KU LEUVEN

ARENBERG DOCTORAL SCHOOL
Faculty of Bioscience Engineering

Methods For Online Sequential Process Improvement

Comparison And Applicability On Contemporary Processes

Koen Rutten

Supervisor:
Prof. dr. ir. J. De Baerdemaeker
Dr. ir. B. De Ketelaere, co-supervisor

Dissertation presented in partial fulfillment of the requirements for the degree of Doctor in Bioscience Engineering

September 2015

Methods For Online Sequential Process Improvement

Comparison And Applicability On Contemporary Processes

Koen RUTTEN

Examination committee:
Prof. dr. ir. C. Courtin, chair
Prof. dr. ir. J. De Baerdemaeker, supervisor
Dr. ir. B. De Ketelaere, co-supervisor
Prof. dr. ir. P. Goos
Prof. dr. ir. L. Pintelon
Prof. dr. ir. M. Vandebroek
Prof. dr. ir. M. Reis
(University of Coimbra, Portugal)

Dissertation presented in partial fulfillment of the requirements for the degree of Doctor in Bioscience Engineering

September 2015

© 2015 KU Leuven – Faculty of Bioscience Engineering
Uitgegeven in eigen beheer, Koen Rutten, Kasteelpark Arenberg 30, bus 2456, 3001 LEUVEN, BELGIUM

Alle rechten voorbehouden. Niets uit deze uitgave mag worden vermenigvuldigd en/of openbaar gemaakt worden door middel van druk, fotokopie, microfilm, elektronisch of op welke andere wijze ook zonder voorafgaande schriftelijke toestemming van de uitgever.

All rights reserved. No part of the publication may be reproduced in any form by print, photoprint, microfilm, electronic or any other means without written permission from the publisher.

Preface - Voorwoord

Deze tekst is het eindresultaat van verschillende jaren noeste arbeid aan de afdeling Mechatronica, Biostatistiek en Sensoren (MeBioS) van de KU Leuven. Geen enkel werk van zo'n omvang kan gerealiseerd worden door een enkeling en velen dragen bij tot het succesvol afronden ervan, zowel rechtstreeks als onrechtstreeks. Graag wil ik dan ook terugblikken op deze mooie (en soms zware) jaren op MeBioS.

I would like to start by thanking the members of the examination committee for the interesting discussions and helpful comments, not only at the time of the preliminary defence but also during the years leading up to the defence. This work has greatly benefited from your input!

Ook wens ik de financierende instantie te bedanken want zonder hun steun en vertrouwen was dit werk niet mogelijk geweest. Dit onderzoek werd gefinancierd met een doctoraatsbeurs van strategisch onderzoek (IWT-SB-091169) en de praktische experimenten in hoofdstuk 8 werden uitgevoerd onder het Prognositics for Optimal Maintenance (POM2, IWT-100031) project, beiden gefinancierd door het Agentschap voor Innovatie door Wetenschap en Technologie (IWT).

Gedurende de jaren dat ik op MeBioS mocht vertoeven zijn er veel collega's geweest en in plaats van hen allen bij naam te noemen, wil ik hen hierbij allen bedanken voor de fijne samenwerking en de ludieke momenten tijdens onder andere de lunchpauzes.

Een aantal collega's en ex-collega's wil ik toch met naam bedanken. Laat ik beginnen met mijn promotor, professor Josse De Baerdemaeker voor de kans, en het vertrouwen, om te mogen doctoreren aan MeBioS. Bart De Ketelaere, mijn co-promotor, die aan het begin van mijn doctoraat nog geen grijze haren had maar er doorheen de afgelopen jaren waarschijnlijk een aantal gekregen heeft: bedankt voor alle moeite die je de afgelopen jaren in mijn doctoraat gestoken hebt op alle gebieden en de grote moeite die je gedaan heb in het verbeteren van deze tekst! Niels Wouters en Bart Lenaerts, bedankt voor de tips in het

gebruik van LaTeX, het is de layout van dit document zeker ten goede gekomen! Professor Paul Darius die zijn strijd tegen kanker verloor en het einde van dit doctoraat niet meer heeft kunnen meemaken: bedankt voor de inspirerende lessen die mijn interesse in de statistiek aangewakkerd hebben en de waardevolle inzichten. En Bram Kamers, eerst co-promotor van mijn masterproef en dan collega: bedankt voor de fijne samenwerking.

Buiten collega's zijn er nog steunpilaren die dit doctoraat mogelijk gemaakt hebben. De vele vrienden en familie die, ieder op hun manier, hun aanmoediging en steun gegeven hebben zijn te ontelbaar om op te noemen. Maar aan de volgende personen wil ik toch graag een persoonlijk woord van dank richten: mijn ouders en zus —Eric, Mariet & Ellen —die me altijd aangemoedigd hebben om door te gaan, hoe moeilijk het ook werd. Mijn grootmoeder, Alice "Liske" Cuypers, die een gedeelte van dit verhaal heeft mee beleefd en vandaag zo trots geweest zou zijn, ik wilde dat je er bij kon zijn om te zien waar dit alles toe geleid heeft! Mijn grootvader, Mathieu Rutten, die ik slechts enkele jaren heb mogen kennen maar die een blijvende indruk op mijn leven nagelaten heeft. En als belangrijkste mijn "partner in crime" Sara Lavrysen: het is niet simpel geweest om de laatste twee jaren met mij samen te leven. Weinig tijd om eens weg te gaan en veel werk 's avonds en in het weekend om alles toch in goede orde af te werken. Dank je voor je niet aflatende steun, het enorme begrip en de goede zorgen (zeker in de vorm van lekker en gezond voedsel wanneer ik weer eens aan het doorwerken was)!

Ongetwijfeld ben ik nog een aantal personen vergeten want in de jaren dat ik aan deze dissertatie gewerkt heb, zijn er vele personen geweest die direct of indirect geholpen hebben. Daarom nogmaals voor eenieder: een woord van welgemeende dank!

Koen Rutten
Augustus 2015

Abstract

In industry, guaranteeing good product quality is essential and requires continuous monitoring and control of the production process. When the quality is not optimal, the process settings need to be changed. Under laboratory conditions this would be performed using an experimental design during which well-chosen combinations of the process settings are imposed so that the optimal settings can be defined. However, this approach is not feasible when searching for optimal settings in the case of a full-scale process since it typically requires exploring the extreme regions of the process where the probability of producing unsaleable product is very high. In order to overcome the drawbacks of classical experimentation, methods for improving full scale processes are investigated in this dissertation. Special attention will be paid to methods which are easy to implement and which are applicable to processes which involve a large number of factors that possibly interact, since this is the situation which is found in contemporary processes.

In the first part of the thesis, a literature survey of methods that are suitable for Online Sequential Process Improvement was conducted. This literature study resulted in two potential candidates for online experimentation, being the Evolutionary Operation (EVOP) and Basic Simplex method. Extensions to the EVOP methodology were developed to allow for an automated implementation along with a novel way to deal with the borders of the experimental domain. Furthermore, a steepest ascent search was combined with the EVOP method. In conjunction with these extensions, a Matlab® software package was created that allows the easy execution of these methods for process improvement in practice. The final part of the methodology research was focussed on presenting a framework for selecting the starting point for these methods should no prior be established by process experience or offline experimentation. Space-filling designs combined with Gaussian Process modelling was shown to achieve a good initial starting point, using a small number of measurements.

In the second part of the dissertation simulation studies were performed

to thoroughly investigate the applicability of the methods on contemporary processes. The EVOP and Simplex methods were compared and this study showed that Simplex is the preferred choice when dealing with deterministic or low-noise systems. The EVOP method proved to be attractive to improve processes characterized by the presence of a substantial amount of noise and/or a dimensionality above three factors. Therefore, the feasibility of a more efficient design—such as a fractional factorial of at least resolution III—was investigated. It was concluded that applying such minimalistic designs offers a significant improvement in the total number of measurements required to reach the optimum and opens up possibilities for the use of EVOP in processes in which fast decisions are required. In the final simulation chapter, first results about the statistical power that is required for an efficient improvement was researched. It was shown that the optimal statistical power increases when the dimensionality increases. However, the exact choice of the power is not too critical and shows a broad, almost flat valley for high dimensions. This allows for much flexibility in the base design for EVOP, depending on the process under study. For processes with a low sampling rate or with a non-stationary behaviour a low power is recommended, whereas a higher power is advised when a high sample rate is possible or the process is stationary.

In the third part, the methodology was validated on a practical case study. Using the developed method, the energy-efficiency of a badminton robot was improved. The problem under study was a minimization of consumed energy subject to a time constraint. A novel approach was presented in which this problem was treated as a multi-objective problem which was then transformed to a single-objective criterion using desirability functions. The energy consumption was reduced by 5% compared to the current implemented energy-efficient solution. Furthermore—by applying more stringent time constraints—the precision of the system could be improved to the maximum precision possible, but with a reduction in energy consumption of 52% compared to the current maximum precision implementation.

In conclusion, this work presents the extensions necessary for the Online Sequential Process Improvement methodology to deal with contemporary processes and shows the potential of using the methodology under practical conditions. Furthermore, a software package was developed that allows for the fast execution of the improvement methods, which was shown on a practical case study.

Beknopte samenvatting

Het garanderen van een kwaliteitsvolle, goede opbrengst is essentieel in de industrie en vereist het continu monitoren en onder controle houden van het productieproces. Wanneer de kwaliteit niet optimaal is, moeten de procesparameters aangepast worden. Op laboschaal wordt dit gedaan door een experimenteel ontwerp uit te voeren waarin slim gekozen combinaties van de procesparameters aangelegd worden zodanig dat de optimale instellingen bepaald worden. Deze aanpak is echter niet valabel indien de optimale parameters bepaald moeten worden in het productieproces zelf aangezien er typisch ook in gebieden van het proces gemeten wordt waar de kans om onverkoopbare producten te produceren zeer groot is. Om deze nadelen van klassiek experimenteren te overwinnen, werden in deze dissertatie methodes voor procesverbetering tijdens het productieproces onderzocht. Aangezien in hedendaagse processen vaak veel instelparameters aanwezig zijn—die mogelijk een invloed hebben op elkaar—werd hier speciale aandacht aan besteed.

In het eerste gedeelte van deze thesis werd een diepgaande literatuurstudie naar methodes die geschikt zijn voor online sequentiële procesVerbetering. Deze studie resulteerde in twee potentiële methodes voor online procesverbetering—met name Evolutionary Operation (EVOP) en Basic Simplex. Samen met een nieuwe manier om te kunnen omgaan met de grenzen van het experimenteel gebied, werden uitbreidingen voor EVOP ontwikkeld om de methode automatisch te kunnen toepassen. Bovendien werd een steepest ascent search gecombineerd met de EVOP methode. Samen met deze uitbreidingen werd een Matlab® software pakket ontwikkeld om de methodes in de praktijk toe te kunnen passen voor procesverbetering. In het laatste gedeelte methodologie onderzoek werd een kader voorgesteld om het startpunt van dergelijke methodes te bepalen indien er geen voorafgaand startpunt bepaald is door proceskennis of offline experimenten. Ruimte-vullende ontwerpen gecombineerd met Gaussiaanse Procesmodellering werden voorgesteld om dit startpunt te bekomen, gebruikmakend van een klein aantal metingen.

In het tweede gedeelte van dit doctoraatsonderoek werden simulatiestudies uitgevoerd om de bruikbaarheid van de methodes grondig te onderzoeken voor hedendaagse processen. De EVOP en Simplex methodes werden vergeleken en deze studie toonde aan dat Simplex de aangewezen keuze is in deterministische of lage-ruis situaties. De EVOP methode is de aangewezen keuze bij processen die gekenmerkt worden door een hoog ruisniveau en/of een dimensionaliteit hoger dan 3 factoren. Echter—voor situaties waarbij een groot aantal factoren in rekening werd gebracht—bleek het aantal experimenten dat nodig is voor het bepalen van de richting van procesverbetering te groot om praktisch haalbaar te zijn. Daarom werd de mogelijkheid onderzocht om een efficiënter ontwerp—zoals het fractioneel factorieel experiment met minstens resolutie III—te gebruiken. Er werd besloten dat dit een duidelijke verbetering gaf in het totaal aantal metingen die nodig zijn om het optimum te bereiken. In de laatste simulatiestudie werden eerste resultaten gepresenteerd over het onderscheidingsvermogen dat benodigd is voor een efficiënte procesverbetering. Hier werd aangetoond dat het optimale onderscheidingsvermogen stijgt als de dimensionaliteit stijgt. De keuze van het onderscheidingsvermogen is echter niet al te kritisch aangezien de functie een brede, bijna vlakke vallei toont voor hoge dimensionaliteiten. Dit staat toe om het onderscheidingsvermogen te bepalen afhankelijk van het type proces; voor processen met een lage bemonstering of niet-stationair gedrag werd een laag onderscheidingsvermogen aangeraden, terwijl een hoger onderscheidingsvermogen geadviseerd werd indien een snelle bemonstering mogelijk is of als het proces stationair is.

In het derde deel werd de methodologie gevalideerd met een praktische casestudy. De energie-efficiëntie van een badminton robot werd verbeterd gebruikmakend van de ontwikkelde methode. Het gestelde probleem was een minimalisatie van het energieverbruik onderhevig aan een tijdsbeperking. Een nieuwe aanpak werd voorgesteld waarin dit werd bekeken als een multi-objectief probleem dat dan getransformeerd werd tot een enkel-objectief criterium gebruikmakend van desirability functies. Het energieverbruik werd verminderd met 5% vergeleken met de huidig geïmplementeerde energie-efficiënte oplossing. Voorts werd de precisie van het systeem verbeterd—door gebruikmaking van striktere tijdsbeperkingen—tot de maximum precisie maar met een reductie in het energieverbruik van 52% vergeleken met de huidige maximum precisie implementatie.

Samengevat presenteert dit werk de uitbreidingen noodzakelijk om online sequentiële procesverbetering toepasbaar te maken op hedendaagse processen en toont het het gebruikspotentieel van deze methodes in praktische omstandigheden. Voorts werd er een software pakket ontwikkeld dat het eenvoudig gebruik van de procesverbeteringsmethodes toelaat, wat werd aangetoond met een praktische casestudy.

Abbreviations

2D	Two-dimensional
3D	Three-dimensional
ANOVA	Analysis Of Variance
BLUP	Best Linear Unbiased Predictor
CCD	Central Composite Design
DACE	Design And Analysis of Computer Experiments
DOE	Design Of Experiments
EGO	Efficient Global Optimization
EOMPC	Energy-Optimal Model Predictive Control
EVOP	Evolutionary Operation
EVOPFD	EVOP Factorial-Design Technique
EVOPSA	Evolutionary Operation Steepest Ascent
FFT	Fast Fourier Transform
FMTC	Flanders Mechatronics & Technology Centre
GA	Genetic Algorithm
GP	Gaussian Process
i.i.d.	independent and identically distributed
IQR	InterQuartile Range
LHC	Latin HyperCube
LoF	Lack of Fit

MPC	Model Predictive Control
MSSD	Mean Square of the Successive Differences
OFAT	One Factor At a Time analysis
OLS	Ordinary Least Squares
OOP	Object Oriented Programming
OSPI	Online Sequential Process Improvement
PEOS	Proximate Energy-Optimal Servo
PTOS	Proximate Time-Optimal Servo
REVOP	Random Evolutionary Operation
ROVOP	Rotating Square Evolutionary Operation
RSM	Response Surface Methodology
SD	Standard Deviation
SE	Standard Error
SFS	Single Factor Search, also called OFAT
SNR	Signal-to-Noise Ratio
TEVOP	Traditional Evolutionary Operation
TOMPC	Time-Optimal Model Predictive Control

Definitions

Coded regression coefficient	The coefficient in a regression model if the factors are coded (in the case of this dissertation between $[-1; 1]$).
Contemporary process	A process with a high degree of complexity sampled with high frequency and a large number of factors that possibly interact.
Cycle	One execution of a statistical design within an EVOP *phase*.
Design region	The region bounded by the minimum and maximum factor levels of the small perturbations in the current improvement step.
Experimental domain	The domain defined by the factor limits imposed by the process (in the case of min-max boundaries a hypercube defined by these factor limits).
Factor	A measurable (and, in the context of this dissertation, controllable) quantity, usually used in the context of the independent variables of an experimental design.
Factor level	A specific setting of a factor.
Factorstep	The distance between the minimum and maximum factor levels in the design region for every factor (determines the size of the *design region*).

In silico	Research or experiments conducted or produced by means of computer modelling or computer simulation.
Improvement methods	Methods that gradually improve the process but not necessarily find the global optimum.
Online improvement	Experimentation is performed while the process or machine is in standard operation.
Phase	A phase in EVOP consists out of a number of *cycles*. A phase is concluded when a decision is taken in which direction to move. (e.g. replicating a design is executing several cycles, while the phase would consist out of all measurements).
Reference condition	The current best known combination of factor levels (at the start of the improvement, later on the centre of the design region).
Simplex	The Simplex improvement method.
simplex	A geometric figure defined by $k+1$ points, for k dimensions (factors).

List Of Symbols

$\mathbf{1}_n$	An $n \times 1$-vector of ones
A_{391}	Spectral amplitude of the bearing setup at 391 Hz
$\hat{a}$	Acceleration setpoint for badminton robot controller
α	The significance level of a statistical test
$\boldsymbol{\beta}$	The parameter vector in a model
β_d	Regression coefficient of the linear term for the d-th factor in a regression model
$\beta_{d,c}$	A coded linear regression coefficient for the d-th main effect
β_{dd}	Regression coefficient for the pure quadratic of the d-th factor in a regression model
$\beta_{dd'}$	Regression coefficient for the interaction term between the d-th and d'-th factor in a regression model
$\hat{\boldsymbol{\beta}}$	The ordinary least squares estimator of the parameter vector in a regression model
β_i	Regression coefficient (in the Gaussian Process model)
β_0	Intercept in a regression model
c_r	The number of cycles in a phase (or replications in classical design)
D	Desirability index, the combination of the individual desirabilities using the geometric mean
D_c	The global desirability index for the badminton robot responses using the c-th time restriction

d_E	desirability transform of energy
d_{ED}	The Euclidean distance between the true and estimated optima of a simulation model
$d_{ED,GP}$	The Euclidean distance between the true and an estimated optima for a GP model
$d_{ED,OLS}$	The Euclidean distance between the true and an estimated optima for a second-order model with a space-filling design
$d_{ED,RSM}$	The Euclidean distance between the true and an estimated optima for a second-order model with a classical design
$\mathcal{D}$-efficiency	The $\mathcal{D}$-efficiency of a design
δ	EVOPSA step size for the badminton robot improvement
δ_{EVOP}	The step size amplitude for an EVOP improvement
$\boldsymbol{\delta}_{\mathbf{EVOP}}$	The step size vector for an EVOP or EVOPSA improvement
$\delta_{EVOP,c,d}$	The coded steepest ascent delta in the d-th direction
df_ε	Degrees of freedom for error
d_d	desirability of the d-th response variable
D'_c	The transformed global desirability index for the badminton robot responses using the c-th time restriction
$d_{t,c}$	desirability transform of time for the c-th time constraint
dx_a	Acceleration factorstep (badminton robot improvement)
dx_d	The factorstep in the d-th dimension
dx_v	Speed factorstep (badminton robot improvement)
E	Energy consumption of the badminton robot
ε	The random error term in the simulation models used
f_a	The number of main effects in the reduced linear model after stepwise regression
$F_{ax,dyn}$	The peak force generated by the sinusoidal dynamic axial load under static load (of the bearing setup)

$f_{ax,dyn}$	The frequency of the sinusoidal dynamic axial load (of the bearing setup)
$F_{ax,dyn,set}$	The peak setting for the sinusoidal dynamic axial load (of the bearing setup)
$F_{ax,st}$	The static axial load (of the bearing setup)
$F_{rad,st}$	The static radial load (of the bearing setup)
f_t	The total number of parameters in a regression model, excluding the intercept
γ	The nugget parameter in a Gaussian Process Model
h_i	Known fixed functions in the Kriging model
$\mathbf{I}$	The $n \times n$ identity matrix
k	The number of factors (dimensions) in a design
l	The number of levels for the factors in a design
M	The number of succesive reflections in Dynamic Simplex
$\mu(\cdot)$	The mean of a quantity
N	The total number of measurements during experimentation or simulation
n	The number of measurements in a Latin Hypercube Design
n_c	The number of times the reference condition is measured (or centerpoint in classical design)
n_T	The total number of observations in one design (phase)
ν	Degrees of freedom of a distribution
p_{enter}	The p-value to check in the stepwise regression procedure whether to enter a term into the model
p_f	Denotes the $1/2^{p_f}$ fraction of a full factorial
ϕ	The noncentrality parameter of the noncentral t-distribution
π	The power to detect a statistical effect or regression coefficient
$\mathbf{\Pi}$	An $n \times k$ matrix whose columns are different randomly selected point permutations of the intervals $\{1, \ldots, n\}$, used in the construction of Latin Hypercube Designs

$\pi_{\mathcal{D}}$	The statistical power value of a $\mathcal{D}$-optimal design (calculated after design construction)
π_{orth}	The statistical power value, specifically for an orthogonal design (see chapter 7)
π_{req}	The statistical power value as a setting of the simulation model in chapter 7
p_{remove}	The p-value to check in the stepwise regression procedure whether to remove a term from the model
$\mathbf{R}$	An $n \times n$ spatial correlation matrix in Gaussian Process modelling
$\mathbf{r}$	An $n \times 1$ vector of estimated correlations of the unobserved responses $\hat{y}(\mathbf{x}_*)$ at new coordinates $\mathbf{x}_*$ based on the observed data points $\mathbf{y}(\mathbf{X})$
$\mathbf{R}_\gamma$	A spatial correlation matrix in Gaussian Process modelling with a nugget parameter
r	The number of replications of the factor levels in a design
$\mathbf{S}$	The normalized design space in Kriging
σ	The error (noise) standard deviation
$\sigma_{\beta_{d,c}}$	The estimated parameter variance of parameter $\beta_{d,c}$
σ^2	Process variance
t	t-distribution (in chapter 6)
t	Time (in chapter 8)
T_{NC}	Non-central, t-distribution with non-centrality parameter ϕ
t_c	The c-th restriction on the arrival time of the badminton robot
$\boldsymbol{\theta}$	A parameter in the spatial correlation matrix of a Gaussian Process model
t_{max}	Arrival time of the badminton robot
$\boldsymbol{\varepsilon}$	The $n \times 1$ vector of random error terms
$\hat{v}$	Speed setpoint for badminton robot controller
v_{set}	Constant rotation speed of the bearing setup

$\mathbf{X}$	The $n \times k$-dimensional matrix of input values or design points
$\mathbf{x_c}$	The $1 \times k$ coordinates for the centre of a design region
$\mathbf{x_{co}}$	The initial corner point for the Corner and Tilted methods to construct the first simplex
x_d	The value of the factor level to be coded
$x_{d,c}$	A coded factor level value
x_d^-	The minimum value of the factor level in the design region
x_d^+	The maximum value of the factor level in the design region
$\mathbf{X_{lim}}$	A $k \times 2$ matrix with the lower and upper bounds for every factor
$\mathbf{x_{lim}^-}$	A $k \times 1$ vector containing the lower bounds for every factor
$\mathbf{x_{lim}^+}$	A $k \times 1$ vector containing the upper bounds for every factor
$\mathbf{x}_*$	A set of unobserved coordinates in Gaussian Process modelling, i.e. a coordinate set that was not included in the design
$\mathbf{x_{start}}$	The reference coordinates for an improvement
$\mathbf{X_0}$	The design matrix of the initial design in an EVOP improvement
$\mathbf{y}$	The response vector in a regression model
y	The response variable in a regression model
y_{act}	Actual badminton robot position
$\bar{y}$	The estimate for the average of the response variable
$\bar{y}_{\sigma=0}$	The average response when no noise is present
$y_{ci,\sigma=0}$	The median noise-free response $y_{ci,\sigma=0}$ in the centre of the i-th phase for EVOP simulations
y_{des}	Interception reference of the badminton robot (distance to travel)
$\hat{y}(\mathbf{x}_*)$	The estimated response at unobserved coordinates $\mathbf{x}_*$
y_i	The i-th observation of the response variable

y_{i+1}	The $i+1$-th observation of the response variable
$y_{i,\sigma=0}$	The i-th noise-free response
$\mathcal{Z}$	Gaussian Random Field (in Kriging)

Contents

List Of Figures

List Of Tables

List Of Algorithms

1 Introduction

1.1 Offline Process Improvement In Industry

When a new machine or process is implemented, experimentation is often required to choose the settings (called factor levels) so that the output quality is maximized. Since real-life processes mostly involve multiple factors that possibly interact, a well-defined strategy needs to be followed. This is the topic of Design Of Experiments (DOE), and a wide range of design types have been proposed in literature [100, 104, 107, 112, 159] and combining such a design with an appropriate regression model leads to a comprehensive optimization strategy.

A widely used method for offline experimentation and optimization within DOE is Response Surface Methodology (RSM) [1, 56, 93, 105, 107]. Typically, RSM asks for substantial changes in the factor levels to determine the most influential factors and to pinpoint the optimal region. Because of those substantial changes, several experimental measurements will result in unacceptable output quality which is not desired, especially in full scale production processes. As a result, RSM is typically an offline, research-oriented approach that is applied on lab scale processes, and factor levels labelled as optimal for the lab scale process are then used on the full scale process.

Another approach to offline optimization is performing it in silico. Nowadays, specialized modelling software as well as strong computational power are available so that often accurate computer models can be built for a given process, either by first principle models (a first principles physics model is one that seeks to calculate a physical quantity starting directly from established laws of physics) or by more advanced techniques such as finite [8, 28] or discrete [85, 87, 150] element modelling. Here the approach also consists of two phases: First, there is an experimental design phase for defining at which factor level combinations to perform the simulations. This phase is then followed by an

analysis phase to find the optimum. This approach is labelled Design And Analysis of Computer Experiments (DACE). Because of its specific nature—the deterministic nature of the simulations and the high complexity of the response surface—it uses specialized designs (i.e. space-filling designs) as well as analysis methods (e.g. Genetic Algorithms [27, 28, 68, 114], Artificial Neural Networks [5, 55, 101] or Kriging methods [8, 165]). The optimal settings determined from the analysis step are then applied to the full scale process.

Most often, however, there is a discrepancy when upscaling from the lab to the full scale process. Figure 1.1 shows an example of such a (hypothetical) upscaling issue in which a shift and distortion of the process contours manifest when shifting from laboratory (or pilot scale) to plant scale operation. It is clear that factor levels found optimal in the lab scale are not optimal on the full (plant) scale and further improvement on the plant scale is desirable.

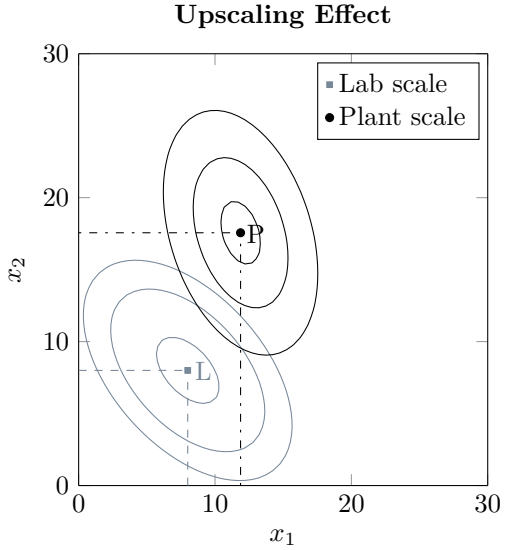


Figure 1.1: Possible appearance of process contour surfaces for a process conducted on the laboratory scale (L) and on the plant scale (P).

Applying RSM on the full scale process, however, is not desirable. As mentioned above, classical RSM requires substantial changes to be invoked in the factor levels with the risk of producing unsaleable product. Besides, in RSM it is generally assumed that the optimum of the process is stationary, i.e. located at a fixed point in the input space. In reality this assumption often fails due to warming up and cooling down of equipment, changes in observable but

uncontrollable factors such as environmental conditions, machine wear, input variability commonly encountered in the agro-food industry, sensor drift, etc.

Also, when experimenting in silico, it is usually not possible to capture all variability of the physical process correctly. This results in a plant-model mismatch and the estimated optimal factor levels in such a case can be further improved by experimentation on the physical process.

Due to the aforementioned reasons of upscaling effects, non-stationary optima and plant-model mismatch, it is evident that a strategy should be implemented to further improve the process and find or track the true optimum by online experimentation on the full scale process.

1.2 Online Process Improvement

To efficiently deal with the aforementioned disadvantages of offline optimization a shift to online, full scale experimentation is advised after offline methodologies have been used to determine the region of the optimum. As one does not wish to impede production or standard operation, the term online in this aspect denotes that experimentation is performed while the full scale process or machine is running in standard operation, i.e. during regular production. Whereas offline methods of process improvement have been reported for many years and in many applications, it is surprising to see that not many methods are described that can truly be classified as online. Before describing these methods, the conditions for online methods have to be defined.

Since online methods have to be applied while the production process is running it necessitates that the optimization may not lead to interruption of the process, or to unacceptable output quality. This can be achieved by introducing small shifts in the factor levels around the current best known combination of factor levels—also called the *reference condition*—analysing the results, and defining new factor levels close to the old ones. With this approach, the process is gradually shifted towards a better output and the risk of producing unacceptable output is minimized due to the small changes in the factor levels. In most of the methods currently proposed for online improvement, the analysis step itself is usually straightforward because of the small perturbations, as the relation between the output and the input (factors) can be well approximated using a very simple model (such as a polynomial model) in this small region.

The region in which these small shifts in the factor levels are introduced is called the *design region*—the region bounded by the factor levels of the small perturbations around the current reference condition. It is a small fraction of

the *experimental domain*—the domain defined by the factor limits imposed by the process (in the case of min-max boundaries a hypercube defined by these factor limits). Although a global optimization is preferred (i.e. find the global optimum of the process), the restrictions imposed upon the optimization (only small perturbations around the reference condition are allowed) induce a gradual shift of the operating conditions towards a higher quality output. If the current reference condition is close to a local optimum, the methods following this philosophy are prone to become stuck in this local optimum. However, the goal of improving the process is still achieved. As such, online improvement is in no way a substitute for offline experimentation but the logical next step. Offline experimentation efforts should focus on pin-pointing the region of the global optimum after which online methods can be used to fine-tune the current reference condition. The online methods presented under this philosophy shall be called *improvement* methods as they gradually *improve* the process but not necessarily find the global optimum.

There is a marked difference in the typical settings when working online on full scale processes as compared to the previously discussed offline methods. In table 1.1 the typical settings for *offlin*e and *online* methods are summarized.

Table 1.1: Different settings for offline and online experimentation.

	Offline methods	**Online methods**
Inline experimentation	Production usually has to be halted.	Production can continue.
Number of experimental measurements	Should be kept to a strict minimum.	Typically not limited, i.e. performed during production.
Region in which to experiment	Large in order to find relevant factors and optimal factor levels (often as large as the experimental domain).	Small to avoid the production of unsaleable product (the design region is a fraction of the experimental domain).
Stationarity	Generally required.	Not required.
Condition of final product	High probability that some of the experimental output is unsaleable.	Low probability that some of the experimental output is unsaleable.
Global/Local optimum	Can find global optimum, depending on technique.	Can become stuck in local optimum close to the initial reference condition.

Most traditional methods described for offline optimization are not suitable for use in online optimization. Of the previously described methods only Genetic Algorithms (GA) have been used online. However, to adhere to the principles for online improvement many adjustments have to be made to GA that are specific to the process under study. Furthermore, GA requires a large initial population of measurements and it might not be guaranteed that all the initial measurements are within specifications (i.e. the end product is saleable) which violates one of the restrictions in table 1.1. An example where GAs are used in an online industrial setting is published by Yüzgeç *et al.* [163] where biomass concentration has to be maximized and ethanol formation minimized in an industrial fed-batch yeast fermentation process. However, without an appropriate fitness function (mathematical model) the application would have failed. This leads to the conclusion that the use of GAs in an online context requires much (process-specific) experience to perform.

An approach that fully adheres to the principles set forth in table 1.1, was proposed almost 60 years ago. Based on the principles of RSM, George Box introduced in the 1950's [19, 20] the concept of Evolutionary Operation (EVOP). EVOP has been successfully applied to full scale production processes [69, 83] and laboratory processes [11, 108, 151] and due to its simplicity and ease of operation is a strong candidate for use in an online setting.

The denominator Evolutionary Operation has been used in literature both to describe the original method by Box and all derived methods, as well as the philosophy behind the approach. A clear distinction will be made in this text to avoid confusion and the following terminology will be used: the term Traditional EVOP (TEVOP) will be used for the original method by Box, derived methods will be referenced by their proper name when they are introduced and the philosophy driving the methodology will be referred to as "Online Sequential Process Improvement" (OSPI). The denominator EVOP itself will be used to refer to the automated software algorithm described in chapter 3.

The basic philosophy behind OSPI is that "it is nearly always inefficient to run an industrial process to produce product alone. A process should be run so as to generate product *plus information on how to improve the product*" [19]. This differs from routine operation were the production process is operating at the best conditions of operation known at that time (which only fulfils the requirement of generating product). To generate additional information on how to improve the product (or output), changes in the factor levels have to be applied and the subsequent change in response (product quality) observed, as detailed at the start of this section.

The OSPI methodology was introduced to deal with problems in the chemical industry where plant processes are usually fine-tuned after considerable

experimentation on a smaller scale, be it on a pilot plant or on lab scale (which was exemplified in figure 1.1). However, the optimal operating conditions on the small scale are usually only a good approximation of the full scale optimal conditions. These *upscaling* effects might occur due to the use of different equipment, changes in the environmental conditions, modification of the process to work at plant scale and so on. A shift in the optimal operating conditions can also come into play in discontinuous processes due to a change of input materials such as in batch-to-batch processes. OSPI can be started when such a change comes in effect to find the shifted optimal conditions. Another application of OSPI lies in the tracking of an optimum when the response drifts in time. Such a dynamic drift can be attributed to, for instance, machine wear or slowly changing environmental conditions. If the drift in time is slower than the capability of the OSPI method to react, the optimum can be tracked.

In the two decades after its publication OSPI methods were used moderately but with great success in industry. During this time, the main limitation of the methodology was the absence of computational power and sophisticated sensor technology to fully automate its workings. Most applications in these decades were manually implemented and therefore also related to low dimensional applications (i.e. two or three factors under investigation). In the contemporary setting however, the computational power and sensor technology is available to fully automate the procedures yet improvement problems often deal with a higher dimensionality than the three factors that were described in the traditional approach.

Although OSPI methods have been around for over six decades, not much research has been devoted to their implementation in modern production environments, especially related to their automation and the effect of the increased dimensionality and complexity of modern machines. Neither has in-depth research been devoted to the comparison of the classical OSPI methods. It is the goal of this dissertation to adjust the traditional OSPI methods for automatic use and compare them in-depth to validate their use in modern, contemporary processes. Furthermore, the increased dimensionality of contemporary problems warrants a detailed investigation into the use of sparse designs in OSPI implementations where the number of experiments per improvement step is the prohibitive factor to implement them on processes.

Indeed, much research has been conducted on experimental designs and models but has been mainly focussed on lab scale or offline applications. Surprisingly, very limited research has been devoted to online improvement of processes even though this is the industry default. Table 1.2 summarizes a topic search on the Web of Knowledge for publications ranging from 2000 to 2015. When looking at the keywords for EVOP only 104 publications are found and these contain a mix of all OSPI methods (both EVOP and all its derivatives) as well as

some publications in the field of numerical optimization which also refer to the original EVOP paper as the fundamental concept for Evolutionary Strategies. For Simplex, a derivative of EVOP, many publications are found. However, the keyword Simplex is not enough to restrict the search to OSPI methods. The keyword simplex is also used in numerical optimization and in classical experimental design (for instance when an experiment is performed using the simplex-lattice design). A more strict search "sequential simplex" only offers 53 publications which also include applications that do not adhere to the OSPI requirements set forth here (as will be discussed in the next chapter). When searching for publications about "online optimization", one obtains more results, however many of these results are not related to OSPI. The denominator online optimization is used extensively when real process evaluations are used, such as in the tuning of sensors and digital filters. Remark that the distinction made in this work between offline optimization and online improvement is generally not made in literature, which leads to confusion as to the exact nature and goal of the performed experimentation effort.

Table 1.2: Search of Research Topics on Web Of Knowledge for publications from 2000 to 2014.

Topic	Search results (2000-2014)
"Response Surface Methodology"	37,135
"On-line optimization" OR "Online optimization"	1,461
"EVOP" OR "Evolutionary Operation"	107
"Simplex"	80,211
"Sequential Simplex"	53

Search of the Web of Knowledge, http://www.webofknowledge.com, search conducted at 24 April 2015. Approximate search results as returned by this search engine.
Both "optimization" and "optimisation" were used in the search.

This search clearly shows that research devoted to OSPI methods is very limited. Traditionally sensor technology and computational power were very limited and manual computation and measurement were the OSPI standard. Yet during the last decades, there is an ever increasing trend towards the implementation of fast, non-destructive and accurate sensors for monitoring the quality of a broad range of industrial processes such as, (near) infrared spectra that are taken inline to monitor chemical processes [39] or the use of vibration sensors for monitoring (production) processes [147]. The tremendous increase in computational power has opened up the possibility to process and manipulate large data streams in very limited time. Having available such process data at any time, combined with large computational power, opens the door towards automated, online adaptation and improvement of a contemporary process.

1.3 Objectives And Outline Of The Dissertation

This dissertation aims to put forth the fundamentals to deal with the increased dimensionality of contemporary problems and the fully automated implementation of the OSPI methodology.

The dissertation is divided in three parts: *Methodology*, *Simulation Studies* and *Case Study*. The *Methodology* part, which consists of three chapters, introduces the methodology necessary for the research study. The *Simulation Studies* part, also consisting of three chapters, focusses on simulation studies that allow for an in-depth comparison of several settings that are difficult to replicate on one physical process. The *Case Study* part, consisting of the final chapter, presents a case study on a practical setup which aims to treat a constrained problem by interpreting it as a multi-variate problem and using desirability functions to transform it to a uni-variate problem.

1.3.1 A Review Of Methods For Online Sequential Process Improvement

The two basic OSPI methods used in this text, EVOP and Simplex, are introduced and explained in detail in chapter 2. A review of pertinent applications and state-of-the-art research is presented. Existing derivatives of these two methods are briefly discussed and it is explained why these derivatives are not used in this dissertation. It is shown that for Traditional EVOP no clear-cut rules for the direction in which to move in every improvement step are provided, which is a problem that needs to be addressed for automated implementations.

1.3.2 Implementation Of And Extensions To Online Sequential Process Improvement For Evolutionary Operation

The Traditional EVOP implementation has no formal rules for the direction in which to move nor for the size of the move, as has been elucidated in chapter 2. To automate the EVOP procedure, calculations for both directionality and size of the move are formalized and presented.

An OSPI method which was mentioned but never explored in previous publications is also presented in this chapter, named Evolutionary Operation Steepest Ascent (EVOPSA) which combines the statistical part of EVOP and

the limited sample size in subsequent steps of Simplex (these traits of EVOP and Simplex are explained in chapter 2).

Additionally, a recommendation is made on how to deal with the borders of the experimental domain in an automated EVOP or EVOPSA procedure which has not been described before since the methods have mainly been used manually.

1.3.3 Choosing An Appropriate Starting Point When No Prior Information Is Available

The OSPI methods start from the currently best known operating settings, or *reference condition*. However, such reference condition is not always available. In such a case, offline experimentation has to be performed to locate the region in which the optimal settings are located. Classical designs used in RSM are not always ideal candidates since their use is limited to fitting simple polynomial models and they might not capture the global optimum in the presence of local optima and high non-linearity of the process. In this chapter, space-filling designs and Gaussian Process (GP) modelling—which are used in conjunction for fitting complex models that stretch beyond polynomials—are proposed as a flexible framework for fitting a broad class of functions.

The goal of the process model is not to model the process in great detail (i.e. achieve accurate prediction), but rather to assess with a minimal effort the factor levels to start from with an OSPI improvement. This methodology is applied to a simulation study to determine its appropriateness and is compared with executing classical RSM and Ordinary Least Squares (OLS) regression on the same simulation study. Afterwards, the GP methodology is applied to a practical case and the estimated optimal settings are used as the reference condition for an EVOP improvement to fine-tune the settings.

1.3.4 A Comparison Of Evolutionary Operation And Simplex For Process Methods

To the author's best knowledge, no in-depth comparison between the two traditional OSPI methods has ever been made on the same process. Nor has formally been tested which OSPI settings and level of noise in a process will favour the use of which method. In this chapter EVOP and Simplex are compared on a simulation study with varying settings, being: the dimensionality of the problem, the size of the design region and the level of noise present in the simulation. To compare the methods in their most basic form and eliminate any adjustments that might skew the comparison, no replication of any sort is used

in the measurements and the simulation results are presented and compared in detail.

1.3.5 Efficient Designs For Evolutionary Operation

It is shown in chapter 5 that EVOP performs better than Simplex when the dimensionality and the noise-level increase. However, the traditionally used full factorial design in EVOP becomes more and more prohibitive in sample size as the dimensionality goes up. To minimize the sample size and ensure the feasibility of EVOP for a high dimensionality, other designs should be used within the EVOP framework. This chapter explores the use of classical designs with a low sample size to determine the impact of the lower sample size on the performance of the EVOP method. A comparison is made with the traditional full factorial for the simulations performed in the previous chapter and additional simulations are executed for higher dimensionalities. It is shown that it is certainly feasible to use efficient designs but that one has to bear in mind the efficiency of the improvement.

1.3.6 Optimal Statistical Power For Evolutionary Operation

The discussion which concludes chapter 6 indicates that finding a good value for the statistical power is not trivial. The previous simulation results indicate that the statistical power for OSPI can be taken lower than in classical experimental design. The question arises how much lower the power value can be compared to classical experimentation to achieve satisfactory results. This is a topic that will require study and will really advance the usability of EVOP with a basis for sample size estimations for sequential experimentation. This chapter will provide the basis for further study into this domain by presenting simulation results on a linear model for varying dimensionality and requested powers.

1.3.7 Constrained Online Improvement Using Evolutionary Operation Steepest Ascent: A Case Study About Energy-Optimal Robot Control

In many contemporary processes the quality of output is often of a multi-variate nature and the combination of several responses define the overall quality. The use of desirability functions can combine these multiple characteristics into a global index that can be used for improvement. This chapter shows the use of

Derringer Desirability functions in the novel context of constrained improvement on a physical system. The EVOPSA method is used to minimize the energy-consumption of a badminton robot while motion time is kept within the required constraints.

Part I

Methodology

2 A Review Of Methods For Online Sequential Process Improvement

2.1 Introduction

As stated in the introduction, contemporary processes are typically characterized by a high degree of complexity—they are sampled with high frequency and involve a large number of factors that possibly interact—and are therefore difficult to model. This complexity has a direct implication on the optimization as it might be very tedious or even impossible to develop a model that adequately describes process behaviour over the whole input space. When a model-based framework is difficult to develop, other techniques should be used in order to improve such processes in an online way. Furthermore, due to the problems of upscaling, non-stationary optima and plant-model mismatch; it is preferred to switch to an online approach to fine-tune the factor levels.

Online Sequential Process Improvement (OSPI) is a general philosophy that is applicable for such situations. OSPI uses sequential experimentation, during which a new experimental run is based on the outcome of previous measurements; this in contrast with classical experimentation in the framework of Design Of Experiments (DOE) where the experimental plan is developed in one phase.

The field of sequential improvement is mainly covered by two basic approaches: one based on statistics developed by Box, and a completely heuristic approach developed by Spendley *et al.*. The approach proposed by Box (and its derivatives) is labelled Evolutionary Operation (EVOP), and the heuristic approach will be referred to as Simplex. This last method is further divided into the basic Simplex as proposed by Spendley *et al.* [143], the Nelder-Mead or Variable Simplex [109] and the Super Modified Simplex [125].

In the nearly sixty years since the philosophy was introduced a large amount of methodological and application-oriented papers were published. This chapter

provides an overview of the OSPI philosophy, the main methods used and their applications. The applications are ordered into several categories to cluster those that are related in some way and make the material more presentable. In 1966, Hunter & Kittrell [71] provided an excellent review of literature of traditional EVOP, Simplex and industrial applications in which companies implemented these methodologies with great profitability. This text will not rehash all these references but will focus on papers published after this review, except when the references are important for the methodology such as the original articles in which the methods were proposed.

This chapter is structured as follows: one separate section will be devoted to Evolutionary Operation and another to Simplex. Both sections will be further divided into an explanation of the method and related derivatives, an overview of applications and avenues for further research. It is concluded by summarizing pertinent research questions that have not been answered in literature, and that motivated the research topic of this dissertation.

2.2 Evolutionary Operation

The term EVOP is used in literature to refer to both the OSPI metholodogy, the EVOP method as described here and its derivatives (and also to refer to Simplex). To make a clear distinction between the several EVOP methods, it was opted to use TEVOP for the traditional method and EVOP from chapter 3 onwards for the adapted, automated scheme presented in this dissertation.

2.2.1 Manual Implementation

Traditional EVOP By Box

The original method proposed by George Box will be labelled Traditional EVOP (TEVOP) to distinguish it from other implementations that will be discussed later on. Evolutionary Operation was originally developed to be used manually by plant personnel to improve a working process. In his original 1957 paper [19], Box introduces this concept and in a later article [22], Box & Hunter detail the simplified calculations and worksheets for plant personnel. All relevant articles about the methodology that have appeared by the hand of Box have been bundled in a book, originally published in 1969, and republished in 1998 [20].

The methodology is based on the analogy with evolutionary processes in nature where one has genetic variability in offspring and where the unfavourable variants of the offspring disappear due to natural selection. In TEVOP well-chosen but small perturbations to the factor levels of the process around the current best known factor levels, or reference condition, are introduced (in analogy with genetic variability). The perturbations are defined based on the concepts of Design Of Experiments, where the reference condition is taken as the centerpoint of a design defined in a small design region. Traditionally, a two-level full factorial design (2-factor example of such a design is given in figure 2.1) is chosen as a basis, having 2^k experimental measurements, with k the number of factors included in the study. Optionally, the reference condition (point 1 in figure 2.1) can be measured as well, which coincides with the centerpoint of a factorial design. The number of measurements is then $2^k + n_c$ with n_c the number of times the reference condition is measured. More information about factorial designs can be found in the following chapter.

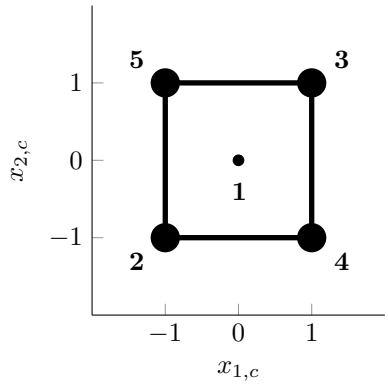


Figure 2.1: Design points for a two-factor full factorial with centerpoint included.

Introducing these small changes allows retrieving information about the direction in which favourable improvement is expected. When this information is obtained, a move is made in this favourable direction, after which the process of measuring small perturbations is repeated. The TEVOP methodology consists of four steps, shown in the flowchart in figure 2.2, which will be discussed in detail.

Design: Factors should be perturbed slightly so that their effect can be estimated but with the restriction that the output (e.g. produced product) is still within specification. In TEVOP, small factor level changes are introduced around the current best known conditions—the reference condition—to gather information about the process. These changes in the factors should be kept small to ensure that no output is produced that falls outside of process specifications.

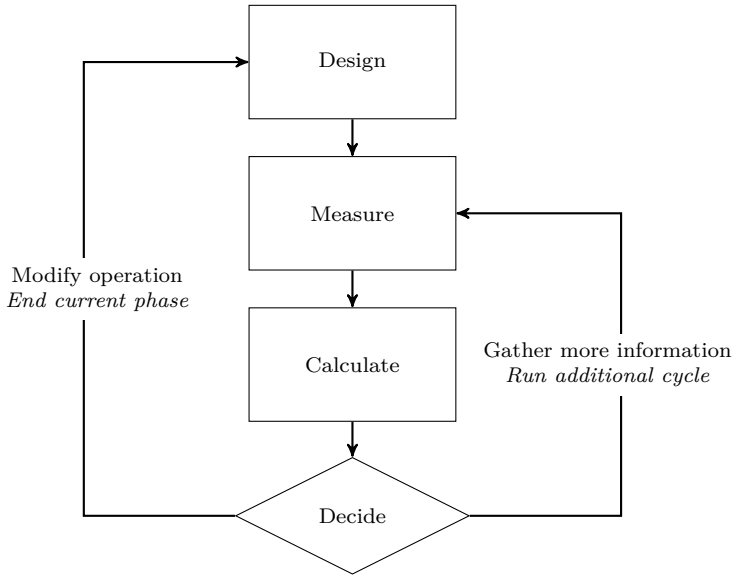


Figure 2.2: Flowchart TEVOP.

If the experimental domain consists of the entire factor space (i.e. all possible factor combinations) then TEVOP will introduce only very small factor level changes—which are a subset of the experimental domain—around the reference condition. These factor changes should be introduced in such a way that meaningful conclusions can be drawn. This is done using a statistical design, which will allow making the decisions using a statistical model. Any design can be used with TEVOP but Box proposed the use of the two-level full factorial design as it is easy to construct and analyse. For two factors, the design including one centerpoint consists of five measurements which are presented in figure 2.1.

Measure: In this second step, the responses at the design points are measured.

Calculate: After measurements are collected, a simple regression model is fitted to the data. All possible effects (main and possibly two-way interactions) are estimated and—if centerpoints are included—a lack-of-fit test to check for curvature is calculated.

Decide: After all effects have been calculated, their statistical significance is checked. If an effect is significant, it will be considered when taking a decision; if it is not significant, it is not taken into account. However—since only small factor changes are allowed—it is possible that no effect is found significant due

to the noise present in the process. The experimenter has to decide whether there truly is no effect active (i.e. the process cannot be improved) or that more information has to be gathered to determine if an effect is significant. This leads to two main decisions that can be made:

1. *Gather more information*: By re-measuring the design and adding the newly collected data to the previous data more information is available to determine whether effects are significant. Every time a design is measured (without changing the reference condition and thus the location of the design) is called a *cycle*. This can be equated to the replication of a design in classical DOE.
2. *Modify operation*: If the experimenter decides that enough information is gathered to determine which effects are significant (or not significant) a decision can be made on how to modify the operation. When the experimenter decides that enough experimental measurements have been measured to modify operation a *phase* of TEVOP is concluded and a new phase started. A *phase* consists of $c_r(2^k + n_c)$ experimental measurements, with c_r the number of *cycles* ran. In general, the following decisions could be made at the conclusion of a phase:
    a. Some effects are significant:
        i. Adopt one of the design points as the new reference condition and commence a new phase about this point. An example of this decision is presented in figure 2.3a;
        ii. Explore an indicated favourable direction of advance (by, for instance, a line search) and commence a new phase around the best conditions found in this exploration. An example is presented in figure 2.3b but other decisions are also possible and it is up to the experimenter to decide on the direction. This type of decision is also the basis for the extension Evolutionary Operation Steepest Ascent which is proposed in the next chapter.
    b. No effects are significant:
        i. Change the design region to one in which the factor levels are more widely spaced (thus strengthening the signal). For an example see figure 2.3c;
        ii. Additional cycles can be run to increase the probability to detect an effect;
        iii. Substitute new factors for one or more of the old factors (if one of the factors and the interactions that were included are not statistically significant, another factor can be selected that might be of interest);

iv. No further improvement is possible and TEVOP operation is stopped.

c. Some effects are significant:

i. Substitute new factors for one or more of the old factors (if one of the factors and the interactions that were included are not statistically significant, another factor can be selected that might be of interest).

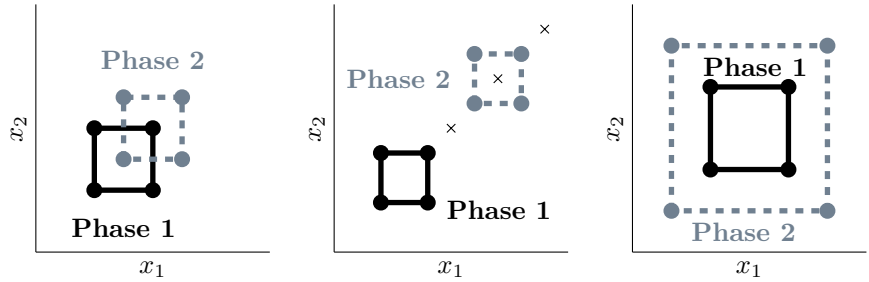


Figure 2.3: Examples of TEVOP decisions: (a) Adopt one of the design points as new reference condition, (b) Explore in favourable direction, points with an x-marker denote the line search, (c) Change design region.

In TEVOP no clear-cut decision rules are given, neither for when to conclude a *phase* as for the determination of the direction in which to move. Box posits that a committee—consisting of plant personnel, management and a statistician—should decide *when* and *how* to move by taking into account their experience of the process, the specific requirements and recommendations set forth by management, etc. In every published application of TEVOP, the authors themselves choose the number of cycles after which to move and in which manner should be moved from one phase to another. Some examples can be found in [17, 19, 20, 136]. An example of a possible two-factor TEVOP improvement is graphically plotted in figure 2.4 on which the contour lines of the underlying process model are shown.

EVOP Factorial-Design Technique

In TEVOP, no clear decision rules are given for the direction in which to move after the conclusion of a phase (the direction in which to move was subjective). It was only after more than 30 years that EVOP was further elaborated by

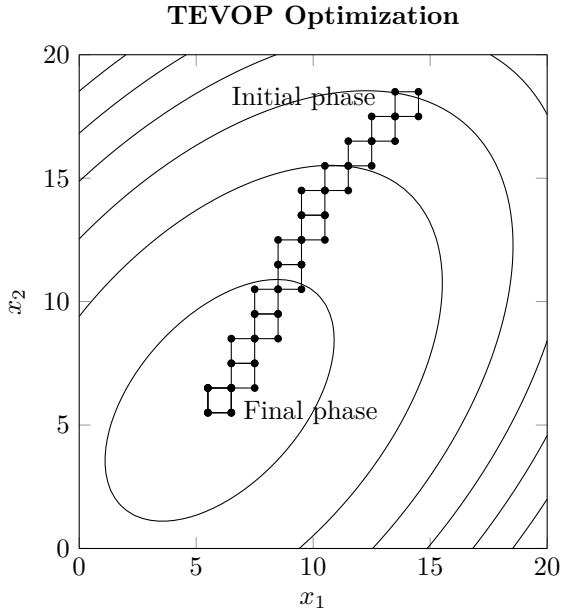


Figure 2.4: 2-factor TEVOP optimization.

Banerjee & Bhattacharyya [9, 10] who proposed the EVOP Factorial-Design technique (EVOPFD) for a three-factor system. Later, it was extended by Tunga et al [151] for an n-variable system. The term factorial design might seem contradictory since TEVOP also builds on factorial designs, but the addition proposed by these authors is to include the principles of Response Surface Methodology [107] to define decision rules for the direction in which to move, an option that statisticians undoubtedly incorporated before in TEVOP but which was never formalized up until now. In table 2.1 the decision rules for EVOPFD are summarized. Although these give a clear indication on which parameters to change, they still offer no clear-cut decision on the exact direction in which to move or how far one should move in a direction.

