Disturbance-Robust MPC for Output Tracking of Underactuated Systems with Ellipsoidal Terminal Set

Max Dam (6055745), Mukil Saravanan (6195474)

***Abstract*—We study a model predictive control (MPC) approach for drones with tethered cargo. The designed controller uses output MPC with disturbance rejection to steer a drone to the desired steady state while under linear state and output constraints. To ensure recursive stability, a terminal constraint that utilizes an ellipsoidal level set approximation to approximate the control invariant set is implemented. Asymptotic stability is mathematically proven and compared to an empirical result. The solution is compared to the performance of an LQR controller, and the tuned cost matrices are compared to identity matrices.**

I. Introduction

In recent years, organizations have started to explore the use of drones for last-mile delivery or transport of items. The volume and form factor of the cargo is generally constrained by the container attached to the drone. To overcome this limitation we envision a niche for cargo drones with loads suspended by a tether instead of rigidly attached to the drone body. Off-the-shelf drone control systems are designed with the assumption that the drone body and cargo act as a single rigid body, performance of these controllers is therefore not optimal for the case of a suspended load. Furthermore, constraint on load deflection are required to make sure the load does not impact the drone body or thrusters. For ease of reading, we will refer to the load as the "pendulum" in the rest of this report.

To address the challenges described above, we have designed a model predictive controller that utilizes the linearized dynamics of the drone-pendulum system. The controller is used steer the system to a stable reference state while respecting state, control, and terminal constraints - and while rejecting constant disturbances caused by model mismatch between the linearized and true dynamics. In addition, we show that the system can achieve this control in real time. We simulate the system in a custom simulation environment that implements the system's full nonlinear dynamics as empirical proof.

A. System dynamics and assumptions

As the focus is on controller design and not on system identification, we have limited the current work to a case of a planar drone with a configuration in $SE(2) \times S$. The system therefore has an 8-dimensional state space, 6 dimensions for the drone, and 2 for the pendulum. We have opted to neglect drag in the equations.

Max Dam and Mukil Saravanan are MSc Robotics students at TU Delft, The Netherlands. E-mail addresses: {m.dam-2, m.saravanan-1}@student.tudelft.nl.

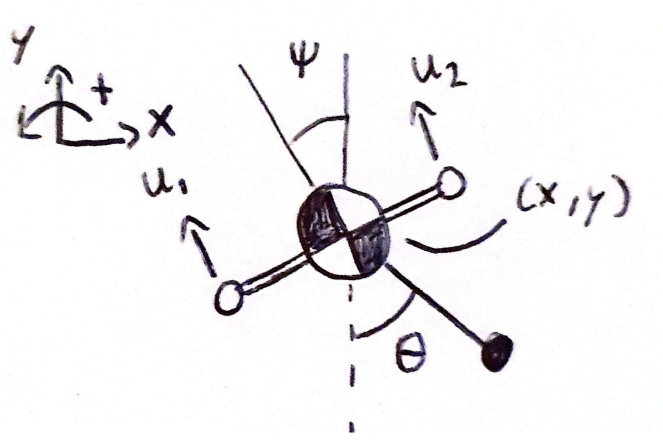

Fig. 1. Configurations of a planar drone

For the planar case, we assume the drone has 2 thrusters without spin-up time as actuators. Each of these can generate a force that can be positive or negative. This gives the drone 2 scalar control inputs, making it an underactuated system.

The full state and input vectors are:

$$\mathbf{x} = \begin{bmatrix} x & y & \psi & \theta & \dot{x} & \dot{y} & \dot{\psi} & \dot{\theta} \end{bmatrix}^\top$$

$$\mathbf{u} = \begin{bmatrix} u_1 & u_2 \end{bmatrix}^\top$$

Where:

- x, y: Drone translation in meters (m)
- ψ: Drone rotation angle in radians (rad)
- θ: Pendulum angle in radians (rad)
- $\dot{x}, \dot{y}, \dot{\psi}, \dot{\theta}$: Velocities of the corresponding states
- u_1, u_2: Thruster forces in Newtons (N)

Of these states, x and y are observed by the controller.

The controller uses a linearized version of the full nonlinear dynamics. The dynamics have been derived by solving the system's Lagrangian using a computer algebra system. The dynamics have been linearized around steady-state hovering using a small angle approximation for ψ and θ. For the full formulation, see the git repository [3]. For the full dynamics and linear system matrices, see VI.

