Model Predictive Control for a Tilt-Rotor Tricopter UAV considering uncertain payload

India van Doornen, Niels Stienen

***Abstract*—This paper investigates the application of model predictive control (MPC) for a Tilt-Rotor Tricopter with an unknown payload. The cost function features a rate-of-change penalty to mitigate abrupt changes in control input. A stability analysis is performed to prove asymptotic stability of the closed-loop system. A comparison is made between the MPC controller and a constrained Linear Quadratic Regulator (LQR) controller. Furthermore, a 3D reference trajectory is simulated to assess the performance of the MPC controller in trajectory-tracking scenarios.**

I. Introduction

This project builds upon the control framework of a tilt-rotor tricopter as established by Prach and Kayacan [1]. The primary focus is on developing a model predictive control (MPC) design that incorporates state, input, and input rate constraints, in contrast to the linear quadratic regulator (LQR) controller approach. Additionally, the paper investigates trajectory tracking and the drop of an uncertain payload with adapted control strategies. Compared to the work by Prach and Kayacan, the PID position controller and MPC are merged unto a unified MPC framework. Furthermore, the control allocation algorithm is simplified such that the MPC directly outputs the relative control inputs.

A. Tricopter Dynamics

The nonlinear dynamical equations describing the motion of the tricopter are given below as derived in [1]. The attitude of the tricopter is described by the Euler angles (ϕ, θ, ψ) where the corresponding Euler rates are given through a transformation from the angular body rates (p, q, r). Moreover, the position in the inertial frame $\mathcal{N}$ is described by a transformation from the body-fixed frame $\mathcal{F}$.

$$^{\mathcal{N}}\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = {}^{\mathcal{N}}\mathbf{C}^{\mathcal{F}}(\phi, \theta, \psi) \; {}^{\mathcal{F}}\begin{bmatrix} X_b \\ Y_b \\ Z_b \end{bmatrix} \tag{1}$$

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & \sin(\phi)\tan(\theta) & \cos(\phi)\tan(\theta) \\ 0 & \cos(\phi) & -\sin(\phi) \\ 0 & \frac{\sin(\phi)}{\cos(\theta)} & \frac{\cos(\phi)}{\cos(\theta)} \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \tag{2}$$

India van Doornen and Niels Stienen, are master students at TU Delft, The Netherlands. E-mail addresses: {k.i.vandoornen, n.l.stienen}@student.tudelft.nl.

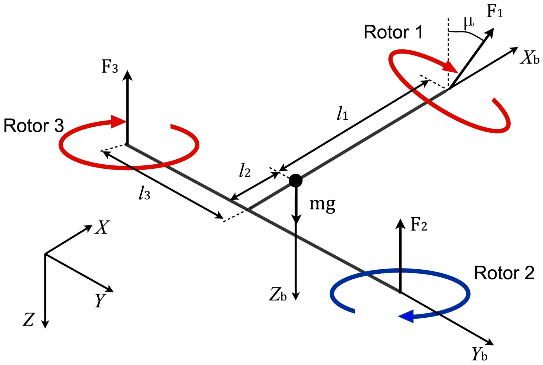

Fig. 1. Schematic overview of a tricopter, illustrating the rotor configuration and orientation. The parameter μ represents the rotation of the rotor around the X_b axis.

$$\begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \\ \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} rv - qw - g\sin(\theta) + \frac{F_x}{m} \\ -ru + pw + g\cos(\theta)\sin(\phi) + \frac{F_y}{m} \\ qu - pv + g\cos(\theta)\cos(\phi) + \frac{F_z}{m} \\ \frac{I_{yy}-I_{zz}}{I_{xx}}qr + \frac{M_x}{I_{xx}} \\ \frac{I_{zz}-I_{xx}}{I_{yy}}pr + \frac{M_y}{I_{yy}} \\ \frac{I_{yy}-I_{xx}}{I_{zz}}pq + \frac{M_z}{I_{zz}} \end{bmatrix} \tag{3}$$

The forces and moments acting on the tricopter, dependent on the control inputs (Ω_1, Ω_2, Ω_3, μ), are described by

$$\begin{bmatrix} F_x \\ F_y \\ F_z \\ M_x \\ M_y \\ M_z \end{bmatrix} = \begin{bmatrix} 0 \\ K_F\Omega_1^2\sin(\mu) \\ -K_F\left(\Omega_1^2\cos(\mu) + \Omega_2^2 + \Omega_3^2\right) \\ -l_3K_F\left(\Omega_2^2 - \Omega_3^2\right) \\ -l_2K_F\left(\Omega_2^2 + \Omega_3^2\right) + l_1K_F\Omega_1^2\cos(\mu) \\ l_1F_y - K_M\left(\Omega_1^2\cos(\mu) - \Omega_2^2 + \Omega_3^2\right) \end{bmatrix} \tag{4}$$

B. Linearized Dynamics

For the design of the MPC controller, the system dynamics are linearized to determine the control action that will subsequently be applied to the continuous nonlinear dynamics. This section outlines the process of linearizing the dynamics.