Other EVOP Techniques

Other modifications of TEVOP have been proposed such as Random Evolutionary Operation (REVOP) and Rotating Square Evolutionary Operation (ROVOP) [91]. As far as the author is aware, no publications have been made in which these adaptations have been used and this was already indicated in 1974 by Lowe [90]. Readers who are interested in the details of these adaptations are

Table 2.1: EVOPFD decision rules.

Observations on Effects and significance tests	**Action to maximize (minimize) objective function**
Effect(s): positive and significant, LoF[1] not significant	Increase (decrease) the level of corresponding parameter(s) from its initial value
Effect(s): negative and significant, LoF[1] not significant	Decrease (increase) the level of corresponding parameter from its initial value
Effect(s): not significant, LoF[1] significant	Maximization (minimization) achieved, if change in mean[2] is negative (positive)
Effects: not significant, LoF[1] not significant	Advisable to select a new search region and start a new phase of EVOP

[1] LoF: Lack-of-Fit test which tests for the presence of curvature. Can be formally tested only when including centerpoints.
[2] change in mean: (average responses at all design points without centerpoint – average responses at centerpoint)
Rules table derived from: [9, 10, 151]

referred to the following references [20, 90, 91]. These two adaptations will not be discussed in this text as they have, to the author's best knowledge, never been used in industry.

Disadvantages of EVOP Techniques

EVOP has not been widely used in industry which might be attributed to the disadvantages of the current implementations: (1) there is no clear indication of when to stop experimenting, this pertains to the number of cycles before one can conclude that there is no effect and when the complete improvement procedure should be stopped; (2) The traditional full factorial design requires a minimum of 2^k experiments per cycle for k factors, this restricts the use of the method to only a few key variables; (3) Achievement of a significant effect of a factor can depend on the chosen factor levels for this variable, i.e. the closer the factor levels are located to each other, the more cycles will be necessary to estimate (small) effects.

Process supervisors usually require the smallest measurable changes in the factor levels [90], i.e. the smallest change in settings possible (for example a change in the reference speed of $\pm 0.25\,\mathrm{m\,s^{-1}}$), and this can become a serious problem as the sample size (and consequently the number of cycles) necessary to detect such a small effect will become too large to be practically feasible to execute.

2.2.2 Automated Implementation

With the increase in computational power and use of sensors in production processes, it becomes interesting to develop an EVOP software algorithm that can automatically adjust the process settings, execute the required measurements and decide in which direction the process settings must be changed to improve. In 2002, Holmes & Holmes [67] filed a US patent for a methodology based on TEVOP that can be used in such a way (evidently the authors included some decision rules on how to move but these are not documented). This clearly shows that there is a benefit and will to implement EVOP automatically. However, not many automated applications have been found and those found use TEVOP as a basis, that is: they do not formally describe the decision rules implemented to execute the moves. Currently there is also no (commercially available) software that can execute an EVOP improvement, which is a must for automated implementations.

An automated implementation opens possibilities for higher-dimensional applications and consequently alternative base designs that are more efficient in terms of measurements. Additionally, more complex designs can be constructed which provide more flexibility in terms of constraints on the factor levels.

2.2.3 Current Directions Of Research

With the possible inclusion of more than three factors in an EVOP procedure it becomes worthwhile to advocate the use of base designs that use a more efficient number of measurements than the 2^k of the full factorial design. By reducing the amount of measurements in each phase, improvement can be achieved faster. A logical extension is using a fraction of the design points of the full factorial design in a fractional factorial design. Mathur *et al.* [97] state that the use of fractional factorials increases the probability of obtaining incorrect directions of improvement if there are non-linearities present in the true (noise-free) response of the process. To deal with this, they proposed a minimax design procedure in 1994 to select two-level fractional factorials that are robust (in their inferences) to nonlinearities in the true response.

The use of optimal designs to allow for any sample size or restrictions (factor constraints) on the design region lead Chu *et al.* [34] in 2002 to propose the use of $\mathcal{D}$-optimal designs. They compared the improvement of a $\mathcal{D}$-optimal design with the traditional full factorial design on the simulation of a pulp digester process with six factors. They found that the $\mathcal{D}$-optimal design outperforms the full factorial design. Interesting as these results may be, the comparison they made between the use of the full factorial and $\mathcal{D}$-optimal is biased. The

comparison was executed on deterministic simulations and there was a difference in the estimated model between the two designs (interaction for full factorial, second order for $\mathcal{D}$-optimal) and with a sample size of 33 experiments for the $\mathcal{D}$-optimal design and 64 experiments for the full factorial design it stands to reason that the design with the smallest sample size will outperform the other on a deterministic simulation. Unfortunately, no clear description is given in the paper which decision rules are applied between phases, it is only indicated that they were changed when changing from full factorial to $\mathcal{D}$-optimal design, and it is difficult to ascertain their impact on the comparison.

Not only the dimensionality has increased in contemporary processes, quality is nowadays also often of a multivariate nature. In order to cope with such situations, Byun & Kim [25] wrote a short article in 2003 about the use of desirability functions to improve multiple responses in a TEVOP setting. By combining several responses into an overall desirability index it is possible to scale a multi-response problem to a single-response problem using the desirability index.

As explained previously, after the conclusion of each cycle a decision has to be made whether additional cycles have to be run (to significantly detect effects) or whether to conclude the current phase. The question on how many cycles to run before concluding that no effects are significant, was briefly addressed in 1968 by Box & Draper [21] when they showed—using the concept of statistical power—that, even with a small number of cycles, a surprising amount of improvement is possible.

In an EVOP scheme those factors that drive the response(s) under consideration are often called key factors. When, for any reason, the key factors change during operation this should be detected. Holmes & Mergen [66] postulate that—when no additional or new factors are active—the time series of observations should be stable (if the underlying process is stationary). This is done by comparing the usual variance estimate s^2 (equation 2.1), which is sensitive to changes in the average of the response variable over time, with the variance estimator q^2, which takes into account the time order of the data (equation 2.2) and is rather insensitive to movements of the average of the response variable over time.

$$s^2 = \frac{\sum_{i=1}^{n_T} (y_i - \bar{y})^2}{n_T - 1} \tag{2.1}$$

with, in equation 2.1, y_i the i-th observation of the response variable, $\bar{y}$ the average of the response variable and n_T the total number of observations. If

there are movements in the response variable's average over time, they are reflected in this estimate.

$$q^2 = \frac{MSSD}{2} = \frac{1}{2\left(n_T - 1\right)} \sum_{i=1}^{n_T - 1} \left(y_{i+1} - y_i\right)^2 \tag{2.2}$$

with, in equation 2.2, y_{i+1} the $i+1$-th observation of the response variable. This variance estimator is based on the Mean Square of the Successive Differences (MSSD) and looks only at the successive differences (taking into account the time order of the data) and represents the variation that a process could display if some of the non-random elements, such as trends and cycles, were eliminated. If the factors involved in the current design account for the majority of the variation in the response, these two variance estimates should be essentially equal (i.e. the current factors are still the key factors). By testing whether these two variance estimators are statistically different, a conclusion about a change in key factors can be made. An example of their use is given in the applications section.

2.2.4 Applications

The following characteristics define processes to which EVOP can easily be applied [63] and for which its application is most useful:

- High-volume production;
- The potential benefits of process improvements are large (the process is an important one and not already operating at optimum conditions);
- Important process factors can be identified;
- The identified factors can be perturbed easily;
- The process stabilizes rapidly after a process change;
- The process response can be rapidly obtained and measured.

Although sequential experimentation can offer significant improvements to a process, its use in industry is not widespread. Both Lowe [90] and Hahn & Dershowitz [64] provided some insight in their 1974 papers as to why the use of EVOP was not more widespread after the first two decades it existed. Lowe argues that process supervisors refuse to accept statistical tests when their eyes and experience leads them to believe that there just cannot be a real

difference between the numbers presented. Furthermore, they are not willing to undertake any experimentation where there is a chance that substandard product may be made or that worse-than-standard costs may result. Hahn & Dershowitz conducted a survey and presented their results in their paper. The most frequently given reason for not using EVOP was "the reluctance to perturb manufacturing processes", i.e. why change operating conditions of a process which is producing? According to them, the key to promoting EVOP and implementing it in industry has to be found in the active participation of management and a positive environment suitable for innovative ideas for process improvement.

Many success stories in industry were reported before 1966 and were summarized by Hunter & Kittrell [71] but (industrial) published applications in the subsequent decades are somewhat scarce. There might be two reasons to which this can be attributed: (1) the advent of modelling techniques in the 1970's based on first principle modelling shifted the focus towards these methods, (2) EVOP might still be used in industry but is usually not published unless there is some academic involvement.

Published applications after 1966 are listed in table 2.2. These have been grouped by the number of factors in the problem, the EVOP technique used, application field and the type of application within the field. If the application type is listed as "R&D" then this application was executed at laboratory scale and "research study" indicates that these were simulation-based applications. The "Implementation" column lists if the application was run manually or automated. In two applications changes to EVOP were made that were very specific to the case under consideration and require too much detail to explain fully in this text. For these "Adapted EVOP" is used as denominator for the technique, indicating that changes have been made that are case-specific and can be found in the original article.

As can be seen from this table, the EVOP Factorial-Design Technique has become popular in the field of lab scale biological experiments, most notably solid state fermentation [11, 115]. In 2002 Shin & Cha [140] improved the transfection conditions for secretion of foreign proteins from insect Drosophila S2 Cells using a Green Fluorescent Protein Reporter. In their paper they used one-factor-at-a-time analysis (OFAT, sometimes called Single Factor Search – SFS) to do a first improvement; afterwards they selected four factors and executed a face centred cube design to build a second order model. This model could not fit the data within the range adequately and they used the endpoint of the OFAT search as the reference condition to start an EVOPFD optimization to see if it could pinpoint the direction of the optimum. They used the factor levels from their RSM-phase as data for the initial EVOP phase . They reported a 1.8-fold increase in the secreted yield after applying EVOPFD. In

Table 2.2: Overview of EVOP applications.

# Factors	Technique	Application field	Application type	Implementation	Reference(s)
2	EVOPFD	Biochemistry, fermentation processes	R&D	Manual	[9, 30, 164]
2	TEVOP	Photographic processing	Industry	Manual	[124]
2	TEVOP	Chemistry, instrumentation optimization	R&D	Manual	[103]
2	TEVOP	Milling industry	Industrial, determine optimal machine conditions	Manual	[122]
2	TEVOP	Control theory	Research study, tuning of parameters of PI-controller	Manual	[23]
2	TEVOP	Pet food manufacturing	Industrial, changing key variables	Automated	[66]
2	TEVOP	Cement industry	Industrial	Automated	[69]
3	EVOPFD	Biochemistry, fermentation processes	R&D	Manual	[4, 10, 12, 31–33, 38, 79, 81, 108, 117, 151]
3	TEVOP	Biochemistry	Industrial	Manual	[83]
3	TEVOP	Chemistry	R&D, lab setup for graduate students	Manual	[142]
3	TEVOP	Chemistry	Industry	Manual	[70]
4	EVOPFD	Biochemistry, fermentation processes	R&D	Manual	[11, 140]
5	EVOPFD	Biochemistry, fermentation processes	R&D	Manual	[77, 92]
6	EVOPFD	Biochemistry, fermentation processes	R&D	Manual	[106, 116]
Variable	Adapted TEVOP	Automatic tuning of controller parameters	Research study	Automated	[73]
Variable	Adapted TEVOP	Mixture Experiments	Research study	Manual	[86]

2010 Pansuriya & Singhal [117] optimized whey-independent serratiopeptidase production (SRP) from Serratia marcescens NRRL B-23112. Three factors were improved with EVOPFD and the SRP production was increased 17-fold. Also in 2010 Bankar & Singhal [12] used a Plackett-Burman design to screen and select the most significant factors to include in an EVOPFD scheme. Seven factors were screened and three were determined to be significant and of value to use in EVOPFD and the yield of the process was increased $\sim$ 2.3-fold. The use of screening designs to select the key factors to be included in an EVOP scheme is preferable if such a screening can be carried out (i.e. in industrial processes such a screening might not be possible since there is the possibility of producing substandard product).

In order to teach the principles of TEVOP, Smart [142] published an article in 2004 that describes how a laboratory setup is used to teach undergraduate students about TEVOP and give them some hands-on experience with the method. The students were asked to improve the operation of a gas absorber with three variables under research which might have an effect on the percentage reduction of CO_2 when using a packed scrubber to remove CO_2 from (simulated) industrial stack gas. The paper describes the setup and an execution of TEVOP after which a discussion follows that indicates which changes could be made, and their effect on the process. Such lab scale applications allow a prospective practitioner to get a good grasp of the concept as well as how to deal with the physical side of the experimentation.

In 1966 Rickmers [124] proposed TEVOP as a tool for persons in the photographic processing industry. An example is given where two variables in a photographic processor—machine speed and developer temperature—might influence the maximum density of the sensitometric control strip. It was not indicated if this was run as an example or in a true industrial process.

Bacon [7] published a short exposé in 1967 on the TEVOP methodology and states that "a limited number of applications have been attempted in the mineral dressing industry. Some of the successes have been most encouraging. Difficulties have arisen in other instances primarily because of large uncontrollable variations in the feed supply" but no example of such an application is given.

Hunter & Chacko [70] published a paper in 1971 that detailed how developing countries could benefit from the use of EVOP. They describe how TEVOP was successfully applied to a chemical manufacturing unit in industry; the object was to improve the optical density of polymer latex. Three key variables were included in the TEVOP program: addition time, temperature and stirring rate. After application of the methodology the throughput of the process was increased by approximately 25% since lower production times could be used to attain the required response.

Ramberg [122] prepared a technical report for the Manufacturing Methods and Technology Program of the US Army Material Command in 1974. A computer program called MACHOP is proposed to do the calculation and decision step in a TEVOP program of two key factors, speed and feed rate, for determining optimal machining conditions. The cost per piece is considered as the response. The data for an example machine is given in the report, however no data on a real process is presented. This is not classified as an automated application since responses have to be entered by hand. In essence all possible combinations of significant effects have been added to a decision table in MACHOP (for the specific application) and—after manually entering the response data—the appropriate decision is selected.

In the biotechnological process industry Kvist & Thyregod [83] published an article in 2005 in which they used TEVOP on the fermentation process of an industrial enzyme. Three factors were identified being pH, temperature and nutrient dosage with as response the yield of the process. The experiments were split in two blocks to ensure that the trials only require stable conditions for four or five batches in sequence (in one block) thus making the trials more robust against longer-term variations in the production process. Applying the TEVOP methodology increased the process yield by 45%.

The applications discussed up until now were run manually, with the calculations often done with statistical or spreadsheet software. With the advance of computational power and sensor technology since the methods conception, automated implementations have become possible and in the last two decades a few of these were published. Holmes & Mergen [66] use such an implementation in the production cycle of a pet food manufacturer. EVOP is executed on the production step of baking pet food pellets with as response the moisture content of the finished product as it leaves the cooling stage. Two key factors are continuously monitored to keep the response at the desired level while there are eight factors in total. The two factors that are initially included in the EVOP methodology are two temperatures during the baking process. By looking at the time series of the data (by comparing s^2 and q^2 as explained previously) an instability in the response was noted after 18 hours. This indicated that extraneous variation has influenced the process and other key variables should be selected. An investigation was performed and two new factors were identified as key variables for the EVOP scheme, being the moisture content of the pellets as they arrive at the entrance of the oven and the flow rate through the oven. These two factors replaced the previous two ones and the response was kept at the desired value. Investigation of the potential reason for the shift in control factors led to the conclusion that the ambient humidity level had undergone significant increase due to a large rainstorm taking place.

In 2011, Hulthén & Magnus Evertsson [69] used EVOP for the automatic control of a cone crusher operation. Two parameters in the control algorithm were improved, eccentric speed when the run is started and speed change, the response to be optimized was the product yield. EVOP was implemented since the process varies continuously due to varying feed material and wear of the crushers and screens. It is therefore not optimal to choose fixed parameters for speed control. Using EVOP, the magnitude of improvement after every phase was around 15%. The authors report that the overall magnitude of improvement potential compared to a fixed speed operation ranges from 5% to 20%. They also note their application of EVOP cannot deal with short-term variations such as changes in the raw material properties.

An area in which it is suggested to use EVOP is the estimation of control parameters. Box & Luceño [23] indicated that EVOP could be used to optimize the control parameters of, in this paper, a PI-controller. Another example of automatic control parameter estimation is published by Jen & Jian [73]. In their controller algorithm EVOP is used to estimate new model parameters for the controller at certain times when the algorithm indicates that an update of the controller parameters is necessary. A simulation study is carried out on a model of an industrial process that compares the proposed system with the traditionally implemented self-tuning controller and concludes that the new algorithm has better control performance.

2.3 Simplex Improvement

2.3.1 General Principles Of Basic Simplex

The Simplex method was originally proposed by Spendley *et al.* [143] in 1962 and starts from an initial set of design points that constitute the initial simplex; the response values for these initial points are measured after which heuristic rules are used to determine where a new measurement should be located.

The initial simplex is a geometric figure defined by $k+1$ points (vertices), where k is the number of factors under investigation. How to select the initial vertex points will be discussed later in this text. As a convention the geometrical figure will be referred to as a simplex, with lower-case lettering, while the improvement method will be referred to as Simplex, with a capital S.

After measuring the responses of the initial simplex, simple heuristic rules are implemented to calculate one new vertex. These rules can be summarized as follows [143, 155]:

1. Rank vertices in decreasing order of response, label the coordinates of the worst response as $\mathbf{w}$; the next-to-worst as $\mathbf{n}$ and denote the coordinate matrix of all responses that are better than $\mathbf{w}$ by $\mathbf{C}$;
2. Calculate the reflection point $\mathbf{r}$, $\mathbf{r} = \frac{2}{k} \sum \mathbf{C} - \mathbf{w}$;
3. Remove coordinate row $\mathbf{w}$ from the simplex;
4. Transfer coordinate row $\mathbf{n}$ to coordinate row labeled $\mathbf{w}$;
5. Measure response at reflection point $\mathbf{r}$;
6. Rank the remaining vertices, leaving out row $\mathbf{w}$ and return to step 2.

This algorithm is graphically represented in a flowchart in figure 2.5 and a two-factor example is plotted in figure 2.6.

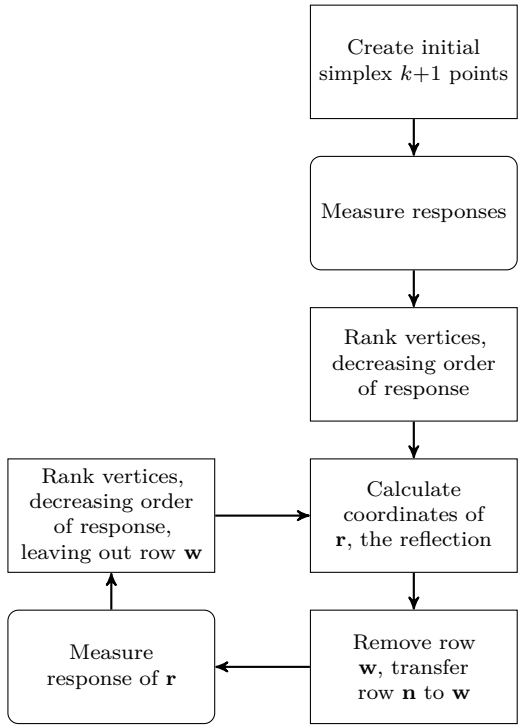


Figure 2.5: Flowchart Simplex.

To deal with noise in stochastic processes step 2 of the procedure is often amended with the so-called $(k + 1)$-rule and becomes:

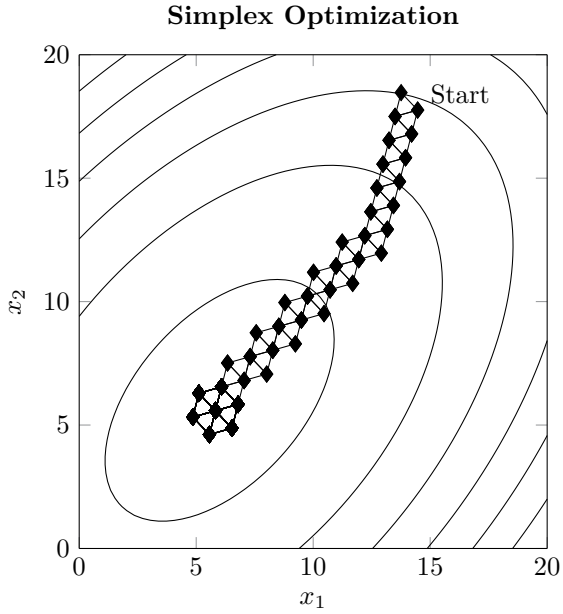


Figure 2.6: 2-factor Basic Simplex improvement, tilted algorithm.

2.a if a point is retained in $(k+1)$ succesive simplexes, and is not eliminated from the simplex, do not move but discard this result and replace it by a new observation at the same point, i.e. move this point to the row labeled w. Otherwise, apply rule 2.b;

2.b Calculate the reflection point $\mathbf{r}$, $\mathbf{r} = \frac{2}{k}\sum \mathbf{C} - \mathbf{w}$.

2.3.2 Constructing The Initial Simplex

There exist several ways of constructing the initial simplex. In 1998 Öberg [110] showed on well-defined functions that the importance of this initial design can affect the speed of convergence. It is shown on two polynomial functions that the $\mathcal{D}$-optimal simplex outperforms the tilted simplex, which outperforms the corner simplex.

Manual Construction

The initial $k+1$ points could be chosen by the experimenter if there is a strong motivation (based on previous experimentation) to experiment at certain settings.

Corner Method

In the corner method, one point is fixed and each other point is placed a certain distance (called factorstep dx_d in this text) in one factor from this point. Let this starting point have coordinates $\mathbf{x_{co}} = \{x_{co,1}, x_{co,2}, \ldots, x_{co,k}\}$, with k the number of factors under investigation, the vertices of the corner simplex are then placed according to table 2.3.

Table 2.3: Initial corner simplex.

Vertex	Factor 1	Factor 2	Factor 3	...	Factor k
1	$x_{co,1}$	$x_{co,2}$	$x_{co,3}$	$\ldots$	$x_{co,k}$
2	$x_{co,1} + dx_1$	$x_{co,2}$	$x_{co,3}$	$\ldots$	$x_{co,k}$
3	$x_{co,1}$	$x_{co,2} + dx_2$	$x_{co,3}$	$\ldots$	$x_{co,k}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$k+1$	$x_{co,1}$	$x_{co,2}$	$x_{co,3}$	$\ldots$	$x_{co,k} + dx_k$

If all factors are measured in the same metric and the factorsteps are equal for every dimension d then this method leads—in two dimensions—to an isosceles right triangle. In such a case the geometry of the resulting simplex is retained throughout the reflections—of which an example is given in figure 2.7a—but the Euclidean step size between resulting reflections depends on the point that is rejected, as graphically shown in figure 2.7b.

Tilted Method

For the tilted algorithm the initial vertex coordinate is given as $\mathbf{x_{co}} = \{x_{co,1}, x_{co,2}, \ldots, x_{co,k}\}$, which is the one vertex chosen by the experimenter, the vertices of the tilted simplex are then placed according to table 2.4.

The parameters p and q are calculated using equations 2.3 and 2.4, with dx_d the factorstep in the d-th dimension as before.

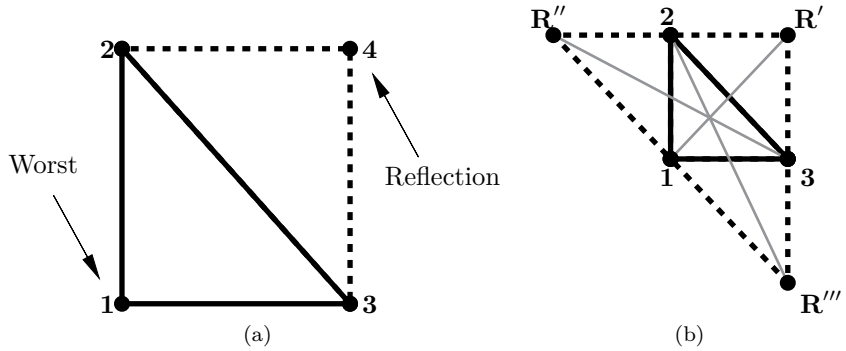


Figure 2.7: Simplex Corner Algorithm (a) Example of 2-factor Simplex reflection, (b) All 2-factor Simplex reflection possibilities.

Table 2.4: Initial tilted simplex.

Vertex	**Factor 1**	**Factor 2**	**Factor 3**	**...**	**Factor k**
1	$x_{co,1}$	$x_{co,2}$	$x_{co,3}$	...	$x_{co,k}$
2	$x_{co,1} + p_1$	$x_{co,2} + q_2$	$x_{co,3} + q_3$	...	$x_{co,k} + q_k$
3	$x_{co,1} + q_1$	$x_{co,2} + p_2$	$x_{co,3} + p_3$	...	$x_{co,k} + q_k$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$k+1$	$x_{co,1} + q_1$	$x_{co,2} + q_2$	$x_{co,3} + q_3$	...	$x_{co,k} + p_k$

$$p_d = dx_d \cdot \frac{\sqrt{k+1} + k - 1}{k \cdot \sqrt{2}} \tag{2.3}$$

$$q_d = dx_d \cdot \frac{\sqrt{k+1} - 1}{k \cdot \sqrt{2}} \tag{2.4}$$

Details about the tilted simplex construction and the origin of the used constants are discussed by Beveridge & Schechter [16]. The use of these parameters ensures that, when all factors are measured in the same metric and the factorstep is equal for all factors ($dx_d = dx$), the resulting simplex is regular (regularity is achieved when a polygon is equiangular and equilateral). Due to the equations used to calculate p and q, the sides of such a regular polygon all have length dx. For a regular, two-factor simplex, an example of the reflection is given in figure 2.8a, figure 2.8b shows that all reflection possibilities in such a regular simplex

result in the same Euclidean step size between the points labelled **w** and the reflection **r**.

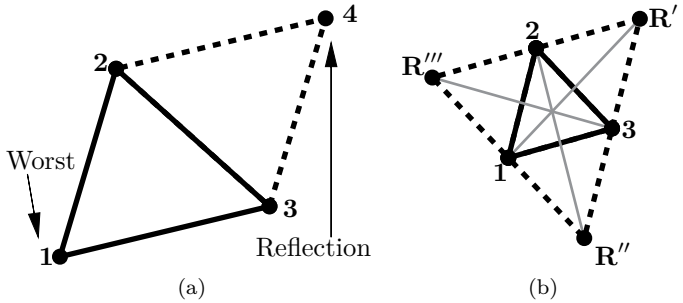


Figure 2.8: Simplex Tilted Algorithm (a) Example of 2-factor Simplex reflection, (b) All 2-factor Simplex reflection possibilities.

$\mathcal{D}$-Optimal Method

The initial simplex can also be chosen using a $\mathcal{D}$-optimal design. To select the points in this way, a $\mathcal{D}$-optimal design for a linear model with $k + 1$ measurements is estimated. These design points then form the initial simplex. More information about $\mathcal{D}$-optimal designs will be presented in section 3.2.3.

For the construction of the initial Simplex a coordinate exchange algorithm (see section 3.2.3 for more information about the algorithm) is used since this does not require the explicit specification of a candidate set. Since the use of a $\mathcal{D}$-optimal initial simplex is not wide-spread and is more difficult to implement in low-level software, it will not be used in the comparison chapter of this dissertation, which aims to compare the basic EVOP and Simplex implementations.

2.3.3 Dealing With The Borders Of The Experimental Domain

To ensure that no experiments outside of the boundaries are performed the Simplex procedure will allocate an infinitely bad response to points outside of the experimental domain defined by a hypercube, forcing the procedure to reflect back inside this hypercube. These points outside of the cube are phantom measurements—that is, they do not require experimentation effort—and they are set by the procedure.

2.3.4 Basic Simplex Implementation

The Basic Simplex has been implemented in a Matlab® (Matlab R2010b, The Mathworks Inc., Natick, Massachusetts USA) class that wraps all the functions necessary to perform the improvement. The pseudo-code in algorithm 2.1 shows how Basic Simplex was programmed. To construct the initial simplex the algorithms described above are used for the corner and tilted method. For the $\mathcal{D}$-optimal initial simplex, the Matlab® command *cordexch* is used, which uses a coordinate-exchange algorithm to generate the $\mathcal{D}$-optimal design. The algorithm generates several designs and selects the best one from them, currently the number of designs generated is set at 100, and the maximum number of coordinate-exchange iterations to arrive at a design is also set at 100. This $\mathcal{D}$-optimal construction of an initial simplex will not be used in this dissertation but is included for reference purposes.

In appendix A an example of how to run the developed Matlab® software package is presented.

2.3.5 Other Simplex Procedures

The most popular derivative of the basic Simplex is the Nelder-Mead Simplex, also called the Variable or Modified Simplex [109]. In this derivative the simplex figure is changing size by expansions and contractions. This makes the procedure more adventurous in its exploration of the process surface and can lead to more rapid convergence. This is usually not wanted behaviour in full scale processes: (1) large expansions are not desired as the risk to produce unacceptable output increases and (2) successive contractions are not desired as the change in factor levels would become so small that no distinction between the responses might be made in the presence of noise [14]. Therefore, it is a technique that is primarily suited for numerical optimization and use in research and development where it is relatively safe to make large changes in the factor levels [111]. Also numerical experiments and carefully constructed examples have revealed that the Nelder-Mead algorithm may be unreliable even in fairly simple situations [149].

Another derivative is the Super-Modified Simplex [125] where the new measurement is extrapolated from a second-order polynomial curve. Within production processes the Super-Modified Simplex method has, to the author's best knowledge, never been used. It is primarily used in the field of chemistry where the method originated from.

Algorithm 2.1. Basic Simplex pseudo-code.

Require: Input settings: number of factors k, starting point $\mathbf{x_{co}}$, bounds of experimental domain $\mathbb{D}$, type of initial Simplex (corner, tilted, $\mathcal{D}$-optimal)

1. Generate initial simplex $\mathbf{S}_0$, a $(k+1) \times k$ matrix, and randomize the rows (vertices).
2. Measure responses, gather measured responses $\mathbf{y}$ in matrix $\mathbf{G}$, a $(k+1) \times 2$ matrix consisting of the vertex numbers (the first column) and the measured responses (second column)
3. Rank and sort rows on descending responses (second column of $\mathbf{G}$)

   $y_1 \geqslant y_2 \geqslant \ldots \geqslant y_{k+1}$

   If responses are equal, rank descending on vertex number (ensures that oldest vertices are removed first).

if current simplex is initial simplex **then**
  Sort rows of $\mathbf{S_0}$
else
  Sort rows of $\mathbf{C}$
end if

4. Label responses, only for initial simplex

if current simplex is initial simplex **then**
  label $\mathbf{w} = \{\mathbf{x}_{k+1}\}$
  label $\mathbf{C} = \{\mathbf{x}_j\}_{j=1}^{k}$
end if

5. Calculate reflection $\mathbf{r}$

   $\mathbf{x_r} = \frac{2}{k} \sum \{\mathbf{C}\} - \{\mathbf{w}\}$

6. Check if $\mathbf{r}$ is within the experimental domain. If yes, measure response at reflection. The vertex number of $\mathbf{r}$ is $k+1+i$. With i the number of the current reflection.

if $\{\mathbf{x}_r\} \notin \mathbb{D}$ **then**
  $y_r = -\infty$
else
  Measure response y_R at coordinates $\{\mathbf{x}_r\}$
end if
$\{\mathbf{G}_r\} = \{k+1+i; y_r\}$

6. Label rows: previous $\mathbf{n}$ goes to $\mathbf{w}$, matrix $\mathbf{C}$ becomes the new matrix in which the responses have to be ranked.

label $\mathbf{w} = \{\mathbf{x}_k\}$
label $\mathbf{C} = \left\{\{\mathbf{x}_j\}_{j=1}^{k-1}; \{\mathbf{x}_r\}\right\}$
label $\{\mathbf{G}\} = \left\{\{\mathbf{G}\}_{j=1}^{k-1}; \{\mathbf{G}_r\}\right\}$

7. Go to 3

2.3.6 Current Directions Of Research

For the Basic Simplex method no real innovative advances have been made as attention in R&D (especially in analytical chemistry) has shifted towards the use of the Variable and Super-Modified Simplex. Due to the immense popularity of Variable Simplex in numerical optimization, a vast amount of literature is devoted to refinements of this method; however this is out of the scope of this text as the interest is not numerical optimization.

In 2003 Xiong & Jutan [160] proposed a method, called Dynamic Simplex, for improving processes in which the optimal conditions change with time. They propose an algorithm to track these moving optima based on a fixed-size simplex with only reflection operations. Since fixed-size simplexes are used, this application can be seen as an extension of Basic Simplex. Although Dynamic Simplex is not used in this dissertation it is presented here for reference purposes as it could be a valid method to track non-stationary optima. Algorithm 2.2 presents the pseudo-code for Dynamic Simplex for maximizing a response.

The major change compared to the Basic Simplex algorithm is the fact that, instead of continuously moving towards the optimum by successive reflections and move the next-to-worst point to worst, M reflections of a simplex are made. These M simplices represent M choices for the subsequent move. Since the process is continuously moving (drifting) one cannot always expect an increase (when maximizing) of the average function value at each successive reflection. The goal is to track a moving optimum, which sometimes may have a decreased function value when it drifts in time. Therefore these M directions are explored and the direction with the best average function value is selected for a move. The authors tested their algorithm on 2D and 3D functions and on a simulation of the Williams-Otto Continuously Stirred Tank Reactor of which the tracking results are presented graphically in the paper. Tracking the optima in these cases was reported to work satisfactorily.

Simplex is often used to deal with sensory data, since not many measurements can be performed and one wants to make rapid decisions based upon the sensory input. The use of intelligent functions to quantify responses made by human observations is a field that might deserve more attention. The expertise of operators in an industrial process and their in-line observations might be used in conjunction with the Simplex technique to rapidly improve processes. In 2004, Curt *et al.* [36] used fuzzy membership functions to calculate a response based on sensory data (see also the applications section). The use of fuzzy membership functions succeeded in transforming expert knowledge (i.e. sensory information) into functions that can be processed by computers.

Algorithm 2.2. Dynamic Simplex pseudo-code.

loop

1. Let the starting simplex in iteration d be $\mathbf{S_0} = \{\mathbf{x_j}\}_{j=1}^{k+1}$ and the objective function values on these vertices be $\{g_j\}_{j=1}^{k+1}$
2. Rank in descending order and sort the vertices of the simplex, such that $g_1 \leqslant g_2 \leqslant \ldots \leqslant g_{k+1}$
3. Successive M reflections

   A single point of reflection of the worst (g_1) point of simplex $\mathbf{S}_0$, thus reflecting the point whose coordinates are labelled $\mathbf{x}_1$ after ranking, can be written as:

   $\mathbf{x}_{k+2} = \frac{2}{k} \sum_{j=2}^{k+1} \mathbf{x}_j - \mathbf{x}_1$

   This is repeated M times, as below. After the reflection is repeated M times, the last simplex M has coordinates $\mathbf{S}_M = \{\mathbf{x}_j\}_{j=M+1}^{k+M}$ and a series of simplices now exist $\{\mathbf{S}_p\}_{p=1}^{M}$

   for 1 to M **do**

   $\mathbf{x}_{k+1+M} = \frac{2}{k} \sum_{j=M+1}^{k+1+M} \mathbf{x}_j - \mathbf{x}_M$

   end for
4. Choose the start simplex for iteration $d+1$ by calculating the average function values in every simplex (save the initial simplex $\mathbf{S}_0$.

   $\bar{\mathbf{g}}_\mathbf{S} = \frac{1}{k+1} \sum_{j=p+1}^{k+p+1} g_j$ with $p = 1, 2, \ldots, M$
5. Select simplex $\mathbf{S_q}$ that satisfies:

   $\bar{\mathbf{g}}_{\mathbf{S_q}} = \max \left\{ \bar{\mathbf{g}}_{\mathbf{S_p}} \right\}_{p=1}^{M}$
6. Remeasure the response at point $\mathbf{x}_{k+1}$
7. Set $\mathbf{S}_q$ as the new start simplex $\mathbf{S}_0$ for iteration $d+1$

end loop

2.3.7 Applications

Simplex is especially suited to improve processes that have one or more of the following properties [62]:

- The process variables can be perturbed only at special times, such as at shift change (e.g. tool change in milling, change of temperature of an oven);
- Process performance is changing over time, i.e. the process is non-stationary and the response drifts over time;
- Statistical calculations need to be minimized.

Some advantages of using Simplex are [143]:

- The direction of advance depends only on the ranking of the responses and not on their values on any absolute scale. Thus the procedure can readily applied even in those instances when a response can only be judged and not measured;
- A regular simplex (constructed using the tilted algorithm for example) in $k+1$ dimensions can be derived from a regular simplex in k dimensions by the addition of only one additional point. Thus a new factor, previously held constant, may easily be incorporated into a Simplex scheme at any stage. However, the elimination of factors is less straightforward, either the method has to be restarted by building a new simplex with fewer factors or regularity will be lost.

Many Simplex applications have been described, particularly in analytical chemistry. However, many of these applications use Variable or Super Modified Simplex rather than the Basic Simplex. Since large steps or successive contractions are not typically a problem in an R&D environment this is a logical choice: the focus is on reaching the optimum as fast as possible, not on keeping the output within acceptable bounds. Due to its popularity in R&D, and mainly chemistry, a number of reviews of the methodology (Basic, Variable and Super Modified) have been published throughout the years, such as [15, 43, 44, 126]. In their reference work Walters *et al.* [155] included an extensive list of application references, which was updated by Walters in 1999 [154]. In this reference list, no separation was made as to which Simplex method was used but it is a valuable reference guide for readers interested in Simplex methods in general.

In table 2.5, applications using Basic Simplex are presented to give an idea about the type of processes for which the method is used, both in- and outside of R&D environments. Again, the implementation column is added and all implementations in which the response was not measured automatically are labelled as manual implementations. For one of the implementations it could not be determined from the publication if the algorithm was run manually or not, this has been indicated by putting "?" in the implementation column.

In 1969 Long [89] improved the sensitivity of a *p*-rosaniline agent for determining sulphur dioxide. Two factors, formaldehyde and hydrochloric acid, were selected in a previous factorial experimentation as having the highest effect on the sensitivity. An initial improvement was performed and a new, smaller, simplex was constructed around the optimal settings from this initial improvement further fine-tune them. By shrinking the simplex the response was improved approximately 1.1-fold from the conditions after the initial improvement.

In 1980 McDevitt & Barker [98] used Simplex to effectively improve the synergic extraction of a bis-diketo copper(II) complex by optimizing three factors (pH and concentrations of isoquinoline, acetylacetone). A 2.5-fold increase in the response was observed between the highest response in the initial simplex and the endpoint of the improvement.

In 1991 Mathieu *et al.* [96] used Simplex as a method to follow a discontinuity in ternary mixtures. These discontinuities can be, for instance, the limit of miscibility between two liquids, a phase change in a solid, an instability zone for a catalyst or the limit of appearance of a phenomenon (e.g. occurrence of errors in production, corrosion of a protective coating). They give two examples, based on previously published data, in which the tracking of discontinuities by Simplex is successful.

In 1996 Danzer & Schwedt [37] developed biosensors for pesticide and heavy metal screening. To minimize the practical expenses a combination of experimental designs and subsequent Simplex improvements were used. The paper deals primarily with the testing of the developed biosensors and not much information is given about the Simplex procedure. This paper is included since it is an excellent example that Simplex has merit when sampling is expensive and has to be kept to a minimum.

In 1999 Prater *et al.* [121] used a two-factor and a three-factor Simplex (with other reagents in both) to improve the crystal growth of lysozyme crystals. The authors proposed this method as an alternative to OFAT to conserve materials when executing crystal growth experiments on the International Space Station. Each crystal is given a quality score on an ordinal scale and this is used as the response for Simplex. The authors conclude that the performance of the

Table 2.5: Overview of Simplex applications.

# Factors	Application field	Application type	Implementation	Reference(s)
2	Chemistry	R&D	Manual	[47, 89]
2	Chemistry	R&D, numerical optimization of model after classical design	Automated	[18]
2	Cement industry	Industrial	Automated	[65]
2	Food industry	Industrial	Manual	[36]
2, 4	Paper industry	Industrial	Manual	[78]
2, 3	Biochemistry	R&D, ordinal responses	Manual	[121]
3	Chemistry	R&D	Manual	[37, 47, 98]
3	Textile industry	R&D, dual-response optimization	Manual	[26]
4	Chemistry	R&D	Manual	[139]
4	Chemistry	R&D, multi-response optimization	?	[118]
Variable	Chemistry	R&D, research study, mixture experiments	Manual	[96]

Basic Simplex is sensitive to the size of the initial Simplex. Furthermore, they remark that, in a two-factor Simplex, no absolute scoring system is necessary to rank the responses since a comparison has only to be made between the new crystal and the previously best crystal. Remember, there are three vertices in a two-factor Simplex, the heuristic rules state that the next-to-worst vertex becomes the worst in the next simplex, leaving only a comparison between the previously best and the new reflection in the new Simplex and thus foregoing any scoring system that would have to assign a rank to all vertices. Additionally, the Simplex heuristic rules essentially allow to execute two new experiments at one time if the process or study allows to execute multiple experiments at once (in this case, several crystal can be grown at once). Again this is due to the fact that the next-to-worst condition **n** is moved to the row labelled worst or **w** in a new move of Simplex. The calculation of a reflection **r** is dependent upon knowing all $k + 1$ vertex coordinates and the identification of the vertex with label **w** (see the rules in section 2.3.1). If a reflection is executed, and this point should be called **r**, then all vertex coordinates of the new simplex are known. The heuristic rules indicate which vertex will be labelled **w** (the vertex labelled **n** previously) and the subsequent reflection $\mathbf{r}'$ can also be calculated. Being able to conduct the measurements at both **r** and $\mathbf{r}'$ simultaneously essentially doubles the rate of improvement.

In 2000 Santos-Delgado *et al.* [139] improved an esterification reaction in which the four factors of interest were methylation temperature and time, hexane volume and shaking time. The improvement led to a decrease in reagents and shaking time which reduced the costs and the generation of residuals. Also, the authors note that a very small number of experiments were used and experimentation time was therefore decreased with respect to performing an OFAT analysis.

In 2006 Pasamontes & Callao [118] used a fractional factorial design to screen the key factors from a pool of six candidates in a sequential injection analysis. Two responses, related to the spectra of the images, are combined using Derringer desirability functions [45] to obtain a single response. Four key factors were identified based on the screening step and the Simplex procedure is used to improve this desirability index. The desirability index is increased from 0.936 to 0.979 after eleven experimental measurements.

The previous applications were all in R&D, some applications of the use of Simplex improvement in industry are reported but are rather scarce. They are outlined in the next few paragraphs.

In 1967 Kenworthy [78] described the application of Simplex in the paper industry in which TEVOP cannot be used because of the fact that (1) sampling is limited and (2) the optimum may move rapidly in time which results in the

need for rapid decisions. The author does not state in detail what is meant by limited sampling, it is expected that in this application sampling can only take place at certain times (end of a production run) and decisions need to be taken rapidly as the intervals between measurements can be large. TEVOP cannot be used in such a situation since it requires a minimum of 2^k samples before a decision can be made in each phase, while Simplex requires a minimum of one sample. In an initial four-factor example the thickness of building board across the width of the boards needs to be controlled. Two Simplex studies were executed which did improve the range of the thickness. Kenworthy describes a second study in which two factors are controlled to increase the tensile strength of paper, the increase in tensile strength was of the order of 15%.

In 1972 Cannon *et al.* [26] described the application of a three-factor Simplex to a textile heat transfer unit. Two responses needed to be improved and desirability functions were implemented. In the first phase of the application an 81.3% improvement in the desirability was achieved, in the second (shrinking the initial simplex and restarting around the end conditions of the initial Simplex improvement) a 76.3% improvement was achieved. This study was conducted on an industrial machine in an R&D environment.

In 2003 Holmes [65] used Simplex in a cement plant to increase the time between the formation of kiln balls. This application is fully automated and uses software to change the settings and to measure the responses. Two factors, the use of limestone ore and iron ore, are taken into account. The result is a 37% increase in the time between the formation of kiln balls by decreasing the expensive limestone ore and increasing the less expensive iron ore. As such the improvement not only succeeded in increasing the time between formations but also decreased the material costs.

In 2004 Curt *et al.* [36] used Simplex to improve two process parameters, mixing duration and mixer rotation speed, on a meat emulsification process. The response dealt with sensory characteristics determined by operators in-line. These characteristics were scaled to a numeric response variable "chopping degree" using fuzzy membership functions. This allowed for the improvement of the in-line intermediary sensory quality.

2.4 Conclusions

In this chapter Online Sequential Process Improvement methods have been discussed, most notably the Traditional EVOP method based on factorial designs, its derivative the EVOPFD technique and the Basic Simplex method. Where the former two are based on statistical principles and can rely on inference,

the latter uses heuristic rules to determine in which direction to move. For TEVOP and its derivative, EVOPFD, several applications exist. Not that many publications for industrial applications are reported but in the last decade some noteworthy publications were made in this context, indicating that there is a renewed interest in this method.

The Basic Simplex method is a largely forgotten method as its more popular derivative Variable Simplex has replaced it, most notably in numerical optimization and in R&D environments. However, the use of Variable Simplex in an industrial environment is often unwanted as the expansion moves can increase the probability of producing unacceptable output and the contraction moves might decrease the difference between responses in such a manner that they cannot be distinguished from each other in the presence of noise. Moreover there are some concerns about general convergence of both Simplex methods [84] that have not been addressed sufficiently.

The main strength of the Basic Simplex method is not its capability to improve industrial processes but its power to track a previously determined optimum when it is subject to time-drift. It is the belief of the author that, should a process be subject to continuously changing optimal conditions (i.e. a drift of the process mean), the Basic Simplex method or its extension the Dynamic Simplex, could be a simple and effective tool of tracking these changes.

There are still some questions left unresolved in the literature. In modern, complex processes, there are often more than three to six key variables that influence the process and there is no literature that investigates how performant these methods are if the dimensionality increases. If there are many key variables for EVOP, the number of experiments per cycle should be reduced to a more efficient number than the 2^k of the full factorial. How do such efficient designs perform compared with the traditional full factorial implementation?

Also the effect of noise has never been compared on Simplex and EVOP. From such a study, conclusions may be drawn on when to use which method. Indeed, no systematic literature has been found that compares EVOP and Simplex.

The multi-variate nature of quality should be taken into account. For Simplex, optimizations using desirability functions for multi-response improvement have already been executed. For EVOP, the application was mentioned in literature but no industrial case study has been found.

In the current process industry, these questions are pertinent and should be answered in depth to assess the applicability of these two methods and this dissertation attempts to answer these questions to determine where and how these methods can best be applied in industry.

3 Implementation Of And Extensions To Online Sequential Process Improvement For Evolutionary Operation

3.1 Introduction

In the previous chapter the basic principles of Online Sequential Process Improvement (OSPI) were explained. Evolutionary Operation (EVOP) and Basic Simplex were handled in detail, and an overview of recent literature was provided. It was concluded that most of the applications were still performed on small scale processes where automation and the restrictions for online improvement are no issues. As a consequence, the implementation of EVOP in those cases was performed manually, often using standard spreadsheets such as Microsoft Excel®. For the purpose of improving full scale processes in industry, however, these issues are important and require a rigorous framework so that EVOP can be run in an automated fashion.

For Basic Simplex the automatic implementation is straightforward and the rules presented for the original method (see chapter 2) can be readily implemented in software code. For EVOP on the other hand, additional work is required to extend the basic principles so that automated operation is feasible. This chapter will describe these extensions in detail.

The extensions discussed and implemented deal primarily with the calculation of the statistical model during each phase and the calculation of the direction in which to move. The proposed extensions are created in such a way that only basic matrix operations are necessary to implement the algorithm. This ensures that even low level controllers, i.e. controllers that can perform a limited number of mathematical operations, could be used to run the algorithm. By choosing for this type of extensions, in the long run, the algorithms could even be implemented in the machine software itself.

As a basis for extending the EVOP technique, the same principles as presented in the flowchart of TEVOP, figure 2.2 of the previous chapter, are used. A *design* is generated, after which the responses are *measured* at the factor level combinations dictated by the design. When all measurements are collected, a statistical model is *estimated* (calculated) online and based on this model a *decision* is made on the direction in which to move. Most adjustments take place in the *calculation* and *decision* step of the procedure which govern the statistical model built and the direction in which to move.

The proposed automated EVOP scheme is presented in the flowchart in figure 3.1 and will be explained in this chapter. It also forms the backbone of the implementation in a Matlab® (Matlab R2010b, The Mathworks Inc., Natick, Massachusetts USA) program which is used throughout the dissertation.

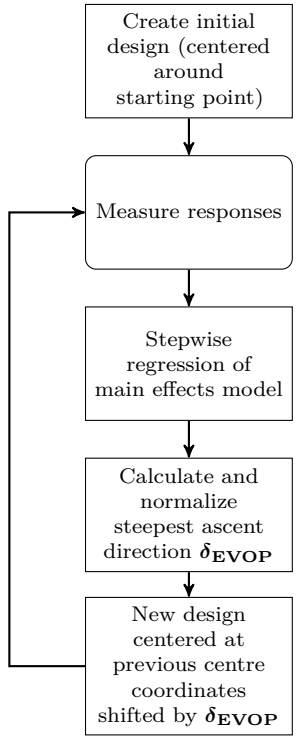


Figure 3.1: Flowchart Automated EVOP algorithm.

The first step consists of choosing an appropriate base design. Appropriate here means that the design should be efficient in estimating the direction of improvement based on a regression model, but should also adhere to the

basic OSPI philosophy that perturbations should be limited so that the risk of producing unacceptable output is minimal (the size of these perturbations will be called factorstep in the remainder of the dissertation). In a next step, the different experimental measurements are executed and the responses stored to disk. Furthermore, it is important that the order in which the experimental measurements are executed should be randomized in order to avoid the risk of including the effect of unknown factors. After all measurements are collected, a regression model is fitted to the data points. The estimation of the significant terms can easily be done using stepwise regression approaches. Once the final model is available, the direction of the improvement is calculated. The procedure then ends by moving the base design in the direction of maximal improvement and starting over at this new location.

Just as it was the case for the factorstep, it is important as well that the step size is limited. Typically, it will be chosen to equal a predefined amplitude δ_{EVOP}, which equals the magnitude of the step size vector $\boldsymbol{\delta}_{\mathbf{EVOP}}$.

An improvement using this EVOP algorithm is shown in figure 3.2 for a two-factor problem.

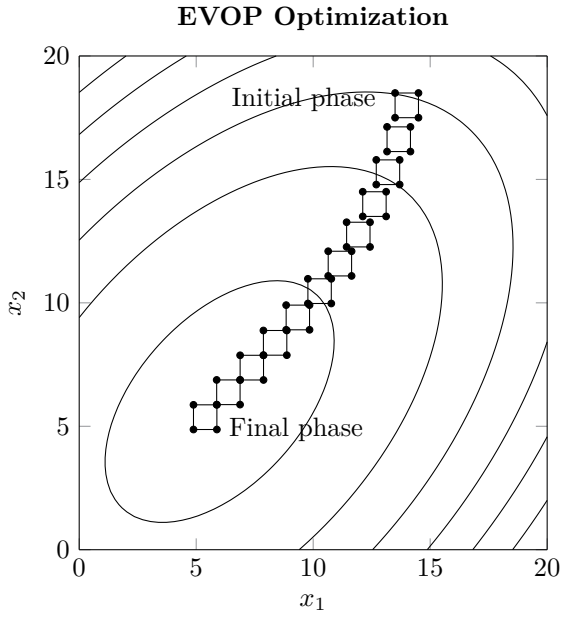


Figure 3.2: Example of 2-factor EVOP improvement.

Once the direction and size of the movement is defined, a plausible approach

is to perform only one experimental run and to compare the response with the past responses of the previous measurements. This is in fact a steepest ascent procedure as found in many statistical textbooks [82, 107] combined with the EVOP methodology. This approach will be denoted by Evolutionary Operation Steepest Ascent (EVOPSA) and can be regarded as the middle ground between EVOP—using statistical model building principles in every phase— and Simplex—augmenting the measurements with one point after the initial simplex is measured.

In order to take the aforementioned steps, several decisions must be made that will be explained in this chapter. This chapter is organized as follows: the creation of the initial design will be handled in section 3.2 labelled "Choice Of Base Design". The statistical model that is built based on these experimental measurements is given in a separate section "Choice Of Regression Model", while the calculation of the direction of improvement as well as the related step size are handled in the section "Calculation Of The Move". In the next section the implementation of the steepest ascent approach EVOPSA is detailed, followed by a section that covers the issue of dealing with the borders of the experimental domain. To conclude the EVOP and EVOPSA software algorithms and their settings are presented in pseudo-code.

3.2 Choice Of Base Design

In the TEVOP procedure, Box advocated the use of a two-level full factorial design since it allows estimating the local trend of the response surface of the process by means of a simple linear model. This reasoning holds well in case the number of factors k is limited but becomes prohibitive for larger problems as will be discussed later. As a result, TEVOP improvements are typically performed when only two or three key variables are present. Based on the sparsity of effects principle this situation is valid for a substantial portion of real life applications. Yet, in several contemporary processes it might be interesting to include more than three key variables. Since the EVOP method does allow to use any design as a basis [13], it is interesting to investigate the use of more efficient designs in situations where the number of key factors increases.

3.2.1 Two-Level Full Factorial Design

A factorial design is a design where all the possible combinations of levels from all factors are used the same number of times. The number of experimental measurements in such a design can be written as rl^k, where k denotes the

number of factors, l the number of levels for the factor and r the number of replicates for each design point (or combination of factor levels). When the number of replications $r = 1$, the design is called an *unreplicated* design.

One of the most commonly used designs is the two-level factorial design ($l = 2$) and it is this type of design that is proposed by Box to be used in EVOP. Figure 3.3 shows the graphical representation of such a two-level design for two and three factors. In this figure, the designs are centred on the origin, and scaled to the interval $[-1; 1]$. Such a standardisation—often labelled "coding"—has some interesting properties that will be discussed in the next section.

The full factorial is an interesting design to use when two or three factors are under investigation (as in TEVOP) but when the dimensionality of the problem increases, the number of measurements becomes prohibitive. In classical design of experiments, full factorial designs are almost never used in experimental studies of over five factors. Indeed, of the 33 applications discussed in chapter 2, only 4 were performed for $k \geqslant 5$ (two for $k = 5$ and two for $k = 6$). For an unreplicated full factorial design of five factors, 32 experimental measurements are necessary per cycle. Every added factor increases the number of measurements by a factor two. Hence, more efficient base designs are to be implemented in EVOP for higher dimensions, yet the literature that has been devoted to efficiently dealing with this increase in dimensionality is very limited.

