2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 14-18, 2024. Abu Dhabi, UAE

# Disturbance-Aware Model Predictive Control of Underactuated Robotics Systems

Jiwon Kim and Min Jun Kim

***Abstract*— While robust model predictive control (MPC) has been studied extensively in recent decades, addressing unmatched disturbances in underactuated robotic systems is still challenging. In this paper, we propose a method to enhance the robustness of the MPC through the online estimation of disturbances using a nonlinear disturbance observer (NDOB). We call this method disturbance-aware MPC (DA-MPC), because the proposed method explicitly utilizes the estimated disturbance in the future prediction. We provide a performance analysis of the NDOB, establishing the boundedness between the predicted and real states. The main advantages of the DA-MPC include its applicability to real-time control and its compatibility with off-the-shelf optimal control problem (OCP) solvers. We demonstrate the application of the proposed method using an underactuated quadrotor system. The simulation validation shows the effectiveness of the proposed method compared to $\mathcal{L}_1$-adaptive MPC, which is one of the state-of-the-art robust MPC methods.**

## I. Introduction

Recent advancements in computing power, combined with the development of efficient algorithms [1]–[3], have made it possible to apply model predictive control (MPC) to nonlinear robotic systems. One of the key strengths of the MPC lies in the versatility achieved through optimization schemes. With user-specified objectives and constraints, the MPC generates a natural robot motion by solving the associated optimization problem in real-time [4]–[6]. It should be emphasized that, as the name implies, the MPC generates motion by predicting future behavior using the dynamic model of the robot. Therefore, having accurate knowledge of the robot model is essential to obtain plausible results.

Consequently, in the presence of unknown disturbances that cannot be modeled, predictions of the MPC may significantly deviate from the actual behavior. As a result, the degradation of control performance becomes inevitable, potentially leading to the violation of constraints which are often associated with safety conditions (see the standard MPC case in Fig. 1). To deal with unknown disturbances, a number of robust MPC methods have been studied. One of the most common approaches is to suppress the disturbance, so that the robot is ensured to stay within a specific bound around the MPC's prediction. To this end, various robust control techniques have been combined, such as game-theoretic min-max formulation [7]–[9], sliding mode control [10], incremental input-to-state stability [11], and contraction theory [12]. However, solving robust MPCs for nonlinear robotic systems is often computationally demanding, rendering them challenging to use as real-time controllers.

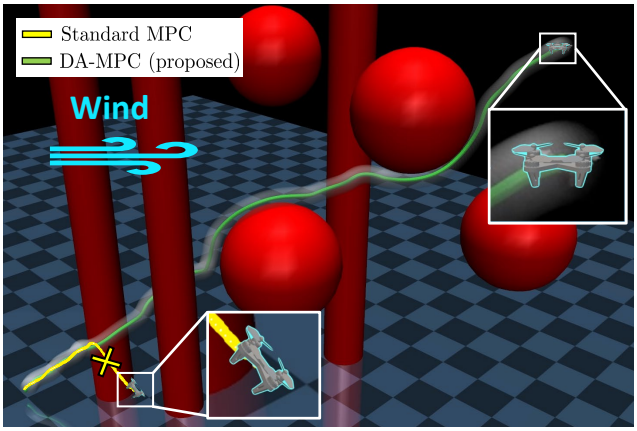


Fig. 1. This paper aims to enable the dynamics model in MPC to be aware of the current disturbances affecting the robot. A nonlinear disturbance observer (NDOB) is employed for online estimation of the disturbance acting on the system. As a result, the proposed method enhances the robustness of the MPC as it can predict the future horizon more accurately.

To enable a real-time implementation of a robust MPC, a noteworthy approach is to combine the MPC with a disturbance observer (DOB) that can compensate for unknown disturbances without heavy computations. However, although some interesting findings have been demonstrated for linear cases [13]–[15], extending these insights to nonlinear robotic systems remains an active area of research [16], [17]. Moreover, for underactuated systems, it is not possible to compensate for all disturbances directly. Consequently, the MPC with the disturbance compensator framework typically compensates only for the input-matched disturbances, while leaving the unmatched terms uncompensated.

In this paper, we take an alternative approach, which utilizes the DOB as an estimator rather than the compensator for underactuated robotic systems. Similar to the previous, although some promising results have been presented for linear systems [18]–[20], few studies have been conducted for nonlinear systems (see, e.g., [21]). For this reason, we first present a nonlinear DOB (NDOB) for real-time estimation of unknown disturbances, including modeling uncertainties and external disturbances. The estimated disturbance is then included in the robot model, so that the MPC can predict the future behavior more accurately. Since the estimated

This work was supported in part by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) under Grant 2021R1C1C1005232, and in part by the ITECH R&D program of MOTIE/KEIT (Project No. 20014398, Development of a kinematically versatile, easy-to-use, safety-supported, AI-integrable, and high speed (>=5kHz) robot controller).

The authors are with Intelligent Robotic Systems Laboratory, Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Republic of Korea. E-mail: {jwkim30, minjun.kim}@kaist.ac.kr

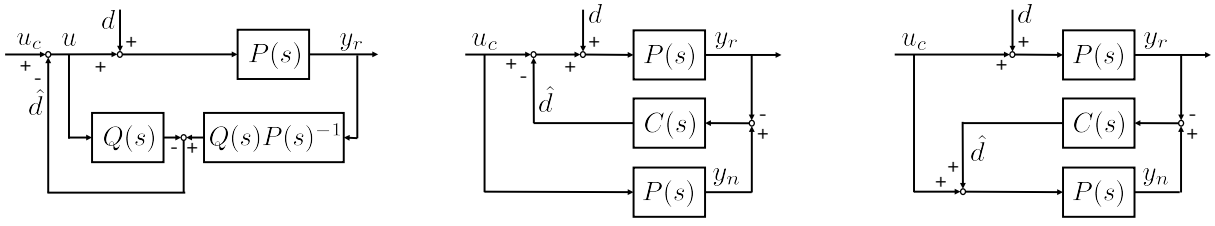


Fig. 2. (a) DOB for compensation, (b) DOB without model inversion for compensation, and (c) DOB without model inversion for estimation.

disturbance is included in the MPC explicitly, the proposed method is called a disturbance-aware MPC (DA-MPC).

To summarize, in this paper, we present DA-MPC which utilizes an NDOB for online estimation of the unknown disturbance. We believe that this paper introduces a novel direction for DOB-based MPC of underactuated robotic systems. The proposed method has the following advantages. First, the DA-MPC approach is well-suited for real-time implementation because the disturbance estimation requires lightweight computation. The disturbance estimate can be computed using a simple PD-type law with a one-step simulation of the rigid body dynamics. Second, although the estimated disturbance is included in the robot model, we do not modify the MPC framework itself. This implies that any off-the-shelf OCP solvers can be employed. In particular, we use the interior-point differential dynamic programming (IP-DDP) [3] to leverage its real-time capability with explicit state constraint. Third, the proposed DA-MPC ensures the boundness between the predicted and real robot states, which can be used to tighten the state constraints. Lastly, it is worth emphasizing once again that the proposed method can be applied to underactuated systems since it involves only the estimation of disturbances without compensation.

The paper is organized as follows. Section II presents preliminaries, and Section III presents the NDOB framework for online estimation of the disturbances. Utilizing this, DA-MPC is proposed in Section IV. The proposed method is validated with simulations in Section V, and Section VI concludes the paper.

*Notations*: Throughout the paper, the subscript $r$ denotes signals from the real system, while subscript $n$ denotes those from the virtual nominal system. In addition, $\tilde{(\cdot)}$ denotes the nominal modeled parameter of $(\cdot)$.