A trim analysis establishes the hover condition around which the system is linearized, with detailed derivation provided in [1]. The resulting equations for the hover condition are characterized by $(u, v, w, p, q, r)_{\text{trim}} = 0$ and

$$\begin{bmatrix}\phi\\ \theta\\ \mu\\ \Omega_1\\ \Omega_2\\ \Omega_3\end{bmatrix}_{\text{trim}} = \begin{bmatrix}\tan^{-1}\left(\frac{-l_2K_M}{l_1(l_1+l_2)K_F}\right)\\ 0\\ \tan^{-1}\left(\frac{K_M}{l_1K_F}\right)\\ \sqrt{\frac{l_2gm}{(l_1+l_2)K_F}\frac{\cos(\phi_{\text{trim}})}{\cos(\mu_{\text{trim}})}}\\ \sqrt{\frac{l_1gm}{2(l_1+l_2)K_F}\cos(\phi_{\text{trim}})}\\ \Omega_{2_{\text{trim}}}\end{bmatrix} \tag{5}$$

The tricopter's location (X, Y, Z) and yaw angle (ψ) can be chosen freely for linearization around the hover condition, resulting in a LTI system represented by the state-space equations:

$$\dot{x} = Ax + Bu, \tag{6}$$

$$y = Cx \tag{7}$$

where C is the identity matrix for state feedback control. The 12 states and 4 control inputs for the system are defined as

$$x = [u\ v\ w\ \phi\ \theta\ \psi\ p\ q\ r\ X_b\ Y_b\ Z_b]^T \tag{8}$$

$$u = [\Omega_1\ \Omega_2\ \Omega_3\ \mu]^T \tag{9}$$

- u, v, w, the translational body rates expressed in $\mathcal{F}$
- p, q, r, the angular body rates expressed in $\mathcal{F}$
- ϕ, θ, ψ, the Euler angles expressed in $\mathcal{N}$
- X_b, Y_b, Z_b, the tricopter position expressed in $\mathcal{F}$
- $\Omega_1, \Omega_2, \Omega_3$, the rotor speeds
- μ, the angle of rotor 1 (as seen in Figure 1)

II. Model predictive control design

For the design of the MPC controller, the linearized dynamics are used. As MPC control uses discrete time steps, the dynamics are discretized using zero-order hold with a sampling time $T_s = 0.1$ seconds. The matrices A and B mentioned hereafter are their discretized counterparts. The discrete dynamics are described by

$$\begin{aligned} x_{k+1} &= Ax_k + Bu_k\\ y_k &= Cx_k \end{aligned} \tag{10}$$

An MPC is designed to stabilize the tricopter in the origin of the body-fixed frame $\mathcal{F}$ using full-state feedback. Here, the hover condition is the origin of the linearized system. The horizon time is set to be equal to 2.5 seconds (around settling time of closed-loop system), which results in prediction horizon $N = 25$.

A. Cost function

The general cost function used in the minimization algorithm is defined as

$$V_N(x_0, \mathbf{u}_N) = \sum_{k=0}^{N-1} l(x_k, u_k) + V_f(x_N) \tag{11}$$

To prevent abrupt changes in rotor rpm, a penalty on the rate of change of the control input was imposed. The resulting stage and terminal cost for a regulation problem are then given by

$$\begin{aligned} l(x_k, u_k) &= \frac{1}{2}\left(x_k^TQx_k + u_k^TRu_k + \Delta u_k^TL\Delta u_k\right)\\ V_f(x_N) &= \frac{1}{2}x_N^TPx_N \end{aligned} \tag{12}$$

After the introduction of the augmented state $\tilde{x}_k = \begin{bmatrix}x_k & u_{k-1}\end{bmatrix}^T$, the terminal and stage costs can be reformulated to resemble those of a standard LQR with cross terms:

$$\begin{aligned} l(\tilde{x}_k, u_k) &= \frac{1}{2}\left(\tilde{x}_k^T\tilde{Q}\tilde{x}_k + u_k^T\tilde{R}u_k + 2\tilde{x}_k^T\tilde{M}u_k\right)\\ V_f(\tilde{x}_N) &= \frac{1}{2}\tilde{x}_N^T\tilde{P}\tilde{x}_N. \end{aligned} \tag{13}$$

with

$$\tilde{A} = \begin{bmatrix}A & 0\\ 0 & 0\end{bmatrix},\ \tilde{B} = \begin{bmatrix}B\\ I\end{bmatrix},\ \tilde{Q} = \begin{bmatrix}Q & 0\\ 0 & L\end{bmatrix},$$

$$\tilde{R} = R + L,\ \tilde{M} = \begin{bmatrix}0\\ -L\end{bmatrix} \text{ and } \tilde{P} = \begin{bmatrix}P & 0\\ 0 & 0\end{bmatrix}$$

where $\tilde{P}$ is the unique (positive semi-definite) solution to the Discrete Algebraic Riccati Equation.

$l(\tilde{x}_k, u_k)$ can be expressed as a quadratic function

$$\frac{1}{2}\begin{bmatrix}\tilde{x}_k\\ u_k\end{bmatrix}^T N \begin{bmatrix}\tilde{x}_k\\ u_k\end{bmatrix},\quad N = \left[\begin{array}{c|c}\tilde{Q} & \tilde{M}\\ \hline \tilde{M}^T & \tilde{R}\end{array}\right] \tag{14}$$

It is assumed matrix L is symmetric, from which follows that

$$N = \left[\begin{array}{cc|c}Q & 0 & 0\\ 0 & L & -L\\ \hline 0 & -L^T & R+L\end{array}\right] \sim \left[\begin{array}{cc|c}Q & 0 & 0\\ 0 & L & 0\\ \hline 0 & -L^T & R\end{array}\right] \tag{15}$$

Given that $l(\tilde{x}_k, u_k)$ must be positive semi-definite and $\tilde{R}$ must be positive definite, it follows from the block-triangular structure of N that R must be positive definite ($R \succ 0$) and both Q and L must be positive semi-definite ($Q, L \succeq 0$), validating the assumption on matrix L.

