Aggressive Optimal Control for Agile Flight with a Slung Load

Cédric de Crousaz, Farbod Farshidian and Jonas Buchli

***Abstract*— A possible design method for learning motion control consists of using a model-based optimal control algorithm to initialize the policy of a sampling based reinforcement learning algorithm. In this paper, the initial control trajectory design is performed for a quadrotor with a cable-suspended load using the iterative LQG (iLQG) algorithm. The hybrid model of the quadrotor with the load is introduced, and four sample tasks are presented. These tasks consist of a simple waypoint task as well as a task where the quadrotor and the load have to pass through a small window. The window task requires more aggressive control, exploiting the underactuated system dynamics including system mode switches, where the load is in free fall for some period of time.**

I. INTRODUCTION

Agile and robust motion control is the key to autonomous robot operation. One approach for designing the motion controller is to design the reference trajectory and the tracking controller separately. In contrast, the design process can be introduced as a single optimization problem, similar to the approach in optimal control and reinforcement learning. In these frameworks, the task is introduced by the means of a cost function which can be optimized either through a model-based (optimal control) or a sample-based method (reinforcement learning). While the model-based approaches leverage the knowledge of the system model, the sample-based approaches interact directly with the real world.

A combined approach can benefit from both the system model knowledge and the real world information. In the author's previous work [1], a combined design methodology for robotic platforms was introduced. This approach proposes a design pipeline which first it uses the iterative LQG (iLQG) algorithm (as a model-based method) to design the controller. This controller is then used to initialize the policy of a learning algorithm, namely PI^2-01 (as a sample-based method). With this approach, while the iLQG algorithm provides a good initial controller for the given task, the PI^2-01 algorithm improves its performance on the real hardware by adapting the controller to the unmodelled dynamics. Since both of these algorithms are using the same cost function, it relieves the designer from tuning the cost function for each algorithm separately. This design pipeline is summarized in Fig 1. The shaded parts are completed in simulation while the solid white parts require trials on the real hardware.

The applicability of this design approach to a ball balancing robot, which is an unstable, nonlinear, and non-minimum phase system, was shown by the authors in [1]. The first half of the design pipeline (the shaded part in Fig 1) is now applied to a quadrotor with a slung load (Fig 2), showing that the design methodology also scales to such a challenging system.

Cédric de Crousaz {cedricd@student.ethz.ch}, Farbod Farshidian {farbodf@ethz.ch} and Jonas Buchli {buchlij@ethz.ch} are with the Agile & Dexterous Robotics Lab at the Institute of Robotics and Intelligent Systems, ETH Zürich, Switzerland.

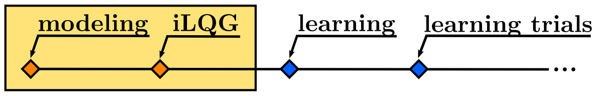


Fig. 1: Overview of the design pipeline

A quadrotor with a slung load is a system with relatively high degree of underactuation. Furthermore, depending on the cable being taut or free, the system has two dynamics modes with a different number of states and degrees of underactuation. Under aggressive manoeuvres, like passing through a small window, the system switches between theses modes. This high number of underactuation plus the hybrid characteristic of the system makes the motion control of the system challenging, especially for aggressive manoeuvres.

In order to perform dynamic and agile tasks like passing through windows, the quadrotor needs to exploit mode-switches. In this work the iLQG algorithm is used to automate the design process of motion control. The generated controller is not only able to fulfill the task but also tries to minimize the control effort.

II. RELATED WORK

Recently using quadrotor UAVs (Unmanned Aerial Vehicles) for manipulation has become popular for several reasons such as inexpensive hardware and the ability to access remote areas. Although adding a gripper to quadrotor for manipulation is a viable solution [2], it adds extra weight and reduces the agility of the aircraft. However hanging the load from a cable retains the agility of UAVs at the expense of making the motion control problem harder as the degree of under-actuation increases. In this area, a large body of the research focuses on keeping the load swing in a minimum possible level [3], [4]. Both model-based [4]–[8] and sample-based [3], [9] methods have been used in this scenario to reduce the energy of the suspended load's residual oscillation.

Using the differentially flatness property of the system, Sreenath et al. [10], [11] introduce a motion controller which lets the load undergo large swings. This controller tracks the reference trajectories in the space of flat outputs. These reference trajectories are provided by the designer. As the complexity of the task increases (like passing-through-window task), the intuition of the designer will not be sufficient anymore. In order to automate the design process of the reference trajectories, an optimization approach is suggested by [12], [13]. By using these optimized reference trajectories, Mellinger et al. [12] demonstrate dynamical manoeuvres like passing through a window only a bit wider than the quadrotor.

The work in this paper has similar elements to [13] in using an optimization to design the motion controller. There are however major differences between these two works. The first and the most important one is that in our proposed approach, the whole controller is optimized, and not just the reference trajectories. This allows to design a motion controller which adapt its disturbance rejection gains with regard to the requirement of the given task. Furthermore in [13] the optimization is performed in the space of flat outputs while here it is in state space. Finally this works tested the designed controller performance in more challenging tasks.