## II. Preliminaries

### A. Disturbance observer (DOB)

Fig. 2(a) shows the standard DOB structure which is widely used for linear systems. Consider the linear system $P(s)$ subject to disturbance $d$ with input $u$ and output $y_r$, where $s$ denotes the Laplace variable. In this section, we assume that $P(s)$ is known to simplify the analysis and provide clear insight. The output of the system $y_r$ and disturbance estimate $\hat{d}$ can be written as follows:

$$y_r = P(s)u_c + \big(1 - Q(s)\big)d, \tag{1}$$

$$\hat{d} \ = Q(s)d, \tag{2}$$

where $u_c$ is control input for the nominal system, and $Q(s)$ is a low-pass filter. Consequently, in the low frequency range where $Q(s) \approx 1$, one can easily see that $y_r = P(s)u_c$ and $\hat{d} = d$. This implies that the system behaves like the nominal system even in the presence of an unknown disturbance.

### B. DOB Without Model Inversion

The standard DOB introduced in the previous section requires the inverse of the system model, which complicates straightforward extension to nonlinear robotic systems. To address this, this section introduces an alternative structure shown in Fig. 2(b) which does not require model inversion [22]. In this framework, the nominal plant is defined as an ideal copy of the real plant, and nominal output $y_n$ is obtained through the simulation. Therefore, the dynamics of the overall system can be expressed as

$$y_r = P(s)(u_c - \hat{d} + d), \tag{3}$$

$$y_n = P(s)u_c. \tag{4}$$

Subtracting (3) from (4), we obtain

$$y_n - y_r = P(s)(\hat{d} - d). \tag{5}$$

The main idea of this structure is to make $y_r$ follow the nominal behavior $y_n$. To this end, in (5), $\hat{d}$ is designed so that $y_n - y_r$ is attenuated against the disturbance $d$. We remark that, with $Q(s) = P(s)C(s)/(1 + P(s)C(s))$, Fig. 2(b) becomes identical to the standard DOB shown in Fig. 2(a).

As another variant, now we apply $\hat{d}$ to the nominal plant instead of the real plant, as shown in Fig. 2(c). For this structure, the overall dynamics are expressed as

$$y_r = P(s)(u_c + d), \tag{6}$$

$$y_n = P(s)(u_c + \hat{d}). \tag{7}$$

Here, by subtracting (6) from (7), we can obtain the same equation as (5). Therefore, the same $\hat{d}$ can be used to attenuate $y_n - y_r$. However, the interpretation is slightly different from the previous: in Fig. 2(c), the disturbance is only estimated without compensation [23]. Consequently, $\hat{d}$ makes $y_n$ follow $y_r$ by applying the estimated disturbance to the nominal plant.

### C. MPC for robotics system

Consider a general robotic system subject to disturbance:

$$\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q}) = \boldsymbol{B}\boldsymbol{u} + \boldsymbol{d}, \tag{8}$$

where $\boldsymbol{q}$ is the generalized coordinate of the $n$-dimensional configuration manifold $\mathcal{Q}^n$, $\boldsymbol{M}(\boldsymbol{q}) \in \mathbb{R}^{n\times n}$ is inertia matrix, $\boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}}) \in \mathbb{R}^{n\times n}$ is Coriolis and centrifugal matrix, $\boldsymbol{g}(\boldsymbol{q}) \in \mathbb{R}^n$ is gravity vector, $\boldsymbol{u} \in \mathbb{R}^{n_u}$ is control input, $\boldsymbol{B} \in \mathbb{R}^{n\times n_u}$ is a matrix that maps control input into generalized force, and $\boldsymbol{d} \in \mathbb{R}^n$ is the unknown external disturbance. This robot dynamics can be represented in input-affine state-space form as follows:

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{g}_u(\boldsymbol{x})\boldsymbol{u} + \boldsymbol{g}_d(\boldsymbol{x})\boldsymbol{d}, \tag{9}$$

with

$$\boldsymbol{f}(\boldsymbol{x}) = \begin{bmatrix} \dot{\boldsymbol{q}} \\ \boldsymbol{M}(\boldsymbol{q})^{-1}(-\boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} - \boldsymbol{g}(\boldsymbol{q})) \end{bmatrix}, \tag{10}$$

$$\boldsymbol{g}_u(\boldsymbol{x}) = \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{M}(\boldsymbol{q})^{-1}\boldsymbol{B} \end{bmatrix}, \quad \boldsymbol{g}_d(\boldsymbol{x}) = \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{M}(\boldsymbol{q})^{-1} \end{bmatrix}, \tag{11}$$

where $\boldsymbol{x} = (\boldsymbol{q}, \dot{\boldsymbol{q}})$. Note that $\boldsymbol{g}_u(\boldsymbol{x})$ and $\boldsymbol{g}_d(\boldsymbol{x})$ in (11) are different. This implies that if the system is underactuated, $\boldsymbol{d}$ cannot be directly compensated.

MPC is based on iteratively solving an optimal control problem (OCP) in a receding manner. The following OCP is solved in standard MPC:

*Problem 1: Standard-MPC*-OCP

$$\boldsymbol{u}^*(\cdot) = \underset{\boldsymbol{u}(\cdot)}{\arg\min} \int_{t_0}^{t_0+T} l(\boldsymbol{x}(t),\boldsymbol{u}(t))dt + m(\boldsymbol{x}(t_0+T)) \tag{12a}$$

$$\text{subject to} \quad \dot{\boldsymbol{x}} = \tilde{\boldsymbol{f}}(\boldsymbol{x}) + \tilde{\boldsymbol{g}}_u(\boldsymbol{x})\boldsymbol{u}, \tag{12b}$$

$$\boldsymbol{x}(t_0) = \boldsymbol{x}_0, \tag{12c}$$

$$\boldsymbol{u}(t) \in \mathcal{U}, \quad \forall t \in [t_0, t_0+T], \tag{12d}$$

$$\boldsymbol{x}(t) \in \mathcal{X}, \quad \forall t \in [t_0, t_0+T], \tag{12e}$$

where $T$ is prediction horizon time, $\boldsymbol{u}^*(\cdot)$ is the optimized control input trajectory from $t_0$ to $t_0+T$, and $l$ and $m$ are the running and terminal cost functions, respectively. (12b) is the robot dynamics constraint using nominal parameters, without unknown disturbance $\boldsymbol{d}$. (12c) is the initial state condition with the current plant's state $\boldsymbol{x}_0$, (12d) is the input constraint to account for the actuation limits, and (12e) is the state constraint employed to guarantee the robot's safety. Problem 1 is solved at each iteration, and only the $\boldsymbol{u}^* = \boldsymbol{u}^*(t_0)$ is applied to the robot.

## III. Disturbance Estimation using NDOB

### A. NDOB framework for disturbance estimation

The proposed NDOB is shown in Fig. 3, in which the linear plant $P(s)$ of Fig. 2(c) is replaced by nonlinear robotic systems. This extension can be made because the structure in Fig. 2(c) requires only a single-step integration of the dynamics, which is the process that can also be performed for nonlinear systems. As shown in Fig. 3, the overall structure consists of the real and nominal robot dynamics which can be written as follows:

$$\boldsymbol{M}_r\ddot{\boldsymbol{q}}_r + \boldsymbol{C}_r\dot{\boldsymbol{q}}_r + \boldsymbol{g}_r = \boldsymbol{B}\boldsymbol{u} + \boldsymbol{d}, \tag{13}$$

$$\tilde{\boldsymbol{M}}_n\ddot{\boldsymbol{q}}_n + \tilde{\boldsymbol{C}}_n\dot{\boldsymbol{q}}_n + \tilde{\boldsymbol{g}}_n = \tilde{\boldsymbol{B}}\boldsymbol{u} + \hat{\boldsymbol{d}}, \tag{14}$$

where a shorthand notation $(\cdot)_r$ (or $(\cdot)_n$) is used for dynamics parameters indicating that $(\cdot)$ is a function of $\boldsymbol{q}_r, \dot{\boldsymbol{q}}_r$ (or $\boldsymbol{q}_n, \dot{\boldsymbol{q}}_n$), and $\hat{\boldsymbol{d}}$ is a disturbance estimate to be designed. Similar to Section II-B, let us subtract (13) from (14):