For use with the controller, linear dynamics have been discretized with the use of the matrix exponential [4]. This turns the linear system from the continuous form

$$\dot{x} = A_c x + B_c u$$
$$y = Cx$$

to the discrete form

$$x^+ = A_d x + B_d u$$
$$y = Cx$$

II. Model predictive control design

A. Horizon and frequency

The MPC is designed to steer towards a stable reference state while respecting state, input, and terminal constraints. For practical reasons, all states were constrained to create a "box" in order to keep the required horizon N relatively low. In the end, a horizon $N = 40$ was chosen with a control frequency of $10Hz$. This setup preserved stability and feasibility during all of testing. The combination was obtained by setting the reference output to the extremes of the state constraints of the x and y positions and increasing N until the problems became feasible from any stationary starting position within the space.

B. Cost functions

The controller tries to find a sequence of control inputs that minimizes the following cost function:

$$J(x_0, u) = \sum_{k=0}^{N-1} \ell(x(k), u(k)) + V_f(x(N))$$
$$\text{s.t. } u \in \mathbb{U}, x \in \mathbb{X}, x[N] \in \mathbb{X}_f$$

where:

- $\ell(x(k), u(k))$ is the stage cost function, which quantifies the cost incurred at each time step k as a function of the state $x(k)$ and the control input $u(k)$.
- $V_f(x(N))$ is the terminal cost function, which quantifies the cost associated with the final state $x(N)$ at the end of the prediction horizon N.

The stage and terminal cost functions are:

$$\ell(x(k), u(k)) = \frac{1}{2}(x(k)^T Q x(k) + u(k)^T R u(k))$$
$$V_f(x(N)) = x(N)^T P x(N)$$

where:

- $Q = Diag(1, 1, 20, 20, 1, 1, 5, 5)$ is the state cost matrix.
- $R = 0.1I$ is the input cost matrix
- P is the solution to the discrete algebraic Riccati equation.

Q and R were manually tuned to bias the controller to keeping ψ and θ close to the equilibrium point.

C. State and input constraints

The system makes use of linear state and input constraints. These are formulated in matrix form as:

$$H_x x \leq h_x$$

for the state, and

$$H_u \leq h_u$$

for the input.

The state constraint are:

$$-\frac{\pi}{6} \leq \psi \leq \frac{\pi}{6}$$
$$-\frac{\pi}{6} \leq \psi - \theta \leq \frac{\pi}{6}$$

The first constraint is in place to minimize the discrepancy between the true and linear dynamics caused by the small angle approximation. The second constraint imposes that the angle between the drone body and the pendulum does not exceed some limit, this is to satisfy the idea that the cargo should not swing around relative to the drone body as it might hit itself.

All other state constraints were taken as

$$-10 \leq x_i \leq 10$$

In order to keep the problem constrained and make troubleshooting easier during development.

The input constraints are:

$$-20 \leq u_i \leq 20 \quad i \in \{0, 1\}$$

These values were chosen as they allow the controller to accelerate the drone upwards with an acceleration of one g.

D. Terminal set

To guarantee recursive stability, a terminal set constraint of the form $x[N] \in \mathbb{X}_f$ is implemented into the MPC formulation.

The terminal set is implemented as a convex ellipsoidal constraint based on Riccati's P, obtained by solving the discrete algebraic Riccati equation (DARE). This constraint is written as

$$(x - x_{ref})^\top P (x - x_{ref}) \leq \gamma$$

Where γ is a constant scaling factor calculated using the algorithm described in 2. This level set satisfies the Lyapunov decrease condition, as described in III-B, and thus guarantees stability.

In addition, the controller checks if an admissible sequence exists for x_0 during setup time. If not, the controller informs the user before starting the optimizer.

E. Optimal Target Selection (OTS)

In order to not have to manually find stable reference outputs, OTS was added to the controller. During setup, the controller is passed a reference output y_{ref} and solves the following optimization problem:

$$\begin{aligned} \underset{x_{\text{ref}}, u_{\text{ref}}}{\text{minimize}} \quad & x_{\text{ref}}^T I x_{\text{ref}} + u_{\text{ref}}^T I u_{\text{ref}} \\ \text{subject to} \quad & (I - A_d)x_{\text{ref}} - B_d u_{\text{ref}} = d_{\text{gravity}} \\ & C x_{\text{ref}} = y_{\text{ref}} \\ & H_x x_{\text{ref}} \leq h_x \\ & H_u u_{\text{ref}} \leq h_u \end{aligned}$$

where:

- $x_{\text{ref}} \in \mathbb{R}^8$ is the desired state.
- $u_{\text{ref}} \in \mathbb{R}^2$ is the desired control input.
- $A_d, B_d, C, H_x, H_u, h_x, h_u$ are the system dynamics and constraints matrices and vectors.
- d_{gravity} is the gravity term that the system needs to reject.
- I is the identity matrix of appropriate dimension (8x8 for x_{ref} and 2x2 for u_{ref}).
- y_{ref} is the desired output

The OTS solver computes x_{ref} and u_{ref} that minimize the cost function while satisfying the given constraints.

