Available online at www.sciencedirect.com

ScienceDirect

IFAC PapersOnLine 53-2 (2020) 9405–9411

Sequence-Constrained Trajectory Planning and Execution for a Quadrotor UAV with Suspended Payload

JB Ubbink * **JAA Engelbrecht** *

* *Dept Electrical and Electronic Engineering, Stellenbosch University, Stellenbosch, South Africa (e-mail: johan.ubbink@gmail.com; jengelbr@sun.ac.za).*

Abstract: The paper presents the design of a trajectory planner and feedback control system to autonomously navigate a quadrotor UAV with a suspended payload through a confined environment consisting of horizontal and vertical tunnels. The trajectory planning task is formulated as an optimal control problem and solved by applying an A* search algorithm. A novel sequence-constrained action space is implemented to encourage the use of input shaping actions, which is an open-loop control technique for reducing vibrations in a response. To execute the planned trajectory, a trajectory regulator is designed to work in conjunction with the trajectory planner. The trajectory regulator uses feedback control to provide disturbance rejection and robustness to parameter uncertainty. The planning and execution is verified in simulation, using a system that is constrained to two dimensions. The trajectory planner successfully plans a collision-free path for the quadrotor with suspended payload through an environment with obstacles, tunnels and vertical chimneys. The regulator successfully controls the quadrotor with suspended payload to follow the planned trajectory through the environment, in the presence of external wind disturbances.

Keywords: Autonomous Vehicles, Transportation, Trajectory Planning, Obstacle Avoidance, Tree Searches

1. INTRODUCTION

Unmanned aerial vehicles (UAVs) show great potential in surveillance, photography and military applications. The ability to transport payloads further transform UAVs into versatile machines able to assist in rescue missions, package deliveries and construction. One method of payload transportation is to attach the payload directly to the body of the quadrotor UAV. This increases the vehicle's moment of inertia and thereby slows down the attitude dynamics of the vehicle (Nicotra et al., 2014). Another method is to suspend the payload beneath the quadrotor with a link (rigid or flexible). This introduces additional degrees of freedom that the control system must take into account. Flying with a suspended load can be a challenging task as the suspended load alters the flight characteristics of the vehicle (Palunko et al., 2012a).

This paper focuses on controlling a quadrotor UAV with a suspended payload, and navigating it through a confined environment. The problem is illustrated in Fig. 1. A quadrotor UAV with a suspended payload is shown at an initial position and is tasked to autonomously navigate to the goal region while avoiding the obstacles and walls.

Current approaches of suspended payload transportation can be divided into two broad categories, namely developing controllers for stabilisation of the payload, and using trajectory planning methods to generate a path with the desired payload swing (Wang and Xian, 2018).

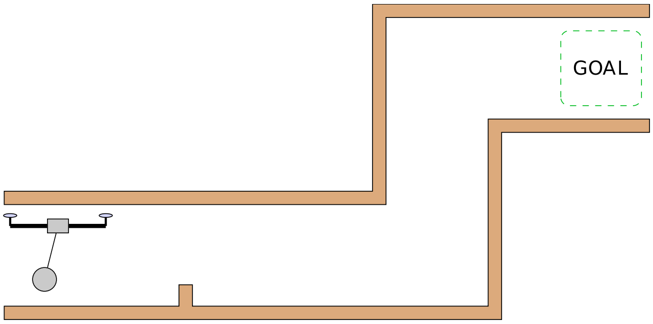

Fig. 1. Quadrotor with suspended payload shown at an initial position. The quadrotor UAV is tasked to autonomously navigate to the goal region, while avoiding the obstacle and boundaries.

Various types of controllers have been implemented with the goal of stabilising the payload, with various complexity (Nicotra et al., 2014; Goodarzi et al., 2014; Klausen et al., 2015; Liang et al., 2016; Tang et al., 2018). Nicotra et al. (2014) proposed a nested saturation control law, capable of steering the quadrotor UAV to a desired reference while simultaneously limiting the sway of the payload. Klausen et al. (2015) implemented a non-linear controller, based on a backstepping technique, that ensures trajectory tracking of the UAV regardless of the pendulum motion. Controllers designed to stabilise the payload usually need to determine the state of the payload. Methods to determine the payload state range from vision-based systems beneath the quadro-

10.1016/j.ifacol.2020.12.2410

tor (Tang et al., 2018), to adaptive control methods where the payload states are estimated (Goodarzi et al., 2014).

Trajectory planning methods use motion planning techniques in order to plan a trajectory which would result in the desired payload swing. Various optimisation methods have been applied to the problem (Palunko et al., 2012a,b; Faust et al., 2013; Tang and Kumar, 2015; Foehn et al., 2017; Wang and Xian, 2018). Dynamic programming is used by Palunko et al. (2012a) to generate swing-free trajectories. Tang et al. (2018) make use of Mixed Integer Quadratic programming to plan trajectories through an obstacle filled environment.

