2020 American Control Conference
Denver, CO, USA, July 1-3, 2020

Hierarchical MPC with Coordinating Terminal Costs*

Vignesh Raghuraman, Venkatraman Renganathan, Tyler H. Summers, and Justin P. Koeln

Abstract— **The performance of hierarchical Model Predictive Control (MPC) is highly dependent on the mechanisms used to coordinate the decisions made by controllers at different levels of the hierarchy. Conventionally, reference tracking serves as the primary coordination mechanism, where optimal state and input trajectories determined by upper-level controllers are communicated down the hierarchy to be tracked by lower-level controllers. As such, significant tuning is required for each controller in the hierarchy to achieve the desired closed-loop system performance. This paper presents a novel terminal cost coordination mechanism using constrained zonotopes, designed to improve system performance under hierarchical control. These terminal costs allow lower-level controllers to balance both short- and long-term control performance without the need for controller tuning. Unlike terminal costs widely used to guarantee MPC stability, the proposed terminal costs are time-varying and computed on-line based on the optimal state trajectory of the upper-level controllers. A numerical example demonstrates the provable performance benefits achieved using the proposed terminal cost coordination mechanism.**

I. INTRODUCTION

Model Predictive Control (MPC) is well-suited for the control of constrained systems since input and state constraints are directly imposed in the underlying optimization problem [1, 2]. However, for systems that require fast control update rates and long prediction horizons, real-time implementation of centralized MPC is hindered by the time required to solve large optimization problems. In such situations, hierarchical MPC reduces computational costs by decomposing control decisions across multiple levels of controllers operating in different timescales [3].

Coordination between controllers at different levels of the hierarchical controller is typically achieved via reference tracking, where state and input trajectories determined by upper-level controllers are communicated down the hierarchy to be tracked by lower-level controllers. Since the closed-loop system behavior is heavily dependent on the weightings used to add reference tracking to the cost function of each controller, guaranteeing constraint satisfaction and control performance is very difficult.

To guarantee input and state constraint satisfaction, coordination mechanisms based on terminal constraints were introduced in [4, 5]. Specifically, *waysets* were defined based on reachability analysis that represent a subset of states at a future point in time from which there exist feasible state and input trajectories for the remainder of system operation. Thus, driving the system states to a wayset provides a short-term control objective that guarantees long-term constraint satisfaction. While similar wayset-based coordination strategies have been used in [6], those waysets are computed off-line in a feed-forward fashion.

To provide improved control performance in the presence of disturbances, [5] computes waysets on-line based on the optimal state trajectories determined by upper-level controllers. To achieve the computational efficiency required for on-line computation, constrained zonotopes [7] were shown in [4, 5] to provide several orders-of-magnitude reduction in wayset computation time compared to conventional set representations.

While using waysets to achieve guaranteed constraint satisfaction, this paper focuses on improving the coordination between controllers within a hierarchy using specifically designed terminal costs to provide closed-loop control performance guarantees. Terminal costs are widely used to guarantee MPC stability by quantifying system operation cost beyond the finite prediction horizon [1]. Within the proposed hierarchical MPC formulation, terminal costs are imposed on the lower-level controller to quantify a specific state transition cost subject to constraints. For a controller with quadratic stage costs, capturing this state transition cost as a function of a terminal state would result in a time-varying piecewise quadratic cost [8]. However, the present paper shows that it is possible to efficiently compute the desired terminal costs on-line in terms of the same constrained zonotope variables used to define the wayset terminal constraint. Thus, the proposed addition of terminal costs to a wayset-based hierarchical MPC controller provides provable control performance guarantees without any additional complexity.

The contributions of this paper are 1) the development of a two-level hierarchical MPC framework with guaranteed constraint satisfaction and control performance, 2) the formulation of terminal costs that allow the lower-level controller to balance both short- and long-term control performance, and 3) the novel representation of the terminal cost using the same variables that define the wayset terminal constraints as constrained zonotopes to achieve computational efficiency.