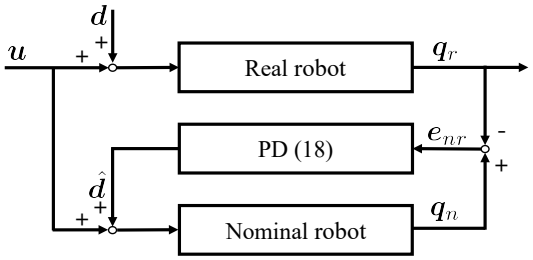


Fig. 3. Nonlinear DOB framework for disturbance estimation.

$$\tilde{\boldsymbol{M}}_n(\ddot{\boldsymbol{e}}_{nr} + \boldsymbol{K}_p\dot{\boldsymbol{e}}_{nr}) + \tilde{\boldsymbol{C}}_n(\dot{\boldsymbol{e}}_{nr} + \boldsymbol{K}_p\boldsymbol{e}_{nr}) = \hat{\boldsymbol{d}} - \boldsymbol{w}, \tag{15}$$

where $\tilde{\boldsymbol{M}}_n\boldsymbol{K}_p\dot{\boldsymbol{e}}_{nr} + \tilde{\boldsymbol{C}}_n\boldsymbol{K}_p\boldsymbol{e}_{nr}$ is added to both sides after subtracting. Here, $\boldsymbol{K}_p = k_p\boldsymbol{I} > 0$ is a gain matrix with identity matrix $\boldsymbol{I}$, $\boldsymbol{e}_{nr} = \boldsymbol{q}_n - \boldsymbol{q}_r$,[1] and $\dot{\boldsymbol{e}}_{nr} = \dot{\boldsymbol{q}}_n - \dot{\boldsymbol{q}}_r$. Moreover,

$$\boldsymbol{w} = \boldsymbol{d} + \boldsymbol{w}_{me} - (\tilde{\boldsymbol{M}}_n\boldsymbol{K}_p\dot{\boldsymbol{e}}_{nr} + \tilde{\boldsymbol{C}}_n\boldsymbol{K}_p\boldsymbol{e}_{nr}) \tag{16}$$

is called an extended disturbance where

$$\begin{aligned}\boldsymbol{w}_{me} =& (\tilde{\boldsymbol{M}}_n\boldsymbol{M}_r^{-1}\boldsymbol{B} - \tilde{\boldsymbol{B}})\boldsymbol{u} + (\tilde{\boldsymbol{C}}_n - \tilde{\boldsymbol{M}}_n\boldsymbol{M}_r^{-1}\boldsymbol{C}_r)\dot{\boldsymbol{q}}_r \\ &+ (\tilde{\boldsymbol{g}}_n - \tilde{\boldsymbol{M}}_n\boldsymbol{M}_r^{-1}\boldsymbol{g}_r) + (\tilde{\boldsymbol{M}}_n\boldsymbol{M}_r^{-1} - \boldsymbol{I})\boldsymbol{d}\end{aligned} \tag{17}$$

corresponds to the disturbance which is mainly caused by modeling error.

As explained in Section II-B, we should design $\hat{\boldsymbol{d}}$ so that $\boldsymbol{x}_n = (\boldsymbol{q}_n, \dot{\boldsymbol{q}}_n)$ follows $\boldsymbol{x}_r = (\boldsymbol{q}_r, \dot{\boldsymbol{q}}_r)$, because this essentially leads $\hat{\boldsymbol{d}} \approx \boldsymbol{w}$ according to (15). To this end, we use the following PD-type disturbance estimation:

$$\hat{\boldsymbol{d}} = -(\boldsymbol{K} + \frac{1}{\gamma^2}\boldsymbol{I})(\dot{\boldsymbol{e}}_{nr} + \boldsymbol{K}_p\boldsymbol{e}_{nr}), \tag{18}$$

where $\boldsymbol{K} > 0$ is a diagonal gain matrix and $\gamma > 0$ is a scalar gain chosen to be a small value. Note that (18) can be interpreted as a control input that forces (15) to behave like $(\dot{\boldsymbol{e}}_{nr} + \boldsymbol{K}_p\boldsymbol{e}_{nr}) \to 0$, akin to the sliding mode control approaches.

*Remark 1:* A subtlety may arise because the gain $\boldsymbol{K}_p$ and the input $\boldsymbol{u}$ are included in (16). This can be problematic if $\boldsymbol{w}$ increases as the observer gets stronger, as it implies that the disturbance estimate also increases. Fortunately, this is not the case for our formulation, because only a small value of $\gamma$ is relevant to the high-gain nature of the observer, while $\boldsymbol{K}_p$ and $\boldsymbol{u}$ are not. Similar treatments can be found, for example, in [23]–[25].

[1]Strictly speaking, the distance between two generalized coordinates should be computed with respect to the configuration manifold. For the sake of simplicity, however, we use the standard arithmetic in the sense of Euclidean throughout the paper. Nevertheless, as we evaluate the proposed method using the quadrotor system whose configuration space is $SE(3)$, an appropriate definition of configuration error will be used in Section V.

### B. Performance analysis of the NDOB

As stated previously, disturbance estimation is achieved by making $\boldsymbol{x}_n \to \boldsymbol{x}_r$. Therefore, by defining $\boldsymbol{x}_{nr} = (\boldsymbol{e}_{nr}, \dot{\boldsymbol{e}}_{nr})$, the performance of the NDOB can be measured by $\|\boldsymbol{x}_{nr}\|$. In the following, we derive the bound of $\|\boldsymbol{x}_{nr}\|$ with respect to the gain $\gamma$. This result will be utilized in Section IV to obtain the state bound of the DA-MPC. We begin with the following theorem that states the finite $\mathcal{L}_2$-gain stability of the proposed NDOB.

*Theorem 1:* Consider the NDOB framework shown in Fig. 3 with $\hat{\boldsymbol{d}}$ defined in (18). If $\boldsymbol{K}$ and $\boldsymbol{K}_p$ are positive definite diagonal matrices, then the following $\mathcal{L}_2$-gain attenuation is satisfied with the finite $\mathcal{L}_2$-gain $\gamma > 0$:

$$\int_0^t (\boldsymbol{x}_{nr}^T \boldsymbol{Q} \boldsymbol{x}_{nr} + \hat{\boldsymbol{d}}^T \boldsymbol{R} \hat{\boldsymbol{d}}) d\tau \leq \gamma^2 \int_0^t (\boldsymbol{w}^T \boldsymbol{w}) d\tau, \tag{19}$$

where

$$\boldsymbol{Q} = \begin{bmatrix} \boldsymbol{K}_p^2 \boldsymbol{K} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{K} \end{bmatrix}, \quad \boldsymbol{R} = (\boldsymbol{K} + \frac{1}{\gamma^2}\boldsymbol{I})^{-1}. \tag{20}$$

*Proof:* The state-space representation of (15) is as follows:

$$\dot{\boldsymbol{x}}_{nr} = \boldsymbol{A}\boldsymbol{x}_{nr} + \boldsymbol{B}(\hat{\boldsymbol{d}} - \boldsymbol{w}), \tag{21}$$

with

$$\boldsymbol{A} = \begin{bmatrix} \boldsymbol{0} & \boldsymbol{I} \\ -\tilde{\boldsymbol{M}}_n^{-1}\tilde{\boldsymbol{C}}_n\boldsymbol{K}_p & -\tilde{\boldsymbol{M}}_n^{-1}\tilde{\boldsymbol{C}}_n - \boldsymbol{K}_p \end{bmatrix}, \ \boldsymbol{B} = \begin{bmatrix} \boldsymbol{0} \\ \tilde{\boldsymbol{M}}_n^{-1} \end{bmatrix}. \tag{22}$$

Let us define the matrix $\boldsymbol{P}$ as

$$\boldsymbol{P} = \begin{bmatrix} \boldsymbol{K}_p\tilde{\boldsymbol{M}}_n\boldsymbol{K}_p + \boldsymbol{K}_p\boldsymbol{K} & \boldsymbol{K}_p\tilde{\boldsymbol{M}}_n \\ \tilde{\boldsymbol{M}}_n\boldsymbol{K}_p & \tilde{\boldsymbol{M}}_n \end{bmatrix}. \tag{23}$$