These trajectory planning methods work offline, planning open-loop commands to navigate through the environment. Control systems are then implemented to ensure the planned trajectory is executed. To address the problem of trajectory execution, Wang and Xian (2018) implemented an online trajectory planning system. Sreenath et al. (2013) proposed a hybrid solution consisting of both trajectory planning as well as load stabilising controllers. In their research, trajectory planning is used to swing the payload over an obstacle, and the feedback control system is then implemented to stabilise the payload after the action.

Transportation of a suspended payload is not unique to aerial vehicles, but is a common problem for crane operators. Crane operators often want to move a payload as fast as possible, but without swinging the payload in a dangerous manner. An experienced crane operator can sometimes produce the desired payload motion by pressing the accelerator button multiple times at proper instances (Singh and Singhose, 2002). Input-shaping, also known as command shaping, involves altering the shape of the actuator commands such that system oscillations are reduced (Singer and Seering, 1990). Homolka et al. (2017) and Ichikawa et al. (2018) successfully implemented input-shaping as a method to reduce the oscillations of a slung load attached beneath a quadrotor. It was however only used as an open-loop control input, and not incorporated with a trajectory planner.

This paper presents an offline trajectory planning method that uses input shaping and plans an open-loop trajectory through a confined environment. The trajectory planning task is formulated as an optimal control problem and is solved by applying the A* search algorithm (Hart et al., 1968). A trajectory regulator, making use of state variable control, is designed to ensure that the planned trajectory is executed in the presence of disturbances.

The rest of this paper is structured as follows: The architecture of the system is discussed in Section 2. The equations of motion describing the system is given in Section 3. Input shaping is briefly discussed in Section 4 with the goal to provide background for the trajectory planner in Section 5. The vehicle flight controller is presented in Section 6, with the trajectory regulator in Section 7. In Section 8 the performance of the complete system is verified through simulation. Finally, a conclusion is given in Section 9.

2. SYSTEM ARCHITECTURE

The system architecture showing the trajectory planner with the trajectory regulator is illustrated in Fig. 2. The trajectory planner generates the open-loop sequence needed to navigate through the environment. The open-loop sequence is provided to the flight controller, which adjusts the motor thrust to control the quadrotor to the desired attitude. Due to parameter uncertainty in the vehicle model, as well as outside disturbances, the quadrotor with suspended payload might deviate from the planned trajectory. To compensate for this, an additional trajectory regulator is implemented. The trajectory regulator compares the expected states of the system to the actual states, and applies additional forces to correct the deviations.

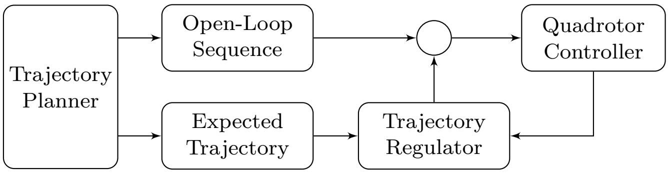

Fig. 2. The system architecture consisting of a trajectory planner and regulator.

3. MATHEMATICAL MODEL

This section defines the equations of motion describing the quadrotor-payload system by making use of the Euler-Lagrange method as derived by Nicotra et al. (2014). The system is constrained to two dimensions.

Fig. 3 shows a two-dimensional model of a quadrotor with a suspended payload. The positions of the centres of mass of the quadrotor and the suspended payload are given as p_Q and p_m respectively. The motor thrusts are represented by f_1 and f_2. The angle θ is the tilt angle between the local horizontal and the UAV. The rotational angle between the local vertical and the cable is represented by α. The cable connecting the quadrotor (p_Q) and the payload (p_m) has a fixed length of L and is approximated as a rigid link. The weight of the cable is neglected. To simplify the notation, system inputs are defined as the total motor thrust, $u_1 = f_1 + f_2$, and the moment around the quadrotor centre of mass, $u_2 = (f_1 - f_2)b$, where b is the distance between p_Q and the motors.

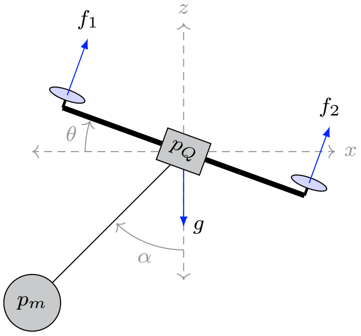

Fig. 3. 2D Model of a quadrotor UAV with a payload suspended through a rigid link.

