B. Demeulenaere

M. Verschuure

J. Swevers

J. De Schutter

e-mail: joris.deschutter@mech.kuleuven.be

Department of Mechanical Engineering,
Katholieke Universiteit Leuven,
Leuven B-3001, Belgium

A General and Numerically Efficient Framework to Design Sector-Type and Cylindrical Counterweights for Balancing of Planar Linkages

This paper extends previous work concerning convex reformulations of counterweight balancing by developing a general and numerically efficient design framework for counterweight balancing of arbitrarily complex planar linkages. At the numerical core of the framework is an iterative procedure, in which successively solving three convex optimization problems yields practical counterweight shapes in typically less than 1 CPU s. Several types of counterweights can be handled. The iterative procedure allows minimizing and/or constraining shaking force, shaking moment, driving torque, and bearing forces. Numerical experiments demonstrate the numerical superiority (in terms of computation time and balancing result) of the presented framework compared to existing approaches. [DOI: 10.1115/1.4000532]

1 Introduction

High-speed linkages exert fluctuating inertia forces and moments on their supporting frame. Counterweight (CW) addition constitutes a classical approach to suppress the resulting frame vibration by reducing or eliminating the resultant force (*shaking force*) and resultant moment (*shaking moment*) exerted by the linkage on the frame.

The main challenge in counterweight design is to achieve a satisfactory balancing result (reduction in shaking force and/or shaking moment), while keeping in check the counterweight dimensions, as well as the increase in driving torque and/or bearing forces [1–5]. Most recent work in the area of balancing tackles this design endeavor through the use of numerical optimization techniques. In this respect, Demeulenaere et al. [6] showed previously that for planar crank-rocker four-bar mechanisms, designing point-mass counterweights can be reformulated as a convex nonlinear optimization problem (convex program). As opposed to nonconvex nonlinear optimization problems, which are generally difficult to solve due to the presence of local optima, the global optimum of convex programs is guaranteed to be found very efficiently [7]. Formulating design problems as convex programs therefore has great advantages, as numerically demonstrated for counterweight balancing in Ref. [6] and the subsequent work [8].

Verschuure et al. [8] extended the basic result [6] by showing that counterweight design can still be formulated as a convex program if the following additional aspects are considered: (i) direct minimization of frame vibration by considering a model of the supporting frame and its mounting and (ii) use of (practical) cylindrical counterweights instead of (theoretical) point-mass counterweights. Numerical results were provided for a crank-rocker four-bar mechanism supported by a rigid frame with three degrees of freedom, as in Ref. [9]. The advantage of this problem

formulation is the certainty that the optimal counterweights actually improve the machine performance in terms of frame vibration and force transmission to the machine floor [9]. However, a model of the frame and its mounting may not be available or difficult to obtain.

The present work extends [6,8] in the following respects.

- (i) A different, completely automatic procedure is developed in Sec. 3 which allows considering any force in the optimization problem (thus not only shaking force, shaking moment, or driving torque, as in Ref. [8]), which is relevant, since bearing forces may increase significantly due to balancing [10].
- (ii) Four extensions of the basic framework are developed in Sec. 4, giving rise to a fast procedure in which solving three instances of a convex program yields practical counterweight shapes: It is shown that also sector-type counterweights can be incorporated into the formulation as a convex program. A systematic and automatic procedure is presented to choose the attachment point of the counterweight, which also includes upper limits on counterweight inertia on counterweight radii and thicknesses.

The presented framework fills up a gap in literature, since only few results are known concerning optimal balancing of complex linkages. A systematic and numerically efficient methodology for optimal counterweight design (involving the essential trade-off [10,11] between the various dynamic forces and counterweight size) seems, to the best of our knowledge, to be nonexistent thus far.

To demonstrate the numerical efficiency of this approach, the framework is then applied in Sec. 5 to a planar ten-bar mechanism, previously considered by Smith [12], meanwhile showing the effect and relevance of the framework extensions and elaborating on the potential of partial force balance compared to full force balance.

In order to make this paper self-contained, Sec. 2 first reviews the relevant aspects of the *basic* convex optimization framework for counterweight balancing, developed in Refs. [6,8].

Contributed by the Mechanisms and Robotics Committee of ASME for publication in the JOURNAL OF MECHANICAL DESIGN. Manuscript received March 5, 2009; final manuscript received October 10, 2009; published online December 9, 2009. Editor: Panos Y. Papalambros.

2 Basic Convex Optimization Framework

This section reviews the choice of optimization variables (Sec. 2.1), objective function (Sec. 2.2), and constraints (Sec. 2.3) of the basic framework. For most mathematical proofs, the reader is referred to Refs. [6,8].

The framework is valid for N -bar planar linkages under the assumption that all forces can be determined through an inverse dynamic analysis. This implies that (i) all links are rigid, (ii) all joints are free of clearance, and (iii) the input crank is driven with a known (constant or fluctuating) speed [11].

2.1 Optimization Variables. The counterweight parametrization is the key aspect in reformulating counterweight design as a convex program. The particular parametrization chosen here describes a counterweight (attached to a moving link i) by its four independent mass parameters and enforces a particular shape through additional constraints. These four mass parameters are, however, not the classical mass parameters (mass m_i (kg), centroidal inertia J_i (kg m²), and COG coordinates (u_i, v_i) (m) in the link coordinate system (LCS)), but the following parameters used in Ref. [13] and denoted as the μ -parameters in Ref. [6]:

$$\mu_{1i} = m_i, \quad \mu_{2i} = m_i \cdot u_i, \quad \mu_{3i} = m_i \cdot v_i \quad (1a)$$

$$\mu_{4i} = J_i + m_i \cdot (u_i^2 + v_i^2) \quad (1b)$$

Compared to the classical mass parameter set $\{m_i, u_i, v_i, J_i\}$, the μ -parameter set $\{\mu_{1i}, \mu_{2i}, \mu_{3i}, \mu_{4i}\}$ has three key properties that are indispensable to obtain a convex optimization problem and which are detailed below.

Superposition. The μ -parameters are additive

$$\mu = \mu^o + \mu^* \quad (2)$$

where μ^o , μ^* , and $\mu \in \mathbb{R}^P$ group the μ -parameters of the original linkage, the counterweights, and the balanced linkage, respectively, and where $P=4 \cdot (N-1)$, that is, $P=36$ in the case of a ten-bar linkage.

Linearly dependent vectors. Any force or moment $d(t)$ obtained through an inverse dynamic analysis of a linkage can be written as an inner product of a time-dependent vector $\phi(t)$, solely depending on the mechanism kinematics, and a time-independent vector μ depending on the mechanism mass parameters

$$d(t) = \phi(t)^T \cdot \mu \quad (3)$$

as will be shown in Sec. 3. Expressions like Eq. (3) are very similar to the linearly independent vector expressions introduced by Berkof and Lowen [14]. The main difference, however, is that the elements of $\phi(t)$ are *not* linearly independent. Hence the name coined for the elements of $\phi(t)$: *linearly dependent vectors* (LDVs).

Convex mass constraints. Upper and lower bounds on the counterweight mass and COG coordinates and enforcing a particular counterweight shape can be expressed as convex constraints.