The remainder of the paper is organized as follows. Section II presents the class of constrained discrete-time linear systems and the proposed wayset-based two-level hierarchical MPC formulation with proven constraint satisfaction. Section III provides the main result of the paper by guaranteeing the hierarchical control performance through the use of terminal costs. The details of terminal cost representation and computation through the use of constrained zonotopes

*This material is based upon work supported by the National Science Foundation under Grant No. 1849500 and the Air Force Office of Scientific Research under Award No. FA2386-19-1-4073.

The authors are with the Department of Mechanical Engineering, The University of Texas at Dallas, 800 W. Campbell Rd, Richardson, TX, USA. Email: (vignesh.raghuraman, vrengana, tyler.summers, justin.koeln)@utdallas.edu.

are presented in Section IV. A numerical example in Section V demonstrates the value of terminal costs in hierarchical MPC. Finally, Section VI summarizes the conclusions of the paper and provides future research directions.

Notation

For a discrete-time system, the notation $x(k)$ denotes the state x at time step k. For MPC, the double index notation $x(k+j|k)$ denotes the predicted state at future time $k+j$ determined at time step k. The bracket notation $k \in [0, k_F]$ denotes all integer values of k from 0 to k_F. The state trajectory over these time indices is denoted $\{x(k)\}_{k=0}^{k_F}$. The set of positive integers is $\mathbb{Z}_+$. The weighted norm is defined as $\|x\|_\Lambda^2 = x^T \Lambda x$, where Λ is a positive definite diagonal matrix. For sets $\mathcal{Z}, \mathcal{W} \subset \mathbb{R}^n$, $\mathcal{Y} \subset \mathbb{R}^m$, and matrix $R \in \mathbb{R}^{m \times n}$, the linear transformation of $\mathcal{Z}$ under R is $R\mathcal{Z} = \{Rz \mid z \in \mathcal{Z}\}$, the Minkowski sum of $\mathcal{Z}$ and $\mathcal{W}$ is $\mathcal{Z} \oplus \mathcal{W} = \{z + w \mid z \in \mathcal{Z}, w \in \mathcal{W}\}$, and the generalized intersection of $\mathcal{Z}$ and $\mathcal{Y}$ under R is $\mathcal{Z} \cap_R \mathcal{Y} = \{z \in \mathcal{Z} \mid Rz \in \mathcal{Y}\}$. The standard intersection, corresponding to the identity matrix $R = I_n$, is simply denoted as $\mathcal{Z} \cap \mathcal{W}$.

II. PROBLEM FORMULATION

As in [4], consider the discrete linear time-invariant system

$$x(k+1) = Ax(k) + Bu(k), \tag{1}$$

where $x \in \mathbb{R}^n$ are the states, $u \in \mathbb{R}^m$ are the inputs, $A \in \mathbb{R}^{n \times n}$ is invertible, $B \in \mathbb{R}^{n \times m}$, and the pair (A, B) is stabilizable.

Assumption 1. *With a fixed time step Δt, the system operates for a finite length of time starting from $t = 0$ and ending at $t = t_F = k_F \Delta t$ with time steps indexed by $k \in [0, k_F]$.*

Starting from an initial condition $x(0)$, the goal is to plan and execute an input trajectory $\{u(k)\}_{k=0}^{k_F-1}$ and corresponding state trajectory $\{x(k)\}_{k=0}^{k_F}$ satisfying the system dynamics from (1), the state and input constraints

$$x(k) \in \mathcal{X} \subset \mathbb{R}^n, \; u(k) \in \mathcal{U} \subset \mathbb{R}^m, \; \forall k \in [0, k_F - 1], \tag{2}$$

and the terminal constraint

$$x(k_F) \in \mathcal{T} \subseteq \mathcal{X}. \tag{3}$$

Assumption 2. *The sets $\mathcal{X}, \mathcal{U}, \mathcal{T}$ are compact and convex.*

For notational simplicity, the state and input constraints from (2) are represented as the output constraints

$$y(k) \triangleq \begin{bmatrix} x(k) \\ u(k) \end{bmatrix} = Cx(k) + Du(k) \in \mathcal{Y} \triangleq \mathcal{X} \times \mathcal{U}, \tag{4}$$

where $[C \; D] = I_{n+m}$.

