Balancing an Inverted Pendulum on a Quadrotor: A Model Predictive Approach

Alexander Berndt, Matthijs Bekendam,

***Abstract*—A model predictive control (MPC) approach for an inverted pendulum balanced on a quadrotor is implemented and simulated. Various MPC controllers are compared to LQR control for different scenarios. We consider a state-based MPC, output MPC with reference tracking and disturbance rejection, as well as a rotational trajectory where we use an adaptive MPC scheme. A stability analysis of the quadrotor is performed to prove stability of the MPC controller.**

I. Introduction

This project involves the problem of balancing an inverted pendulum using a quadrotor. The dynamics are mostly based on work in [1], but are expanded in three ways. Firstly, we consider each individual rotor thrust input as well as the quadrotor's rotational inertia, resulting in a larger, coupled MIMO system, as opposed to a system comprising independent subsystems obtained by using the rotational angle rates as inputs. Secondly, we focus on a model predictive control design with state and input constraints as opposed to the simpler LQR design. Thirdly, we also consider a time-varying equilibrium point which is tackled using an adaptive MPC scheme.

A. Quadrotor and Pendulum Dynamics

Using Lagrangian mechanics, the nonlinear dynamic equations describing the quadrotor-pendulum interaction can be derived as shown in [1]. The main equations governing the motion are.

$$\begin{bmatrix}\ddot{x}\\ \ddot{y}\\ \ddot{z}\end{bmatrix} =^{O}_{V} R(\alpha,\beta,\gamma)\begin{bmatrix}0\\0\\a\end{bmatrix}+\begin{bmatrix}0\\0\\-g\end{bmatrix} \tag{1}$$

$$\begin{bmatrix}\dot{\gamma}\\ \dot{\beta}\\ \dot{\alpha}\end{bmatrix} = \begin{bmatrix}\cos(\beta)\cos(\gamma) & -\sin(\gamma) & 0\\ \cos(\beta)\sin(\gamma) & \cos(\gamma) & 0\\ -\sin(\beta) & 0 & 1\end{bmatrix}^{-1}\begin{bmatrix}\omega_x\\ \omega_y\\ \omega_z\end{bmatrix} \tag{2}$$

$$\begin{bmatrix}\ddot{r}\\ \ddot{s}\end{bmatrix} = \mathbf{f}(r,s,\dot{r},\dot{s},\ddot{x},\ddot{y},\ddot{z}) \tag{3}$$

The function $\mathbf{f}$ of Equation 3 is defined in Equation 13-14 in [1]. The rotational velocity and relative upwards velocity as seen from $\mathcal{V}$ can be related to the individual motor thrusts as

$$\begin{bmatrix}\omega_x\\ \omega_y\\ \omega_z\\ \dot{v}_z\end{bmatrix} = \begin{bmatrix}-I_{xx}^{-1} & I_{xx}^{-1} & 0 & 0\\ 0 & 0 & -I_{xx}^{-1} & I_{xx}^{-1}\\ -I_{zz}^{-1} & -I_{zz}^{-1} & I_{zz}^{-1} & I_{zz}^{-1}\\ m^{-1} & m^{-1} & m^{-1} & m^{-1}\end{bmatrix}\begin{bmatrix}F_1\\F_2\\F_3\\F_4\end{bmatrix} \tag{4}$$

Since the focus of this assignment is the design of model-predictive controllers, the derivations and coordinate transformations used to obtain Equations 1-4 are omitted here. The reader is referred to [1] [2] for further details.

B. Linearized System Dynamics

Equations 1-4 can be linearized around the $(x,u)=(0,0)$ equilibrium point. This results in the familiar LTI form

$$\begin{aligned}\dot{x} &= Ax+Bu\\ y &= Cx\end{aligned} \tag{5}$$

The matrix C represents the sensor array and is either identity for state-feedback control, or a subset of the states as detailed in the output-MPC design. The system has 16 states where the states are

- r, $\dot{r}$: relative x-displacement and velocity of pendulum
- x, $\dot{x}$: x-displacement and velocity of quadrotor's COM
- β, $\dot{\beta}$: pitch angle and rate of quadrotor body
- s, $\dot{s}$: relative y-displacement and velocity of pendulum
- y, $\dot{y}$: y-displacement and velocity of quadrotor's COM
- γ, $\dot{\gamma}$: roll angle and rate of quadrotor body
- z, $\dot{z}$: z-displacement and velocity of quadrotor's COM
- α, $\dot{\alpha}$: yaw angle and rate of quadrotor body

$$x = \begin{bmatrix}r\\ \dot{r}\\ x\\ \dot{x}\\ \beta\\ \dot{\beta}\\ s\\ \dot{s}\\ y\\ \dot{y}\\ \gamma\\ \dot{\gamma}\\ z\\ \dot{z}\\ \alpha\\ \dot{\alpha}\end{bmatrix} \qquad u = \begin{bmatrix}T_1\\T_2\\T_3\\T_4\end{bmatrix} \tag{6}$$

Alexander Berndt and Matthijs Bekendam are master students in the MSc Systems and Control at TU Delft, The Netherlands. E-mail addresses: {a.e.berndt, m.bekendam}@student.tudelft.nl.