To enable trajectory tracking, the cost function of the MPC is adapted to penalize deviations from a desired reference trajectory x_{ref}. The revised cost function is formulated as

$$\begin{aligned} l(\tilde{x}_k, u_k) &= \frac{1}{2}(\tilde{x}_k - x_{\text{ref},k})^T\tilde{Q}(\tilde{x}_k - x_{\text{ref},k})\\ &\quad + u_k^T\tilde{R}u_k + 2(\tilde{x}_k - x_{\text{ref},k})^T\tilde{M}u_k\\ V_f(\tilde{x}_N) &= \frac{1}{2}(\tilde{x}_N - x_{\text{ref},N})^T\tilde{P}(\tilde{x}_N - x_{\text{ref},N}). \end{aligned} \tag{16}$$

B. Constraints

It is assumed that the electric motors of the propellers have a range of $0 - 3000$ rpm. The rotor tilt angle is constrained between $\frac{-\pi}{2}$ and $\frac{\pi}{2}$. To prevent singularities in the transformation matrix and prevent unrealistic pitch and roll behaviour, their angles are constrained between $\frac{-\pi}{2}$ and $\frac{\pi}{2}$. The yaw is bounded between -2π and 2π, as the yaw is periodic with 2π. The constraints need to be applied to the system linearized around the trim conditions, leading to the trim-dependent input constraints

$$\begin{aligned}-\Omega_{i,\mathrm{trim}} \leq \Omega_i \leq 3000 - \Omega_{i,\mathrm{trim}} \\ \frac{-\pi}{2} - \mu_{\mathrm{trim}} \leq \mu \leq \frac{\pi}{2} - \mu_{\mathrm{trim}}\end{aligned} \tag{17}$$

and state constraints

$$\begin{aligned}\frac{-\pi}{2} \leq \theta, \phi \leq \frac{\pi}{2} \\ -2\pi \leq \psi \leq 2\pi\end{aligned} \tag{18}$$

where the effect of the trim condition on the roll is neglected.

These constraints can compactly be represented as $Fx_k \leq f$ and $Gu_k \leq g$. For prediction horizon N, this representation can easily be transformed for the whole state sequence $\boldsymbol{x_{N,k}}$ and input sequence $\boldsymbol{u_{N,k}}$, which is especially useful for implementation of the terminal constraint set $\mathbb{X}_{\mathrm{f}} = \{x \in \mathbb{R}^n \mid Hx \leq h\}$. The state and input sequence are constraint with $\tilde{F}\boldsymbol{x_{N,k}} \leq \tilde{f}$ and $\tilde{G}\boldsymbol{u_{N,k}} \leq \tilde{g}$, where

$$\tilde{F} = \mathrm{diag}(F_1, \cdots, F_N, H), \quad \tilde{f} = [f_1 \; \cdots \; f_N \; h]^T \tag{19a}$$

$$\tilde{G} = \mathrm{diag}(G_1, \cdots, G_N), \quad \tilde{g} = [g_1 \; \cdots \; g_N]^T. \tag{19b}$$

with $F_i = F$, $f_i = f$, $G_i = G$, and $g_i = g$.

Since the optimization parameter is $\boldsymbol{u_{N,k}}$, the state constraints (19a) need to be mapped to the input space through[2]

$$\begin{aligned}\tilde{F}(\mathcal{A}x_k + \mathcal{B}\boldsymbol{u_{N,k}}) \leq \tilde{f} \\ \tilde{F}\mathcal{B}\boldsymbol{u_{N,k}} \leq \tilde{f} - \tilde{F}\mathcal{A}x_k\end{aligned} \tag{20}$$

where x_k is the initial state a time k, and

$$\mathcal{A} = \begin{bmatrix} I \\ A \\ A^2 \\ \vdots \\ A^N \end{bmatrix}, \quad \mathcal{B} = \begin{bmatrix} 0 & 0 & \cdots & 0 \\ B & 0 & \cdots & 0 \\ AB & B & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ A^{N-1}B & A^{N-2}B & \cdots & B \end{bmatrix} \tag{21}$$

The admissible control sequence at time k is then given by

$$\mathcal{U}_N(x_k) = \left\{ \boldsymbol{u_{N,k}} \;\middle|\; \begin{bmatrix} \tilde{F}\mathcal{B} \\ \tilde{G} \end{bmatrix} u_{N,k} \leq \begin{bmatrix} \tilde{f} \\ \tilde{g} \end{bmatrix} - \begin{bmatrix} \tilde{F}\mathcal{A} \\ 0 \end{bmatrix} x_k \right\} \tag{22}$$

In the context of trajectory tracking, the right-hand side of equation (22) is modified to

$$\begin{bmatrix} \tilde{f} \\ \tilde{g} \end{bmatrix} - \begin{bmatrix} \tilde{F} \\ 0 \end{bmatrix} (\mathcal{A}x_k - \boldsymbol{x_{\mathrm{ref}}}) \tag{23}$$

where $\boldsymbol{x_{\mathrm{ref}}}$ represents the state references defined for the entire prediction horizon.

C. Terminal set $\mathbb{X}_{\mathrm{f}}$

Algorithm 1 is used to determine the maximal constraint admissible set $\mathbb{X}_{\mathrm{f}}$ for the regulation problem. This algorithm is an adapted version of that presented by Gilbert and Tan[3]. When $\mathbb{Z} = \mathbb{X} \times \mathbb{U}$ is a polytope, the maximization in 1 is efficiently solved through linear programming.[3] The result is another polytope $\mathbb{X}_{\mathrm{f}} = \{x \in \mathbb{R}^n \mid Hx \leq h\}$ containing 576 constraints, allowing for quadratic programming of the MPC problem.