2.2 Objective Function. The dimensionless *balancing effect index* α is introduced in Ref. [10] as the root-mean-square (rms) (over one mechanism period T (s)) of the optimized dynamic force $f(t)$ with respect to the rms of the original dynamic force, indicated with a superscript $(\cdot)^o$

$$\alpha = \frac{\text{rms}(f(t))}{\text{rms}(f^o(t))}$$

We opt here for a problem formulation similar to [6,15]

$$\text{minimize} \quad \alpha_{\text{msh}} \quad (4a)$$

$$\text{subject to} \quad \alpha_{\text{fsh}} \leq \alpha_{\text{fsh}}^M, \quad \alpha_{\text{drv}} \leq \alpha_{\text{drv}}^M \quad (4b)$$

$$\alpha_{\text{bearing}} \leq \alpha_{\text{bearing}}^M \quad (4c)$$

That is, instead of minimizing a weighted combination of the balancing effect indices the balancing trade-off is controlled by minimizing the shaking moment α_{msh} subject to upper bounds α_{fsh}^M , α_{drv}^M , and $\alpha_{\text{bearing}}^M$, that is, the maximum allowed increase ($\alpha^M > 1$), or the minimum wanted reduction ($\alpha^M < 1$) of the rms shaking force, driving torque, and bearing forces.

2.3 Mass Constraints

2.3.1 Linear Mass Constraints. For physical reasons, counterweight mass m_i^* and inertia J_i^* need to be nonnegative (Eqs. (5a) and (5b)), while additional constraints are required to prevent excessive counterweight mass (the upper bound (5c) on the total counterweight mass) or counterweights that lie at too large a distance from the link to which they are attached (the upper and lower bounds (5d) and (5e) on the COG coordinates u_i^* and v_i^*). As demonstrated below, all these constraints are linear in μ_{1i}^* , μ_{2i}^* , μ_{3i}^* , and μ_{4i}^*

$$m_i^* \geq 0 \Leftrightarrow \mu_{1i}^* \geq 0 \quad (5a)$$

$$J_i^* \geq 0 \Rightarrow \mu_{4i}^* \geq 0 \quad (5b)$$

$$u_i^m \leq u_i^* \leq u_i^M \Leftrightarrow \mu_{1i}^* \cdot u_i^m \leq \mu_{2i}^* \leq \mu_{1i}^* \cdot u_i^M \quad (5c)$$

$$v_i^m \leq v_i^* \leq v_i^M \Leftrightarrow \mu_{1i}^* \cdot v_i^m \leq \mu_{3i}^* \leq \mu_{1i}^* \cdot v_i^M \quad (5d)$$

$$\sum_{i=1}^{N-1} m_i^* \leq m_{\text{tot}}^{*,M} \Leftrightarrow \sum_{i=1}^{N-1} \mu_{1i}^* \leq m_{\text{tot}}^{*,M} \quad (5e)$$

$(\cdot)^m$ and $(\cdot)^M$ denote lower and upper limits.

2.3.2 Nonlinear Mass Constraints. A particular counterweight shape can be enforced through the nonlinear constraint (expressed in the μ parameters, Eq. (5b) is a necessary but not a sufficient constraint)

$$\mu_{1i}^* \cdot \mu_{4i}^* = \delta((\mu_{2i}^*)^2 + (\mu_{3i}^*)^2) \quad (6)$$

where the shape factor δ is given by the following.

For point-mass, see Ref. [6],

$$\delta = 1 \quad (7a)$$

For minimum-inertia, see Ref. [8],

$$\delta = 1.5 \quad (7b)$$

For sector-type, with angle 2γ , see Sec. 4.1,

$$\delta = \left(\frac{3\gamma}{2\sqrt{2} \cdot \sin \gamma} \right)^2 \quad (7c)$$

To obtain a convex constraint, this constraint is *relaxed* by replacing the equality sign

$$\mu_{1i}^* \cdot \mu_{4i}^* \geq \delta((\mu_{2i}^*)^2 + (\mu_{3i}^*)^2) \quad (8)$$

which can be reformulated as a convex (second-order cone) constraint. This relaxation is inspired by the observation that counterweights with lower inertia are generally more favorable if shaking moment and driving torque are minimized or constrained. That is, in practice, Eq. (8) is very often *active*, i.e., valid with equality sign.

2.3.3 Equivalent Convex Program. Define auxiliary variables $z_d \in \mathbb{R}^n$, $d = \{\text{fsh}, \text{msh}, \text{drv}, \text{bearing}\}$, that simultaneously satisfy

$$\alpha_d = \|z_d\| \quad (9a)$$

$$\mathbf{z}_d = \Psi(k_{\text{link}}(t))_d \cdot \boldsymbol{\mu} \quad (9b)$$

where the matrices $\Psi_d \in \mathbb{R}^{n_d \times n_d}$ only depend on the mechanism kinematics $k_{\text{link}}(t)$ and $0 < n_d \leq P$. The derivation of this result is provided in Refs. [6,8] and is based on (i) the fact that expressions of type (3) are available for the considered force/moment and (ii) the eigenvalue decomposition. By using this result and the superposition result (2), and by introducing the auxiliary variable y the following *equivalent* problem formulation is found, with $\mathbf{x} = (\boldsymbol{\mu}^*, \mathbf{z}_d, y)$, $d = \{\text{msh}, \text{fsh}, \text{drv}, \text{bearing}\}$ as the optimization variable

$$\text{minimize}_{\mathbf{x}} \quad y \quad (10a)$$

$$\text{subject to} \quad \|\mathbf{z}_{\text{msh}}\| \leq y \quad (10b)$$

$$\|\mathbf{z}_{\text{fsh}}\| \leq \alpha_{\text{fsh}}^M, \quad \|\mathbf{z}_{\text{drv}}\| \leq \alpha_{\text{drv}}^M \quad (10c)$$

$$\|\mathbf{z}_{\text{bearing}}\| \leq \alpha_{\text{bearing}}^M \quad (10d)$$

$$\mathbf{z}_d = \Psi_d \cdot (\boldsymbol{\mu}^* + \boldsymbol{\mu}^o), \quad (10e)$$

$$\text{linear mass constraints (5a)–(5e)} \quad (10f)$$

$$\text{shape constraint (6)} \quad (10g)$$

This problem is a second-order cone program (SOCP): objective (10a) is linear, constraints (10b)–(10d) and (10g) are second-order cone constraints, and constraints (10e) and (10f) are linear.

3 Automatic Generation of LDV Expressions

This section develops a completely automatic method to derive LDV expressions of general form (3), for any force/moment $d(t)$ in a planar linkage of arbitrary complexity. The method, which is detailed in Sec. 3.2, is based on a force superposition principle that was inspired by Haines' work [13] and that is stated in Sec. 3.1.

3.1 Force Superposition Principle. Consider a planar N -bar mechanism of arbitrary complexity and three different mass distributions (denoted ${}^p\boldsymbol{\mu}$, ${}^q\boldsymbol{\mu}$, and ${}^r\boldsymbol{\mu}$). Let $d(t)$ be any (inertia) force/moment acting on the mechanism links. Denote the values of $d(t)$ corresponding to the mass distributions p , q , and r as ${}^pd(t)$, ${}^qd(t)$, and ${}^rd(t)$, respectively.