Applying Schur complement, $\boldsymbol{P}$ is positive definite if $\boldsymbol{K}_p\boldsymbol{K} > 0$, which is true by assumption. Notice that, with (23), the NDOB's estimation (18) can be written as

$$\hat{\boldsymbol{d}} = -\boldsymbol{R}^{-1}\boldsymbol{B}^T\boldsymbol{P}\boldsymbol{x}_{nr}. \tag{24}$$

In [24], it is shown that for Euler-Lagrange systems, which describe general robotics systems, (19) is satisfied when the following differential Riccati equation (DRE) is satisfied:

$$\dot{\boldsymbol{P}} + \boldsymbol{P}\boldsymbol{A} + \boldsymbol{A}^T\boldsymbol{P} - \boldsymbol{P}\boldsymbol{B}(\boldsymbol{R}^{-1} - \frac{1}{\gamma^2})\boldsymbol{B}^T\boldsymbol{P} + \boldsymbol{Q} = \boldsymbol{0}. \tag{25}$$

Therefore, it is sufficient to check the above DRE, which can be easily verified by plugging (20), (22), and (23) into (25) and using $\dot{\tilde{\boldsymbol{M}}}_n - \tilde{\boldsymbol{C}}_n^T - \tilde{\boldsymbol{C}}_n = \boldsymbol{0}$. Hence (19) holds. ∎

Utilizing the result of Theorem 1, we can derive of bound of $\|\boldsymbol{x}_{nr}\|$. Before proceeding, the following assumption is made about the bounded external and modeling error disturbances:

$$\sqrt{\|\boldsymbol{d}\|^2 + \|\boldsymbol{w}_{me}\|^2} \leq \bar{d}. \tag{26}$$

Further, let us define a positive definite matrix $\bar{\boldsymbol{K}}$ as

$$\bar{\boldsymbol{K}} = \boldsymbol{Q} + \boldsymbol{P}\boldsymbol{B}\boldsymbol{R}^{-1}\boldsymbol{B}^T\boldsymbol{P}. \tag{27}$$

Note that as $\gamma$ decreases, the eigenvalues of $\bar{\boldsymbol{K}}$ increase. Considering this, with sufficiently small $\gamma$, we can assume

$$\frac{3\gamma^2}{\lambda_{\min}(\bar{\boldsymbol{K}}) - 3k_p^2 c\gamma^2} \approx \frac{3\gamma^2}{\lambda_{\min}(\bar{\boldsymbol{K}})}, \tag{28}$$

where $\lambda_{\min}(\cdot)$ denotes the minimum eigenvalue of $(\cdot)$. The constant $c$ in (28) is from

$$\|\begin{bmatrix} \tilde{\boldsymbol{C}}_n & \tilde{\boldsymbol{M}}_n \end{bmatrix} \boldsymbol{x}_{nr}\|^2 \leq c\|\boldsymbol{x}_{nr}\|^2. \tag{29}$$

The following theorem states the bound of $\|\boldsymbol{x}_{nr}\|$.

*Theorem 2:* Assume that external and modeling error disturbances are bounded as (26) and $\gamma$ is chosen sufficiently small enough for (28) to be approximated. Then,

$$\|\boldsymbol{x}_{nr}\| \leq \sqrt{\frac{3\gamma^2}{\lambda_{\min}(\bar{\boldsymbol{K}})}}\bar{d}. \tag{30}$$

*Proof:* By substituting (24) into (19), we obtain

$$\boldsymbol{x}_{nr}^T\bar{\boldsymbol{K}}\boldsymbol{x}_{nr} \leq \gamma^2\|\boldsymbol{w}\|^2. \tag{31}$$

Moreover, from (16), $\|\boldsymbol{w}\|^2$ is bounded as

$$\|\boldsymbol{w}\|^2 \leq 3\|\boldsymbol{d}\|^2 + 3\|\boldsymbol{w}_{me}\|^2 + 3\|\boldsymbol{K}_p\begin{bmatrix} \tilde{\boldsymbol{C}}_n & \tilde{\boldsymbol{M}}_n \end{bmatrix}\boldsymbol{x}_{nr}\|^2, \tag{32}$$

where inequality $\|\boldsymbol{a}+\boldsymbol{b}+\boldsymbol{c}\|^2 \leq 3\|\boldsymbol{a}\|^2 + 3\|\boldsymbol{b}\|^2 + 3\|\boldsymbol{c}\|^2$ is used. By substituting (32) into (31), we have

$$\begin{aligned} \boldsymbol{x}_{nr}^T\bar{\boldsymbol{K}}\boldsymbol{x}_{nr} \leq\ & 3\gamma^2\|\boldsymbol{d}\|^2 + 3\gamma^2\|\boldsymbol{w}_{me}\|^2 \\ & + 3\gamma^2\|\boldsymbol{K}_p\begin{bmatrix} \tilde{\boldsymbol{C}}_n & \tilde{\boldsymbol{M}}_n \end{bmatrix}\boldsymbol{x}_{nr}\|^2. \end{aligned} \tag{33}$$

Using $\lambda_{\min}(\bar{\boldsymbol{K}})\|\boldsymbol{x}_{nr}\|^2 \leq \boldsymbol{x}_{nr}^T\bar{\boldsymbol{K}}\boldsymbol{x}_{nr}$ and (29),

$$\|\boldsymbol{x}_{nr}\|^2 \leq \frac{3\gamma^2}{\lambda_{\min}(\bar{\boldsymbol{K}}) - 3k_p^2 c\gamma^2}(\|\boldsymbol{d}\|^2 + \|\boldsymbol{w}_{me}\|^2). \tag{34}$$

Substituting (28) into (34) yields

$$\|\boldsymbol{x}_{nr}\|^2 \leq \frac{3\gamma^2}{\lambda_{\min}(\bar{\boldsymbol{K}})}(\|\boldsymbol{d}\|^2 + \|\boldsymbol{w}_{me}\|^2). \tag{35}$$

Finally, by plugging (26) into (35), we obtain

$$\|\boldsymbol{x}_{nr}\|^2 \leq \frac{3\gamma^2}{\lambda_{\min}(\bar{\boldsymbol{K}})}\bar{d}^2. \tag{36}$$

Therefore (30) follows. ∎

From this theorem, it is once again clear that the NDOB estimates the disturbance accurately with sufficiently small $\gamma$. The bound on $\|\boldsymbol{x}_{nr}\|$ will be used to compute the bound of the proposed DA-MPC in Section IV-B.

## IV. Disturbance-Aware MPC (DA-MPC)

### A. DA-MPC framework

With the disturbance estimate (18) obtained from the NDOB, we can modify the standard MPC to be aware of the disturbances, as shown in Fig 4. Consequently, MPC can predict the future behavior more precisely, because the dynamic model becomes more accurate. This essentially leads to improvement of the control performance.

The OCP solved in the DA-MPC framework can be formulated as follows:

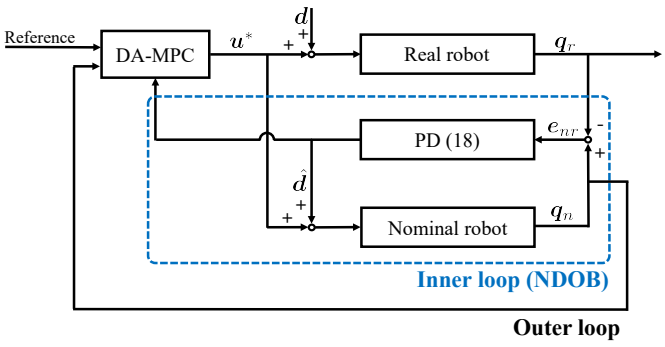


Fig. 4. The proposed DA-MPC framework.

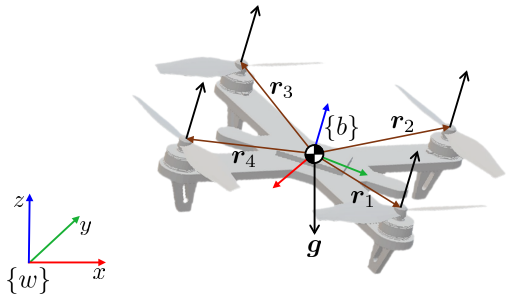


Fig. 5. A quadrotor system used in simulation.