Making use of the Euler Lagrange method, the dynamic equations are obtained as

$$\begin{aligned}\ddot{z} &= f_z(\alpha,\dot{\alpha}) + g_z(\alpha,\theta)u_1 + w_z(\dot{z} - v_{\text{wind}_z})\\ \ddot{x} &= f_x(\alpha,\dot{\alpha}) + g_x(\alpha,\theta)u_1 + w_x(\dot{x} - v_{\text{wind}_x})\\ \ddot{\alpha} &= g_\alpha(\alpha,\theta)u_1 + w_\alpha(\dot{\alpha} - v_{\text{wind}_\alpha})\\ \ddot{\theta} &= \frac{1}{\mathcal{J}}u_2\end{aligned} \tag{1}$$

where

$$\begin{aligned}f_z(\alpha,\dot{\alpha}) &= -g - \frac{m}{M+m}L(\cos\alpha)\dot{\alpha}^2\\ f_x(\alpha,\theta) &= -\frac{m}{M+m}L(\sin\alpha)\dot{\alpha}^2\\ g_z(\alpha,\dot{\alpha}) &= \frac{1}{M+m}(\cos\theta + \frac{m}{2M}(\cos\theta - \cos(\theta - 2\alpha)))\\ g_x(\alpha,\theta) &= \frac{1}{M+m}(\sin\theta + \frac{m}{2M}(\sin\theta - \sin(\theta - 2\alpha)))\\ g_\alpha(\alpha,\theta) &= \frac{1}{ML}\sin(\theta - \alpha)\end{aligned}$$

M and m are the mass of the quadrotor and payload respectively, and g is the gravitational acceleration. $\mathcal{J}$ is the moment of inertia of the quadrotor. External wind disturbances are incorporated through the functions w_z, w_x and w_α, which are functions of the vehicle velocity, payload velocity, and wind velocity.

4. INPUT SHAPING FOR OSCILLATION FREE RESPONSES

Input shaping, a technique to generate a non-oscillatory response for an underdamped plant, was proposed by Smith (1957) and Calvert and Gimpel (1957). This method is briefly discussed here, as it provides the basis for how the action space achieves an oscillation-free response in the next section.

To achieve an oscillation-free response, the original command signal is replaced with two commands. The response of the second command is designed to cancel out the oscillations induced by the first command. This is shown in Fig. 4. The amplitudes of A_1 and A_2, and the time delay for A_2, are carefully designed to cancel out the oscillations. The natural frequency w_n and damping ratio ζ for the plant must be known. Equation (2) can be used to calculate the sequence of two impulses, that produces zero residual vibration, where $K = \exp(\frac{-\zeta\pi}{\sqrt{1-\zeta^2}})$ and T_d is the damped period of vibration (Singh and Singhose, 2002).

$$\begin{bmatrix} A_1 & A_2 \\ t_1 & t_2 \end{bmatrix} = \begin{bmatrix} \frac{1}{1+K} & \frac{K}{1+K} \\ 0 & 0.5T_d \end{bmatrix} \tag{2}$$

However, it is not practical to control a real system with impulse commands. A method is needed to convert (2) into a usable form. To achieve this, the original input command for the system is shaped by convolving it with the impulse sequence determined in (2). If the impulse sequence causes no vibration, then the convolution product will also cause no vibration (Singer and Seering, 1990).

5. TRAJECTORY PLANNER

The trajectory planner uses the A* search algorithm, developed by Hart et al. (1968), to find a collision-free state trajectory that will navigate the quadrotor with suspended payload through the confined environment. The algorithm starts at an initial starting node and attempts to find a path to a goal node with the lowest cost. Actions from a finite action space are applied to the node that is explored. By applying these actions, new nodes are created which need to be explored. These new nodes are placed in a queue of nodes to explore next. In the case of the A* algorithm, this queue is sorted based on the cost to reach that node (cost to come) plus a heuristic (cost to go) to reach the goal state.

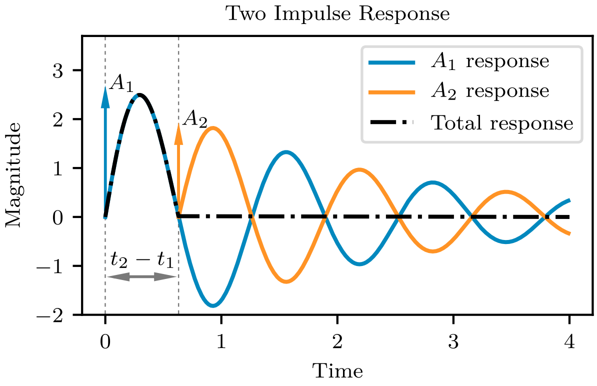

Fig. 4. Two impulse responses resulting in oscillation-free response (Singh and Singhose, 2002). The second impulse, A_2, is designed to cancel out the response of the first impulse, A_1.