If now ${}^r\boldsymbol{\mu}$ is a linear combination of ${}^p\boldsymbol{\mu}$ and ${}^q\boldsymbol{\mu}$

$${}^r\boldsymbol{\mu} = v \cdot ({}^p\boldsymbol{\mu}) + w \cdot ({}^q\boldsymbol{\mu}) \quad (11)$$

where $v, w \in \mathbb{R}$, then ${}^rd(t)$ is also a linear combination of ${}^pd(t)$ and ${}^qd(t)$, with the same coefficients v and w

$${}^rd(t) = v \cdot ({}^pd(t)) + w \cdot ({}^qd(t)) \quad (12)$$

Proving this results in lengthy calculations for which the reader is referred to Ref. [16].

3.2 Automated LDV Procedure. The superposition principle is now used to develop an automated procedure for deriving Eq. (3). It is assumed that the kinematic parameters and the input crank speed are given. The procedure is applied to some dynamic reaction $d(t)$ (which can be any force component, moment, or linear combination of them, such as the shaking moment) and consists of four steps, given below.

Step 1. Construct the mass distribution matrix $\mathbf{M}_\mu \in \mathbb{R}^{P \times P}$

$$\mathbf{M}_\mu = [\boldsymbol{\mu}^1 \boldsymbol{\mu}^2 \cdots \boldsymbol{\mu}^P]$$

Each vector $\boldsymbol{\mu}^i \in \mathbb{R}^P$ represents a possible mass distribution of the mechanism. These vectors are chosen such that they are linearly independent. Hence $\mathbf{M}_\mu$ is of full rank P and constitutes a basis for representing any mass distribution of the linkage. The coordinates $\mathbf{s} \in \mathbb{R}^P$ of any linkage mass distribution $\boldsymbol{\mu}$ can therefore be determined based on a simple matrix inversion

$$\mathbf{M}_\mu \cdot \mathbf{s} = \boldsymbol{\mu} \Rightarrow \mathbf{s} = \mathbf{M}_\mu^{-1} \cdot \boldsymbol{\mu} \quad (13)$$

Step 2. First consider some force $d(t)$ in the mechanism and assume it is sampled at E equidistant time instants over one period T . These samples are grouped in the force vector

$$\mathbf{d} = [d(T_s) \ d(2T_s) \ \dots \ d(E \cdot T_s)]^T \quad (14)$$

where $T_s = T/E$ (s) denotes the sample period. Model and simulate the mechanism (in a standard multibody simulation package for instance) in order to generate the forces/moments corresponding to the P mass distributions $\boldsymbol{\mu}^i$. This results in $\mathbf{d}^i \in \mathbb{R}^E$, $i = \{1 \dots P\}$ vectors containing the E sampled values of the considered force/moment during one period.

Construct the matrix $\mathbf{D} \in \mathbb{R}^{E \times P}$ (not necessarily of full rank)

$$\mathbf{D} = [\mathbf{d}^1 \mathbf{d}^2 \cdots \mathbf{d}^P]$$

Step 3. Apply the force superposition principle in Sec. 3.1

$$\mathbf{d} = \mathbf{D} \cdot \mathbf{s} = \underbrace{(\mathbf{D} \cdot \mathbf{M}_\mu^{-1})}_{\boldsymbol{\Phi}} \cdot \boldsymbol{\mu} \quad (15)$$

where $\boldsymbol{\Phi} \in \mathbb{R}^{E \times P}$, and $\mathbf{d} \in \mathbb{R}^E$ is the vector of sampled $d(t)$ -values, corresponding to the arbitrary mass distribution $\boldsymbol{\mu}$. $\boldsymbol{\Phi} \in \mathbb{R}^{E \times P}$ only depends on the mechanism's kinematics and is given by

$$\boldsymbol{\Phi} = [\phi(T_s) \ \phi(2T_s) \ \dots \ \phi(E \cdot T_s)]^T$$

Comparing Eq. (15) with Eq. (3) reveals that the proposed procedure does not yield analytical expressions for $\phi(t)$ but a matrix $\boldsymbol{\Phi}$ of which the rows are the elements of $\phi(t)$ evaluated at the given time instants. Analytical expressions, however, are not needed since our only use of $\phi(t)$ is to generate matrix Ψ of Eq. (9b), for which the numerical matrix $\boldsymbol{\Phi}$ suffices [6,8].

3.3 Discussion. The force superposition principle of Sec. 3.1 has been developed based on a hint in the appendix of Ref. [13] but is, in the present form, novel in literature. The same goes for the procedure that results in Eq. (15). Furthermore, the procedure has two significant advantages over existing procedures.

- Generating $\mathbf{D}$ does not require symbolic manipulation of the equations of motion: it suffices to model and simulate the mechanism in a standard multibody simulation package, in order to generate the $\mathbf{d}^i$ vectors, corresponding to the P mass distributions $\boldsymbol{\mu}^i$. As a result, complexity of the mechanism is no longer an issue in deriving LDV expressions.
- The procedure is equally applicable to all dynamic reactions, including the bearing forces. To the best of our knowledge, no methodologies exist that generate LDV expressions for bearing forces in an automated way and for mechanisms of arbitrary complexity. Only for four-bars, dedicated LDV results for bearing forces are available [17].

4 Framework Extensions

This section develops four extensions of the basic framework outlined in Sec. 2.

The nonlinear inequality constraint (8) constitutes the Achilles' heel of the SOCP reformulation. Imposing the counterweight shape through Eq. (8) renders the optimization result dependent on the choice of LCS origin for minimum-inertia and sector-type counterweights. The relaxation on the other hand, might induce problems if it is not active in the optimum, which implies that the resulting counterweight does not have the shape chosen by the designer.

While both issues are mainly of theoretical interest for simple four-bar mechanisms, they do arise in practice for more complicated linkages [18]. Solutions for both issues are presented in Secs. 4.2 and 4.3, respectively. Based on these solutions, a three-step counterweight balancing procedure is proposed in Sec. 4.4.

Fig. 1 Sector-type counterweight (left) compared to minimum-inertia counterweight (right)

Section 4.5 shows that upper limits on the counterweight size can be expressed as a convex constraint. Before all of that, however, new results concerning sector-type counterweights are developed in Sec. 4.1.

4.1 Sector-Type Counterweights. The derivation of counterweight shape constraint (6) is based on the observation that if a particular counterweight shape is chosen, it suffices to provide values of μ_{1i}^* , μ_{2i}^* , and μ_{3i}^* in order to determine μ_{4i}^* .

If the counterweight is a point-mass, Eq. (6) is found with $\delta = 1$, by observing that $J_i^* = 0$ and expressing this constraint in terms of the μ -parameters [6].

If the counterweight is minimum-inertia with respect to the LCS origin, it follows that the diameter D_i^* (m) of this cylindrical counterweight and its centroidal inertia J_i^* equal [8]

$$D_i^* = 2\sqrt{(u_i^*)^2 + (v_i^*)^2} \quad (16a)$$

$$J_i^* = \frac{m_i^* \cdot (D_i^*/2)^2}{2} = \frac{m_i^* \cdot ((u_i^*)^2 + (v_i^*)^2)}{2} \quad (16b)$$

Now expressing Eq. (16b) in terms of the μ -parameters gives rise to Eq. (6) with $\delta = 1.5$, as shown more elaborately in Ref. [8].