F. Constant disturbance rejection

In order to overcome steady-state errors due to model mismatch, disturbance rejection for constant (or slowly changing) disturbances was added. In our case, this was used to compensate for gravity, but it can be used for other sources of uncertainty.

In order to capture the disturbance, the dynamics as implemented in the controller are augmented with an extra term d_{est} to become

$$x^+ = A_d x + B_d u + d_{est}$$

d_{est} is initialized as a zero vector. After each prediction of the controller, the initial state and control input are stored. In the next prediction, this saved state and control input are used to make a prediction. This prediction is then used in conjunction with the actual next state (the current initial state) to update the estimated disturbance using a filter:

$$\begin{aligned} x_{pred} &= A_d x_{prev} + B_d u_{prev} \\ d_{raw} &= x_0 - x_{pred} \\ d_{est} &= \alpha_d d_{est} + (1 - \alpha_d) d_{raw} \end{aligned}$$

where:

- x_{pred} is the predicted output.
- d_{raw} is the measured disturbance at the current timestep.
- α_d is the filter coefficient.
- d_{est} is the estimated disturbance vector.

III. Asymptotic stability

This section presents a mathematical and empirical validation demonstrating that the proposed regulation Model Predictive Control (MPC) design achieves asymptotic stabilization of the closed-loop system within the established Region of Attraction (ROA). The verification is conducted by confirming the sufficient conditions delineated in [1]

A. Linearized System Stability Analysis

Proposition 2.1 (Continuity of system solution):

The non-linear dynamics of the system as described in VI $f(\mathbf{x}, \mathbf{u})$ is Lipschitz continuous[1]. The linearized equations as described in VI-B are inherently Lipschitz continuous due to their Linear Time-Invariant (LTI) property. Thus, the system solution $\phi(k; \mathbf{x}, \mathbf{u})$ is continuous for $k \in \mathbb{Z}$

Assumption 2.2 (Continuity of system and cost):: From proposition 2.1 in III-A, the system is Lipschitz continuous. The stage cost (Lagrange cost) $l(\mathbf{x}, \mathbf{u})$ and the terminal cost (Mayer cost) $V_f(\mathbf{x})$ as defined in II are continuous. At equilibrium point $\mathbf{x}_{eq} = \{0\}$ under no input $\mathbf{u}_{null} = 0$, $f(\mathbf{x}_{eq}, \mathbf{u}_{null}) = l(\mathbf{x}_{eq}, \mathbf{u}_{null}) = V_f(\mathbf{x}_{eq}) = 0$. Additionally, the cost penalties $\mathbf{Q}, \mathbf{R}, \mathbf{P}$ are Positive Definite ($\mathcal{PD}$). Since cost terms and penalties are quadratic and $\mathcal{PD}$ respectively, then $l : \mathbb{Z} \to \mathbb{R}_{\geq 0}, V_f : \mathbb{X} \to \mathbb{R}_{\geq 0}$

Assumption 2.3 (Properties of constraint sets):: This assumption is satisfied as $\mathbb{X}$ is closed in $\mathbb{R}^8$ and $\mathbb{U}$ is closed and bounded (i.e., compact) in $\mathbb{R}^2$. Hence, for all pairs of states and inputs, $\mathbb{X} \times \mathbb{U} \to \mathbb{Z}$ is also compact. As terminal set $\mathbb{X}_f$ is designed as the level set of quadratic terminal cost (which is closed in $\mathbb{R}^8$), $\mathbb{X}_f$ is closed and bounded in $\mathbb{R}^8$, hence $\mathbb{X}_f \subseteq \mathbb{X}$.

The equilibrium point is found by finding some $\epsilon > 0$ such that $\mathbf{x}_{eq}$ belongs within that ϵ. This implies origin is not on the boundary of $\mathbb{X}_f$ but in the neighborhood around the origin of $\mathbb{X}_f$ (i.e,) $\mathbf{x}_{eq} \in int(\mathbb{X}_f)$. This is depicted in Figure 2.