The search space for the trajectory planner is selected as the state space of the quadrotor with suspended payload, consisting of the position and velocity of the quadrotor as well as the angle and angular rate of the payload. Symbolically it is described as the set $X = [z, \dot{z}, x, \dot{x}, \alpha, \dot{\alpha}]$. Obstacles in the environment are represented by state constraints on the position of the quadrotor and payload. To ensure the payload does not swing more than physically possible, the payload angle is constrained to an angle between $-70° < \alpha < 70°$.

To determine the state of a child node generated by an action, an initial value problem is solved. The new state at the next time step (child node) is determined by simulating the dynamic equations for one sampling period with the chosen action applied, starting from the initial state at the current time step (parent node). The forward simulation represents the state transition function for the system. This new state is then stored in a child node, and added to the queue to be investigated later.

The trajectory planner needs a library of actions, called the action space, to propagate the system state and to grow the search tree. For our planner, we designed a library of complex actions that use input shaping. The actions are designed to move the system through the environment, while minimizing payload swing. Fig. 5 shows the *start moving* action that starts the quadrotor-payload system moving from an initial stationary state. The *start moving* action uses an input shaping manoeuvre applied to a simulated velocity controller to start the vehicle moving without inducing payload oscillations. The timing and magnitude of the steps is calculated using (2). Fig. 6 shows the *continue moving* action that keeps the vehicle moving at a constant velocity, after it has already started moving.

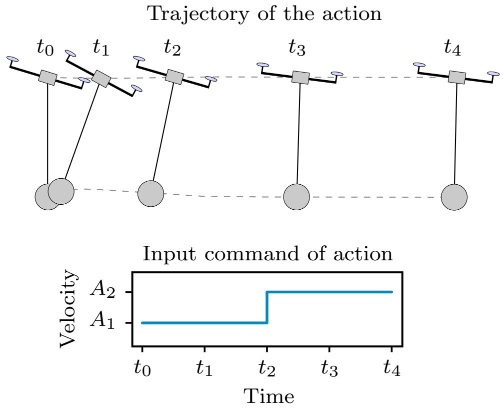

Fig. 5. Action to start moving from standstill.

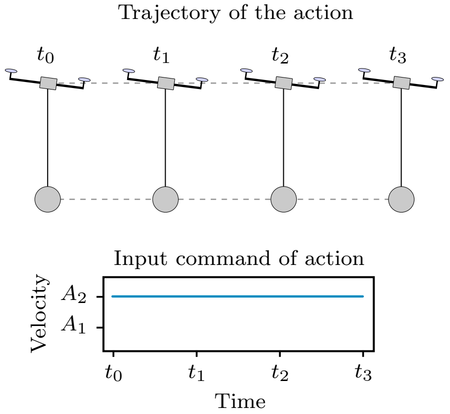

Fig. 6. Action to continue moving.

Fig. 7 shows the *stop moving* action, which brings an already moving quadrotor with payload to standstill with minimal swing by applying an input shaping action. The same actions are repeated for moving in the opposite direction, as well as upwards and downwards.

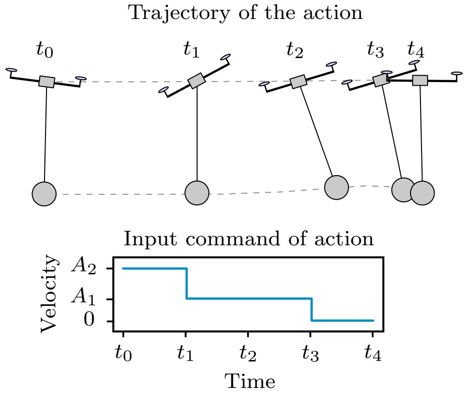

Fig. 7. Action to bring the quadrotor to a standstill.

A special *obstacle avoidance* action is shown in Fig. 8 to demonstrate the ability of the planner to use different, more aggressive actions, would the environment require it. The *obstacle avoidance* action allows the quadrotor and payload system to "jump a hurdle" in its path. The action consists of supplying a large force reference in the forward direction, followed by a force reference in the backwards direction. A desirable characteristic of this input sequence is that the cumulative forces in the forward and backwards directions are equal, resulting in a zero net velocity increase. The oscillations of the forward forces cancels the oscillations of the backwards movement, stopping the payload swing after the movement. (A drawback of this specific obstacle avoidance action is that the size of the obstacle it can clear is relatively small in comparison to the distance that the quadrotor needs to travel to clear it. To address this problem, more advanced obstacle avoidance actions can be designed using trajectory optimisation techniques. However, the current action is sufficient to demonstrate the ability of the planner to include different actions in its action space.)