Now assume that the counterweight is a sector-type counterweight, as depicted in Fig. 1, that is, with the sector center in the origin o_i of the LCS, sector radius R_i^* , sector thickness t_i^* , and a sector angle equal to 2γ (rad), ρ (kg/m³) the material density and r_i^* the distance between o_i and the COG. In that case, the following relations are valid [11]:

$$R_i^* = \sqrt[3]{\frac{3 \cdot m_i^* \cdot r_i^*}{2\rho \cdot t_i^* \cdot \sin \gamma}} \quad (17a)$$

$$m_i^* = \rho \cdot t_i^* \cdot \gamma \cdot (R_i^*)^2 \quad (17b)$$

$$J_i^* + m_i^* \cdot (r_i^*)^2 = \frac{m_i^* \cdot (R_i^*)^2}{2} \quad (17c)$$

$$r_i^* = \sqrt{(u_i^*)^2 + (v_i^*)^2} \quad (17d)$$

Eliminating $\rho \cdot t_i^*$ from Eqs. (17a) and (17b) leads to

$$r_i^* = \frac{2 \sin \gamma}{3\gamma} \cdot R_i^* \quad (18)$$

which substituted into Eq. (17c), yields

$$J_i^* + m_i^* \cdot (r_i^*)^2 = \left(\frac{3\gamma}{2\sqrt{2} \cdot \sin \gamma} \right)^2 \cdot m_i^* \cdot ((u_i^*)^2 + (v_i^*)^2)$$

Expressing this result in terms of the μ -parameters gives rise to Eq. (6) with δ given by Eq. (7c).

4.2 Dependency on Origin of Local Coordinate System. The derivation of Eq. (8) assumes that for minimum-inertia counterweights, o_i always lies on the counterweight's circumference and that for sector-type counterweights the sector center coincides with o_i . This renders the optimization result dependent on the

Fig. 2 Poor choice of LCS origin q . The sector-type counterweight with sector center in q' is the truly optimal coupler counterweight.

choice of LCS origin for minimum-inertia and sector-type counterweights. Conversely, point-mass counterweights are completely independent of the LCS choice.

Assume that the truly optimal sector-type counterweight configuration were the one depicted in Fig. 2, having q' as the sector center of the counterweight attached to the link. If the designer chose q instead of q' as the LCS origin, the counterweight would be forced to have its sector center in q , giving rise to an overly large counterweight and a balancing result that is potentially much worse.

The aforementioned problem might look somewhat artificial for four-bar mechanisms, where commonly two counterweights are attached to the ground points and only the attachment point of the coupler counterweight is free, but it is practically relevant for more complex linkages. However, based on the observation that (i) point-mass counterweights are independent of the LCS choice and (ii) the COGs of point-mass and sector-type counterweights generally lie fairly close to each other, the intuitive iterative procedure described in Sec. 4.4 is proposed to optimally choose the LCS origin for each link.

For technical details concerning the implications of this procedure for solving the SOCP of Eq. (10) the reader is referred to Ref. [18]. Figure 3 graphically illustrates the procedure with two typical cases, where only the behavior of the coupler counterweight of a four-bar is analyzed.

In the first, more frequently occurring case (Figs. 3(a) and 3(b)), the point-mass counterweight (Fig. 3(a)) is such that the closest point on the binary link is one of its end-points, in this case r . Since the LCS is initially in q , it is translated to r for the second iteration. The result of the sector-type counterweight optimization is shown in Fig. 3(b) and reveals that the point on the binary link, closest to the counterweight COG, is again r . Hence, the procedure stops.

In the second case (Figs. 3(c)–3(e)), the point-mass counterweight (Fig. 3(c)) is such that the closest point on the binary link is the point v lying between the end-points. Translating the LCS to v and determining the optimal sector-type counterweights results in Fig. 3(d), which shows that the point closest to the link of the resulting counterweight is w , which is already very close to v . Moving the LCS origin to w and making one final iteration results in Fig. 3(e): the counterweight is perpendicular to the link and no further iterations are necessary.

As Fig. 3 reveals, the procedure enforces the resulting sector-type counterweights to be *in contact* with the link, meaning that the sector center always lies on the link. The procedure also works well for minimum-inertia counterweights.

4.3 Upper Limit on Counterweight Inertia. As already mentioned, relaxation (8) might induce problems if it is not active in the optimum, which implies that the resulting counterweight does not have the shape chosen by the designer. This phenomenon

Fig. 3 Two typical cases for iterative procedure to optimally choose LCS origin: (i) closest point to point-mass counterweight on the binary link is one of its end-points (a–b); (ii) closest point to point-mass counterweight on the binary link is the point v lying between the end-points (c–d–e).

was first observed in Ref. [19] for the Watt six-bar, in which one of the links was given a counterweight with zero mass m_i^* and nonzero inertia J_i^* . In Ref. [19] the name *point-inertia* was chosen for such counterweights, and a limit mass distribution was proposed that realizes such counterweights.

Point-inertias are impractical and should therefore be avoided. While this problem was solved in Ref. [19] by setting μ_{1i}^* and μ_{4i}^* to zero for the problematic link, here a more elegant and completely automatic approach is proposed, by providing an upper limit on the counterweight inertia.

The upper inertia limit is derived from the exact counterweight constraint (6), by taking into account the bounds (5c) and (5d) on the counterweight coordinates

$$\begin{aligned}\mu_{1i}^* \cdot \mu_{4i}^* &= \delta \cdot ((\mu_{2i}^*)^2 + (\mu_{3i}^*)^2) \\ \mu_{1i}^* \cdot \mu_{4i}^* &= \delta \cdot ((\mu_{1i}^*)^2 \cdot (u_i^*)^2 + (\mu_{1i}^*)^2 \cdot (v_i^*)^2) \\ \mu_{1i}^* \cdot \mu_{4i}^* &\leq \delta \cdot ((\mu_{1i}^*)^2 \cdot D_{iu}^2 + (\mu_{1i}^*)^2 \cdot D_{iv}^2)\end{aligned}$$

where

$$D_{iu} = \max(|u_i^m|, |u_i^M|), \quad D_{iv} = \max(|v_i^m|, |v_i^M|) \quad (19)$$

Hence, taking into account Eq. (5a), we obtain

$$\mu_{4i}^* \leq \delta \cdot \mu_{1i}^* \cdot (D_{iu}^2 + D_{iv}^2) \quad (20)$$

which is a linear inequality constraint in μ_{1i}^* and μ_{4i}^* , and hence preserves convexity if added to the SOCP. Combining zero mass

and nonzero inertia is rendered impossible by Eq. (20), since zero mass reduces Eq. (20) to $\mu_{4i}^* \leq 0$, which in conjunction with Eq. (5b) and the zero mass, enforces $J_i^* = 0$.