C. Time-Varying Trajectory Tracking

Using a time-varying coordinate transformation $R_z(\Omega t)$, a new time invariant coordinate reference $\mathcal{C}$ is defined with (u, v, w) and (η, μ, ν) describing the position and euler angles in $\mathcal{C}$ respectively

$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} = \begin{bmatrix} \cos(\Omega t) & -\sin(\Omega t) & 0 \\ \sin(\Omega t) & \cos(\Omega t) & 0 \\ 0 & 0 & 1 \end{bmatrix}^{-1} \begin{bmatrix} x \\ y \\ z \end{bmatrix} \tag{7}$$

Linearizing the system around a rotational trajectory of radius R with rotational velocity Ω yields

$$\begin{aligned} \ddot{\tilde{p}} &= \frac{\zeta_0^2}{L^2}\Big[\tilde{p}\Big(\Omega^2 + \tfrac{gL^2}{\zeta_0^3}\Big) + 2\dot{\tilde{q}}\Omega + \tilde{\mu}() + \tilde{a}()\Big] \\ \ddot{\tilde{q}} &= \tilde{q}(\Omega^2 + \frac{g}{\zeta_0}) - 2\dot{\tilde{p}}\Omega + \tilde{\nu}\alpha_0 \\ \ddot{\tilde{u}} &= \tilde{\alpha}\sin\mu_0 + \tilde{\mu}\alpha_0\cos\mu_0 + 2\dot{\tilde{v}}\Omega + \tilde{u}\Omega^2 \\ \ddot{\tilde{v}} &= -\tilde{\nu}\alpha_0 - 2\dot{\tilde{u}}\Omega + \tilde{v}\Omega^2 \\ \ddot{\tilde{w}} &= \tilde{\alpha}\cos\mu_0 - \tilde{\mu}\alpha_0\sin\mu_0 \end{aligned} \tag{8}$$

the details of which are specified in [1]. The equations can be condensed into the linear state-space equations

$$\dot{e} = A\big(R(t), \Omega(t)\big)e + B\big(R(t), \Omega(t)\big)u \tag{9}$$

where e represents the deviation from the nominal trajectory and u the change in control input at this nominal trajectory.

D. Dynamics Summary

In summary, the inverted-pendulum-quadrotor system is modelled as a linearizable nonlinear system in two different reference frames:

1) Static reference frame represented by $\mathcal{V}$ corresponding to an equilibrium point of $(x, u) = (0, 0)$
2) Dynamic reference frame which is time invariant when viewed in the reference frame $\mathcal{C}$. This reference represents the deviation of the quadrotor and inverse pendulum from a circular trajectory of radius $R(t)$, rotational velocity $\Omega(t)$ and altitude $z_0(t)$.

II. Model Predictive Control Design

Based on the dynamics presented in Section I, different MPC designs were designed to control the quadrotor-pendulum system. The three different designs were done for three different scenarios:

1) Regulation of system in coordinate frame $\mathcal{V}$ using regular state-based MPC
2) Output MPC design in coordinate frame $\mathcal{V}$ with disturbance rejection
3) Adaptive MPC control design for the circular trajectory tracking represented in coordinate frame $\mathcal{C}$ with parameter dependent state-space matrices

The general cost function used to solve the MPC problem at each iteration is given in [3] and is defined as follows

$$\begin{aligned} J(x_0, u) &= \sum_{k=0}^{N-1} \ell\big(x(k), u(k)\big) + V_f\big(x(N)\big) \\ &\text{s.t. } u \in \mathbb{U}, x \in \mathbb{X} \end{aligned} \tag{10}$$

A. Regulation MPC

In this case, full-state information is assumed, and the goal is to stabilize the system as described in the $\mathcal{V}$ reference frame. The 0 equilibrium represents a constant hovering point with the pendulum vertically balanced on top of the quadrotor. The MPC optimization problem to be solved at each iteration is represented by Equation 10 with the stage and terminal costs defined as

$$\begin{aligned} \ell\big(x(k), u(k)\big) &= x(k)^T Q x(k) + u(k)^T R u(k) \\ V_f\big(x(N)\big) &= x(N)^T P x(N) \end{aligned} \tag{11}$$

where $Q = 10I$, $R = 0.1I$ and P is the solution of the discrete algebraic Riccati equation (DARE) used to solve the unconstrained infinite-horizon LQR problem. The choices of Q and R were obtained following an iterative tuning procedure which would ensure a balance between control action and settling-time. This tuning procedure is illustrated in Section IV. P is the DARE solution as it would help the stability proof in Section III. Note that Q, R and P are positive definite.

The control inputs represent the scaled thrust available to each of the four rotors. Note that since the control inputs u are effectively a change in input offset by a nominal input u_0 equal to a quarter of the quadrotor's weight, the control thrust will never be negative. Using the thrust limits obtained in experimental data in [2], the scaled inputs are bounded by 0.1. The input constraints $u \in \mathbb{U}$ in Equation 10 are therefore defined as

$$|u_i| \leq T_{\text{max}} = 0.1 \quad \forall i \in [1, .., 4N] \tag{12}$$

To ensure the linearized system is sufficiently representative of the real non-linear system, the state constraints can be used to ensure the state trajectories do not allow states which are dominated by the system's non linearities. This is done by placing limits on the angles and velocities in the state $x(0)$. The roll and pitch angles are limited to 0.5 rad and the velocities limited to 4 m/s to avoid aerodynamic affects. The pendulum displacements are limited to 0.25m which correspond to an angle of 28^o. No constraints are posed on the positions. These constraints are represented by $x_{\text{i, limit}}$. corresponding to the i'th state

$$|x_i(k)| \leq x_{\text{i, limit}} \quad \forall i \in \{1, 4, 5, 7, 10, 11, 14\}, k \in [0, .., N] \tag{13}$$