Assumption 2.14 (Basic stability assumption):: Under the above two assumptions and the MPC formulation, any controllable (initial) state $x_0 \in \mathcal{X}_N$ is driven into the terminal set at the end of the time-horizon N. Once the state is in $\mathbb{X}_f$, there exists a control law $\kappa(\mathbf{x}) = K\mathbf{x}$, where K is Riccati gain obtained by solving the Discrete Algebraic Riccati Equation (DARE) of infinite horizon unconstrained LQR equation. The terminal set $\mathbb{X}_f$ is constructed in a way such that all the inputs computed from the control law are well-compact (i.e.) $\kappa(\mathbf{x}) \in \mathbb{U}$.

To prove the positive control invariance of $\mathbb{X}_f$ and Lyapunov decrease condition, we prove the following conditions,

(a) $l(\mathbf{x}, \kappa_N(\mathbf{x})) \geq \alpha_1(|\mathbf{x}|) \quad \forall \mathbf{x} \in \mathcal{X}_N; \forall \mathbf{u} \in \mathbb{U}$
(b) $V_f(\mathbf{x}) \leq \alpha_f(|\mathbf{x}|) \quad \forall \mathbf{x} \in \mathbb{X}_f$

where $\alpha_1(.), \alpha_f(.)$ are κ_∞ comparison functions.

[1]All the analyses can be found in https://github.com/BaCyka1/SC42125_project

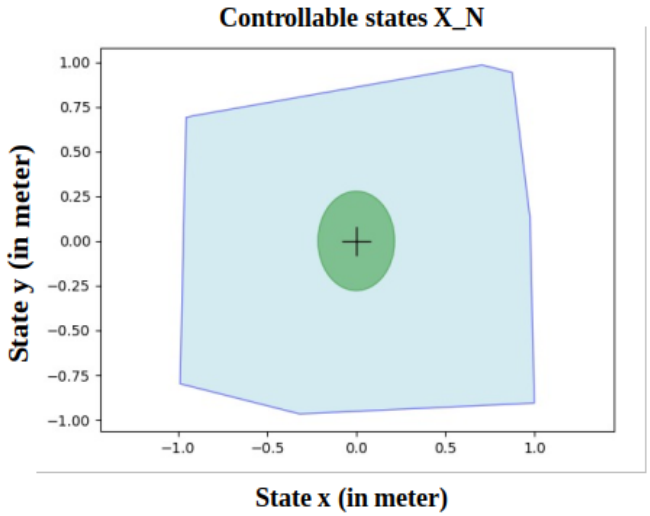

Fig. 2. 2-D mapping of the estimate of $\mathcal{X}_N$ for different x and y initial conditions. The green ellipse represents the terminal state $\mathbb{X}_f$ for states x and y. The plus sign represents the equilibrium point $\{\mathbf{0}\}$

Lemma 2.14(a): Under the LQR control law $\kappa(\mathbf{x}) = K\mathbf{x}$,

$$\begin{aligned} l(\mathbf{x}, \kappa(\mathbf{x})) =& \frac{1}{2}(\mathbf{x}^\top Q\mathbf{x} + (K\mathbf{x})^\top R(K\mathbf{x})) \\ =& \mathbf{x}^\top (Q + K^\top RK)\mathbf{x} \end{aligned} \tag{1}$$

Since $\mathbf{Q}, \mathbf{R}$ is $\mathcal{PD}$ and $K^\top K > 0$, then $(Q + K^\top RK)$ is also $\mathcal{PD}$.

From Rayleigh's inequality (bounds of a positive definite matrix) [2], we write equation 1 as

$$\ell(\mathbf{x}, \kappa(\mathbf{x})) \geq \lambda_{\min}(Q + K^\top RK) \cdot \|\mathbf{x}\|^2 \tag{2}$$

where $\lambda_{\min}$ is the smallest eigenvalue of $(Q + K^\top RK)$

From the equation 2, we write $\alpha_1(.) \in \kappa_\infty$ as

$$\alpha_1(|\mathbf{x}|) = \lambda_{\min}(Q + K^\top RK) \cdot ||\mathbf{x}||^2 \quad \Rightarrow \quad \ell(\mathbf{x}, \kappa(\mathbf{x})) \geq \alpha_1(|\mathbf{x}|)$$

Hence, the assumption 2.14 (a) is proven.