III. PROBLEM DEFINITION

In order to use iLQG for motion control design, the model of the quadrotor with a suspended load is derived. The quadrotor has six degrees of freedom (DOF), but only four actuators, the rotors. The load, which is modelled as a point mass, adds another three DOF. If the cable is taut, the distance r between the quadrotor and the load is fixed, resulting in a system with four degrees of underactuation. When there is no tension on the cable, the two bodies are essentially independent, increasing the degree by one.

A. A Hybrid System

The system at hand consists of a quadrotor with mass m_Q and moment of inertia $\mathbf{I}_Q$, from which a load with mass m_L is hung by a massless cable of length L_c as shown in Fig 2. As the cable can only handle tensile forces along

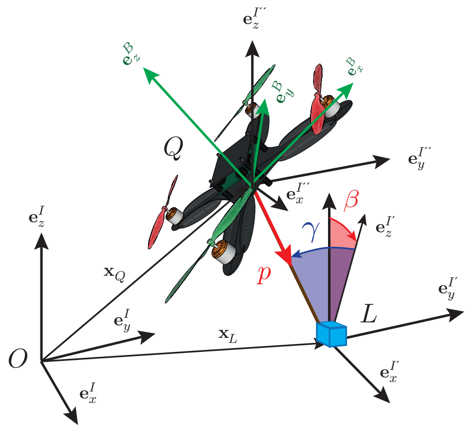


Fig. 2: Overview of coordinate description

the direction of the string, this is a hybrid system with two different system dynamics, referred to as modes. In the first mode, the cable is taut and transferring a force between load and quadrotor, while in the second, there is no tension in the cable and the load is in free fall.

A transition between the system modes happens whenever the state trajectory intersects with one of the switching surfaces, denoted by $\mathcal{S}$. The free fall mode is entered as soon as the tension in the cable is zero, which represents the switching surface $\mathcal{S}_1$. As soon as the load moves away from the quadrotor and has reached the distance $r = L_c$ from the center of mass of the quadrotor, represented by $\mathcal{S}_2$, the tension inside the cable is non-zero again, and mode 1 is entered. Using a similar notation as [10], this can be summarized as

Mode 1) taut cable

$$\Sigma_1 = \begin{cases} \dot{\mathbf{x}}_1 = f_1(\mathbf{x}_1, \mathbf{u}), & \mathbf{x}_1 \notin \mathcal{S}_1 \\ \mathbf{x}_2^+ = \Delta_{1\to 2}(\mathbf{x}_1^-), & \mathbf{x}_1 \in \mathcal{S}_1 \end{cases}$$

$$\Delta_{1\to 2} : \text{identity map}$$

$$\mathcal{S}_1 = \{\mathbf{x}_1 \,|\, F_c = m_L\,(\ddot{\mathbf{x}}_L + g\,\mathbf{e}_3)\cdot \mathbf{p} \equiv 0\}$$

Mode 2) free fall

$$\Sigma_2 = \begin{cases} \dot{\mathbf{x}}_2 = f_2(\mathbf{x}_2, \mathbf{u}), & \mathbf{x}_2 \notin \mathcal{S}_2 \\ \mathbf{x}_1^+ = \Delta_{2\to 1}(\mathbf{x}_2^-), & \mathbf{x}_2 \in \mathcal{S}_2 \end{cases}$$

$$\Delta_{2\to 1} : \text{inelastic collision}$$

$$\mathcal{S}_2 = \{\mathbf{x}_2 \,|\, |r| = L_c,\ \frac{d}{dt} r > 0\}$$

$f_i(\mathbf{x}_i, \mathbf{u})$ are the nonlinear system dynamics of mode i with the corresponding state and input vector, $\Delta_{i\to j}$ denotes the transition map from mode i to mode j and the exponents $(\cdot)^-$ and $(\cdot)^+$ symbolize the time instances right before and after the mode switch.

By continuity, the transition map $\Delta_{1\to 2}$ is simply identity. The only thing to consider is that the state vector $\mathbf{x}_1$ of mode 1 does not contain r, the distance from the center of mass of the quadrotor to the load, nor $\dot{r}$, which are set to $r = L_c$ and $\dot{r} = 0$ when the cable is taut.

To switch from the free fall mode to the model where the cable is taut, the transition is modelled as an inelastic collision. The position and orientation of the bodies is again unaltered by continuity, but since the energy is not conserved, the velocities will change. The transition map $\Delta_{2\to 1}$ is obtained by considering the conservation of linear and angular momentum of the system.

To avoid having to handle changing dimensions during the swing process, the code was implemented using an extended system where the variables r and $\dot{r}$ are also included in the state of mode 1. The state thus always has dimension 18.

B. The Model

The quadrotor and the load are modelled as a rigid body and a point mass respectively [10]. A position vector $\mathbf{x}_L$ or $\mathbf{x}_Q$ defines the position of the load or quadrotor respectively. The relation between the two position vectors is given as

$$\mathbf{x}_Q = \mathbf{x}_L - r\cdot \mathbf{p}, \tag{1}$$

where $\mathbf{p}$ is a unit vector pointing from the center of mass of the quadrotor to the load and r is the distance between the two. The angles γ, β together with the distance r describe the location of the quadrotor relative to the load as depicted in Fig 2. γ and β are introduced such that no discontinuities or singularities appear around the equilibrium position. The orientation of the quadrotor with respect to the inertial world frame I is described by the angles ϕ, θ, and ψ. Starting in the inertial frame, ϕ first rotates around $\mathbf{e}_x^I$, followed by a rotation of θ around the new y-axis and then ψ around z.