*Problem 2: DA-MPC*-OCP

$$\boldsymbol{u}^*(\cdot) = \arg\min_{\boldsymbol{u}(\cdot)} \int_{t_0}^{t_0+T} l(\boldsymbol{x}(t), \boldsymbol{u}(t))dt + m(\boldsymbol{x}(t_0+T)) \quad (37a)$$

$$\text{subject to} \quad \dot{\boldsymbol{x}} = \tilde{\boldsymbol{f}}(\boldsymbol{x}) + \tilde{\boldsymbol{g}}_u(\boldsymbol{x})\boldsymbol{u} + \tilde{\boldsymbol{g}}_d(\boldsymbol{x})\hat{\boldsymbol{d}}, \quad (37b)$$

$$\boldsymbol{x}(t_0) = \boldsymbol{x}_{n,0}, \quad (37c)$$

$$\boldsymbol{u}(t) \in \mathcal{U}, \quad \forall t \in [t_0, t_0+T], \quad (37d)$$

$$\boldsymbol{x}(t) \in \bar{\mathcal{X}} = \mathcal{X} - \mathcal{S}, \quad \forall t \in [t_0, t_0+T], \quad (37e)$$

where (37b) is robot dynamics constraint with inclusion of $\hat{\boldsymbol{d}}$. The initial condition is set with the nominal system's current state $\boldsymbol{x}_{n,0}$ as shown in (37c). Moreover, $\mathcal{S}$ in (37e), which will be introduced in the following section, can be used to tighten the state constraint to enhance the robustness of constraint satisfaction [26].

The main difference between the Problem 2 and the Problem 1 lies in the inclusion of $\hat{\boldsymbol{d}}$ in the robot model in (37b). This allows the OCP to handle difficult-to-model disturbances, particularly modeling uncertainties and unknown external disturbances. It is important to point out that the DA-MPC modifies the robot model to include $\hat{\boldsymbol{d}}$, rather than compensating for it. In fact, for underactuated systems, it is impossible to compensate for the disturbance when an unmatched term exists. From this point of view, the DA-MPC framework provides a means to incorporate input-unmatched disturbances into the MPC formulation. Moreover, we finally remark that the inner NDOB loop can be implemented at a higher frequency than the outer MPC loop so that the most recent disturbance estimate is provided to the DA-MPC.

### B. Boundness of DA-MPC framework

This section presents the maximum deviation of the real system's state $\boldsymbol{x}_r$ from the state $\boldsymbol{x}_p$ predicted in the DA-MPC, within the time interval $\Delta t$ of the MPC iterations. To begin with, let us write the state error between them as follows:

$$\begin{aligned}\|\boldsymbol{x}_r - \boldsymbol{x}_p\| &= \|-\boldsymbol{x}_{nr} + \boldsymbol{x}_n - \boldsymbol{x}_p\| \\ &\le \|\boldsymbol{x}_{nr}\| + \|\boldsymbol{x}_n - \boldsymbol{x}_p\|. \end{aligned} \quad (38)$$

To investigate $\|\boldsymbol{x}_n - \boldsymbol{x}_p\|$ over $\Delta t$, note that $\boldsymbol{x}_n$ and $\boldsymbol{x}_p$ evolve in the inner loop according to

$$\dot{\boldsymbol{x}}_n = \tilde{\boldsymbol{f}}(\boldsymbol{x}_n) + \tilde{\boldsymbol{g}}_u(\boldsymbol{x}_n)\boldsymbol{u}^* + \tilde{\boldsymbol{g}}_d(\boldsymbol{x}_n)\hat{\boldsymbol{d}}, \quad (39)$$

$$\dot{\boldsymbol{x}}_p = \tilde{\boldsymbol{f}}(\boldsymbol{x}_p) + \tilde{\boldsymbol{g}}_u(\boldsymbol{x}_p)\boldsymbol{u}^* + \tilde{\boldsymbol{g}}_d(\boldsymbol{x}_p)\hat{\boldsymbol{d}}_0, \quad (40)$$

where $\hat{\boldsymbol{d}}_0 = \hat{\boldsymbol{d}}(t_0)$ is the disturbance estimate used in the OCP. Since $\boldsymbol{x}_p(t_0) = \boldsymbol{x}_n(t_0)$ and $\boldsymbol{u}^*$ is constant over $\Delta t$, we can approximate $\int_{t_0}^{t_0+\Delta t} \big(\tilde{\boldsymbol{f}}(\boldsymbol{x}_n) - \tilde{\boldsymbol{f}}(\boldsymbol{x}_p) + \tilde{\boldsymbol{g}}_u(\boldsymbol{x}_n)\boldsymbol{u}^* - \tilde{\boldsymbol{g}}_u(\boldsymbol{x}_p)\boldsymbol{u}^*\big)dt \approx \boldsymbol{0}$ and $\tilde{\boldsymbol{g}}_d(\boldsymbol{x}_n) \approx \tilde{\boldsymbol{g}}_d(\boldsymbol{x}_p)$ over $\Delta t$. Additionally, suppose that the change of the disturbance estimate over a short time interval $\Delta t$ is bounded as

$$\|\hat{\boldsymbol{d}}(t_0+\Delta t) - \hat{\boldsymbol{d}}_0\| \le \bar{r}_d. \quad (41)$$

Then,

$$\begin{aligned}&\|\boldsymbol{x}_n(t_0+\Delta t) - \boldsymbol{x}_p(t_0+\Delta t)\| \\ =&\|\boldsymbol{x}_n(t_0) - \boldsymbol{x}_p(t_0) + \int_{t_0}^{t_0+\Delta t} (\dot{\boldsymbol{x}}_n - \dot{\boldsymbol{x}}_r)dt\| \\ \approx&\|\int_{t_0}^{t_0+\Delta t} \tilde{\boldsymbol{g}}_d(\boldsymbol{x}_n)(\hat{\boldsymbol{d}} - \hat{\boldsymbol{d}}_0)dt\| \le \int_{t_0}^{t_0+\Delta t} \|\tilde{\boldsymbol{g}}_d(\boldsymbol{x}_n)\bar{r}_d\|dt. \end{aligned} \quad (42)$$

Noting that $\|\tilde{\boldsymbol{g}}_d(\boldsymbol{x}_n)\| = \|\tilde{\boldsymbol{M}}_n^{-1}\| = \bar{\sigma}$ where $\bar{\sigma}$ is the maximum singular value of $\tilde{\boldsymbol{M}}_n^{-1}$,

$$\|\boldsymbol{x}_n(t_0+\Delta t) - \boldsymbol{x}_p(t_0+\Delta t)\| \le \max_{\boldsymbol{q}} \big(\bar{\sigma}(\boldsymbol{q})\big)\bar{r}_d\Delta t. \quad (43)$$

Substituting (43) into (38), we get

$$\begin{aligned}\|\boldsymbol{x}_r - \boldsymbol{x}_p\| &\le \|\boldsymbol{x}_{nr}\| + \|\boldsymbol{x}_n - \boldsymbol{x}_p\| \\ &\le \sqrt{\frac{3\gamma^2}{\lambda_{\min}(\bar{\boldsymbol{K}})}}\bar{d} + \max_{\boldsymbol{q}} \big(\bar{\sigma}(\boldsymbol{q})\big)\bar{r}_d\Delta t. \end{aligned} \quad (44)$$

Therefore, the set $\mathcal{S}$ that represents the future bound of $\boldsymbol{x}_r$ can be obtained as follows:

$$\mathcal{S} = \{\boldsymbol{x}_r |\ \|\boldsymbol{x}_r - \boldsymbol{x}_p\| \le \sqrt{\frac{3\gamma^2}{\lambda_{\min}(\bar{\boldsymbol{K}})}}\bar{d} + \max_{\boldsymbol{q}} \big(\bar{\sigma}(\boldsymbol{q})\big)\bar{r}_d\Delta t\}. \quad (45)$$

This set can be used to tighten the state constraint in (37e), thereby enhancing robustness in ensuring the constraint satisfaction of the real state.