Lemma 2.14(b): From the definition of $V_f(\mathbf{x}) = \mathbf{x}^\top P\mathbf{x}$, where $P \in \mathcal{PD}$, obtained from solving DARE.

$$V_f(\mathbf{x}) \begin{cases} > 0 & \text{if } \mathbf{x} \in \mathbb{X}_f \setminus \{\mathbf{0}\} \\ = 0 & \text{if } \mathbf{x} = \mathbf{0} \end{cases}$$

From Rayleigh's inequality (bounds of a positive definite matrix), we write $\alpha_f(.) \in \kappa_\infty$ as

$$\alpha_f(|\mathbf{x}|) = \lambda_{max}(P)||\mathbf{x}||^2 \quad \Rightarrow \quad V_f(\mathbf{x}) \leq \alpha_f(|\mathbf{x}|) \tag{3}$$

where λ_{max} is the largest eigenvalue of the matrix P. Additionally, this is proven with prepositions 2.15 and 2.18 in [1]. Hence, the assumption 2.14 (b) is proven.

Under the above lemmas, we first prove the Lyapunov decrease condition within the terminal set,

$$V_f(f(\mathbf{x}, \kappa(\mathbf{x}))) - V_f(\mathbf{x}) \leq -\ell(\mathbf{x}, \kappa(\mathbf{x})), \quad \forall \mathbf{x} \in \mathbb{X}_f.$$

Substituting the linear dynamics, $A_K := A + BK$ for x^+

$$V_f(x^+) - V_f(x) = x^\top (A_K^\top P A_K - P)x$$

Rearranging DARE, we get $A_K^\top P A_K - P = -Q - K^\top RK$ and from equation 1, we get

$$V_f(x^+) - V_f(x) = -x^\top (Q + K^\top RK)x = -\ell(\mathbf{x}, \kappa(\mathbf{x})) \tag{4}$$

Hence, the Lyapunov decrease condition within the terminal set is proven.

To check the positive control invariance of the ellipsoidal terminal set under LQR control law, we need to show if $V_f(\mathbf{x}) = \mathbf{x}^\top P\mathbf{x} \leq \gamma \quad \forall x \in \mathbb{X}_f$ then,

$$V_f(\mathbf{x}^+) = {\mathbf{x}^+}^\top P\mathbf{x}^+ \leq \gamma$$

From the Lyapunov decreases condition in 4, we get

$$\begin{aligned} V_f(\mathbf{x}^+) &\leq V_f(\mathbf{x}) - \ell(\mathbf{x}, \kappa(\mathbf{x})) \\ {\mathbf{x}^+}^\top P\mathbf{x}^+ \leq \mathbf{x}^\top P\mathbf{x} - \ell(\mathbf{x}, \kappa(\mathbf{x})) &\leq \mathbf{x}^\top P\mathbf{x} \leq \gamma \\ \Rightarrow \ {\mathbf{x}^+}^\top P\mathbf{x}^+ &\leq \gamma \\ \Rightarrow \ \mathbf{x}^+ &\in \mathbb{X}_f \end{aligned} \tag{5}$$

Hence, the ellipsoidal terminal set is a positive control invariant set.

Under assumptions 2.2, 2.3, and 2.14, it is proven that the equilibrium point $\mathbf{x}_{eq} = \{0\}$ is asymptotically stable, locally with respect to the RoA set $\mathcal{X}_N$ for the linearized dynamics (A, B) under the described MPC control law $\kappa_N(\mathbf{x})$

B. Construction of ellipsoidal terminal set $\mathbb{X}_f$

Following the establishment of stability for the linear system, the objective is to determine an ellipsoidal control invariant and constraint admissible set, denoted as $\mathbb{X}_f$, and a Region of Attraction set, denoted as $\mathcal{X}_N$, such that initial conditions within these sets are guaranteed to converge asymptotically towards the origin.

To ensure that the MPC optimal control input behaves as an unconstrained infinite-horizon LQR control input, the terminal set $\mathbb{X}_f$ is defined. Within this set, the MPC control law $\kappa_N(\mathbf{x})$ simplifies to the linear feedback law Kx, where K is the optimal LQR gain obtained from solving DARE.

The terminal set is designed as the level set of terminal Lyapunov function $V_f = \frac{1}{2}\mathbf{x}^\top P\mathbf{x}$, where P is the solution of DARE.