Neglecting rotor dynamics, each rotor i generates a thrust $F_{t,i} = k_F \cdot \omega_i^2$ and a moment $M_i = k_M \cdot \omega_i^2$, saturated by $\omega \in [125.7, 816.8]\,[rad/s]$. Unless specified, all the parameter values in this paper are taken from [14]. The matrix for the moment of inertia of the quadrotor in the body-fixed frame is given as $\mathbf{I}_Q = diag(0.03, 0.03, 0.05)\,[kg\, m^2]$. The load is a point mass and thus has no moment of inertia.

The inputs to the system are $\mathbf{u}^\top = [F_z, M_x, M_y, M_z]$, where F_z is the total thrust, and M_x, M_y, and M_z the moments along the corresponding body-fixed axes.

The generalized coordinate vector consists of the elements listed in Table I. Two model descriptions are used, one using the load position $\mathbf{x}_L$, the other one using the quadrotor position $\mathbf{x}_Q$ as a base. The two descriptions are dynamically equivalent, but since the relation (1) is nonlinear, the different formulations change the form of the cost function. This will be discussed in more detail in section IV-A.

TABLE I: Generalized coordinates

Symbol	Description
$\mathbf{x}_L, \mathbf{x}_Q$	Position vector of load and quadrotor in inertial frame
ϕ, θ, ψ	Angles describing the orientation of quadrotor body-fixed frame with respect to inertial frame
γ, β	Angles describing the position of quadrotor with respect to the load
r	Distance from quadrotor to load in direction of $\mathbf{p}$

C. Optimal Control Algorithm

The iLQG algorithm is an iterative method which returns a locally optimal linear feedback controller able to work with arbitrary nonlinear cost functions [15]. In each step of the iteration, the system is linearized around a nominal trajectory and a quadratic approximation of the cost is minimized to obtain a new control input. For more details about the iLQG algorithm the reader is referred to [15].

The initial trajectory for the iLQG algorithm is generated using a linear quadratic regulator (LQR) controller. Note that this only stabilizes the quadrotor at its initial position and does not introduce any moving trajectory. The equilibrium state and input vectors around which the LQR solution is centered are

$$\begin{aligned} \mathbf{x}_{eq}^\top &= [\cdot,\cdot,\cdot,0,0,0,0,0,0,L_c,0,0,0,0,0,0,0,0] \\ \mathbf{u}_{eq,1}^\top &= [(m_L + m_Q)\cdot g, 0, 0, 0]\,, \quad \text{if cable taut} \\ \mathbf{u}_{eq,2}^\top &= [m_Q \cdot g, 0, 0, 0]\,, \quad \text{if load in free fall} \end{aligned}$$

The first three elements of $\mathbf{x}_{eq}$ are arbitrary, since the equilibrium state is independent of the position of the quadrotor.

D. Cost Function

The general cost function is defined as

$$J = \mathrm{E}\left[h(\mathbf{x}(t_f))) + \textstyle\sum_{t=0}^{t_f-1} l(t, \mathbf{x}(t), \mathbf{u}(t))\right] \tag{2}$$

For the examples in this paper, a terminal cost of the form

$$h(\mathbf{x}(t_f)) = (\mathbf{x}(t_f) - \mathbf{x}_{goal})^\top \mathbf{C}_1 (\mathbf{x}(t_f) - \mathbf{x}_{goal}) \tag{3}$$

where $\mathbf{x}_{goal}$ is the desired terminal state at the final time $t = t_f$, and an immediate cost defined as

$$\begin{aligned} l(t, \mathbf{x}(t), \mathbf{u}(t)) &= (\tilde{\mathbf{x}}(t) - \mathbf{x}_{\text{eq}})^\top \mathbf{C}_2 (\tilde{\mathbf{x}}(t) - \mathbf{x}_{\text{eq}}) \\ &\quad + (\mathbf{u}(t) - \mathbf{u}_{\text{eq}})^\top \mathbf{C}_3 (\mathbf{u}(t) - \mathbf{u}_{\text{eq}}) + g(t, \mathbf{x}, \mathbf{u}) \end{aligned} \tag{4}$$

are used. The first two terms in $l(t, \mathbf{x}, \mathbf{u})$ are time-independent and assure the stability of the quadrotor flight, while $g(t, \mathbf{x}, \mathbf{u})$ is an arbitrary, possibly time-varying cost function defined differently for each task to achieve the desired goal. $\mathbf{C}_1, \mathbf{C}_2$, and $\mathbf{C}_3$ were chosen as diagonal matrices. For the examples presented in this paper, $\tilde{\mathbf{x}}$ is always the state vector with the quadrotor position and velocities to penalize the speed of the quadrotor. The time dependent cost function for the tasks at hand is defined as