For the reference tracking problem, at each time instance the terminal set of the regulation problem is translated by the vector x_{ref}

$$\mathbb{X}_{\mathrm{f}}(y_{\mathrm{ref}}) = \{x_{\mathrm{ref}}\} \oplus \mathbb{X}_{\mathrm{f}} \tag{24}$$

D. Payload observer

The UAV is equipped with a payload where the payload mass is randomly distributed in the range 0.1-0.3kg. This uncertainty in mass can be modelled as a constant, unknown, disturbance on the true mass. Consequently, a Luenberger observer can be implemented to estimate the mass of the tricopter.

$$\begin{aligned}\begin{bmatrix} x_{k+1} \\ m_{k+1} \end{bmatrix} = \begin{bmatrix} A & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} x_k \\ m_k \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} u_k \\ y_k = Cx_k \\ \hat{m}_{k+1} = \hat{m}_k + L_{\mathrm{obs}}(m - \hat{m}_k)\end{aligned} \tag{25}$$

The linearized A matrix does not depend on the mass of the tricopter, while in the B matrix only the body-fixed velocity states u, v, w depend on m. To decrease computational load, it was decided not to recalculate A and B at every time instance as the effect on the total dynamics was deemed to be small. Rather, as the effect of a disturbance on the mass has a predominant effect on the hover condition, the trim values are recalculated every time step to account for the evolution of $\hat{m}$.

E. Optimal control problem

The summary of the optimal control problem $\mathbb{P}_N(x_0, t)$ is described by (26). A quadratic cost function with a rate of change penalty is implemented to penalize aggressive control changes. The terminal cost is chosen as a Lyapunov function to ensure (Lyapunov) stability of the closed-loop system. An invariant terminal set is enforced to guarantee recursive feasibility of the MPC problem. It is defined as a polytopic set, allowing for quadratic programming of the MPC problem. The matrices L, Q, and R used in (11) are defined by $0.5 \cdot \mathrm{diag}(1, 1, 1, 100)$ $100 \cdot \mathrm{diag}(1, 1, 1, 0.5, 0.5, 10, 10, 10, 5, 500, 500, 300)$, and $0.1 \cdot \mathrm{diag}(1, 1, 1, 1)$ respectively. The prediction horizon is set to $N = 25$. The tuning process and simulation results can be found in Section IV.

$$\mathbb{P}_N(x_0, t) : \begin{cases} \min\limits_{\boldsymbol{u_N}} V_N(x_0, \boldsymbol{u_N}) \\ \text{s.t. } Ax + Bu \\ Fx_k \leq f \\ Gu_k \leq g \\ x_N \in \mathbb{X}_{\mathrm{f}} \end{cases} \tag{26}$$

III. Asymptotic stability

In this section, the asymptotic stability of the linearized closed-loop system for the previously designed MPC is proven. In Section IV it will be shown through numerical simulations that the proposed MPC is able to stabilize the nonlinear system.

A. Stability analysis of the linearized system

As the system is constrained in both states and inputs, it is (nearly) impossible to find a global CLF. Instead, a local CLF together with a control invariant set $\mathbb{X}_{\mathrm{f}}$ is determined to prove local asymptotic stability. The linearized system dynamics $x_{k+1} = Ax_k + Bu_k$ are subjected to control and state constraints $u \in \mathbb{U}, x \in \mathbb{X}$, where $\mathbb{U}$ is a compact and $\mathbb{X}$ is a closed set, both containing the origin in its interior. As the controllability matrix $K = \begin{bmatrix} B \; AB \; \cdots \; A^{12}B \end{bmatrix}$ of the linearized system has full row rank, the system is controllable.

To prove the origin is exponentially stable in $\mathcal{X}_N$ through Theorems 2.19 and 2.21 in the book [4], it remains to be shown Assumptions 2.2, 2.3, and 2.14 are satisfied and $\mathbb{X}_\mathrm{f}$ has a non-empty interior containing the origin.

Assumption 2.2 *(Continuity of system and cost)*:
The functions $f : \mathbb{Z} \to \mathbb{X}$, $l : \mathbb{Z} \to \mathbb{R}_{\geq 0}$ and $V_f : \mathbb{X}_\mathrm{f} \to \mathbb{R}_{\geq 0}$ are continuous, $f(0,0) = 0$, $l(0,0) = 0$ and $V_f(0) = 0$.

In Section II it was shown that $l(\tilde{x}, u)$ and $V_f(\tilde{x})$ are positive semi-definite quadratic functions. Therefore both functions are continuous mapping to the non-negative real numbers, and $l(0,0) = 0$ and $V_f(0) = 0$. Furthermore, as the state dynamics are linearized, f is also continuous with $f(0,0) = 0$. Therefore assumption 2.2 is met.

Assumption 2.3 *(Properties of constraint sets)*:
The set $\mathbb{Z}$ is closed and the set $\mathbb{X}_\mathrm{f} \subseteq \mathbb{X}$ is compact. Each set contains the origin. As $\mathbb{U}$ is bounded, the set $\mathbb{U}(x)$ is compact $\forall x \in \mathbb{X}$.

In Section II it was described how all inputs and the states ϕ, θ, and ψ are constrained by non-strict inequalities, while all other states are left unconstrained. As there are no mixed constraints, $\mathbb{Z} = \mathbb{X} \times \mathbb{U}$. The control constraint is not state-dependent and is given by $u \in U$. $\mathbb{U}$ is closed and bounded, and therefore by Heine-Borrel's theorem it is compact. The set $\mathbb{X}$ is closed, implying that the cartesian product $\mathbb{Z} = \mathbb{X} \times \mathbb{U}$ is also a closed set. Furthermore, as it is linearized around the trim condition, $\mathbb{Z}$ contains the origin.

From [3] it can be concluded that for $0 \in \mathbb{Z}$ the set $\mathbb{X}_\mathrm{f}$ is non-empty with $0 \in \mathbb{X}_\mathrm{f}$. Furthermore, it is stated that the property of closure of $\mathbb{Z}$ is inherited by $\mathbb{X}_\mathrm{f}$. The closure of an arbitrary subset S in space W is the intersection of all closed sets in W containing S. If the set S is closed, then S is the intersection of all sets containing S, meaning that $S = \mathrm{cl}(S)$. As set $\mathbb{Z}$ is closed and the closure property of $\mathbb{X}_\mathrm{f}$ is inherited from $\mathbb{Z}$, $\mathbb{X}_\mathrm{f} = \mathrm{cl}(\mathbb{X}_\mathrm{f})$, and thus it can be concluded the set $\mathbb{X}_\mathrm{f}$ is closed. Furthermore, as the set $\mathbb{Z}$ is bounded and therefore by [3] the set $\mathbb{X}_\mathrm{f}$ is also bounded, it can be concluded by the Heine-Borrel theorem that $\mathbb{X}_\mathrm{f}$ is compact.