$$\mathbb{X}_f = \{\mathbf{x} \in \mathbb{R}^8 \mid V_f(\mathbf{x}) \leq \gamma\} \tag{6}$$

subject to $\rho > 0$, $\mathbb{X}_f \subseteq \mathbb{X}$, and $K\mathbb{X}_f \subseteq \mathbb{U}$. Algorithm 1 determines the semi-axes of the 8-dimensional ellipsoid by utilizing the eigenvectors and eigenvalues of the matrix P. The admissibility of control inputs within $\mathbb{X}_f$ is verified by evaluating the corner points of an over-approximated polytope encompassing the ellipsoidal terminal set, ensuring that all control actions computed within $\mathbb{X}_f$ satisfy the control constraints. An optimization problem, as detailed in Algorithm 2, is formulated to compute the optimal scaling factor $\gamma^\star$, which maximizes the volume of $\mathbb{X}_f$ while guaranteeing

control admissibility. This formulation introduces a quadratic terminal constraint into the Optimal Control Problem (OCP), necessitating a Quadratically Constrained Quadratic Program (QCQP) solver. An attempt to tightly linearize the ellipsoidal set using convex hulls, without under-approximating $\mathbb{X}_f$, was undertaken. However, this approach resulted in an intractable number (1,261,786) of linear constraints in the 8-dimensional space. Consequently, the ellipsoidal terminal set representation is adopted for subsequent analysis.

C. Estimation of $\mathcal{X}_N$ using $\mathbb{X}_f$

While not directly used in the optimization, a program for approximating the admissible set $\mathcal{X}_N$ was created for visualization purposes.

In the controller, before online optimization, the initial state is checked using the part of the algorithm that evaluates the admissibility of an initial state. See VI-D for the full algorithm.

D. Confirmation of a-priori assumptions

An empirical validation of the Lyapunov decrease condition is depicted in Figure 3. It is observed that the equation 4 when it enters the terminal set $\mathbb{X}_f$. Notably, initially the two graphs (Blue: $V_f(x^+) - V_f(x)$, Orange: $-\ell(\mathbf{x}, \kappa(\mathbf{x}))$ have an offset. This suggests the state $\mathbf{x}$ has not entered $\mathbb{X}_f$. However, the total cost converges to zero over timesteps.

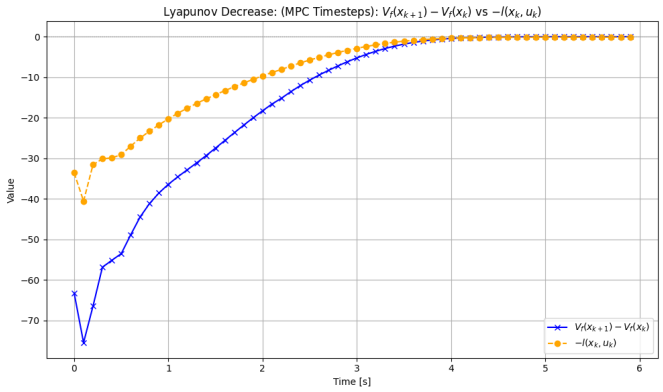

Fig. 3. Empirical validation of Lyapunov decrease condition as in Equation 4

IV. Numerical Simulations

A. MPC with Different Cost Function Weightings

To test the tuning of the Q and R matrices, the tuned controller was compared to the performance of a controller with identity matrices. The controller was made to move from `(0,0)` to `(5,5)`. From Figure 4, it is observed that tuned cost penalty matrices as in section II show a faster time response than identity matrices.

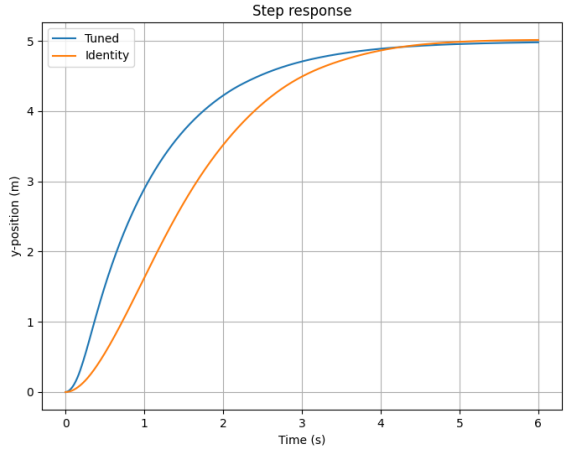

Fig. 4. Step response of MPC with tuned vs untuned matrices

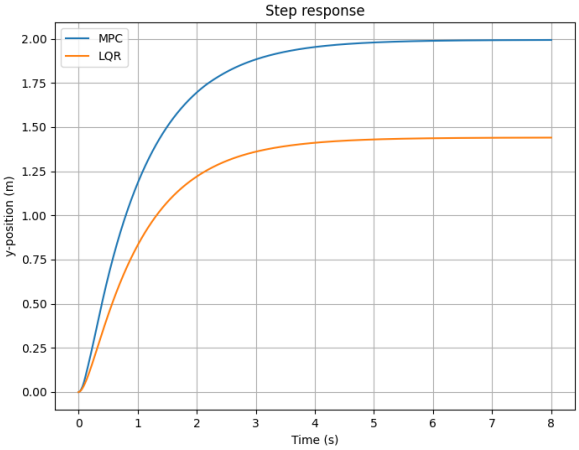

Fig. 5. Step response of MPC with disturbance rejection and unconstrained LQR

B. Output MPC and Disturbance Rejection - Comparison with unconstrained LQR

To test the effectiveness of the disturbance rejection, the controller was compared to an unconstrained LQR controller with a step response. A reference output of (2, 2) was set, and the initial state was the zero vector.