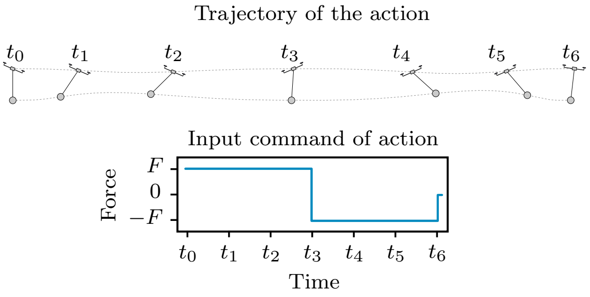

Fig. 8. Action to swing the payload over an obstacle.

As previously stated, the queue of nodes to investigate is sorted based on the cost to reach that node (cost to come) plus a heuristic (cost to go) to reach the goal state. The cost to come is selected as the total distance travelled by the quadcopter, as well as the effort of the actions. The *obstacle avoidance* action takes more effort to executed than the *continue moving* action. It is therefore penalised more by the cost function. The heuristic is selected as the euclidean distance to the goal position.

To limit the size of the action space and the size of the resulting search graph, we implemented a novel sequence-constrained action space. For instance, it would not make sense to apply the *start moving* action, if the system is already moving. Therefore certain actions are removed from the action space based on the previous applied input. The sequence-constrained action space is summarised in Table 1. The second column shows the available actions in the action space based on the previous input in the sequence, shown in column one.

To incorporate the velocity input shaping actions in the trajectory planner, a velocity controller was implemented in the state transition function. However, the *obstacle avoidance* action is an exception, since it bypasses the velocity controllers, and supplies its force commands directly as horizontal and vertical force commands for the inner-loop controllers. For the velocity commands to work with the trajectory regulator, the commands need to be converted to force references. By simulating the path with velocity commands on an ideal model, the force commands generated by the controller can be recorded. The recorded force commands are applied as an open-loop input sequence to the system, when the quadrotor-payload system executes the trajectory.

Table 1. Action space given previous input

Previous Input	Current Action Space
Stop moving	Start moving (forwards) Start moving (backwards) Upwards action Downwards action
Start moving (forwards)	Continue moving Stop moving Avoid obstacle
Continue moving	Continue moving Stop moving Avoid obstacle
Avoid obstacle	Continue moving Avoid obstacle Stop Action
Upwards action	Upwards action Standstill action

6. FLIGHT CONTROLLER DESIGN

The flight control system is designed to receive horizontal and vertical force commands, and uses the individual motor thrusts, f_1 and f_2, in order to produce these forces. The flight control system is designed based on the vehicle model only, and does not use the vehicle-payload model. The payload is treated as an external disturbance.

The control architecture is illustrated in Fig. 9. The horizontal and vertical force references are converted to a corresponding tilt angle θ and total thrust u_1. A tilt angle controller uses the moment u_2 produced by the differential thrust of the quadrotor motors to rotate the quadrotor to the desired angle θ. The total thrust, u_1, and differential thrust, u_2, are converted to individual motor thrusts f_1 and f_2 through the motor mixing component.

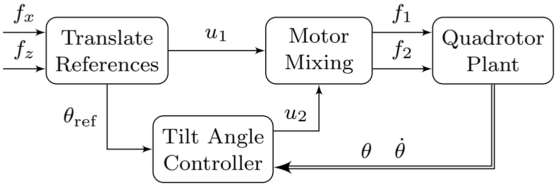

Fig. 9. The flight control system which receives force references to adjust the attitude of the quadrotor UAV.

The motors can only produce a finite amount of thrust, therefore the reference values for f_1 and f_2 might not match the actual motor output. A general rule of thumb is that the motors should be able to provide at least twice as much thrust as the weight of the quadrotor (Javir et al., 2015). Therefore the individual maximum motor thrust is limited to $f_{\max} = f_{\text{trim}}$, where f_{trim} is the force required to hover the quadrotor.

The control law of the tilt angle controller is given by

$$u_2 = k_{\theta 1}(-\dot{\theta} + k_{\theta 2}(\theta_{\text{ref}} - \theta)), \quad (3)$$

where $k_{\theta 1}$ and $k_{\theta 2}$ are controller gains. The controller should allow enough time for the transient response of the motor controller to adjust the motor thrusts. The controller is designed to be optimally damped with a settling time of $t_{2\%} = 0.15$s. This settling time is designed to allow enough time for the motors to rotate the quadrotor without saturating when a tilt angle reference is applied. The reference angle θ_{ref} is limited to $\theta_{\max} = 45°$, to ensure that the tilt angle controller does not receive unrealistic reference angles that would tilt the quadrotor too far.

7. TRAJECTORY REGULATOR

As described in Section 2, a trajectory regulator is designed to work in conjunction with the trajectory planner. The trajectory planner provides the planned open-loop force commands to navigate through the environment, while the trajectory regulator compensates for parameter uncertainty and external disturbances. If the executed quadrotor position or velocity trajectory deviates from the planned trajectory, then restoring feedback control force commands are superimposed on the planned open-loop force commands to return the quadrotor to the planned trajectory.