## V. Simulation Validation

### A. System and implementation

We aim to verify the proposed method in the MuJoCo simulation environment [27] for the quadrotor system, shown in Fig. 5. $\{w\}$ and $\{b\}$ represent the world frame and the body frame, and $\boldsymbol{r}_i$ for $i = 1, 2, 3, 4$ denotes the position of each rotor in the body frame. In the simulation study, the

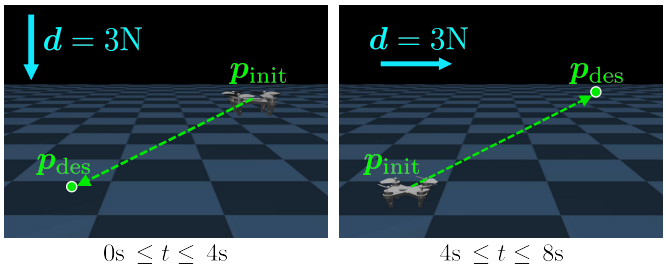


Fig. 6. Scenario 1: Illustration of the simulation scenario.

quadrotor model is assumed to be known, and its dynamic model is given as follows:

$$\begin{bmatrix} m\boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{J} \end{bmatrix} \begin{bmatrix} {}^b\dot{\boldsymbol{v}} \\ {}^b\dot{\boldsymbol{w}} \end{bmatrix} + \begin{bmatrix} m(\boldsymbol{R}_b)^T\boldsymbol{g} \\ {}^b\boldsymbol{w} \times \boldsymbol{J}\,{}^b\boldsymbol{w} \end{bmatrix} = \boldsymbol{B}\boldsymbol{u} + \boldsymbol{d}, \tag{46}$$

where ${}^b\boldsymbol{v}$ and ${}^b\boldsymbol{w}$ are body linear and angular velocity, $\boldsymbol{R}_b \in SO(3)$ is the body rotation matrix, and $\boldsymbol{g} = [0,\ 0,\ 9.81]^T$. $m \in \mathbb{R}$ is body mass, and $\boldsymbol{J} \in \mathbb{R}^{3\times3}$ is the moment of inertia with respect to the body frame. $\boldsymbol{u} \in \mathbb{R}^4$ is thrust vector of 4 rotors, and $\boldsymbol{d} \in \mathbb{R}^6$ is disturbance vector. $\boldsymbol{B} \in \mathbb{R}^{6\times4}$ is defined as

$$\boldsymbol{B} = \begin{bmatrix} \boldsymbol{e}_3 & \boldsymbol{e}_3 & \boldsymbol{e}_3 & \boldsymbol{e}_3 \\ r_{1,y} & r_{2,y} & r_{3,y} & r_{4,y} \\ -r_{1,x} & -r_{2,x} & -r_{3,x} & -r_{4,x} \\ -\kappa & \kappa & -\kappa & \kappa \end{bmatrix}, \tag{47}$$

where $\boldsymbol{e}_3 = [0,\ 0,\ 1]^T$, and $\kappa = c_m/c_f$ is the constant with torque coefficient $c_m$ and thrust coefficient $c_f$. The configuration error is defined as $\boldsymbol{q}_1 - \boldsymbol{q}_0 = \log(\boldsymbol{q}_1^{-1}\boldsymbol{q}_0) \in \mathbb{R}^6$, for $\boldsymbol{q}_0, \boldsymbol{q}_1 \in SE(3)$ [28]. In the following simulations, the system's body mass and moment of inertia are set as $m = 2.18\text{kg}, \boldsymbol{J} = \text{diag}(0.0406,\ 0.0416,\ 0.0813)\text{kgm}^2$.

We implement the IP-DDP algorithm [3] in C++ as the OCP solver, through the discretization of the Problem 2. Dynamics parameters and their derivatives are obtained using the *Pinocchio* C++ library [29]. Through such settings, real-time implementation is achieved. In subsequent experiments, we use prediction time horizon 1s with a step time 0.05s. DA-MPC operates at 500Hz when the state constraint is not considered, and 200Hz when it is taken into account. The NDOB running in the inner loop operates at 1000Hz, so that the DA-MPC can utilize the most recent disturbance estimate. The input constraint for each rotor is set as a box constraint with the range of $[0\text{N}, 20\text{N}]$. In the following comparative studies, the costs of the MPC for the comparison group are set identically, which consists of the position regulation cost, the velocity penalization cost, and the input penalty cost. All the simulation videos can be found in the video attachment and [30].

### B. Comparative study

In this subsection, we aim to validate the effectiveness of using NDOB as the disturbance estimator for underactuated systems. To this end, we compare the proposed DA-MPC with (1) vanilla MPC where Problem 1 is solved, (2) $\mathcal{L}_1$-NMPC [17] where $\mathcal{L}_1$ adaptive controller is additionally

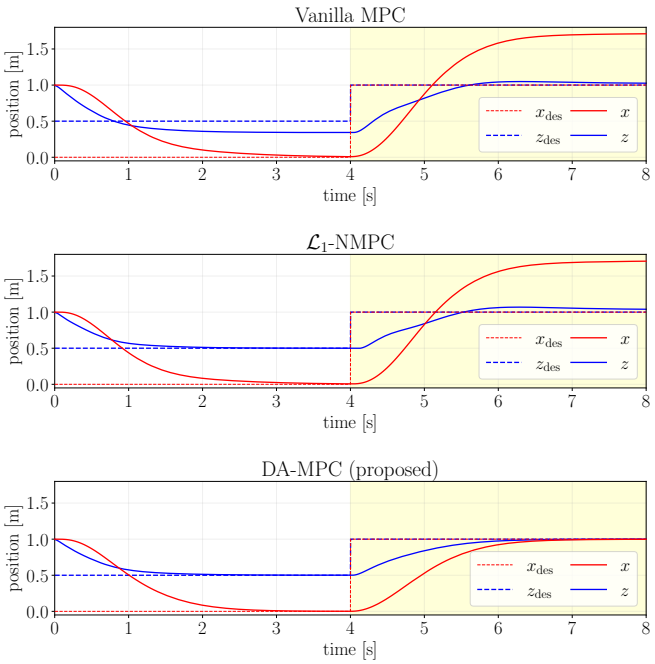


Fig. 7. Scenario 1: The desired and real $x$ and $z$ position for Vanilla MPC (top), $\mathcal{L}_1$-NMPC (middle), and DA-MPC (bottom). In the white and yellow areas, the constant disturbance with the magnitude of 3N is applied in $-z$ direction, $+x$ direction, respectively.

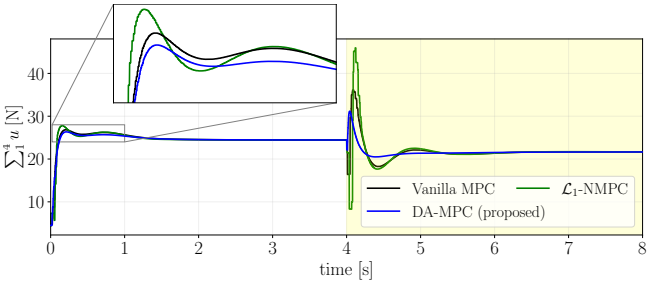


Fig. 8. Scenario 1: Total control input comparison.

used as the disturbance compensator. $\mathcal{L}_1$-NMPC is known for its effectiveness for easy implementation and outstanding performance. However, it only handles matched disturbances which have associated control channels, allowing for direct cancellation. Two scenarios are considered: scenario 1) a regulation problem with constant disturbance, and scenario 2) a tracking problem with varying disturbance.

Fig. 6 depicts the simulation scenario 1. Initially, the quadrotor is located at $\boldsymbol{p}_{\text{init}} = (1, 0, 1)$ and desired position is $\boldsymbol{p}_{\text{des}} = (0, 0, 0.5)$. The desired position is changed to $\boldsymbol{p}_{\text{des}} = (1, 0, 1)$ at $t = 4\text{s}$. A constant disturbance force with a magnitude of 3N is applied to the specific axis of the world frame for each time interval: $-z$ direction during $0\text{s} \leq t \leq 4\text{s}$, and $+x$ direction during $4\text{s} \leq t \leq 8\text{s}$.