Since the MPC problem is ultimately an optimization on the control inputs, the state constraints $x \in \mathbb{X}$ are related to the input constraints by using the prediction relationship $x_N = Px_0 + Zu_N$. The state constraints are therefore defined as

$$
\begin{aligned}
&|Px_0 + Zu_N| \leq x_{\text{limit}} \\
\implies &\begin{cases} Zu_N \leq -Px_0 + x_{\text{limit}} \\ Zu_N \geq -Px_0 - x_{\text{limit}} \end{cases}
\end{aligned} \tag{14}
$$

The continuous time dynamics are discretized with a sampling time of 0.1 seconds. This ensures the discrete system is still controllable and observable. The discretization method used was zero-order hold to simulate the usual DAC and ADC sensors typically used in these applications.

The prediction horizon N was chosen to be 20, which with a sampling time of 0.1 equates to a horizon of 2 seconds (approximately the settling time of the closed-loop system). The horizon was chosen based on iterating through different sampling times and horizon times which would ensure the discretized system is controllable and the online MPC problem is not too computationally demanding.

B. Output MPC and Disturbance Rejection

To simulate output-based control, an output MPC controller was designed. The motivation behind this is to simulate limited direct access to state information via sensor hardware. As with the experiments in [2], the sensor array is represented by a motion capture system and an on-board IMU. This means the following can be measured:

1) Pendulum relative position r, s
2) Quadrotor absolute position x, y, z
3) Roll and pitch angular rates $\dot{\beta}$, $\dot{\gamma}$
4) Yaw angle α

This sensor array represents an observable state-space system meaning that the state measurement errors asymptotically converge to zero. In order to add disturbance rejection properties to the closed-loop system, an augmented state-space system is required. This augmented state-space system can be used to estimate the disturbances acting on the system in order to counter them as explained in Section 5.5 of [3]. The augmented state-space system is represented by

$$
\begin{aligned}
\begin{bmatrix} x^+ \\ d^+ \end{bmatrix} &= \begin{bmatrix} A & B_d \\ 0 & I \end{bmatrix} \begin{bmatrix} x \\ d \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} u + \begin{bmatrix} w \\ w_d \end{bmatrix} \\
y &= \begin{bmatrix} C & C_d \end{bmatrix} \begin{bmatrix} x \\ d \end{bmatrix} + v
\end{aligned} \tag{15}
$$

where the matrices B_d and C_d are used to represent the affect of a constant disturbance on the x and y states, simulating a constant side-wind acting on the quadrotor. In order to ensure the augmented system is observable, it is required that the original (A, C) system is observable as well as

$$
\text{rank} \begin{bmatrix} I - A & -B_d \\ C & C_d \end{bmatrix} = n + n_d \tag{16}
$$

where Equation 16 is satisfied with $n + n_d = 18$. For reference tracking and disturbance rejection, an optimal target selection (OTS) needs to be solved online to obtain the required x_{ref} and u_{ref} to reject disturbances on x and y. The OTS problem to be solved at each iteration is described as in Equation 17.

$$
(x_{\text{ref}}, u_{\text{ref}})(\hat{d}, y_{\text{ref}}) \in \begin{cases} \arg\min_{x_r, u_r} J(x_r, u_r) \\ s.t. \begin{bmatrix} I - A & -B \\ C & 0 \end{bmatrix} \begin{bmatrix} x_r \\ u_r \end{bmatrix} = \begin{bmatrix} 0 \\ y_{\text{ref}} - \hat{d} \end{bmatrix} \\ (x_r, u_r) \in \mathbb{Z} \\ Cx_r + \hat{d} \in \mathbb{Y} \end{cases} \tag{17}
$$

Based on information of typical sensor noise described in [2], a Kalman filter is used to obtain the state-observer gain with process and measurement covariance matrices set to $Q_{KF} = I$ and $R_{KF} = I$ respectively. In a real experiment, iterative tuning of the weighting matrices would be required to ensure adequate noise filtering. In this case, Q_{KF} and R_{KF} were chosen just to ensure the observer error converges faster to zero than the closed-loop system.

With the references x_{ref} and u_{ref} solved using the OTS at each iteration, the reference tracking MPC problem as described in Equation 10 is solved. The stage and terminal costs are just adjusted to account for the reference signals as described in Equation 18.

$$
\begin{aligned}
\ell\big(x(k), u(k)\big) &= \big(x(k) - x_{\text{ref}}\big)^T Q \big(x(k) - x_{\text{ref}}\big) \\
&\quad + \big(u(k) - u_{\text{ref}}\big)^T R \big(u(k) - u_{\text{ref}}\big) \\
V_f\big(x(N)\big) &= \big(x(N) - x_{\text{ref}}\big)^T P \big(x(N) - x_{\text{ref}}\big)
\end{aligned} \tag{18}
$$

Following a similar procedure as described for the Regulation MPC design, the cost weighting matrices were chosen to be $Q = 10I$, $R = I$ and P is the solution to the DARE of the unconstrained problem. Finally, the prediction horizon N was selected to be 20 with a sampling time of 0.1 seconds for the same reasons as explained in the Regulation MPC design.