As the derivation of Eq. (20) reveals, the upper inertia limit reduces to a linear *equality* constraint if $u_i^* = D_{iu}$ and $v_i^* = D_{iv}$. In that special case, the combination of relaxation (8) and upper limit (20) is equivalent to the original nonconvex constraint (6). That is, in that case, introducing Eq. (20) enforces the proper counterweight shape. From this special case it can be intuitively understood that the more u_i^* and v_i^* approach D_{iu} and D_{iv} , the more the resulting counterweight shape will resemble the desired counterweight shape. Interestingly enough, situations with inactive Eq. (8) very often coincide with u_i^* and v_i^* approaching their upper or lower limits imposed by Eqs. (5c) and (5d). Hence, with a slight modification detailed below, Eq. (20) enforces a counterweight shape very close to the desired counterweight shape.

The slight modification mentioned below concerns a more tight determination of D_{iu} and D_{iv} . If counterweights have first been determined with the upper limit Eq. (20) and D_{iu} and D_{iv} given by Eq. (19), the following stricter values D_{iu} and D_{iv} can be used for Eq. (20) in a subsequent iteration

$$D_{iu} = \begin{cases} |u_i^m| & \text{if } u_i^* \leq 0 \\ |u_i^M| & \text{if } u_i^* > 0 \end{cases} \quad (21a)$$

$$D_{iw} = \begin{cases} |v_i^m| & \text{if } v_i^* \leq 0 \\ |v_i^M| & \text{if } v_i^* > 0 \end{cases} \quad (21b)$$

where u_i^* and v_i^* denote the counterweight COG coordinates determined in the first iteration. This procedure assumes that the counterweight quadrant remains the same, which, in our numerical experience, is very likely.

The advantage of using Eq. (21) instead of Eq. (19) is that the former values are tighter upper bounds, resulting in a stricter upper inertia limit (20) and hence better chances to obtain a nearly desired counterweight shape. The drawbacks are the need to first do an iteration using Eq. (19) and the lack of theoretical guarantee that the counterweight quadrant remains the same. However, the latter event is unlikely, while the need to iteratively design counterweights already arose to determine the LCS. It hence suffices to include the update of D_{iu} and D_{iv} in that procedure to obtain the iterative completely automatic procedure detailed in Sec. 4.4.

4.4 An Automated Three-Step Counterweight Balancing Procedure. The iterative procedure to obtain the best choice of LCS is given here (nonitalic), while the iterative update of the bounds D_{iu} and D_{iv} is included (italic).

Iteration 1: initialization based on point-mass

11-1. Determine LCS-independent point-mass counterweights by solving (10h) with $\delta=1$, including the upper inertia limit (20), where D_{iu} and D_{iv} are given by Eq. (19).

11-2. For each link i for which $m_i^* > 0$: set LCS origin to point on link lying closest to the COG location of corresponding point-mass counterweight (the new LCS is parallel to the previous LCS; do not change the LCS origin if $m_i^* = 0$).

11-3. For each link i : determine D_{iu} and D_{iv} according to Eq. (21) if $m_i^* > 0$ and according to Eq. (19) if $m_i^* = 0$. Call these values $D_{iu}^{(1)}$ and $D_{iv}^{(1)}$.

Iteration 2: first sector-type counterweight iteration

12-1. Determine LCS-dependent sector-type counterweights by solving Eq. (10) with updated LCS origins and δ given by Eq. (7c), including the upper inertia limit (20), where $D_{iu} = D_{iu}^{(1)}$ and $D_{iv} = D_{iv}^{(1)}$.

12-2. For each link i for which $m_i^* > 0$: set LCS origin to point on link lying closest to the COG location of corresponding sector-type counterweight (the new LCS is parallel to the previous LCS; do not change LCS origin if $m_i^* = 0$).

12-3. For each link i : determine D_{iu} and D_{iv} according to Eq. (21) if $m_i^* > 0$ and according to Eq. (19) if $m_i^* = 0$. Call these values $D_{iu}^{(2)}$ and $D_{iv}^{(2)}$. (These values are highly likely to be equal to $D_{iu}^{(1)}$ and $D_{iv}^{(1)}$.)

Iteration 3: fine-tuning of sector-type counterweights¹

13-1. Determine LCS-dependent sector-type counterweights by solving Eq. (10) with updated LCS origins and δ given by Eq. (7c), including the upper inertia limit (20), where $D_{iu} = D_{iu}^{(2)}$ and $D_{iv} = D_{iv}^{(2)}$.

4.5 Upper Limit on Counterweight Size. As a final extension, this section proposes an effective way to avoid large counterweights by including (i) an upper bound L^M on the diameter of minimum-inertia counterweights or the radius of sector-type counterweights (ii) an upper bound on the counterweight thickness t^M .

Counterweight's radius. The upper bound on the counterweight's radius is given by

$$2 \cdot \sqrt{(u_i^*)^2 + (v_i^*)^2} \leq L^M \quad (22a)$$

¹The number of iterations is fixed to three: doing more iterations generally results in only marginal an improvement.

$$\frac{3\gamma}{2 \sin \gamma} \cdot \sqrt{(u_i^*)^2 + (v_i^*)^2} \leq L^M \quad (22b)$$

for minimum-inertia and sector-type counterweights, respectively. By introducing κ as

$$\kappa = 2 \quad (\text{minimum-inertia counterweights})$$

$$\kappa = \frac{3\gamma}{2 \sin \gamma} \quad (\text{sector-type counterweights})$$

constraint (22) can be written as

$$\kappa \cdot \sqrt{(u_i^*)^2 + (v_i^*)^2} \leq L^M \quad (23)$$

which can be reformulated as a second-order cone constraint in the variables μ_{1i}^* , μ_{2i}^* , and μ_{3i}^* .

Hence, including this upper bound in the SOCP of the three-step procedure does not destroy the SOCP structure. Since Eq. (23) does not make sense for point-mass counterweights, it is only added to the SOCP in steps I2-1 and I3-1 of the three-step procedure.

If the maximum distance from the LCS to the COG gives rise to a counterweight size larger than L^M , that is, if $\kappa \sqrt{D_{iu}^2 + D_{iv}^2} > L^M$, the upper limit on inertia (20) should be replaced by

$$\mu_{4i}^* \leq \delta \cdot \mu_{1i}^* \cdot \left(\frac{L^M}{\kappa} \right)^2$$

Counterweight's thickness. For sector-type counterweights substituting Eq. (17b) in Eq. (17c) for a given m_i^* yields

$$\mu_{4i} = \frac{m_i^* \cdot R_i^{*2}}{2} = \frac{1}{2 \cdot \rho \cdot t_i^* \cdot \gamma} \cdot m_i^{*2}$$

μ_4 is therefore minimal if $t = t^M$. Similar reasoning applies to cylindrical counterweights. As a result, counterweights with maximum thickness t^M must satisfy

$$\mu_4 \geq \lambda \cdot \mu_1^2 \quad (24)$$

where

- $\lambda = 3 / (2 \cdot \pi \cdot \rho \cdot t^M)$ for minimum-inertia CWs
- $\lambda = 1 / (2 \cdot \gamma \cdot \rho \cdot t^M)$ for sector-type CWs

which can be reformulated as a second-order cone constraint.