From Figure 5, it is evident that the unconstrained LQR has a state-steady error of around 0.5 m while the MPC controller converges the target reference state.

C. Demo animation

Our custom simulation environment includes graphics. We would like to share one of the test runs as an approachable result. Link to video.

V. Conclusion

The nonlinear dynamics of an underactuated 2D drone with an inverted pendulum were linearized around the hovering equilibrium. While the system is fully controllable, its

under-actuation necessitates a focus on controlling specific output states (position or velocity) through appropriate output matrix selection. Comparative simulations of unconstrained LQR and output-MPC with disturbance rejection demonstrated the significant impact of MPC design parameters on closed-loop system behavior and stability. A rigorous mathematical stability analysis was developed and validated numerically. Future research should explore trajectory tracking in a realistic 3D quadrotor scenario, incorporating unknown disturbance rejection.

REFERENCES

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

[2] Wikipedia. Rayleigh quotient — Wikipedia, The Free Encyclopedia, 2025. http://en.wikipedia.org/w/index.php?title=Rayleigh%20quotient&oldid=1273982356. [Online; accessed 13-April-2025].

[3] M. Dam, M. Saravanan. SC42125_project, 2025. https://github.com/BaCyka1/SC42125_project. [Online; accessed 13-April-2025].

[4] R. DeCarlo. *Linear Systems: A State Variable Approach with Numerical Implementation*. Prentice Hall, 1989.

VI. APPENDIX

A. Non-linear dynamics of the system

The non-linear dynamics equations of a 2D drone are derived using Newton-Euler formulation. Refer to Figure 1 for notations.

$$\begin{aligned} D_{x,y} =& -m_d^2 + m_d m_l \sin^2(\theta) + m_d m_l \cos^2(\theta) \\ & - 2m_d m_l + m_l^2 \sin^2(\theta) + m_l^2 \cos^2(\theta) \\ & - m_l^2 \end{aligned}$$

$$\begin{aligned} \ddot{x} =& \frac{1}{D_{x,y}}[F_T m_d \sin(\psi) - F_T m_l \sin(\psi)\sin^2(\theta) + F_T m_l \sin(\psi) \\ & - F_T m_l \sin(\theta)\cos(\psi)\cos(\theta) - l_r m_d m_l \sin(\theta)\dot{\theta}^2 \\ & + l_r m_l^2 \sin^3(\theta)\dot{\theta}^2 + l_r m_l^2 \sin(\theta)\cos^2(\theta)\dot{\theta}^2 - l_r m_l^2 \sin(\theta)\dot{\theta}^2] \end{aligned}$$

$$\begin{aligned} \ddot{y} =& \frac{1}{D_{x,y}}[-F_T m_d \cos(\psi) - F_T m_l \sin(\psi)\sin(\theta)\cos(\theta) \\ & + F_T m_l \cos(\psi)\cos^2(\theta) - F_T m_l \cos(\psi) \\ & + g m_d^2 - g m_d m_l \sin^2(\theta) \\ & - g m_d m_l \cos^2(\theta) + 2g m_d m_l \\ & - g m_l^2 \sin^2(\theta) - g m_l^2 \cos^2(\theta) \\ & + g m_l^2 + l_r m_d m_l \cos(\theta)\dot{\theta}^2 \\ & - l_r m_l^2 \sin^2(\theta)\cos(\theta)\dot{\theta}^2 \\ & - l_r m_l^2 \cos^3(\theta)\dot{\theta}^2 \\ & + l_r m_l^2 \cos(\theta)\dot{\theta}^2] \end{aligned}$$

$$\ddot{\psi} = \frac{-F_1 l_d + F_2 l_d}{I_d} \tag{7}$$

$$D_\theta = -l_r m_d + l_r m_l \sin^2(\theta) + l_r m_l \cos^2(\theta) - l_r m_l$$

$$\begin{aligned} \ddot{\theta} =& \frac{1}{D_\theta}[-F_1 \sin(\psi)\cos(\theta) + F_1 \sin(\theta)\cos(\psi) \\ & - F_2 \sin(\psi)\cos(\theta) + F_2 \sin(\theta)\cos(\psi)] \end{aligned}$$