$$\begin{aligned} &C(t_p, \mathbf{x}_p, \mathbf{W}_p, \rho) = \\ &\quad (\mathbf{x} - \mathbf{x}_p)^\top \mathbf{W}_p (\mathbf{x} - \mathbf{x}_p) \cdot \sqrt{\frac{\rho}{2\pi}} e^{\left(-\frac{\rho}{2}(t - t_p)^2\right)} \end{aligned} \tag{5}$$

which penalizes the deviation of the system from a desired state $\mathbf{x}_p$ at time t_p. The parameter $\rho \in \mathbb{R}$ defines how precise the timing has to be. This flexible function is used in different forms, summarized in Table II. The weight $\mathbf{W}_p \in \mathbb{R}^{s \times s}$ is usually chosen as a diagonal matrix, where s is the dimension of $\mathbf{x}_p$. For small values of ρ, the cost is spread out over time without a strong peak, which can be useful for general waypoint tasks to give the algorithm leeway in choosing the time when to pass the waypoint. If the quadrotor should only be at $\mathbf{x}_p$ for a short time instance, as is the case in sections IV-B–IV-D, ρ has to be bigger. This strong peak can lead to a failure of the iLQG procedure, in which case the cost weights need to be incremented gradually as in section IV-C.

TABLE II: Different forms of the waypoint cost (5)

Name	What it penalizes
$C_{wp,Q}$	Deviation of quadrotor position from $\mathbf{x}_p$, dimension 3
$C_{wp,L}$	Deviation of load position from $\mathbf{x}_p$, dimension 3
$C_{sp,Q}$	Deviation of quadrotor state from $\mathbf{x}_p$, dimension 18
$C_{sp,L}$	Deviation of load state from $\mathbf{x}_p$, dimension 18

IV. TASKS AND RESULTS

Four tasks as well as their resulting trajectories are presented here for a system with $m_Q = 0.5\,[kg]$, $m_L = 0.05\,[kg]$, and $L_c = 1\,[m]$. In section IV-A, a simple waypoint task is solved following points positioned in a figure-8 pattern. Sections IV-B to IV-D investigate the learning capabilities for more agile tasks with potential mode switches. For this, the quadrotor and the load have to move through a window of width w_w and height w_h centered at w_{pos}. The window height is chosen smaller than the cable is long, forcing the quadrotor to swing up the load to make it pass through the window. In a first step in section IV-B, the quadrotor is guided towards a solution by asking the load to pass first. In section IV-C, this guideline is removed, and the system needs to find a way on its own. Finally, in section IV-D, the quadrotor is guided again, but this time, the window is too narrow for the quadrotor to fly through level.

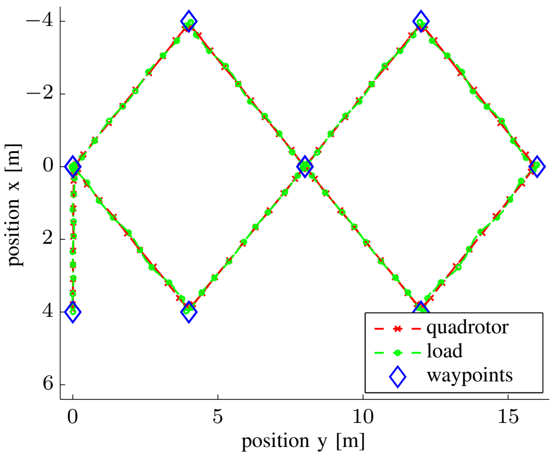


Fig. 3: Top view of figure 8 task

For each of the tasks, the initial state is the hovering state in mode 1, i.e. the load is hanging vertically with zero velocity and the quadrotor is level. The detailed cost functions for are listed in the appendix.

A. Figure 8 Task

The quadrotor is required to pass several waypoints arranged in a figure-8 pattern, forming an "8" when looking from the "top" onto the xy-plane, but also positioned at different z coordinates. The waypoints are located at

$$\begin{array}{lll} \mathbf{x}_{p0}^{\top} = [0,0,0] & \mathbf{x}_{p1}^{\top} = [4,4,2] & \mathbf{x}_{p2}^{\top} = [0,8,4] \\ \mathbf{x}_{p3}^{\top} = [-4,12,2] & \mathbf{x}_{p4}^{\top} = [0,16,0] & \mathbf{x}_{p5}^{\top} = [4,12,-2] \\ \mathbf{x}_{p6}^{\top} = [0,8,-4] & \mathbf{x}_{p7}^{\top} = [-4,4,-2] & \mathbf{x}_{p8}^{\top} = [0,0,0] \end{array}$$

After this path, the quadrotor should move to the goal position at $\mathbf{x}_{goal}^{\top} = [4,0,0]$. The time given to move from one waypoint to the next is $6\,[s]$, and the starting point is left $5\,[s]$ into the task, resulting in a slight swinging of the load around the quadrotor trajectory.

The cost function in this task has the general format as in the equations (3), (4) and (5) (for more details refer to the appendix). Furthermore, the model with the quadrotor position and velocity as states is used here. The model with the load position is also a valid option, but since the relation (1) is nonlinear, the cost is not quadratic anymore. The quadratic approximation of the cost can then lead to a divergence of iLQG if e.g. the time frame is reduced.