Assumption 2.14 *(Basic stability assumption)*:
$V_f(\cdot)$, $\mathbb{X}_\mathrm{f}$ and $l(\cdot)$ have the following properties:

1) $\forall x \in \mathbb{X}_\mathrm{f}, \exists u \in \mathbb{U}$:
$$\begin{aligned} &f(x,u) \in \mathbb{X}_\mathrm{f} \\ &V_f(f(x,u)) - V_f(x) \leq -l(x,u) \end{aligned}$$
2) $\exists \alpha_1(\cdot), \alpha_f(\cdot) \in \mathcal{K}_\infty$:
$$\begin{aligned} &l(x,u) \geq \alpha_1(|x|), \quad \forall x \in \mathcal{X}_N, \forall u \in \mathbb{U} \\ &V_f(x) \leq \alpha_f(|x|), \quad \forall x \in \mathbb{X}_\mathrm{f} \end{aligned}$$

The adapted algorithm as presented in [3] generates a (close approximation) of the maximum invariant constraint admissible set $\mathbb{X}_\mathrm{f}$ for the closed-loop system $x^+ = A_K x$. This is the largest set W for which

$$\begin{aligned} W &\subseteq \{x \in \mathbb{X}_\mathrm{f} \mid Kx \in \mathbb{U}\} \\ x \in W &\Rightarrow x(i) = A_K^i x \in W, \quad \forall i \geq 0 \end{aligned} \tag{27}$$

It remains to be proven that $\mathbb{X}_\mathrm{f}$ is then control invariant for $x^+ = Ax + Bu$, $u \in \mathbb{U}$;

$$\forall x_0 \in \mathbb{X}_\mathrm{f}, \; x(i) = A_K^i x \in \mathbb{X}_\mathrm{f}, \; u(i) = Kx(i), \quad \forall i \geq 0.$$

For such a system, the controller $u = Kx$ will therefore keep all trajectories starting in $\mathbb{X}_\mathrm{f}$ within it, satisfying the state and input constraints for all future times.

For any finite prediction horizon N, the value function $V_N^{uc}(x)$ for the unconstrained optimal control problem $\mathbb{P}_N^{uc}(x) \; : \; V_N^{uc}(x) = \min\limits_{\mathbf{u}} V_N(x, \mathbf{u})$ satisfies (11), with $V_f(x) := V_\infty^{uc}(x)$ the value function of the infinite horizon unconstrained optimal control problem. It follows that

$$V_N^{uc} = V_\infty^{uc} = V_f(x) = \frac{1}{2} x^T P x \tag{28}$$

As the finite horizon constrained optimal problem $\mathbb{P}_N(x)$ is the same as the unconstrained problem $\mathbb{P}_N^{uc}(x)$ except for the inclusion of constraints, the constrained optimal cost $V_N^0(x)$ will be higher than the unconstrained optimal cost

$$V_N^{uc}(x) = V_f(x) \leq V_N^0(x), \quad \forall x \in \mathcal{X}_N \supseteq \mathbb{X}_\mathrm{f} \tag{29}$$

By Lemma 2.18 (Monotonicity of the value function) from [4]

$$V_N^0(x) \leq V_f(x), \quad \forall x \in \mathbb{X}_\mathrm{f} \tag{30}$$

it can be concluded that

$$V_N^{uc}(x) = V_N^0(x) = V_f(x) = V_\infty^{uc}, \quad \forall x \in \mathbb{X}_\mathrm{f} \tag{31}$$

By the uniqueness of the solutions to problems $\mathbb{P}_N(x)$ and $\mathbb{P}_N^{uc}(x)$

$$\kappa_N(x) = \kappa_\infty^{uc}(x) = Kx, \quad \forall x \in \mathbb{X}_\mathrm{f} \tag{32}$$

It can therefore be concluded that if $V_f(x) := V_\infty^{uc}$ with corresponding controller $u = Kx$, and if $\mathbb{X}_\mathrm{f}$ is invariant for $x^+ = A_K x$, then the constrained and controlled system $x^+ = Ax + B\kappa_N(x)$ is also positive invariant to $\mathbb{X}_\mathrm{f}$.

To verify the Lyapunov decrease condition is met, for all vertices of $\mathbb{X}_\mathrm{f}$ it should be checked if there exists an input u such that $V_f(f(x,u)) - V_f(x) \leq -l(x,u)$ holds. Unfortunately, determining the vertices of a polytope is not as easy as it seems, and for a polytope with 576 constraints the problem is NP-complete, e.i. a solution can be verified "quickly" but there is no known way to find a solution quickly. It is assumed *a prior* this condition holds, and in Section IV it will be shown *a posteriori* the Lyapunov decrease condition is met numerically. The first property of Assumption 2.14 is thereby met.

The stage cost $l(\tilde{x}, u)$ can be written in quadratic form (14), with $N \succeq 0$. The lower bound can be given by

$$l(z) = \frac{1}{2} z^T N z \geq \frac{1}{2} \lambda_{\min}(N) |z|^2 \geq \frac{1}{2} \lambda_{\min}(N) |x|^2 \tag{33}$$

where $z = [\tilde{x}\ u]^T$. Repeating the same procedure for the upper bound on the terminal cost $V_f(x)$ with $P \succeq 0$ results in

$$V_f(x) = \frac{1}{2}x^T P x \leq \frac{1}{2}\lambda_{\max}(P)|x|^2 \qquad (34)$$

Therefore the state cost is lower bounded by $\alpha_1(|x|) = \lambda_{\min}(N)|x|^2$ and terminal cost upper bounded by $\alpha_f(|x|) = \frac{1}{2}\lambda_{\max}(P)|x|^2$, where $\alpha_1, \alpha_f \in \mathcal{K}_\infty$. Hence the second property of assumption 2.14 is also met.