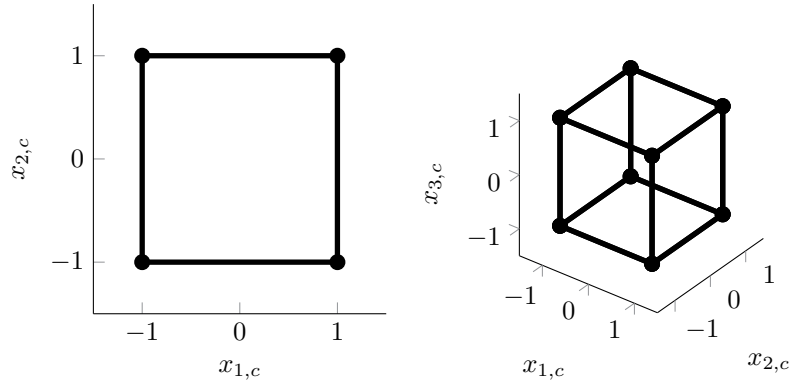


Figure 3.3: A (a) 2-factor and (b) 3-factor full factorial 2-level design.

3.2.2 Screening Designs

In classical experimental design screening designs are used to "screen" out the active (statistically significant) factors from a larger set of factors. These designs can also be used in terms of EVOP. The fractional factorial screening design will be discussed in this section.

In a fractional factorial design an informative subset of the design points of a full factorial design is used. The reduction of the number of points will inevitably lead to some loss in capabilities as compared to the full factorial design, but for the purpose of EVOP this might be acceptable.

The basic idea behind fractional factorial designs is to drop a number of measurements from a complete factorial design, in such a way that as much as possible of the balance is kept. Suppose the fraction of a three-factor, two-level design that is used corresponds to the coordinates in coded units in table 3.1 and visualized in figure 3.4.

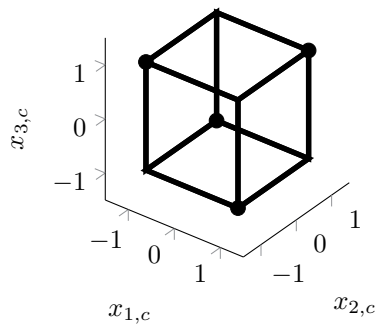


Figure 3.4: 1/2 fraction of 2^3 design.

Table 3.1: Fractional factorial: ½ fraction of 2^3 full factorial.

x_1	x_2	x_3
-1	-1	1
1	-1	-1
-1	1	-1
1	1	1

The portion of experimental measurements that is dropped from the full factorial can be deliberately chosen, and some of the often used fractional factorials in a classical DOE setting (to determine a main effects regression model, as explained in the next section) are presented in table 3.2.

Table 3.2: Often used screening designs for $k = 6, 8, 10, 12$.

# factors	Fractional factorial
6	8
8	16
10	16
12	16

The choice of the design inevitably sets the type of regression model that can be constructed, a topic that is dealt with in the next section. Also, the smaller the fraction of retained points (so the smaller the design), the more difficult it becomes to estimate a given effect.

3.2.3 Optimal Designs: $\mathcal{D}$-Optimality

Optimal designs are a class of experimental designs that are optimal with respect to some statistical criterion, such as $\mathcal{D}$-optimality. In the framework of this dissertation optimal designs are also called flexible designs since one can generate an optimal design for any number of measurements (with as limiting factor the minimal number of measurements for the model structure). Since accurate parameter estimates are essential for the estimation of the improvement direction, the $\mathcal{D}$-optimality criterion will be used in this dissertation as this criterion minimizes the generalized variance of the parameter estimates. The $\mathcal{A}$-optimality criterion would be a better choice since it seeks to minize the trace of the inverse of the information matrix and, hence, minimizes the average variance of the parameter estimates. Yet $\mathcal{D}$-optimal designs are much easier to generate, especially considering the restriction that in the long run the code should run on the machine itself.

The $\mathcal{D}$-optimality criterion will be discussed briefly by using the example of Ordinary Least Squares Regression (OLS). More relevant information on $\mathcal{D}$-optimal design can be found in other works, such as [3, 6, 42, 58]. In equation 3.1 an OLS regression model is written in matrix notation. In this equation, $\mathbf{y}$ is the $n \times 1$ vector of responses, $\boldsymbol{\varepsilon}$ the $n \times 1$ vector of random error terms which are normally distributed, $\boldsymbol{\beta}$ the $(f_t + 1) \times 1$ vector of parameter estimates and $\mathbf{X}$ is the $n \times (f_t + 1)$ design matrix, with n being the number of experiments and f_t the number of model terms excluding the intercept.

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon} \tag{3.1}$$

The theory behind $\mathcal{D}$-optimal design can be explained by starting with the OLS estimator $\hat{\boldsymbol{\beta}}$ of the parameter vector, which can be written as equation 3.2. The general principle of $\mathcal{D}$-optimal design can be derived from the variance-covariance matrix of the vector $\hat{\boldsymbol{\beta}}$, which can be written as equation 3.3 [82].

$$\hat{\boldsymbol{\beta}} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y} \tag{3.2}$$

$$\operatorname{cov}\left\{\hat{\boldsymbol{\beta}}\right\} = \sigma^2 (\mathbf{X}'\mathbf{X})^{-1} \tag{3.3}$$

From equation 3.3, one can see that if $(\mathbf{X}'\mathbf{X})^{-1}$—called the *dispersion* matrix (which is the inverse of the *information* matrix $\mathbf{X}'\mathbf{X}$)—is minimized, the variances of the parameters are minimized and an accurate model can be obtained. For $\mathcal{D}$-optimal designs, the information matrix is represented by a scalar that represents the magnitude of this matrix, in this case the determinant of the information matrix. A design is said to be $\mathcal{D}$-optimal when the parameter variances are as small as possible by arranging the experiment positions as broadly as attainable according to the criterion of the maximization of the determinant of the information matrix: $\max(|\mathbf{X}'\mathbf{X}|)$. Like with all optimality criteria, an optimal design is constructed for a specific model and might be sub-optimal for other models.

In most cases computerized algorithms, so-called exchange algorithms, are necessary to construct $\mathcal{D}$-optimal designs. The two main types of exchange algorithms will be briefly explained.

The first kind of algorithm is the *point-exchange* algorithm, also sometimes called the row-exchange algorithm, which was popularized by Fedorov [50]. A key feature of point-exchange algorithms is that they require a user-specified candidate set as input. This candidate set is the set of all possible combinations of factor levels as specified by the user. Every possible factor level combination is typically called a candidate point (or a *row* in the candidate set, hence the alternative name *row-exchange*). The point-exchange algorithms use this set of candidate points to generate a starting design of which at least a part is a randomly selected subset of the candidate set. It then tries to improve this starting design by replacing factor level combinations of the current design by candidate points. The algorithm continues exchanging factor level combinations until the determinant of the covariance matrix $\operatorname{cov}\left\{\hat{\boldsymbol{\beta}}\right\}$ can no longer be improved. The generation of a starting design and the subsequent point-wise

improvement procedure are repeated a pre-specified number of times to increase the likelihood that the best design in terms of the $\mathcal{D}$-optimality criterion is found.

In general, point-exchange algorithms have two drawbacks [42]. First, their computation time grows exponentially with the number of factors and with the number of levels considered for each of these. Second, in the presence of constraints on the factor levels, it may be difficult to produce a good candidate set. This is because all the combinations in the candidate set have to satisfy the constraints.

The second kind of algorithm, the *coordinate-exchange* algorithm, was introduced in literature by Meyer & Nachtsheim [102] and offers the advantage that it does not require the explicit specification of a candidate set. The starting design for this algorithm is determined completely at random in the experimental domain. Then, the coordinate-exchange algorithm attempts to improve the level of each factor at each run of the starting design, one by one. If a better value is found for a level of one of the factors in a run, then that level is changed to the better value. This coordinate-wise improvement is continued until none of the individual factor levels can be improved any more. As in the point-exchange algorithm, the generation of the starting design and the improvement procedure are repeated a pre-specified number of times. This algorithm often leads to the most efficient design, in comparison with the point-exchange algorithm, since it is not bounded by a given candidate set of points. However, since no candidate set of factor levels is given, the implementation of the design might be more difficult. In contrast, the point-exchange algorithm draws from a candidate set of points defined by the experimenter beforehand. The design might not become as efficient as a coordinate-exchange generated design but might be easier to implement. If there are very complex constraints on the experimental region, it might be more tedious to calculate the convex hull with related constraints when generating a design in a coordinate-exchange approach.

For the construction of the optimal designs in this dissertation a coordinate exchange algorithm is used since this does not require the explicit specification of a candidate set.

3.3 Choice Of Regression Model

Before detailing the choice of the regression model the use of coded units has to be introduced as the properties of coding will be important when calculating the move. Often, in experimental design, the units of each factor are transformed to scale the range in every factor to the interval $[-1; +1]$. By default all calculations

in the developed software are executed with these coded variables where -1 conforms to the lower bound of the factor and $+1$ to the upper bound of the factor in the *design region* (the region bounded by the factor levels in the current design). Coding the factors can be done by applying equation 3.4 in which x_d is the value of the factor to be coded, x_d^+ the maximum factor level in the design region and x_d^- the minimum factor level in the design region.

$$x_{d,c} = \frac{x_d - \frac{x_d^+ + x_d^-}{2}}{\frac{x_d^+ - x_d^-}{2}} \tag{3.4}$$

For two-level designs, the low level of the factor conforms to -1 and the high level to $+1$. If a centerpoint is included it is located at the midpoint between low and high level and is coded as 0. As mentioned above, two-level factorial designs are the basis of the EVOP scheme. This type of designs allows estimating a model with linear terms and (for a full factorial) all $(k-1)$-factor interactions. However, an experimenter is usually only interested in, at most, a model with all linear terms (main effects) and two-way interactions since the effect hierarchy principle [159] states that lower-order effects are (usually) more important than higher-order effects. A model with main effects and two-way interactions can be written as equation 3.5, with y the response, β_0 the intercept, β_d the parameter estimate of the d-th factor, $\beta_{dd'}$ the parameter estimate of the interaction between the d-th and d'-th factor and $\varepsilon \sim \mathcal{N}\left(0, \sigma^2\right)$ the noise which is assumed identical and independently distributed (i.i.d.) according to a normal distribution with mean zero and standard deviation σ.

$$y = \beta_0 + \sum_{d=1}^{k} \beta_d x_d + \sum_{d=1, d'>d}^{k} \beta_{dd'} x_d x_{d'} + \varepsilon \tag{3.5}$$

For an interaction model of k factors (main effects + two-way interactions) the number of parameter estimates t_p equals equation 3.6.

$$t_p = k + \frac{k!}{2 \cdot (k-2)!} \tag{3.6}$$

In the case of a fractional factorial, however, it is not always possible to estimate this simple model. Let us take a three factor case as an example, and consider both the full factorial (eight experimental measurements) and a half fraction of the points (four measurements, see figure 3.4). Using tables 3.3 and 3.4 the full factorial and fractional factorial designs can be compared by looking at the effects. Each row in the table represents a design point, the full factorial having

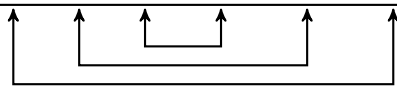
eight points (experimental measurements or factor level combinations) and the half fraction design four points. For each point, the numbers in the columns "x_1", "x_2" and "x_3" denote the coordinate of the respective point. For example, the first row represents the point with coordinates $\{-1; -1; -1\}$ for the full factorial, and $\{-1; -1; 1\}$ for the fractional factorial.

Table 3.3: Effects of full factorial 2^3 design.

x_1	x_2	x_3	x_1x_2	x_1x_3	x_2x_3
-1	-1	-1	$+1$	$+1$	$+1$
-1	-1	$+1$	$+1$	-1	-1
-1	$+1$	-1	-1	$+1$	-1
-1	$+1$	$+1$	-1	-1	$+1$
$+1$	-1	-1	-1	-1	$+1$
$+1$	-1	$+1$	-1	$+1$	-1
$+1$	$+1$	-1	$+1$	-1	-1
$+1$	$+1$	$+1$	$+1$	$+1$	$+1$

Table 3.4: Effects of ½ factorial 2^3 design.

x_1	x_2	x_3	x_1x_2	x_1x_3	x_2x_3
-1	-1	$+1$	$+1$	-1	-1
$+1$	-1	-1	-1	-1	$+1$
-1	$+1$	-1	-1	$+1$	-1
$+1$	$+1$	$+1$	$+1$	$+1$	$+1$



Alias Pattern

If one would use a main effects only model, the first three columns constitute the design matrix, and the columns of this matrix are orthogonal for both designs. If one includes also the two-way interactions, the design matrix consists of all six columns, and those are still orthogonal for the full factorial design, but this is not the case for the fractional factorial design. Indeed, the main effect of x_1 and the interaction between x_1 and x_3 are completely confounded. When two effects are confounded, we can only estimate their joint effect. This means that for the case of three factors and considering only a half fraction, only a main effects model (as presented in equation 3.7) can be fitted to the data.

$$y = \beta_0 + \sum_{d=1}^{k} \beta_d x_d + \varepsilon \tag{3.7}$$

It will be shown later that such a main effects model will be sufficient for EVOP if one uses the designs in coded form, i.e. coding the factor levels in the design region to the interval $[-1; 1]$. The amount of confounding in a fractional factorial design is denoted by the resolution of the design, which is written in roman numerals. The different resolutions and their amount of confounding are summarized in table 3.5. In general, when the resolution increases, the larger the fraction of the full factorial that has to be taken (i.e. the larger the sample size).

Table 3.5: Resolution and measure of confounding.

Resolution	**Confounding**
III	Some main effects confounded with 2-factor interactions
IV	Some main effects confounded with 3-factor interactions, some 2-factor interactions with each other
V	Some main effects confounded with 4-factor interactions, some 2-factor interactions with 3-factor interactions
VI	Main effects unconfounded by 4-factor (or less) interactions, 2-factor interaction effects unconfounded by 3-factor (or less) interactions, some 3-factor interaction effects are confounded with other 3-factor interactions

Besides the confounding pattern, it is also evident that reducing the number of experimental measurements decreases the power to detect a given effect. Since EVOP operates at a local scale, the expected effects are quite small so that power is an important issue when considering using fractional factorial designs as a base design. The power issue is discussed in detail in chapter 6 and 7 and is used to make recommendations on the type of design to use for a given dimensionality and signal-to-noise level present.

The notation for a fractional design is $rl_{Res}^{k-p_f}$ with p_f indicating that a $1/2^{p_f}$ fraction of the full factorial measurements is taken and the subscript Res denotes the resolution in roman numerals. The variable k denotes the number of factors, l the number of levels for the factor and r the number of replicates, as for the full factorial.

Summarizing: When the dimensionality increases fractional factorial designs are plausible options, and can have their merit in EVOP, but face the drawback of a loss in power and capacity to estimate interactions.

In order to determine the active factors, the regression model has to be reduced to its significant parameters. Several options exist for model reduction such as all possible regression (also called all possible subsets regression) in which every

possible model is estimated from one parameter up to an arbitrary number of parameters (arbitrary up to a maximum number of pre-defined parameter estimates, without counting the intercept). Using one, or a combination of, selection criteria the best model is chosen.

Another approach is the use of stepwise regression in which parameters are iteratively added or removed from the model to increase the explanatory power of the model. In stepwise regression the model is fit iteratively and, as a result, has the possibility to lead to different final models which may not all perform equally well. Depending on the terms included in the initial model and the order in which parameters are moved in and out, the method may build different models from the same potential parameter set. The method terminates when no single step improves the model. There is no guarantee, however, that a different initial model or a different sequence of steps will not lead to a better fit. In this sense, stepwise models are locally optimal, but may not be globally optimal. In all possible regression one does not have this model dependency on the iteration since all possible models are fitted. However, since all combinations of models have to be fitted, all possible regression is computationally much more expensive then stepwise regression and might not be plausible to implement in industrial hardware. Therefore stepwise regression is used to reduce the model to its significant parameters. A stepwise approach with *bidirectional* elimination is used, which means that in every step of the algorithm tests are performed for variables to be in- or excluded.

Matlab®'s *stepwisefit* procedure is used with a $p_{enter} = 0.05$ and $p_{remove} = 0.1$ and as initial model the full main effects model is selected. This procedure in Matlab® 2010 begins with the initial model (in this case the full linear model is used as the initial model) and compares the explanatory power of incrementally larger and smaller models. At each step of the algorithm, the p-value of an F-statistic is computed to test models with and without a potential term:

The following hypothesis test is used for every term β:

$$H_0 : \beta = 0$$

$$H_a : \beta \neq 0$$

If the term is not in the model and there is sufficient evidence to reject the null hypothesis ($p_{enter} < 0.05$), the term is added to the model. If the term is in the model and there is insufficient evidence to reject the null hypothesis ($p_{remove} > 0.1$), the term is removed from the model. The algorithm proceeds iteratively as shown in the pseudo-code in algorithm 3.1.

Algorithm 3.1. Matlab® 2010 stepwise regression pseudo-code.

1. Fit the initial model.
2. If any parameter not in the model has a p-value less than p_{enter} (that is, if it is unlikely that they would have zero coefficient if added to the model), add the one with the smallest p-value and repeat this step; otherwise, go to step 3.
3. If any parameter in the model has a p-values greater than p_{remove} (that is, if it is unlikely that the hypothesis of a zero coefficient can be rejected), remove the one with the largest p-value and go to step 2; otherwise, end.

3.4 Calculation Of The Move

Calculating the move after a phase has been concluded consists of two parts: (1) estimating the direction and (2) normalizing the step size.

The estimation of the direction to move follows directly from the model fitted to the data. The basic idea is to move in the direction of the optimum using the partial derivatives. When including the main effects and the two-way interactions in the estimated model, the partial derivative of this model in dimension d includes the main effect β_d and the terms $\beta_{dd'}x_{d'}$ for every two-way interaction in which x_d is present.

If those calculations are based on the coded design, i.e. in the range $[-1; 1]$, and if the steepest ascent is defined in the centre of the design (so in $\{0, \ldots, 0\}$), the interaction terms $\beta_{dd'}x_{d'}$ vanish in the partial derivatives and a simple main effects regression suffices as a base model. All main effects that are not included in the model based on the stepwise selection (see higher) have a partial derivative of zero. The vector which comprises these partial derivatives determines the direction of the move. If a move is executed by shifting the centre of the design, one wishes to have control over the length of the vector between the two consecutive centres. In order to do so, the partial derivatives are normalized and, since the partial derivatives of the main effects model are equal to the parameter estimates or betas, these are called the normalized betas. Afterwards, these normalized betas can be multiplied with a step length δ_{EVOP} after decoding, or a coded step length $\delta_{EVOP,c}$ before decoding back to the full range of the factors.

The step length $\delta_{EVOP,c}$ can be set by the experimenter and either made variable or be fixed. For the current approach, it was chosen to ensure that, when all

regression coefficients are of equal importance, a corner of the old and new design region will coincide as is shown in figure 3.5 for two factors.

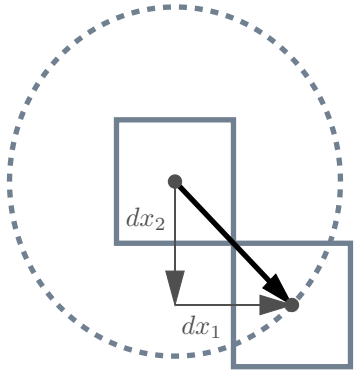


Figure 3.5: EVOP stepsize for 2 factors.

Defined this way, the step length δ_{EVOP} depends on the dimensionality of the problem k and is expressed, in coded units, by equation 3.8, which is an application of Pythagoras' theorem. In this equation f_a is the number of main effects in the reduced model, which is at most equal to the number of factors k.

$$\delta_{EVOP,c} = \sqrt{2^2 f_a} = 2\sqrt{f_a} \tag{3.8}$$

In equation 3.9, the coded step size vector $\boldsymbol{\delta}_{\mathbf{EVOP,c}}$ is written as a function of the normalized betas and its predetermined vector length $\delta_{EVOP,c}$. In this equation $\boldsymbol{\beta}$ is the $1 \times k$ vector of the parameter estimates.

$$\boldsymbol{\delta}_{\mathbf{EVOP,c}} = 2\sqrt{f_a} \frac{\boldsymbol{\beta}}{\sqrt{\sum_{d=1}^{k} \beta_d^2}} \tag{3.9}$$

The coded, normalized step size vector can be rescaled to uncoded values and is used to calculate the coordinates of the new design points. The steepest ascent delta's can be decoded by using equation 3.10, in which $\delta_{EVOP,c,d}$ is the steepest ascent delta in the d-th direction.

$$\boldsymbol{\delta}_{\mathbf{EVOP}} = \left\{ \delta_{EVOP,c,d} \cdot \frac{x_d^+ - x_d^-}{2} \right\}_{d=1}^{k} \tag{3.10}$$

By adding $\boldsymbol{\delta_{EVOP}}$ to the uncoded coordinates of the centre of the current design region, the centre of the new design region is determined.

3.5 Evolutionary Operation Steepest Ascent

Evolutionary Operation Steepest Ascent is the automatic implementation of the steepest ascent procedure as described in many statistical textbooks. To the author's best knowledge an application of steepest ascent within the framework of the EVOP methodology has not been published before although such exploration was already proposed by Box in the first publication about EVOP [19].

EVOPSA works on the same principles as the EVOP algorithm but, instead of estimating a new steepest ascent direction after every optimization phase, such a direction is estimated once, after which a line search in the direction of steepest ascent is performed. A new measurement point is placed at a distance δ_{EVOP} from the centre of the first design. If the response value at this point is more desirable than the maximum response value of the previously acquired points a new measurement point is placed, again at a distance δ_{EVOP}, in this direction. This procedure is repeated until a response value is worse than the previously acquired one. In such a case, a new statistical design is executed around the previous measurement and a new path of steepest ascent calculated, after which this procedure is repeated. EVOPSA combines two interesting properties of EVOP and Simplex: (1) estimation of the direction based on a statistical design, (2) sequential augmentation with only a limited amount of measurements in the direction of steepest ascent.

An example of a two-factor EVOPSA optimization is plotted in figure 3.6 and the flowchart for this algorithm can be found in figure 3.7.

3.6 Dealing With The Borders Of The Experimental Domain

The experimental domain is usually constrained by boundaries, be it either thresholds that certain factors cannot exceed due to process limitations—so-called single-factor boundaries—or more complex constraints in which certain combinations of factor levels are infeasible. The focus lies on single-factor min-max boundaries, which can be seen as a hypercube that defines the experimental domain.

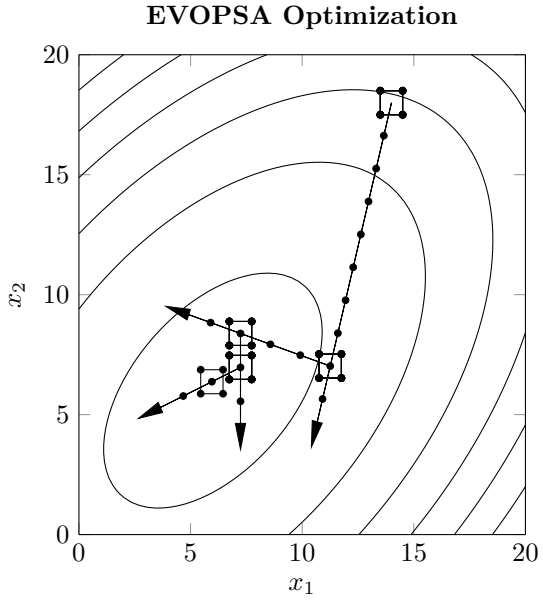


Figure 3.6: Example of 2-factor EVOPSA improvement.

Should a move bring the new design region outside of the experimental domain, a decision has to be made on how to constrain the direction of steepest ascent to the experimental domain. If the addition of $\delta_{EVOP,d}$ in the d-th dimension should lead to a factor setting that exceeds this factors boundary constraints, this part of the steepest ascent vector is set to zero, as illustrated for two factors in figure 3.8.

3.7 EVOP Software Implementation

3.7.1 Settings

Certain parameters have to be set when starting the software to be able to run an EVOP scheme. They will be discussed here.

Number of factors k: The number of factors that will be taken into the EVOP scheme, this is important for building the design.

Type of design: A string indicating the design that should be built, currently

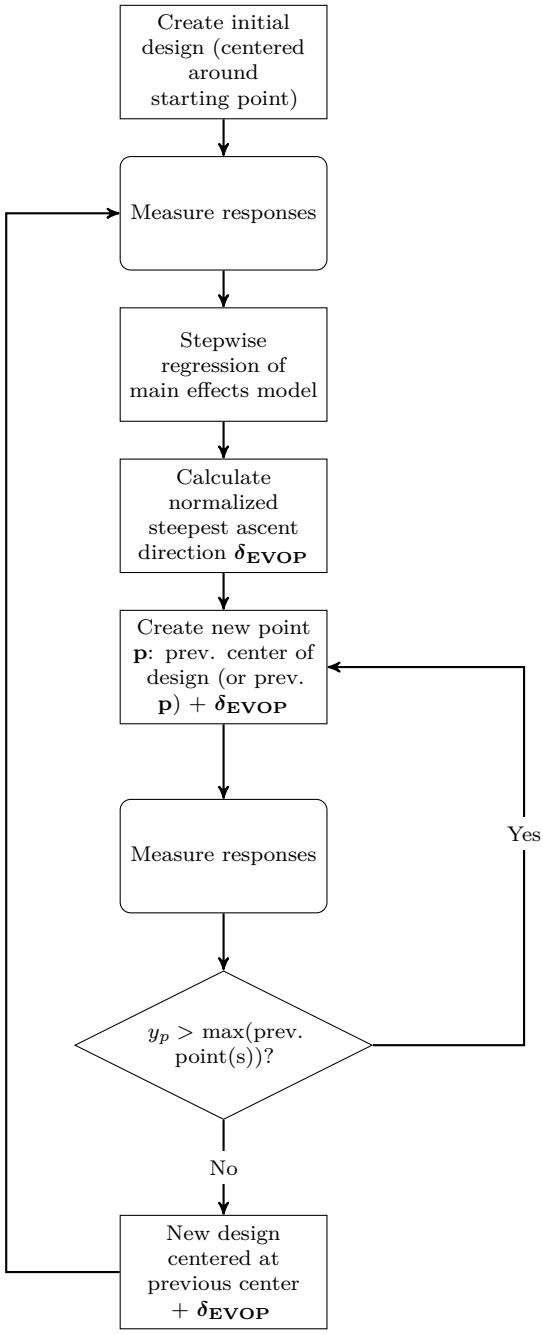


Figure 3.7: Flowchart EVOPSA algorithm.

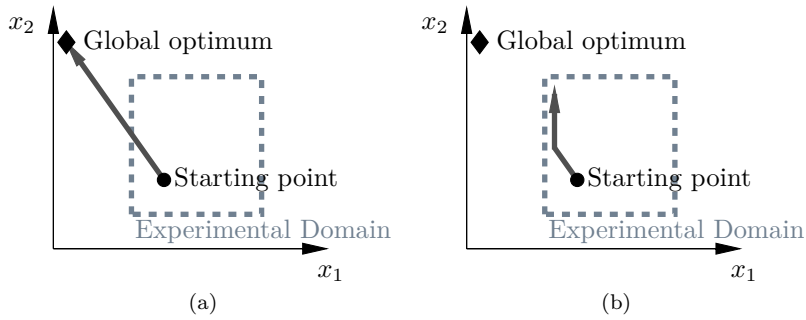


Figure 3.8: A 2-factor example of dealing with boundary constraints in EVOP, (a) Unconstrained improvement path, (b) Improvement path constrained to experimental region.

"FullFactorial" or "FracFactorialX", in which the X in the fractional factorial string is replaced by the numeric value for the resolution of the design. The resolution must be at minimum 3 to make a useful design.

Replication r: The number of times the design needs to be replicated. Minimum is one, as the design needs to be measured once.

Factorstep $\mathbf{dx}$: a $1 \times k$ vector that denotes the maximum perturbation size (in original, uncoded units) in every dimension (i.e. the maximum change in factor levels in one dimension), see figure 3.9.

Starting point $\mathbf{x_{start}}$: a $1 \times k$ coordinate vector denoting the reference condition, i.e. the point around which the first design region will be centred. The initial design region will form a hypercube which borders are defined by the coordinates $\{x_{start,d} \pm 0.5 \cdot dx_d\}_{d=1}^{k}$.

Boundaries of the experimental domain $\mathbf{X_{lim}} = \left[\mathbf{x_{lim}^-}; \mathbf{x_{lim}^+}\right]$: a $k \times 2$ matrix with the lower (first column $\mathbf{x_{lim}^-}$) and upper (second column $\mathbf{x_{lim}^+}$) bounds on every factor.

The combination of the factorstep $\mathbf{dx}$ and starting point $\mathbf{x_{start}}$ defines the initial design region. The design points of a fractional or full factorial design will always be located at the cube points of the hypercube that defines the design region. Figure 3.9 shows the design region for a two-factor example with the design points of a full factorial, which are located at the cube points.

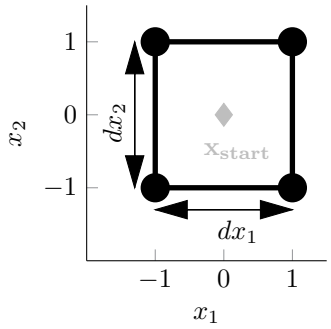


Figure 3.9: EVOP design region, example for 2-factor full factorial.

3.7.2 Pseudo-Code

The base designs are generated using Matlab® algorithms: *ff2n* to generate full factorial designs and *fracfactgen* in combination with *fracfact* to generate fractional factorial designs. The pseudo-code for EVOP is presented in algorithm 3.2.

In appendix A an example is presented on how to run an EVOP improvement using the developed Matlab® software package.

3.8 EVOPSA Software Implementation

3.8.1 Settings

The settings for the EVOPSA algorithm are exactly the same as for the EVOP algorithm. Programmatically, the EVOPSA class is a child of the EVOP class and inherits all its properties from its parent.

3.8.2 Pseudo-Code

The EVOPSA pseudo-code is presented in algorithm 3.3. Since EVOPSA is programmatically a child class of the EVOP algorithm, all functions to generate the designs are exactly the same as for EVOP.

In appendix A an example is presented on how to run an EVOPSA improvement using the developed Matlab® software package.

Algorithm 3.2. EVOP Pseudocode.

Require: Number of factors k, number of replications r, factorstep $\mathbf{dx}$, design type "FullFactorial"/"FracFactorialX", starting point $\mathbf{x_{start}}$, Boundaries of experimental domain $\mathbf{X_{lim}}$

1. Generate and randomize measurements of the initial design $\mathbf{X}_0$. Save coordinates of the center of the design region $\mathbf{x_c} = \mathbf{x_{start}}$;
2. Measure responses;
3. Build main effects model and reduce with stepwise regression to significant terms. Extract the $\hat{\boldsymbol{\beta}}$-vector, a $1 \times k$ vector of parameter estimates (excluding the intercept). Effects not in the model have a parameter estimate of zero.
4. Calculate $\boldsymbol{\delta}_{\mathbf{EVOP,coded}}$ and transform to $\boldsymbol{\delta}_{\mathbf{EVOP}}$;
5. Check if new design region is within boundaries of experimental domain;

for i from 1 to k **do**
  if $(x_{c,i} + \delta_{EVOP,i} + 0.5 \cdot dx_i) > x^{+}_{lim,i}$ **then**
    $\delta_{EVOP,i} = 0$
  else if $(x_{c,i} + \delta_{EVOP,i} - 0.5 \cdot dx_i) < x^{-}_{lim,i}$ **then**
    $\delta_{EVOP,i} = 0$
  end if
end for

6. Generate and randomize the measurements of a new design $\mathbf{X}$ in design region centered at $\mathbf{x_c} = \mathbf{x_c} + \boldsymbol{\delta}_{\mathbf{EVOP}}$ and go to step 2.

3.9 Conclusions

In this chapter Online Sequential Process Improvement methods were adapted and described for the contemporary setting. The original Traditional EVOP algorithm discussed in chapter 2 was adapted to be run automatically on a computer and the specific calculations for every subsequent step were shown. Evolutionary Operation Steepest Ascent, a combination of EVOP and a line search is formally described as well. Since it is a subsidiary of EVOP, all formal calculations remain the same, only some decision rules were adapted. Furthermore, a novel method to deal with the borders of the experimental domain for EVOP and EVOPSA was presented. To conclude, the pseudocode for the software implementation for both algorithms was given and these algorithms will be used throughout this dissertation for EVOP and EVOPSA.

Algorithm 3.3. EVOPSA Pseudocode.

Require: Number of factors k, number of replications r, factorstep $\mathbf{dx}$, design type "FullFactorial"/"FracFactorialX", starting point $\mathbf{x_{start}}$, Boundaries of experimental domain $\mathbf{X_{lim}}$

1. Generate and randomize measurements of the initial design $\mathbf{X}_0$. Save coordinates of the center of the design region $\mathbf{x}_c = \mathbf{x_{start}}$;
2. Measure responses $\mathbf{y_J}$ for this iteration J;
3. Build main effects model and reduce with stepwise regression to significant terms. Extract the $\hat{\boldsymbol{\beta}}$-vector, a $1 \times k$ vector of parameter estimates (excluding the intercept). Effects not in the model have a parameter estimate of zero;
4. Calculate $\boldsymbol{\delta}_{\mathbf{EVOP,coded}}$ and transform to $\boldsymbol{\delta}_{\mathbf{EVOP}}$;
5. Save responses of this iteration in vector $\mathbf{y_{J-1}} = \mathbf{y_J}$
6. Generate coordinates of new point $\mathbf{p} = \mathbf{x_c} + \boldsymbol{\delta}_{\mathbf{EVOP}}$ and check if new design point is within boundaries of experimental domain;

for i from 1 to k **do**
  if $(p_i + \delta_{EVOP,i}) > \mathbf{x}^{+}_{\mathbf{lim,i}}$ **then**
    $\delta_{EVOP,i} = 0$
  else if $(p_i + \delta_{EVOP,i}) < \mathbf{x}^{+}_{\mathbf{lim,i}}$ **then**
    $\delta_{EVOP,i} = 0$
  end if
end for

6. Measure response $\mathbf{y_J}$ at point $\mathbf{x}_c = \mathbf{x_c} + \boldsymbol{\delta}_{\mathbf{EVOP}}$
7. Compare response $\mathbf{y_J}$ to maximum of previous response(s) $\mathbf{y_{J-1}}$

if any element of $\mathbf{y_J} \geqslant \max(\mathbf{y_{J-1}})$ **then**
  Go to step 5;
else
  Go to step 8;
end if

8. Save responses of this iteration in vector $\mathbf{y_{J-1}} = \mathbf{y_J}$;
9. Generate and randomize the measurements new design $\mathbf{X}$ in design region centred at $\mathbf{x_c} = \mathbf{x_c} - \boldsymbol{\delta}_{\mathbf{EVOP}}$ and go to step 2.

An iteration J in the sense of this code is started every time (a set of) responses are measured, thus in step 2 and step 6.

4 Choosing An Appropriate Starting Point When No Prior Information Is Available

4.1 Introduction

In industry, achieving a constant, high quality of the end product at reasonable cost is essential and requires finding the optimal factor levels of the process that will achieve such quality. This, in turn, requires knowledge about the process itself which can be based on historical insight and experience or, in its most desirable form, on a process model that relates the response (the quality of the end product) to the factors (input settings of the process). There are several procedures to construct such a process model, such as first principles models and data-driven models built with Response Surface Methodology (RSM), as explained in the introduction.

As explained there, when using first principles models a plant-model mismatch can occur by incorrect scaling of the factors or failing to capture a source of variation. This results in a shift between the estimated optimum, as determined by the model, and the true process optimum. Similarly, for RSM-modelling, or data-driven modelling in general, it is hard to include all potential variation into the model so that a mismatch can occur. Although a mismatch is likely, in practice the result of these models is accepted since it already provides an improvement when compared to the situation before the experimentation started.

Results section submitted for publication, under review in *Applied Stochastic Models in Business and Industry*

In order to further improve the process, fine-tuning the factor levels of the process to reflect more accurately the true optimum can yield additional benefit, but this follow-up experimentation should be executed on the process itself which poses serious challenges and restrictions. In order to reduce the cost, Online Sequential Process Improvement (OSPI) was introduced, to have a minimal risk of producing unacceptable output products, i.e. the product should be still saleable at any time.

One of the main disadvantages of OSPI methods—due to the sequential nature of small changes in the factor levels—is its proneness to get trapped in local optima. As a result, having available a good prior estimate of the global optimum as a starting point is an essential aspect. In most applications, prior knowledge about the process is available and the current best settings can be used as a starting point for OSPI. However, when such a prior is not available, experimentation and modelling on the full scale process is still required, with the higher-mentioned RSM approach as a potential solution.

Yet RSM is based on classical designs and (regression) models inherently posing limitations for modelling complex processes. Those designs operate at only two (in case of a full or fractional factorial design) to five (such as a Central Composite Design) pre-defined factor levels which may not be sufficient to model complex process behaviour over the full experimental domain. As a consequence, those designs are not ideal candidates since their use is limited to fitting simple polynomial models and might not capture the global optimum in the presence of local optima and high non-linearity of the process. Indeed, if the non-optimal coverage of the design space excludes points located close to the global optimum it might be missed completely. The evident question then is how to pinpoint the global optimum with a limited experimental effort spanning a sufficiently broad class of models, potentially non-linear and having local optima.

In this text, the use of so-called space-filling designs, which were popularized in the field of the Design and Analysis of Computer Experiments (DACE) where fitting complex models that stretch beyond polynomials are common sense, is proposed. In the same field of DACE, one does typically not use the simple polynomial models to fit the complex process behaviour. Instead, Gaussian Process (GP) modelling is used as a flexible framework for fitting a broad class of functions.

GP modelling has its origins within the field of geostatistics and borrows its ideas from a method called Kriging. This method was originally developed by the French mathematician Georges Matheron [94] based on the master's thesis of Danie G. Krige, after whom the method is named. In geostatistics, the spatial correlation of considered points is an important aspect which is

explicitly taken into account. Take as an example soil samples that are collected a certain distance from one another. The composition of these soil samples is not independent from one another but rather correlated with the distance between the sampled points—soil characteristics at locations close to each other do have a tendency to be similar, whilst the properties of remote locations are less related. Given this specific setting the general idea behind the method is to interpolate the value of the response (e.g. soil composition) at an unobserved location from observations of its value at nearby locations (the sample points). This very same idea forms the basis of GP modelling as introduced by O'Hagan & Kingman in 1978 [113] in fields other than geostatistics. This was further developed to the concept of DACE by Sacks et al. [137] in 1989. Gaussian Processes gained much interest during the late 1990's, especially when the concept was introduced to the machine learning community in 1996 by Williams & Rasmussen [157] since the rapid advance of computational power helped facilitate the implementation of GP modelling for larger data sets [24, 75, 80, 138]. GP modelling has already been used to model complex industrial [29] and chemical [148, 162] processes with the focus to accurately predict the output at untested factor levels and the development of adaptive soft sensors [61]. The authors of these publications showed that the flexibility of GP for modelling such processes has clear advantages over the classical RSM approach. In those cases where the response behaviour is complex, it is evident that the number of design points that is required to approximate the surface in every part of the experimental domain can become quite substantial. For the purpose of providing a good starting point for OSPI methods such as Evolutionary Operation (EVOP), however, it is not required to approximate the full process dynamics in the experimental domain, limiting the number of experimental measurements to perform.

Apart from EVOP, one might use the Efficient Global Optimization (EGO) approach, proposed by Jones *et al.* [75], that can be sequentially executed in conjunction with GP modelling. This has not been done in this text for two main reasons: (1) The EGO algorithm places additional sequential samples by balancing the requirement of finding the optimum and sampling in regions in which the prediction error is high. It is not guaranteed that samples taken in regions where the prediction error is high, will result in saleable product (although it could be adjusted to deal with this). (2) EGO combined with GP modelling is computationally expensive, especially when the sample size increases, and it is assumed that—once real production is started—executing algorithms that are computationally expensive might not be desired in a large part of the process industry.

As its name suggests, the goal of space-filling designs is to spread out the design points as evenly as possible over the whole experimental domain. This can be

achieved in several ways leading to design types that are labelled as uniform designs, maximum entropy designs, Latin Hypercube designs or sphere-packing designs. For more detail about different types of space-filling designs, the reader is referred to the book by Santner *et al.* [138]. Latin Hypercube designs (LHC) are the most used ones, and were first proposed in 1979 as one of the first designs for computer experimentation [99]. Their popularity is mainly due to their good space-filling properties and the fact that they are computationally inexpensive to generate.

In view of the above, the goal of this text is to provide a strategy for finding an efficient starting point for online improvement methods when no prior information is available, based on space-filling designs followed by Gaussian Process modelling. In order to evaluate the performance of this approach, two test cases are defined. In the first case a simulation study is performed on a non-linear test function (Ackley function) having a global optimum as well as local optima. During the simulation the amount of noise is varied and the effect on pinpointing the optimum quantified. The proposed methodology is compared with an RSM approach in which a classical Central Composite Design (CCD) combined with a polynomial model is used to estimate the optimum. To test how much the design influences the behaviour of the Ordinary Least Squares (OLS) regression used in RSM, OLS regression is also applied to the data from the space-filling LHC designs.

The second case deals with a lab scale setup in which rolling element bearing behaviour is tested. The specific focus of this case is the application of static forces to reduce bearing displacement when under a dynamic radial load, or to pin-point regions where resonant frequencies occur. In the simulation case the Euclidean distance between the estimated and true optimum can be quantified and used to assess the methodology, in the bearing case the true optimum is not known and an EVOP improvement is started to fine-tune the factors.

4.2 Statistical Designs & Methods

4.2.1 The Latin Hypercube Design

Latin Hypercube (LHC) designs are a popular class of space-filling designs, and will be used here. The LHC design is based on a generalization of the well-known Latin Square. A Latin Square is an $n \times n$ array consisting of n different symbols, each occurring exactly once in each row and exactly once in each column. An example of a 7×7 Latin Square is given in figure 4.1.

D	G	A	F	B	C	E
E	A	B	G	C	D	F
F	B	C	A	D	E	G
G	C	D	B	E	F	A
A	D	E	C	F	G	B
B	E	F	D	G	A	C
C	F	G	E	A	B	D

Figure 4.1: 7×7 Latin Square.

A Latin Square in the framework of a space-filling design can generate a two-factor design. To generate this design one factor is attributed to the column property, and one to the row property. As such, the row as well as the column factor is divided into equally spaced intervals (seven in the example of figure 4.1). If one now selects one of the symbols from this Latin Square, say G, a total of n points in the experimental domain is selected. Those points then form the different experimental measurements that are to be executed. Evidently, for a given number of bins, n different Latin Squares can be obtained, some having better space-filling properties than others. The worst choice is to consider a reduced Latin Square in which there is a natural order in the symbols used, such as is represented in figure 4.2a. A randomly selected order for each row allows for better space-filling properties, as is shown in figure 4.2b where the grey ("G") symbols do span the experimental region well compared to the logical row permutation in figure 4.2a.

The space-filling properties of a design are measured by the *discrepancy*, which compares the uniformity of the design against the uniform distribution (the more uniform, the better the space-filling properties). So one needs an additional criterion for producing Latin Squares with appealing (space-filling) properties. There exist several criteria but the one used in constructing the designs in this text is the maximin criterion which tries to maximize the minimum distance between design points while maintaining even spacing between the factor levels.

A Latin Hypercube is a generalization of a Latin Square to an arbitrary number of dimensions. In view of the above, an LHC design can be generated using a well-chosen Latin Hypercube in a given dimension. The general procedure for constructing an LHC design of size n given k continuous, independent factors can be summarized as:

A	B	C	D	E	F	G
B	C	D	E	F	G	A
C	D	E	F	G	A	B
D	E	F	G	A	B	C
E	F	G	A	B	C	D
F	G	A	B	C	D	E
G	A	B	C	D	E	F

(a)

D	G	A	F	B	C	E
E	A	B	G	C	D	F
F	B	C	A	D	E	G
G	C	D	B	E	F	A
A	D	E	C	F	G	B
B	E	F	D	G	A	C
C	F	G	E	A	B	D

(b)

Figure 4.2: 2-factor, 7-run, Latin Square Design: (a) natural ordering and bad space-filling properties, (b) no natural ordering, better space-filling properties.

1. Divide domain of each factor into n intervals;
2. Construct an $n \times k$ matrix $\mathbf{\Pi}$ whose columns are different randomly selected point permutations of the intervals $\{1, \ldots, n\}$;
3. Each row of $\mathbf{\Pi}$ corresponds to a cell in the hyper-rectangle induced by the interval partitioning from step 1. Sample one point from each of these cells. (This implies that, in an LHC design, there are as many factor levels as there are measurements).

It is important to note that—since the LHC design is a computer-generated design with randomly selected point permutations—constructing a new LHC design can lead to a different placement of the design points, therefore—in the simulation study—multiple LHC designs will be generated. The JMP® software (version 11, The SAS Institute, Inc., Cary, NC, USA) is used to generate the LHC designs in this text. For more information about the generation of LHC designs, the reader is referred to [80, 138].

4.2.2 The Gaussian Process Model

Gaussian Process modelling [24, 123, 157] is a terminology popularized in the field of computer experimentation, and the model as discussed here is related to Simple Kriging. The Kriging model [74, 75, 138] can be thought of as a two component model, with a linear model part and a departure part: $Response = LinearPart + Departure$.

The first component (the linear part) consists of a general linear model while the second component is treated as the realization of a stationary Gaussian random field, denoted $\mathcal{Z}$. Define now $\mathbf{S} = [-1, 1]^k$ to be the normalized design space and let $\mathbf{X} \in \mathbf{S}$ be a scaled $n \times k$-dimensional matrix of input values (design points). The Kriging approach models the associated response $\mathbf{y}(\mathbf{X})$ as equation 4.1.

$$\mathbf{y}(\mathbf{X}) = \sum_{i=1}^{k} \beta_i \cdot h_i(\mathbf{X}) + \mathcal{Z}(\mathbf{X}) \tag{4.1}$$

In the above equation the h_i's are known fixed functions, the β_i's are unknown coefficients to be estimated and $\mathcal{Z}(\mathbf{X})$ is a stationary Gaussian random process with:

$$\mathrm{E}[\mathcal{Z}(\mathbf{X})] = 0 \tag{4.2}$$

$$\mathrm{Corr}[\mathcal{Z}(x_i), \mathcal{Z}(x_j)] = R(x_j - x_i) \tag{4.3}$$

In other words, the correlation ($\sim$ dependence) of the Gaussian random function evaluated at the two design points x_i and x_j depends on the distance function $R(\cdot)$. This is why the term spatial correlation is often used, and differentiates Kriging from OLS regression for which the departures (error terms) are supposed to be independent and identically distributed (i.i.d.). It turns out that modelling this spatial correlation using a distance function is so powerful that the stochastic process model in equation 4.1 can be simplified to equation 4.4:

$$\mathbf{y}(\mathbf{X}) = \mu + \mathcal{Z}(\mathbf{X}) \tag{4.4}$$

where the design matrix $\mathbf{X}$ is an $n \times k$ matrix, k denotes the dimensionality of the input space and n the number of observations; μ is the mean of the stochastic process and $\mathcal{Z}(\mathbf{X})$ the Gaussian Random field. The response $\mathbf{y}$ is represented as an $n \times 1$ vector with mean $\mathbf{1}_n\hat{\mu}$ (also an $n \times 1$ vector where $\mathbf{1}_n$ denotes an n-vector of ones) and variance-covariance matrix equal to equation 4.5.

$$\mathrm{Var}(\mathbf{y}) = \sigma^2\mathbf{R} \tag{4.5}$$

where σ^2 denotes the process variance and $\mathbf{R}$ an $n \times n$ spatial correlation matrix. The correlation matrix is modelled using a product distance function which relates certain properties (dependent on the used product distance function) to the correlation structure. Several forms of this function are proposed (see [138]), of which the Gaussian and Cubic correlation function are popular choices. The Gaussian correlation function is appropriate if the simulation or output is known to be analytic and is often the preferred choice in computer simulation studies and takes into account all other points while calculating the correlations at a specific point. This behaviour might be unwanted in physical processes were the presence of noise could lead to incorrect conclusions about the correlation structure by including all measurements. In the context of optimizing real-life processes, the cubic correlation structure is advisable since it does not take into account all other data-points when calculating the correlations at a specific point; only those within a certain distance are considered which reduces the risk of an inappropriate model due to outliers in the data. In the real-life situation, the underlying model structure is unknown, therefore the cubic correlation function is also used on the Ackley function.

If the cubic correlation function is used, an element R_{ij} of this cubic correlation matrix is given by equation 4.6.

$$R_{ij} = \prod_{k=1}^{n} \rho\left(d_{ij,k}; \theta_k\right) \tag{4.6}$$

with $d_{ij,k} = x_{ik} - x_{jk}$ the distance between points i and j in dimension k where:

$$\rho\left(d_{ij,k}; \theta_k\right) = \begin{cases} 1 - 6\left(d_{ij,k}\theta_k\right)^2 + 6\left(\left|d_{ij,k}\right|\theta_k\right)^3, & \left|d_{ij,k}\right| \leq \frac{1}{2\theta_k} \\ 2\left(1 - \left|d_{ij,k}\right|\theta_k\right)^3, & \frac{1}{2\theta_k} < \left|d_{ij,k}\right| < \frac{1}{\theta_k} \\ 0, & \frac{1}{\theta_k} < \left|d_{ij,k}\right| \end{cases} \tag{4.7}$$

where in this notation $\theta_k \geq 0$ so that if $\theta_k = 0$, the correlation is 1 across the range of the k-th factor (i.e. the surface is flat in that direction and the factor has no influence).

The JMP® software is used to estimate the GP models in this text. The notation presented here is the one used in the JMP® implementation which differs from most literature where θ_k tends to infinity if the correlation is 1, i.e. in the formulation as defined here θ_k is the reciprocal of the parameter commonly used in literature. This definition is preferred by the author so that it intuitively coincides with what we classically have as regression parameters—a value of zero meaning no influence of the considered parameter—and the use of

the cubic correlation function in this aspect can be considered a generalization of a cubic spline.

The Best Linear unbiased Predictor (BLUP) is given by equation 4.8 [74, 75].

$$\hat{y}\left(\mathbf{x}_*\right) = \hat{\mu} + \mathbf{r}'\left(\mathbf{x}_*, \hat{\boldsymbol{\theta}}\right) \cdot \mathbf{R}^{-1}\left(\mathbf{X}, \hat{\boldsymbol{\theta}}\right) \cdot \left(\mathbf{y}\left(\mathbf{X}\right) - \mathbf{1}_n \hat{\mu}\right) \tag{4.8}$$

With $\hat{y}\left(\mathbf{x}_*\right)$ the estimated response at unobserved coordinates $\mathbf{x}_*$; and $\hat{\sigma}$, $\hat{\mu}$ and $\hat{\boldsymbol{\theta}}$ the maximum likelihood estimates of σ, μ and $\boldsymbol{\theta}$; $\mathbf{r}$ an $n \times 1$ vector of estimated correlations of the unobserved responses $\hat{y}\left(\mathbf{x}_*\right)$ at new coordinates $\mathbf{x}_*$ based on the observed data points $\mathbf{y}\left(\mathbf{X}\right)$. That is, element i of $\mathbf{r}$ is $r_i\left(\mathbf{x}_*\right) \equiv \mathrm{Corr}\left[\mathcal{Z}\left(\mathbf{x}_*\right), \mathcal{Z}\left(\mathbf{X}_i\right)\right]$.

The BLUP above interpolates exactly through the response values at the design points and is suited for deterministic functions or computer simulations. However, when noise is present, it is not desirable that the GP model interpolates through the response values but rather includes an estimate of the noise. This is possible by estimating a so-called nugget parameter γ which introduces a ridge parameter in the correlation matrix as shown in equation 4.9, where $\mathbf{I}$ is the $n \times n$ identity matrix. This ridge parameter allows to deal with additive Gaussian white noise.

$$\mathbf{R}_\gamma = \mathbf{R} + \gamma \mathbf{I} \tag{4.9}$$

4.2.3 Response Surface Methodology And Ordinary Least Squares Regression

Response Surface Methodology uses OLS regression to fit polynomial models to data collected from well-chosen designs. As stated in the introduction, the type of polynomial model depends on the number of factor levels in the design. In classical RSM a quadratic polynomial model is often fitted to the data when a minimum or maximum is expected. In this text, a rotatable CCD design is used which has five factor levels and a number of measurements equal to $r\left(2^k + 2 \cdot k\right) + n_c$ with k the number of factors under investigation, n_c the number of centerpoints and r the number of times the design points (save the centerpoints) are replicated. More information about RSM and the design can be found in many excellent references and textbooks, such as [104, 107, 112]. A representation of the placement of the design points for a two-factor rotatable CCD is given in figure 4.3.

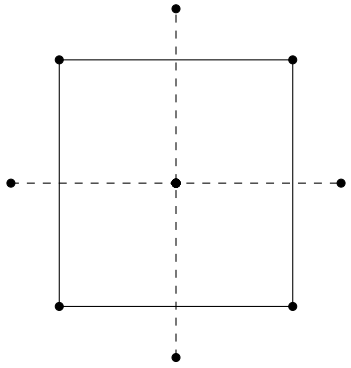


Figure 4.3: 2-factor rotatable CCD.

In general, a second-order design is fitted which can be written for k factors as equation 4.10.

$$y = \beta_0 + \sum_{d=1}^{k} \beta_d x_d + \sum_{d=1, d'>d}^{k} \beta_{dd'} x_d x_{d'} + \sum_{d=1}^{k} \beta_{dd} x_d^2 + \varepsilon \tag{4.10}$$

With y the response, β_0 the intercept, β_d the regression coefficient of the d-th factor, $\beta_{dd'}$ the regression coefficient of the interaction between the d-th and d'-th factor and β_{dd} the regression coefficient for the pure quadratic of the d-th factor and ε the random error term which is assumed i.i.d. following the normal distrubution with mean zero and error standard deviation σ, $\varepsilon \sim \mathcal{N}\left(0, \sigma^2\right)$. In the simulations we did not opt for reducing the full quadratic model using selection procedures (e.g. stepwise, all possible subsets, etc) because the results of the study would then depend on the chosen procedure, which is undesirable. The impact of this should be modestly small since the non-significant terms will have small coefficients.

In a normal RSM procedure one would first use a screening design to select the most significant factors, then build a linear regression model (using for instance a full factorial with centerpoints) and—if a lack-of-fit test indicates there is a lack of fit—a second-order design might be run (or the full factorial is augmented to a Central Composite Design) to estimate a regression function which includes pure quadratics. However, this is out of scope for the Ackley function as one knows that curvature is present within the model. Indeed, the full second-order model will not be able to capture the full function dynamics. It is of interest whether this second-order model captures sufficiently the region of the optimum.

4.3 Simulation Study

The simulation study is performed on a non-linear test function that has several local minima in the region considered. A two-dimensional function was chosen to be able to visualize the optimization results.