Fig. 7 shows resulting $x$ and $z$ position trajectories. Since the vanilla MPC cannot handle the disturbance, the control performance is degraded in both time intervals. When the matched disturbance is applied ($0\text{s} \leq t \leq 4\text{s}$), the position converges well to the desired position for both $\mathcal{L}_1$-NMPC

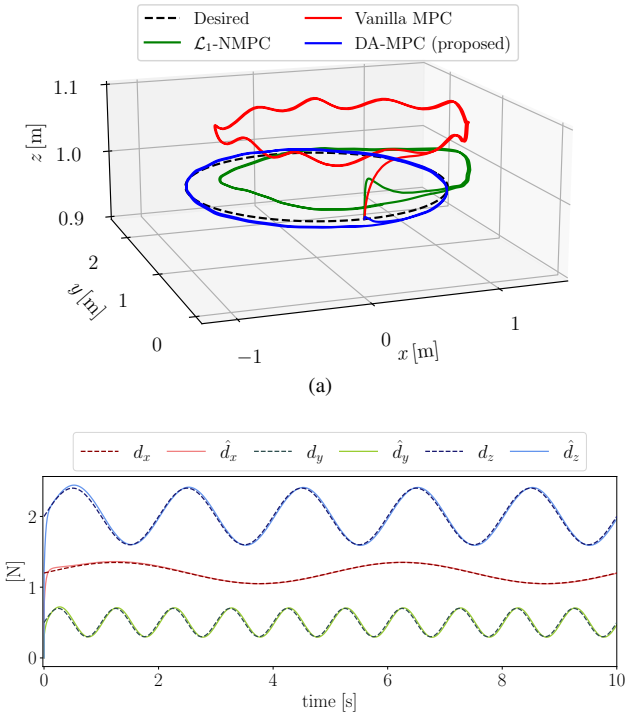


Fig. 9. Scenario 2: (a) Tracking performance comparison. (b) Disturbance estimation performance during 10 seconds ($\gamma^2 = 0.02$).

and DA-MPC. However, when the disturbance includes an unmatched component ($4\text{s} \leq t \leq 8\text{s}$), $\mathcal{L}_1$-NMPC's control performance is degraded because it can compensate only for the input-matched disturbance. In contrast, the DA-MPC uses NDOB as an estimator to adapt the robot dynamics model, and therefore, it can deal with the unmatched disturbance in optimizing the future prediction.

Moreover, Fig. 8 depicts the sum of control inputs (i.e. the total thrust). The DA-MPC tends to use smaller control input than vanilla MPC and $\mathcal{L}_1$-NMPC. Note that, for both time intervals, the disturbance is in a direction that helps move toward the goal position. Since the disturbance is included in the dynamics of the robot, MPC tries to leverage this when optimizing the future behavior. The DA-MPC exploits the disturbance to reach the goal with less control input, whereas the $\mathcal{L}_1$-NMPC tends to use more inputs because it tries to compensate for the disturbance.

In scenario 2, a circular trajectory of 1m radius is used as the reference position for a duration of 20 seconds. The sinusoidal disturbance is applied: $\boldsymbol{d} = [1.2 + 0.15\sin(0.4\pi t),\ 0.5{+}0.2\sin(2\pi t),\ 2{+}0.4\sin(\pi t),\ 0,\ 0,\ 0]$. Fig. 9(a) shows the position plot throughout two complete rotations. For the vanilla MPC, it can be observed that the control performance is significantly degraded. $\mathcal{L}_1$-NMPC shows improved tracking performance in $z$ direction as it can compensate for the matched disturbances. The proposed DA-MPC shows improved control performance in all directions as expected. In addition, Fig. 9(b) confirms that the NDOB can effectively estimate the unknown disturbances.

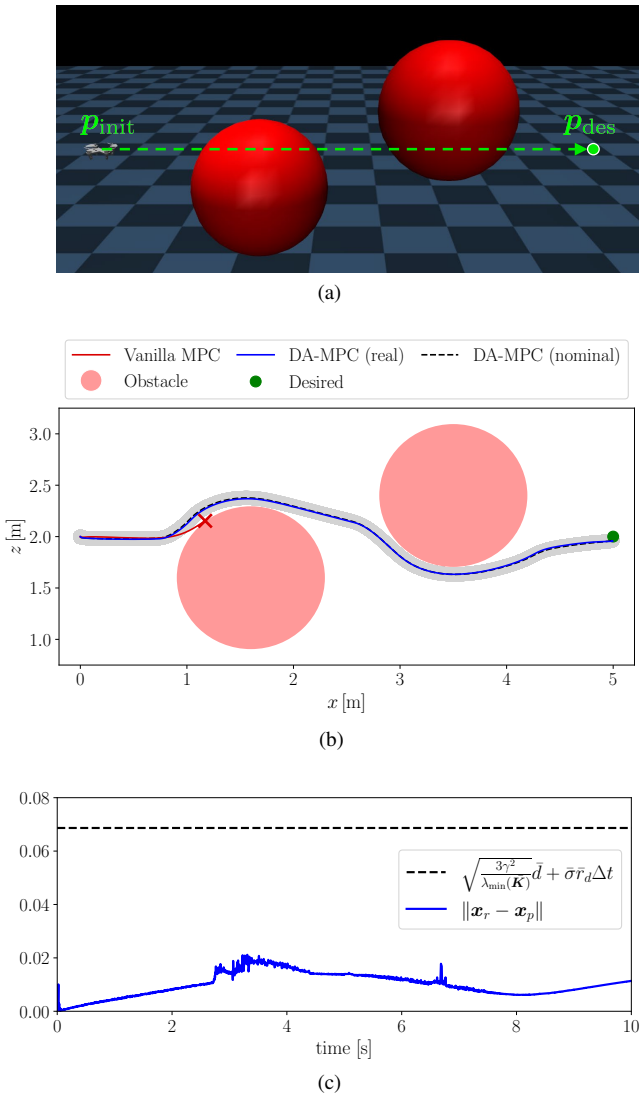


Fig. 10. Scenario 3: (a) Illustration of the simulation environment. (b) A resulting trajectory in $xz$ plane. The gray area throughout the nominal trajectory indicates the maximum norm of $\boldsymbol{x}_r - \boldsymbol{x}_p$, which is used in state constraint tightening. (c) Boundness of $\|\boldsymbol{x}_r - \boldsymbol{x}_p\|$.

### C. Obstacle avoidance scenario

In this subsection, we validate the proposed method for the obstacle avoidance task by additionally considering state constraints. Fig. 10(a) illustrates the simulation environment of scenario 3. The desired trajectory is defined as a linear interpolation from the initial position $\boldsymbol{p}_{\text{init}} = (0, 0, 2)$ to the desired position $\boldsymbol{p}_{\text{des}} = (5, 0, 2)$ over a period of 10 seconds. To avoid collision with obstacles, they are incorporated as position state constraints. Moreover, sinusoidal disturbance is applied: $\boldsymbol{d} = \sin(\pi t/7)[0.5,\ 0.5,\ -0.5,\ 0.1,\ 0.1,\ 0.1]^T$. The state bound $\mathcal{S}$ (45) is computed by setting $\bar{d}$ as maximum value $\|\boldsymbol{d}\|$, and $\bar{r}_d$ as a maximum change of $\|\boldsymbol{d}\|$ over $\Delta t = 0.005\text{s}$.

Fig. 10(b) illustrates the resulting trajectories in $xz$ plane. The DA-MPC can avoid the obstacles robustly by exploiting the estimated disturbance together with state constraint tightening. In contrast, the vanilla MPC resulted in a crash due to the disturbance. Fig. 10(c) shows the norm of $\boldsymbol{x}_r - \boldsymbol{x}_p$ with

its theoretical bound (45). We can observe that this bound is sufficiently conservative to be used in constraint tightening.