The result after only three iterations starting from the LQR initialization is shown in Fig 3. The convergence of the algorithm in this case is very fast, as can be seen from the total cost listed in Table III.

TABLE III: Cost for the figure 8 task

iteration	0	1	2	$\cdots$	10
total cost	259060.11	2899.32	2899.02	$\cdots$	2899.08

B. Guided Window Task

In this task, the quadrotor and the load have to move through a window of width $w_w = 0.6L_c$ and height $w_h = 0.4L_c$ centered at w_{pos}. The window height being smaller

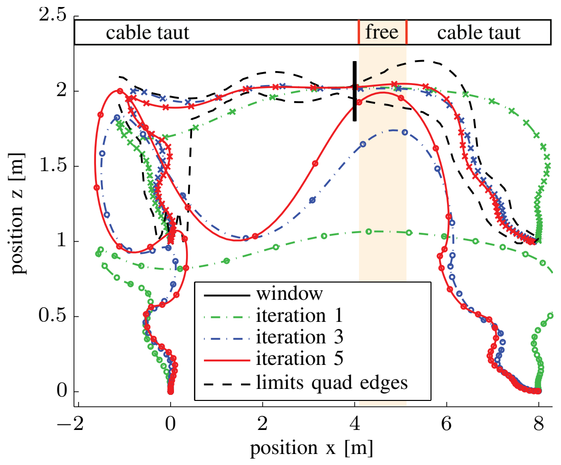


Fig. 4: Side view of the window task for different iterations. The quadrotor curve is marked with "x", the load with "o". The dashed black lines named *limits quad edges* are the outermost points of the quadrotor arms for any time instance. The system mode is displayed in the bar at the top, the shaded area marking the distance over which the load is in free fall.

than the cable length, the quadrotor is forced to swing up the load to make it pass through the window.

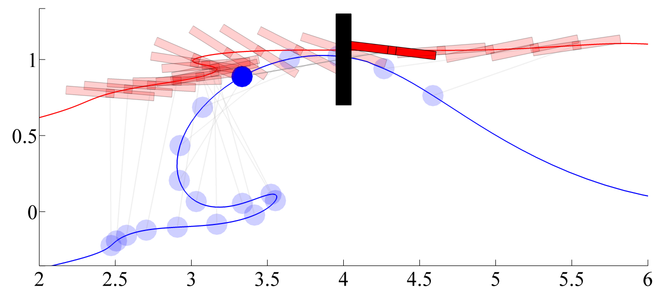
The cost function was designed as in (3), and (4) with $g(t,\mathbf{x},\mathbf{u})$ given in (6).

$$\begin{aligned} g(t,\mathbf{x},\mathbf{u}) &= C_{sp,L}(t_1,\mathbf{x}_{win},\mathbf{W}_{win},14) + \dots \qquad (6)\\ &\quad C_{wp,Q}(t_1+0.01,w_{pos},diag(0,300,500),5) \\ w_{pos} &= [4,0,2]\,, \quad t_1 = 0.5\cdot t_f \\ \mathbf{x}_{win}^{\top} &= [\,w_{pos}(1),w_{pos}(2),w_{pos}(3)+0.8w_h,\ \dots \\ &\quad 0,0,0,0,-0.5\pi,0,8,zeros(1,8)\,] \\ \mathbf{W}_{win} &= diag\,(5,5,4,0,0,0,10,0,0,1,zeros(1,8))\cdot 10^2 \\ t_f &= 12\,[s], \quad \mathbf{x}_0^{\top} = [\,0,0,0\,]\,, \quad \mathbf{x}_{end}^{\top} = [\,8,0,0\,] \end{aligned}$$

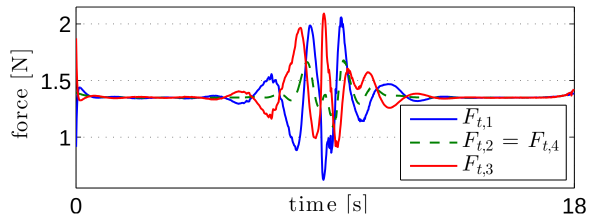
The solution is partially guided in this case as the load is required to pass in front of the quadrotor by the penalty $\gamma(t_1) = -0.5\pi$ in $\mathbf{x}_{win}$. To make sure that the quadrotor actually moves through the window and does not simply swing the load in position, $\dot{x}(t_1) = 8[m/s]$ is set. The exact value being arbitrary, the penalty weight in $\mathbf{W}_{win}$ for the velocity needs to be low to only influence the behaviour without requiring the quadrotor to actually reach that specific velocity. Since both the load and the quadrotor position are penalized, there is no clear advantage of using one of the other formulation. Often, better results were obtained by taking the formulation of the object for which the cost function is more restricting. Here, this is the load, since the precision parameter ρ is higher in the load cost.

Starting from the LQR solution, the first, third and fifth iteration are displayed in Fig 4.