4.3.1 Ackley Function

The Ackley function is a multi-dimensional test function proposed in 1987 [2], that has six parameters that control its shape. A two-dimensional Ackley function (equation 4.11) is used, with shape parameters $a_1 = 0.8, a_2 = 20, a_3 = 0.2, a_4 = 2, a_5 = 2\pi, a_6 = 5.7$.

$$\begin{aligned} y\left(x_1, x_2\right) = & \frac{1}{a_1}\Bigg\{-a_2 \exp\left(-a_3\sqrt{\frac{1}{a_4}\left(x_1^2 + x_2^2\right)}\right) \\ & - \exp\left(\frac{1}{a_4}\cos\left(a_5 x_1\right) + cos\left(a_5 x_2\right)\right) \\ & + a_2 + \exp\left(1\right) + a_6\Bigg\} \end{aligned} \tag{4.11}$$

The experimental domain for the Ackley function is bounded in this simulation by:

$$x_1 \in [-0.5; 1.5] \tag{4.12}$$

$$x_2 \in [-0.5; 1.5] \tag{4.13}$$

In this domain, the function has three local minima and one global minimum, whose coordinates are given in table 4.1. A representation of the function is given in figure 4.4.

4.3.2 Simulation Procedure

For deterministic computer experiments a sample size of $n = 10k$, where k is the number of factors, is a reasonable rule of thumb for an initial experiment [88].

Table 4.1: Minima of the implemented 2D Ackley function.

Coordinates	**Minimum type**
$\{0.9522; 0\}$	local
$\{0; 0.9522\}$	local
$\{0.9685; 0.9685\}$	local
$\{0; 0\}$	global

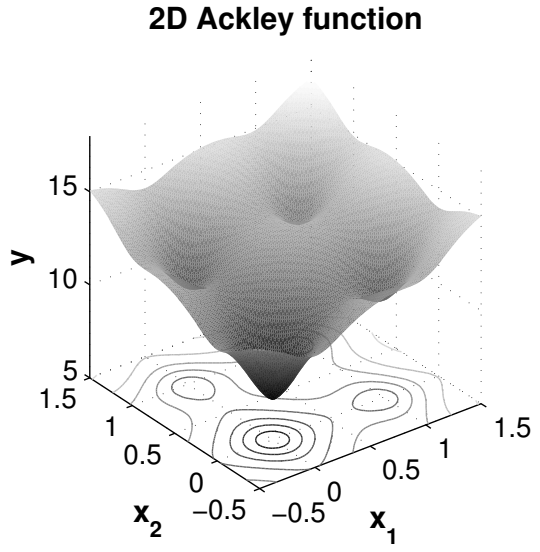


Figure 4.4: 2-factor Ackley Function in the experimental domain

Although in this study random noise is added to the response, we adhered to this rule of thumb and created LHC designs consisting of only 20 measurements. The reason being that the main goal is to pinpoint the region of the global optimum, rather than fitting the function in the whole space with great detail. In figure 4.5 the placement of the LHC design sample points of one of the generated designs is visualized by the black dots on the contour lines.

In contrast with the LHC design, which is computer-generated, the rotatable CCD is constructed according to set rules and will remain the same for all measurements. A rotatable CCD is constructed within the experimental domain of the Ackley function and a representation of the placement of the design points in the experimental domain is given in figure 4.6 with the actual coordinates of the design points presented in table 4.2. The number of measurements for the

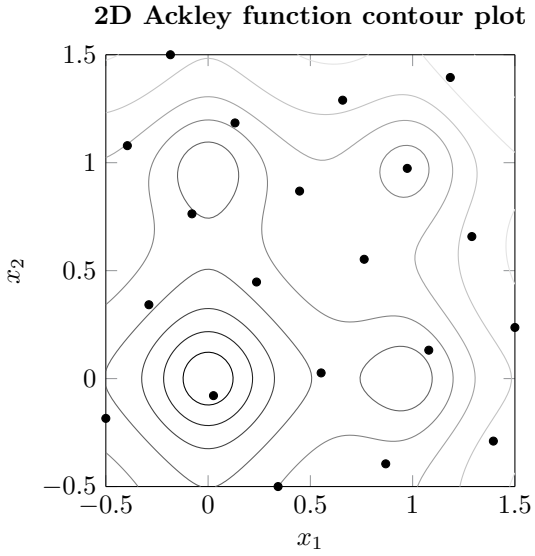


Figure 4.5: Contour plot of 2D Ackley function, the dots denote the placement of the LHC design points.

CCD was chosen to be equal to that of the LHC design, namely 20 measurements. The design points (save the centerpoints) are replicated once ($r = 2$) and four centerpoints ($n_c = 4$) are added, which leads to $2\left(2^2 + 2 \cdot 2\right) + 4 = 20$ measurements.

Table 4.2: Design points for the rotatable CCD, coded axial points at $\alpha = \sqrt{2}$ used for the Ackley function.

x_1	x_2	**# replicated**
-0.5	0.5	2
-0.2071	-0.2071	2
-0.2071	1.2071	2
0.5	-0.5	2
0.5	0.5	4
0.5	1.5	2
1.2071	-0.2071	2
1.2071	1.2071	2
1.5	0.5	2

In order to investigate the influence of the noise level on the accuracy of finding

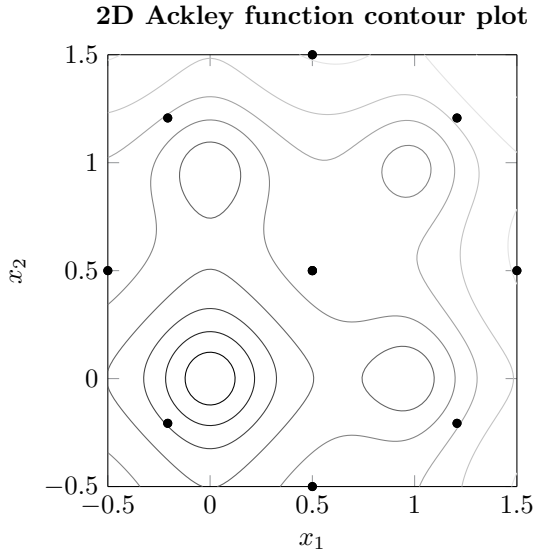


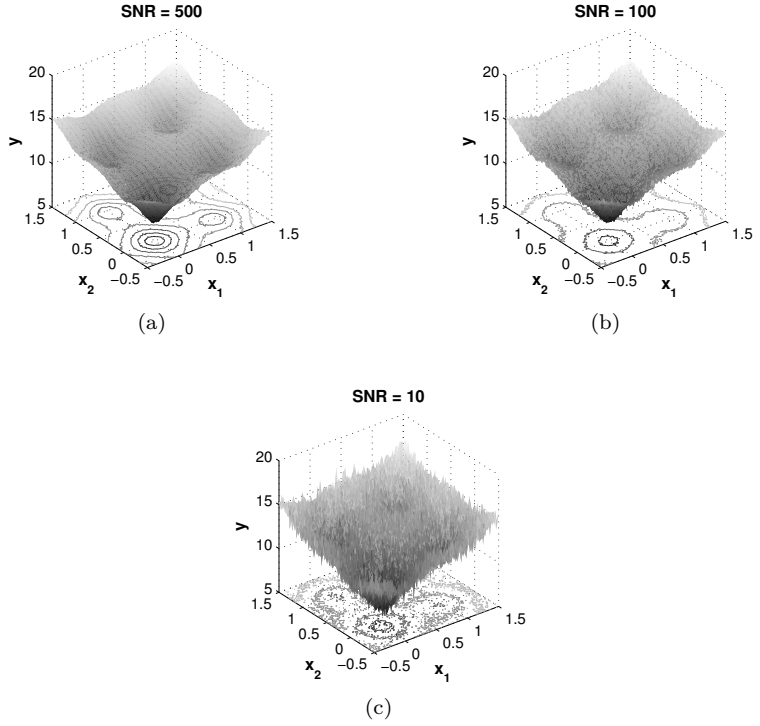
Figure 4.6: Contour plot of 2D Ackley function, the dots denote the placement of the CCD points.

the global optimum a relative measure for noise was used. This Signal-to-Noise Ratio (SNR), which is often used in engineering disciplines, is defined as the magnitude (variance) of the noise-free signal (sometimes referred to as the RMS value) divided by the magnitude (variance) of the noise as presented in equation 4.14.

$$\text{SNR} = \frac{\frac{1}{N-1}\sum_{i=1}^{N}\left(y_{i,\sigma=0} - \bar{y}_{\sigma=0}\right)^2}{\frac{1}{N-1}\sum_{i=1}^{N}\left(y_i - y_{i,\sigma=0}\right)^2} \tag{4.14}$$

where σ is the error (noise) standard deviation, N the total number of measurements, $\bar{y}_{\sigma=0}$ the average response when no noise is present, y_i the i-th response with noise and $y_{i,\sigma=0}$ the i-th noise-free response. The SNR was set at three discrete values being 500, 100 and 10. The error standard deviation σ is calculated using the SNR setpoint and a calculated magnitude (variance) of the noise-free signal and this σ is used to generate noise by multiplying it with the output of Matlab®'s *randn* function, which generates random numbers drawn from the standard normal distribution.

In order to have a clear understanding of the different signal-to-noise levels that were chosen in this study, the noise-corrupted responses for the Ackley function are plotted on figure 4.7. As can be seen from this figure, for SNR = 500 the noise has a minimal effect on the shape of the function and in the case of a low SNR, e.g. SNR = 10, the surface becomes distorted but the general characteristics are still maintained.



(a) (b) (c)

Figure 4.7: Plot of the noise-corrupted Ackley function for: (a) SNR = 500, (b) SNR = 100, (c) SNR = 10.

To study the robustness of the proposed approach in the presence of noise, 30 LHC designs were generated for each noise level and 30 GP models were built based on these designs, which results in a total of $3 \cdot 30 = 90$ designs and 90 models. The JMP® software is used to construct GP models, based on equation 4.8, and for each GP model the estimated optimum was calculated using the same software. A nugget parameter was estimated for all levels of the SNR. To compare with RSM, the CCD design is executed 30 times for every SNR, resulting in 30 measurements of the same design and 90 models.

To compare GP and OLS modelling on the LHC designs, an OLS regression model is constructed as well on every LHC design, leading to 90 OLS models.

In order to quantify the precision with which GP modelling and OLS regression find the global optimum, the coordinates of the fitted global minimum were compared to the underlying ground truth optimum which is located at $\{0, 0\}$. The Euclidean distances d_{ED} between the true and the estimated optima are calculated and the median of these distances for each SNR are reported as well as the interquartile range (IQR), which is a measure of the variation of d_{ED}. The median and IQR are chosen as measures of location and variability since they are more robust to outliers than the mean and standard deviation.

4.3.3 Results

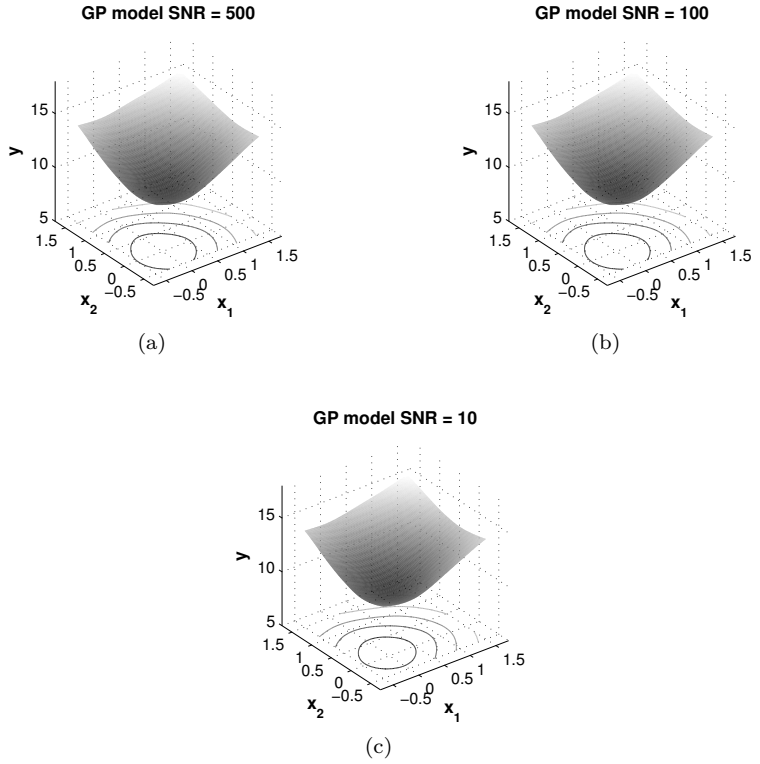
Gaussian Process Modelling

For every SNR, 30 LHC designs were generated, and on each of these a GP model was fit, totalling 90 GP models. Figure 4.8 presents one realization of a GP model for each of the noise-levels under consideration.

Given the limited number of 20 design points, the model is not able to perfectly capture the behavioural details of the real Ackley function. However, the proposed procedure is sufficient to pinpoint the region of the optimum which was the goal of this research. This same reasoning holds for all SNR levels. As expected, increasing the noise-level decreases the quality of the GP model. Yet in all cases the fitted model succeeds in pinpointing the global minimum adequately since the GP models always selected the region in which the global optimum is located and never a region of a local optima.

The Euclidean distances $d_{ED,GP}$ between the true and the estimated optima are calculated and the median of these distances per SNR are reported as well as the IQR, in table 4.3. As can be clearly seen from this table, the median $d_{ED,GP}$, and its variation, are very small. The higher the noise in the system, the larger one expects variation and the median distance to become as there is more uncertainty about the location of the optimum which is also evidenced in the table.

In figure 4.9 the estimated optima for the 30 models per noise-level are shown. As expected, they show to be randomly scattered around the true optimum, and lower noise scenarios lead to a more precise estimation of the optimum. From the point of view of determining the region of the global optimum the GP approach has succeeded even if the fitted models, of which examples were given in figure 4.8, clearly show that they are over-smoothed (compared to the



(a) (b) (c)

Figure 4.8: GP models of the Ackley function for different noise-levels: (a) SNR = 500, (b) SNR = 100, (c) SNR = 10.

Table 4.3: Results of Ackley simulation study for GP modelling with LHC designs, median of Euclidean distance true-estimated optimum, interquartile range of Euclidean distance.

SNR	**Median of $d_{ED,GP}$**	**IQR**
500	0.0373	0.0315
100	0.0478	0.0416
10	0.0965	0.0811

deterministic function in figure 4.4). This can be explained by the fact that the inclusion of the nugget parameter ensures that the estimated model does not interpolate exactly through the response values at the design points, as would be the case when using GP modelling on a deterministic simulation, but tries to fit a smooth function that best describes the data given the presence of noise.

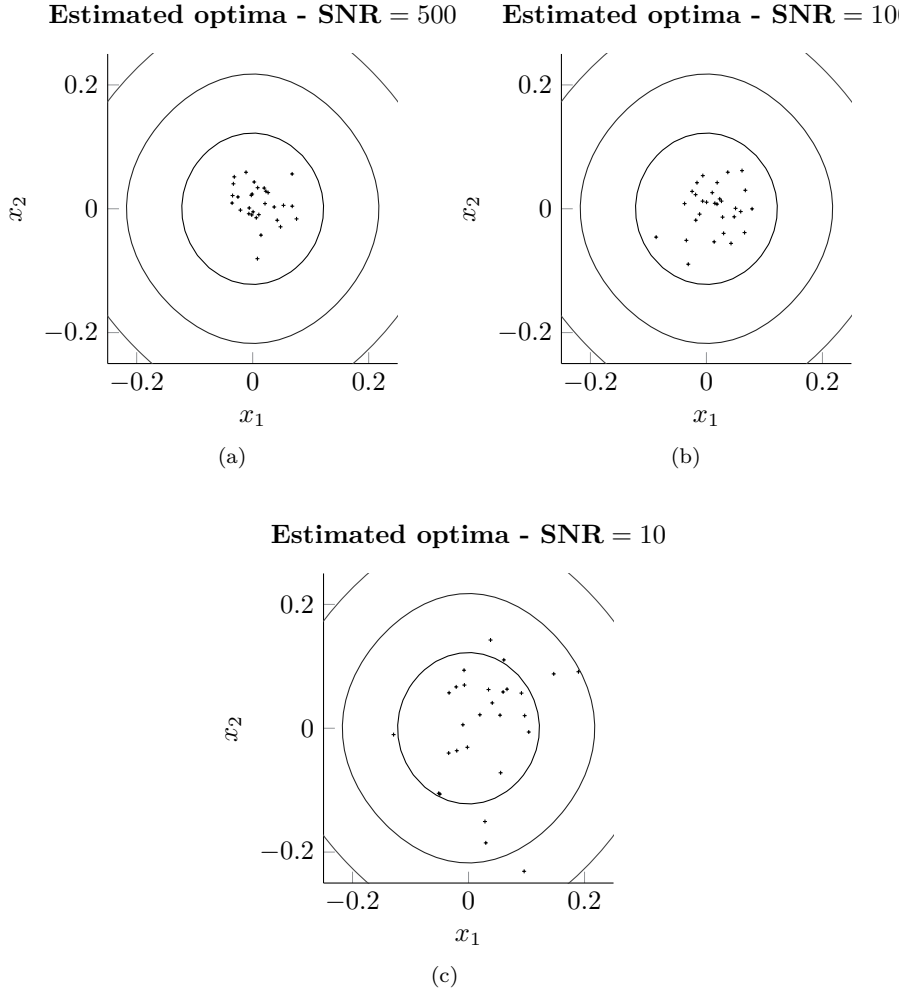


Figure 4.9: GP-estimated optimal factor settings plotted on deterministic Ackley function contours for: (a) SNR = 500, (b) SNR = 100, (c) SNR = 10.

Comparison With Response Surface Methodology

For RSM, second-order models were built according to equation 4.10 for 30 simulations and the three Signal-to-Noise Ratios. It is clear that the polynomial model will struggle in capturing the process dynamics due to the high non-linearity of the Ackley function and, given the location of the design points of the CCD (figure 4.6), it is expected that the resulting second-order polynomial will have difficulty in pin-pointing the optimal region.

Indeed, the estimated optimum of the constructed second-order models is always located outside of the experimental domain (outside of the bounds imposed on the Ackley function). Even more so, for SNR = 10, six of the 30 models do not have a global minimum. In a practical situation the optimal settings would be bounded to the experimental domain. In this aspect, the optimum of the second-order models was searched, constrained to the experimental domain. This optimum was always located $\{-0.5, -0.5\}$, which corresponds to a corner of the experimental domain.

In table 4.4 the results for this constrained optimization per SNR is shown in terms of the median of the Euclidean distance $d_{ED,RSM}$ between the true and the estimated optimum as well as the IQR. The distances are not extremely large, since one of the design points is located close to the global optimum, which will influence the tendency of the second-order model to decrease in this direction. However, the estimated RSM factor levels of the optimum are located further from the true optimum compared to the results from the GP modelling and more sequential improvement steps would be necessary to find the true optimal settings when starting from the optimal values determined by RSM.

Table 4.4: Results of Ackley simulation study for RSM, median of Euclidean distance true-estimated optimum, interquartile range of Euclidean distance.

SNR	**Median of $d_{ED,RSM}$**	**IQR**
500	0.7071	0
100	0.7071	0
10	0.7071	0

The IQR is zero even though the estimated optima can be quite distant from the true optimum due to the fact that the estimated optima are located at $\{-0.5; -0.5\}$, which is logical as all models are influenced by the design and will tend to decrease in this direction.

Ordinary Least Squares Regression On The Space-Filling Design

When applying RSM and examining the placement of the design points of the CCD (figure 4.6), it was already expected that the model would fail to capture the optimum. A valid question would be to test whether OLS regression would perform better if a design is used that spreads the design points more uniformly throughout the experimental domain, such as the space-filling LHC design used for GP modelling. For this purpose, the same second-order models as used for the CCD are built for the LHC designs. The Euclidean distance from the estimated to the true optimum $d_{ED,OLS}$ and the related IQR's are reported in table 4.5.

Table 4.5: Results of Ackley simulation study for OLS modelling with LHC designs, median of Euclidean distance true-estimated optimum, interquartile range of Euclidean distance.

SNR	**Median of $d_{ED,OLS}$**	**IQR**
500	0.0639	0.0598
100	0.0789	0.0663
10	0.1670	0.2215

To assess the difference between the OLS and GP model built on the same LHC design, the difference between the Euclidean distances to the optimum $d_{ED,OLS} - d_{ED,GP}$ is calculated for every LHC design and graphically represented in figure 4.10. It can be seen from this figure that the difference in OLS and GP modelling is not very large but that GP does offer an improvement over OLS since the median of the distances is positive. A Wilcoxon Signed Rank test [54] was executed to test whether the median of the differences was different from zero and for the three SNRs it was concluded that they are statistically significantly different from zero at the 0.01 significance level.

Furthermore, when looking at the IQR for both GP and OLS in table 4.3 and table 4.5 it is clear that GP modelling offers less variability in the resulting distance from the true optimum between the different LHC designs.

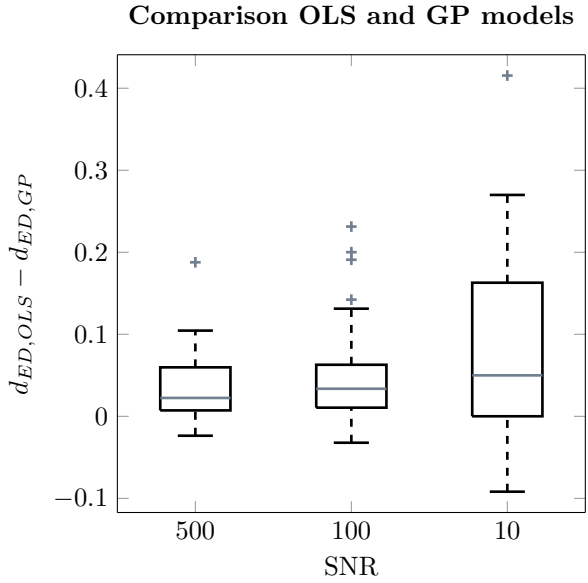


Figure 4.10: Difference in Euclidean distances between OLS and GP models for the Ackley simulation study.

4.4 Lab Scale Experiment

4.4.1 Bearing Test Setup

The bearing test setup is a lab scale setup developed by the department of Mechanical Engineering at the KU Leuven to analyse the behaviour of rolling element bearings subjected to highly varying loads. A detailed description of the setup can be found in [72], only those settings relevant for the executed experiment are detailed below.

To test the applicability of GP modelling on a practical setup, the following problem was considered: a bearing is placed within the bearing housing and rotated at a constant speed v_{set} of 1000 rpm. A shaker applies a sinusoidal dynamic axial load $F_{ax,dyn,set}$ with a frequency $f_{ax,dyn}$ of 391 Hz. Besides, static forces were applied in the axial ($F_{ax,st}$) and radial ($F_{rad,st}$) direction of the bearing housing using air springs. The shaker inducing the dynamical loads is voltage-controlled meaning that a constant voltage is applied to excite it. Depending on the static forces applied to the bearing housing, the peak force $F_{ax,dyn}$ generated by this dynamic excitation will change. To get an indication

of the size of such a peak force, it was measured when $F_{ax,st} = 400\,\text{N}$ and $F_{rad,st} = 800\,\text{N}$, which resulted in a peak force of $\sim$69 N.

Due to the dynamic axial load, bearing displacement will happen within the bearing housing. It was the interest of this experiment to find those input settings (static forces) that maximize the bearing displacements so to find the most problematic combination under the given conditions of dynamical load and rotation speed.

Figure 4.11a shows the bearing setup with all relevant hardware whilst figure 4.11b shows where the forces relevant for this experiment are applied. The bearing dynamics are measured using a capacitive sensor (Lion Precision Elite Series CPL190) and the spectral amplitude at the excitation frequency of 391 Hz, denoted by A_{391} , is used as an indicator for the bearing displacement. The spectral amplitude was estimated using a Fast Fourier Transform (FFT) of the data from the sensor. Table 4.6 presents the variables involved in the experiment and whether they were included in the experiment as a factor (independent variable), a response (dependent variable) or not taking into account (constants).

Table 4.6: Variables involved in the bearing setup and their role in the experiment.

Name	**Role**	**Setpoint or exp. range**
$F_{ax,st}$	factor	$[0; 800]$ N
$F_{rad,st}$	factor	$[0; 1600]$ N
$f_{ax,dyn,set}$	constant	391 Hz
v_{set}	constant	1000 rpm
A_{391}	response	-

4.4.2 Experimental Procedure

The two factors that were to be varied are the axial and radial static loads, $F_{ax,st}$ and $F_{rad,st}$ while the response under investigation is the peak amplitude A_{391} at the dynamic shaker excitation frequency which was set at 391 Hz.

Since the response was expected to be highly non-linear an LHC design of 25 sample points was constructed, which was the maximum amount of design points that could be measured within the allotted timespan for model-building. The experimental domain of interest was bounded, as presented in table 4.6.

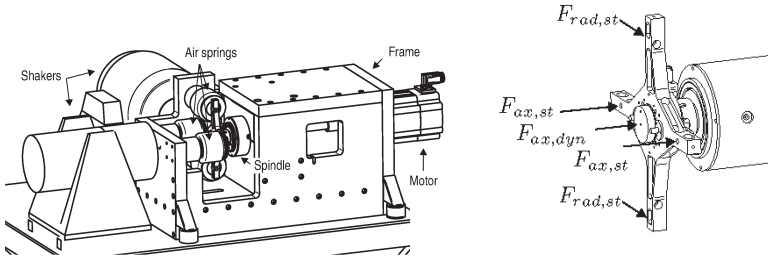


Figure 4.11: Schematic representation of the bearing setup: (a) a representation of the complete setup, (b) Detail of bearing housing and applied forces for this experiment, images courtesy of the department of mechanical engineering, KU Leuven, Belgium.

The generated settings from the LHC design were rounded towards the nearest integer as the setup can only accept integer force values for the set-points of the air springs. After new settings are applied to the setup the machine runs for 60 seconds to remove any transient behaviour that might result from the change of the forces. After this stabilization period the displacement sensor measures the axial displacement for 10 seconds at a sample frequency of 10 kHz. On this raw signal an FFT is performed and the performance indicator A_{391}, defined as the power (energy content) at 391 Hz is extracted.

Using the 25 responses measured at the LHC design points, a GP model with a nugget parameter is fitted using the JMP® software. As in this case the real optimum is not known, the assessment of the quality of this optimum cannot be investigated by looking at the median distance between the estimated and real optimum. Instead, this example is used to show the proposed strategy of applying EVOP as a next step. All estimated parameters from the EVOP program are also rounded towards the nearest integer for the reasons mentioned above.

4.4.3 Evolutionary Operation

In this real-life example, the GP modelling was followed by the sequential improvement step to pinpoint the true optimum. Evolutionary Operation was chosen as the sequential improvement method.

For each EVOP phase, a replicated 2^2 full factorial is used as the statistical

design, resulting in $2 \cdot 2^2$ measurements per phase. The design points are replicated since it is expected that this increased sample size is necessary to have a proper estimate of the noise. Based on the full factorial design, a main effects regression model is fitted as equation 4.15.

$$y = \beta_0 + \sum_{d=1}^{k} \beta_d x_d \tag{4.15}$$

where β_0 denotes the intercept and the β_d's denote the regression coefficients. A stepwise procedure, in which all model terms were included in the starting model, with a $p_{enter} = 0.05$ and $p_{remove} = 0.1$ was adopted in order to determine the most appropriate model on which the direction in which to move is calculated using the partial derivatives. This direction is then normalized such that the step length δ_{EVOP} is constant as determined by equation 4.16.

$$\delta_{EVOP} = \sqrt{dx_{axial}^2 \cdot f_{axial} + dx_{radial}^2 \cdot f_{radial}} \tag{4.16}$$

Where f_{axial} and f_{radial} are binary operators that are one when the corresponding main effect in the regression model is included and zero when the corresponding main effect is not included at the end of the stepwise regression. If the stepwise regression omits all terms from the main effects model the subsequent phase is stationary (i.e. $\delta_{EVOP} = 0$).

A new full factorial is placed at distance δ_{EVOP} from the centre of the previous design and this procedure is repeated sequentially, as explained in the previous chapters.

The starting point of the EVOP improvement is the centre of the first full factorial design and coincides with the estimated optimum from the GP model. The design points for the first phase are then placed at $\pm 0.5 dx_d$ from this point with dx_d the factorstep, being the distance between the cube points of the factorial design in the d-th dimension or factor. For the bearing setup the settings, the factorsteps were chosen to be $dx_{axial} = 16\,\mathrm{N}$ and $dx_{radial} = 8\,\mathrm{N}$. This implies that the cube points are placed $\pm 8\,\mathrm{N}$ for the radial force and $\pm 5\,\mathrm{N}$ for the axial force around the coordinates for the reference condition. The reference condition itself is not measured and no centerpoints are used in the factorial designs. The practical limitation in this real-life example was the execution of ten phases (80 measurements) at most.

4.4.4 Results

Gaussian Process Modelling

A GP model is built from the data of the 25-point LHC design to estimate which combination of static forces results in the largest bearing displacement. In figure 4.12 the GP model is plotted which shows that a large radial force and a medium axial force give the largest bearing displacement. The estimated settings for maximal bearing displacement were $F_{rad,st} = 1566\,\mathrm{N}$ and $F_{ax,st} = 320.95\,\mathrm{N}$, and coincide with the experience of the mechanical engineers. This last value was rounded to 321 N since the setup can only accept integers. These settings were used as the starting point for a sequential improvement method to determine if they are the actual maximum settings of the setup.

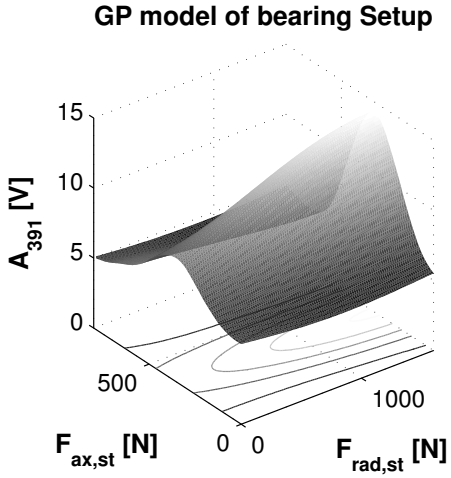


Figure 4.12: GP model bearing setup.

Online Improvement Of Optimal Factor Levels

The estimated settings for maximal bearing displacement are used as the starting point for the EVOP improvement step. Table 4.7 details the coordinates of the centerpoint of each phase, together with the number of phases executed at this location and the maximum measured value of the performance indicator A_{391} at the design points. The full improvement is plotted in figure 4.13 on the contour lines of the estimated GP model. Executing the improvement increased the performance indicator A_{391} by 7.5%.

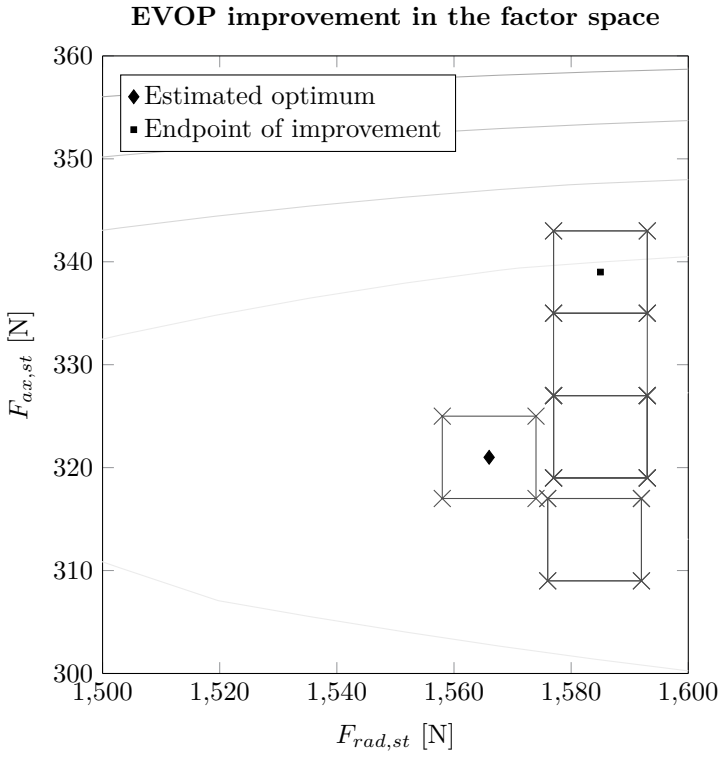


Figure 4.13: EVOP improvement from estimated GP optimum.

Table 4.7: Centerpoints of consecutive EVOP phases for the bearing setup improvement.

$F_{rad,st}$	$F_{ax,st}$	# phases	max A_{391}
1566	321	1	156.961
1584	313	2	158.734
1585	323	4	164.584
1585	331	1	165.845
1585	339	2	168.739

During EVOP, every decision to move resulted in an increase in the performance indicator. However, in between the moves, there are some stationary phases which imply that the signal (underlying slope of the response) was too weak when compared to the noise present so that the power to detect the direction of improvement was low. In such cases, it is advisable to increase the amount of replication (the sample size on which the model is based). The relative high amount of measurement noise can be attributed to the control of the air springs that are used in the setup to generate the static forces. A discussion with the engineers revealed that those springs are hard to adjust so that the repeatability in generating those forces is quite low.

Looking at the improvement path it can be seen that the endpoint of the improvement is quite close to the estimated optimum. The radial force $F_{rad,st}$ was increased by 19 N and the axial force $F_{ax,st}$ with 18 N compared to the estimated GP optimal values, which is a shift of $\sim 1.2\%$ for the former and $\sim 2.3\%$ for the latter, compared to the full experimental range of the factors.

4.5 Conclusions

Space-filling designs combined with Gaussian Process modelling to determine the region of the global optimum as a starting point for sequential improvement methods such as Evolutionary Operation was proposed in cases where the experimenter has no prior information about the location of the optimum. By applying the proposed approach, it was shown that even in noisy situations the region of the optimum was found with a limited number of measurements.

This method allows for fast exploration of the entire experimental domain after which production can be started in a feasible candidate point. By implementing an OSPI method, the response can then be sequentially improved during the production process. It will prove most effective in processes in which the underlying process model is (expected) to be highly non-linear as the broad span of models covered by Gaussian Process modelling can deal with such non-linearities.

Part II

Simulation Studies

5 A Comparison Of Evolutionary Operation And Simplex

5.1 Introduction

The era when Online Sequential Process Improvement (OSPI) was introduced was characterized by the limited availability of sensor technologies (to "sense" the process status) and virtually non-existent computer power (to process the information). As a result the first proposed OSPI method, Traditional Evolutionary Operation (TEVOP), is based on very simple underlying models and simplified calculations so that the process owner can easily compute it by hand. However, due to this manual procedure TEVOP was only applied on a low frequency basis, e.g. the improvement was run only once for each production lot to compensate for inter-lot variability.

The introduction of the OSPI way of thinking did also inspire other researchers to search for alternatives that adhered to the framework of OSPI such as Basic Simplex. Although the Basic Simplex methodology has been advocated as a technique for process improvement, the literature that deals with practical examples is rather scarce, as was shown in chapter 2. In general, Simplex did only modestly impact the process industry, but had a huge impact on the optimization of numerical functions. This is mainly due to the work of Nelder & Mead [109] who adapted the basic Simplex scheme to allow for variable perturbation sizes. Although efficient in numerical optimization this Variable Simplex procedure is not suited for tracking the optimum of real-life processes

The results section was published in:
K. Rutten et al. "A Comparison of Evolutionary Operation And Simplex For Process Improvement". In: *Chemometrics and Intelligent Laboratory Systems* 139 (Dec. 2014), pp. 109–120. DOI: 10.1016/j.chemolab.2014.09.011

due to several reasons. As mentioned earlier in this dissertation, perturbations should be small enough so that the risk of producing nonconforming products is minimized. On the other hand, due to the noise that is inherent to practical experimentation, the perturbation size cannot be too small since the Signal-to-Noise Ratio (SNR) will be insufficient in such a case to pinpoint the direction of the optimum. Because of those drawbacks, reports on the application of Variable Simplex on real-life industrial processes are very limited. Variable Simplex is most often used on numerical optimization studies, an example of which is the optimization of injection moulding on a finite element model [49]. Usually, this method is used in research and development where it is relatively safe to make large changes in the levels of the factors [111]. Therefore it is not a valid technique for use in full scale processes.

Since the introduction of OSPI, the way processes are monitored changed completely. Modern processes are sampled at high frequency using multiple sensor technologies; making the original (manual) EVOP and Simplex schemes, which were typically applied in case of only two or three factors, unfeasible. This complicated experimental setting is largely compensated by the increased computation power making both methods interesting for modern processes.

Yet, to the author's best knowledge, there is no systematic publication discussing EVOP and Simplex in the modern setting, i.e. in higher ($k > 3$) dimensions. The only article found where examples of both methods appear is written by Lowe [90] where advantages and disadvantages for both EVOP and Simplex are given, elucidated with different examples. The most distinct advantage listed for EVOP is that it can be used for either quantitative or qualitative factors. And the most marked disadvantage of EVOP that is listed is the fact that the inclusion of many factors would incur too many measurements and make the experimentation prohibitive. For Simplex a clear advantage is the simplicity of the calculations and the fact the only a minimal number of experiments need to be performed to move through the experimental domain. A distinct disadvantage of the method is that wrong moves can be made more easily in the presence of noise. However, the paper does not compare the two methods with each other on the same case. Instead, the conclusions drawn are based on personal experience only. According to Lowe, process supervisors do not see the additional benefits of the methods since they are not familiar with the underlying concepts and want to see a significant improvement as quickly as possible, while these methods aim to improve a process by small perturbations only to ensure no non-conforming products are manufactured.

Although two main applications of the methodology exist, being: (1) improving a stationary process by finding the direction of the optimum and (2) tracking the optimum of a non-stationary process (i.e. a process that drifts in time), this chapter is restricted to assessing and to comparing the performance of

EVOP and Simplex for situations with different dimensions (up to 8 factors), perturbations sizes and noise levels on stationary processes only. An in-depth insight in the methodologies is presented and recommendations about when to use which method are given. The comparison is based on a simulation study to gain deeper insight in the strengths and weaknesses of both methods under varying conditions that cannot be replicated on real processes since the number of factors and the amount of noise in the process are known to be important and, in general, cannot be changed independently and in a controlled way on a real process.

For this comparison the automated, EVOP procedure that was developed in chapter 3 will be used.

5.2 Materials & Methods

EVOP and Simplex are compared based on a simulation study with multiple scenarios. Main settings studied are (1) the perturbation size in every dimension, which will be labelled factorstep dx_d; (2) the signal-to-noise ratio (SNR) and (3) the dimensionality k (number of factors) of the problem. The quality of the improvement is quantified using several criteria, such as the number of measurements that are needed to attain a well-defined optimal region (see further), the interquartile range (IQR) as a measure for the repeatability (see further) and the number of cases the optimal region is reached ("success rate").

5.2.1 Underlying Model

As mentioned higher, the tested methods are developed especially to seek/track the optimum when the process is already operating near this optimum as determined during prior (offline) experimentation. As a result, they are not designed to optimize multimodal processes with multiple local optima when no prior information about the region of the global optimum is available. Based on this philosophy the chosen underlying model has only one global optimum in the considered design space, so that issues of being trapped in local optima are excluded. A quadratic function, defined by equation 5.1, was chosen to describe the region of the optimum.

$$y_i = 200 - 128 \sum_{d=1}^{k} x_d^2 + \varepsilon_i \tag{5.1}$$

In this equation y_i is the i-th response variable, x_d $(d = 1, \ldots, k)$ the k factors and a random error term ε. The optimal response for any number of inputs k is 200 and is located at $\{0, 0, \ldots, 0\}$. The number of factors used in the simulations ranges from 2 to 8. For each factor the experimental domain was bounded by $[-1; 1]$. It was opted not to include interactions in the underlying model since a higher dimensionality enables a higher number of possible interactions (two-factor, three-factor, ..., k-factor). Evidently, it is not representative to take all possible interactions into account, raising the question which number of them to include, and whether only two-factor interactions should be considered. Besides, also the size and direction of all interactions would need to be selected as they will significantly impact the model behaviour. Such a study would completely lack focus on dimensionality and is therefore avoided here. Furthermore the effect hierarchy principle [159] states that interactions (especially higher order interactions) are generally of less importance than the main effects.

The quadratic coefficients are set at the arbitrary value of 128, but its influence is standardized using a relative amount of noise present in the data through the use of the SNR comparing the signal ($\sim$ coefficients) to the noise ($\sim \varepsilon$), as is explained in the following section.

5.2.2 Simulation Settings

Three settings are controlled in the simulation study: the SNR to control the noise level relative to the amount of information present, the size of the perturbations which are induced by setting the factorstep dx_d and the dimensionality k of the problem. The definition of these three settings is given below, as well as other important performance characteristics.

Signal-to-Noise Ratio: In order to investigate the influence of the noise level, a relative measure for noise is used. This signal-to-noise ratio, which is often used in engineering disciplines, is defined as the magnitude (variance) of the noise-free signal (sometimes referred to as the RMS value) divided by the power (variance) of the noise, and is presented in equation 5.2 (which is the same as equation 4.14 and is repeated here for ease of reference).

$$SNR = \frac{\frac{1}{N-1} \sum_{i=1}^{N} \left(y_{i,\sigma=0} - \bar{y}_{\sigma=0}\right)^2}{\frac{1}{N-1} \sum_{i=1}^{N} \left(y_i - y_{i,\sigma=0}\right)^2} \tag{5.2}$$

The SNR was set at 8 discrete values being ∞ (no noise), 1000, 500, 250, 100, 50, 25 and 10.

Factorstep: The factorstep dx_d represents the initial perturbation in every dimension for both EVOP and Simplex and will be schematically presented in section 5.2.4 where the Simplex and EVOP starting point is discussed. Since all dimensions were scaled to the same range (see higher), the factorstep in every direction is taken equal so that the subscript d is omitted. The factorstep dx was fixed at three discrete values, being 1%, 5% and 10% of the experimental range for each variable.

Dimensionality: In order to investigate the properties of EVOP and Simplex as a function of the dimensionality of the problem, k was varied between 2 and 8.

Number of repetitions: In order to have a clear view on the performance of both methods in the presence of noise at each combination of SNR, factorstep and dimensionality the simulation was performed 30 times for each method to be able to calculate the median value and the IQR which are used as performance indicators for the methods (see later) within a reasonable time. This leads to a total of number of improvements of:

$$2\,(\text{methods}) \times 8\,(\text{noise-levels}) \times 3\,(\text{factorsteps}) \times 7\,(\text{dimensions})$$
$$\times\, 30\,(\text{repetitions}) = 10,080$$

Starting point: The coordinates of the starting point $\mathbf{x}_{start}$ for all simulations were kept fixed according to equation 5.3.

$$\{x_{start,d}\}_{d=1}^{k} = \frac{l_{start}}{\sqrt{k}} \tag{5.3}$$

with k as defined higher and l_{start} the radius of the chosen contour line. In this way the starting point is located at the same contour line regardless of the dimensionality. The radius l_{start} was chosen to be 0.95 in all simulations thus fixing the noise-free initial response at 84.48. The exact implications of the starting point for EVOP and Simplex are explained further for each method separately.

Stopping criterion: Several stopping criteria are used in literature and have a large impact on the number of measurements and success rates. Since there is no equivalent stopping criterion for EVOP and Simplex the introduction of such different criteria would have a large impact on the comparison. Since the

interest is in comparing the two methodologies, it was opted not to use any (subjective) stopping criterion in this comparison study, but only to define the maximum number of measurements to complete. This number is chosen to be $51,200$, which is large enough to ensure that a simulation has a reasonable chance of reaching the optimum in all scenarios. This number is based on the full factorial design for $k = 8$, which has the largest sample size ($n_T = 256$). In that case, 200 phases are allowed to reach the optimum, which is approximately 4 times the required number if the optimal path was followed.

Success rate: A "success" was defined as an improvement that resulted in attaining a noise-free response value which is equal to or larger than 95% of the theoretical optimum (in this case 190). When this level was not reached using the maximum number of measurements it was labelled "failure".

Median number of measurements: The median of the 30 repetitions for each simulation setting is used as a measure for the number of measurements needed to reach the region of the optimum.

Interquartile Range (IQR): The IQR of the 30 repetitions for each simulation setting is used as a measure for the stability of the method to reach the region of the optimum, as it gives a clear indication of the amount of randomness involved in the followed path. The smaller the IQR, the less randomness is involved.

5.2.3 Sequential Improvement Methods Used

Both Simplex and EVOP were implemented in this study and will be detailed below. For Simplex, a capital "S" will be used to denote the method, whereas lower case "s" will be used to denote the geometric figure, as before.

Simplex

Basic Simplex has been implemented as described in chapter 2. The initial $k+1$ vertices were chosen according to the tilted method [143, 155]. A previous study [143] has indicated that orientation of the initial simplex (i.e. same construction procedure and rotating the simplex around its point of gravity) has an effect on the performance, but Spendley *et al.* illustrated that this effect is negligible when noise is present and when $k > 2$. The tilted method was chosen since a study by Öberg [110] has shown that this method performs better than the other, classically used, corner method to select the vertex points. This was also observed here (results not shown).

For the tilted algorithm the initial vertex coordinate is given as $\mathbf{x}_{co} = \{x_{co,1}, x_{co,2}, \ldots, x_{co,k}\}$, which is the only vertex that needs to be chosen by the experimenter, all other vertices are calculated by the rules set forth for the tilted simplex in chapter 2.

In this simulation study the boundaries are simple single-factor boundaries. This gives a feasible region in the factor space which lies within a k-dimensional hypercube. To ensure that no experiments outside of the boundaries are performed, the Simplex procedure will allocate an infinitely bad response to points outside of the domain, forcing the procedure to reflect back inside the hypercube. These points outside of the cube are phantom measurements, meaning they do not require experimentation effort; they are set by the Simplex procedure itself.

Evolutionary Operation

Evolutionary Operation was implemented as described in chapter 3 and for each EVOP phase, a 2^k full factorial design is used as the statistical design, as is common in the classical interpretation of EVOP.

Based on the full factorial design, a main effects regression model is fitted. The fact that only a main effects model (and not the possible interactions) are considered, results from the fact that the direction of improvement is calculated from the centre of the design, which was implemented in coded units, as explained in chapter 3.

A stepwise procedure, in which all model terms were included in the starting model, with a $p_{enter} = 0.05$ and $p_{remove} = 0.1$ was adopted in order to determine the most appropriate model on which the steepest ascent deltas ($\boldsymbol{\beta}$) are calculated using the partial derivatives. This vector of steepest ascent deltas is then normalized to $\boldsymbol{\delta}_{\mathbf{EVOP}}$ such that the step size is always constant.

To ensure that no experiments fall outside the considered hypercube the procedure set forth in chapter 3 was implemented: If a calculated steepest ascent delta for a dimension would cause a new EVOP phase to fall outside the experimental region, the method is forced to traverse alongside the border of the experimental domain by setting the steepest ascent delta to zero for that dimension.

5.2.4 Definition Of The Starting Point For EVOP And Simplex

For EVOP, the starting point $\mathbf{x_{start}}$ will refer to the center of the first full factorial design, so that the first design points are placed at $\pm 0.5 \cdot dx$ from this point (see figure 5.1). For the Simplex approach, the initial corner point $\mathbf{x_{co}}$ was defined as $\{x_{co,d}\}_{d=1}^{k} = \{x_{start,d} - 0.5 \cdot dx\}_{d=1}^{k}$ so that the initial simplex vertices lie within the same region as the EVOP points, as illustrated in figure 5.1 for two factors.

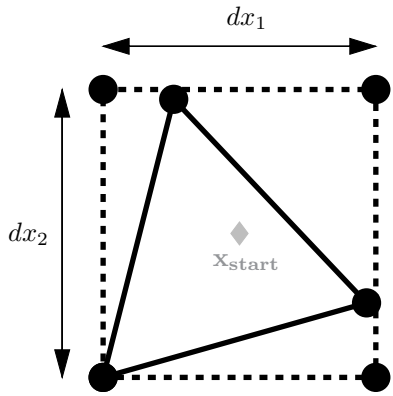


Figure 5.1: Initial EVOP (dashed line) and Simplex (full line) design.

5.2.5 Definition Of The Step Size

For EVOP, the vector including the step sizes in each dimension $\boldsymbol{\delta}_{\mathbf{EVOP}} = (\delta_{EVOP,1}, \ldots, \delta_{EVOP,k})$ consists of the normalized steepest ascent deltas in every dimension. This vector $\boldsymbol{\delta}_{\mathbf{EVOP}}$ determines the distance between the corresponding points of two consecutive EVOP phases. The step size was fixed by equation 5.4.

$$\delta_{EVOP} = ||\boldsymbol{\delta}_{\mathbf{EVOP}}|| = dx \cdot \sqrt{f_a} \tag{5.4}$$

With f_a the number of active effects resulting from running the stepwise regression model.

In analogy with EVOP, also for Simplex, the step size $\delta_{Simplex}$ is calculated based on the factorstep dx and is defined as the Euclidean distance between the worst point $\mathbf{w}$ (that will be omitted in the subsequent Simplex step) and

its reflection **r** (the point to measure in the subsequent step, see figure 2.8a for the 2D example) as presented in equation 5.5.

$$\delta_{Simplex} = ||\mathbf{r} - \mathbf{w}|| \tag{5.5}$$

With **r** and **w** the coordinate vectors of the reflection point and the point labelled as worst respectively. Since the initial simplex is a regular polygon every reflection also results in a regular polygon with the same hypervolume and geometry. As a result the step size is the same, irrespectively of the rejected vertex **w**, (but still a function of the dimensionality) as was already illustrated in chapter 2 (figure 2.8b).

5.3 Results & Discussion

5.3.1 Visualisation Of The Effect Of The Signal-To-Noise Ratio

In order to have a clear understanding of the different signal-to-noise ratios that were chosen in this study, the noise-corrupted response for several Signal-to-Noise Ratios were plotted for the two-factor model (figure 5.2). The noise effect becomes clearly visible when the SNR value drops below 250, whereas for an SNR of 1000 the noise has only a marginal effect.

5.3.2 Comparing Step Sizes

As mentioned in 5.2, the EVOP step size δ_{EVOP} is determined by the active factors included in the reduced linear model. As a consequence the maximum step size is obtained when all factors are active and is defined solely by the dimensionality of the problem and the size of the factorstep. On the contrary, the Simplex step size $\delta_{Simplex}$ is completely defined by the initial simplex that is used. Since it is a regular polygon, the step size remains constant during an improvement and is also solely dependent on the dimensionality of the problem, as the constants p and q of the tilted algorithm (see 2) only depend on the dimensionality for a given factorstep.

Given the settings that are applied in this study the maximum EVOP step size is smaller than the Simplex step size for two dimensions, whilst for higher dimensions the reverse holds. As the dimensionality of the problem increases, the EVOP step size will increase while the Simplex step size will decrease

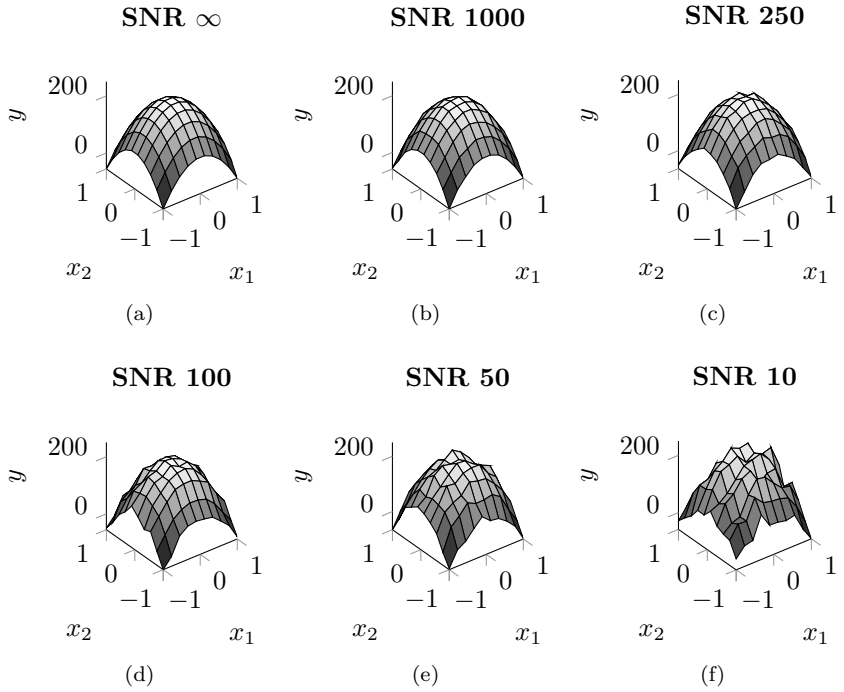


Figure 5.2: Plot of several signal-to-noise ratios for two factors: (a) SNR $= \infty$, (b) SNR $= 1000$, (c) SNR $= 250$, (d) SNR $= 100$, (e) SNR $= 50$, (f) SNR $= 10$.

slightly. Figure 5.3 shows this behaviour as a function of k for one noise-free improvement, for which all factors are active.

One can argue that the larger EVOP step size for high dimensions gives the method an unfair advantage to Simplex, but the cost of one additional EVOP phase is much higher than one Simplex phase. Indeed, one Simplex phase augments the number of measurements by one, whereas 2^k measurements are added during each EVOP phase, resulting in a much higher experimental cost. In addition, while the Simplex procedure is always forced to move to a new point based on the heuristic rules, this is not the case for the EVOP procedure. If at least one term is found to be significant by stepwise regression, it results in moving the full factorial; if no terms are found to be significant (due to noise or to approaching the optimum at which the derivatives go to zero), the step size equals zero ("stationary phase"). In the noise-free case depicted in figure 5.3, EVOP always finds significant terms in the model if it is not yet at the optimum, thus no stationary phases are performed.

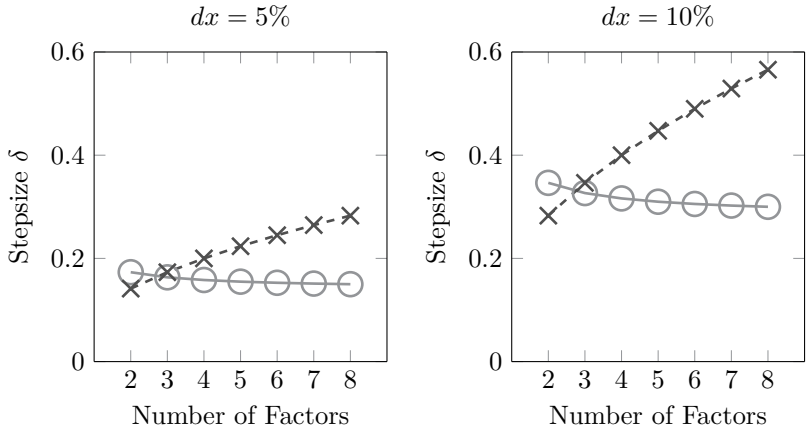


Figure 5.3: Step sizes for EVOP (X) and Simplex (O) with SNR = ∞: (a) $dx = 5\%$ and (b) $dx = 10\%$.