In scenario 4, we validate the proposed method in an environment cluttered with obstacles, as depicted in Fig. 1. This scenario is similar to scenario 3, where the reference trajectory is linearly interpolated from the initial position to the desired position, and the same sinusoidal disturbance is applied. The only difference from scenario 3 lies in the increased number of obstacles. Fig. 1 illustrates the resulting trajectory for both the proposed DA-MPC and the standard MPC.

## VI. Conclusion

In this paper, we propose the DA-MPC (Disturbance-Aware Model Predictive Control) framework for underactuated robotic systems. Within this framework, the MPC can predict the future horizon more accurately, using the online estimation of the disturbance provided by the NDOB. We presented the performance bounds of the NDOB and DA-MPC, which can be used to tighten the state constraint. In addition, we highlight that the proposed method (i) can effectively deal with unmatched disturbance which is challenging for existing methods; and (ii) is suitable for real-time implementation, because the augmentation of the NDOB does not require heavy computation. The proposed method can be applied to the existing state-of-the-art OCP solvers, and our particular choice was IP-DDP, which can include state constraints explicitly in real-time. Comparative studies conducted in simulation validate the effectiveness of the proposed DA-MPC framework. Future work includes providing theoretical guarantees for safety and validating the proposed method in real-world scenarios.

## References


[1] Y. Tassa, T. Erez, and E. Todorov, "Synthesis and stabilization of complex behaviors through online trajectory optimization," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2012, pp. 4906–4913.

[2] C. Mastalli, R. Budhiraja, W. Merkt, G. Saurel, B. Hammoud, M. Naveau, J. Carpentier, L. Righetti, S. Vijayakumar, and N. Mansard, "Crocoddyl: An efficient and versatile framework for multi-contact optimal control," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2020, pp. 2536–2542.

[3] A. Pavlov, I. Shames, and C. Manzie, "Interior point differential dynamic programming," *IEEE Transactions on Control Systems Technology*, vol. 29, no. 6, pp. 2720–2727, 2021.

[4] D. Falanga, P. Foehn, P. Lu, and D. Scaramuzza, "Pampc: Perception-aware model predictive control for quadrotors," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018, pp. 1–8.

[5] M. V. Minniti, R. Grandia, K. Fäh, F. Farshidian, and M. Hutter, "Model predictive robot-environment interaction control for mobile manipulation tasks," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 1651–1657.

[6] S. W. Han, M. Iskandar, J. Lee, and M. J. Kim, "Online multi-contact feedback model predictive control for interactive robotic tasks," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2024, pp. 11 556–11 562.

[7] M. Lazar, D. M. De La Pena, W. M. H. Heemels, and T. Alamo, "On input-to-state stability of min–max nonlinear model predictive control," *Systems & Control Letters*, vol. 57, no. 1, pp. 39–48, 2008.

[8] D. M. Raimondo, D. Limon, M. Lazar, L. Magni, and E. F. ndez Camacho, "Min-max model predictive control of nonlinear systems: A unifying overview on stability," *European Journal of Control*, vol. 15, no. 1, pp. 5–21, 2009.

[9] S. L. Herbert, M. Chen, S. Han, S. Bansal, J. F. Fisac, and C. J. Tomlin, "Fastrack: A modular framework for fast and guaranteed safe motion planning," in *IEEE 56th Conference on Decision and Control (CDC)*, 2017, pp. 1517–1522.

[10] B. T. Lopez, J.-J. E. Slotine, and J. P. How, "Dynamic tube mpc for nonlinear systems," in *American Control Conference (ACC)*, 2019, pp. 1655–1662.

[11] J. Köhler, R. Soloperto, M. A. Müller, and F. Allgöwer, "A computationally efficient robust model predictive control framework for uncertain nonlinear systems," *IEEE Transactions on Automatic Control*, vol. 66, no. 2, pp. 794–801, 2020.

[12] S. Singh, B. Landry, A. Majumdar, J.-J. Slotine, and M. Pavone, "Robust feedback motion planning via contraction theory," *The International Journal of Robotics Research*, vol. 42, no. 9, pp. 655–688, 2023.

[13] J. Yang, S. Li, X. Chen, and Q. Li, "Disturbance rejection of dead-time processes using disturbance observer and model predictive control," *Chemical Engineering Research and Design*, vol. 89, no. 2, pp. 125–135, 2011.

[14] J. Yang, W. X. Zheng, S. Li, B. Wu, and M. Cheng, "Design of a prediction-accuracy-enhanced continuous-time mpc for disturbed systems via a disturbance observer," *IEEE Transactions on Industrial Electronics*, vol. 62, no. 9, pp. 5807–5816, 2015.

[15] Y. Zhang, C. Edwards, M. Belmont, and G. Li, "Robust model predictive control for constrained linear system based on a sliding mode disturbance observer," *Automatica*, vol. 154, p. 111101, 2023.

[16] H. Xie, L. Dai, Y. Lu, and Y. Xia, "Disturbance rejection mpc framework for input-affine nonlinear systems," *IEEE Transactions on Automatic Control*, vol. 67, no. 12, pp. 6595–6610, 2021.

[17] D. Hanover, P. Foehn, S. Sun, E. Kaufmann, and D. Scaramuzza, "Performance, precision, and payloads: Adaptive nonlinear mpc for quadrotors," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 690–697, 2021.

[18] G. Pannocchia and A. Bemporad, "Combined design of disturbance model and observer for offset-free model predictive control," *IEEE Transactions on Automatic Control*, vol. 52, no. 6, pp. 1048–1053, 2007.

[19] U. Maeder, F. Borrelli, and M. Morari, "Linear offset-free model predictive control," *Automatica*, vol. 45, no. 10, pp. 2214–2222, 2009.

[20] G. Pannocchia, "Offset-free tracking mpc: A tutorial review and comparison of different formulations," in *European Control Conference (ECC)*, 2015, pp. 527–532.

[21] M. Morari and U. Maeder, "Nonlinear offset-free model predictive control," *Automatica*, vol. 48, no. 9, pp. 2059–2067, 2012.

[22] B. K. Kim, H.-T. Choi, W. K. Chung, and I. H. Suh, "Analysis and design of robust motion controllers in the unified framework," *J. Dyn. Sys., Meas., Control*, vol. 124, no. 2, pp. 313–321, 2002.

[23] M. J. Kim, Y. J. Park, and W. K. Chung, "Design of a momentum-based disturbance observer for rigid and flexible joint robots," *Intelligent Service Robotics*, vol. 8, pp. 57–65, 2015.

[24] M. J. Kim, Y. Choi, and W. K. Chung, "Bringing nonlinear $\mathcal{H}_\infty$ optimality to robot controllers," *IEEE Transactions on Robotics*, vol. 31, no. 3, pp. 682–698, 2015.

[25] M. J. Kim, F. Beck, C. Ott, and A. Albu-Schäffer, "Model-free friction observers for flexible joint robots with torque measurements," *IEEE Transactions on Robotics*, vol. 35, no. 6, pp. 1508–1515, 2019.

[26] J. B. Rawlings, D. Q. Mayne, and M. Diehl, *Model Predictive Control: Theory, Computation, and Design, 2nd Edition*. Nob Hill Publishing LLC, 2017.

[27] E. Todorov, T. Erez, and Y. Tassa, "Mujoco: A physics engine for model-based control," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2012, pp. 5026–5033.

[28] J. Sola, J. Deray, and D. Atchuthan, "A micro lie theory for state estimation in robotics," *arXiv preprint arXiv:1812.01537*, 2018.

[29] J. Carpentier, G. Saurel, G. Buondonno, J. Mirabel, F. Lamiraux, O. Stasse, and N. Mansard, "The pinocchio c++ library : A fast and flexible implementation of rigid body dynamics algorithms and their analytical derivatives," in *IEEE/SICE International Symposium on System Integration (SII)*, 2019, pp. 614–619.

[30] KAIST IRSL, "Disturbance-Aware Model Predictive Control of Underactuated Robotics Systems (IROS2024)," (Sep. 11, 2024). Accessed: Sep. 11, 2024. [Online Video]. Available: https://youtu.be/gh37BL7gssw