As one can see, the quadrotor initially focuses on moving from the starting point $x_0^{\top} = [0,0,0]$ to the goal at $x_f^{\top} = [8,0,0]$, but rises first to pass the window. After that, the cost on the load angle becomes dominant, and the load is gradually swung up, until both the quadrotor at the load pass the window in the fifth iteration. Just when passing the window, the cable tension vanishes and there is a short



(a) Side view



(b) Rotor thrust with thrust saturation at $0.088\,[N]$ and $3.716\,[N]$

Fig. 5: Quadrotor passing through window without any penalty to enforce a method

mode switch, displayed in the bar at the top of Fig 4.

C. Non-Guided Window Task

This first task is promising, but the quadrotor is partially told how to perform the task through a detailed cost function, in which the load is told to pass the window in a horizontal position in front of the quadrotor. Removing the terms for the cable orientation leads to a more general cost function, but also a divergence of the iLQG algorithm. The cost terms are thus added step by step, increasing the weights gradually, a common approach within the control optimization framework [16]. For each set of parameters, 5 iterations are performed with the model using the quadrotor position as coordinates. The setup of the cost function can be summarized as such:

1) Small penalty asking the load to pass through center of window, small ρ
2) Increase the penalty on the load
3) Add penalty for quadrotor to pass through window, increase ρ for the load
4) Require load to be moving when passing the window, increase position penalties, reduce penalties on velocities, and add a cost on the load for being close to the x coordinate of the window

Item 4 is necessary, as the load is only required to pass through the window, but has no knowledge of the wall the window is in. If this cost on the window position is omitted, the load will be swung up "through the wall" before passing neatly through the window. Note that despite the need to gradually increase the cost weights, the quadrotor is not told *how* to pass the window. The final result is shown in Fig 5.

As becomes visible in Fig 5a, the solution differs from Fig 4. The quadrotor gently moves in front of the window, but instead of swinging up the load to fling it through the window, it only initiates a small swing and rapidly moves through the opening, pulling the load behind. The input gains

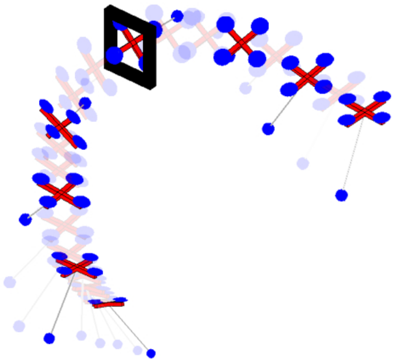


Fig. 6: Quadrotor passing through window at an angle

in Fig 5b are very low, and far from the saturation limits. Again, a mode switch occurs when passing the window.

D. Narrow Window Task

As a last case, the quadrotor has to pass a window which not only is not high enough for the cable to be vertical, but is also too narrow for the quadrotor to fly through horizontally. A window with $w_w = 0.45L_c$ and $w_h = 0.6L_c$ is used. Similar to the first window task, a solution is sought for which the load passes in front of the quadrotor, using the model description with the load position as states. An extra penalty is added to require the quadrotor to pass through with an angle $\phi > 0$. Fig 6 shows how the task is solved.

Instead of moving straight towards to window, the load is swung through the window sideways. The movement is such that no mode switch occurs. Surprisingly, this aggressive manoeuvre can be initialized with the LQR trajectory directly Fig 6 if good control weights are chosen.

Videos of the different simulations can be found at `https://www.youtube.com/user/ADRLabETH`.

V. CONCLUSIONS

The iLQG method was applied to a hybrid system consisting of a quadrotor with suspended load. In the simulations presented, the switching between the modes is handled well by the algorithm, but small stepwise changes of the penalty weights may be necessary to avoid divergence of iLQG. The simulations indicate that three main reasons for iLQG to diverge are immediate costs with either high peaks, non-quadratic costs, or the violation of the input saturation bounds. Short saturated control peaks are usually not problematic, but can destabilize the system if too long or too frequent. If the initial iLQG iterations are performed in a simulation where the input saturation can be turned off, it can help to do so in a first step to get an estimate of how much the saturation is violated, and adjust the input cost accordingly. Simply increasing the input penalty $\mathbf{C}_3$ often does not lead to the expected result, and can even lead to instability, as the quadrotor will focus on minimizing the input instead of solving the task. If a task requires more aggressive control, as is the case for the window task, better results are obtained by penalizing the quadrotor velocity. In general, this of course also reduces the input applied. Once a working trajectory is found, the algorithm is far more robust to changes in the input gain and velocity penalties.

A feasibility criterion however, in particular to know when a cost function is "too non-quadratic" for iLQG to handle, is not available and could be the topic of future works.

Acknowledgement

This research has been funded through a Swiss National Science Foundation Professorship award to Jonas Buchli.

Appendix

A. Cost Functions and Weights for the iLQG Tasks

Using the expressions as defined by (3) to (5), the different weights used are listed below.