5.3.3 Comparison Of The Number Of Measurements

Table 5.1 and 5.2 provide an overview of the simulation results with respect to the number of measurements that are needed to reach the region of the optimum. Relevant information is shown for each combination of the factorstep dx (expressed as a percentage of the experimental range), for each SNR and for each dimensionality k. All values in these tables have been rounded towards plus infinity for readability. As stated before; the median is presented as a measure of the location of the number of measurements necessary to reach the optimal region, while the interquartile range is a measure for the stability of the method. The noise-free case will be discussed first as this is a special scenario representing deterministic simulations. Afterwards, the effect of each of the changed settings will be discussed in detail.

Results For Deterministic Simulations

A situation where no noise is present (SNR = ∞) is never found in practical (physical) systems, but is often present when dealing with computer simulations or analytical functions. In such cases, Simplex always outperforms EVOP in terms of the number of measurements that are needed to reach the region of the optimum (see table 5.1 and 5.2). Due to the higher number of measurements

Table 5.1: EVOP simulation data [median of number of measurements to reach optimum | interquartile range (number of times optimum was reached)].

		Number of Factors						
dx	SNR	2	3	4	5	6	7	8
1 %	∞	100 | 0 (30)	160 | 0 (30)	288 | 0 (30)	512 | 0 (30)	960 | 0 (30)	1,792 | 0 (30)	3,328 | 0 (30)
	1,000	642 | 168 (30)	936 | 160 (30)	1,344 | 304 (30)	2,048 | 256 (30)	2,880 | 512 (30)	4,352 | 640 (30)	6,144 | 768 (30)
	500	870 | 256 (30)	1,272 | 304 (30)	1,984 | 432 (30)	3,072 | 672 (30)	4,640 | 576 (30)	6,528 | 768 (30)	9,856 | 1,280 (30)
	250	1,140 | 312 (30)	1,716 | 352 (30)	2,712 | 1,008 (30)	4,672 | 640 (30)	6,848 | 1,216 (30)	10,368 | 1,280 (30)	13,824 | 2,560 (30)
	100	1,798 | 556 (30)	3,008 | 1,064 (30)	4,696 | 1,440 (30)	7,056 | 2,080 (30)	11,616 | 2,304 (30)	17,792 | 3,328 (30)	25,216 | 5,120 (30)
	50	2,464 | 1,012 (30)	3,888 | 1,544 (30)	5,976 | 1,744 (30)	11,104 | 2,528 (30)	16,736 | 2,880 (30)	24,640 | 6,784 (30)	35,584 | 8,064 (29)
	25	3,446 | 1,372 (30)	5,380 | 2,400 (30)	9,872 | 5,232 (30)	16,992 | 6,240 (30)	22,624 | 8,064 (30)	39,360 | 11,008 (26)	46,080 | 6,080 (7)
	10	5,416 | 2,608 (30)	9,176 | 6,544 (30)	15,520 | 8,352 (30)	23,936 | 9,888 (29)	37,056 | 9,920 (20)	40,704 | 11,104 (5)	/
5 %	∞	24 | 0 (30)	40 | 0 (30)	64 | 0 (30)	128 | 0 (30)	256 | 0 (30)	512 | 0 (30)	768 | 0 (30)
	1,000	42 | 12 (30)	40 | 0 (30)	64 | 0 (30)	128 | 0 (30)	256 | 0 (30)	512 | 0 (30)	768 | 0 (30)
	500	52 | 24 (30)	48 | 16 (30)	64 | 16 (30)	128 | 0 (30)	256 | 0 (30)	512 | 0 (30)	768 | 0 (30)
	250	64 | 32 (30)	56 | 16 (30)	80 | 16 (30)	160 | 32 (30)	256 | 0 (30)	512 | 0 (30)	768 | 0 (30)
	100	98 | 40 (30)	84 | 32 (30)	144 | 48 (30)	240 | 96 (30)	320 | 64 (30)	512 | 128 (30)	1,024 | 256 (30)
	50	102 | 48 (30)	152 | 64 (30)	224 | 64 (30)	320 | 96 (30)	512 | 128 (30)	768 | 128 (30)	1,024 | 256 (30)
	25	120 | 76 (30)	196 | 120 (30)	312 | 144 (30)	496 | 224 (30)	640 | 192 (30)	1,024 | 256 (30)	1,792 | 512 (30)
	10	172 | 68 (30)	376 | 264 (30)	536 | 352 (30)	816 | 416 (30)	1,376 | 384 (30)	1,792 | 512 (30)	2,560 | 768 (30)
10 %	∞	12 | 0 (30)	24 | 0 (30)	48 | 0 (30)	64 | 0 (30)	128 | 0 (30)	256 | 0 (30)	512 | 0 (30)
	1,000	12 | 0 (30)	24 | 0 (30)	48 | 0 (30)	96 | 0 (30)	128 | 0 (30)	256 | 0 (30)	512 | 0 (30)
	500	12 | 4 (30)	24 | 0 (30)	48 | 0 (30)	96 | 0 (30)	128 | 0 (30)	256 | 0 (30)	512 | 0 (30)
	250	16 | 12 (30)	24 | 0 (30)	48 | 0 (30)	96 | 0 (30)	128 | 0 (30)	256 | 0 (30)	512 | 0 (30)
	100	20 | 12 (30)	24 | 8 (30)	48 | 0 (30)	96 | 0 (30)	128 | 0 (30)	256 | 0 (30)	512 | 0 (30)
	50	28 | 12 (30)	32 | 16 (30)	48 | 0 (30)	96 | 0 (30)	128 | 0 (30)	256 | 0 (30)	512 | 0 (30)
	25	36 | 12 (30)	44 | 16 (30)	64 | 32 (30)	96 | 32 (30)	128 | 64 (30)	256 | 0 (30)	512 | 0 (30)
	10	60 | 44 (30)	76 | 40 (30)	128 | 64 (30)	192 | 96 (30)	256 | 128 (30)	384 | 256 (30)	512 | 256 (30)

Table 5.2: Simplex simulation data [median of number of measurements to reach optimum | interquartile range (number of times optimum was reached)].

dx	SNR	Number of Factors						
		2	3	4	5	6	7	8
1 %	∞	79 | 0 (30)	114 | 0 (30)	157 | 0 (30)	206 | 0 (30)	257 | 0 (30)	313 | 0 (30)	373 | 0 (30)
	1,000	449 | 1,248 (30)	34,064 | 26,738 (12)	/	/	/	/	/
	500	2,945 | 2,489 (23)	/	/	/	/	/	/
	250	12,439 | 28,228 (14)	/	/	/	/	/	/
	100	/	/	/	/	/	/	/
	50	/	/	/	/	/	/	/
	25	/	/	/	/	/	/	/
	10	/	/	/	/	/	/	/
5 %	∞	17 | 0 (30)	26 | 0 (30)	34 | 0 (30)	45 | 0 (30)	55 | 0 (30)	64 | 0 (30)	77 | 0 (30)
	1,000	17 | 0 (30)	28 | 2 (30)	43 | 7 (30)	83 | 30 (30)	143 | 44 (30)	279 | 241 (30)	948 | 1,045 (30)
	500	17 | 0 (30)	31 | 7 (30)	70 | 20 (30)	181 | 208 (30)	472 | 513 (30)	2,487 | 5,761 (29)	8,221 | 11,406 (27)
	250	17 | 3 (30)	44 | 25 (30)	142 | 135 (30)	729 | 1,400 (30)	2,463 | 5,082 (29)	14,693 | 16,168 (23)	27,613 | 16,807 (5)
	100	22 | 10 (30)	116 | 188 (30)	833 | 2,003 (28)	5,747 | 14,742 (22)	35,457 | 24,819 (9)	/	/
	50	35 | 40 (30)	315 | 956 (30)	6,157 | 10,576 (25)	27,663 | 13,047 (3)	/	/	/
	25	78 | 283 (30)	2,855 | 9,422 (27)	19,553 | 26,027 (8)	29,902 | 0 (1)	/	/	/
	10	195 | 1,333 (27)	19,416 | 15,026 (12)	/	/	/	/	/
10 %	∞	11 | 0 (30)	15 | 0 (30)	19 | 0 (30)	23 | 0 (30)	30 | 0 (30)	35 | 0 (30)	40 | 0 (30)
	1,000	11 | 0 (30)	15 | 1 (30)	20 | 2 (30)	26 | 2 (30)	31 | 4 (30)	42 | 7 (30)	54 | 15 (30)
	500	11 | 0 (30)	16 | 0 (30)	21 | 2 (30)	28 | 6 (30)	37 | 14 (30)	53 | 17 (30)	85 | 54 (30)
	250	11 | 0 (30)	16 | 2 (30)	24 | 5 (30)	34 | 8 (30)	52 | 32 (30)	110 | 77 (30)	196 | 202 (30)
	100	11 | 0 (30)	19 | 3 (30)	33 | 35 (30)	58 | 87 (30)	197 | 231 (30)	472 | 775 (30)	3,379 | 3,777 (29)
	50	11 | 2 (30)	22 | 7 (30)	74 | 96 (30)	241 | 458 (30)	963 | 1,257 (30)	3,404 | 9,839 (28)	15,238 | 20,601 (23)
	25	15 | 9 (30)	45 | 40 (30)	200 | 422 (30)	834 | 2,950 (30)	3,263 | 9,438 (28)	5,845 | 17,713 (11)	16,180 | 21,851 (5)
	10	24 | 35 (30)	84 | 402 (30)	1,322 | 2,409 (29)	11,620 | 33,629 (21)	21,646 | 26,223 (7)	17,619 | 13,314 (2)	/

in each single phase (one for Simplex but 2^k for EVOP), the total number of measurements required to reach the optimum is higher in EVOP since both methods traverse along the fastest path possible. This difference is most pronounced for high dimensions. For $k = 8$, EVOP will need 256 measurements for each new phase, while Simplex still needs only one. An illustration of the improvement path for $k = 2$ is given for each method in figure 5.4. This superior behaviour of Simplex explains the huge success of it and especially its derivative method, Variable Simplex or Nelder-Mead Simplex [109], in numerical optimization.

The effect of the number of measurements for each phase is far more pronounced than the relatively small increase in step size for EVOP (figure 5.3) which gives EVOP a small advantage. In general, choosing a large factorstep seems a good choice for lowering the number of measurements necessary to attain the optimal region. However, choosing the factorstep too large can have negative effects: the optimum might not be located accurately enough or might be missed completely, should the method step over the optimal region. In the simulations discussed in this paper, this did not occur, not even with the chosen maximal step size of 10%.

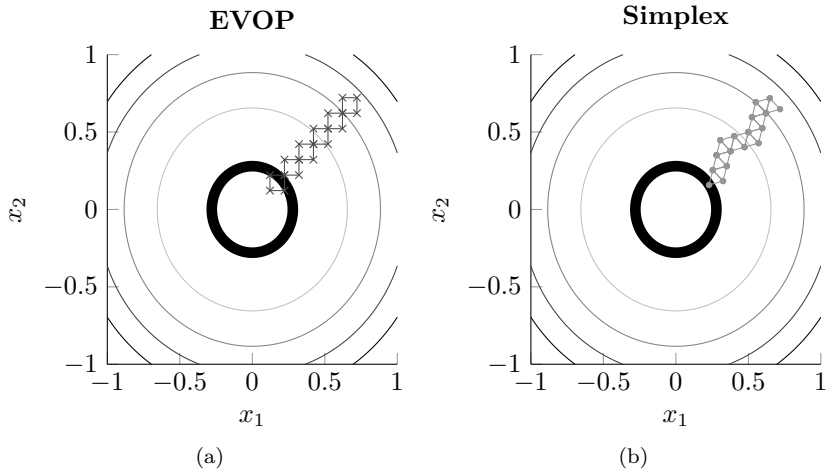


Figure 5.4: 2-factor improvement for $dx = 5\%$ and no noise; optimum region denoted by central black circle: (a) EVOP, (b) Simplex.

Influence Of The SNR

As explained above, a measure for the robustness of the methods against noise is the IQR of the number of measurements necessary to reach the region of the optimum for all 30 repetitions and consequently, it quantifies the amount of random behaviour in the improvement path. This random behaviour is illustrated for a two-factor improvement in figure 5.5 where two repetitions of EVOP and Simplex improvements are plotted for the same simulation settings (the Simplex results are mirrored on the figure to show both EVOP and Simplex simultaneously). Remark that in an EVOP improvement a stationary phase can be encountered and that for Simplex new measurement points can coincide with older points if one point is retained (incorrectly) as the optimal settings long enough so that the method circles around it. This behaviour is not visible in the figure so that the number of measurements denoted in the tables might seem higher when compared to the plotted ones.

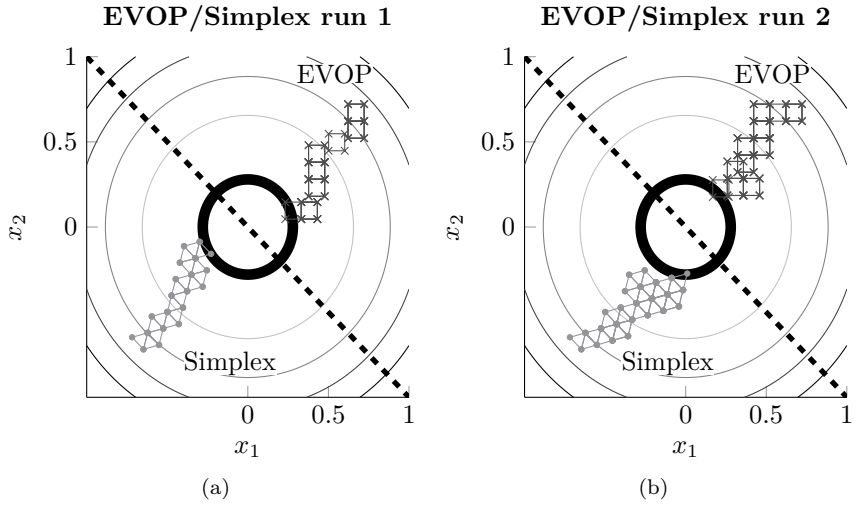


Figure 5.5: EVOP 2-factor improvement, $dx = 5\%$, SNR $= 50$, optimum region denoted by central black circle, EVOP in top right part of figures, Simplex in bottom left part: (a) First repitition, (b) Second repitition.

Increasing the noise-level (decreasing the SNR) while keeping the factorstep and number of factors constant, shows that EVOP is more robust to the introduced noise than Simplex. This can be validated by looking at the change in IQR for the EVOP and Simplex method respectively. For example for $dx = 5\%, k = 6$: the IQR for EVOP stays zero until the SNR is lowered to 250, while the IQR for

Simplex increases by 44 measurements between the deterministic case and the simulations with SNR = 1000. Examining table 5.1 and 5.2, this observation holds in general: the increase in IQR for EVOP is much slower than the increase for Simplex, when keeping the dx and k constant and lowering the SNR.

The power to deal with noise in the EVOP method is contained in the fact that the variation of the underlying process can be estimated by statistics because more points are measured than there are parameters in the used model.

Influence Of The Number Of Factors k

Increasing the dimensionality k will increase the number of measurements in every EVOP phase to 2^k, while for Simplex only the number of initial measurements $(k+1)$ changes and in every following phase only one measurement is used. The increase of measurements in every EVOP phase has a direct positive impact on the robustness of the method against noise. Since a main effects only model is fitted in every EVOP phase, only a small amount of the degrees of freedom for error are used to estimate the model. When estimating a main effects model, $k + 1$ degrees of freedom are necessary for the estimation of the model terms, all other degrees of freedom are used for the error sum of squares. For the two-factor case, there are three model degrees of freedom and only one degree of freedom is left for the error sum of squares. Contrarily, for eight factors there are nine model degrees of freedom and $(256 - 9) = 247$ degrees of freedom for the error sum of squares, increasing the power of the method for higher dimensions. So inherently, EVOP as implemented in this study has a higher power at higher dimensions—an observation that was also made when discussing the influence of the SNR above.

As an example: for a fixed factorstep $dx = 5\%$ and fixed signal-to-noise ratio SNR = 500, increasing the number of factors k, increases the power of EVOP which results in a decrease in the IQR to zero when $k \geqslant 5$. It is important to note that the IQR is expressed in number of measurements and that each EVOP phase consists out of 2^k measurements. Expressed in EVOP phases the IQR decreases from six $(k = 2)$, to zero $(k \geqslant 5)$ phases.

When considering the same fixed SNR and dx for the Simplex methodology, the IQR increases rapidly when the number of factors is increased. Simplex only adds one measurement in every phase and estimates no model, using only the acquired response values to decide on a new direction. It is for this reason that the method performs poorly when the dimensionality of the problem increases for the decision of the direction is solely based on $k + 1$ response values from which a new direction is estimated, not taking into account noise or the relative importance of each direction.

The (statistical) power of the traditional full factorial design that forms the base of EVOP is also its weakness and chapter 7 will investigate how statistical power influences the EVOP improvement. While the power of the method increases with increasing k, the experimental cost per phase increases with a power of two. In practice, EVOP improvements with more than five variables become too big to run. For instance, with six factors each EVOP phase will need 64 measurements and depending on the process this might make the improvement so costly that its benefit does not outweigh the experimentation cost.

The EVOP procedure could be improved for higher dimensions by using more efficient base designs. Since only a main effects model is estimated a fractional factorial might be used to drastically reduce the number of measurements, e.g. a full factorial for eight factors has 256 measurements while a fractional factorial can have as few as 16 measurements, this is investigated in chapter 6. Also, instead of running the classic EVOP where in each phase a full base design is performed, a pure steepest ascent approach could be used, as described in chapter 3. These are not the only improvements possible, and the whole field of online improvement based on EVOP-related methods is largely unexplored and deserves further attention, as the state of most contemporary processes is quantified by online sensors that can give immediate feedback on the response.

Influence Of The Factorstep dx

The factorstep determines the spread of the points within the same phase, and thus influences the differences in responses in this phase. How it affects this difference depends on the underlying "true" model behaviour. Two extreme situations are graphically shown for a simple one-variable quadratic function in figure 5.6: figure 5.6a depicts a situation far away from the optimum and figure 5.6b a situation where the two design points are located at either side of the optimum. Changing dx in the right hand case has no effect whatsoever on the response due to the fact that the points are located symmetrically around the optimum.

The left side figure shows a situation where the points are taken away from the optimum. This is the practical situation at the start of the simulations. Here, an increase in dx will result in an increase in the difference in response values, giving the method more power to detect the right direction of process improvement. Additional to the increased power, increasing the factorstep dx also results in a lower number of phases that is required to step towards the optimum, since the step size is proportional to the factorstep (see section 5.2.5).

When noise is introduced into the system the size of the factorstep plays an important role in the ability to discriminate the difference in response values.

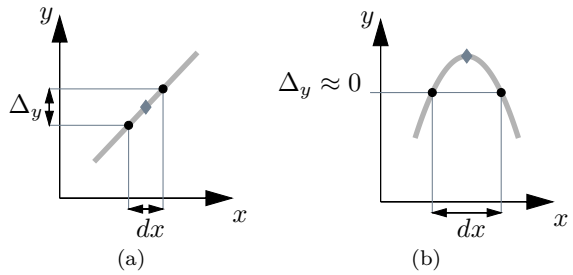


Figure 5.6: Simple example of the effect of the factorstep dx: (a) "away" from optimum, (b) around optimum.

For instance in the case of $k = 2, \text{SNR} = 1000$, an increase in dx decreases the relative IQR, showing that there is less uncertainty in the path the method takes towards the optimum. The interquartile range IQR is very large for most of the simulations where $dx = 1\%$ (or impossible to calculate when the method never reached the optimum in the 30 repetitions). This indicates that the factorstep is too small and the movement of the methods, especially Simplex, is largely driven by randomness (the method does not have the power to detect the effects, it cannot discriminate the response values in a correct way).

It is concluded from the results shown in table 5.2 that one should take the step size in Simplex large enough, relatively larger than for EVOP, to cope with the noise in the system. Setting the factorstep dx requires more attention for Simplex compared to EVOP, especially when the dimensionality increases. In any case, the difference in the measured responses should be significantly larger than the variation introduced by noise; otherwise the movement of the method will be largely driven by randomness.

In table 5.2 these influences can be observed clearly. In the most extreme case where $dx = 1\%$, even the introduction of a small amount of noise increases the number of measurements drastically. If dx increases, more noise can be introduced before the number of measurements increases to the same values of these related to a lower dx.

5.3.4 Overall Recommendations

In general, the Simplex method performs well for low-noise situations, and in the two-factor case it also outperforms EVOP in many high-noise situations. It is therefore not surprising that Simplex is used on lab scale equipment for the improvement of HPLC setups [15, 144] where the number of factors remains

limited, and SNR is typically high. For higher dimensionalities the increasing freedom of directions in which to move will hamper Simplex performance when dx or SNR decreases. For both methods the accuracy for estimating the direction of improvement has to be high enough, either by increasing the number of replications (not executed in this research since basic algorithms are compared) or by increasing dx.

All changed simulation settings have an effect on the performance of the methods and it is often the combination of these settings that is of importance. In figure 5.7 the comparison of both methods is visualized, where black denotes Simplex outperformed EVOP and white for the reverse (gray denotes where both methods performed poorly). For every combination of factorstep dx, number of factors k and signal-to-noise ratio SNR it is visualized which of the two methods performed best in terms of the combination of number of successes, median number of measurements and interquartile range.

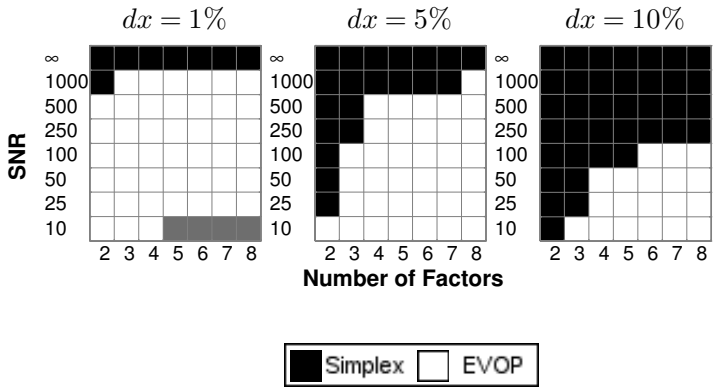


Figure 5.7: Binary comparison matrices of the methods (gray means no success for both methods).

This graphical comparison shows clearly that the factorstep is one of the most important settings to determine which method should be used as it not only affects the step size between improvement phases but also the ability to detect the direction of the optimum. Simplex is more susceptible to changes in the factorstep than EVOP. This can be clearly seen for $dx > 1\%$ where Simplex only outperforms EVOP in low-noise cases, while increasing the factorstep will allow Simplex to outperform EVOP in more cases.

However, the true effect of the factorstep on the response is often unknown at the start of an improvement and can change during the improvement depending on the behaviour of the underlying process. A change from the situation in

figure 5.6a to something similar to figure 5.6b has a large impact on the effect of the factorstep. Since this effect is unknown, the success of Simplex is uncertain as it heavily relies on the effect of the factorstep (i.e. Simplex fails with small changes in response values (determined by factorstep) and large noise-level). Moreover, in online improvement one usually starts the improvement close to the optimum and wishes to make the probability of unsaleable product as small as possible, thus making the factorstep small. For these improvements, Simplex is not suited since it performs poorly with small factorsteps. The decreasing performance of the Simplex method when the factorstep decreases also explains why the use of the Variable Simplex in production environments might be unreliable: when the Variable Simplex method reaches a perceived optimal region, it will start to shrink which will result in a decrease in the factorstep. Apart from the smallest factorstep, the Simplex method performs quite well for low-noise cases. Depending on dx, even in higher dimensionalities Simplex outperforms EVOP if the noise-level is low enough.

This leads to the decision tree in figure 5.8 which allows a user to select the most suitable method for his circumstance. Specific values in this tree are based on this simulation study and should not be taken as a rule for a specific case. Experience with the process under study should allow to make appropriate assumptions for the decision rules.

The combined effect of factorstep and noise on the Simplex method is as expected from previous publications, such as the 1974 paper by Lowe [90] which was discussed in the introduction.

5.4 Conclusions

In this chapter the basic EVOP and Simplex sequential improvement methods were compared in a simulation study. Three settings were varied in the simulation: the signal-to-noise ratio SNR—which controls the amount of random noise—, the factorstep dx—which quantifies the distance between the initial measurement points—and the number of factors k. The simulation data showcases the strengths and weakness of both methods, as well as the importance of choosing an appropriate factorstep.

The factorstep dx is an essential parameter when performing the improvement, not only because the step size is a function of the factorstep, but also because it influences the robustness of the methods against noise. Choosing it too small will drastically decrease the robustness of the method against noise, while choosing it too large might not pinpoint the optimum accurately enough. In the presence of noise, the EVOP method is more robust to changes in the

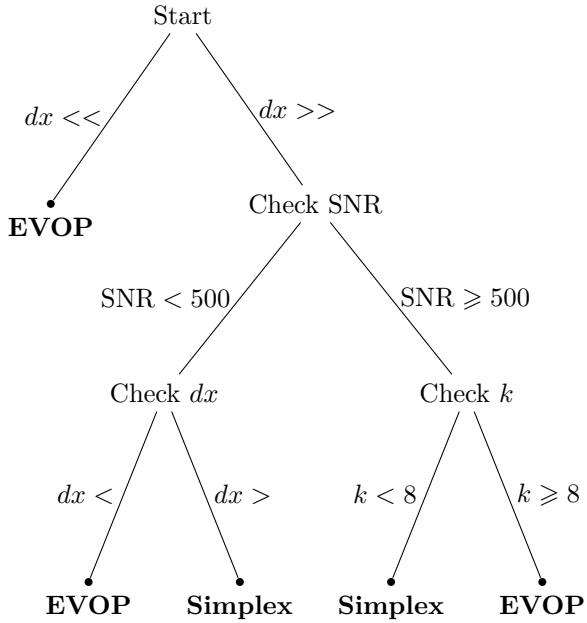


Figure 5.8: Decision tree for which method to use.

factorstep than the Simplex method since it incorporates an estimation of the noise present in the measurements. However, the effect of this setting is often unknown at the start of the improvement which decreases the usability of the Simplex method since its performance is tied in directly with the effect of the factorstep. If the goal of the improvement is the determination of the optimal region, and one expects that this region lies far away from the current operating conditions, the factorstep will be chosen larger and Simplex is the best choice for the improvement, otherwise EVOP is a safer choice that is more robust to changes in the factorstep.

Overall, EVOP proved to be an attractive method to improve processes characterized by a high amount of noise present in the system and/or a dimensionality above three factors. The higher the dimensionality, the higher the accuracy of estimation. As a disadvantage, a higher number of factors render the number of measurements prohibitive when implementing the full factorial design. Further research devoted to implementing more efficient designs in the EVOP methodology, such as fractional factorial designs, should be performed to make EVOP applicable to high dimensional cases. Additionally when no effect is active during one complete unreplicated design (which is called a cycle), the

same design can be repeated to get a better estimate of the process variance to discern the effects. As such one improvement phase consists of multiple cycles. In this study every phase consists out of only one cycle, resulting in multiple stationary phases (no effect is detected). The power to combine these to increase the accuracy of the estimated process variance is a powerful property of EVOP that was not incorporated in this study since the basics of both methods were to be compared.

Simplex, on the other hand, is the preferred choice when dealing with deterministic or low-noise systems and is also very susceptible to changes (for non-deterministic cases) in the factorstep setting, the smaller the factorstep becomes, the more unsure one is about the accuracy of the Simplex methodology. This implies that the implementation of Simplex needs more knowledge about the process to accurately determine the factorstep or that the factorstep should be chosen quite large. It is therefore a technique better suited to quickly reach the region of the optimum, if one is far away from it, and then switch to EVOP to accurately pin-point the optimum.

In a real process, the SNR is usually known and the influence of the factorstep might be derived from process knowledge. This allows us to make an initial determination about the method which is suited using the conclusions of the previous paragraphs. The proposed decision tree in figure 5.8 can then be used as a guideline to refine the design selection even more.

6 Efficient Designs For Evolutionary Operation

6.1 Introduction

In the previous chapter, the performance of EVOP and Simplex was compared based on a simulation study with up to eight factors. It was shown that Simplex performed poorly when dimensionality increased, especially when there was noise in the data (the practical setting). Because of this poor performance, even with low noise levels, Simplex is not a valid option when further increasing the dimensionality. EVOP on the other hand performed well but faces the drawback that the number of measurements that are required for each phase becomes prohibitive when one adheres to the full factorial design as a basis. Based on those observations, this chapter aims at investigating the use of more efficient designs in the framework of EVOP for higher dimensions.

Compact designs that still offer good properties can be generated by allowing for some degree of confounding (see chapter 3) which was recognized by Sir R.A. Fisher in 1926 [53]. It took some years before the main framework that detailed the concept of confounding was firmly developed by Yates around 1935 [161]. This framework lead to the use of designs such as the fractional factorial that—as the name suggest—considers only a fraction of the full factorial design. Fractional factorial designs were first discussed in this context by Finney in 1943 [52]. Information on fractional factorial designs was set forth in chapter 3 and will not be repeated in this chapter.

In order to investigate the appropriateness of fractional factorials in the EVOP context a simulation study was performed, much in analogy with the simulation settings in the previous chapter. The dimensionality of the study is extended to 16 factors, compared to eight factors in chapter 5. Simulations up to eight factors will be referred to as the "low dimensional cases" and a comparison between the full and fractional factorial will be made for these cases. For

the high dimensional cases ($k > 8$), using a full factorial is not feasible, and only results using the fractional factorial as a base design will be listed and discussed. For interpretation purposes, the concept of statistical power will be used extensively.

6.2 Materials & Methods

The designs were compared based on a simulation study with multiple scenarios. In analogy with chapter 5, the main settings under study are: (1) the perturbation size in every dimension (so for every factor), denoted factorstep dx_d; (2) the signal-to-noise ratio (SNR) and (3) the dimensionality k (number of factors) of the problem. The improvement is quantified using the same criteria that were previously used, being (1) the number of measurements that are needed to attain the defined optimal region, (2) the interquartile range IQR as a measure for the repeatability and (3) the number of cases in which the optimal region is reached ("success rate").

6.2.1 Underlying Model

The simulation model described by equation 6.1 is used, which is the same as in chapter 5. By using the same model it is possible to compare the results of the efficient designs with those of the previous full factorial simulations for "low" k. The experimental domain for each factor was bounded by $[-1; 1]$.

$$y_i = 200 - 128 \sum_{d=1}^{k} x_{d,i}^2 + \varepsilon_i \tag{6.1}$$

The number of factors k used in the simulations ranges from 4 to 16. This study does not start from two factors—as the previous simulation study—since no fractional factorial designs exist for two factors.

6.2.2 Simulation Settings

Three settings are controlled in the simulation study: the signal-to-noise ratio (SNR) to control the noise level, the size of the perturbations which are induced by setting the factorstep dx_d and the dimensionality of the problem k. The definition of these three settings was given in chapter 5. Most settings will be

kept the same as in the previous study. For ease of reference, the used settings are briefly summarized below:

Signal-to-Noise Ratio: The SNR was set at eight discrete values, being ∞ (no noise), 1000, 500, 250, 100, 50, 25 and 10. To test whether the improvements perform better than a random walk, additional simulations were performed in which the simulation model was replaced by Gaussian white noise with $\mu = 0$ and a chosen σ. These simulations are presented in the results under SNR $= 0$ as there is no signal, only noise.

Factorstep: The factorstep in every direction is taken equal so that the subscript d is omitted. The factorstep dx was fixed at three discrete values, being 1%, 5% and 10% of the experimental range for each factor.

Dimensionality: In order to investigate the properties of EVOP as function of the dimensionality of the problem, k was varied between 4 and 16.

Number of repetitions: The simulation was performed 30 times for each design in order to have a clear view on the performance of both designs in the presence of noise; at each combination of SNR, factorstep and dimension. This leads to a total of number of simulations of:

$$2\,(\text{designs}) \times 8\,(\text{noise-levels}) \times 3\,(\text{factorsteps}) \times 7\,(\text{dimensions})$$
$$\times\, 30\,(\text{repetitions}) = 10,080$$

Starting point: The starting point for all simulations was kept fixed at the coordinates determined by equation 6.2. With k as defined higher and l_{start} the radius of the chosen contour line, as before. The radius l_{start} was chosen to be 0.95 in all simulations thus fixing the noise-free initial response at 84.48.

$$\{x_{start,d}\}_{d=1}^{k} = \frac{l_{start}}{\sqrt{k}} \tag{6.2}$$

Stopping criterion: As before, it was opted not to use any (subjective) stopping criterion in this comparison study, but only to define the maximum number of measurements to complete. This number was chosen to be $51,200$ and was chosen to allow the design with the largest sample size per phase (the full factorial for $k = 8$) to use 200 phases.

Success rate, number of measurements needed, number of phases needed: A "success" was defined as an improvement that resulted in attaining a noise-free response value which is equal to or larger than 95% of the theoretical optimum.

The data in the tables is presented in the median number of measurements needed to have at least one of the noise-free responses of a phase within the optimal region, median (N), and the median number of phases, median (N/n_T) with n_T the sample size per phase.

6.2.3 Evolutionary Operation And Statistical Power

The implemented algorithm is the same as the one described in detail in chapter 3 and implemented in the previous simulation study. The definition of the step size δ_{EVOP} is the same as in the previous simulations (equation 6.3).

$$\delta_{EVOP} = ||\boldsymbol{\delta}_{\mathbf{EVOP}}|| = dx \cdot \sqrt{f_a} \tag{6.3}$$

where f_a denotes the number of active (significant) factors after using a stepwise regression procedure with $p_{enter} = 0.05$ and $p_{remove} = 0.1$. As before, the starting point $\mathbf{x_{start}}$ will refer to the centre of the first design, with the design region bounded by $\{x_{start,d} \pm 0.5 \cdot dx\}_{d=1}^{k}$. Fractional factorials will be used as the efficient designs in the simulations. A main effects model is fitted to the data points and forms the basis of the calculation of the direction of improvement as explained in chapter 3.

To easily distinguish between the designs, the design notation introduced in chapter 3 is used. The two-level full factorial design is denoted by writing the number of measurements as $r2^k$ with r the amount of replication (in these studies one and thus omitted) and k the number of factors. The fractional factorial design is denoted by writing the number measurements as $r2_{Res}^{k-p_f}$, the number of measurements in such a fractional factorial are a $1/2^{p_f}$ fraction of the full factorial measurements and the subscript that denotes the resolution Res in roman numerals. For the fractional factorial designs used in the simulation study, the resolution is minimum III.

Reducing the number of measurements (from full to fractional factorial) while estimating the same model as we did for the full factorial, will also reduce the number of degrees of freedom for error df_ε. The reduced sample size and thus df_ε, will have a direct impact on the statistical power of the design. The statistical power is the probability of accepting the alternative hypothesis when the alternative hypothesis is true—that is, the ability of a test to detect an effect, if the effect actually exists. In this case the power is the probability to detect a main effect (of a certain size), given a certain noise-level (determined by the SNR), if this main effect actually exists.

Consider the general test on a main effect coefficient β_d from a regression context:

$$H_0 : \beta_d = 0$$

$$H_a : \beta_d \neq 0$$

For an orthogonal two-level design in coded units (the low factor level transformed to -1 and the high factor level to $+1$), the power π of this test is given by equation 6.4. In this equation T_{NC} is the noncentral t-distribution with degrees of freedom $\nu = n_T - 1 - f_t$ and ϕ the noncentrality measure; t the t-distribution with degrees of freedom ν on which a two-sided test with significance level α is executed. The degrees of freedom for the distributions are defined by n_T the total number of measurements and f_t the number of parameters in the estimated model (excluding the intercept), which in this case is the number of active main effects f_a when fitting a stepwise linear regression model ($f_t = f_a$).

$$\begin{aligned}\pi =& P\left\{-t\left(1-{}^{\alpha}\!/_{2}; n_T - 1 - f_t\right) < T_{NC}\right\} \\ &+ P\left\{t\left(1-{}^{\alpha}\!/_{2}; n_T - 1 - f_t\right) > T_{NC}\right\}\end{aligned} \tag{6.4}$$

The noncentrality measure ϕ to detect a coefficient for a given coded (linear) regression coefficient $\beta_{d,c}$ can then be written as equation 6.5 with σ the error standard deviation.

$$\phi = \frac{\beta_{d,c}}{\frac{\sigma}{\sqrt{n_T}}} = \sqrt{n_T}\frac{\beta_{d,c}}{\sigma} \tag{6.5}$$

As can be seen from equation 6.4 and 6.5, the statistical power to detect a difference is directly related to the sample size n_T, the amount of noise in the system σ, the number of parameters in the model f_t and the size of the coded effect $\beta_{d,c}$ (size of the regression coefficient for coded factor levels) one is interested in detecting. For more information about power analysis, the reader is referred to [57, 82, 95]. For all power calculations in this text the level of significance is set at $\alpha = 0.05$. In appendix B, an example Matlab® (Matlab R2010b, The Mathworks Inc., Natick, Massachusetts USA) program is given that implements these equations to calculate the power.

In a classical DOE setting one would execute a power analysis before actually starting the experiment, as to determine the sample size necessary to detect a difference $\beta_{d,c}$ of a certain size. However, the true size of the main effects is not known at the start of the experiment (indeed if they were known there would be no need for experimentation), nor might an estimator of σ be known. Therefore a sample size estimation is often done by defining the ratio $\beta_{d,c}/\sigma$, which quantifies the size of the effect to be detected relative to the size of the noise in the system σ.

In this text, the sample size is kept to the bare minimum: no replication of the designs is performed and the smallest feasible fractional factorial is used which is of resolution III or IV, depending on the number of factors. This leads to designs which have a limited power to detect a given effect, and the discussion of the results are backed using the power of those designs, which show to be quite different for the different dimensions. It is noted that in the field of Optimal Design one could choose designs with a (close to) fixed power irrespective of the dimensionality, but this falls beyond the scope of this comparison study and is reserved for future research directions and will be used extensively in chapter 7. A lower power will result in more randomness in the improvement paths, a measure of which is expressed by the interquartile range IQR.

Table 6.1: Number of measurements n_T in the full(F) and fractional factorial (Fr) designs, degrees of freedom for error (df_ε) when estimating a main effects model and power for $\frac{\beta_{d,c}}{\sigma} = 0.5$, $\alpha = 0.05$.

	Full Factorial				Fractional Factorial			
# Factors	**Design**	n_T	$df_{\varepsilon F}$	π_F	**Design**	n_T	$df_{\varepsilon Fr}$	π_{Fr}
4	2^4	16	11	0.45	2^{4-1}_{IV}	8	3	0.17
6	2^6	64	57	0.98	2^{6-3}_{III}	8	1	0.09
8	2^8	256	247	1.00	2^{8-4}_{IV}	16	7	0.41
10	-	-	-	-	2^{10-6}_{III}	16	5	0.37
12	-	-	-	-	2^{12-8}_{III}	16	3	0.29
14	-	-	-	-	2^{14-10}_{III}	16	1	0.13
16	-	-	-	-	2^{16-11}_{IV}	32	15	0.75

When the dimensionality increases, the number of measurements and subsequently the power increases. However, the number of measurements becomes prohibitive and the resulting power is excessively high. A lower power could still be acceptable and opens the door for using compact designs, such as the fractional factorial. Table 6.1 shows—for the full and fractional factorial designs used—the number of measurements, the degrees of freedom for error df_ε when the full main effects model is estimated and the power π to detect a

ratio $\beta/\sigma = 0.5$. The example ratio is chosen to detect effects smaller than the noise present in the system as this is often the case when applying EVOP in a small design region and which usually leads to small parameter estimates.

One can easily see that the minimalistic fractional factorial designs are generally not suitable for finding small effects; only for $k = 16$ the power for detecting an effect half the size of the error standard deviation is larger than 0.7. The power is especially low for $k = 4, 6$ and 14. It is expected that the EVOP simulations using such designs will perform weak. However, as will be explained later on, a surprising amount of improvement is still possible.

The aforementioned tables are an example when one wishes to detect $\beta_{d,c}/\sigma = 0.5$. If the ratio is larger, the power will also increase, whilst the power will decrease when the ratio is smaller. When performing classical design of experiments the design region usually equals the experimental domain. In EVOP, however, the design region is much smaller than the experimental domain and moves through the experimental domain after the conclusion of each phase. This has two implications: (1) the main effects in EVOP are, in general, smaller than in classical experimentation, (2) in every phase the main effects will be different as the behaviour of the underlying model in this small region changes.

The fact that the main effects in EVOP are, in general, smaller than in classical experimentation leads to a reduced power if the same design is used. Therefore, replication of the design can be considered to increase the power. However, such replication is not necessarily required depending on the demands imposed on the improvement as will be discussed based on the simulation results. The simulation results involve no replication, to showcase the most extreme possibility.

As stated previously, when moving in the experimental domain the main effects in EVOP can (and in this simulation will) change from one phase to another due to the fact that the underlying process model exhibits other behaviour in the new region. Since in this study the underlying simulation model is known, one can calculate the true main effects for a given factorstep and design region to get a feeling how they change. Since the function is pure quadratic and is approximated in every design region by a linear model, the first order partial derivatives of the simulation function (equation 6.6) give a clear indication on how the relative magnitude of the main effects will change when moving through the experimental domain.

$$\frac{\partial y}{\partial x_d} = -64x_d \tag{6.6}$$

It is evident from this equation that, the closer one gets to the optimum, the smaller the main effects become, which will lower the ratio $\beta_{d,c}/\sigma$. As a result,

the power to detect an effect becomes lower if one approaches the optimum. This observation suggests that the type of design that fits the improvement procedure "from start (far away from optimum) to finish (near optimum)" changes along the procedure. However, in the typical setting considered throughout this Ph.D. one does not know how far the optimum is and what real size of the β's is. This implies that in a real situation an appropriate sample size to use throughout the improvement should be determined with careful consideration.

Decreasing the sample size also decreases the power and leads to more randomness in the improvement paths. The number of phases that is needed is the most informative indicator when quantifying the amount of randomness in these paths. Indeed, when comparing the full and fractional design for a given k, the number of measurements n_T within a phase is very different, so that the number of measurements is not ideal for comparison purposes. Remark, however, that the total number of measurements N (sum of all measurements in every executed phase, thus a multiple of n_T) required for improvement is of utmost importance in practice, since typically this number needs to be minimized. Therefore both the number of measurements as well as the number of phases (and their related IQR) are calculated for discussion purposes.

For the non-deterministic cases (SNR $\neq \infty$), changing the simulation settings will have a direct effect on the statistical power of the method. Equations 6.4 and 6.5 show that the degrees of freedom for error df_ε, the sample size n_T, the amount of noise in the process as expressed by σ and the size of the main effects $\beta_{d,c}$ will change the power. These four parameters can be directly related to the three simulation settings. (1) Changing dx will change the size of the coded main effects $\beta_{d,c}$: the smaller dx, the smaller the main effects, the lower the power. (2) Changing the signal-to-noise ratio SNR will change the amount of noise in the process σ: the lower the SNR, the higher σ, the lower the power. (3) Changing the dimensionality k has an impact on the degrees of freedom for error df_ε and can have an impact on the sample size n_T (always in case of the full factorial, in case of the fractional factorial the sample size will not increase with every increase of the dimensionality). Summarized for the full factorial: the higher the dimensionality k, the higher n_T and df_ε, the higher the power. For the fractional factorial the influence is not as straightforward since the sample size n_T does not necessarily increase when increasing k, this has an effect on the degrees of freedom for error df_ε and it is easiest to reference table 6.1 to see the changes in power when changing the dimensionality k.

Example Of Power Analysis For Classical Design

This section shows how an a priori sample size estimation to determine the minimal sample size is executed for a classical experimentation procedure i.e. only one phase is measured, usually as large as the experimental region. This section is presented to exemplify how the concept of statistical power can be used—a priori—for sample size estimation. This minimal sample size can be used to determine an appropriate statistical design and the degree of replication for this design (if needed). A sample size estimation based on statistical power can be performed using formulas to approximate the sample size [95], using look-up tables and interpolating between the presented values in the tables [82] or by using software, such as the example Matlab® program that is presented in appendix C for orthogonal two-level designs.

In this example an experiment with six factors ($k = 6$) and a model with all linear terms and intercept has to be estimated ($f_t = k$). One wishes to be able to detect a coded main effect that is half the size of the error standard deviation ($\beta_{d,c}/\sigma = 0.5$) with a power of $\pi = 0.5$ (usually the power is significantly higher in classical experimentation, but 0.5 will be used to allow some more freedom in design selection) at a significance level $\alpha = 0.05$. Using the program in appendix C shows that the sample size has to be at least $n_T = 19$ to achieve a power that is 0.5 or higher (in case of $n_T = 19$, the program calculates the power as $\pi = 0.52$). This Matlab® program uses the formulas for power in equations 6.4 and 6.5, which hold for a (coded) orthogonal, two-level design. Thus the experimenter has to find an orthogonal two-level design which has a sample size of at least $n_T = 19$. An unreplicated fractional factorial design of resolution III, 2^{6-3}_{III}, has 8 measurements, a fractional factorial design of resolution IV, 2^{6-2}_{IV}, has 16 measurements. One could replicate these designs to get a sample size that fits the requirements: replicate the resolution III design three times for $3 \cdot 2^{6-3}_{III} = 24$ measurements or replicate the resolution IV design two times for $2 \cdot 2^{6-2}_{IV} = 32$ measurements. Other types of orthogonal two-level designs could also be used, such as a Plackett-Burman design [120] which can be constructed for several sample sizes where the sample size is a multiple of 4. The nearest multiple of four to $n_T = 19$ is a sample size of $n_T = 20$.

Table 6.2 summarizes the sample sizes of these three two-level designs with the resulting power to detect $\beta/\sigma = 0.5$. The experimenter can now choose which design to use. If the sample size is the limiting factor, the Plackett-Burman design will be selected as it fits the requirements for the power and has the smallest sample size of the three proposed designs.

Table 6.2: Summary of different two-level designs for sample size estimation for $k = 6$, $\pi = 0.5$, $\frac{\beta_{d,c}}{\sigma} = 0.5$, $\alpha = 0.05$.

Design	n_T	π
PB	20	0.54
$3 \cdot 2^{6-3}_{III}$	24	0.64
$2 \cdot 2^{6-2}_{IV}$	32	0.78

6.3 Results & Discussion

The fractional factorial simulations were compared to the full factorial simulations (up to eight factors) of the previous simulation study given in chapter 5. The results of these previous simulations are pertinent for the comparison and are represented, for ease of reference, in table 6.3. For the high dimensional case, it is no longer feasible to execute a full factorial design sequentially and therefore no comparison between the two designs is possible. In table 6.4 an overview of the simulation results for the fractional factorial EVOP is given in the number of measurements. Table 6.5 shows the results in phases for both full and fractional factorial EVOP (rounded towards plus infinity for readability) for the low dimensional cases. Relevant information is shown for each combination of the factorstep dx (expressed as a percentage of the experimental range), SNR and dimensionality k. All values in these tables have been rounded towards plus infinity for readability.

6.3.1 Influence Of Simulation Settings For The Low Dimensional Case

The influence of the effect of the different simulation settings (signal-to-noise ratio SNR, factorstep dx and dimensionality k) on EVOP was already discussed in detail in chapter 5 of this dissertation. In this section, the simulations for the traditional (full factorial) and efficient (fractional factorial) designs will be compared for the low dimensional case ($k \leqslant 8$).

Deterministic Simulations

When the simulation is deterministic (SNR $= \infty$), the shortest path towards the optimum was taken irrespective of the dimensionality. In this case, the fractional factorial has a clear advantage as the number of measurements per

Table 6.3: EVOP full factorial simulation data [median of number of measurements needed to reach optimum | interquartile range (number of times optimum was reached)], number of experiments per phase denoted as n_T.

		Number of Factors		
dx	SNR	4 ($n_T = 16$)	6 ($n_T = 64$)	8 ($n_T = 256$)
1%	∞	288 | 0 (30)	960 | 0 (30)	3,328 | 0 (30)
	1,000	1,344 | 304 (30)	2,880 | 512 (30)	6,144 | 768 (30)
	500	1,984 | 432 (30)	4,640 | 576 (30)	9,856 | 1,280 (30)
	250	2,712 | 1,008 (30)	6,848 | 1,216 (30)	13,824 | 2,560 (30)
	100	4,696 | 1,440 (30)	11,616 | 2,304 (30)	25,216 | 5,120 (30)
	50	5,976 | 1,744 (30)	16,736 | 2,880 (30)	35,584 | 8,064 (29)
	25	9,872 | 5,232 (30)	22,624 | 8,064 (30)	46,080 | 6,080 (7)
	10	15,520 | 8,352 (30)	37,056 | 9,920 (20)	/
	0	/	/	/
5%	∞	64 | 0 (30)	256 | 0 (30)	768 | 0 (30)
	1,000	64 | 0 (30)	256 | 0 (30)	768 | 0 (30)
	500	64 | 16 (30)	256 | 0 (30)	768 | 0 (30)
	250	80 | 16 (30)	256 | 0 (30)	768 | 0 (30)
	100	144 | 48 (30)	320 | 64 (30)	1,024 | 256 (30)
	50	224 | 64 (30)	512 | 128 (30)	1,024 | 256 (30)
	25	312 | 144 (30)	640 | 192 (30)	1,792 | 512 (30)
	10	536 | 352 (30)	1376 | 384 (30)	2,560 | 768 (30)
	0	12,448 | 11,404 (11)	/	/
10%	∞	48 | 0 (30)	128 | 0 (30)	512 | 0 (30)
	1,000	48 | 0 (30)	128 | 0 (30)	512 | 0 (30)
	500	48 | 0 (30)	128 | 0 (30)	512 | 0 (30)
	250	48 | 0 (30)	128 | 0 (30)	512 | 0 (30)
	100	48 | 0 (30)	128 | 0 (30)	512 | 0 (30)
	50	48 | 0 (30)	128 | 0 (30)	512 | 0 (30)
	25	64 | 32 (30)	128 | 64 (30)	512 | 0 (30)
	10	128 | 64 (30)	256 | 128 (30)	512 | 256 (30)
	0	8,920 | 11,312 (30)	17,248 | 15,328 (4)	/

phase is smaller, i.e. for less experimental effort the region of the optimum was reached. For the fractional factorial the experimental effort is 50% less for $k = 4$ and up to 93.75% less for $k = 8$. This decrease is explained by the reduction of the number of measurements per phase (see table 6.1). Since the step size (equation 6.3) depends on the factorstep dx and the number of active effects, and all main effects are significant, the number of phases for both fractional and full factorial EVOP in the deterministic case are the same for a given dimensionality k and factorstep dx, and independent from the used design. One can also see that the number of phases necessary to reach the optimal region decreases as the dimensionality increases. This can be attributed to the increase in the stepsize as defined by equation 6.3.

Simulations With noise

Further comparisons will always refer to the situations in which noise is present in the process (the practical case). Looking at table 6.5, it can be seen that the full factorial is much more robust against noise as the number of phases and IQR increase much less than for the fractional factorial. However when one looks at tables 6.3 and 6.4, it shows that even though the EVOP scheme for the full factorial is more efficient, the total amount of measurements to reach the optimum is lower using the fractional factorial.

Table 6.4: EVOP fractional factorial simulation data [median of number of measurements necessary to reach optimum | interquartile range (number of times optimum was reached)], number of experiments per phase denoted as n_T.

		Number of Factors						
dx	SNR	4 ($n_T = 8$)	6 ($n_T = 8$)	8 ($n_T = 16$)	10 ($n_T = 16$)	12 ($n_T = 16$)	14 ($n_T = 16$)	16 ($n_T = 32$)
1%	∞	144 | 0 (30)	120 | 0 (30)	208 | 0 (30)	192 | 0 (30)	176 | 0 (30)	160 | 0 (30)	288 | 0 (30)
	1,000	936 | 144 (30)	748 | 200 (30)	1,792 | 256 (30)	1,864 | 336 (30)	1,800 | 416 (30)	1,392 | 368 (30)	3,584 | 896 (30)
	500	1,208 | 320 (30)	1,060 | 216 (30)	2,408 | 544 (30)	2,504 | 880 (30)	2,496 | 1,376 (30)	2,200 | 736 (30)	5,232 | 1,792 (30)
	250	1,736 | 496 (30)	1,580 | 744 (30)	3,696 | 1,232 (30)	4,000 | 1,568 (30)	4,384 | 960 (30)	4,264 | 1,936 (30)	9,008 | 4,448 (30)
	100	2,928 | 688 (30)	2,468 | 1,584 (30)	5,880 | 2,032 (30)	6,856 | 2,528 (30)	8,696 | 5,232 (30)	10,280 | 10,592 (30)	16,576 | 9,528 (29)
	50	3,996 | 1,880 (30)	3,604 | 2,648 (30)	9,456 | 2,448 (30)	13,120 | 5,840 (30)	12,632 | 10,352 (30)	21,568 | 15,928 (24)	32,640 | 13,272 (19)
	25	5,776 | 2,704 (30)	5,744 | 4,248 (30)	15,000 | 5,552 (30)	22,784 | 18,608 (28)	16,048 | 11,196 (13)	32,992 | 14,656 (8)	33,984 | 8,240 (9)
	10	10,428 | 6,072 (30)	11,752 | 7,960 (30)	27,432 | 15,240 (24)	36,120 | 14,736 (12)	37,296 | 24,424 (4)	/	/
	0	39,800 | 0 (1)	/	/	/	/	/	/
5%	∞	32 | 0 (30)	32 | 0 (30)	48 | 0 (30)	48 | 0 (30)	48 | 0 (30)	48 | 0 (30)	96 | 0 (30)
	1,000	48 | 16 (30)	72 | 24 (30)	64 | 16 (30)	80 | 32 (30)	96 | 32 (30)	80 | 32 (30)	128 | 32 (30)
	500	56 | 24 (30)	72 | 40 (30)	96 | 32 (30)	104 | 32 (30)	112 | 32 (30)	96 | 32 (30)	192 | 64 (30)
	250	64 | 24 (30)	80 | 24 (30)	128 | 48 (30)	128 | 32 (30)	128 | 64 (30)	136 | 64 (30)	272 | 96 (30)
	100	116 | 56 (30)	104 | 72 (30)	208 | 64 (30)	264 | 96 (30)	256 | 144 (30)	264 | 208 (30)	448 | 224 (30)
	50	168 | 80 (30)	128 | 64 (30)	320 | 160 (30)	304 | 208 (30)	376 | 240 (30)	488 | 432 (30)	720 | 448 (30)
	25	216 | 88 (30)	212 | 128 (30)	424 | 240 (30)	576 | 256 (30)	680 | 1,056 (30)	1,440 | 2,800 (30)	1,680 | 1,312 (30)
	10	368 | 208 (30)	472 | 432 (30)	848 | 480 (30)	1,576 | 1,360 (30)	2,712 | 3,264 (30)	14,512 | 10,228 (29)	8,640 | 13,472 (30)
	0	11,876 | 19,720 (26)	37,416 | 28,220 (5)	/	/	/	/	/
10%	∞	24 | 0 (30)	16 | 0 (30)	32 | 0 (30)	32 | 0 (30)	32 | 0 (30)	32 | 0 (30)	64 | 0 (30)
	1,000	24 | 0 (30)	16 | 8 (30)	32 | 0 (30)	32 | 0 (30)	32 | 0 (30)	56 | 32 (30)	64 | 0 (30)
	500	24 | 0 (30)	24 | 8 (30)	32 | 0 (30)	32 | 0 (30)	48 | 16 (30)	56 | 32 (30)	64 | 0 (30)
	250	24 | 8 (30)	32 | 16 (30)	32 | 0 (30)	48 | 16 (30)	48 | 16 (30)	56 | 32 (30)	96 | 0 (30)
	100	32 | 0 (30)	32 | 16 (30)	48 | 16 (30)	64 | 32 (30)	80 | 32 (30)	96 | 32 (30)	160 | 32 (30)
	50	44 | 16 (30)	48 | 16 (30)	72 | 16 (30)	88 | 64 (30)	96 | 48 (30)	104 | 80 (30)	256 | 192 (30)
	25	52 | 24 (30)	56 | 48 (30)	96 | 48 (30)	136 | 64 (30)	160 | 160 (30)	256 | 176 (30)	528 | 352 (30)
	10	72 | 64 (30)	84 | 48 (30)	160 | 80 (30)	208 | 176 (30)	336 | 256 (30)	1,096 | 1,024 (30)	928 | 1,024 (30)
	0	2,856 | 4,496 (30)	16,736 | 20,108 (29)	21,072 | 0 (1)	/	/	/	/

Table 6.5: EVOP full and fractional factorial simulations for the low dimensional case, expressed in phases: [median of phases to reach the optimum | interquartile range of phases to reach the optimum (success rate)], number of measurements per phase denoted as n_T.