5 Numerical Results: Smith Ten-Bar

To illustrate the generality and the numerical superiority of the presented convex framework, the framework is applied to a complex planar mechanism that is the ten-bar previously considered by Smith [12], see Fig. 4. This ten-bar has four groundpoints A, D, G, and M. It has four ternary moving links: link 1 (ABE), 2 (BCJ), 5 (FGH), and 9 (LMN), where N describes the output trajectory and five binary moving links: link 3 (CD), 4 (EF), 6 (HK), 7 (JK), and 8 (KL). The fixed ground is denoted as link 10. The reference point for the shaking moment calculation is chosen equal to o , which lies in the middle between A and D. Since none of the parameters were available in Ref. [12], they were estimated. Table 1 summarizes the linkage numerical parameters. The input crank, link 1, makes an oscillating movement from -22.5° deg to 45° deg.

5.1 Design Choices. Regardless whether the original nonconvex optimization problem or the reformulated convex problem is solved, the designer has to make a few design choices. These choices include the upper limits α_{ish}^M , α_{drv}^M , and $\alpha_{bearing}^M$ defined by Eqs. (4b) and (4c), the upper limit $m_{tot}^{*,M}$ (kg) on the total counterweight mass (5e), the upper and lower limits on the COG coordinates defined by Eqs. (5c) and (5d), the type of counterweight defined by δ , and the upper limit L^M and t^M on the CW radius or diameter (21) and the counterweight thickness.

In order to simplify choosing $m_{tot}^{*,M}$, the COG bounds and the

Fig. 4 Kinematic scheme of the considered ten-bar.

radii bound the dimensionless design choices η_m , η_b , and η_R are introduced. $\eta_m > 0$, called the mass factor, is introduced to express the upper limit on the total counterweight mass as a fraction of the total mass m_{tot}^o of the unbalanced links

$$m_{\text{tot}}^{*,M} = \eta_m \cdot m_{\text{tot}}^o \quad (25)$$

$\eta_b > 0$, termed the box-factor, is introduced to describe a box around the link where the counterweight COG is allowed, as shown in Fig. 5. The bounds are given by

$$u_i^m = -\eta_b \cdot a_i, \quad u_i^M = (1 + \eta_b) \cdot a_i \quad (26a)$$

Table 1 Kinematic and dynamic parameters of the considered ten-bar linkage. a_b , b_b , u_i^o , v_i^o in millimeters, ξ_i in degrees, m_i^o in grams, and J_i^o in 10^{-4} (kg m)².

i	1	2	3	4	5	6	7	8	9
a_i	100	240	120	210	100	120	150	140	141
b_i	110	280	-	-	110	-	-	-	100
ξ_i	80	15	-	-	115	-	-	-	45
m_i^o	425	683	94	165	391	094	118	101	393
u_i^o	40	170	60	105	018	60	075	70	71
v_i^o	36	24	0	0	33	0	0	0	024
J_i^o	400	200	114	607	560	114	222	180	440

Fig. 5 Bounds on the COG coordinates of counterweight for (a) binary link and (b) ternary link

Fig. 6 Trade-off curves α_{msh} as a function of α_{fsh} with $\eta_m = 1.5$, $\eta_b = 1$, $\eta_R = 1$, and $t^M = 0.03$ (a) with $\alpha_{\text{bearing}}^M = 1.5$ for $\alpha_{\text{driv}}^M = \{1.15, 1.2, 1.25, 1.3\}$, respectively, (b) with $\alpha_{\text{driv}}^M = 1.25$ for $\alpha_{\text{bearing}}^M = \{1.5, 2, 10\}$, respectively

$$v_i^m = -\eta_b \cdot \frac{a_i}{2}, \quad v_i^M = b_i \cdot \sin \xi_i + \eta_b \cdot \frac{a_i}{2} \quad (26b)$$

with respect to the LCS (x_i, y_i) , see Fig. 5. a_i (m) and b_i (m) denote the distance between two joints of link i , and ξ_i (rad) denotes the angle between two sides of a ternary link i ($b_i = 0$, $\xi_i = 0$ in the case of a binary link). Similar equations describe the boundary box around other configurations of ternary links than the one in Fig. 5. η_b allows expressing the upper and lower bounds on the COG coordinates as a fraction of the size of the link, while taking into account the link shape. For a slider, a rectangle fitting the shape of the slider is multiplied by η_b to obtain the COG bounds.

The radius factor $\eta_R > 0$ is introduced to express the upper limit on the counterweight radius of link i as a fraction of the link length

$$L_i^M = \eta_R \cdot \max(a_i, b_i)$$

The three-step counterweight balancing procedure of Sec. 4.4 is implemented in MATLAB. Each convex optimization is solved using Yalmip, a MATLAB toolbox for modeling optimization problems independent of the numerical solver [20]. The SOCP numerical solver used here is SEDUMI, a dedicated software package for optimization problems over symmetric cones [21].

5.2 Effects of Design Choices. Given the very short computation time required to solve one instance of the SOCP, it becomes feasible to assess in an exhaustive manner the influence of the design choices identified here (α_{fsh}^M , α_{driv}^M , $\alpha_{\text{bearing}}^M$, η_m , η_b , η_R , δ , and t^M). This is done by solving the SOCP for all combinations of interest.

A very useful feature of convex programs, like the SOCP at hand, is the way in which infeasible problems are handled. If the considered combination is such that it is impossible to comply simultaneously with all constraints, SEDUMI does not crash, but yields a so-called *certificate of infeasibility*, which is a numerical proof that the problem is infeasible.

Trade-off plots can then be used to make a sensible design choice. For reason of conciseness, only a few of these trade-offs are shown here.

Figure 6 shows the trade-off curves α_{msh} as a function of α_{fsh}^M for different design choices.

Figure 6(a) shows that a driving torque of $\alpha_{\text{driv}}^M = 1.15$ results in significantly worse balancing results. $\alpha_{\text{driv}}^M = 1.25$ seems a good compromise, since a further increase in driving torque results in less significant improvement of the remaining dynamic reactions.

Figure 6(a) also reveals that full force balance ($\alpha_{\text{fsh}} = 0$) is infeasible for all presented trade-offs.

Figure 6(b) shows the effect of the constraint on the bearing forces. The strict constraint of $\alpha_{\text{bearing}}^M = 1.5$ results in significantly worse shaking moment and shaking force. However, not constraining the bearing forces ($\alpha_{\text{bearing}}^M = 10$) results in bearing forces that increase up to five times the original value.

Fig. 7 Trade-off curves α_{msh} as a function of α_{fsh} with $\alpha_{drv}^M = 1.25$, $\alpha_{bearing}^M = 1.5$, $\eta_m = 1$, and $\eta_b = 1$ for (a) $\eta_R = 0.06$ for point-mass (dashed-dotted line), minimum-inertia counterweights (dashed-line) and 90 deg-sector counterweights (solid line), (b) $\eta_R = \{0.4, 0.5, 0.6\}$, respectively

Effect of counterweight shape. Figure 7(a) shows that considering practical counterweights instead of impractical point-masses results in significantly less balancing potential. Point-mass counterweights in fact give the limits of counterweight balancing [6]. The 90 deg-sector-type counterweights give slightly better balancing results than minimum-inertia counterweights.

Effect of inertia constraint. Since a lower inertia is generally better when strict constraints on the driving torque have to be met, the extra inertia constraint has little effect in the considered range of values for this particular case. Not including the inertia constraint can however give rise to wrong expectations of the balancing result [18].