Fig. 10 shows the block diagram for a general state-space regulator for the horizontal dynamics. The same architecture is used for the vertical regulator. The goal is to design a gain $\mathbf{k}_x = [k_{x1} \quad k_{x2}]^T$ to place the closed-loop poles at the desired positions.

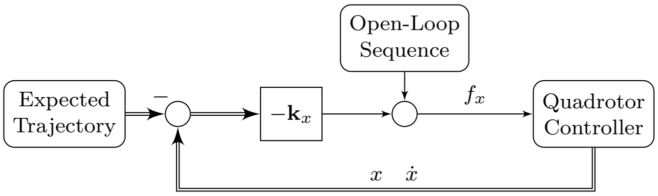

Fig. 10. Control architecture of the horizontal trajectory regulator.

It is important to choose the regulator dynamics to be slow enough to give the inner-loop system enough time to adjust, but to be fast enough to effectively reject any disturbances. The horizontal system is constrained by the response time of the tilt angle controller and is designed to be optimally damped with a 2% settling time of $t_{s2\%} = 1.5s$. The settling time is chosen to be significantly longer than the response time of the inner-loop tilt angle controller. The vertical dynamics is constrained by the time constant of the motors, as well as their maximum thrusts. The regulator is designed to be optimally damped with a 2% settling time of $t_{s2\%} = 1.3s$, ensuring that the actuators do not saturate when a step reference is applied.

8. RESULTS

The trajectory planner and regulator were tested in a simulated environment similar to the scenario shown in Fig. 1. The confined test environment consists of a lower horizontal tunnel that contains an obstacle that needs to be “jumped”, followed by a vertical chimney, followed by another horizontal tunnel leading to a goal region where the quadrotor must come to a standstill. The distance between the obstacle and the ceiling is less than the height of the quadrotor with suspended payload. The trajectory planner therefore needs to swing the payload to clear the gap. The scenario is selected as it is a

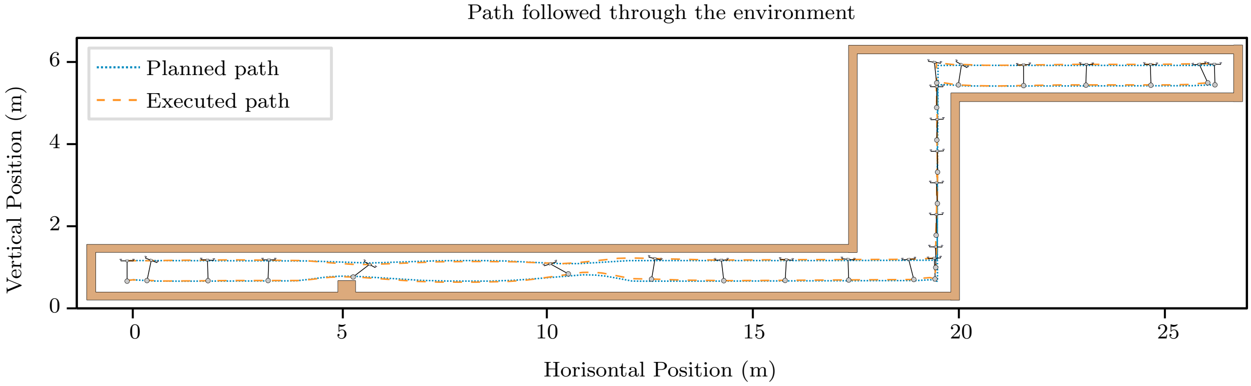

Fig. 11. Path executed by the quadrotor and payload system including actuator dynamics.

challenging environment, showcasing the ability of the planning algorithm to apply a wide variety of the actions from its sequence-constrained action space. The same algorithm can be applied in open air environments, to plan paths resulting in minimal payload swing.

The trajectory planner uses only the slower quadrotor and payload translational dynamics to perform the path planning, and neglects the fast rotational dynamics of the quadrotor's attitude motion. However, the simulation model[1] that is used to test the trajectory planner and regulator is more representative, and includes the fast rotational dynamics, realistic sensor noise on the quadrotor's position and velocity sensors, and external wind disturbance. The mathematical model in (1) is used, with the controllers described in Section 6. Turbulence, modeled as shaped white noise, is applied as the wind disturbances.

The generated action sequence is shown in Fig. 12, consisting of the input shaping velocity commands, vertical velocity commands, and obstacle avoidance force commands.