Assumptions 2.2, 2.3, and 2.14 are therefore all satisfied, and $\mathbb{X}_\mathrm{f}$ has a non-empty interior containing the origin. Hence by Theorems 2.19 and 2.21, the origin is exponentially stable in $\mathcal{X}_N$.[4]

IV. Numerical simulations

In this section, several numerical simulations show the effectiveness of the MPC controller proposed in Section II for stabilizing the nonlinear system. As the MPC controller uses discrete time steps and the nonlinear dynamics are continuous the simulation uses an ODE solver to solve the continuous nonlinear dynamics for one time step T_s in every iteration. Additionally the MPC controller outputs a control input relative to the trim condition. The block diagram in Figure 2 shows how the conversion between the MPC and nonlinear dynamics occurs. All the code used to run the simulations is available in [5].

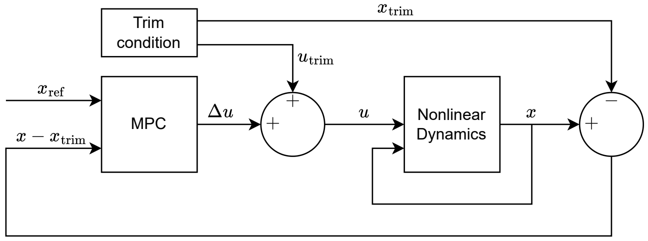

Fig. 2. Block diagram illustrating the implementation of the trim condition within the MPC framework

A. Horizon and sampling time

The relation between the prediction horizon and sampling time of the discrete dynamics is examined. The optimal sampling time and corresponding prediction horizon will be chosen based on this analysis.

For the stabilizing full-state feedback control, the settling time averages around 3 seconds. The prediction horizon is defined as the number of timesteps T_s required to simulate this 3-second interval.

Selecting a proper sampling time is making a compromise between the resolution of the result and the computational time required. Figure 3 illustrates the impact of different sampling times on the controller's performance. Based on this, the sampling time was set to $T_s = 0.1$s.

Setting $T_s = 0.1s$, several prediction horizons were evaluated in Figure 4, with the optimal prediction horizon determined to be $N = 25$. This selected horizon, although slightly shorter than the average settling time, exhibited no significant changes in the state and input evolution. Choosing a slightly lower horizon was found to reduce computational time while maintaining satisfactory performance.

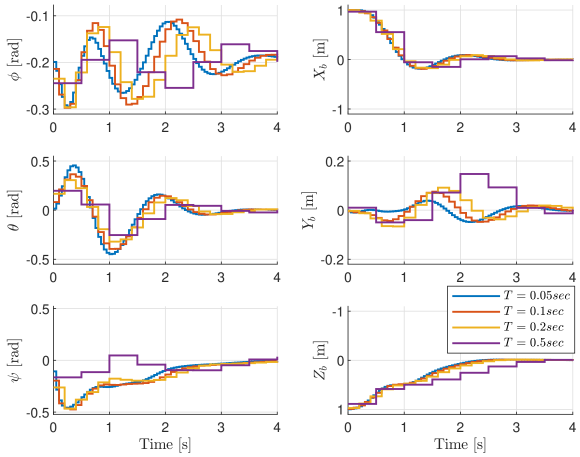

Fig. 3. Comparison of different sampling times.

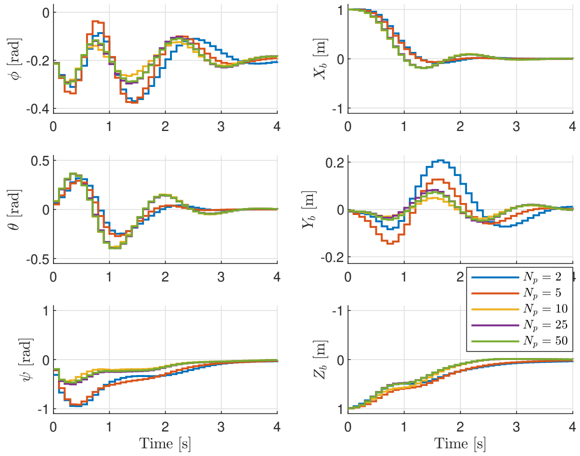

Fig. 4. Effect of different prediction horizons on controller performance

B. Weighing of cost function matrices

Next, the relationship between the rate of change weighting matrix L and its impact on control action and state evolution was investigated. The Q and R matrices were tuned to achieve uniform convergence rates across all states. The results depicted in Figure 5 were analyzed to determine an appropriate scaling factor for L. A scaling factor of 0.5 was selected as it effectively reduced significant changes in control action while maintaining fast convergence.

To evaluate the robustness of the MPC controller under various initial conditions, multiple sets of random initial conditions were generated and simulated, as shown in Figure 6.