		Number of Factors					
		4		6		8	
dx	SNR	Full ($n_T = 16$)	Frac. ($n_T = 8$)	Full ($n_T = 64$)	Frac. ($n_T = 8$)	Full ($n_T = 256$)	Frac. ($n_T = 16$)
1%	∞	18 | 0 (30)	18 | 0 (30)	15 | 0 (30)	15 | 0 (30)	13 | 0 (30)	13 | 0 (30)
	1,000	84 | 19 (30)	117 | 18 (30)	45 | 8 (30)	94 | 25 (30)	24 | 3 (30)	112 | 16 (30)
	500	124 | 27 (30)	151 | 40 (30)	73 | 9 (30)	133 | 27 (30)	39 | 5 (30)	151 | 34 (30)
	250	170 | 63 (30)	217 | 62 (30)	107 | 19 (30)	198 | 93 (30)	54 | 10 (30)	231 | 77 (30)
	100	294 | 90 (30)	366 | 86 (30)	182 | 36 (30)	309 | 198 (30)	99 | 20 (30)	368 | 127 (30)
	50	374 | 109 (30)	500 | 235 (30)	262 | 45 (30)	451 | 331 (30)	139 | 32 (29)	591 | 153 (30)
	25	617 | 327 (30)	722 | 338 (30)	354 | 126 (30)	718 | 531 (30)	180 | 24 (7)	938 | 347 (30)
	10	970 | 522 (30)	1,304 | 759 (30)	579 | 155 (20)	1,469 | 995 (30)	/ (0)	1,715 | 953 (24)
	0	/	4,975 | 0 (1)	/	/	/	/
5%	∞	4 | 0 (30)	4 | 0 (30)	4 | 0 (30)	4 | 0 (30)	3 | 0 (30)	3 | 0 (30)
	1,000	4 | 0 (30)	6 | 2 (30)	4 | 0 (30)	9 | 3 (30)	3 | 0 (30)	4 | 1 (30)
	500	4 | 1 (30)	7 | 3 (30)	4 | 0 (30)	9 | 5 (30)	3 | 0 (30)	6 | 2 (30)
	250	5 | 1 (30)	8 | 3 (30)	4 | 0 (30)	10 | 3 (30)	3 | 0 (30)	8 | 3 (30)
	100	9 | 3 (30)	15 | 7 (30)	5 | 1 (30)	13 | 9 (30)	4 | 1 (30)	13 | 4 (30)
	50	14 | 4 (30)	21 | 10 (30)	8 | 2 (30)	16 | 8 (30)	4 | 1 (30)	20 | 10 (30)
	25	20 | 9 (30)	27 | 11 (30)	10 | 3 (30)	27 | 16 (30)	7 | 2 (30)	27 | 15 (30)
	10	34 | 22 (30)	46 | 26 (30)	22 | 6 (30)	59 | 54 (30)	10 | 3 (30)	53 | 30 (30)
	0	778 | 713 (11)	1,485 | 2,465 (26)	/ (0)	4,677 | 3,528 (5)	/	/
10%	∞	3 | 0 (30)	3 | 0 (30)	2 | 0 (30)	2 | 0 (30)	2 | 0 (30)	2 | 0 (30)
	1,000	3 | 0 (30)	3 | 0 (30)	2 | 0 (30)	2 | 1 (30)	2 | 0 (30)	2 | 0 (30)
	500	3 | 0 (30)	3 | 0 (30)	2 | 0 (30)	3 | 1 (30)	2 | 0 (30)	2 | 0 (30)
	250	3 | 0 (30)	3 | 1 (30)	2 | 0 (30)	4 | 2 (30)	2 | 0 (30)	2 | 0 (30)
	100	3 | 0 (30)	4 | 0 (30)	2 | 0 (30)	4 | 2 (30)	2 | 0 (30)	3 | 1 (30)
	50	3 | 0 (30)	6 | 2 (30)	2 | 0 (30)	6 | 2 (30)	2 | 0 (30)	5 | 1 (30)
	25	4 | 2 (30)	7 | 3 (30)	2 | 1 (30)	7 | 6 (30)	2 | 0 (30)	6 | 3 (30)
	10	8 | 4 (30)	9 | 8 (30)	4 | 2 (30)	11 | 6 (30)	2 | 1 (30)	10 | 5 (30)
	0	558 | 707 (30)	357 | 562 (30)	1,078 | 958 (4)	2,092 | 2,514 (29)	/	1,317 | 0 (1)

The dissimilarity in the improvement efficiencies can be explained using the concept of statistical power. The power denotes the probability of detecting the right direction, and should be high for attaining a high efficiency in the improvement path. When the power becomes too low, the efficiency is lower causing a longer path in terms of the number of phases. When the difference in power between the full and fractional factorial is smallest, as is the case for $k = 4$, the results (table 6.5) of the full and fractional factorial are comparable in terms of the median number of phases and IQR. When increasing the dimensionality the results are not as comparable since the difference in power between the two designs is much larger, so that the fractional factorial designs require more phases to reach the optimum. A clear example is the case when $k = 6, dx = 5\%$ where it can be seen that the results (in phases) of the full factorial for $\text{SNR} = 25$ are comparable with those of the fractional factorial for $\text{SNR} = 250$. The mentioned case of $k = 6$ is an extreme one as mentioned above, with the full factorial having a power of 0.98 when ${}^{\beta_{d,c}}/_{\sigma} = 0.5$ and the corresponding fractional factorial showing a power of only 0.09. It is of interest to investigate whether this smaller power—which requires more phases—is counteracted by the fact that each phase has less measurements when compared to the full factorial design.

To ensure that the large number of measurements $(51,200)$ before an improvement is labelled a failure does not allow the simulation to reach the optimal region 30 times by chance, additional simulations were executed, labelled $\text{SNR} = 0$. These simulations consist of random walks in which the simulation model was replaced by Gaussian white noise. Comparing the results of the random walk with the other simulations, it is clear that EVOP performs better than a random walk for all simulation settings. For those random walk results for which the median and IQR were not zero, Kruskall-Wallis tests were performed to formally test whether the results for the random walk and EVOP simulations came from different distributions. These tests all indicated that the results came from different distributions at the 0.01 significance level.

In essence, one wishes to maximize the amount of information (high power) and expend a minimal experimental effort (low number of samples). These two requirements contradict each other since a higher power, while keeping all other settings the same, will require a larger sample size. Take, for instance, $dx = 5\%, k = 8, \text{SNR} = 100$: the full factorial EVOP needs 4 phases while the fractional factorial needs 13 phases which again shows that the improvement path for the former is more efficient. However, the experimental effort to reach the optimal region is almost five times smaller for the fractional factorial (208 points compared to 1024 points for the full factorial). Even though the fractional factorial improvement might be less efficient in number of phases, it is much more efficient in terms of experimental effort. So from a practical point of view,

the fractional factorial is advisable in this situation. Balancing the requirements of high power and minimal experimental effort is one of the challenges when executing any type of experimentation.

Apart from the dimensionality, the factorstep dx has also a marked influence on the efficiency. It determines the perturbation size in every dimension according to the definition in section 6.2.3. As explained in chapter 5, an increase in dx will result in an increase in the statistical power for the used simulation model. Such an increase also increases the size of the coded main effects $\beta_{d,c}$ (the factor levels are coded between -1 and 1, changing the range of the uncoded factor levels, will change the coded β's) and the effect will be easier to detect in terms of a statistical test. If one would perform a power analysis for two values of β where $\beta_1 < \beta_2$, when the noise-level stays the same, then, in general, the power to detect these effects is $\pi_1 < \pi_2$.

Summarizing the above observations, the statistical power is an essential parameter to determine the sample size for the improvement and should be high enough. The power can be increased in the simulations by increasing the factorstep dx (increase the coded regression coefficients β), by increasing the sample size (changing the design or amount of replication) or by changing the SNR (this can be achieved in the real process by increasing the number of samples per phase n_T, which results in a smaller $\sigma_{\beta_{d,c}}$).

An example of changing the factorstep and SNR in the simulations shows this clearly: when $k = 8$ and $dx = 10\%$ the IQR for the full factorial does not change (zero phases) when decreasing the SNR from 1000 to 25. When the factorstep is decreased to $dx = 5\%$ the IQR does not change from SNR = 1000 (zero phases) to SNR=250, after which it slowly increases until the IQR is three phases for SNR = 10. For the fractional factorial, on the other hand, the increase in IQR when decreasing the SNR is much larger when decreasing the factorstep from $dx = 10\%$ to $dx = 5\%$ for the same dimensionality. This is evidenced by looking at SNR = 10 where for $dx = 10\%$ the IQR is five phases while for $dx = 5\%$ the IQR is 30 phases. This shows that dx and SNR, which lower the power for both methods when decreased, will have a more marked influence on the fractional factorial then on the full factorial EVOP scheme.

The time to conclude one phase is much more important from the viewpoint of sequential experimentation (and related to this, the experimental effort per phase). Consecutive phases ("steps" in the improvement) are executed to gradually move the process towards the optimum. When a machine is not continuously operated (only during working hours, a limited batch of a certain product) the amount of time available for sampling is limited. Since EVOP is a sequential process with only small shifts in the factor settings, a number of consecutive phases is usually necessary to determine the optimal factor

levels. Should the sampling time per phase be large compared to the total available sampling time then it is not feasible to run the method. Therefore the number of measurements per phase should be kept to a minimum. This aspect is also important when optima drift in time as the rate of improvement (i.e. moving from one phase to the other) should not be slower than the speed of the time-drift.

Interpreting The Results From An Improvement Perspective

There are situations in which an EVOP simulation does not reach the optimal region. For instance for the full factorial for $k = 8$, $dx = 1\%$, SNR $= 10$, EVOP never reaches the optimal region. It is wrong to conclude that EVOP failed to improve the response since these simulations were labelled a failure. EVOP is an improvement method, so in real processes in which the response is not known, the interest is in seeing whether the application of EVOP did in fact improve the response. Let us compare two situations for $k = 8$, $dx = 1\%$: the situation of interest, SNR $= 10$, and a situation with less noise, SNR $= 1000$. In figure 6.1, the median noise-free response $y_{ci,\sigma=0}$ in the centre of the i-th phase is plotted against the number of measurements. This implies that $y_{ci,\sigma=0}$ is plotted after the conclusion of each phase (after each 256 measurements for the full factorial and 16 measurements for the fractional factorial) up to $51,200$ measurements. In addition the region of the optimum is displayed to show when a simulation is labelled as a success.

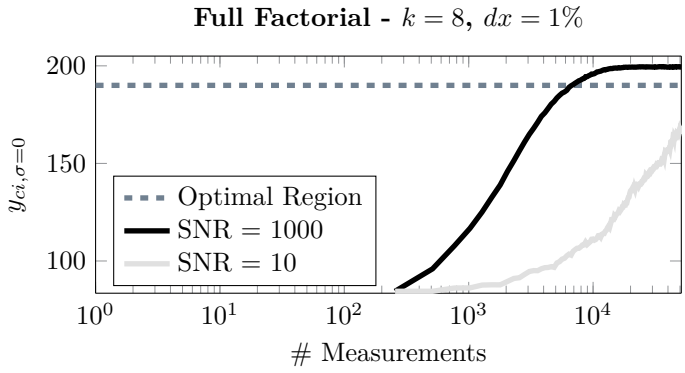


Figure 6.1: Median of noise-free response $y_{ci,\sigma=0}$ for full factorial, $k = 8$; $dx = 1\%$; SNR $= 10, 1000$.

As can be seen from this figure, even in the case of SNR $= 10$ there is a definite improvement in the response but since there is more noise, the median

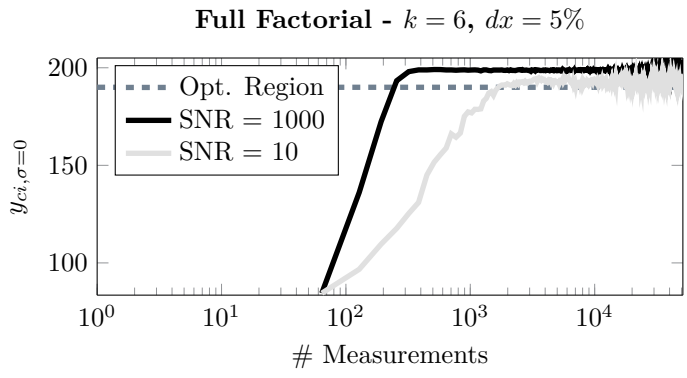
improvement in every phase is much smaller than in the case of SNR = 1000. This, combined with the small factorstep—which results in small moves between phases as per equation 6.3—and the large sample size per phase, ensures that the 51, 200 samples are insufficient to reach the region of the optimum (maybe it won't ever reach the optimum since the power will also keep on decreasing when one approaches the optimum due to the smaller coded regression coefficients). Yet EVOP does succeed in *improving* the response, as is its goal.

To investigate the effect of different sample sizes —and thus powers— between the full and fractional factorial designs, figures are created for $k = 6$, the case for which both full and fractional factorial are simulated and where the difference in power between the two designs is the most extreme. Both fractional and full factorial improvements are plotted with factorstep $dx = 5\%$. For each design two figures are created: on the first figures (figure 6.2a and 6.3a) the median noise-free response response ($y_{ci,\sigma=0}$) at the centre of each phase is plotted for SNR = 10 and SNR = 1000; and on the second figures (figure 6.2b and 6.3b) the median power of each consecutive phase for the aforementioned SNR-values is plotted. The power calculation is based on the largest linear effect $\beta_{d,c}$ of the full coded linear model (including a, intercept) for the current design region, which can be calculated since the underlying model is known.

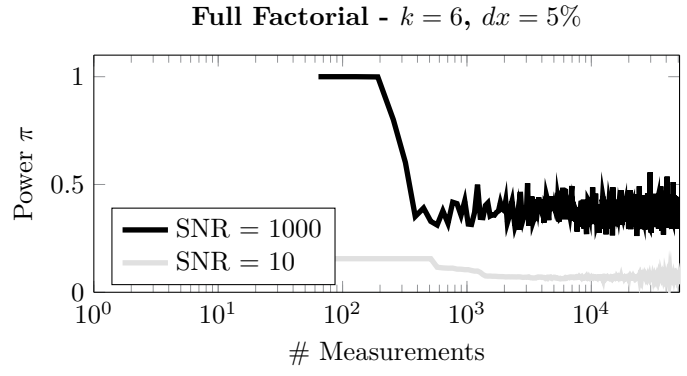
When one approaches the optimal region (effect size becomes smaller and the response moves closer to the dotted line on figures 6.2a and 6.3a), the power drops and, since the power for the fractional factorial is lower than the power of the full factorial, it will stop—on average—further away from the optimum (the fractional factorial fails to detect the smaller effects near the optimal region). This can be seen when comparing figures 6.2a and 6.3a where the median response at the centre of the design region after the improvement is larger for the full factorial compared to the fractional factorial. This difference becomes more pronounced when the noise decreases (SNR = 10 versus SNR = 1000).

Figures 6.2a and 6.3a show that the median end response for the full and fractional factorial after improvement is within the optimal region but that the full factorial will reach a higher response level than the fractional factorial. These figures also show that the initial improvement (slope) in the total number of measurements for both full and fractional factorial is equal for this SNR. Since the sample size per phase is smaller for the fractional factorial, the amount of improvement after every phase for the fractional factorial is much smaller than for the full factorial. Yet, the time it takes to conclude one phase is much shorter for the fractional factorial (smaller sample size) which is an advantage when the process is non-stationary (drifting in time).

Looking at the plotted examples, for the fractional factorial in case SNR = 10 the resulting power is extremely low and the power curve is almost completely flat

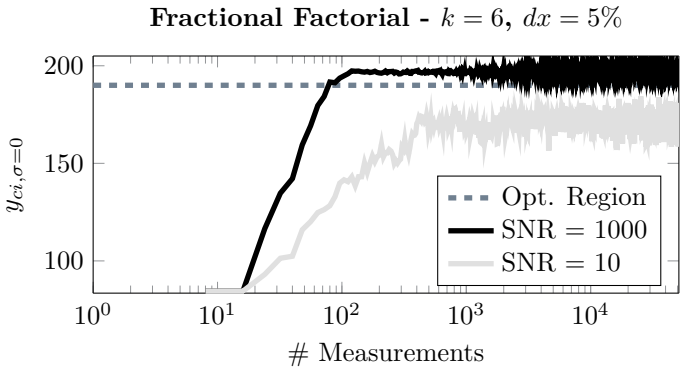


(a)

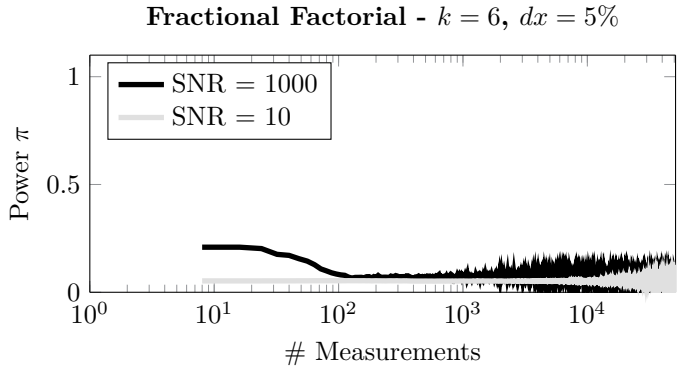


(b)

Figure 6.2: Full factorial, $k = 6$; $dx = 5\%$; SNR $= 10, 1000$: (a) median of noise-free response $y_{ci,\sigma=0}$, (b) median of power for consecutive phases.



(a)



(b)

Figure 6.3: Fractional factorial, $k = 6$; $dx = 5\%$; SNR $= 10, 1000$:(a) median of noise-free response $y_{ci,\sigma=0}$, (b) median of power for consecutive phases.

and it might come as a surprise that the method still achieves an improvement when the power is extremely low. This can be attributed to the use of the stepwise regression. The power to detect an effect depends on the degrees of freedom for error df_ε that are related to the number of parameters f_a in the model. The more parameters are removed from the model, the higher the power to detect the remaining effects. This will play an important role when the effect size β is relatively large ("far away" from the optimum) but will have less and less effect as the effect size decreases.

When the degrees of freedom for error are small, every parameter removed from the model will have a large impact on the power. This is elucidated in table 6.6 where—for the used 2^{6-3}_{III} design—the power is calculated for main effects models ranging from all main effects ($f_a = 6$) to one main effect ($f_a = 1$) for a ratio $\beta/\sigma = 0.5$. The removal of one parameter from the full main effects model will increase the power by 5%, removing an additional parameter will only result in an additional increase of 3%. The higher the degrees of freedom for error, the less the influence of the removal of one parameter on the power. It should be noted that the removal of parameters from the model and the relation between degrees of freedom for error and power is generally more complex—see equations 6.4 and 6.5—but the previous example gives a good indication of how the change in df_ε changes the statistical power.

Table 6.6: Number of parameters f_a , degrees of freedom for error df_ε in model and related power π for a 2^{6-3}_{III} design with $\frac{\beta_{d,c}}{\sigma} = 0.5$, $\alpha = 0.05$.

f_a	df_ε	π
6	2	0.09
5	3	0.14
4	4	0.17
3	5	0.19
2	6	0.21
1	7	0.22

This evidently leads to the question when which design should be implemented in EVOP. The answer to this is not straightforward as it depends on the quality of the improvement. In section 6.3.3, an overview of how such quality might be rated in industry is given.

6.3.2 Feasibility Of Using Efficient Designs In The High-Dimensional Case

When the dimensionality is high—and a full factorial is used—one would expend too much experimental effort, i.e. every phase would require too much time to conclude to achieve the desired improvement when experimentation time is limited (e.g. a machine that runs only 8 hours a day). Even if the improvement path is not as efficient as is the case for the full factorial, the significantly reduced experimental effort makes it clear that efficient designs are a much-needed improvement over the traditional full factorial design.

Although no comparison can be made with the classical full factorial in the high dimensional case ($k > 8$), the simulations allow to draw conclusions about the feasibility of using the fractional factorial for this case. Using the data in tables 6.5 and 6.7—fractional factorial results for the low and high dimensional case in phases respectively—one can surmise that the results for the number of phases are comparable when the noise-level is not too high (SNR $\geqslant 50$). The data from these tables for $k = 4, 8, 12, 16$ is represented graphically in figure 6.4. It can be seen clearly that the fractional factorial designs perform equally efficient in high dimensionalities (when compared to low dimensionalities) unless the SNR becomes too small. This is logical since the small sample size of the fractional factorial will have a very low power at low SNR values. The question arises once more what the optimal power for an EVOP improvement is. This will be handled in the next chapter.

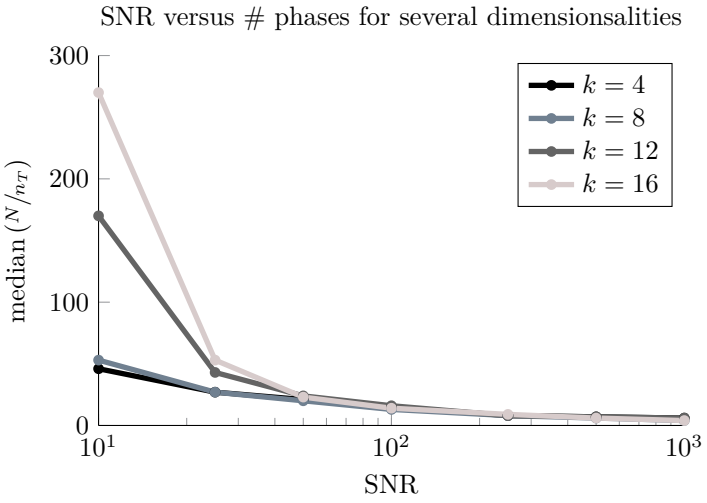


Figure 6.4: SNR versus median number of phases for $k = 4, 8, 12, 16$.

Table 6.7: EVOP fractional factorial simulations for the high dimensional case, expressed in phases: [median of phases to reach the optimum | interquartile range of phases to reach the optimum (success rate)].

		Number of Factors			
dx	SNR	10 ($n_T = 16$)	12 ($n_T = 16$)	14 ($n_T = 16$)	16 ($n_T = 32$)
1%	∞	12 | 0 (30)	11 | 0 (30)	10 | 0 (30)	9 | 0 (30)
	1,000	117 | 21 (30)	113 | 26 (30)	87 | 23 (30)	112 | 28 (30)
	500	157 | 55 (30)	156 | 86 (30)	138 | 46 (30)	164 | 56 (30)
	250	250 | 98 (30)	274 | 60 (30)	267 | 121 (30)	282 | 139 (30)
	100	429 | 158 (30)	544 | 327 (30)	643 | 662 (30)	518 | 298 (29)
	50	820 | 365 (30)	790 | 647 (30)	1,348 | 996 (24)	1,020 | 415 (19)
	25	1,424 | 1,163 (28)	1,003 | 700 (13)	2,062 | 916 (8)	1,062 | 258 (9)
	10	2,258 | 921 (12)	2,331 | 1,527 (4)	/	/
5%	∞	3 | 0 (30)	3 | 0 (30)	3 | 0 (30)	3 | 0 (30)
	1000	5 | 2 (30)	6 | 2 (30)	5 | 2 (30)	4 | 1 (30)
	500	7 | 2 (30)	7 | 2 (30)	6 | 2 (30)	6 | 2 (30)
	250	8 | 2 (30)	8 | 4 (30)	9 | 4 (30)	9 | 3 (30)
	100	17 | 6 (30)	16 | 9 (30)	17 | 13 (30)	14 | 7 (30)
	50	19 | 13 (30)	24 | 15 (30)	31 | 27 (30)	23 | 14 (30)
	25	36 | 16 (30)	43 | 66 (30)	90 | 175 (30)	53 | 41 (30)
	10	99 | 85 (30)	170 | 204 (30)	907 | 640 (29)	270 | 421 (30)
10%	∞	2 | 0 (30)	2 | 0 (30)	2 | 0 (30)	2 | 0 (30)
	1000	2 | 0 (30)	2 | 0 (30)	4 | 2 (30)	2 | 0 (30)
	500	2 | 0 (30)	3 | 1 (30)	4 | 2 (30)	2 | 0 (30)
	250	3 | 1 (30)	3 | 1 (30)	4 | 2 (30)	3 | 0 (30)
	100	4 | 2 (30)	5 | 2 (30)	6 | 2 (30)	5 | 1 (30)
	50	6 | 4 (30)	6 | 3 (30)	7 | 5 (30)	8 | 6 (30)
	25	9 | 4 (30)	10 | 10 (30)	16 | 11 (30)	17 | 11 (30)
	10	13 | 11 (30)	21 | 16 (30)	69 | 64 (30)	29 | 32 (30)

6.3.3 Improvement Quality

When talking about the improvement of a process a clear distinction can be made between two measures of improvement "quality", being:

1. The *quality of output* after improvement compared to the quality before improvement;
2. The *rate of improvement* while running EVOP, which is the change in average response (quality) between phases.

The first measure "*quality of output*" relates to the improvement in quality after an EVOP scheme is run. In other words: how much better (or worse) is the output compared to the initial situation. Since the true optimum is not known when these methods are applied in practical situations, this is the only way to quantify the overall improvement in response.

It is shown that—for the executed simulations—the effect size decreases as the method gets closer to the optimum and, by extension, the power drops. At a certain point, the power might become insufficient to detect an effect with a high enough probability and the method will stop moving. This can be seen in figures 6.2 and 6.3. In general one can say that the design with the highest power will move closest towards the optimum and have a higher quality of

output. Thus the full factorial (high power) achieves a better end-result, in the presence of noise, than the fractional factorial.

Yet the quality of the output is not the only determining factor to take into account when choosing the design to use. Experimental effort plays an important role as well. In the case of this simulation study, the full factorial always achieves a higher final response value (in the presence of noise) than the fractional factorial. For the case of SNR = 1000, the rate of improvement (in total number of measurements) in figures 6.2a and 6.3b is almost equal and is equated with the slope of the improvement in response. However, the *rate of improvement* can only be measured at the conclusion of every phase and if one takes this into account it is not always feasible to wait for a long period of time (design with high sample size, full factorial) for a decision on the move. The main reasons why this is not desired can be summarized by (1) a constraint on the number of experiments (e.g. production might be limited in time or sampling the process might be slow which puts a constraint on the number of measurements one can use), and (2) possibility of non-stationarity of the process (e.g. when measurement time is longer than a process drift in time the improvement method will fail).

These measures of how well the improvement scheme performed have to be taken into account when deciding on an appropriate design to use.

Furthermore, experimental time can be equated to cost (e.g. the cost of running the machine for the duration of the improvement process or the cost of a machine supervisor during the improvement process). This cost cannot be uniquely quantified as every process supervisor or production facility might have different requirements, equipment and personnel. Relating the number of measurements to a cost function will help set limits to the maximum measurements allowed in one phase or during the improvement if it is only run for a limited time. Such a cost function can, for example, be comprised of: the time to acquire one sample, operating cost, cost of personnel, profit of the produced product, cost of supplies, etcetera. Such a cost function is a good way to visualize the various demands the improvement method should take into account.

6.4 Conclusions

The traditional base design of EVOP, the two-level full factorial, uses 2^k measurements per phase which becomes more and more prohibitive in higher dimensions. In this chapter the two-level fractional factorial design is investigated as an alternative base design for EVOP. It is shown that even in

high dimensions a fractional factorial base design is often preferable since a lot of improvement is still possible using the reduced sample size.

In classical design the statistical power of a design is often $\pi \geqslant 0.80$. In EVOP, however, it is shown that even with a very low power a large amount of improvement is still possible. The simulations prove that EVOP has the capacity to correct its improvement path after every phase, whereas in classical experimentation only one "phase" is performed.

The observation that the statistical power can be lower compared to classical design leads to the question what the optimal power for the EVOP methodology is. This warrants further investigation and is dealt with in chapter 7.

To conclude, more efficient designs offer a significant improvement in total number of measurements to the optimum. Their improvement rate might be slower after every phase than for the full factorial base design but, since the sample size of the efficient designs is lower, faster decisions (in time) can be taken. This is an advantage in non-stationary processes in which a time-drift is present. One does have to take into account that, in general, the efficient designs have a tendency to stop farther away from the optimum if the noise-level increases as the statistical power will become too low to detect the very small effects. The statistical significance of an effect should always be balanced with the practical relevance, i.e. very small shifts in the response might become statistically significant when increasing the sample size (power) but might not be relevant for the experimenter (e.g. detecting an increase of 0.01% in the response is not worth the experimental effort) .

7 Optimal Statistical Power For Evolutionary Operation

7.1 Introduction

In the previous chapter efficient designs were introduced as a solution to reduce the sample size in each phase. In contemporary processes were the possibility exists that many settings have a significant influence, this is of the utmost importance to allow for a feasible improvement in an acceptable time. However, it is generally known that the power to detect an effect is often quite low when using such sparse designs, and the concept of power was used extensively to explain the observations that were made. One of the striking results of the previous chapter was that the power does not need to be as high as classically assumed for experimental design (where values of 0.7–0.9 are common). Indeed, even with powers as low as 0.2 good results were obtained. Even more surprising, the improvement using a low power was often faster than when a high power was chosen.

This leads to the question what power to use for EVOP schemes. To the author's best knowledge, research into the required power for EVOP is completely lacking, and this chapter provides first results and recommendations in that direction. In order to focus on the power, the effect of changing the factorstep and the SNR will be excluded here for obvious reasons. The effect of the power on the improvement will be studied by changing the sample size of the design, and this will be done for different dimensions of the problem. In order to make results interpretable, a simple linear function will be used throughout the simulation study.

7.2 Materials & Methods

7.2.1 Underlying Model

In the previous chapters a quadratic model was used for running the simulations since it is reasonable to assume that such a model approximates the region around the optimum well. However, such a model complicates research into optimal power as the slope of the function and thus, the coded linear effect, changes when one moves closer to the optimum. To exclude the changes in power due to the changing slope of the underlying function, a simple linear model (equation 7.1) will be used, where k is the dimensionality of the problem.

$$y = \sum_{d=1}^{k} 0.5 \cdot x_d + \varepsilon \tag{7.1}$$

The noise $\varepsilon \sim N\left(0, \sigma^2\right)$ in this equation is assumed independent and identically distributed (i.i.d.) according to a normal distribution with mean zero and standard deviation $\sigma = 1$.

7.2.2 Simulation Settings

The goal of the simulation study is to determine whether there is an optimal statistical power in an EVOP improvement scheme. This will be achieved by studying two settings: (1) the power π_{req} and (2) the dimensionality k of the problem. The performance characteristics used to indicate how good the EVOP improvement performs are the median total number of measurements, median (N), required to reach the optimal region (as defined later) of the simulation and the median number of phases to reach the optimal region, median (N/n_T), with n_T the number of measurements per phase.

Dimensionality: The dimensionality k of the problem was varied from 2 to 16 to investigate its influence on the optimal power.

Statistical power: The statistical power will affect the sample size (as explained in the next section) and is changed from 0.05 to 0.99 using a linearly spaced vector of 21 levels.

Random walk: To test whether the improvement performs better than just randomly moving through the experimental domain additional simulations were performed where the model terms were removed from the simulation model,

keeping only the noise term. To significantly reduce computation time, these simulations were stopped when the number of measurements reaches $67,700$ (or the nearest sample size that is larger than this number).

Number of repetitions: In order to have a clear view on the performance indicators the simulation was performed 100 times for each combination of power and dimensionality and the median value of the total number of measurements is reported. This leads to a number of simulations runs of:

$$22 \text{ (power-levels + random walk)} \times 15 \text{ (dimensions)}$$
$$\times 100 \text{ (repetitions)} = 33,000$$

7.2.3 Evolutionary Operation Implementation

Evolutionary Operation Settings

To make the interpretation of the results more clear, the simulation settings and definition of the step size in this chapter are simplified when compared to the previous chapters. All this information is represented in figure 7.1 where examples of EVOP improvements are visualized for $k = 2$ for several different powers. The following paragraphs will refer to the relevant information on this figure. Since the sample size will change, depending on the requested power, $\mathcal{D}$-optimal designs will be used as the base designs in this chapter (see next section).

Starting point: The starting point $\mathbf{x_{start}}$, which is the center of the base design as was the case in previous chapters, was fixed at $\{0, \ldots, 0\}$ for all simulations.

Factorstep: The factorstep was fixed for all dimensions at $dx = 1$ as shown by the EVOP phases depicted in figure 7.1.

Step size & direction of the move: The step size δ_{EVOP} is defined equivalently to the previous chapters, and is given by equation 7.2. Contrary to the previous chapter the direction of the move will be restricted here in every dimension to only three options: $+1$ if a regression coefficient is significant and positive (correct move in this direction), 0 if the regression coefficient is not significant and -1 if a regression coefficient is significant and negative (incorrect move in this direction).

$$\delta_{EVOP} = dx \cdot \sqrt{f_a} \tag{7.2}$$

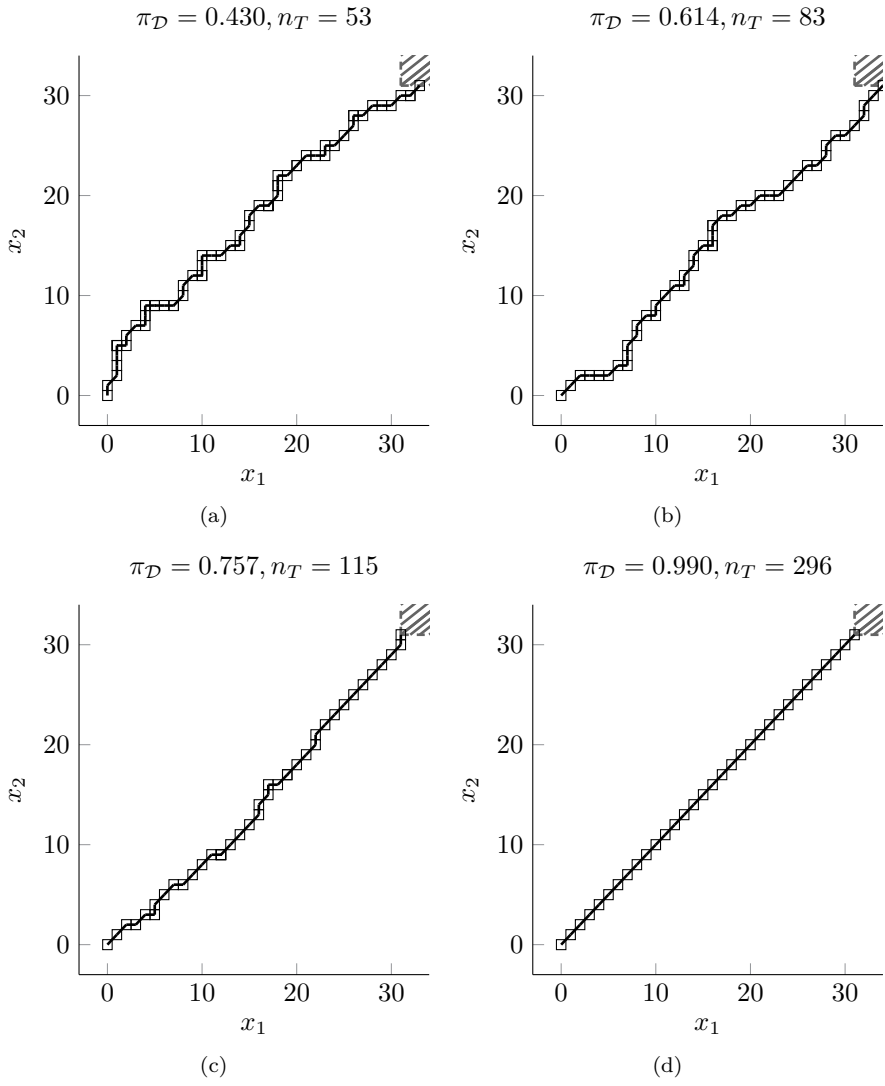


Figure 7.1: Representation of the experimental domain and an example of one improvement run (moving design region) until the optimal region is reached for $k = 2$ and different powers: (a) $\pi_{\mathcal{D}} = 0.430$ and $n_T = 53$, (b) $\pi_{\mathcal{D}} = 0.614$ and $n_T = 83$, (c) $\pi_{\mathcal{D}} = 0.757$ and $n_T = 115$, (d) $\pi_{\mathcal{D}} = 0.990$ and $n_T = 296$.

Region of the optimum: There are several options on how to choose the optimal region. Contrary to previous chapters, the optimal region is not defined as a percentage of some desirable (maximum) response value. Since the simulation model is linear in the parameters, the response value will keep increasing if the factor levels are increased. An optimal region could be determined by defining a noise-free response that needs to be reached. However, as the dimensionality increases, the possible factor level combinations to reach such a response would also drastically increase. Since all dimensions are of equal importance in the simulation model in equation 7.1, it was opted to fix the optimal region a number of steps in every dimension from the starting point. It is now defined by the region where $\forall x_i : x_i \geqslant 30$. This definition of the optimal region is exemplified by the shaded areas in figure 7.1. Defined this way, the minimal number of phases to reach the region of the optimum is 31, irrespective of the dimensionality.

Number of phases to optimal region: the number of phases to the optimal region is calculated by dividing the number of measurements N required to reach the region of the optimum by the sample size n_T per phase.

Boundaries of the experimental domain: No boundaries are imposed on the movement of the design region. Since the regression coefficients are the same, irrespective of the location of the design region, constraining it to a bounded experimental domain does not make sense from the perspective of this simulation.

Statistical Power And Design Selection

As explained in the previous chapter, for an orthogonal two-level design in coded units (the low factor level transformed to -1 and the high factor level to $+1$), the power π of the statistical significance test is given by equation 7.3.

$$\begin{aligned}\pi =& P\left\{-t\left(1-{}^{\alpha}/_{2}; n_T-1-f_t\right) < T_{NC}\right\} \\ &+ P\left\{t\left(1-{}^{\alpha}/_{2}; n_T-1-f_t\right) > T_{NC}\right\}\end{aligned} \tag{7.3}$$

In this equation T_{NC} is the noncentral t-distribution with degrees of freedom $\nu = n_T - 1 - f_t$ and ϕ the noncentrality measure; t the t-distribution with degrees of freedom ν on which a two-sided test with significance level α is executed. For both distributions the degrees of freedom are defined by n_T the total number of measurements and f_t the number of parameters in the estimated model (excluding the intercept), which equals the number of active main effects $f_t = f_a$ when fitting a main effects model.

The noncentrality measure to detect a coded coefficient of size $\beta_{d,c}$ is then given in equation7.4 with σ the error standard deviation.

$$\phi = \frac{\beta_{d,c}}{\frac{\sigma}{\sqrt{n_T}}} = \sqrt{n_T}\frac{\beta_{d,c}}{\sigma} \tag{7.4}$$

The statistical power to detect an effect is directly related to the sample size n_T, the amount of noise in the system σ, the number of parameters in the model f_a and the size of the effect $\beta_{d,c}$ (size of the coded linear regression coefficient) one wishes to detect.

The noise σ in the system is constant throughout the simulations, all main effects are nonzero in the true model and $\beta_{d,c} = 0.25$, fixed at the coded slope of the linear simulation model (equation 7.1) in the design region. This ensures that the only parameter in equations 7.3 and 7.4 that can affect the power is the sample size n_T as all others are fixed for the given significance level $\alpha = 0.05$. The Matlab® program in appendix C can be used to recursively search for the sample size that matches the requested power most closely.

The sample size that gives a resulting power closest to the requested value can be any number equal to or greater then $k + 2$ samples. The sample size is at minimum $k + 2$ since saturated designs are not allowed as no classical statistical tests can be executed. This leads to a minimum degrees of freedom to estimate the k main effects, the intercept and at least one degree of freedom for error to estimate the noise.

Requiring fixed set points for the power poses a challenge for design selection as the classical designs used up until now will not allow to build designs for arbitrary sample sizes. The use of optimal designs which can generate near-orthogonal designs for an arbitrary sample size will be used to allow for the necessary flexibility in sample size. Since in EVOP accurate parameter estimates are important as the direction of improvement is directly derived from these coefficients, the $\mathcal{D}$-optimality criterion was chosen for the generation of the optimal designs. This criterion minimizes the generalized variance of the parameter estimates for a pre-specified model which, in this case, is a linear model of the dimensionality required by the simulation. In other words, the algorithm tries to maximize the $\mathcal{D}$-efficiency of the design, which is expressed for this specific case by equation 7.5. More information about the construction of $\mathcal{D}$-optimal designs can be found in section 3.2.3.

$$\mathcal{D}\text{-efficiency} = 100 \cdot \frac{|\mathbf{X}'\mathbf{X}|^{\frac{1}{k+1}}}{n_T} \tag{7.5}$$

One $\mathcal{D}$-optimal design is constructed for each combination of requested power and dimensionality and this generated design is used for every phase and for every repetition to exclude correlating the results with a change in the design matrix. Such a change could happen since the possibility exists that different runs of the construction algorithm lead to different designs. It is important to remember that the designs are computer-generated and not every generated design will have the same $\mathcal{D}$-efficiency. To ensure that a design with a high $\mathcal{D}$-efficiency is selected, the *cordexch* algorithm in Matlab® is used and set to perform 100 iterations to generate a design (default is 10). Furthermore, 100 designs are generated and the design with the highest $\mathcal{D}$-efficiency is selected.

It is important to note that the use of equation 7.3 is valid as long as a two-level design is used. Since the pre-specified model structure is linear, this requirement is met and one does not need to generalize the power equations. However, equation 7.4 for the noncentrality parameter only holds for coded orthogonal designs. In general, the non-centrality parameter for coded designs can be written as equation 7.6, with $\sigma_{\beta_{d,c}}$ the estimated parameter variance.

$$\phi = \frac{\beta_{d,c}}{\sigma_{\beta_{d,c}}} \tag{7.6}$$

The construction of a $\mathcal{D}$-optimal design does not necessarily lead to orthogonality. This implies that equation 7.4 for the noncentrality measure no longer holds. Therefore, the exact statistical power can only be calculated after design construction. However, $\mathcal{D}$-optimal designs do often lead to orthogonal or near-orthogonal designs and therefore equation 7.4, which defines the noncentrality measure for orthogonal designs, is an acceptable choice to determine the sample size and significantly reduces the computational time necessary to select a sample size.

The procedure for design generation can be summarized as follows:

1. Based on the required power π_{req}, calculate the sample size using equations 7.3 and 7.4 (implemented in the Matlab® program in appendix C);
2. Generate a $\mathcal{D}$-optimal design with the required sample size;
3. Check the actual power of the parameters of the $\mathcal{D}$-optimal design (using equation 7.6) after it is generated and select the smallest power $\pi_{\mathcal{D}}$.

Since only a limited amount of power levels are simulated (21 levels) and the interest is in investigating the effect of power on the total number of measurements and the number of phases for increasing dimensionality, two

models will be fitted on the grid of simulation data. Since it is expected that the underlying behaviour is highly non-linear, Gaussian Process (GP) modelling will be used, as described in chapter 4.

The JMP® software (version 11, The SAS Institute, Inc., Cary, NC, USA) will be used to estimate two GP models in which a nugget parameter is included since the simulation data is noisy. The independent variables of the model are: (1) the statistical power $\pi_{\mathcal{D}}$ for the used $\mathcal{D}$-optimal design and (2) the dimensionality k for which this simulation is executed. The dependent variables used for the models are the median[1] total number of measurements (of the 100 repetitions)—median (N)— and the median[2] number of phases—median (N/n_T)—to reach the optimal region for each combination of the independent variables. Using the GP model for the median number of measurements, the JMP® profiler is used to find the minimal median total number of measurements for each dimensionality, which corresponds to the optimal power for this model. Contrary to chapter 4—where the cubic correlation function was used—the gaussian correlation function is implemented since the responses are the median of the number of measurements and phases, a large amount of the noise will already be filtered out and therefore using the gaussian correlation—as represented in equation 7.7—is a good choice. The correlation between two points x_{ik} and x_{jk} is always non-zero, no matter the distance between the points (i.e. taking into account all information).

$$r_{ij} = \exp\left(-\sum_{k=1}^{n} \theta_k \left(x_{ik} - x_{jk}\right)^2\right) \tag{7.7}$$

7.2.4 Simulation Pseudo-Code

Algorithm 7.1 represents the pseudo-code of the software algorithm as implemented in Matlab®. It can be seen from this pseudo-code that the worst case scenario is simulated:

1. The data from previous stationary phases will be forgotten and not used to get a more accurate estimate of the noise;
2. Since the same design is used for every subsequent step, unbalanced designs will tend to favour a move in a specific direction (i.e. if $k = 2, n_T = 5$ one

[1]The full data of the simulations, i.e. the total number of measurements N to reach the optimal region for each combination of the independent variables could not be used since the amount of data was too large for the JMP® software. Therefore, median (N) was chosen as the dependent variable.

[2]The same remark as in footnote 1 holds for the number of phases N/n_T.

of the design points is replicated which will always be the same point in every subsequent phase).

It should also be noted that—for the very specific simulation settings here—another simulation approach is possible. One can conceptually view the estimation of the statistical model in every phase as a chance to "succeed" or "fail" to detect a move in any dimension. This can be seen, in essence, as a Bernouilli trial since the probability of success is the same every time the estimation is conducted.

Based on this observation, one can use negative binomial distributions (a discrete probability distribution of the number of successes in a sequence of independent and identically distributed Bernoulli trials) to construct a much faster, analytical simulation. The reader is referred to [59] for an introduction on the negative binomial distribution and to [60] for a theoretical explanation about the mean of the maximum of N random variables distributed by the negative binomial distribution. Since this chapter aims at laying the foundations for this type of simulation studies in general, the approximate approach as described above will be used. It was verified for certain combinations of power and dimensionality that the approximate solution does approach the analytical solution quite well for the specific settings used in this chapter.

7.3 Results & Discussion

7.3.1 Statistical Power And Sample Size

The actual power $\pi_{\mathcal{D}}$ is always close to the requested power π_{req}. To illustrate this, table 7.1 presents the powers and sample sizes for $k = 2$ and $k = 14$. This table presents the requested power π_{req}, the power π_{orth} for the closest orthogonal design (as given by the Matlab® program in appendix C), the power $\pi_{\mathcal{D}}$ (as calculated based on the generated $\mathcal{D}$-optimal design), the $\mathcal{D}$-efficiency of the selected design and the sample size per phase n_T.

7.3.2 Results Of The Random Walk

In none of the power × dimensionality combinations the optimal region was reached within the maximum number of measurements allowed (67, 700). This maximum number of measurements is more than double the highest number of measurements ever encountered in any other simulation (33, 839), which leads

Algorithm 7.1. Optimal Power Simulations Pseudocode.

for dimension from 2 to 16 **do**
  for every power π_{req} **do**
    Calculate sample size n_T
    Generate $\mathcal{D}$-optimal design $\mathbf{X}$ with $dx = 1$, $\mathbf{x_{start}} = \{0, \ldots, 0\}$
    Calculate power $\pi_{\mathcal{D}}$
    for repetition from 1 to 100 **do**
      $count = 0$
      $\boldsymbol{\Delta} = 0$
      while any element of $\boldsymbol{\Delta} \leqslant 30$ **do**
        Simulate response for design $\mathbf{X}$
        Execute stepwise regression
        Move design region by $\boldsymbol{\delta}_{\mathbf{EVOP}}$, calculated in every dim. as:
        - +1 for positive sign. coefficient;
        - -1 for negative sign. coefficient;
        - 0 for non-sign. coefficient.

        $\boldsymbol{\Delta} + \boldsymbol{\delta}_{\mathbf{EVOP}}$
        $count = count + 1$
      end while
      Save total number of measurements $N = count \cdot n_T$
    end for
  end for
end for

to the conclusion that for all powers, even the lowest value of only 0.05, EVOP outperformed a simple random walk in the experimental domain.

7.3.3 Total Experimental Effort And Path Efficiency

The Gaussian Process models that were fit to the simulation data are visualized in figure 7.2. Using the JMP® profiler function the minimum median total number of measurements as fitted by the GP model was determined for each dimension. These minima are plotted as dots for each dimensionality k on figure 7.2a.

When considering the *total experimental effort* (measurements) for reaching the region of the optimum, it can be seen from figure 7.2a that when the power is very low ($\pi_{\mathcal{D}} < 0.25$) and the dimension is high ($k > 10$) the number of measurements required to reach the optimal region increases drastically.

Table 7.1: Requested power π_{req}, power for the orthogonal design π_{orth} generated by the sample size program, power for the $\mathcal{D}$-optimal design $\pi_{\mathcal{D}}$ calculated after design generation, $\mathcal{D}$-efficiency and sample size n_T per phase for $k = 2, 14$.

$k = 2$					$k = 14$				
π_{req}	π_{orth}	$\pi_{\mathcal{D}}$	$\mathcal{D}$-eff.	n_T	π_{req}	π_{orth}	$\pi_{\mathcal{D}}$	$\mathcal{D}$-eff.	n_T
0.050	0.055	0.054	96.4	5	0.050	0.068	0.068	> 99.9	16
0.097	0.102	0.102	98.6	10	0.097	0.116	0.112	97.0	18
0.144	0.151	0.151	> 99.9	16	0.144	0.148	0.139	97.7	20
0.191	0.199	0.197	99.7	22	0.191	0.194	0.181	98.2	24
0.238	0.238	0.238	99.9	27	0.238	0.240	0.234	98.9	29
0.285	0.292	0.292	99.9	34	0.285	0.291	0.283	99.0	35
0.332	0.337	0.337	> 99.9	40	0.332	0.338	0.333	99.4	41
0.379	0.381	0.381	99.9	46	0.379	0.383	0.379	99.5	47
0.426	0.431	0.430	> 99.9	53	0.426	0.426	0.423	99.7	53
0.473	0.478	0.478	> 99.9	60	0.473	0.474	0.469	99.8	60
0.520	0.522	0.522	> 99.9	67	0.520	0.526	0.521	99.8	68
0.567	0.570	0.570	> 99.9	75	0.567	0.568	0.565	99.8	75
0.614	0.614	0.614	> 99.9	83	0.614	0.618	0.616	99.9	84
0.661	0.665	0.665	> 99.9	93	0.661	0.663	0.662	99.9	93
0.708	0.710	0.710	> 99.9	103	0.708	0.709	0.707	99.9	103
0.755	0.757	0.757	> 99.9	115	0.755	0.757	0.756	99.9	115
0.802	0.804	0.804	> 99.9	129	0.802	0.804	0.803	> 99.9	129
0.849	0.851	0.851	> 99.9	146	0.849	0.850	0.850	99.9	146
0.896	0.896	0.896	> 99.9	168	0.896	0.896	0.896	> 99.9	168
0.943	0.943	0.943	> 99.9	203	0.943	0.943	0.943	> 99.9	203
0.990	0.990	0.990	> 99.9	296	0.990	0.990	0.990	> 99.9	296

However, when the dimensionality is low, a low power still results in reaching the region of the optimum the fastest. Another observation is the fact that for all dimensions, there is a quite flat region in between the very low and the very high power values. In other words, the exact choice of the power is not so critical but the optimal power shifts towards a higher value when the dimensionality increases. This is logical as the probability to make a wrong move or no move is present in every dimension. The higher the number of dimensions, the higher the number of coefficients that can be incorrectly estimated.

Besides the total number of measurements required, also the *efficiency*—expressed as the number of phases that is required—can be considered. The median number of phases to the optimal region for an increasing power (increasing sample size per phase) for three selected dimensions ($k = 2, 8, 14$) are plotted in figure

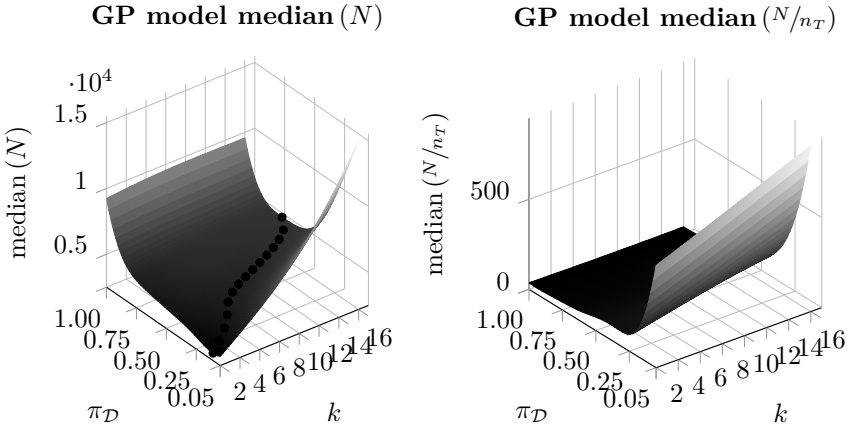


Figure 7.2: Gaussian Process models for the power simulations: (a) Model for the median total number of measurements, (b) model for the median total number of phases.

7.3. One can clearly see from this figure that the higher the power, the higher the efficiency (lower median number of phases). Due to the way the optimal region is defined, the most efficient improvement path from start to optimal region always counts the same number of phases—namely 31—regardless of the dimensionality. Figure 7.3 also shows that the efficiencies for the three dimensions are rather close to each other when, say, $\pi_{\mathcal{D}} \geqslant 0.50$. This observation is not surprising when inspecting table 7.1, from which can be derived that the number of measurements per phase for $k = 2$ and 14 are always identical from a power of 0.661 onwards. A similar observation can then also be made from figure 7.2a where the total experimental effort for low and high dimensions are quasi identical at the highest power levels. The fact that the values are not completely identical results from the fact that the power denotes the probability of correctly detecting one main effect. When dimensionality increases for a given power, there is thus a probability in each dimension that the effect is overlooked, explaining why the experimental effort in the higher dimensions is higher than in lower dimensions, even if the sample size per phase is identical.