C. Adaptive MPC for Time-Varying Trajectory

Equation 9 represents the quadrotor and pendulum in the dynamic coordinate frame $\mathcal{C}$ which is a parameter-dependent linear state-space system. In this case, the dynamic coordinate frame depends on desired trajectory radius and rotational velocity. An adaptive MPC controller was designed to account for time-varying circle radius R and rotational velocities Ω.

The MPC problem to be solved at each iteration is the same as defined in Equation 10 for the Regulation MPC case, expect that here, the prediction matrices need to be recalculated depending on the current desired radius and rotational velocity as shown in Equation 19.

$$\begin{bmatrix} x(0) \\ x(1) \\ \vdots \\ x(N) \end{bmatrix} = \underbrace{\begin{bmatrix} I \\ A(R,\Omega) \\ \vdots \\ A(R,\Omega)^{N-1} \end{bmatrix}}_{T(R,\Omega)} x(0) + \underbrace{\begin{bmatrix} 0 & 0 & \dots & 0 \\ B(R,\Omega) & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ A(R,\Omega)^{N-2}B(R,\Omega) & A(R,\Omega)^{N-3}B(R,\Omega) & \dots & 0 \end{bmatrix}}_{Z(R,\Omega)} \begin{bmatrix} u(0) \\ u(1) \\ \vdots \\ u(N) \end{bmatrix} \tag{19}$$

The MPC problem is solved based on the same quadratic cost stage and terminal cost function of the Regulation MPC as defined in Equation 11 with $Q = 15I$, $R = 0.8I$. The terminal cost was chosen to be $P = 20I$ to avoid solving the *DARE* at each iteration.

III. Asymptotic stability

In this section, the designed regulation MPC controller is shown to asymptotically stabilize the closed-loop system provided the system's initial conditions are sufficiently close to the origin.

A. Linearized System Stability Analysis

Considering the linearized system

$$x^+ = Ax + Bu \tag{20}$$

where A and B represent the linearized system dynamics derived in Section I. The linearized system (A, B) is controllable since $\mathcal{W}_c = \begin{bmatrix} B & AB & A^2B & .. & A^{15}B \end{bmatrix}$ is full rank. With stage cost $\ell(x,u) = \frac{1}{2}\left(x^TQx + u^TRu\right)$. By selecting $Q = qI, q > 0$ and $R = rI, r > 0$, which are hence positive definite.

The terminal cost is chosen to be the solution P of the *DARE* with the aforementioned Q and R.

$$V_f(x) = \tfrac{1}{2}x^TPx \tag{21}$$

As described in Section 2.5.4 of [3] (which discusses the linear MPC case with quadratic cost functions), based on Assumptions 2.2, 2.3 and 2.14 in [3], by Theorems 2.19 and 2.21, the origin of the aforementioned system design is exponentially stable in $\mathcal{X}_N$.

Since the estimation of $\mathbb{X}_f$ yields a linear constraint set $Hx \leq h$ and the quadratic assumption of this space $x^TPx \leq c$ is very small, we decided to remove the quadratic terminal constraint $x(N) \in \mathbb{X}_f$ and instead use the β stability formulation as defined in Section 2.6 of [3]. Following an iterative analysis, it was found that choosing $\beta = 1$ would still result in a system which is steered to $\mathbb{X}_f$ in N steps. Hence, by Theorem 2.41, the origin is asymptotically stable for the closed-loop system.

Assumption 2.2 is satisfied since the function f is linear and $f(0,0) = 0$, $\ell(0,0) = 0$ and $V_f(0) = 0$ and all cost functions are positive definite. Also, $\ell(x,u) \geq q||x||^2 + r||u||^2$ and $V_f(x) \geq p||x||^2$ for $q = 10$, $r = 0.1$ and $p = \min \text{eig}(P) = 10.01$.

Assumption 2.3 is satisfied since the constraints $x \in \mathbb{X}$, $u \in \mathbb{U}$ both are closed, compact and contain the origin in its interior as discussed in Section II.

Assumption 2.14 (a) is shown to hold a posteriori. (b) holds as discussed for Assumption 2.2.

B. Estimating $\mathbb{X}_f$

Having proven stability for a the linear system, the goal is to find the maximum invariant constraint admissible set $\mathbb{X}_f$ and the region of attraction $\mathcal{X}_N$ such that initial conditions starting in $\mathcal{X}_N$ will be guaranteed to converge towards the origin asymptotically.

$\mathbb{X}_f$ is the set which ensures the MPC optimal control input will be the same as the unconstrained control input defined by the infinite-horizon LQR problem. Within this set $\mathcal{K}_N(x) = Kx$ where K is the optimal LQR gain.

By extending Algorithm 3.2 presented in [4], an invariant set $\mathbb{X}_f$ can be estimated which adheres to the control input and state constraints. This is done by replacing the state-to-output mapping C with the augmented state-to-input and state-to-state mapping $[K; I]$. Instead of finding an output admissible set as in [4], the new algorithm will find a state and input admissible invariant set which will contain all states which ensure the unconstrained LQR control law will keep the state therein. This algorithm is shown in Appendix A.

Using Algorithm 1, the set $\mathbb{X}_f$ is found to contain 320 linear inequalities represented by $Hx \leq h$. $\mathbb{X}_f$ is formally defined as

$$\mathbb{X}_f = \left\{x \in \mathbb{R}^{16} \mid Hx \leq h\right\}$$

C. Using $\mathbb{X}_f$ to estimate $\mathcal{X}_N$

With an estimation of $\mathbb{X}_f$, an empirical analysis is done to reconstruct $\mathcal{X}_N$. This is done by considering a large set of initial conditions x_0 and checking the feasibility of the MPC problem with the terminal constraint $x(N) \in \mathbb{X}_f$. If the problem is feasible, $x(0)$ lies in $\mathcal{X}_N$.