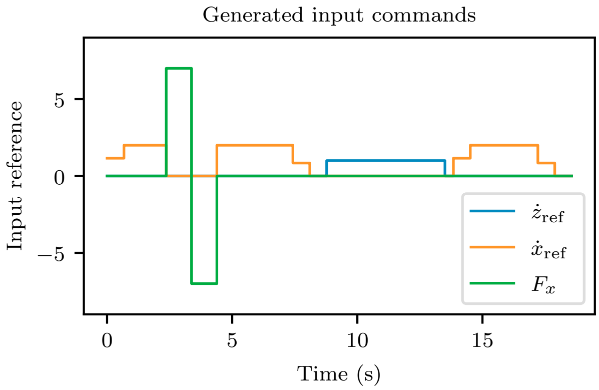

Fig. 12. Input sequence generated by the trajectory planner, consisting of velocity and force commands.

The velocity commands are converted through simulation to corresponding open-loop force references, shown in Fig. 13. These force commands are applied in an open-loop manner to the plant. The planned and executed trajectory are shown in Fig. 11. With the help of the trajectory regulator, the executed path closely matches the planned path. Input shaping actions are used to start and stop the quadrotor, and the obstacle avoidance action is used to swing the payload over the obstacle.

The swing of the payload is plotted in Fig. 14. The payload swing is kept to a minimum, with slight oscillations due to the quadrotor trajectory deviating slightly from the planned path. The payload swings up to 60° when passing over the obstacle.

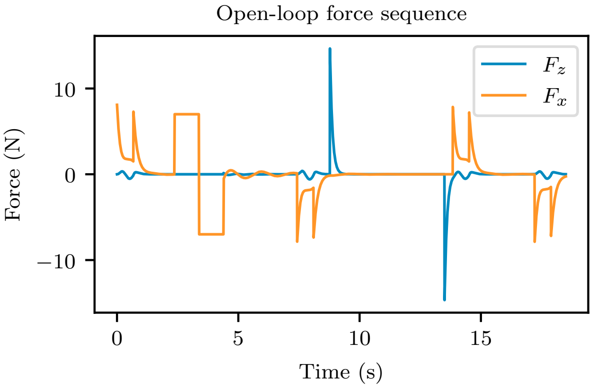

Fig. 13. Input sequence converted to open-loop force commands.

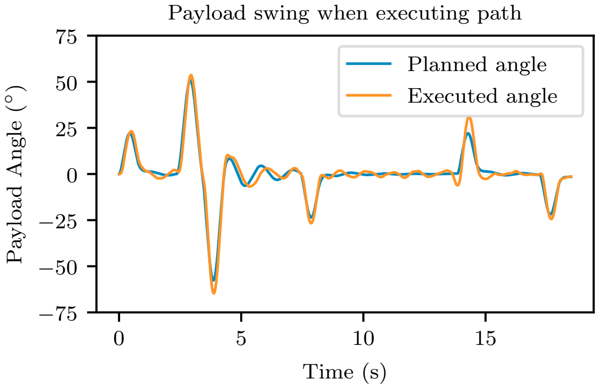

Fig. 14. Payload swing while executing the path.

[1] Simulation model is available at `www.github.com/johanubbink/trajectory-planning-quad-payload`

9. CONCLUSION

The trajectory planner successfully plans a collision-free path for the quadrotor with suspended payload through a complex confined environment that contains tunnels, chimneys, and obstacles that create narrow gaps. The trajectory regulator then successfully controls the quadrotor with suspended payload to follow the planned trajectory through the environment in the presence of external wind disturbances.

Currently the trajectory planning is performed offline for a 2D model in a known environment. Expanding from a 2D model to a 3D model will increase the dimensions of the search space. However, our sequence-constrained approach limits the growth of the search tree, increasing the feasibility of high-dimensional search. To implement the system in an online manner, the timing consistency needs to be improved. The algorithm can be implemented in a more optimized compiled language. Variants of the A* algorithm that are more suited for real-time planning, such as the Real Time A* algorithm (Korf, 1990), may also be considered. Once an online strategy with a real-time planner is implemented, navigating an uncertain, dynamic environment becomes more realistic, since the trajectory planner could be designed to re-plan when new information regarding the environment is obtained.

REFERENCES

Calvert, J.F. and Gimpel, D. (1957). Method and Apparatus for Control of System Output Response to System Input. *U.S. Patent #2,801,351.*

Faust, A., Palunko, I., Cruz, P., Fierro, R., and Tapia, L. (2013). Learning swing-free trajectories for UAVs with a suspended load. In *2013 IEEE International Conference on Robotics and Automation*, 4902–4909. doi:10.1109/ICRA.2013.6631277.

Foehn, P., Falanga, D., Tedrake, R., Scaramuzza, D., and Kuppuswamy, N. (2017). Fast Trajectory Optimization for Agile Quadrotor Maneuvers with a Cable-Suspended Payload. In *Robotics: Science and Systems XIII.* Robotics: Science and Systems Foundation. doi:10.15607/rss.2017.xiii.030.