C. Lyapunov decrease condition

It was previously established that verifying the Lyapunov decrease condition through the vertices of the polytopic

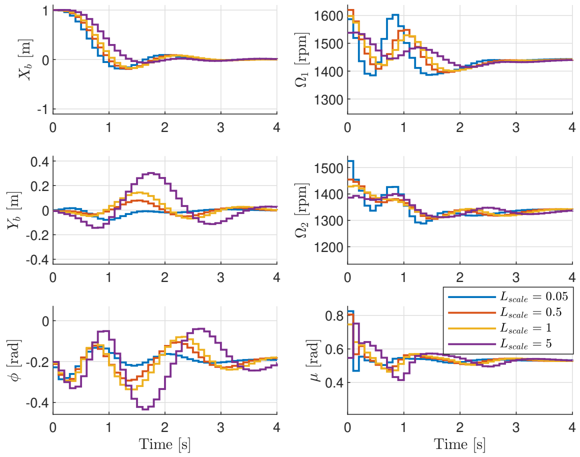

Fig. 5. Simulation of different weightings of L

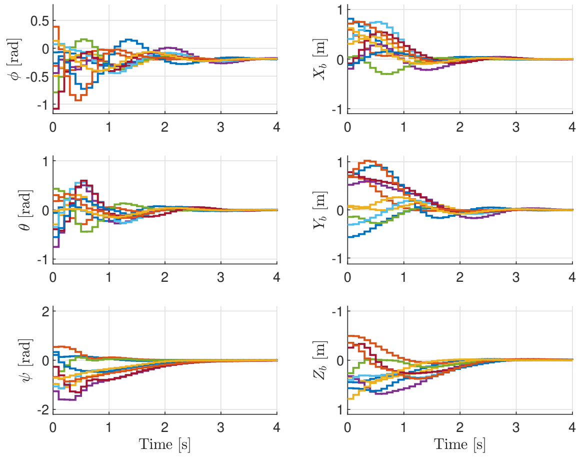

Fig. 6. Simulation of 10 sets of random initial conditions

terminal set is an NP-complete problem. Instead, a numerical approach is adopted to assess the condition by plotting and comparing the left- and right-hand sides of $V_f(f(x,u)) - V_f(x) \leq -l(x,u)$, as depicted in 7. In this figure, the dashed vertical line indicates the timestep at which the state enters the set $\mathbb{X}_\mathrm{f}$. Through numerical analysis, it is concluded *a posteriori* that the Lyapunov decrease condition is satisfied.

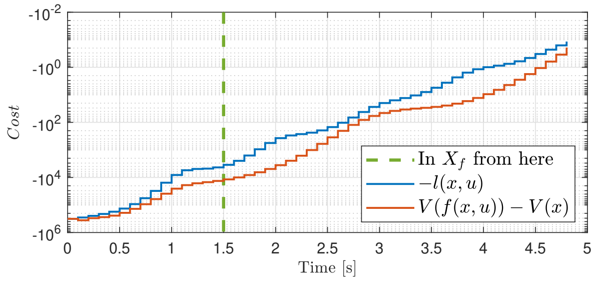

Fig. 7. Numerical verification of Lyapunov decrease condition, shown in log scale

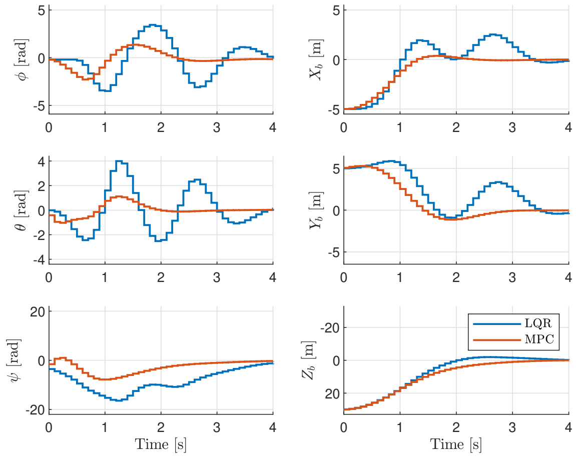

Fig. 8. Comparison of MPC and LQR for linearized system dynamics

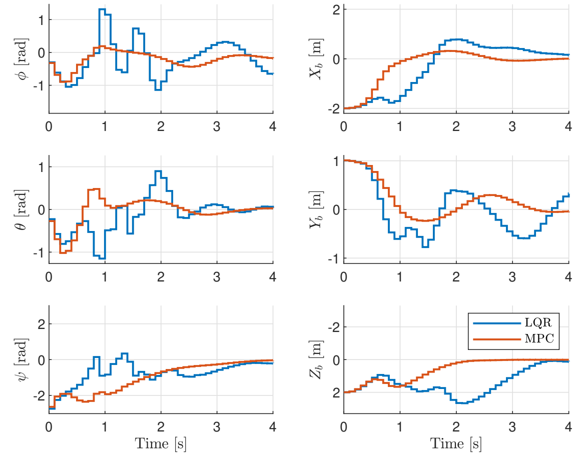

Fig. 9. Comparison of MPC and LQR for nonlinear system dynamics

D. Comparison of LQR and MPC control

Based on the state and input constraints specified in Section II, the regulation MPC and constrained LQR are compared. Both controllers were initialized from the same initial condition and utilized identical gain matrices. The initial condition was deliberately chosen outside the terminal set $\mathbb{X}_\mathrm{f}$. Figure 8 illustrates both controllers applied to the linearized dynamics of the tricopter. The results indicate that the MPC controller exhibits superior performance, characterized by reduced oscillations and faster settling times. A similar comparison was conducted using the nonlinear dynamics, as illustrated in Figure 9. Once again, the MPC is the better-performing controller, highlighting the effectiveness of the MPC framework.