Effect of counterweight size. Figure 7(b) shows the effect of the constraint on the counterweight radius. $\eta_R = 0.4$ gives significantly worse balancing results than $\eta_R = 0.5$ or $\eta_R = 0.6$. Similar trade-off plots can be generated for different t^M .

Final counterweight design. The final counterweight design corresponding to the trade-off ($\alpha_{drv}^M = 1.25$, $\alpha_{fsh}^M = 0.45$, $\alpha_{msh} = 0.30$, $\alpha_{bearing}^M = 1.5$, $\eta_m = 1$, $\eta_b = 1$, $\eta_R = 0.6$, and $t^M = 0.05$ (m)) is given in Table 2, columns 2–4. The three-step procedure detects a necessary change of LCS for CW5. This LCS change however has little effect in this particular case ($\alpha_{msh} = 0.2954$ instead of 0.3032).

The CW configuration, which clearly is a practical one, is shown in Fig. 8, and the resulting forces and moments are shown in Fig. 9.

5.3 Nonconvex Optimization. The art and challenge in convex optimization are in formulating the problem. Once a problem is formulated as a convex program, it is relatively straightforward to solve it. To demonstrate that convex reformulation is worth the effort, a comparison is made between the convex optimization framework and commonly used algorithms to solve the following straightforward nonconvex formulation of the problem. The optimization variable $\mathbf{x} \in \mathbb{R}^P$, $P = 5 \cdot (N-1)$, groups the parameters (u_i^* , v_i^* , t_i^* , $d_{x,i}$, and $d_{y,i}$) for all links i , where ($d_{x,i}$, $d_{y,i}$) locate the sector center in the LCS. The optimization problem in $\mathbf{x}$ is

Table 2 Optimal counterweight parameters for partial balancing of the considered ten-bar linkage, corresponding to the trade-off ($\alpha_{drv}^M = 1.25$, $\alpha_{fsh}^M = 0.45$, $\alpha_{msh} = 0.30$, $\alpha_{bearing}^M = 1.5$, $\eta_m = 1$, $\eta_b = 1$, $\eta_R = 0.6$, and $t^M = 0.05$ (m))

Link i	1	3	5	9
m_i^*/m_{tot}^M (kg)	0.34	0.26	0.15	0.25
u_i^*/u^m (m)	0.36	0.25	0.23	0.18
v_i^*/v^m (m)	0.02	0.45	-0.46	0.10
R_i^*/L^M (m)	1	0.96	0.58	0.53

Fig. 8 Configuration of sector-type counterweights for partially balanced ten-bar of Table 2 columns 2–4.

$$\text{minimize}_{\mathbf{x}} \quad \alpha_{msh} \quad (27a)$$

$$\text{subject to} \quad \alpha_{fsh} \leq \alpha_{fsh}^M, \quad \alpha_{drv} \leq \alpha_{drv}^M \quad (27b)$$

$$\alpha_{bearing} \leq \alpha_{bearing}^M \quad (27c)$$

$$t_i^* \geq 0 \quad (27d)$$

$$u_i^m \leq u_i^* \leq u_i^M, \quad v_i^m \leq v_i^* \leq v_i^M \quad (27e)$$

$$u_i'^* = u_i^* - d_{x,i}, \quad v_i'^* = v_i^* - d_{y,i} \quad (27f)$$

$$(R_i^*)^2 = \left(\frac{3\gamma}{2 \sin \gamma} \right)^2 \cdot ((u_i'^*)^2 + (v_i'^*)^2) \quad (27g)$$

$$m_i^* = \rho \cdot t_i^* \cdot \gamma \cdot (R_i^*)^2 \quad (27h)$$

$$\sum_{i=1}^{N-1} m_i^* \leq m_{tot}^{*,M} \quad (27i)$$

$$R_i^* \leq R_i^M \quad (27j)$$

$$0 \leq d_{x,i} \leq a_i, \quad d_{y,i} = 0 \quad \text{for binary links} \quad (27k)$$

Equations (27a)–(27j) follow from previously explained physical reasons. Equation (27k) follows from the requirement that the sector center needs to be situated on the link (similar equations are

Fig. 9 Comparison of the (a) shaking force, (b) bearing force in A, (c) shaking moment, and (d) driving torque of the original linkage (dashed-line) and the partially balanced ten-bar (solid line)

Fig. 10 Histograms of the results for the Mstart-SQP for the considered ten-bar (a) exitflag, (b) optimal value α_{msh} , and (c) computation time

used for ternary links).

Two different strategies are used for numerically solving the optimization problem: a *multistart sequential quadratic programming* (Mstart-SQP) and a *genetic algorithm* (GA). Both methods are classical methods in the area of solving (nonconvex) nonlinear problems.

Mstart-SQP implies that a large set of (e.g., randomly generated) initial guesses is provided to the SQP algorithm, in this case the one provided by MATLAB's *fmincon* algorithm. Afterwards, the global optimum is found as the best of all local optima provided by different instances of the SQP algorithm. In this particular case, 64 different initial guesses are randomly generated for u_i^* , v_i^* , and t_i^* , but such that they comply with constraints Eqs. (27d) and (27e). $d_{x,i}$ and $d_{y,i}$ are initially taken as equal to zero.

The GA implementation used here is the one provided by the genetic algorithm toolbox of MATLAB. Standard values for the algorithm parameters were used. Since this method does not guarantee to find the global optimum, a sequence of 64 optimizations is solved, the final optimum is found as the best of all local optima.

After optimization, the algorithm returns an exitflag, corresponding to the status of the result, that is, >0 if the algorithm converged.

Figure 10 gives an overview of the results using the Mstart-SQP algorithm to balance the ten-bar with the design choices of Sec. 5.2, with counterweights allowed on all links. Figure 10(a) reveals that out of the 64 optimizations, only 29 converged (within the 10,000 function evaluations). Out of these 29, 23 start points lead to the final global optimum $\alpha_{msh}^M = 0.30$, see Fig. 10(b), while the other start points get stuck into local optima up to a shaking moment of $\alpha_{msh}^M = 0.68$. Finally, Fig. 10(c) gives an idea of the necessary computation time for each start point with a total of 5766 s.² The computation time for the three-step procedure was 3.5 s. This 3.5 s is much lower than the smallest necessary CPU-time for one instance of the Mstart-SQP. Due to this large computation time it is hardly practical to generate trade-off plots using the Mstart-SQP algorithm.

The GA method did not result in the global optimum, the lowest optimum found was $\alpha_{msh} = 0.51$ in 157 CPU s (for one of the 128 optimizations).

6 Discussion

The three-step procedure is a well-functioning heuristic because the point-mass counterweights provide an LCS-independent and extremely good initial guess of the COG location of counterweights. It is, however, a heuristic, which has a major consequence that the global optimum is no longer guaranteed. That is, for a given LCS choice, we are guaranteed to find the globally optimal counterweights, but we do not have a guarantee that the

LCS choice is globally optimal. In the considered example it was shown to be globally optimal. Our experience is that it is globally optimal as long as the mass constraint is active, and the thickness constraint is not active on the counterweights subject to LCS changes. If the thickness constraint is active, less mass can be added than might be necessary and therefore it might be interesting to keep a LCS that is not closest to the COG to add more mass. In that case an optimal LCS could be found by bisection.