Goodarzi, F.A., Daewon Lee, F.A., and Taeyoung Lee, F.A. (2014). Geometric stabilization of a quadrotor UAV with a payload connected by flexible cable. In *American Control Conference (ACC), 2014*, 4925–4930. American Automatic Control Council.

Hart, P.E., Raphael, B., and Nilsson, N. (1968). A Formal Basis for the Heuristic Determination of Minimum Cost Paths. *IEEE Transactions on Systems Science and Cybernetics.* doi:10.3305/nh.2014.29.1.7039.

Homolka, P., Hromcik, M., and Vyhlidal, T. (2017). Input shaping solutions for drones with suspended load: First results. In *Proceedings of the 2017 21st International Conference on Process Control, PC 2017*, 30–35. Institute of Electrical and Electronics Engineers Inc. doi:10.1109/PC.2017.7976184.

Ichikawa, S., Castro, A., Johnson, N., Kojima, H., and Singhose, W. (2018). Dynamics and command shaping control of quadcopters carrying suspended loads. *IFAC-PapersOnLine*, 51(14), 84 – 88. doi:https://doi.org/10.1016/j.ifacol.2018.07.203. 14th IFAC Workshop on Time Delay Systems TDS 2018.

Javir, V., Pawar, K., Dhudum, S., Patale, N., and Patil, S. (2015). Design, Analysis and Fabrication of Quadcopter. *Journal of Advance Research in Mechanical & Civil Engineering (ISSN: 2208-2379)*, 2(3), 16–27.

Klausen, K., Fossen, T.I., and Johansen, T.A. (2015). Nonlinear control of a multirotor UAV with suspended load. *2015 International Conference on Unmanned Aircraft Systems (ICUAS)*, 176–184. doi:10.1109/ICUAS.2015.7152289.

Korf, R.E. (1990). Real-time heuristic search. *Artificial Intelligence*, 42(2), 189 – 211. doi:https://doi.org/10.1016/0004-3702(90)90054-4.

Liang, X., Fang, Y., and Sun, N. (2016). A hierarchical controller for quadrotor unmanned aerial vehicle transportation systems. In *2016 35th Chinese Control Conference (CCC)*, 6148–6153. doi:10.1109/ChiCC.2016.7554322.

Nicotra, M.M., Garone, E., Naldi, R., and Marconi, L. (2014). Nested saturation control of an UAV carrying a suspended load. *Proceedings of the American Control Conference*, 3585–3590. doi:10.1109/ACC.2014.6859222.

Palunko, I., Cruz, P., and Fierro, R. (2012a). Agile load transportation : Safe and efficient load manipulation with aerial robots. *IEEE Robotics and Automation Magazine*, 19(3), 69–79. doi:10.1109/MRA.2012.2205617.

Palunko, I., Fierro, R., and Cruz, P. (2012b). Trajectory generation for swing-free maneuvers of a quadrotor with suspended payload: A dynamic programming approach. *Proceedings - IEEE International Conference on Robotics and Automation*, 2691–2697. doi:10.1109/ICRA.2012.6225213.

Singer, N.C. and Seering, W.P. (1990). Preshaping Command Inputs to Reduce System Vibration. *Journal of Dynamic Systems, Measurement, and Control*, 112(1), 76. doi:10.1115/1.2894142.

Singh, T. and Singhose, W. (2002). Tutorial on Input Shaping/Time Delay Control of Maneuvering Flexible Structures. *Proceedings of the 2002 American Control Conference (IEEE Cat. No.CH37301)*, 1717–1731 vol.3. doi:10.1109/ACC.2002.1023813.

Smith, O.J.M. (1957). Posicast Control of Damped Oscillatory Systems. *Proceedings of the IRE*, 45(9), 1249–1255. doi:10.1109/JRPROC.1957.278530.

Sreenath, K., Michael, N., and Kumar, V. (2013). Trajectory generation and control of a quadrotor with a cable-suspended load-A differentially-flat hybrid system. *Proceedings of the 2013 IEEE International Conference on Robotics and Automation (ICRA)*, 4888–4895. doi:10.1109/ICRA.2013.6631275.

Tang, S. and Kumar, V. (2015). Mixed Integer Quadratic Program trajectory generation for a quadrotor with a cable-suspended payload. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, 2216–2222. doi:10.1109/ICRA.2015.7139492.

Tang, S., Wüest, V., and Kumar, V. (2018). Aggressive Flight With Suspended Payloads Using Vision-Based Control. *IEEE Robotics and Automation Letters*, 3(2), 1152–1159. doi:10.1109/LRA.2018.2793305.

Wang, S. and Xian, B. (2018). An Anti-swing Trajectory Approach for an Unmanned Aerial Vehicle with A Slung Payload. In *2018 37th Chinese Control Conference (CCC)*, 5560–5565. doi:10.23919/ChiCC.2018.8483176.