Since the system has 16 states, the region $\mathcal{X}_N$ cannot be plotted, but a 2-D projection of the 16-D $\mathcal{X}_N$ can be constructed. Figure 1 shows a 2-D plot of initial conditions, in this case, the pendulum's initial x-displacement r and velocity $\dot{r}$, which can be steered to $\mathbb{X}_f$ in $N = 20$ steps.

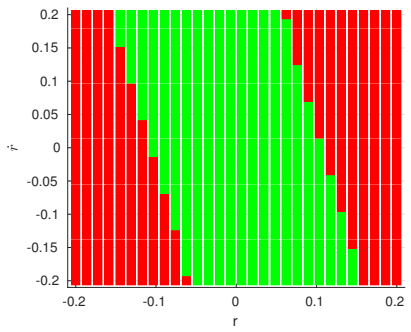

Fig. 1. 2-D mapping of the estimate of $\mathcal{X}_N$ for different r and $\dot{r}$ initial conditions. Green/red indicates an IC within/outside $\mathcal{X}_N$

Similar figures can be obtained for other combinations of varied initial conditions. The reader is referred to the repository in [5] for more plots like Figure 1.

Initially, the goal was to construct a convex set estimating $\mathcal{X}_N$. However, since the quadrotor-pendulum system has 16 states, even only considering 3 different values per state will result in $3^{16} = 43046721$ calculations. And even then, a resolution of 3 values per state would not provide an accurate estimate of the true $\mathcal{X}_N$. The obtained solution can however validate whether a certain initial condition lies in $\mathcal{X}_N$ or not.

D. Confirmation of a-priori assumptions

For the stability proof, one of the key assumptions was a Lyapunov decrease as stated in Assumption 2.14 in [3] and repeated in Equation 22. For all $x \in \mathbb{X}_f$ there exists a u $((x,u) \in \mathbb{Z})$ satisfying

$$V_f(f(x,u)) - V_f(x) \leq -\ell(x,u) \tag{22}$$

Figure 2 shows Equation 22 holds for an initial condition at the edge of $\mathbb{X}_f$. The same result was found for all other initial conditions tested within $\mathbb{X}_f$.

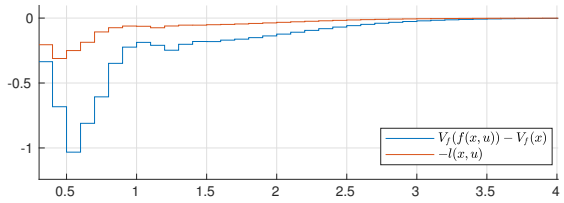

Fig. 2. A posteriori confirmation of Assumption 2.14 of [3]. $V_f(x) = x^T P x$ and $\ell(x,u) = \frac{1}{2}(x^T Q x + u^T R u)$

Additionally, the assumption that $\mathbb{X}_f$ is invariant was also confirmed a posteriori, ensuring that for all $x \in \mathbb{X}_f$ there exists a u $((x,u) \in \mathbb{Z})$ satisfying

$$f(x,u) \in \mathbb{X}_f \quad \forall x \in \mathbb{X}_f$$

This was confirmed by considering the state trajectories and ensuring a posteriori that all states within $\mathbb{X}_f$ remain in $\mathbb{X}_f$ using the $Hx \leq h$ inequality obtained using Algorithm 1.

IV. Numerical simulations

In this section, we show several numerical simulations where we compare the various MPC controllers shown in Section II as well as LQR control used in [1].

A. Prediction Horizon N

Figure 3 shows the system response for different prediction horizons. Note that for $N = 3$ and $N = 5$, the system is unstable. The choice of $N = 10$ and $N = 20$ clearly shows a stable response. Despite minimal differences in the response for $N = 10$ and $N = 20$, a larger N ensures a larger region of attraction $\mathcal{X}_N$. Therefore, the prediction horizon was kept at $N = 20$.

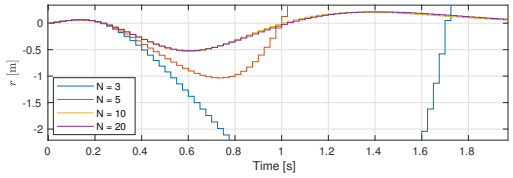

Fig. 3. System response for different prediction horizons N

B. Regulation MPC vs Constrained LQR

Given the defined input constraints based on the maximum available thrust, the closed-loop system using constrained LQR and MPC can be compared. The constrained LQR system was tuned until the output was stable despite the input saturation. This is then compared to the system's response using constrained MPC. Figure 4 shows the vast improvement shown by MPC over LQR for an initial condition outside the terminal set $\mathbb{X}_f$.