B. Linear system dynamics

To linearize the derived non-linear equations, two approximations are made as follows

1) Linearized around hovering of the drone
2) Small angle approximation for $\psi \approx 0$ and $\theta \approx 0$

Note that the system matrices A, and B are in the continuous time domain.

$$A = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & \frac{g(-m_d - m_l)}{m_d} & \frac{g m_l}{m_d} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{g(m_d + m_l)}{l_r m_d} & -\frac{g(m_d + m_l)}{l_r m_d} & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$B = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ \frac{1}{m_d + m_l} & \frac{1}{m_d + m_l} \\ -\frac{l_d}{I_d} & \frac{l_d}{I_d} \\ 0 & 0 \end{bmatrix}$$

C. Appendix - Computing ellipsoidal terminal set

Algorithm 1 computes an ellipsoidal terminal set for MPC and verifies its control feasibility. It is done by checking control input bounds at the vertices of an over-approximating polytope.

Algorithm 1 Constructing Ellipsoid Polytope and Input Constraint Check

procedure CONSTRUCTANDCHECKFEASIBLITY(P, γ, K, u_{lb}, u_{ub})
  $x^T P x \leq \gamma$
  $[\lambda, V] \leftarrow \text{eig}(P)$
  $L \leftarrow \sqrt{\gamma/\lambda}$
  $dim \leftarrow \text{size}(P, 1)$
  $S \leftarrow$ all sign combinations of $[-1, 1]^{\text{dim}}$
  $A \leftarrow \text{diag}(L)$
  $corners \leftarrow (V \cdot A \cdot S^T)^T$
  $u_{lb} \leq u \leq u_{ub}$ for $u = Kx$
  $feasible \leftarrow$ True **for** $i \leftarrow 1$ ***to*** *size*$(corners, 1)$ **do**
    end
    $x_{corner} \leftarrow \text{corners}[i, :]$
  $u \leftarrow K \cdot x_{corner}$ **if** $\neg$*all*$(u_{lb} \leq u \wedge u \leq u_{ub})$ **then**
    end
    $feasible \leftarrow$ False
  return $feasible$
end procedure

Algorithm 2 finds the optimal size ($\gamma^\star$) of an ellipsoidal terminal set by minimizing the terminal set while ensuring control feasibility within the ellipsoidal terminal set.

Algorithm 2 Find Optimal Ellipsoid Size $\gamma^\star$ Under Input Constraints $\mathbb{U}$

function FINDOPTIMALGAMMA($P, K, u_{\text{lb}}, u_{\text{ub}}$)
  Initialize $\gamma \leftarrow 1.0$
  function OBJECTIVE(γ)
    $\mathcal{V} \leftarrow$ Algorithm1(P, γ) **if** $\forall x \in \mathcal{V}, \quad u = Kx \in \mathbb{U}$ **then**
      end
      return $-\gamma$ **else**
      end
      return $+\infty$
  end function
  Solve $\min_\gamma$ Objective(γ) subject to $\gamma > 0$
  return optimal $\gamma^\star$
end function

D. Algorithm for computing feasible/admissible set

Algorithm 3 Estimate Terminal Set $\mathcal{X}_N$ via Sampling

Input: MPC problem $\mathbb{P}_N$, target region definition (H_x, h_x), state space $\mathbb{X}$, input space $\mathbb{U}$, desired number of samples $n_{samples}$
Result: An approximation $\mathcal{X}'_N$ of the terminal set $\mathcal{X}_N$
Initialization:
$\mathcal{X}_f \leftarrow \{x \in \mathbb{R}^n \mid H_x x \leq h_x\}$ ▷ Define target region
$\mathcal{X}'_N \leftarrow \emptyset$ ▷ Initialize empty approximation set
while $|\mathcal{X}'_N| < n_{samples}$ **do**
  end
  Sample a candidate initial condition x_0 ▷ Get a new IC to test (e.g., from $\mathbb{X}$)
Check feasibility of MPC problem $\mathbb{P}_N(x_0, \mathbf{u})$ subject to constraints:
  $x(N) \in \mathcal{X}_f$, $u(k) \in \mathbb{U}$, $x(k) \in \mathbb{X}$ **if** *the MPC problem is feasible for* x_0 **then**
    end
    Result of the check above
$\mathcal{X}'_N \leftarrow \mathcal{X}'_N \cup \{x_0\}$ ▷ Add feasible IC to the set
return $\mathcal{X}'_N$ ▷ Return the estimated terminal set