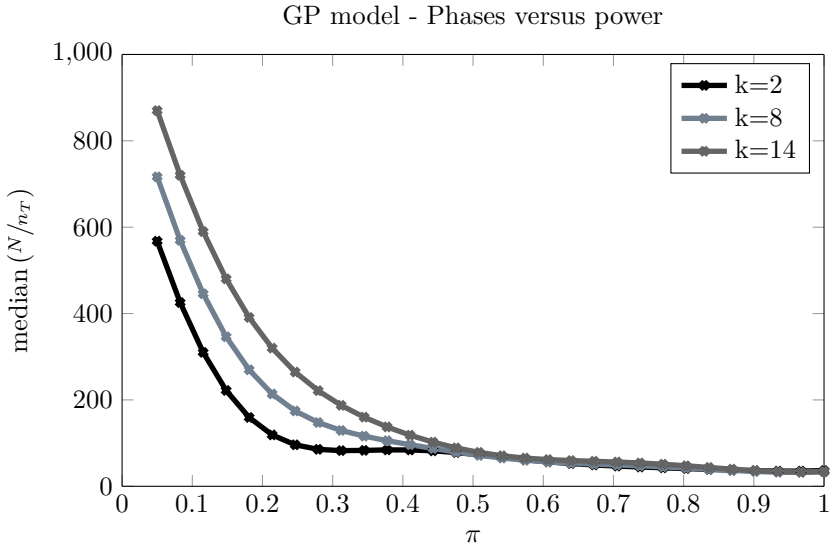


Figure 7.3: GP model - Phases versus power for $k = 2, 8, 14$.

7.3.4 Optimal Power

As mentioned above and visualised in figure 7.2a, the number of measurements that is required to reach the region of the optimum greatly depends on the dimensionality of the problem and the considered power. When visualizing the power at which the minimal experimental effort is achieved for each dimension—see figure 7.4—it can be observed that for low dimensions this optimal power is surprisingly low, namely the lowest power considered in this study when $k = 2$. The reason here is mainly the fact that the sample size for each phase increases drastically for low dimensions when the power is increased. Indeed, for $k = 2$ the sample size doubles when increasing the power from ~ 0.05 to 0.1, while for $k = 14$ the sample size increases with only 12.5%. In other words, the modest increase in power comes at a great cost (experimental effort) in low dimensions, favouring low power. When moving into higher dimensions, figure 7.4, complemented by table 7.1, shows that the gain in path efficiency does weigh more than the increase in sample size for each phase, but how much it weighs depends on dimensionality. For the highest dimension considered, $k = 16$, the optimal power is about 0.7, and for all other dimensions this optimum comes lower. The way the optimal power depends on dimension is not straightforward. It can be clearly seen that there are two important regions: for $k < 8$ a low power is preferable, for $k \geqslant 8$ there is a flat, broad valley with

almost equal experimental effort for a broad range of powers ($\pi \in [0.4; 0.8]$), which will allow considerable freedom in choosing a power (see next section).

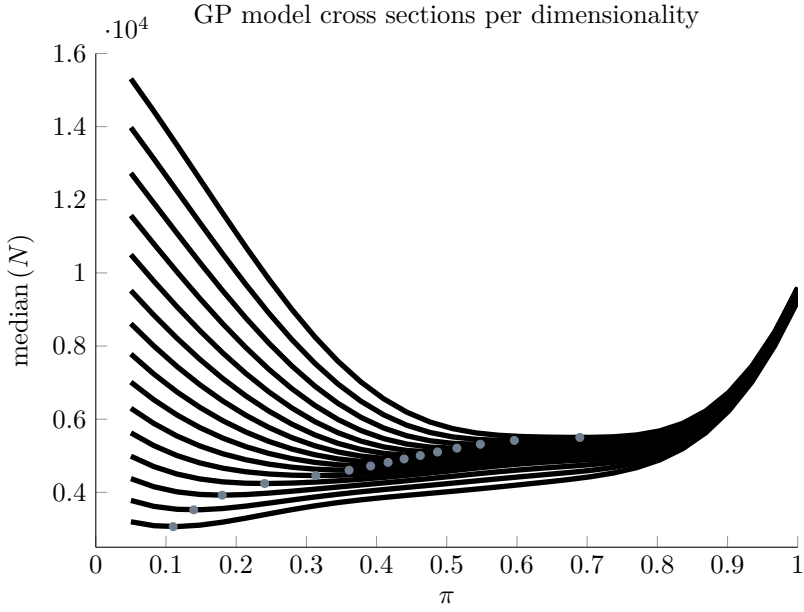


Figure 7.4: Cross sections of GP model for optimal power, bottom line is $k = 2$, top $k = 16$.

7.3.5 Practical recommendations

As discussed above, for low dimensions choosing a base design with a low power results in the least experimental effort that is required to reach the region of the optimum. As for practical recommendations, it is thus recommended to use very sparse designs where the power is lower than 0.4. Although the efficiency of the path might not be ideal in those cases, it is clearly the fastest way because of the great experimental cost that comes with an increase in power.

The situation is somewhat more open for the larger dimensions, since the experimental effort for a broad range of power values is nearly equal. In those cases, the recommendation strongly depends on the type of process and how it is sampled. For processes where each measurement takes a substantial effort, it is advised to use a low power, and, thus, a low sample size for each phase, so that eventual drifting of the process behaviour can be well tracked. For processes

where one can sample (experiment, measure) very fast at low cost, this issue does not play an important role, and choosing a higher power so that a more efficient path is followed is advised. The reason for this is mainly because that way each phase will have a larger probability of producing improved output.

7.4 Conclusions

In this chapter a first framework for investigating the optimal statistical power in EVOP improvements was presented. The optimal statistical power shows up to be a function of the dimensionality and increases when the dimensionality increases.

The experimental effort required to reach the region of the optimum shows a broad, almost flat, valley for high dimensionalities ($k \geqslant 8$). So it is possible to choose any power in this broad range ($\pi \in [0.4; 0.8]$) without affecting the effort to a large extent. The choice which power—and, hence, which sample size for one phase—to choose is then completely determined by the type of process under study; for processes with a low sampling rate or non-stationary process prone to time drift a low power is recommended, whereas a higher power is advised when a high sample rate is possible, or when the process is stationary.

It is stressed here that these observations are first results, and that more data should be gathered for different simulation cases to confirm these numbers.

Part III

Case Study

8 Constrained Online Improvement Using Evolutionary Operation Steepest Ascent: A Case Study About Energy-Optimal Robot Control

8.1 Introduction

In modern processes the output is often of a multi-objective nature and this work would not be complete without investigating such processes. This chapter presents a way to deal with multi-objective or constrained improvement problems by using EVOPSA.

The choice of EVOPSA was motivated by the following constraints: the measurement time for improvements is limited and thus the total sample size has to be kept to a strict minimum. This leaves the Simplex method, or the proposed EVOPSA method which combines the best of EVOP and Simplex. Initial tests, which are not included in this chapter since they were executed using another measurements system and are not comparable with the results of the measurement system used for the improvements, indicated that of these two methods, EVOPSA performed best. It was therefore chosen to execute this study using EVOPSA.

The goal of this chapter is to show the switch from offline optimization to a completely automated online constrained improvement method using a case

The results section was published in:
K. Rutten et al. "Constrained On-line Optimization Using Evolutionary Operation: A Case Study About Energy-Optimal Robot Control". In: *Quality and Reliability Engineering International* (2014). DOI: 10.1002/qre.1662

study about a badminton robot. The starting point of the research was the availability of the optimal region which was determined offline using Genetic Algorithms on a computer simulation model of the robot [146]. Based on these results, follow-up online experimentation was executed to fine-tune the settings. The additional challenge of the constrained nature of the problem was dealt with by handling it as a multi-objective improvement. First the badminton robot case study will be described in detail, then the settings for EVOPSA are documented. After this a benchmark test is described in order to evaluate the performance of the methods which is followed by the results of the study, which are concluded in the final section.

8.2 Case Study: The Badminton Robot

Typical requirements of industrial motion controllers are to realize fast and accurate motions in order to optimize the system's productivity [146], the so-called time-optimal motion control. The raising energy prices and growing environmental awareness have initiated a shift in these requirements to find a trade-off between time-optimal and energy-optimal motion control.

In order to investigate the potential of time and energy-optimal controllers, a case study was implemented at the Flanders Mechatronics & Technology Centre (FMTC). It deals with the optimization of a badminton robot: the robot should consume the least amount of energy possible, subject to the constraint that it arrives on time to intercept a shuttle. The robot, of which a schematic representation is given in figure 8.1, has three degrees of freedom: a rotationary axis to rotate the racket, a linear axis to move the badminton robot and a hit axis to let the racket hit the shuttle. More information about the design of the badminton robot and its subsystems can be found in the relevant articles [145, 156]. The improvements executed in this study will focus on the main energy consumer of the badminton robot, namely the linear axis.

The initial motion controller for the linear axis was a Proximate Time-Optimal Servo (PTOS) controller [158], as described in detail for the badminton robot by Stoev *et al.* [145]. The control scheme can be described as follows: a camera is used in combination with an interception algorithm to calculate the distance to move y_{des} (interception reference) and the time t_{max} that is available to move in position to be able to hit shuttle before it is out of reach. In the classical PTOS approach, the interception reference is fed to the PTOS controller which has two tuning parameters, acceleration a and velocity v, both set at their maximum values to ensure time-optimal motion. The PTOS controller generates the reference signals necessary for linear movement (trajectory generator).

(a) (b)

Figure 8.1: Schematics of the badminton robot: (a) a representation of the complete setup, (b) detail of the robot itself, images courtesy of Flanders Mechatronics & Technology Centre, Belgium

Energy-optimal motions can be achieved by taking the interception time parameter t_{max} into the control strategy. Not every move should be executed as fast as possible, since slower movement modes might consume less energy. By making the control parameters dependent on interception reference *and* interception time, the amount of consumed energy during badminton play might be reduced. To achieve this, a simulation model of the badminton robot was constructed and a Pareto front was built using a genetic algorithm, which permits to construct the functional maps (based on a regression function, see [146, 156]) presented in equation 8.1 and 8.2.

$$(t_{max}, y_{des}) \rightarrow \hat{a} \tag{8.1}$$

$$(t_{max}, y_{des}) \rightarrow \hat{v} \tag{8.2}$$

These maps were used to create the Proximate Energy-Optimal (PEOS) $\hat{a}$ and $\hat{v}$ lookup tables which are a subset from these functional maps. This subset was selected in such a way that the controller parameters $\hat{a}$ and $\hat{v}$ can be calculated by linear interpolation from these lookup tables for a specific motion y_{des} within time interval t_{max} if this combination (t_{max}, y_{des}) is not present in the lookup tables. An advantage of this approach is that the implemented controller does not need to be changed, but only the controller settings $\hat{a}$ and $\hat{v}$ have to be adjusted. Further details of this procedure are published by Stoev *et al.* [146] and Wang *et al.* [156].

A drastic decrease in energy was reported during benchmark testing while the overall capability of the badminton robot to intercept the shuttle, the precision of

the robot, lowered slightly. Another energy-optimal approach was implemented in Wang *et al.* [156], where a model predictive control (MPC) approach, based on time-optimal MPC (TOMPC) [153], was used and this approach was labelled Energy-Optimal MPC, EOMPC. The EOMPC approach outperformed the PEOS approach both in energy as well as in precision for the badminton case. However, this approach faces its challenges during implementation: (1) it requires more performant hardware than PEOS, and (2) it relies on the availability of a cost function, in this case a quadratic function describing the energy consumption. These type of cost functions are often very hard to obtain. In addition, (3) the presence of some model-plant mismatch may even further jeopardize the applicability of the method.

In current practice at the FMTC, the PEOS lookup tables generated in simulation are implemented on the real system. Online improvement techniques will be used to try and improve the PEOS lookup tables to further decrease energy consumption. The proposed online improvement is simple and easy to implement and does not require the availability of an accurate parametric model of the full process.

8.3 Online Improvement

There are two performance criteria that need to be improved: (1) the consumed energy and (2) the precision, defined here as the percentage of hits intercepted during benchmark testing. These performance criteria will be compared in benchmark tests after improvement. As a reference, the PTOS approach will also be measured as this time-optimal motion results in the highest consumed energy and precision. For the PEOS case, the current standard, two functional maps are present: one for velocity $\hat{v}$, one for acceleration $\hat{a}$. These lookup tables were generated during offline optimization and already offer a significant improvement in energy compared to the PTOS. Each of the (t_{max}, y_{des}) pairs in these lookup tables will be used as the starting point of an online improvement. Thus, as much improvements as there are pairs in the lookup table are executed. In order to protect the system against eventual warming-up or other systematic but uncontrollable disturbances, a pair was selected from the lookup tables at random after which all improvements are performed for this coordinate set. The settings for the EVOPSA algorithm are explained below.

8.3.1 Settings For Evolutionary Operation Steepest Ascent

Evolutionary Operation Steepest Ascent (EVOPSA) is, as explained previously, a procedure where statistical estimation of a steepest ascent direction is combined with a line search. Instead of calculating a new design every phase —a phase being every set of measurements necessary to complete before moving step size δ in the suspected steepest ascent direction— as would be done in EVOP, a steepest ascent direction is estimated after which a line search in this direction is executed. When the last response of the line search becomes worse than the previous response, a new statistical design is executed around the previous response and a new steepest ascent path is estimated.

Usage of the EVOPSA algorithm requires following settings, of which most are visualized in figure 8.2.

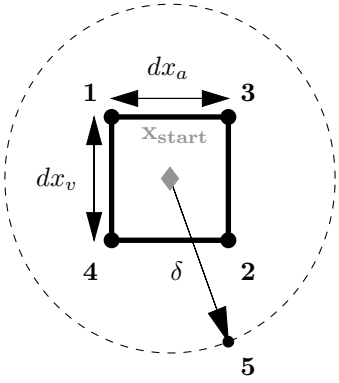


Figure 8.2: Visualization of first and second EVOPSA phase.

Starting point $\mathbf{x_{start}}$: this is the point around which the primary measurements (points 1–4 in the figure) will be spread. In the case of the badminton robot, there are two factors in the improvements, which are the acceleration $\hat{a}$ and velocity $\hat{v}$ of the robot.

Factorstep dx_d: the factorstep represents the distance in every factor between the measurement points when using a full factorial design. In case of the badminton robot, there are two inputs: acceleration and velocity. Subsequently there are two factorsteps: one for acceleration, chosen as $dx_a = 0.3\,\mathrm{m\,s^{-2}}$ and one for velocity set at $dx_v = 0.03\,\mathrm{m\,s^{-1}}$. Determining appropriate values for the factorsteps is not straightforward. Usually experience with the process offers values for the factorsteps.

Stepsize δ: The step size between two consecutive phases is calculated according

to the rules set forth in chapter 3. When both factors are active, this results in a distance $\delta = \sqrt{dx_a^2 + dx_v^2}$.

Boundaries of the experimental domain: Both acceleration and velocity have minimum and maximum constraints. If during improvement such a boundary is met, the setting is restricted to its nearest limit. The experimental range stretches from $0.01\,\mathrm{m\,s^{-2}}$ to $30\,\mathrm{m\,s^{-2}}$ for acceleration, and from $0.01\,\mathrm{m\,s^{-1}}$ to $3\,\mathrm{m\,s^{-1}}$ for velocity.

Number of measurements: the number of measurements for each complete improvement run was limited to 40. This number was chosen based on practical considerations The number of measurements was calculated in such a way that all improvements could be executed in the timespan the setup was available.

8.3.2 Constrained Nature Of The Problem

The problem deals with the minimization of energy E subject to an arrival time constraint t_{max} (for a given interception distance y_{des}). This can be considered as the one-parameter improvement of E subject to the time constraint t_{max}. In this research, another approach was adopted: the problem was treated as a multi-objective problem using desirability functions by incorporating the interception time t_{max} and the energy E as the two responses to be improved (see further). The desirability functions as proposed by Derringer & Suich [45, 46] are used extensively in statistical multi-response optimization as evidenced by a sample out of the body of recent literature [35, 51, 76, 119, 141] and were previously proposed to be used within the framework of EVOP [25]. This motivated the choice to use desirability functions for this study.

With desirability functions, one wishes to attain the best balance among several different response variables or objectives. During this approach, every response is transformed into a desirability function d_d having a range between zero and one, zero being undesirable and one being the most desirable outcome for this response. All these separate desirabilities are then combined into the global desirability index D using the geometric mean as presented in equation 8.3.

$$D = (d_1 \cdot d_2 \cdot \ldots \cdot d_d)^{1/d} \tag{8.3}$$

This implementation of the geometric mean ensures that, when one response is undesirable (zero), the desirability index will also be zero. The badminton problem can be considered as a dual-response system: Energy E, which needs to be minimized, and interception time t_{max} which cannot surpass a certain

value. The interception time t_{max} can be treated as follows: it does not really matter in which timespan the move is executed as long as it is below or equal to the interception time t_{max} and the energy is minimized.

From the previous benchmark results for the PTOS and PEOS case [146, 156], one can surmise that there is a definite trade-off between the time to travel a distance, the energy consumed and the precision of the robot. To investigate this trade-off, five desirability transformations for time were implemented.

These transformation functions for time $d_{t,c}$ (with $c = 0, 1, 2, 3, 4$) and energy d_E are represented schematically in figure 8.3.

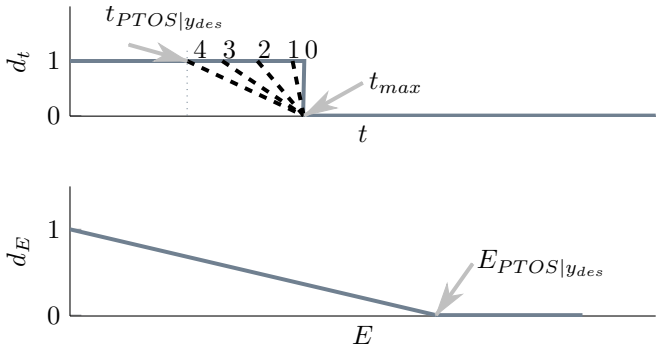


Figure 8.3: Desirability transforms for time and energy.

Mathematically, these transformations can be written as equations 8.4 and 8.5.

$$d_{t,c} = \begin{cases} 0, & t \geq t_{max} \\ {}^{(t_c - t)}/_{(t_{max} - t_c)}, & t_c < t < t_{max} \\ 1, & t \leq t_c \end{cases} \tag{8.4}$$

$$d_E = \begin{cases} 0, & E \geq E_{PTOS|y_{des}} \\ 1 - {}^{E}/_{\left(E_{PTOS|y_{des}}\right)}, & 0 < E < E_{PTOS|y_{des}} \\ 1, & E = 0 \end{cases} \tag{8.5}$$

With t and E the time and energy to transform, t_{max} the interception time, $E_{PTOS|y_{des}}$ the consumed energy in PTOS operation for the current interception reference y_{des} and $t_c < t_{max}$. The value for t_c changes depending on the implemented time-constraint with $c = 0, 1, 2, 3, 4$ and can be written as equations 8.6 through 8.10.

$$t_0 = t_{max} - 0.005 \tag{8.6}$$

$$t_1 = t_{max} - 0.05 \tag{8.7}$$

$$t_2 = t_1 - \frac{1}{3} \cdot (t_1 - t_4) \tag{8.8}$$

$$t_3 = t_4 + \frac{1}{3} \cdot (t_1 - t_4) \tag{8.9}$$

$$t_4 = t_{PTOS|y_{des}} \tag{8.10}$$

Let $t_{PTOS|y_{des}}$ be the smallest time possible to move a certain distance y_{des} (under PTOS operation)—being the time-optimal motion—and t_{max} the on-time constraint (the movement has to be executed within this timespan, the on-time constraints are taken from the currently implemented lookup tables). The difference between these two times is divided in five parts, numbered zero to four ($t_{PTOS|y_{des}}$). This leads to five full improvement schemes of the lookup tables and subsequently five new lookup tables from which to choose. The decision to differ t_0 slightly from t_{max} was made to assure that the desirability gradually drops to zero when approaching the time constraint. This ensures that the method still has a strong estimation of directionality if all measurements in a phase would be located around the time constraint t_{max}. Constraint t_1 was chosen a fixed value from t_0 to investigate how strong the effect is of changing all time-constraints by a fixed time-interval. The other time-intervals are dependent upon the values of t_{max} and $t_{PTOS|y_{des}}$.

The global desirability index can be written as equation 8.11, a function of the used desirability transforms for time and energy with $c = 0, \ldots, 4$.

$$D_c = \sqrt{d_{t,c} \cdot d_E} \tag{8.11}$$

Since the global desirability is a proportion (always between zero and one), the response transformation of equation 8.12 is used to be able to do pairwise comparisons later on, as described in [82].

$$D'_c = 2 \cdot \arcsin\left(\sqrt{D_c}\right) \tag{8.12}$$

These transformed global desirabilities D'_c will also be used for the estimation of the steepest ascent parameters in EVOPSA.

8.4 Benchmark Testing

After improvement, the PTOS, PEOS and new lookup table implementations have to be compared with each other during a validation procedure. In this chapter, the procedure described in Stoev *et al.* [146] was adopted, which is summarized below. Since it is impossible for a human player to hit exactly in the same manner during each validation run, a play sequence was recorded and fed into the badminton robot to perform a benchmark test of the different control methods. The test sequence contains 84 shuttle hits and has a duration of approximately 8.5 minutes. This sequence is repeated three times to test if there is a significant difference in consumed energy between the implemented methods. The benchmark testing of PTOS, PEOS and the five EVOPSA lookup tables result in 21 benchmark tests which were executed randomly.

No changes to the controller were made. The same control scheme as in PEOS is used for the EVOPSA lookup tables, meaning that only the lookup tables themselves have been changed, not the interpolation code or the underlying controller.

The total sum of consumed energy in the linear motor during the benchmark test will be measured and the position errors are calculated. This position error is calculated as the difference between the actual position y_{act} and the desired position as given by the interception reference y_{des}. The position error helps to determine the precision of the badminton system during play. Precision is measured as the percentage of hits out of the total of 84 hits during the benchmark test that the robot is able to intercept. A hit cannot be intercepted when $|y_{act} - y_{des}| \geq 0.05\,\text{m}$.

From the benchmark tests the following information will be extracted and tabulated:

- *Method*: the controller method implemented; either PTOS, PEOS or PEOS-improved using one of the five desirability transforms. For ease of reference, the improved lookup tables were labelled with the name of the improvement method (EVOPSA) and the number of the time constraint that was used ($c = 0, 1, 2, 3, 4$).
- *Consumed energy*: The mean energy consumed on the linear axis during the benchmark tests.
- *Standard Deviation of Energy*: The standard deviation of energy on the three benchmark tests.

- *% improvement versus PTOS*: Percentage improvement in average energy consumption when compared to the average PTOS energy consumption.
- *% improvement versus PEOS*: Percentage improvement in average energy consumption when compared to the average PEOS energy.
- *Precision*: The percentage of hits intercepted during the three benchmark tests. There was no difference in precision between the three repeats of each method thus only one value is shown.

The interest is in comparing the new lookup tables with the current PEOS lookup tables. A one-way ANalysis Of VAriance (ANOVA) model was built with the optimization method as the factor and the consumed energy during a benchmark test as the response. A Dunnett Multiple Comparisons Procedure [48] was used to compare the sample means of the energy of the different EVOPSA results (treatments) with the PEOS method, which is the control. The joint significance level α for all tests was set at 0.05.

8.5 Results & Discussion

After constructing the five new lookup tables, all methods were benchmarked to compare performance. The methods under consideration are PTOS (no lookup tables, time-optimal motion), PEOS (current approach, lookup tables generated offline) and EVOPSA (lookup tables improved online, starting from the PEOS tables).

With EVOPSA, five different sets of lookup tables were constructed. The five implementations change the time constraint from on-time motion ($c = 0$) to time-optimal motion ($c = 4$). The results of the benchmark tests are summarized in table 8.1.

The results for the Dunnett's Multiple Comparisons Procedure are given in table 8.2. All improvements are significantly different from the PEOS control and will be included in the final comparison of the results.

The goal of the initial improvment, EVOPSA-0, was the fine-tuning of the current PEOS lookup tables to reduce the energy consumption. From the results, it can be seen that a reduction of approximately 4 kJ is achieved, which is an improvement of 5% compared to PEOS operation. This shows that the offline lookup tables can indeed be fine-tuned on the full scale system using online methods. In this case, the offline optimizations already did an excellent job of pin-pointing the region of the optimum and the simulation model used

Table 8.1: Summary of benchmark test results.

Method	Consumed Energy ± SD [kJ]			impr. vs. PTOS [%]	impr. vs. PEOS [%]	Precision [%]
PTOS	226.527	±	2.94	-	-	97.62
PEOS	83.782	±	0.18	63.02	-	94.05
EVOPSA-0	79.823	±	0.16	64.76	4.73	94.05
EVOPSA-1	82.982	±	0.11	63.37	0.95	94.05
EVOPSA-2	91.091	±	0.01	59.79	−8.73	95.24
EVOPSA-3	109.317	±	0.10	51.74	−30.48	97.62
EVOPSA-4	109.943	±	0.08	51.47	−31.22	97.62

Table 8.2: Dunnett Multiple Comparison, control = PEOS, $\alpha = 0.05$.

Difference with PEOS	Difference [kJ]	SE Difference [kJ]	p-value
EVOPSA-0	−3.960	0.098	$< 0.001^*$
EVOPSA-1	−0.800	0.098	$< 0.001^*$
EVOPSA-2	7.309	0.098	$< 0.001^*$
EVOPSA-3	25.536	0.098	$< 0.001^*$
EVOPSA-4	26.161	0.098	$< 0.001^*$

does not exhibit a large plant-model mismatch in the region of interest. In situations where this plant-model mismatch is larger, online methods will yield larger improvements.

The four additional improvements were executed to investigate the trade-off between consumed energy and precision of the system. As was expected, the more stringent the time-constraint (higher c), the higher the precision and consumed energy. When comparing methodologies, there are two contrasts that are of special interest: the contrast between EVOPSA-0 and EVOPSA-1, which shifts all time constraints with 0.045 seconds, and the contrast between PTOS and EVOPSA-4, which both yield the highest precision but differ in consumed energy. The tests for both of these contrasts indicated that the difference in consumed energies was statistically significant.

The shift in time from EVOPSA-0 to EVOPSA-1 (0.045 seconds difference between t_0 and t_1) increases the energy by 3.2 kJ, which is approximately 4%,

while keeping the precision the same. It is interesting to note that even with these more stringent time-constraints, the improved lookup tables still perform 1% better in terms of energy than the PEOS tables.

When examining the other implementations, the behaviour was as expected: the more stringent the constraint on arrival time (higher c), the better the overall precision of the system became. The results of precision and average consumed energy are plotted in figure 8.4 for the five improvements to visualize the trade-off between precision and consumed energy.

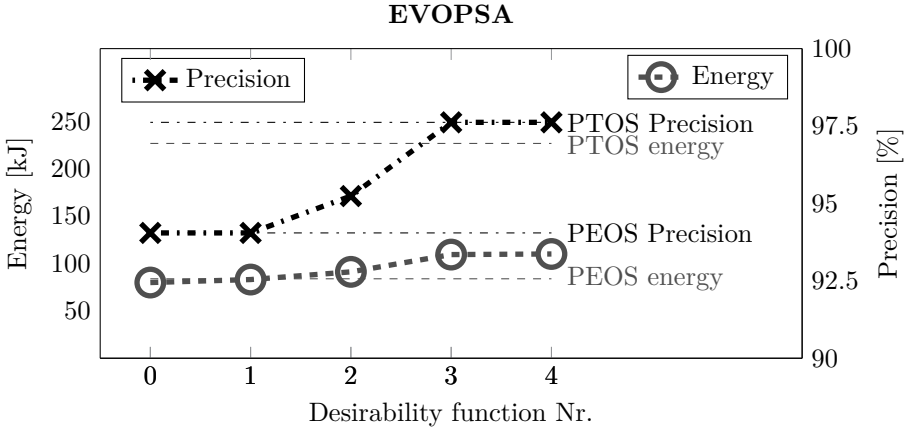


Figure 8.4: Trade-off between energy and precision.

From $c \geq 2$, the energy increased above the PEOS benchmark but also increased the precision above those of the PEOS case. EVOPSA-3 and EVOPSA-4 are equivalent in terms of performance, since one cannot increase the performance above the maximum performance of the currently implemented PTOS controller. In studying figure 8.4 a secondary, interesting result is found: the same precision as the time-optimal (PTOS) method could be reached with EVOPSA-3 but with an energy consumption that is 117.2 kJ lower, or 52% lower, compared to PTOS operation. For high precision operation, this is an extremely important improvement on the energy consumption.

Based on these results, a decision can be made on which implementation to run. If minimization of energy consumption is of paramount concern, then implementation $c = 0$ should be chosen for it has the same precision as the currently implemented tables but consumes less energy. If precision is most important, then implementation $c = 3$ should be selected since it has the highest precision possible at the lowest energy consumption for this precision.

It is clear from these results that the applied EVOPSA method offers an improvement over the currently implemented PEOS case. The main advantages are that the proposed method works online and is easy to implement. If the optimum of a process shifts over time, due to machine wear for instance, the method can be started at the current best known settings and the process can be shifted towards the new optimal settings with a low probability of undesirable output being generated by the improvement. These properties immediately lead to the disadvantages of the method: since it estimates local approximate models, the method cannot deal with local optima. Indeed, it is not the goal of this type of improvement to pin-point the global optima but to gradually improve the process to better operating conditions without sudden, drastic changes. Since the probability of undesirable output should remain low, the settings are changed in small steps during each optimization phase, making the method slow compared to a more adventurous exploration of the experimental domain. Such exploration could be included in the method by varying the step size δ between phases, yet this could also increase the probability of undesirable output.

The inability to deal with local optima and the speed of the improvement clearly indicate that this method needs a good parameter set to start from. In the case of the badminton robot, offline simulations provided the current starting sets (PEOS). In other processes where such a good set of settings is not at hand, a rough exploration of the entire experimental domain is needed, for instance using classical [107] or optimal [58] designs, or by using sparse, space-filling designs, as discussed in chapter 4, to try and extract as much information as possible with a low number of experiments.

8.6 Conclusions

The case study handled in this chapter shows that there is an added value in the further online improvement of processes. After a first offline optimization—in this case a first principles model optimized with Genetic Algorithms— OSPI schemes can be used to fine-tune the optimum. For badminton robot motion, traditionally, the motion controllers are to realize fast and accurate motions in order to optimize the system's productivity, the so-called time-optimal motion control. The raising energy prices and growing environmental awareness have initiated a shift in these requirements to find a trade-off between time-optimal and energy-optimal motion control. In this paper, a constrained optimization dealing with this shift in requirements—minimize energy subject to a time-constraint—was dealt with.

The original time-optimal control (PTOS method) was previously optimized offline using a simulation model and the resulting lookup tables (PEOS method) already provided a drastic improvement over PTOS operation. In this chapter Evolutionary Operation Steepest Ascent—which combines the statistical estimation of a local approximate model with a line search—is used to improve a badminton robot online with as starting parameters the offline optimization results.

The constrained nature of the problem was dealt with by treating it as a multi-objective problem which was then scaled to a single-objective criterion using Derringer Desirability functions. By using the combination of EVOPSA and Derringer desirability functions, the lookup tables could be further improved and a reduction of energy consumption of 5% was achieved while keeping the precision of the system at the same level as the PEOS approach.

Introducing more stringent time-constraints increased the precision of the robot, at the cost of an increased energy consumption. One of the resulting lookup tables with more stringent time-constraints manages to increase the precision to the maximum achievable precision but with a reduction in energy consumption of 52% compared to the current implementation with maximum precision, namely time-optimal motion under PTOS operation. These results clearly indicate that online, full scale process improvement can provide additional benefits to offline optimization.

Part IV

Conclusions

9 Conclusions

9.1 General Conclusions

Improving full scale processes is a challenge that is usually tackled by using offline methods on small scale versions of the process (lab, pilot) and to use the optimal settings of those processes as inputs for the production plant. Alternatively, physics-based computer models that mimic the real-life process are built, and the optima of those "in silico" models are then applied to the real process. Besides issues related to upscaling and plant-model mismatch, non-stationary optima are also a relevant problem, especially in processes where biological material is involved. Also influences that induce a sudden shift in the optima such as changes due to batch-to-batch variation, shifts in environmental conditions and varying machine behaviour after maintenance require an additional adjustment step on the process itself when they occur. Methods that can cope with these situations are required, and are an integral part of this dissertation.

In view of the above limitations of traditional (offline) process improvement, a shift to online process improvement is required. In this dissertation methods of Online Sequential Process Improvement (OSPI) were investigated. The goal was a thorough research of the validity of traditional OSPI methods for contemporary processes, i.e. processes with a high amount of sensors and factors. To achieve this goal, three research objectives were set: (1) an investigation of the methodology and eventual extensions for contemporary processes, (2) an in-depth investigation of the methods through simulation studies and (3) the application to a practical case study. The general conclusions will be discussed per research objective in the following sections.

This dissertation contributes to the state-of-the-art by (1) providing automated procedures for direction selection for EVOP, (2) providing a software package to apply OSPI methods, (3) comparing Simplex and EVOP in a systematic

manner—which was never done before—(3) investigating the use of efficient designs in higher dimensions, (4) presenting the foundations the optimal statistical power and related sample size for EVOP improvements and (5) highlighting the use of desirability functions for multi-objective processes.

9.1.1 Methodology

The traditional OSPI methods, EVOP and Simplex, did not evolve much in the decades since their inception. When reviewing the literature it is shown that Simplex is a largely forgotten method for process improvement as its derivative, the Nelder-Mead or Variable Simplex, has accrued considerable attention in numerical optimization. However, this Variable Simplex method is not suited for online process improvement due to its variable step size which is problematic for noisy systems (when the step size becomes small) and might violate the restrictions of small changes to the factor levels, i.e. low probability of unsaleable output (when the step size becomes large).

While the simplicity of Simplex—due to its heuristic rules—allows for easy automated implementation, the EVOP method is still mostly applied manually with few (i.e. two applications) reports of automated implementations. For an automated implementation formal rules are necessary for defining the direction and step size of the move. Such rules were formalized in this work based on the first order derivatives of the main effects model built in every EVOP phase.

The use of steepest ascent within the EVOP framework—which was already suggested in the original paper—was formally presented as well. Since EVOP was run manually before, no computational logic was needed to deal with the boundaries of the experimental domain. In a truly automated procedure keeping the method inside of the experimental domain requires some computational logic and a novel method to deal with an experimental domain bounded by a hypercube was presented.

All OSPI methods depend on a good prior estimate of the factor levels where to start from since they are known for their proneness to get stuck in local optima. This is logical since only small changes in the factor levels are allowed during every improvement phase to conform to the requirement that there should be a low probability of unsaleable output. The availability of a good prior estimate of the factor levels is of paramount importance to the success of OSPI methods. However, when no such initial estimate is available experimentation has to be carried out to select this prior. Contemporary processes are often characterized by nonlinearities and possible local optima. To deal with this, a framework to establish the region of the optimal factor levels was proposed that uses a limited experimental effort and spans a sufficiently broad class of models,

potentially nonlinear and having local optima. The combination of space-filling designs with Gaussian Process modelling established this prior. This framework was compared extensively with classical Response Surface Methodology and Ordinary Least Squares regression on a simulation study and it was confirmed that this framework offers a good prior estimate of the settings. Afterwards, it was successfully applied to a practical case to establish the prior for an EVOP improvement. Starting from the settings of initial prior, an increase of 7.5% in the output of the system was achieved.

Summarizing, the two traditional OSPI methods, EVOP and Simplex, are reviewed and it is found that Simplex is not applied to processes often. When it is applied, it is usually on processes with a high SNR (low noise) and when rapid improvement is necessary. EVOP on the other hand is still applied but the original method lacks formal rules for the direction or step size of the move impeding automatic implementation. Formal rules for EVOP and a novel method to deal with the boundaries of the experimental domain are presented to enable automation. In order to have a good starting point for EVOP a framework consisting of the use of space-filling designs and Gaussian Process modelling is proposed to determine this prior with limited experimental effort.

9.1.2 Simulation Studies

There is no systematic literature which compares EVOP and Simplex and their performance which is a serious gap when one wishes to asses which method performs best under which conditions. To asses this shortcoming, a study was performed in which the dimensionality of the problem, the amount of noise in the simulation and the size of the perturbations (controls the size of the design region, called factorstep in this dissertation) were varied. Since all these settings cannot be changed on a physical process, it was chosen to execute a simulation study to compare the two methods.

From this study it was evident that the factorstep is one of the most essential parameters when performing an improvement and has a direct effect on the robustness of both EVOP and Simplex. The effect of the factorstep cannot be uniquely quantified as it is process-dependent and will require experience with the process or historical data to assess. From the two methods, the EVOP procedure is more robust against changes in the factorstep, meaning that the performance of the Simplex method is directly tied to the effect of the factorstep.

From the simulations it was concluded that Simplex is the preferred choice when dealing with deterministic or low-noise systems, but the method is very susceptible to changes in the factorstep. This implies that more process knowledge is needed to start a Simplex improvement to accurately determine

to factorstep. In general, if the goal of the improvement is the determination of the optimal region, and one expects that this region lies far away from the current operation conditions, the factorstep will be chosen larger and one can use Simplex. In the other cases EVOP is the safer choice. The EVOP method is advised when the process is characterized by a high amount of noise present in the system and/or a dimensionality above three covariates.

Due to the full factorial that is traditionally used in EVOP, the experimental effort (number of measurements) might become prohibitive as the dimensionality increases. Indeed, using the full factorial with more than five or six covariates might be inefficient as 32 or more measurements need to be performed for every improvement phase. In chapter 6, efficient designs were proposed to tackle the issue of a large experimental effort that is required when a full factorial is used as a base design for cases with more than five covariates. It was shown that such efficient designs provided not only a solution to this issue, but allowed to reach the region of the optimum using less experimental effort. This is an advantage for non-stationary processes in which time-drift is present. It was also shown that they tend to stop farther away from the optimum, especially if the noise-level increases (since the statistical power will become too low to detect the very small effects present in this design region). Moreover, this simulation study showed that the required power for detecting the effects could be much lower than what is classically used in experimental design, but it remained unclear how low this value can be to be acceptable, and how this optimal power depends in the dimensionality of the problem.

This question concerning the relation between optimal power and dimensionaltiy was investigated in chapter 7 were a simulation study on a linear model was performed to offer first insights into this relation. $\mathcal{D}$-optimal designs were used to allow for a varying sample size (dependent on the requested power) and Gaussian Process models were built on the simulation data. It was concluded that the optimal power is a function of the dimensionality: for low dimensionalities ($k < 8$) a very low power ($\pi < 0.4$) leads to a minimal number of measurements, while for high dimensionalities ($k \geqslant 8$) a broad, almost flat valley ($\pi \in [0.4; 0.8]$) was observed. It is possible to choose the power in this valley without affecting the sample size to a large extent. Therefore, a choice can be made base solely on the type of process under study: for processes with a low sampling rate or non-stationary process prone to time drift a low power is recommended, whereas a higher power is advised when a high sample rate is possible, or when the process is stationary. The results from this chapter gave insight into the relation between the optimal power and the dimensionality yet are only initial results and should be verified and further investigated on different simulation cases.

Summarizing, Simplex is a technique better suited to quickly reach the region of

the optimum—if one is far away from it—by using a relatively large factorstep after which one switches to EVOP to accurately pin-point the optimum. However, the traditional full factorial base design for EVOP becomes prohibitive when the dimensionality increases and other, more efficient designs need to be incorporated. The sample size of these designs can be determined by performing a sample size estimation using the concept of statistical power.

9.1.3 Case Study

After extensive testing on simulation studies, the OSPI methodology was applied to a practical case. The controller parameters for a badminton robot were improved starting from tables of optimal (offline estimated) parameter settings for different motion distances and times. These tables provided the prior estimates for the controller parameters and as many improvements as parameter sets available were performed.

Since the amount of measurements that could be performed was very limited traditional EVOP was not feasible to execute. This left the Simplex and EVOP Steepest Ascent method. The EVOPSA method delivered a higher improvement than the Simplex method, therefore it was selected to perform the improvement. This method combines the best of EVOP, statistical model-building and inference, and Simplex, sequential augmentation with only one point in the line search step.

The constrained nature of the improvement, minimize energy subject to the arrival time constraint, was dealt with by interpreting it as a dual-objective: minimize energy and keep arrival time below a certain threshold. This multi-objective problem was scaled to a single-objective by using Derringer desirability functions. The primary objective of the study—being to improve the offline set of lookup tables (one per controller parameter) in an online fashion—was achieved and a 4.7% decrease in energy consumption was attained, without a loss in motion precision. This was measured during benchmark testing and compared to the offline generated lookup tables. The secondary objective of the study was the search for a set of lookup tables which increased motion precision to the precision of time-optimal motion but at a lower energy consumption. This objective was achieved by implementing more restrictive transformations for the arrival times and resulted in the identification of lookup tables with the same precision as time-optimal motion but with an energy consumption that is 52% lower than time-optimal motion.

Summarizing, EVOPSA was used in a case were the number of samples was severely limited to profit both from the model-building capabilities of EVOP and the augmentation with only one measurement during the line search. The

use of desirability functions to deal with the constrained nature of the problem by redefining it as a multi-objective problem proved a valid approach. And apart from the proposed applications in the process industry, this case study also shows the validity of OSPI methods for controller parameter tuning. In analogy with the process industry, every improved parameter set can be seen as one production line that was fine-tuned.

9.2 Future Perspectives

The fundamental work to prepare OSPI methods for their introduction in contemporary processes has been presented in this work yet many avenues of research are still to be considered. There are three main aspects that future work must take into consideration: (1) extensions and refinement of the methodology, (2) a proof-of-concept of a process that fully integrates OSPI and (3) effectively using OSPI in industry.

9.2.1 Methodology

Flexible designs. The use of efficient designs already drastically reduces the number of measurements in every EVOP phase but one is always bound by the construction rules of the designs—which in the case of the efficient design discussed in this text is a $1/2^{p_f}$ fraction of the design points of a full factorial. Being able to choose an arbitrary number of design points per phase opens up the possibility of even more performant improvement, as demonstrated in the simulation study for optimal power. The field of optimal design allows to build designs with an arbitrary number of design points (with the restriction that at least the minimum number of points necessary to estimate a previously defined model have to be measured, this number being $f_t + 1$ with f_t the number of model terms excluding the intercept).

This concept was already used in chapter 7 but should be extended since computer-generated optimal designs allow for the inclusion of additional restrictions. When there is a time-drift, a blocking factor can easily be introduced to try and estimate the effect of this drift. Blocking could also be used to test for the variability between small batches (note: blocking is also possible using classical designs, however in optimal design one has much more freedom in choosing the blocking factors and the levels of these factors). Furthermore, if certain factors are harder to change then others, an optimal design can be generated that incorporates the hard- and easy-to-change factors. For instance, the temperature of an oven could considered hard-to-change since changing the

temperature will have a slow effect on the response, if one is not interested in transients one will have to wait until the temperature is at the correct level. Therefore changing the temperature should be done minimally and not every experimental measurement.

Using computer-generated optimal designs (such as for instance the $\mathcal{D}$-optimal design) poses some additional challenges that need to be investigated. With the main concern being the fact that constructing an optimal design might be computationally more intensive than is possible to execute on a low-level controller. If optimal designs are used, several options need to be investigated: (1) should an optimal design be generated in every EVOP phase, possibly leading to a slower software since the construction is carried out at the start of every phase or (2) should an initial (coded) optimal design be created and the placement of these design points be re-used in every subsequent phase. This second possibility might even lead to the recommendation (3) to use a design library of optimal designs at a low-level controller from which the algorithm can select one. However, running a construction algorithm several times might lead to different optimal designs and, consequently, this different placement of the design points might have its impact on the improvement capability in every EVOP phase and should be investigated thoroughly before making any recommendation on their use.

Inclusion of interaction terms. A powerful feature of EVOP based on a full factorial base design is the fact that interaction terms—which appear quite often in production processes—can be estimated as well. In the approach proposed in this dissertation, where coded designs were used and where the calculation of the steepest ascent parameters were derived from the centre of the design region, the need for interaction terms was partly removed. However, in a next step the performance of the EVOP method needs to be compared against an implementation that takes into account interaction terms. To take advantage of the interaction terms, the direction of the move should not be calculated from the centre of the design region since the interaction terms would disappear but from the estimated optimal settings within the design region. It is expected that such an adaptation will have a large effect on the EVOPSA procedure during—what is now called—the line search step were the direction will no longer be situated along a line but along a curve due to the inclusion of the interaction terms. When many factors are taken into an EVOP improvement it becomes unfeasible to use the full factorial, but interactions that are expected to be relevant might be included by using optimal designs or fractional factorials of resolution IV in which some two-way interactions can be estimated.

Lack-of-Fit test. The inclusion of centerpoints in the factorial designs allows doing a formal lack-of-fit test. Such a test will indicate if the model fitted to the design region exhibits a lack of fit, i.e. if there is an indication of curvature

within the design region. The presence of curvature indicates that the design region might encompass an optimum. In such a case the design could be augmented to be able to fit quadratic terms (e.g. by the addition of star points to form a Central Composite design from a fractional or full factorial design) and try to accurately pin-point the optimal settings within the design region.

Stopping criterion. Some attention should be given to an intelligent stopping criterion for the methods. For Simplex stopping criteria are already proposed in literature, most of them specifically tailored for the Variable (Nelder-Mead) Simplex. One simple criterion for the Simplex is to stop the method after a particular point is retained an arbitrary number of times. Since the Simplex has the tendency to retain the best point and (in 2 dimensions) circle the optimum, the retention of a point after several reflections is a fair indication that the Simplex method will no longer advance. For EVOP possible criteria to investigate are: (1) criteria based on a formal lack-of-fit test when centerpoints are included since the presence of curvature indicates the possible presence of an optimum in the design region or (2) those based on the steepest ascent vector.

When all terms in the steepest ascent vector are zero (i.e. when no coefficient in the regression model is found to be significant) no move is executed (the next phase is "stationary"). If an arbitrary number of such stationary phases are performed the method should also stop since the procedure is no longer moving. If the steepest ascent vector changes signs in between phases (i.e. the EVOP procedure keeps on jumping from one phase to the previous one and vice-versa) no improvement is made and this can be an indication that the optimum is located somewhere in the vicinity of the design regions of these two phases.

Non-stationary processes. In this dissertation only stationary processes are considered yet part of OSPI is its professed ability to track non-stationary optima. Almost no literature has been devoted to this topic and with the current automated extensions proposed it becomes interesting to investigate this further. For Simplex, an extension which is called Dynamic Simplex is proposed in literature [160]. For EVOP such an extension targeted at nonstationarity is not proposed and should be investigated. The same simulation models used in the publication about Dynamic Simplex could then be used both for Basic Simplex and EVOP to compare performances with the Dynamic Simplex.

Sequential Gaussian Process modelling. Although not elucidated in this text, GP modelling is a powerful tool in which an initial design can be sequentially augmented with additional design points to refine the model. Techniques such as the use of the Expected Improvement Criterion (see for instance Boyle's PhD thesis for an explanation [24]) which indicates where the uncertainty about prediction is highest and as such the expected improvement of the model by placing an additional design point is highest, or region-searching

[148] where the design is augmented with points in the region of the expected optimum are an excellent avenue for further research. Replacing current EVOP methods with such an incrementally increased model seems interesting but one has to bear in mind that GP modelling is computationally expensive and becomes drastically more so when the sample size increases. Besides, in order to adhere to the OSPI philosophy, also restrictions related to the maximum step size from one setting to a next one should be implemented. This warrants further investigation to determine whether GP modelling or a combination of current techniques and GP modelling for sequential augmentation might yield promising results. This is a feasible option when no prior knowledge about the initial starting conditions is known and space-filling designs, that span the entire experimental domain, are executed as the initial design.

Optimizing the direction of the move. Currently the steepest ascent parameters are determined by the first order derivatives of the linear model and the direction of the move is plotted from the centre of the (coded) design region. Another valid approach might be to numerically optimize the statistical model and finding the best settings within the design region (which in case of a linear model without interactions will always be located at the edge of the design region). These settings can then be used as the starting point for the move, or for the determination of the steepest ascent vector which does not make use of the first order derivatives (and might be computationally more intensive). Additionally quadratic models could be used and this quadratic information could be included to determine the direction of the move. In such a case Newton's method could be used to approximate the step size (while satisfying the Armijo-Wolfe criteria, i.e. ensure that there is a sufficient decrease in step length and slope of the path in each step). A comparison of the Newton step with a gradient descent procedure is shown in figure 9.1. From this figure, it can clearly be seen that the Newton step is more efficient than the gradient descent procedure (provided quadratic information is available).

Sequentially augmented improvement direction. As implemented in this Ph.D, at every new EVOP phase the data of the previous phase is no longer used. However, this data could be used to more precisely estimate the improvement direction. For instance one could restrict the path of steepest ascent to a smooth curve, i.e. by adding a penalty for the difference between the previous steepest ascent vector and the current estimated one (e.g. is the difference is large, the new direction will be more restricted along the slope of the previous direction). Furthermore if a phase is stationary (i.e. no move is executed) the measurement data of all stationary phases in the same design region can integrally be used to enhance the statistical power.

Variable step size. A variable step size could increase the rate of improvement in both methods. Yet, as stated before, a more adventurous exploration

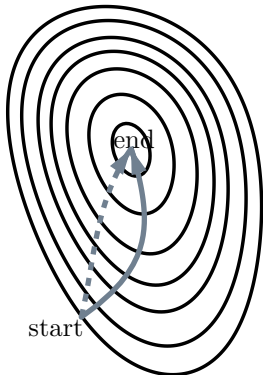


Figure 9.1: Comparison of gradient descent (full) and Newton's method (dashed) for minimizing a function (with small step sizes). Figure adapted from `http://en.wikipedia.org/wiki/Newton's_method_in_optimization`.

of the experimental domain might be unwarranted since the probability to produce unacceptable output might increase. It is interesting to test this theory by implementing a variable step size with restrictions on the minimum and maximum size to avoid problems of the changing length of the move.

9.2.2 Using Online Sequential Process Improvement In Industry

Popularisation & Application. Not many applications in industry are published in the last decades. There is some resurgence in applications since the turn of the century but their number is still quite limited. Often the practical mentality in industry requires methods to be proven before the process supervisor would take the chance of applying it to a process. Therefore more practical case studies need to be performed to convince industrial partners to apply the methods. This can be an arduous road since one has to find companies willing to implement the methods on a practical process and which allow dissemination of the results. Another stumbling block is the total absence of commercial software that implements OSPI. Before becoming acceptable industry-wide, those case studies should trigger software companies to invest in OSPI as a useful tool.

Incorporating Operator Experience. As Box stated when initially introducing the OSPI concept, operator experience is of paramount importance to start up an OSPI improvement. In most cases operators have a good

understanding of which parameters influence the process and how they possibly interact. This experience can be used to select the key factors and to select new ones should external influences change the key factors of a process. It is important in the contemporary, automatic setting to not forget this valuable input.

Challenges. Although the OSPI methodology is sound and easy to understand, its application in industry is still lacking. Apart from the reluctance of process supervisors to incorporate little known methods, the available sensor technology is also often an obstacle. The methods require some (automatically) measured characteristics that one wishes to improve. Not in all industries such measurements are readily available or easy to implement. An example of an industry in which such might be readily available is the chemical industry (e.g. measurements of yield, pH, and concentration [69]). In other environments sensor measurements are becoming more and more the norm and, often, sensors implemented in the machines for maintenance might correlate to the characteristics one wishes to improve (e.g. torque of a conveyer belt could correlate with yield). The challenge is to identify these sensors and, if they are not present, convince management to procure them. A manual or a non-integrated automatic OSPI improvement could be executed to prove the validity of the approach and convince management of the procurement of the necessary sensors.

Integrating OSPI In The Machine. Currently the developed software is connected to existing machines by writing an extension or connector for sensor measurements and machine control but the best way to validate the use of OSPI methods would be a production machine which incorporates OSPI algorithms in the machine software. The methods could then be started automatically by the machine when it deems necessary to improve the process (e.g. when changing batches, at first start-up during the week, continuously, when a shift in response is detected, when a change in key variables is detected as presented in chapter 2). Ideally, such a machine should be compared to one without OSPI integration and their performance over a longer period of time should be assessed.

Economic value of running OSPI. Convincing management to use OSPI in conjunction with classical optimization techniques or by implementing it stand-alone should be done by showing them the economic value of running the OSPI scheme. There are two major economic advantages of running the scheme: (1) The process is being gradually improved which can increase throughput, quality, customer satisfaction and decrease defect rates and complaints. (2) Compared to classical optimization strategies the gradual improvement of the production process allows to keep the production within acceptable bounds. In contrast, a classical RSM approach usually involves halting the production process during experimentation, which will cost the company money without

a guaranteed return on investment. It is the main philosophy of OSPI to not influence the production process too much, therefore it can be run while still producing saleable product and without influencing the production process too much. It would be beneficial to calculate the cost of running a classical approach compared to an OSPI approach for a specific process as a case study. Furthermore additional research is necessary to investigate and optimize the costs and benefits. The simulations in this Ph.D were focussed mainly on a minimal amount of measurements without taking into account the costs of producing product of less than the intended quality. Furthermore implementation and running costs should be investigated to determine whether the method is economically relevant for specific cases.

Opportunities. During the course of this research several visits to companies were made and it was observed that—in for instance the packaging industry—every operater has his own machine settings (factor levels), which can be vastly different. This shows that there is a large room for improvement and that an automated approach that finds the optimal combination of factor levels is much needed. Furthermore several processes are sensitive to environmental conditions (see for instance [66]) and batch-to-batch variations. In such production environments an increased stress-level was observed for the operators since trial-and-error changes to the settings are applied to move the process within acceptable bounds once a significant deviation in the output is observed. Using OSPI could offer an automated tool to keep, or move, the process within acceptable bounds. It is even possible to use environmental data (e.g. humidity, temperature, ...) within the OSPI approach. Moving towards industry 4.0 which will bring more sensors, and sensor fusion, to the production enviroment will offer more and more data for automated improvement methods to take into account while trying to keep a process within bounds (or when improving it). This shows that there is a large, potential application area for the methods within industry.

Part V

Appendix

A Examples And Documentation Of Matlab Software Package For OSPI Methods

A.1 Introduction

A significant effort was delivered to develop a framework of Matlab® classes that make it easy for a user to work with the presented Online Sequential Process Improvement methods.

Object Oriented Programming (OOP) was used for this software package and the following terminology is used in this context: one improvement method is called a *class*, creating an improvement method with settings for executing an improvement is called an *instance* of this class, functions applied to data which are only specified for a specific class or instances of this class are called *methods*. Without going into further detail, this terminology will be used here to explain generally, the advantages of an OOP framework.

- All relevant *methods* for an improvement are "packaged" together, there is no confusion possible. This allows us to use the same *method* name for every *class*, which will make it much easier to switch between improvement methods;
- One placeholder variable will hold the entire *instance*, i.e. all relevant settings and the improvement method are saved using this one variable which makes (global) variable management much easier;
- It is possible to extend classes, in the OOP terminology make a child class of its "parent". This allows the child class to inherit all methods from the parent. This is used for EVOPSA, which is a special application of the EVOP improvement. This allows us to maintain the relevant

settings, which are identical for both methods, in one class. EVOPSA will automatically inherit all methods from EVOP.