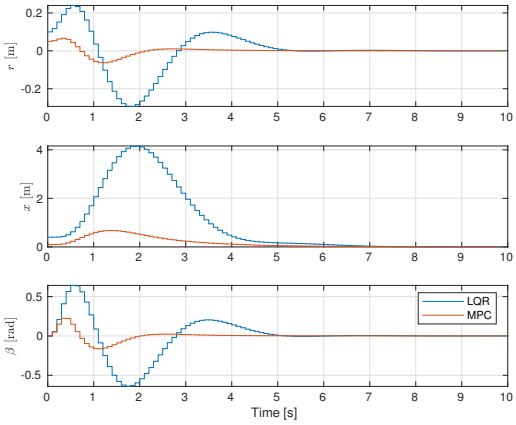

Fig. 4. Comparison between MPC and constrained LQR for initial condition $x(0)$ outside $\mathbb{X}_f$

On the other hand, Figure 5 shows the MPC and LQR control with an initial condition within the set $\mathbb{X}_f$. As was intended with the MPC design and *DARE* terminal cost, the system response is virtually the same as that of the unconstrained LQR problem.

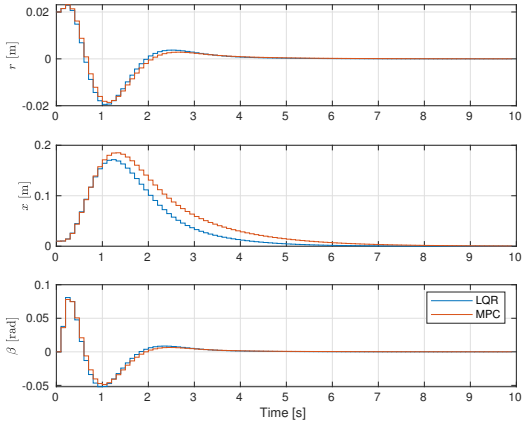

Fig. 5. Comparison between MPC and constrained LQR for initial condition $x(0)$ in $\mathbb{X}_f$

C. Regulation MPC with Different Cost Function Weightings

The regularizing MPC controller is used to illustrate the effect different cost function choices have on the closed-loop system's performance. Figure 6 shows the states related to the quadrotor and pendulum in the x-direction converging to the origin with an initial condition outside $\mathbb{X}_f$ but within $\mathcal{X}_N$.

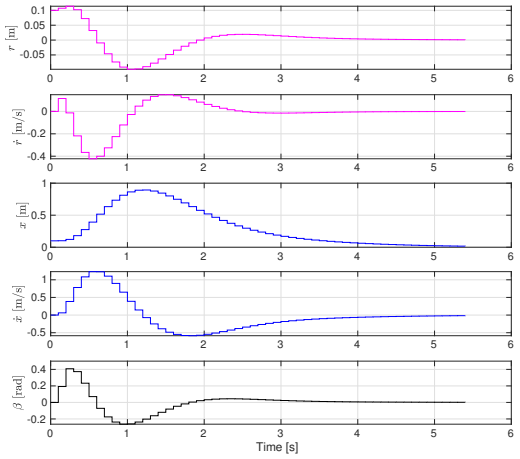

Fig. 6. Asymptotic convergence of the pendulum's position and velocity (r and $\dot{r}$), quadrotor's position, velocity and pitch angle (x, $\dot{x}$, β)

Figures 7, 9 and 8 show the stabilization of the quadrotor using various Q and P weighting matrices respectively. These parameters are used in the MPC cost function as defined in Equation 10. To ensure brevity, only x-direction states are plotted since y-direction states are very similar and not necessary for this comparison.

It is clear from Figure 7 that small state costs result in longer settling times and more overshoot as opposed to larger costs. However, it seems that the stage cost seems to saturate at some point (approximately $Q = 0.1I$). This could be explained by the fact that large inputs are required to ensure

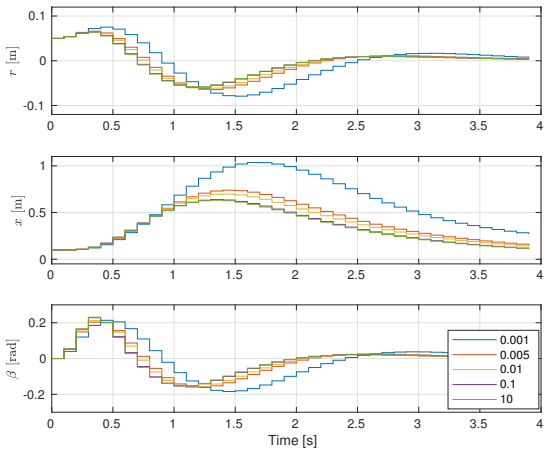

Fig. 7. State trajectories for different state cost matrices Q. $Q = qI$ where the legend refers to the constant q. The states are pendulum displacement r, quadrotor displacement x and pitch angle β.

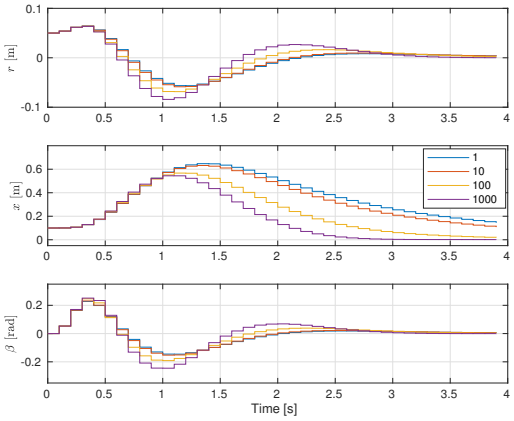

Fig. 8. State trajectories for different terminal cost matrices P. $P = pI$ where the legend refers to the constant p. The states are pendulum displacement r, quadrotor displacement x and pitch angle β.