E. Trajectory tracking

To demonstrate the effectiveness of the MPC tracking various trajectories, a specific trajectory was designed featuring straight segments, sharp corners and curved sections. The entire trajectory spans 21 seconds and covers approximately 18 meters. After stopping briefly to stabilize the UAV, the

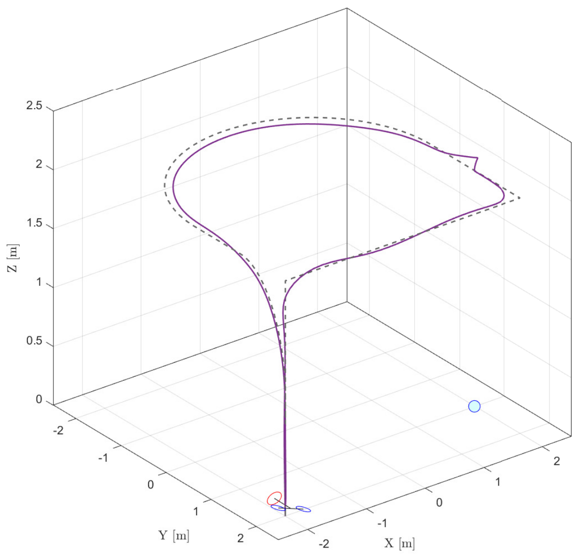

Fig. 10. 3D plot of trajectory tracking results: desired (dotted) vs. actual (solid) trajectory

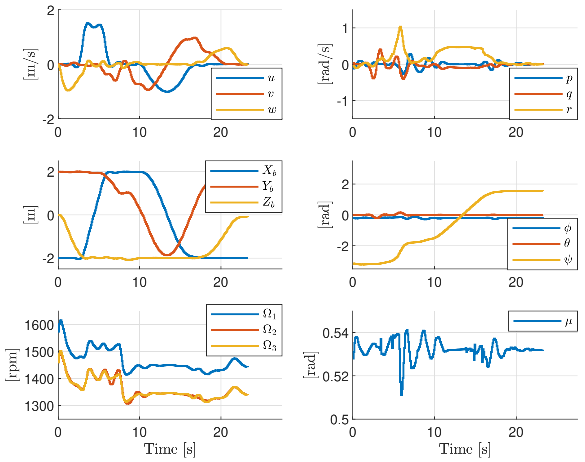

Fig. 11. 2D plot depicting states and inputs corresponding to the trajectory shown in Figure 10

payload is released just after the final 90-degree turn on the third line segment. Figure 10 illustrated the 3D simulation of the trajectory tracking. The dashed grey line represents the desired trajectory, the solid purple line represents the covered trajectory, and the blue ball symbolizes the payload. Notably, the observed sawtooth pattern in the covered trajectory is attributed to the necessity of the observer to estimate the dropped payload mass. The simulation outcome confirms the MPC's capability to accurately track diverse trajectories while accounting for the unknown payload mass.

Figure 11 presents a 2D plot illustrating the states and inputs associated with the trajectory displayed in Figure 10. The bottom left plot specifically highlights a noticeable decrease in rotor rpm following the payload drop. Additionally, incorporating a rate of change penalty in the cost function has proven effective in mitigating substantial changes in control actions. Notably, the plots for roll θ and pitch ϕ exhibit minimal variation throughout the trajectory, suggesting potential for increased speed.

V. Discussion

The Model Predictive Controller developed in this study demonstrates effective stabilization of the tricopter's nonlinear dynamics and enables good trajectory tracking while minimizing abrupt changes in control actions. When initialized from states outside the terminal set $\mathbb{X}_f$, the MPC consistently outperforms the constrained LQR controller.

Nevertheless, there are areas that warrant further investigation in future research. The chosen prediction horizon improves trajectory tracking performance but introduces a computational load that limits real-time applicability. By reducing the prediction horizon in the current setup, e.g. $N = 10$, real-time implementation becomes feasible. A brief analysis of this reduction's impact on trajectory tracking reveals a deteriorated ability to follow the reference trajectory. Future studies should prioritize optimizing the controller implementation for computational efficiency and conduct a more comprehensive investigation into how varying prediction horizons affect trajectory tracking across diverse scenarios. This analysis could yield more insights into balancing performance and computational loads, potentially refining the controller's suitability for practical, real-time applications.

Furthermore, the implemented steady-state target optimizer (SSTO) did not appear to improve trajectory performance compared to directly using the desired trajectory as a reference. It was anticipated that integrating SSTO would lead to improved and more efficient trajectory tracking, particularly in navigating long curved corners, by computing references for pitch, yaw, and roll. It is unclear if this is caused by an error in the implementation of the SSTO, or if the absence of disturbances or absence of discrepancies in the model results in similar performance. Future studies should investigate the cause behind deviations from the curved corned, and if this could be solved by a different implementation of an SSTO.

References

[1] A. Prach and E. Kayacan, "An mpc-based position controller for a tilt-rotor tricopter vtol uav," *Optimal Control Applications and Methods*, vol. 39, no. 1, pp. 343–356, 2018. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/oca.2350

[2] M. Fink, *Implementation of Linear Model Predictive Control – Tutorial*. Technical University of Munich, 2021.

[3] E. Gilbert and K. 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, 1991.

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

[5] K. I. van Doornen and N. Stienen, "Model predictive control of a tricopter," 2024. [Online]. Available: https://github.com/StienenNiels/MPC

Appendix

Algorithm 1 Estimate maximal constraint admissible set $\mathbb{X}_{\mathrm{f}}$

```
Initialization:
    Set $k := 0$
    $K :=$ LQR optimal gain
    $A_K := A + BK$
Iteration:
for all $i = 1, 2, ..., s$ do
    $x_i^* := \begin{cases} \arg\max_x & f_i\left(A_K^{k+1} x\right) \\ \text{s.t.} & f_j\left(A_K^t x\right) \leq 0, \quad \forall j \in \{1, 2, ..., s\}, \\ & \qquad\qquad\qquad \forall t \in \{0, 1, ..., k\} \end{cases}$
end for
if $f_i\left(A_K^{k+1} x_i^*\right) \leq 0,\ \forall j \in \{1, 2, ..., s\}$ then
    set $k^* = k$ and define
    $\mathbb{X}_{\mathrm{f}} := \{x \in \mathbb{R}^n \mid f_j\left(A_K^t x\right) \leq 0,$
            $\forall j \in \{1, 2, ..., s\}, \forall t \in \{0, 1, ..., k^*\}\}$
else
    set $k := k + 1$ and continue.
end if
```