The software code was structured in such a way that it is easy to convert it into other programming languages, foreseeing the implementation of the methods in other programming languages and hardware platforms. This appendix chapter will describe the functions that are made available to the user of the classes as well as (pseudo-) examples of improvements to illustrate how simple running the algorithms is.

Every improvement that is executed on a physical process will adhere to the following flow:

- Use process knowledge to determine relevant factors, responses, noise-level and factorstep;
- Determine hard- and software necessary to run the process, measure its responses and influence the factors and interface with the improvement software;
- Use the developed software package to run the improvement.

Process knowledge will be abstracted here by choosing one response (unnamed), a process with four factors ($k = 4$), a center of the first design region $\mathbf{x_{start}}$, factorsteps $\mathbf{dx}$ for the four factors and boundary constraints $\mathbf{X_{lim}}$ for the four factors. These settings are summarized in table A.1. Hardware and hardware interfacting will be abstracted by representing the interfacing with the software by two generic functions: (1) `IOsetFactors(factorSettings)` which expects a $1 \times k$ vector holding the settings for the factors as the input. The response is gathered from the process by `IOgetResponse()`, it is assumed that the function waits until the measurement procedure is completed before sending data to the program. The examples will be run without a stopping criterion for the improvements, this is something that will have to be chosen by the experimenter.

Table A.1: Improvement settings for the software examples

Setting	Value
k	4
$\mathbf{x_{start}}$	$\{0; 0; 0; 0\}$
$\mathbf{dx}$	$[1; 0.8; 0.5; 3]$
$\mathbf{X_{lim}}$	$\begin{bmatrix} -50 & 10 \\ -40 & 30 \\ -80 & 55 \\ -100 & 100 \end{bmatrix}$

A.2 Software Methods

This section documents the software methods available to the user. To simplify use, all improvement methods share some basic methods, these will be given once under "General Methods" while methods related to a specific improvement scheme are presented seperately.

A.2.1 Simplex

S = Simplex(k,initialSimplex,xco,dx)

Constructor of the Simplex class, creates an instance of the Simplex method with all proper parameters set

Inputs	k	The number of factors to measure
	initialSimplex	The initial simplex, either "$\mathcal{D}$-optimal", "corner" or "tilted"
	xco	A $1 \times k$ vector with the starting coordinates, see chapter 2
	dx	A $1 \times k$ vector with the factorsteps, see chapter 2
Outputs	S	An instance of the Simplex class

resetInitialSimplex(coords)		
Overwrite the initial simplex by one that is set manually		
Inputs	coords	A $(k+1) \times k$ coordinate matrix for the $k+1$ simplex points
Outputs	-	

setReplications(r)		
Set the number of replications for every point. Every point will be measured this number of times and the average response of all measurements will be used internally, see [155]		
Inputs	r	A positive integer denoting the amount of replications, the minimum (and default) is 1, meaning every point is measured once
Outputs	-	-

A.2.2 Evolutionary Operation

E = EVOP(k,dx,xstart,nc,r,design,varargin)

Constructor of the EVOP class, creates an instance of the EVOP method with all proper parameters set

Inputs	k	The number of factors to measure
	xstart	A $1 \times k$ vector with the centerpoint of the first design region, see chapter 3
	dx	A $1 \times k$ vector with the factorsteps, see chapter 3
	nc	Number of centerpoints in the design
	r	The number of times the factor levels (excluding the centerpoint) in a design should be measured to complete a phase
	design	An integer indicating the type of design to use: (0) full factorial, (1) fractional factorial (mininum resolution III), (2) $\mathcal{D}$-optimal design
	varargin	Optional argument that only needs to be set when a $\mathcal{D}$-optimal design is used. Then it should be set as an integer which denotes the sample size of the design (excluding the centerpoints)
Outputs	E	An instance of the EVOP class

A.2.3 Evolutionary Operation Steepest Ascent

ES = EVOPSteepestAscent(k,dx,xstart,nc,r,design)		
Constructor of the EVOPSteepestAscent class, creates an instance of the EVOPSA method with all proper parameters set		
Inputs	k	The number of factors to measure
	xstart	A $1 \times k$ vector with the centerpoint of the first design region, see chapter 3
	dx	A $1 \times k$ vector with the factorsteps, see chapter 3
	nc	Number of centerpoints in the design
	r	The number of times the factor levels (excluding the centerpoint) in a design should be measured to complete a phase
	design	An integer indicating the type of design to use: (0) full factorial, (1) fractional factorial (mininum resolution III), (2) $\mathcal{D}$-optimal
Outputs	ES	An instance of the EVOPSteepesAscent class

A.2.4 General Methods

setLogFolder(folder)		
Set the path to a folder for logging the configuration (settings given in the constructor) and the raw data (if data logging is enabled)		
Inputs	folder	String describing the path to the log folder, either relative or absolute
Outputs	-	-

setEnableData(setData)

Enable or disable logging of the data, data is logged in the folder which is set by setLogFolder

Inputs	setData	A boolean which is set to `True` to enable logging and set to `False` to disable logging
Outputs	-	-

setEnableConfig(setConfig)

Enable or disable logging of the configuration to a file, this file is logged in the folder which is set by setLogFolder

Inputs	setConfig	A boolean which is set to `True` to enable logging and set to `False` to disable logging
Outputs	-	-

setEnableLimits(setLimits)

Enable or disable constrained improvement, the hypercube limits of the experimental domain are set by setLimits

Inputs	setLimits	A boolean which is set to `True` to enable constrained improvement and set to `False` to disable it
Outputs	-	-

setLimits(factorLimits)

Set the factor limits for constrained improvement (i.e. constrained to an experimental domain defined by a hypercube). This method must be called before enabling constrained improvement!

Inputs	factorLimits	A $k \times 2$ matrix with, for every factor, the minimum and maximum factor level that bounds the experimental domain hypercube
Outputs	-	-

coords = readPoint()

Get the coordinates of the points that need to be measured next (this method will always return the same set of coordinates until the setResponse method is called)

Inputs	-	-
Outputs	coords	A $1 \times k$ coordinate vector of the new measurement point

numpoints = getPhasePointsRemaining()

Gives the number of points remaining in the current phase, necessary in toplevel programs to determine if a cycle is finished (for Simplex always gives 0 after the initial cycle since Simplex is augmented with only one point)

Inputs	-	-
Outputs	numpoints	An integer with the number of points remaining in the current cycle

setResponse(response)

Enter the response, this function will call a step procedure internally once a cycle is completed

Inputs	response	A floating point number defining the response of the finished measurement
Outputs	-	-

close()

Finish the improvement: force last data to be written to file and write configuration with current timestamp

Inputs	-	-
Outputs	-	-

A.3 Examples

Algorithm A.1. Matlab® example program for using Simplex improvement in the developed software package.

```
%% Define settings
k = 4;
xstart = [0 0 0 0];
dx = [1 0.8 0.5 3];
initialSimplex = 'tilted';
xco = xstart-dx/2;% The initial point of the Simplex can be
    defined from the centre of the first design region

%% Create Simplex instance
S = Simplex(k,initialSimplex,xco,dx);

%% Set the folder in which everything is logged (default 'log')
S.setLogFolder('data');
S.setEnableData(1); %Log measurement data
S.setEnableConfig(1); %Log measurement configuration data

%% Set factor level limits
% Constrain the improvement to the hypercube bounded by these
    factor levels (one row for every factor)
factorLimits = [-50 10; -40 30; -80 55; -100 100];
S.setLimits(factorLimits);
S.enableLimits(1); % Constrained improvement

while 1,
    % Get the factor levels of the next point to measure
    factorSettings = S.readPoint();
    % Set factor levels on process
    IOsetFactors(factorSettings);
    % Wait until measurement is finished, get response
    response = IOgetResponse();
    % Feed response to improvement method
    S.setResponse(response);
end
S.close(); % Write last data to file, save configuration
```

Algorithm A.2. Matlab® example program for using EVOP improvement in the developed software package.

```
%% Define settings
k = 4;
xstart = [0 0 0 0];
dx = [1 0.8 0.5 3];
nc = 1;
r = 1;
design = 1; % Fractional factorial min Res III

%% Create EVOP instance
E = EVOP(k,dx,xstart,nc,r,design);

%% Set the folder in which everything is logged (default 'log')
E.setLogFolder('data');
E.setEnableData(1); %Log measurement data
E.setEnableConfig(1); %Log measurement configuration data

%% Set factor level limits
% Constrain the improvement to the hypercube bounded by these
    factor levels (one row for every factor)
factorLimits = [-50 10; -40 30; -80 55; -100 100];
E.setLimits(factorLimits);
E.enableLimits(1); % Constrained improvement

while 1,
    % Get the factor levels of the next point to measure
    factorSettings = E.readPoint();
    % Set factor levels on process
    IOsetFactors(factorSettings);
    % Wait until measurement is finished, get response
    response = IOgetResponse();
    % Feed response to improvement method
    E.setResponse(response);
end
E.close(); % Write last data to file, save configuration
```

Algorithm A.3. Matlab® example program for using EVOPSA improvement in the developed software package.

```
%% Define settings
k = 4;
xstart = [0 0 0 0];
dx = [1 0.8 0.5 3];
nc = 1;
r = 1;
design = 1; % Fractional factorial min Res III

%% Create EVOPSteepestAscent instance
ES = EVOPSteepestAscent(k,dx,xstart,nc,r,design);

%% Set the folder in which everything is logged (default 'log')
ES.setLogFolder('data');
ES.setEnableData(1); %Log measurement data
ES.setEnableConfig(1); %Log measurement configuration data

%% Set factor level limits
% Constrain the improvement to the hypercube bounded by these
    factor levels (one row for every factor)
factorLimits = [-50 10; -40 30; -80 55; -100 100];
ES.setLimits(factorLimits);
ES.enableLimits(1); % Constrained improvement

while 1,
    % Get the factor levels of the next point to measure
    factorSettings = ES.readPoint();
    % Set factor levels on process
    IOsetFactors(factorSettings);
    % Wait until measurement is finished, get response
    response = IOgetResponse();
    % Feed response to improvement method
    ES.setResponse(response);
end
ES.close(); % Write last data to file, save configuration
```

B Power Analysis Program In Matlab For Coded, Orthogonal, Two-level Designs

Algorithm B.1. Matlab® example program for calculating the power of coded, orthogonal, two-level designs

```
clear all
close all
clc

% SETTINGS
alpha = 0.05; %significance level
sigma = 1; %error standard deviation
beta = 0.5; %size of coded linear regression coefficient one
    wishes to detect
k = 6; %Number of factors
p = k; %Number of parameters in the model, excluding intercept
nt= 20; %Sample size, total number of measurements
nu1 = nt-1-p; %Degrees of freedom for error

% CALCULATE POWER
phi = beta/sigma*sqrt(nt); %Non-centrality measure
criticalHigh = tinv(1-alpha/2,nu1); %Critical t-value t*
criticalLow = -tinv(1-alpha/2,nu1); %Critical t-value t*
power = (nctcdf(criticalLow,nu1,phi))+ (1-nctcdf(criticalHigh,nu1,
    phi)) %Power by using the cumalative non-central t-distribution
     density function
```

C Estimate Sample Size For Orthogonal, Coded, Two-Level Designs In Matlab

Algorithm C.1. Matlab® example program for calculating sample size of orthogonal, two-level designs starting from a coded regression coefficient (Part 1)

```
%   SETTINGS
power = 0.5; %The power one wishes to attain
alpha = 0.05; %significance level
sigma = 1; %error standard deviation
beta = 0.5; %size of coded linear regression coefficient to detect
k = 6; %Number of factors
p = k; %Number of parameters in the model, excluding intercept
```

Algorithm C.1. Matlab® example program for calculating sample size of orthogonal, two-level designs starting from a coded regression coefficient (Part 2)

```
%   RECURSIVELY SEARCH FOR MATCH
for step = 1
    startnt = (k+2);
    endnt = (k+2)*1000;
    for nt = startnt:step:endnt
        nu1 = nt-1-p; %Degrees of freedom for error
        phi = beta/sigma*sqrt(nt); %Non-centrality measure
        criticalHigh = tinv(1-alpha/2,nu1); %Critical t-value t*
        criticalLow = -tinv(1-alpha/2,nu1); %Critical t-value t*
        powerC =  (nctcdf(criticalLow,nu1,phi))+ (1-nctcdf(
            criticalHigh,nu1,phi));
        if(powerC >= power)
            endnt = nt;
            break;
        end
    end
end
%   DISPLAY SAMPLE SIZE AND POWER
disp(['Minimum sample size: ' num2str(nt) ', Power: ' num2str(
    powerC)])
```

Bibliography

[1] A. Abuhabaya, J. Fieldhouse, and D. Brown. "The optimization of biodiesel production by using response surface methodology and its effect on compression ignition engine". In: *Fuel Processing Technology* 113 (Sept. 2013), pp. 57–62. ISSN: 03783820. DOI: 10.1016/j.fuproc.2013.03.025.

[2] D. H. Ackley. *A connectionist machine for genetic hillclimbing*. Boston: Kluwer Academic Publishers, 1987.

[3] P. de Aguiar, B. Bourguignon, M. Khots, D. Massart, and R. Phan-Than-Luu. "D-optimal designs". In: *Chemometrics and Intelligent Laboratory Systems* 30.2 (Dec. 1995), pp. 199–210. ISSN: 01697439. DOI: 10.1016/0169-7439(94)00076-X.

[4] M. Alimahmoodi and C. N. Mulligan. "Optimization of the anaerobic treatment of a waste stream from an enhanced oil recovery process". In: *Bioresource technology* 102.2 (2011), pp. 690–696. ISSN: 1873-2976. DOI: 10.1016/j.biortech.2010.08.043.

[5] K. Andersen, G. Cook, G. Karsai, and K. Ramaswamy. "Artificial neural networks applied to arc welding process modeling and control". In: *IEEE Transactions on Industry Applications* 26.5 (1990), pp. 824–830. ISSN: 00939994. DOI: 10.1109/28.60056.

[6] A. C. Atkinson, A. N. Donev, and R. D. Tobias. *Optimum Experimental Design, with SAS*. Oxford: Oxford University Press, 2007, p. 478. ISBN: 978-0-19-929659-0.

[7] D. Bacon. "Application of EVOP and other optimization techniques to processing plants". In: *Canadian Mining and Metallurgical Society Bulletin* 60 (1967), pp. 1178–1183.

[8] D. Banabic. *Advanced Methods in Material Forming*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007. ISBN: 978-3-540-69844-9. DOI: 10.1007/3-540-69845-0.

[9] R. Banerjee and B. Bhattacharyya. “Evolutionary operation (EVOP) to optimize protease biosynthesis by Rhizopus oryzae”. In: *Bioprocess Engineering* 8.3-4 (1992), pp. 151–155.

[10] R. Banerjee and B. C. Bhattacharyya. “Evolutionary Operation (EVOP) to Optimize Three-Dimensional Biological Experiments”. In: *Biotechnology and Bioengineering* 41.1 (1993), pp. 67–71.

[11] R. Banerjee and B. Bhattacharyya. “Evolutionary operation as a tool of optimization for solid state fermentation”. In: *Biochemical Engineering Journal* 13.2 (2003), pp. 149–155. ISSN: 1369703X. DOI: 10.1016/S1369-703X(02)00127-4.

[12] S. B. Bankar and R. S. Singhal. “Optimization of poly-epsilon-lysine production by Streptomyces noursei NRRL 5126”. In: *Bioresource technology* 101.21 (2010), pp. 8370–8375. ISSN: 1873-2976. DOI: 10.1016/j.biortech.2010.06.004.

[13] E. H. Barnett. “Introduction to evolutionary operation”. In: *Industrial & Engineering Chemistry* 52.6 (1960), pp. 500–503. DOI: 10.1021/ie50606a026.

[14] R. R. Barton and J. S. J. Ivey. “Modifications of the Nelder-Mead simplex method for stochastic simulation response optimization”. In: *Proceedings of the 1991 Winter Simulation Conference*. 1991, pp. 945–953.

[15] J. C. Berridge. “Simplex optimization of high-performance liquid chromatographic separations”. In: *Journal of chromatography* 485 (1989), pp. 3–14.

[16] G. S. Beveridge and R. S. Schechter. *Optimization: Theory and Practice*. McGraw-Hill, Inc., 1970. ISBN: 978-0-070-05128-7.

[17] R. Boddy and G. Smith. *Effective Experimentation For Scientists and Technologists*. John Wiley & Sons, 2010, p. 256. ISBN: 978-0-470-68460-3.

[18] S. Boumaza, F. Kaouah, T. Berrama, M. Trari, and Z. Bendjama. “Optimization of preparation conditions for activated carbon from waste materials of agricultural origin for the removal of basic red 46”. In: *Chemometrics and Intelligent Laboratory Systems* 118 (Aug. 2012), pp. 311–316. ISSN: 01697439. DOI: 10.1016/j.chemolab.2012.05.002.

[19] G. Box. “Evolutionary operation: A method for increasing industrial productivity”. In: *Applied Statistics* 6.2 (1957), pp. 81–101. URL: http://www.jstor.org/stable/10.2307/2985505.

[20] G. Box and N. Draper. *Evolutionary Operation - A Statistical Method for Process Improvement*. New York: Wiley Classics Library, 1998. ISBN: 978-0-471-25551-2.

[21] G. Box and N. Draper. “Isn’t My Process Too Variable for EVOP?” In: *Technometrics* 10.3 (1968), pp. 439–444. URL: http://www.jstor.org/stable/10.2307/1267100.

[22] G. Box and J. Hunter. “Condensed calculations for evolutionary operation programs”. In: *Technometrics* 1.1 (1959), pp. 77–95. URL: http://www.jstor.org/stable/10.2307/1266311.

[23] G. Box and A. Luceño. “Discrete proportional-integral control with constrained adjustment”. In: *Journal of the Royal Statistical Society. Series D (The Statistician)* 44.4 (1995), pp. 479–495. URL: http://www.jstor.org/stable/2348897.

[24] P. Boyle. “Gaussian processes for regression and optimisation”. PhD thesis. Victoria University of Wellington, 2007, p. 190. URL: http://researcharchive.vuw.ac.nz/handle/10063/421.

[25] J.-H. Byun and Y. B. Kim. “Use of Evolutionary Operation for Optimizing Multiple Performance Characteristics in Large Scale Manufacturing”. In: *International Journal of Industrial Engineering* 10.4 (2003), pp. 490–496.

[26] S. L. Cannon. “Evolutionary Operation Applied to the Optimization of a Textile Heat Transfer Unit”. In: *Transactions of the Transactions of the Institution of Chemical Engineers And The Chemical Engineer* 50.3 (1972), pp. 283–285.

[27] G. Carrillo-Ureta. “Genetic algorithms for optimal control of beer fermentation”. In: *Proceedings of the 2001 IEEE International Symposium on Intelligent Control*. 2001, pp. 391–396. DOI: 10.1109/ISIC.2001.971541.

[28] C. Castro, C. António, and L. Sousa. “Optimisation of shape and process parameters in metal forging using genetic algorithms”. In: *Journal of Materials Processing Technology* 146.3 (Mar. 2004), pp. 356–364. ISSN: 09240136. DOI: 10.1016/j.jmatprotec.2003.11.027.

[29] T. Chen and J. Ren. “Bagging for Gaussian process regression”. In: *Neurocomputing* 72.7-9 (Mar. 2009), pp. 1605–1610. ISSN: 09252312. DOI: 10.1016/j.neucom.2008.09.002.

[30] D. Cheong. “Production of Bio-Hydrogen by Mesophilic Anaerobic Fermentation in an Acid-Pphase Sequencing Batch Reactor”. In: *Biotechnology and Bioengineering* 96.3 (2007), pp. 421–432. DOI: 10.1002/bit.

[31] U.-K. Choi, M.-H. Kim, D.-J. Kwon, O.-J. Kwon, and N.-H. Lee. "Optimization of Extraction Conditions for the Antibacterial Activity by Clove against Streptococcus mutans Using Evolutionary Operation-Factorial Design Technique". In: *Food Science and Biotechnology* 17.5 (2008), pp. 899–903.

[32] U.-K. Choi, M.-H. Kim, and N.-H. Lee. "Optimization of antibacterial activity by Gold-Thread (Coptidis Rhizoma Franch) against Streptococcus mutans using evolutionary operation-factorial design technique". In: *Journal of microbiology and Biotech* 17.11 (2007), pp. 1880–1884.

[33] U.-K. Choi, O.-H. Lee, S.-I. Lim, and Y.-C. Kim. "Optimization of Antibacterial Activity of Perilla frutescens var. acuta Leaf against Pseudomonas aeruginosa Using the Evolutionary Operation-Factorial Design Technique". In: *International journal of molecular sciences* 11.10 (Jan. 2010), pp. 3922–3932. ISSN: 1422-0067. DOI: 10.3390/ijms11103922.

[34] Y.-H. Chu, I.-S. Han, and C. Han. "Improved evolutionary operation based on D-optimal design and response surface method". In: *Korean Journal of Chemical Engineering* 19.4 (July 2002), pp. 535–544. ISSN: 0256-1115. DOI: 10.1007/BF02699292.

[35] R. Colombo, T. C. R. Ferreira, S. a. Alves, R. L. Carneiro, and M. R. V. Lanza. "Application of the response surface and desirability design to the Lambda-cyhalothrin degradation using photo-Fenton reaction". In: *Journal of environmental management* 118 (Mar. 2013), pp. 32–9. ISSN: 1095-8630. DOI: 10.1016/j.jenvman.2012.12.035.

[36] C. Curt, I. Allais, N. Perrot, V. Leblanc, and G. Trystram. "Optimisation of the meat emulsification process using at-line human evaluations and the Simplex method". In: *Journal of Food Engineering* 64.64 (2004), pp. 33–41. DOI: 10.1016/j.jfoodeng.2003.09.010.

[37] T. Danzer and G. Schwedt. "Chemometric methods for the development of a biosensor system and the evaluation of inhibition studies with solutions and mixtures of pesticides and heavy metals Part 1. Development of an enzyme electrodes system for pesticide and heavy metal screening us". In: *Analytica Chimica Acta* 318.3 (Jan. 1996), pp. 275–286. ISSN: 00032670. DOI: 10.1016/0003-2670(95)00460-2.

[38] M. Das, R. Banerjee, and S. Bal. "Multivariable parameter optimization for the endoglucanase production by Trichoderma reesei Rut C30 from Ocimum gratissimum seed". In: *Brazilian Archives of Biology and Technology* 51.1 (2008), pp. 35–41.

[39] T. De Beer, A. Burggraeve, M. Fonteyne, L. Saerens, J. Remon, and C. Vervaet. "Near infrared and Raman spectroscopy for the in-process monitoring of pharmaceutical production processes." In: *International journal of pharmaceutics* 417.1-2 (Sept. 2011), pp. 32–47. ISSN: 1873-3476. DOI: 10.1016/j.ijpharm.2010.12.012.

[42] B. De Ketelaere, P. Goos, and K. Brijs. "Prespecified factor level combinations in the optimal design of mixture-process variable experiments". In: *Food Quality and Preference* 22.7 (2011), pp. 661–670. ISSN: 0950-3293. DOI: 10.1016/j.foodqual.2011.04.006.

[43] B. Dejaegher and Y. Van der Heyden. "Sequential Optimization Methods". In: *Comprehensive Chemometrics*. Ed. by B. Walczak, S. D. Brown, and T. Romá. Vol. 1. Elsevier, 2009. Chap. 17, pp. 547–575. ISBN: 9780444527011. DOI: 10.1016/B978-044452701-1.00036-3.

[44] S. N. Deming, L. R. J. Parker, and M. B. Denton. "A review of simplex optimization in analytical chemistry". In: *CRC Crticial Reviews in Analytical Chemistry* 7.3 (1978), pp. 187–202. DOI: 10 . 1080 / 10408347808542701.

[45] G. Derringer and R. Suich. "Simultaneous optimization of several response variables". In: *Journal of Quality Technology* 12.4 (1980), pp. 214–219.

[46] G. Derringer. "A Balancing Act Optimizing A Product's Properties". In: *Quality Progress* 27.4 (1994), pp. 51–58.

[47] D. Dodoo and M. Vrchlabsky. "Optimization Of The Silver Diethyldithiocarbamate Spectrophotometric Method For The Determination Of Arsenic In Tap Waters". In: *Chemia Analityczna* 26.5 (1981), pp. 867–876.

[48] C. W. Dunnett. "New tables for multiple comparisons with a control". In: *Biometrics* 20.3 (1964), pp. 482–491. URL: http://www.jstor.org/stable/10.2307/2528490.

[49] B. Farshi, S. Gheshmi, and E. Miandoabchi. "Optimization of injection molding process parameters using sequential simplex algorithm". In: *Materials & Design* 32.1 (Jan. 2011), pp. 414–423. ISSN: 02613069. DOI: 10.1016/j.matdes.2010.06.043.

[50] V. V. Fedorov. *Theory of Optimal Experiments*. New York: Academic Press, 1972. ISBN: 978-0-323-16246-3.

[51] L. Ferey, N. Delaunay, D. N. Rutledge, A. Huertas, Y. Raoul, P. Gareil, and J. Vial. "Use of response surface methodology to optimize the simultaneous separation of eight polycyclic aromatic hydrocarbons by capillary zone electrophoresis with laser-induced fluorescence detection". In: *Journal of chromatography. A* 1302 (Aug. 2013), pp. 181–90. ISSN: 1873-3778. DOI: 10.1016/j.chroma.2013.06.027.

[52] D. J. Finney. "The Fractional Replication of Factorial Experiments". In: *Annals of Eugenics* 12.1 (1943), pp. 291–301.

[53] R. A. Fisher. "The arrangement of field experiments". In: *Journal of the Ministry of Agriculture of Great Britain* 33 (1926), pp. 503–513.

[54] J. D. Gibbons and S. Chakraborti. *Nonparametric Statistical Inference*. 4th ed. New York: Marcel Dekker, Inc., May 2003, p. 645. ISBN: 0824740521.

[55] C. Gontarski and P. Rodrigues. "Simulation of an industrial wastewater treatment plant using artificial neural networks". In: *Computers & Chemical Engineering* 24.2 (2000), pp. 1719–1723. DOI: 10.1016/S0098-1354(00)00449-X.

[56] J. González-Ramírez, D. Leducq, M. Arellano, and G. Alvarez. "Energy consumption optimization of a continuous ice cream process". In: *Energy Conversion and Management* 70 (June 2013), pp. 230–238. ISSN: 01968904. DOI: 10.1016/j.enconman.2013.03.015.

[57] P. Goos. *Verklarende statistiek - Schatten en toetsen*. Acco, 2014. ISBN: 978-9-033-49533-5.

[58] P. Goos and B. Jones. *Optimal Design of Experiments: A Case Study Approach*. John Wiley & Sons, 2011, p. 304. ISBN: 978-0-470-74461-1.

[59] P. Goos and D. Meintrup. *Statistics with JMP: Graphs, Descriptive Statistics and Probability*. 1st ed. Wiley, May 2015.

[60] P. J. Grabner and H. Prodinger. "Maximum Statistics of N Random Variables Distributed by the Negative Binomial Distribution". In: *Combinatorics, Probability and Computing* 6 (1997), pp. 179–183.

[61] R. Grbić, D. Slišković, and P. Kadlec. "Adaptive soft sensor for online prediction and process monitoring based on a mixture of Gaussian process models". In: *Computers & Chemical Engineering* 58 (Nov. 2013), pp. 84–97. ISSN: 00981354. DOI: 10.1016/j.compchemeng.2013.06.014.

[62] G. J. Hahn. "Process improvement through simplex EVOP". In: *Triangle* 2.210 (1976), p. 30.

[63] G. Hahn. "Process improvement using evolutionary operation". In: *Chemtech* 6.3 (1976), pp. 204–206.

[64] G. Hahn and A. Dershowitz. "Evolutionary Operation To-day-Some Survey Results and Observations". In: *Journal of the Royal Statistical Society. Series C* 23.2 (1974), pp. 214–218.

[65] D. S. Holmes. "Model-free optimization in cement plants". In: *Cement Industry Technical Conference, 2003. Conference Record. IEEE-IAS/PCA 2003*. IEEE. Dallas, 2003, pp. 159–173. ISBN: 0780379004. DOI: 10.1109/CITCON.2003.1204717.

[66] D. S. Holmes and A. Erhan Mergen. "Identifying Need for New Factors in EVOP". In: *Quality Engineering* 18.2 (July 2006), pp. 219–224. ISSN: 0898-2112. DOI: 10.1080/08982110600567558.

[67] D. S. Holmes and P. M. Holmes. "Method, System and Article of Manufacture for Operating a Process". English. US 6453207 B1. Sept. 17, 2002.

[68] E. Hopper and B. Turton. "A genetic algorithm for a 2D industrial packing problem". In: *Computers & Industrial Engineering* 37.1 (1999), pp. 375–378.

[69] E. Hulthén and C. Magnus Evertsson. "Real-time algorithm for cone crusher control with two variables". In: *Minerals Engineering* 24.9 (Aug. 2011), pp. 987–994. ISSN: 08926875. DOI: 10.1016/j.mineng.2011.04.007.

[70] W. Hunter and E. Chacko. "Increasing industrial productivity in developing countries". In: *Industrial Development Review* 13.3 (1971), pp. 11–16.

[71] W. Hunter and J. Kittrell. "Evolutionary operation: a review". In: *Technometrics* 8.3 (1966), pp. 389–397. URL: http://www.jstor.org/stable/10.2307/1266686.

[72] W. Jacobs, R. Boonen, P. Sas, and D. Moens. "The effect of external dynamic loads on the lifetime of rolling element bearings: accurate measurement of the bearing behaviour". In: *Journal of Physics: Conference Series* 364.1 (May 2012). ISSN: 1742-6596. DOI: 10.1088/1742-6596/364/1/012037.

[73] C.-H. Jen and B. C. Jiang. "Combining on-line experiment and process control methods for changes in a dynamic model". In: *International Journal of Production Research* 46.13 (July 2008), pp. 3665–3682. ISSN: 0020-7543. DOI: 10.1080/00207540601120476.

[74] B. Jones and R. Johnson. "Design and analysis for the Gaussian process model". In: *Quality and Reliability Engineering International* 25.5 (July 2009), pp. 515–524. DOI: 10.1002/qre.

[75] D. R. Jones, M. Schonlau, and W. J. Welch. "Efficient global optimization of expensive black-box functions". In: *Journal of Global optimization* 13.4 (1998), pp. 455–492.

[76] K. Kanlayasiri and P. Jattakul. "Simultaneous optimization of dimensional accuracy and surface roughness for finishing cut of wire-EDMed K460 tool steel". In: *Precision Engineering* 37.3 (July 2013), pp. 556–561. ISSN: 01416359. DOI: 10.1016/j.precisioneng.2012.12.009.

[77] B. Kar, R. Banerjee, and B. C. Bhattacharyya. “Optimization of physicochemical parameters for gallic acid production by evolutionary operation-factorial design technique”. In: *Process Biochemistry* 37.12 (July 2002), pp. 1395–1401. ISSN: 13595113. DOI: 10.1016/S0032-9592(02)00020-1.

[78] I. C. Kenworthy. “Some examples of simplex evolutionary operation in the paper industry”. In: *Applied Statistics* 16.3 (1967), pp. 211–224. URL: http://www.jstor.org/stable/10.2307/2985918.

[79] D.-H. Kim, Y.-C. Kim, and U.-K. Choi. “Optimization of Antibacterial Activity of Perilla frutescens var. acuta Leaf against Staphylococcus aureus Using Evolutionary Operation Factorial Design Technique”. In: *International journal of molecular sciences* 12.4 (Jan. 2011), pp. 2395–407. ISSN: 1422-0067. DOI: 10.3390/ijms12042395.

[80] J. R. Koehler and A. B. Owen. “Computer experiments”. In: *Handbook of statistics, 13: Design and Analysis of Experiments*. Ed. by S. Ghosh and C. R. Rao. Vol. 13. Amsterdam: Elsevier, 1996, pp. 261–308.

[81] S. Kumar, N. Katiyar, P. Ingle, and S. Negi. “Use of evolutionary operation (EVOP) factorial design technique to develop a bioprocess using grease waste as a substrate for lipase production”. In: *Bioresource technology* 102.7 (Apr. 2011), pp. 4909–12. ISSN: 1873-2976. DOI: 10.1016/j.biortech.2010.12.114.

[82] M. Kutner, C. Nachtsheim, J. Neter, and W. Li. *Applied Linear Statistical Models*. 5th ed. McGraw-Hill/Irwin, 2004. ISBN: 978-0-073-10874-2.

[83] T. Kvist and P. Thyregod. “Using Evolutionary Operation to Improve Yield in Biotechnological Processes”. In: *Quality and Reliability Engineering International* 21.5 (Aug. 2005), pp. 457–463. ISSN: 0748-8017. DOI: 10.1002/qre.733.

[84] R. M. Lewis and V. Torczon. “Pattern Search Methods for Linearly Constrained Minimization”. In: *SIAM Journal on Optimization* 10.3 (Jan. 2000), pp. 917–941. ISSN: 1052-6234. DOI: 10.1137/S1052623497331373.

[85] J. Li, C. Wassgren, and J. Litster. “Multi-scale modeling of a spray coating process in a paddle mixer/coater: the effect of particle size distribution on particle segregation and coating uniformity”. In: *Chemical Engineering Science* 95 (2013), pp. 203–210. ISSN: 0009-2509. DOI: 10.1016/j.ces.2013.03.014.

[86] Y. B. Lim and S. H. Park. “Number of Cycles in Evolutionary Operation”. In: *Journal of the Korean Statistical Society* 36.2 (2007), pp. 201–208.

[87] P. Liu, R. Yang, and A. Yu. “DEM study of the transverse mixing of wet particles in rotating drums”. In: *Chemical Engineering Science* 86 (2013), pp. 99–107. DOI: 10.1016/j.ces.2012.06.015.

[88] J. L. Loeppky, J. Sacks, and W. Welch. “Choosing the Sample Size of a Computer Experiment: A practical guide”. In: *Technometrics* 51.4 (2009), pp. 366–376.

[89] D. E. Long. “Simplex optimization of the response from chemical systems”. In: *Analytica Chimica Acta* 46.2 (1969), pp. 193–206.

[90] C. W. Lowe. “Evolutionary operation in action”. In: *Applied Statistics* 23.2 (1974), pp. 218–226. URL: http://www.jstor.org/stable/10.2307/2347005.

[91] C. W. Lowe. “Some techniques of evolutionary operation”. In: *Transactions of the Institution of Chemical Engineers* 42 (1964), pp. 332–344.

[92] P. Mahapatra, A. Kumari, V. K. Garlapati, R. Banerjee, and A. Nag. “Optimization of Process Variables for Lipase Biosynthesis from Rhizopus oligosporus NRRL 5905 Using Evolutionary Operation Factorial Design Technique”. In: *Indian Journal of Microbiology* 50.4 (Oct. 2010), pp. 396–403. ISSN: 0973-7715. DOI: 10.1007/s12088-011-0071-z.

[93] F. Mantzouridou, A. Spanou, and V. Kiosseoglou. “An inulin-based dressing emulsion as a potential probiotic food carrier”. In: *Food Research International* 46.1 (Apr. 2012), pp. 260–269. ISSN: 09639969. DOI: 10.1016/j.foodres.2011.12.016.

[94] G. Matheron. “Principles of geostatistics”. In: *Economic geology* 58.8 (1963), pp. 1246–1266.

[95] P. Mathews. *Sample Size Calculations - Practical Methods for Engineers and Scientists*. Mathews Malnar and Bailey, Inc., 2010. ISBN: 978-0-615-32461-6.

[96] D. Mathieu, E. Puech-Costes, M.-T. Maurette, and R. P. Tan Luu. “The simplex method applied to the detection and following of a discontinuity”. In: *Chemometrics and Intelligent Laboratory Systems* 20.1 (Aug. 1993), pp. 25–34. ISSN: 01697439. DOI: 10.1016/0169-7439(93)80018-D.

[97] A. Mathur, K. Pattipati, and A. Song. “Robust local experiments in sequential process improvement”. In: *Proceedings of IEEE International Conference on Systems, Man and Cybernetics* 3 (1994), pp. 2566–2571. DOI: 10.1109/ICSMC.1994.400257.

[98] R. J. McDevitt and B. J. Barker. “Simplex optimization of the synergic extraction of a bis-diketo copper (II) complex”. In: *Analytica Chimica Acta* 122.2 (1980), pp. 223–226.

[99] M. D. McKay, R. J. Beckman, and W. J. Conover. “A comparison of three methods for selecting values of input variables in the analysis of output from a computer code”. In: *Technometrics* 21.2 (1979), pp. 239–245.

[100] R. W. Mee. *Factorial Two-Level Experimentation*. Springer, 2009. ISBN: 978-0-387-89102-6.

[101] M. Meireles. "A comprehensive review for industrial applicability of artificial neural networks". In: *IEEE Transactions on Industrial Electronics* 50.3 (2003), pp. 585–601. DOI: 10.1109/TIE.2003.812470.

[102] R. K. Meyer and C. J. Nachtsheim. "The coordinate-exchange algorithm for constructing exact optimal experimental designs". In: *Technometrics* 37.1 (1995), pp. 60–69.

[103] A. d. C. P. Monteiro, L. de Andrade, and R. de Campos. "On-line mercury and methylmercury pre-concentration by adsorption of their dithiophosphoric acid diacyl ester chelates on a C 18 column and cold-vapor atomic-absorption detection". In: *Fresenius' Journal of Analytical Chemistry* 371.3 (Oct. 2001), pp. 353–357. ISSN: 0937-0633. DOI: 10.1007/s002160100949.

[104] D. C. Montgomery. *Design and Analysis of Experiments*. 8th ed. Wiley, 2012. ISBN: 978-0-470-39882-1.

[105] M. H. Muhamad, S. R. Sheikh Abdullah, A. B. Mohamad, R. Abdul Rahman, and A. A. Hasan Kadhum. "Application of response surface methodology (RSM) for optimisation of COD, NH3-N and 2,4-DCP removal from recycled paper wastewater in a pilot-scale granular activated carbon sequencing batch biofilm reactor (GAC-SBBR)". In: *Journal of environmental management* 121 (May 2013), pp. 179–90. ISSN: 1095-8630. DOI: 10.1016/j.jenvman.2013.02.016.

[106] G. Mukherjee and R. Banerjee. "Evolutionary operation-factorial design technique for optimization of conversion of mixed agroproducts into gallic acid". In: *Applied biochemistry and biotechnology* 118.1-3 (2004), pp. 33–46. ISSN: 0273-2289.

[107] R. H. Myers, D. C. Montgomery, and C. M. Anderson-Cook. *Response Surface Methodology: Process and Product Optimization Using Designed Experiments*. 3rd ed. Hoboken, New Jersey: Wiley, 2009. ISBN: 0-470-17446-3.

[108] S. Negi and R. Banerjee. "Optimization of Amylase and Protease Production from Aspergillus awamori in Single Bioreactor Through EVOP Factorial Design Technique". In: *Food Technology and Biotechnology* 44.2 (2006), pp. 257–261.

[109] J. Nelder and R. Mead. "A simplex method for function minimization". In: *The computer journal* 7.4 (1965), pp. 308–313.

[110] T. Öberg. "Importance of the first design matrix in experimental simplex optimization". In: *Chemometrics and intelligent laboratory systems* 44.1 (1998), pp. 11–13.

[111] T. Öberg and S. Deming. “Find optimum operating conditions fast”. In: *Chemical engineering progress* 96.4 (2000), p. 53.

[112] G. Oehlert. *A First Course in Design and Analysis of Experiments*. New York: W. H. Freeman and Company, 2000. ISBN: 0-7167-3510-5.

[113] A. O’Hagan and J. F. C. Kingman. “Curve fitting and optimal design for prediction”. In: *Journal of the Royal Statistical Society. Series B* 40.1 (1978), pp. 1–42. URL: http://www.jstor.org/stable/2984861.

[114] C. Onnen, R. Babuška, U. Kaymak, J. Sousa, H. Verbruggen, and R. Isermann. “Genetic algorithms for optimization in predictive control”. In: *Control Engineering Practice* 5.10 (Oct. 1997), pp. 1363–1372. ISSN: 09670661. DOI: 10.1016/S0967-0661(97)00133-0.

[115] B. P. Panda, M. Ali, and S. Javed. “Fermentation Process Optimization”. In: *Research Journal of Microbiology* 2.3 (2007), pp. 201–208.

[116] B. P. Panda, R. Mazumder, and R. Banerjee. “Statistical optimization of process parameters influencing the biotransformation of plant tannin into gallic acid under solid-liquid fermentation”. In: *Journal of Pharmacy And Bioallied Sciences* 1.1 (2009), pp. 37–42. ISSN: 0975-7406. DOI: 10.4103/0975-7406.62685.

[117] R. C. Pansuriya. “Evolutionary Operation (EVOP) to Optimize Whey-Independent Serratiopeptidase Production from Serratia marcescens NRRL B-23112”. In: *Journal of Microbiology and Biotechnology* 20.5 (May 2010), pp. 950–957. ISSN: 10177825. DOI: 10.4014/jmb.0911.11023.

[118] A. Pasamontes and P. Callao. “Fractional factorial design and simplex algorithm for optimizing sequential injection analysis (SIA) and second order calibration”. In: *Chemometrics and Intelligent Laboratory Systems* 83.2 (Sept. 2006), pp. 127–132. ISSN: 01697439. DOI: 10.1016/j.chemolab.2005.10.007.

[119] F. M. Pelissari, M. M. Andrade-Mahecha, P. J. D. A. Sobral, and F. C. Menegalli. “Optimization of process conditions for the production of films based on the flour from plantain bananas (Musa paradisiaca)”. In: *LWT - Food Science and Technology* 52.1 (June 2013), pp. 1–11. ISSN: 00236438. DOI: 10.1016/j.lwt.2013.01.011.

[120] R. L. Plackett and J. P. Burman. “The design of optimum multifactorial experiments”. In: *Biometrika* 33.4 (1946), pp. 305–325. URL: http://www.jstor.org/stable/2332195.

[121] B. D. Prater, S. C. Tuller, and L. J. Wilson. “Simplex optimization of protein crystallization conditions”. In: *Journal of Crystal Growth* 196.2-4 (Jan. 1999), pp. 674–684. ISSN: 00220248. DOI: 10.1016/S0022-0248(98)00862-8.

[122] J. Ramberg. *Development and Adaptation of A Control System for Optimization of Single and Multiple Operation Machining*. Tech. rep. Iowa: Intertech Corporation, 1974.

[123] P. Ranjan, R. Haynes, and R. Karsten. "A computationally stable approach to Gaussian process interpolation of deterministic computer simulation data". In: *Technometrics* 53.4 (2011), pp. 366–378. DOI: 10.1198/TECH.2011.09141.

[124] A. D. Rickmers. "Evolutionary Operations (EVOP)". In: *Journal of the Society of Motion Picture and Television Engineers* 75.7 (1966), pp. 661–665.

[125] M. Routh, P. Swartz, and M. Denton. "Performance of the super modified simplex". In: *Analytical Chemistry* 49.9 (1977), pp. 1422–1428. DOI: 10.1021/ac50017a033.

[126] C. Rozycki. "Application of the simplex method for optimization of the analytical methods". In: *Chemia Analityczna* 38 (1993), pp. 681–698.

[127] K. Rutten, J. De Baerdemaeker, and B. De Ketelaere. "Choosing An Appropriate Starting Point For Sequential Improvement Methods When No Prior Information is Available". In: *Applied Stochastic Models In Business and Industry* (2015).

[136] T. P. Ryan. "Evolutionary Operation". In: *Statistical Methods for Quality Improvement*. 1957. John Wiley & Sons, 2011. Chap. 15, pp. 565–585.

[137] J. Sacks, W. J. Welch, T. J. Mitchell, and H. P. Wynn. "Design and analysis of computer experiments". In: *Statistical science* 4.4 (1989), pp. 409–423.

[138] T. J. Santner, B. J. Williams, and W. I. Notz. *The Design and Analysis of Computer Experiments*. New York: Springer-Verlag, 2003. ISBN: 978-0-387-95420-2.

[139] M. J. Santos-Delgado, E. Crespo-Corral, and L. M. Polo-Díez. "Determination of herbicides in soil samples by gas chromatography: optimization by the simplex method". In: *Talanta* 53.2 (Nov. 2000), pp. 367–77. ISSN: 1873-3573.

[140] H. S. Shin and H. J. Cha. "Facile and statistical optimization of transfection conditions for secretion of foreign proteins from insect Drosophila S2 cells using green fluorescent protein reporter". In: *Biotechnology progress* 18.6 (2002), pp. 1187–94. ISSN: 8756-7938. DOI: 10.1021/bp0255331.

[141] P. I. Silva, P. C. Stringheta, R. F. Teófilo, and I. R. N. de Oliveira. "Parameter optimization for spray-drying microencapsulation of jaboticaba (Myrciaria jaboticaba) peel extracts using simultaneous analysis of responses". In: *Journal of Food Engineering* 117.4 (Aug. 2013), pp. 538–544. ISSN: 02608774. DOI: 10.1016/j.jfoodeng.2012.08.039.

[142] J. L. Smart. "Using the Evolutionary Operation Method To Optimize Gas Absorber Operation". In: *Chemical Engineering Education* 38.3 (2004), pp. 204–211.

[143] W. Spendley, G. Hext, and F. Himsworth. "Sequential application of simplex designs in optimisation and evolutionary operation". In: *Technometrics* 4.4 (1962), pp. 441–461. URL: http://www.jstor.org/stable/10.2307/1266283.

[144] S. Srijaranai, R. Burakham, R. L. Deming, and T. Khammeng. "Simplex optimization of ion-pair reversed-phase high performance liquid chromatographic analysis of some heavy metals". In: *Talanta* 56.4 (Mar. 2002), pp. 655–61. ISSN: 1873-3573.

[145] J. Stoev, A. Bartic, S. Gillijns, and W. Symens. "Badminton playing robot-a multidisciplinary test case in mechatronics". In: *Mechatronic Systems*. Marriott Boston Cambridge, USA, 2010. DOI: 10.3182/20100913-3-US-2015.00028.

[146] J. Stoev, G. Pinte, and W. Symens. "Time-constrained energy-optimal motion control - Application to a Badminton Robot". In: *The 13th Mechatronics Forum International Conference*. 2012.

[147] Z.-g. Su, P.-h. Wang, X.-j. Yu, and Z.-z. Lv. "Experimental investigation of vibration signal of an industrial tubular ball mill: Monitoring and diagnosing". In: *Minerals Engineering* 21.10 (Sept. 2008), pp. 699–710. ISSN: 08926875. DOI: 10.1016/j.mineng.2008.01.009.

[148] Q. Tang, Y. B. Lau, S. Hu, W. Yan, Y. Yang, and T. Chen. "Response surface methodology using Gaussian processes: Towards optimizing the trans-stilbene epoxidation over Co^{2+}-NaX catalysts". In: *Chemical Engineering Journal* 156.2 (Jan. 2010), pp. 423–431. ISSN: 13858947. DOI: 10.1016/j.cej.2009.11.002.

[149] V. Torczon and M. W. Trosset. "From evolutionary operation to parallel direct search: Pattern search algorithms for numerical optimization". In: *Computing Science and Statistics* 29 (1998), pp. 396–401.

[150] G. Toschkoff, S. Just, A. Funke, and D. Djuric. "Spray models for discrete element simulations of particle coating processes". In: *Chemical Engineering Science* 101 (2013), pp. 603–614. ISSN: 0009-2509. DOI: 10.1016/j.ces.2013.06.051.

[151] R. Tunga, R. Banerjee, and B. C. Bhattacharyya. “Optimization of n variable biological experiments by evolutionary operation-factorial design technique”. In: *Journal of bioscience and bioengineering* 87.2 (Jan. 1999), pp. 224–30. ISSN: 1389-1723. DOI: 10.1016/S1389-1723(99)89017-3.

[153] L. Van den Broeck, M. Diehl, and J. Swevers. “A model predictive control approach for time optimal point-to-point motion control”. In: *Mechatronics* 21.7 (Oct. 2011), pp. 1203–1212. ISSN: 09574158. DOI: 10.1016/j.mechatronics.2011.07.008.

[154] F. Walters. “Sequential Simplex Optimization - An Update”. In: *Analytical Letters* 32.2 (Jan. 1999), pp. 193–212. ISSN: 0003-2719. DOI: 10.1080/00032719908542815.

[155] F. H. Walters, L. R. J. Parker, S. L. Morgen, and S. N. Deming. *Sequential Simplex Optimization*. CRC Press LLC, 1991.

[156] X. Wang, J. Stoev, G. Pinte, and J. Swevers. “Classical and modern methods for time-constrained energy optimal motion - Application to a badminton robot”. In: *Mechatronics* 23.6 (2013), pp. 669–676. ISSN: 0957-4158. DOI: 10.1016/j.mechatronics.2013.07.001.

[157] C. K. I. Williams and C. E. Rasmussen. “Gaussian processes for regression”. In: *Advances in Neural Information Processing Systems 8*. Ed. by D. S. Touretzky, M. C. Mozer, and M. E. Hasselmo. MIT, 1996, pp. 514–520.

[158] M. L. Workman. “Adaptive proximate time-optimal servomechanisms”. PhD thesis. Stanford University, 1987.

[159] J. Wu and M. Hamada. *Experiments: Planning, Analysis, and Parameter Design Optimization*. Wiley-Interscience, 2000. ISBN: 0-471-25511-4.

[160] Q. Xiong and A. Jutan. “Continuous optimization using a dynamic simplex method”. In: *Chemical Engineering Science* 58.16 (Aug. 2003), pp. 3817–3828. ISSN: 00092509. DOI: 10.1016/S0009-2509(03)00236-7.

[161] F. Yates. “Sir Ronald Fisher and the design of experiments”. In: *Biometrics* 20.2 (1964), pp. 307–321. URL: http://www.jstor.org/stable/2528399.

[162] J. Yu. “Online quality prediction of nonlinear and non-Gaussian chemical processes with shifting dynamics using finite mixture model based Gaussian process regression approach”. In: *Chemical Engineering Science* 82 (Sept. 2012), pp. 22–30. ISSN: 00092509. DOI: 10.1016/j.ces.2012.07.018.

[163] U. Yüzgeç, M. Türker, and A. Hocalar. “On-line evolutionary optimization of an industrial fed-batch yeast fermentation process.” In: *ISA transactions* 48.1 (Jan. 2009), pp. 79–92. ISSN: 0019-0578. DOI: 10.1016/j.isatra.2008.09.001.

[164] B. Zelić and N. Pavlovic. “Optimization of pH and temperature in the process of bioconversion of glucose to 2,5-diketo-D-gluconic acid”. In: *Chemical and Bochemical Engineering* 16.1 (2002), pp. 7–11.

[165] J. Zhou and L. Turng. “Process optimization of injection molding using an adaptive surrogate model with Gaussian process approach”. In: *Polymer Engineering & Science* 47.5 (2007), pp. 684–694. DOI: 10.1002/pen.20741.

List Of Publications

Publications In Peer-reviewed Journals

K. Rutten, J. De Baerdemaeker, and B. De Ketelaere. "A Comparison of Evolutionary Operation And Simplex For Process Improvement". In: *Chemometrics and Intelligent Laboratory Systems* 139 (Dec. 2014), pp. 109–120. DOI: 10.1016/j.chemolab.2014.09.011

K. Rutten, J. De Baerdemaeker, J. Stoev, M. Witters, and B. De Ketelaere. "Constrained On-line Optimization Using Evolutionary Operation: A Case Study About Energy-Optimal Robot Control". In: *Quality and Reliability Engineering International* (2014). DOI: 10.1002/qre.1662

R. Van Beers, B. Aernouts, L. L. Gutiérrez, C. Erkinbaev, K. Rutten, A. Schenk, B. Nicolaï, and W. Saeys. "Optimal Illumination-Detection Distance and Detector Size for Predicting Braeburn Apple Maturity from Vis/NIR Laser Reflectance Measurements". In: *Food Bioprocess Technology* (2015). DOI: 10.1007/s11947-015-1562-4

K. Rutten, J. De Baerdemaeker, and B. De Ketelaere. "Choosing An Appropriate Starting Point For Sequential Improvement Methods When No Prior Information is Available". In: *Applied Stochastic Models In Business and Industry* (2015) *Publication in review process*

Papers At International Scientific Conferences & Symposia, Published In Full In Proceedings

B. De Ketelaere, C. Bravo, F. Mathijs, K. Rutten, and K. Mertens. "Advanced Multivariate Statistical Process Control (M-SPC) As Basis For Machine

Condition Monitoring". In: *Proceedings of the International Congress on Condition Monitoring and Diagnostics Engineering Management.* Vol. 13. Stavanger, Norway, May 2011

B. De Ketelaere and K. Rutten. "The Garden Sprinkler: An interactive web-based application for teaching design of experiments". In: *Proceedings of the 9th International Conference On Teaching Statistics.* International Association For Statistical Education (IASE) and International Statistical Institute (ISI). Flagstaff, Arizona, July 2014

Meeting Abstracts, Presented At International Scientific Conferences And Symposia, Published Or Not Published In Proceedings Or Journals

K. Rutten and B. De Ketelaere. "Comparison of On-line Design of Experiments Methods on Physical Models". In: *58th ISI World Statistics Congress.* International Statistical Institute (ISI). Dublin, Ireland, Aug. 2011

K. Rutten, J. De Baerdemaeker, and B. De Ketelaere. "Comparison of On-line Design of Experiments Methods on Physical Models". In: *11th Conference of the European Network of Business and Industrial Statistics.* European Network of Business and Industrial Statistics (ENBIS). Coimbra, Portugal, Sept. 2011

K. Rutten, J. De Baerdemaeker, and B. De Ketelaere. "Comparison of Different Forms of Sequential Designs for a Low and High Dimensional Case". In: *12th Conference of the European Network for Business and Industrial Statistics.* European Network of Business and Industrial Statistics. Ljubljana, Slovenia, Sept. 2012

K. Rutten, J. De Baerdemaeker, and B. De Ketelaere. "Simulation Comparison Of EVOP And Simplex-type Statistical Optimization Methods". In: *Spring Research Conference 2013.* Los Angeles, United States, June 2013

K. Rutten, J. De Baerdemaeker, and B. De Ketelaere. "The Garden Sprinkler: An Interactive Web-Based Application For Teaching Response Surface Methodology". In: *13th Conference of the European Network for Business and Industrial Statistics, Ankara Turkey.* European Network for Business and Industrial Statistics (ENBIS). Sept. 2013

K. Rutten, J. De Baerdemaeker, and B. De Ketelaere. "Constrained Online Improvement Using Evolutionary Operation". In: *14th Conference of the*

European Network for Business and Industrial Statistics. European Network for Business and Industrial Statistics (ENBIS). Linz, Austria, Sept. 2014

FACULTY OF BIOSCIENCE ENGINEERING
DEPARTMENT OF BIOSYSTEMS
MECHATRONICS, BIOSTATISTICS & SENSORS
Kasteelpark Arenberg 30, bus 2456
3001 LEUVEN, BELGIUM
koen.rutten@biw.kuleuven.be
http://www.mebios.be