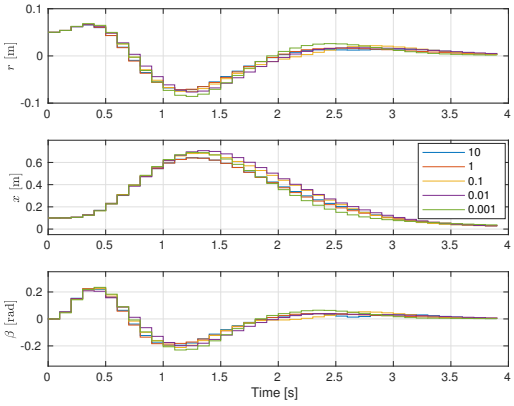

Fig. 9. State trajectories for different input R cost matrices. $R = rI$ where the legend refers to the constant r. The states are pendulum displacement r, quadrotor displacement x and pitch angle β.

smaller state errors, but the input constraints $u \in \mathbb{U}$ prevent this from happening, resulting in no effective change for higher state costs.

Figure 9 shows the state trajectories for different $R = rI$ matrices with the stage state cost Q and terminal cost P constant at I. Figure 10 indicates that the input sequences jump around for smaller values of R, and are more constant for larger values of R.

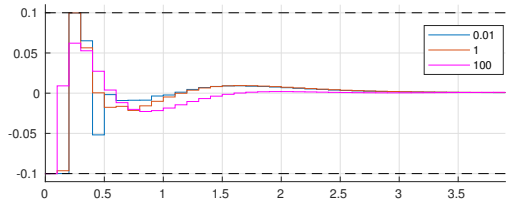

Fig. 10. Input sequences for one of the rotors for different input cost R matrices. $R = rI$ where the legend refers to the constant r. Q and P are constant and equal to I.

D. Output MPC and Disturbance Rejection using Observer

Assuming we only have access to the quadrotor's position (x, y, z), pendulum position (r, s) and the euler angle rates, we can use an observer to reconstruct the states for use in MPC control. As detailed in Section II, this observer is augmented with a disturbance observer for steps in the x and y directions to simulate a constant lateral offset acting on the quadrotor.

Figure 11 shows a simulation where a disturbance is acting on the quadrotor at two distinct time-spans $t = [1, 2]$ sec and $t = [10, 15]$ sec. Additionally, a step reference on the x state is started at $t = 5$ sec. The figure shows how the output MPC controller manages to reject the disturbance and track the input reference.

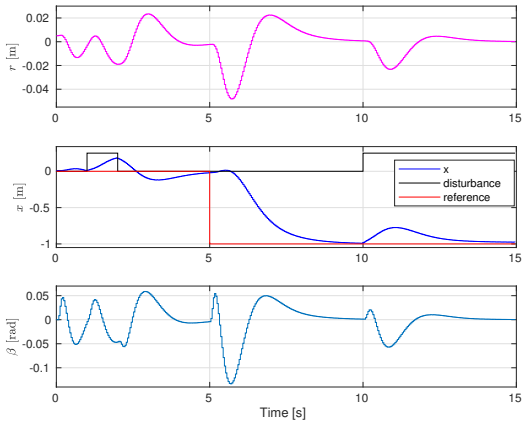

Fig. 11. Reference tracking and disturbance rejection using output MPC. A disturbance on x is active during $t = [1, 2]$ sec and $t = [10, 15]$ sec. A step reference on x is started at $t = 5$ sec

Figure 12 shows the observer's estimate of the disturbance acting on the x and y states. Note the settling time of the observer on the disturbances and states of approximately 0.2 seconds, which is considerably faster than the system's settling time of approximately 3 seconds, implying the observer gains were well chosen.

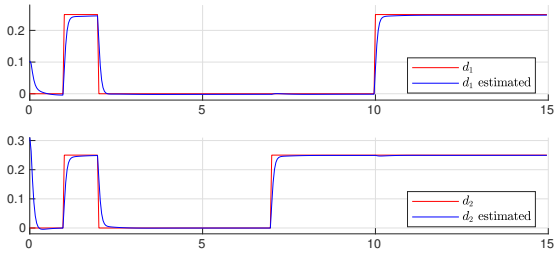

Fig. 12. Plot showing the estimate of the disturbance acting on the x and y states as well as the actual step disturbance acting on these states

Figure 13 shows the change in the reference states x_{ref} as a step disturbance acts on the system's x and y displacement states during the time span $t = [1, 2]$ sec.

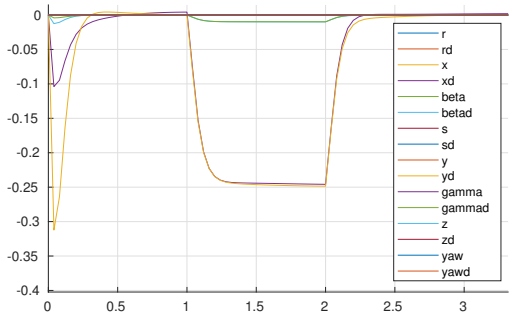

Fig. 13. Reference states x_{ref} determined by the OTS optimization under influence of the disturbance acting on the system's x and y states during $t = [1, 2]$ sec

E. Adaptive MPC

Figure 14 shows the trajectory followed by the quadrotor and pendulum using the adaptive MPC controller. The quadrotor's desired trajectory is denoted by the blue line and the pendulum's position by the pink line. Note how the pendulum's trajectory (pink curve) initially moves away from the blue line due to the initial conditions imposed on the quadrotor and pendulum. The desired rotational velocity was increased with time, resulting in a nominal pendulum position angled more towards the center of the circle.