1) Figure 8 Task:

$$\mathbf{C}_1 = diag\,(100, 100, 100, 2, 2, 0.1, 1, 1, 0, 50, 50, 50, 2, 2, 0.1, 0.2, 0.2, 0)$$
$$\mathbf{C}_2 = diag\,(0, 0, 0, 2, 2, 0.5, 1, 1, 0, 50, 50, 50, 2, 2, 2, 1, 1, 0)$$
$$\mathbf{C}_3 = diag\,(0.5, 0.5, 0.5, 0.5)$$
$$g(t,\mathbf{x},\mathbf{u}) = \frac{25}{8}\sum_{i=1}^{9} C_{wp}(t_i, \mathbf{x}_{p,i}, 100 \cdot diag(1,1,1), 4)$$

with $t_0 = 5[s]$, $\Delta t = 6\,[s]$, and $t_i = t_0 + i \cdot \Delta t$, $i = 0 \ldots 8$.

2) Guided Window Task:

Part of the cost is given in (6). The weights for the time independent state and control penalty used are

$$\mathbf{C}_1 = diag\,(50, 50, 50, 2, 2, 0.1, 1, 1, 0, 50, 50, 50, 2, 2, 0.1, 0.2, 0.2, 0)$$
$$\mathbf{C}_2 = \frac{1}{20} diag\,(0, 0, 0, 20, 20, 5, 10, 10, 0, 30, 50, 50, 20, 20, 20, 10, 10, 0)$$
$$\mathbf{C}_3 = diag\,(10, 10, 10, 10)$$

3) Non-Guided Window Task:

For this Task, several sets of parameters were used. The initial one, applied to the LQR trajectory, was

$$\mathbf{C}_1 = diag\,(100, 100, 100, 2, 2, 0.1, 1, 1, 0, 50, 50, 50, 2, 2, 0.1, 0.2, 0.2, 0)$$
$$\mathbf{C}_2 = \frac{1}{10} diag\,(0, 0, 0, 0, 0, 0, 10, 10, 0, 20, 20, 20, 10, 10, 10, 5, 5, 0)$$
$$\mathbf{C}_3 = diag\,(0.5, 0.5, 0.5, 0.5)$$

$$\begin{aligned} g(t,\mathbf{x},\mathbf{u}) &= 0.1 \cdot C_{sp,L}(t_1, \mathbf{x}_{win}, \mathbf{W}_{win}, 10) \\ w_{pos}^{\top} &= [4, 0, 1]\,, \quad t_1 = 0.5 \cdot t_f \\ \mathbf{x}_{win}^{\top} &= [\,w_{pos}(1), w_{pos}(2), w_{pos}(3), \ \ldots \\ &\quad 0, 0, 0, 0, 0, 0, 5, zeros(8,1)\,] \\ \mathbf{W}_{win} &= diag\,(100, 100, 100, zeros(1,6), 1, zeros(1,8)) \\ t_f &= 18\,[s], \quad \mathbf{x}_0^{\top} = [\,0, 0, 0\,]\,, \quad \mathbf{x}_{end}^{\top} = [\,8, 0, 0\,] \end{aligned}$$

After 5 iterations, the cost was changed to

$$\mathbf{C}_2 = \frac{1}{20} diag\,(0, 0, 0, 0, 0, 0, 10, 10, 0, 20, 20, 20, 10, 10, 10, 5, 5, 0)$$
$$g(t,\mathbf{x},\mathbf{u}) = 0.5 \cdot C_{sp,L}(t_1, \mathbf{x}_{win}, \mathbf{W}_{win}, 10)$$

The changes are marked in red, all the other costs remain the same. After 5 iterations, the cost for the quadrotor was added as

$$\begin{aligned} g(t,\mathbf{x},\mathbf{u}) &= 0.5 \cdot C_{sp,L}(t_1, \mathbf{x}_{win}, \mathbf{W}_{win}, 40) + \ldots \\ &\quad 0.5 \cdot C_{wp,Q}(t_1, \mathbf{x}_{wQ}, \mathbf{W}_{wQ}, 5) \\ \mathbf{x}_{wQ}^{\top} &= [\,0, w_{pos}(2), w_{pos}(3) + 0.5 w_h] \\ \mathbf{W}_{wQ} &= diag(0, 50, 50) \end{aligned}$$

Getting closer to the solution, the cost for the quadrotor can now be weighted more heavily, and a negative cost is added to prevent the load from crashing into the wall.

$$\mathbf{C}_2 = \frac{1}{20} diag\,(0, 0, 0, 0, 0, 0, 10, 10, 0, 5, 20, 20, 10, 10, 10, 5, 5, 0)$$
$$\mathbf{x}_{win}^{\top} = [\,w_{pos}(1), w_{pos}(2), w_{pos}(3), zeros(6,1), 5, zeros(8,1)\,]$$
$$\begin{aligned} g(t,\mathbf{x},\mathbf{u}) &= 0.5 \cdot C_{sp,L}(t_1, \mathbf{x}_{win}, \mathbf{W}_{win}, 40) + \ldots \\ &\quad 5 \cdot C_{wp,Q}(t_1, \mathbf{x}_{wQ}, \mathbf{W}_{wQ}, 5) - \ldots \\ &\quad 4 \cdot C_{wp,L}(t_1, w_{pos}, diag(10, 0, 0), 30) \end{aligned}$$

4) Narrow Window Task:

Parameters applied to LQR trajectory

$$\mathbf{C}_1 = diag\,(50, 50, 50, 2, 2, 0.1, 1, 1, 0, 50, 50, 50, 2, 2, 0.1, 0.2, 0.2, 0)$$
$$\mathbf{C}_2 = \frac{1}{20} diag\,(0, 0, 0, 20, 20, 5, 10, 10, 0, 15, 15, 40, 10, 10, 10, 10, 10, 0)$$
$$\mathbf{C}_3 = diag\,(7.8, 7.8, 7.8, 7.8)$$
$$\begin{aligned} g(t,\mathbf{x},\mathbf{u}) &= 0.04 \cdot C_{sp,L}(t_1, \mathbf{x}_{wL}, \mathbf{W}_{wL}, 25) + \ldots \\ &\quad 0.1 \cdot C_{sp,Q}(t_1 + 0.125, \mathbf{x}_{wQ}, \mathbf{W}_{wQ}, 10) \\ \mathbf{x}_{wL}^{\top} &= [\,w_{pos}(1), w_{pos}(2) - 0.2, w_{pos}(3) - 0.35 w_h, \ \ldots \\ &\quad 0, 0, 0, 0, -\frac{\pi}{2}, 0, 8, zeros(8,1)\,] \\ \mathbf{x}_{wQ}^{\top} &= [\,w_{pos}(1), w_{pos}(2) - 0.1, w_{pos}(3) - 0.25 w_h, \ \ldots \\ &\quad 0.5\pi, 0, 0, 0, 0, 0, zeros(9,1)\,] \\ w_{pos}^{\top} &= [4, 0, 2]\,, \quad t_1 = 0.35 \cdot t_f \\ \mathbf{W}_{wL} &= diag\,(350, 650, 650, 0, 0, 0, 0, 1000, 0, 50, zeros(1,8)) \\ \mathbf{W}_{wQ} &= diag\,(10, 600, 600, 1000, 0, 0, zeros(1,12)) \\ t_f &= 22\,[s], \quad \mathbf{x}_0^{\top} = [\,0, 0, 0\,]\,, \quad \mathbf{x}_{end}^{\top} = [\,12, 0, 0\,] \end{aligned}$$

References

[1] F. Farshidian, N. Neunert, and J. Buchli, "Learning of closed-loop motion control," 2014, in print: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS).

[2] D. Mellinger, Q. Lindsey, M. Shomin, and V. Kumar, "Design, modeling, estimation and control for aerial grasping and manipulation," in *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, 2011.

[3] A. Faust, I. Palunko, P. Cruz, R. Fierro, and L. Tapia, "Learning swing-free trajectories for uavs with a suspended load," in *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, 2013.

[4] I. Palunko, R. Fierro, and P. Cruz, "Trajectory generation for swing-free maneuvers of a quadrotor with suspended payload: A dynamic programming approach," in *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, 2012.

[5] I. Palunko, P. Cruz, and R. Fierro, "Agile load transportation : Safe and efficient load manipulation with aerial robots," *Robotics Automation Magazine, IEEE*, vol. 19, 2012.

[6] J. Schultz and T. Murphey, "Trajectory generation for underactuated control of a suspended mass," in *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, 2012.

[7] G. Starr, J. Wood, and R. Lumia, "Rapid transport of suspended payloads," in *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*, 2005.

[8] D. Zameroski, G. Starr, J. Wood, and R. Lumia, "Rapid swing-free transport of nonlinear payloads using dynamic programming," *Journal of Dynamic Systems, Measurement, and Control*, vol. 130, 2008.

[9] I. Palunko, A. Faust, P. Cruz, L. Tapia, and R. Fierro, "A reinforcement learning approach towards autonomous suspended load manipulation using aerial robots," in *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, 2013.

[10] K. Sreenath, N. Michael, and V. Kumar, "Trajectory generation and control of a quadrotor with a cable-suspended load - a differentially-flat hybrid system," in *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, 2013.

[11] K. Sreenath and V. Kumar, "Dynamics, control and planning for cooperative manipulation of payloads suspended by cables from multiple quadrotor robots," in *Robotics: Science and Systems (RSS)*, 2013.

[12] D. Mellinger, N. Michael, and V. Kumar, "Trajectory generation and control for precise aggressive maneuvers with quadrotors," *The International Journal of Robotics Research*, vol. 31, 2012.

[13] S. Tang, "Aggressive maneuvering of a quadrotor with a cable-suspended payload," Ph.D. qualifiers report, University of Pennsylvania Philadelphia, PA, 2014. [Online]. Available: http://www.seas.upenn.edu/~sytang/docs/2014QualifierReport.pdf

[14] D. Mellinger, M. Shomin, and V. Kumar, "Control of quadrotors for robust perching and landing," in *Proc. Int. Powered Lift Conf*, 2010.

[15] E. Todorov and W. Li, "A generalized iterative lqg method for locally-optimal feedback controlof constrained nonlinear stochastic systems," in *IEEE, American Control Conference, 2005.*, 2005.

[16] I. Mordatch, E. Todorov, and Z. Popović, "Discovery of complex behaviors through contact-invariant optimization," *ACM Trans. Graph.*, vol. 31, no. 4, pp. 43:1–43:8, 2012.