While solving the convex optimization takes less than 1 CPU s, the initial construction of the D matrix will generally be more elaborate but worth the effort to generate trade-off chart. In the considered ten-bar example, this initial construction takes less than 2 CPU s (using MATLAB).

As mentioned earlier the convex optimization framework for the counterweight design of planar linkages assumes that all forces/moments can be determined through an inverse dynamic analysis implying the three assumptions described in Sec. 2. In reality these requirements are not always fulfilled. The framework could therefore be extended to include (i) flexibility of the links, (ii) clearance in the joints, and (iii) unknown speed variations. However, from our experience, we were able to obtain very satisfying balancing results on practical applications using these assumptions even knowing they did not hold completely.

7 Conclusion

This paper extends previous work by the same authors concerning convex reformulations of counterweight balancing. The extensions proposed in Sec. 4 gave rise to a fast procedure in which solving three instances of a convex program yields practical counterweight shapes to minimize and/or constrain shaking force, shaking moment, driving torque, and bearing forces in planar mechanisms of arbitrary complexity.

These extensions are numerically illustrated here by considering a ten-bar mechanism. It has been shown that our method finds the global optimum and this within a few CPU seconds, regardless of the complexity of the mechanism, which is in marked contrast with the large computation time necessary for the nonconvex optimization.

Given the very short computation time required to solve one instance of the convex problem, it becomes practically feasible to assess the effect of several design choices by generating trade-off plots for each of the combinations of interest.

The final result of the present paper is a general and numerically efficient framework for designers for counterweight balancing planar mechanisms of arbitrary complexity. This framework is easy to implement (for it does not require analytical derivation of the linkage kinematics or dynamics) and numerically very efficient (because of its formulation as a convex program).

Acknowledgment

This work has been carried out within the framework of Project No. G.0462.05. Bram Demeulenaere was a postdoctoral fellow of the Research Foundation—Flanders (FWO—Vlaanderen). The support of Project No. IWT 040444 and K.U.Leuven-BOF EF/05/006 Center-of-Excellence Optimization in Engineering is gratefully acknowledged.

References

- [1] Arakelian, V., and Smith, M., 2005, "Shaking Force and Shaking Moment Balancing of Mechanisms: A Historical Review With New Examples," *ASME J. Mech. Des.*, **127**, pp. 334–339.
- [2] Dresig, H., and Schönfeld, S., 1976, "Rechnergestützte optimierung der antriebs- und gestellkraftgrößen ebener koppelgetriebe (Computer-Aided Optimization of Driving and Frame Reaction Forces of Planar Linkages)," *Mech. Mach. Theory*, **11**(6), pp. 363–379.
- [3] Tricamo, S., and Lowen, G., 1983, "Simultaneous Optimization of Dynamic Reactions of a Four-Bar Linkage With Prescribed Maximum Shaking Force," *ASME J. Mech., Transm., Autom. Des.*, **105**, pp. 520–525.
- [4] Yan, H., and Soong, R., 2001, "Kinematic and Dynamic Design of Four-Bar Linkages by Links Counterweighting With Variable Input Speed," *Mech. Mach. Theory*, **36**, pp. 1051–1071.

²This CPU-time is necessary to have a higher probability to obtain the global optimum and is therefore used as comparison.

- [5] Chaudhary, H., and Saha, S., 2007, "Balancing of Four-Bar Linkages Using Maximum Recursive Dynamic Algorithm," *Mech. Mach. Theory*, **42**, pp. 216–232.
- [6] Demeulenaere, B., Aertbeliën, E., Verschuure, M., Swevers, J., and De Schutter, J., 2006, "Ultimate Limits for Counterweight Balancing of Crank-Rocker Four-Bar Linkages," *ASME J. Mech. Des.*, **128**, pp. 1272–1284.
- [7] Boyd, S., and Vandenberghe, L., 2004, *Convex Optimization*, Cambridge University Press, Cambridge, England.
- [8] Verschuure, M., Demeulenaere, B., Swevers, J., and De Schutter, J., 2008, "Counterweight Balancing for Vibration Reduction of Elastically Mounted Machine Frames: A Second-Order Cone Programming Approach," *ASME J. Mech. Des.*, **130**, p. 022302.
- [9] Kochev, I., and Gurdev, G., 1988, "General Criteria for Optimum Balancing of Combined Shaking Force and Shaking Moment in Planar Linkages," *Mech. Mach. Theory*, **23**(6), pp. 481–489.
- [10] Lowen, G., Tepper, F., and Berkof, R., 1974, "The Quantitative Influence of Complete Force Balancing on the Forces and Moments of Certain Families of Four-Bar Linkages," *Mech. Mach. Theory*, **9**, pp. 299–323.
- [11] 1999, "Blatt 1: Getriebedynamik-starrkörper mechanismen (Dynamics of Mechanisms-Rigid Body Mechanisms)," Verein Deutscher Ingenieure—Richtlinien, Paper No. VDI2149.
- [12] Smith, M., 1975, "Optimal Balancing of Planar Multi-Bar Linkages," *Proceedings of the 4th World Congress on the Theory of Machines and Mechanisms*, Newcastle-upon-Tyne, UK, pp. 145–149.
- [13] Haines, R., 1981, "Minimum rms Shaking Moment or Driving Torque of a Force-Balanced 4-Bar Linkage Using Feasible Counterweights," *Mech. Mach. Theory*, **16**, pp. 185–190.
- [14] Berkof, R., and Lowen, G., 1969, "A New Method for Completely Force Balancing Simple Linkages," *ASME J. Eng. Ind.*, **91**(1), pp. 21–26.
- [15] Sadler, J., and Mayne, R., 1973, "Balancing of Mechanisms by Nonlinear Programming," Third Applied Mechanism Conference, Paper No. 29.
- [16] Demeulenaere, B., 2004, "Dynamic Balancing of Reciprocating Machinery With Application to Weaving Machines," Ph.D. thesis, Department of Mechanical Engineering, Katholieke Universiteit Leuven, Belgium.
- [17] Tepper, F., and Lowen, G., 1975, "Shaking Force Optimization of Four-Bar Linkage With Adjustable Constraints on Ground Bearing Forces," *ASME J. Eng. Ind.*, **97**(2), pp. 643–651.
- [18] Verschuure, M., 2009, "Counterweight Balancing of Mechanisms Using Convex Optimization Techniques," Ph.D. thesis, Department of Mechanical Engineering, Katholieke Universiteit Leuven, Belgium.
- [19] Demeulenaere, B., Swevers, J., and De Schutter, J., 2006, "Ultimate and Practical Limits for Counterweight Balancing of Six-Bar Linkages," *ASME Paper No. DETC2006-99219*.
- [20] Löfberg, J., 2004, "Yalmip: A Toolbox for Modeling and Optimization in Matlab," *Proceedings of the 2004 IEEE CCA/ISIC/CACSD*, Taipei, Taiwan, Paper No. SaM09.1.
- [21] Sturm, J., 1999, "Using SeDuMi 1.02, a Matlab Toolbox for Optimization Over Symmetric Cones," *Optim. Methods Software*, **11**, pp. 625–653.