Figure 15 shows the change in the euler angles as seen from the quadrotor's reference frame $\mathcal{C}$. The relative pitch angle decreases with time since the desired rotational velocity Ω increases with time.

V. Conclusion

The quadrotor with inverted pendulum proved to be a fun but challenging system to control, mostly due to its complicated dynamics and large number of states. Due to the

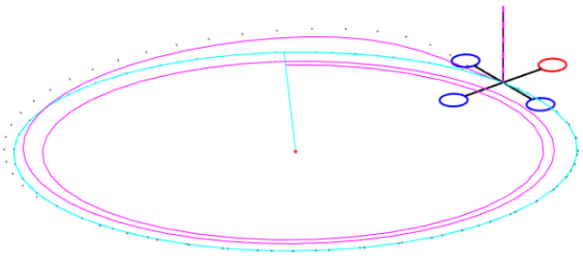

Fig. 14. 3D simulation of adaptive MPC controller following a rotational trajectory with increasing rotational velocity. blue is the desired quadrotor trajectory, pink is the pendulum position (which moves inwards as velocity increases)

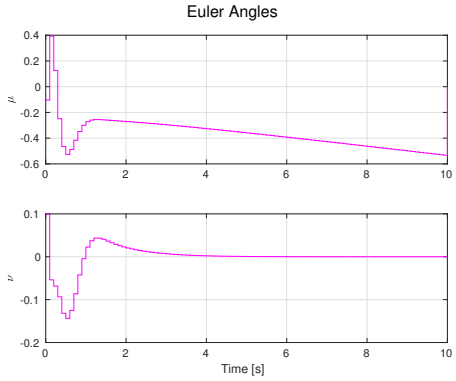

Fig. 15. Quadrotor relative euler angles. The relative pitch angle changes due to the adjusted linearization for increasing rotational velocity Ω

highly nonlinear dynamics, as well as the intrinsic coupling between inputs and states, the nonlinear system proved too difficult to prove stability when using MPC. The linear system, however, was proven to be stable within a region of attraction.

Implementing different MPC designs (state-based MPC, output-MPC with disturbance rejection and adaptive MPC) resulted in good insight into the intricacies of MPC controller design and the effects different design choices have on the closed-system response.

REFERENCES

[1] M. Hehn and R. D'Andrea, "A flying inverted pendulum," in *2011 IEEE International Conference on Robotics and Automation*, May 2011, pp. 763–770.

[2] M. W. Mueller and R. D'Andrea, "Stability and control of a quadrocopter despite the complete loss of one, two, or three propellers," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, May 2014, pp. 45–52.

[3] J. Rawlings and D. Mayne, *Model Predictive Control: Theory and Design*. Nob Hill Publishing, 2008.

[4] E. G. Gilbert and K. T. Tan, "Linear systems with state and control constraints: the theory and application of maximal output admissible sets," *IEEE Transactions on Automatic Control*, vol. 36, no. 9, pp. 1008–1020, Sep. 1991.

[5] A. E. Berndt and J. M. Bekendam, "Quadrotor-balancing-pendulum-mpc," https://github.com/alexberndt/Quadrotor-Balancing-Pendulum-MPC, 2019.

APPENDIX

A. Algorithms to Estimate $\mathbb{X}_f$ and $\mathcal{X}_N$

Extension of Algorithm 3.2 in [4] to approximate a maximal invariant control admissible set $\mathbb{X}_f$.

Algorithm 1: Estimate Set $\mathbb{X}_f$

Result: $Hx \leq h$ representing $\mathbb{X}_f$
Initialization:
$A_K := A - BK$
K := LQR optimal gain
K_{aug} := [K; I]
Set $k := 0$
Iteration:
For all $i = 1, 2, .., s$

$$x_i^* := \begin{cases} \underset{x}{\text{argmax}} \;\; f_i(K_{\text{aug}} A_K^{k+1} x) \\ \text{s.t. } f_j(KA^t x) \leq 0 \;\; \forall j \in \{1, 2, .., s\}, \\ \qquad\qquad\qquad \forall t \in \{0, 1, .., k\} \end{cases}$$

end
if $\text{f}_i(K_{\text{aug}} A_K^{k+1} x) \leq 0 \; \forall i \in \{1, 2, .., s\}$
then

$$\mathbb{X}_f := \{x \in \mathbb{R}^n \mid f_i(K_{\text{aug}} A_K^t x) \leq 0 \forall j \in \{1, 2, .., s\}, \; \forall t \in \{0, 1, .., k\}\}$$

else
set k := k + 1 and continue

Algorithm 2: Estimate Terminal Set $\mathcal{X}_N$

Result: An approximation of $\mathcal{X}_N$ represented by $\mathcal{X}'_N$
Initialization:
$\mathbb{P}_N(x_0, \mathbf{u})$
$\mathbb{X}_f := \{x \in \mathbb{R}^n \mid Hx \leq h\}$
X_{ICs} set of ICs to consider
$\mathcal{X}'_N := \{\}$
Iteration:
For all $x_0 \in X_{\text{ICs}}$
if MPC problem is feasible:
s.t. $x(N) \in \mathbb{X}_f$
$u \in \mathbb{U}$, $x \in \mathbb{X}$
then
$\mathcal{X}'_N = x_0 \cup \mathcal{X}'_N$
end
end