A generic cost function defines the desired system operation using a pre-determined reference trajectory $\{r(k)\}_{k=0}^{k_F}$ where

$$J^*(x(0)) = \min_{\{u(k)\}_{k=0}^{k_F-1}} \sum_{j=0}^{k_F-1} \ell(j) + \ell_T(k_F), \tag{5}$$

with stage costs $\ell(j) = \ell(x(j), u(j), r(j))$ and terminal cost $\ell_T(k_F) = \ell_T(x(k_F))$.

Considering the system (1), terminal constraint (3), output constraints (4), and cost function (5), this paper extends the wayset-based vertical hierarchical MPC approach developed in [4] to include terminal costs for guaranteed control performance in addition to guaranteed constraint satisfaction.

A. Vertical Hierarchical Control

While [4, 5] provide vertical hierarchical MPC formulations with M levels of controllers $\mathbf{C}_i, i \in [1, M]$, this paper will focus on the two-level case, $M = 2$, for clarity of exposition. The prediction horizons and time steps for the upper-level controller, $\mathbf{C}_1$, and the lower-level controller, $\mathbf{C}_2$, satisfy the following assumptions.

Assumption 3. *For each controller $\mathbf{C}_i, i \in [1, 2]$, the prediction horizon $N_i \in \mathbb{Z}_+$ and time step $\Delta t_i > 0$ satisfy*

i) $\Delta t_2 = \Delta t$;
ii) $\Delta t_1 = N_2 \Delta t_2$;
iii) $\Delta t_1 = \frac{t_F}{N_1}$.

These assumptions indicate i) the lower-level controller $\mathbf{C}_2$ and the system (1) have the same time step, ii) $\mathbf{C}_2$ predicts state and input trajectories between consecutive updates of the upper-level controller $\mathbf{C}_1$, and iii) $\mathbf{C}_1$ predicts to the end of system operation. Additionally, let $\nu_i \triangleq \frac{\Delta t_i}{\Delta t} \in \mathbb{Z}_+$, be the time scaling factor where $\nu_1 = N_2$ and $\nu_2 = 1$. The time steps for $\mathbf{C}_1$ are indexed by k_1, where $k_1 \triangleq \frac{k}{\nu_1}$ and the time steps for $\mathbf{C}_2$ are the same as those for the system, $k_2 = k$. Let $k_{1,F} \triangleq \frac{k_F}{\nu_1}$ such that $k_1 \in [0, k_{1,F}]$.

While similar, the optimization problems for each of the two controllers are presented separately to highlight the key differences. Details of these controller formulations are further explained and used to make constraint satisfaction and performance guarantees in Sections II-B and III.

The upper-level controller $\mathbf{C}_1$ updates when $k = \nu_1 k_1$ (i.e. when $k \mod \nu_1 = 0$), by solving the constrained optimization problem $\mathbf{P}_1(x(k))$ defined as

$$J_1^*(x(k)) = \min_{\substack{x_1(k_1|k_1) \\ U_1(k_1)}} \sum_{j=k_1}^{k_{1,F}-1} \ell(j|k_1) + \ell_T(k_{1,F}) + \ell_p(k_1), \tag{6a}$$

s.t. $\forall j \in [k_1, k_{1,F} - 1]$

$$x_1(j+1|k_1) = A_1 x_1(j|k_1) + B_1 u_1(j|k_1), \tag{6b}$$

$$y_1(j|k_1) = Cx_1(j|k_1) + Du_1(j|k_1) \in \mathcal{Y}_1, \tag{6c}$$

$$x_1(k_{1,F}|k_1) \in \mathcal{T}, \tag{6d}$$

$$x_1(k_1|k_1) = x(k) \vee x_1^*(k_1|k_1 - 1). \tag{6e}$$

First, note that $\mathbf{C}_1$ has a *shrinking horizon*, based on the summation limits in (6a), since $\mathbf{C}_1$ predicts to the end of system operation. The input sequence is defined as $U_1(k_1) = \{u_1(j|k_1)\}_{j=k_1}^{k_{1,F}-1}$, with the optimal sequence denoted as $U_1^*(k_1)$. In (6a), the stage cost is a function of states, inputs, and references such that $\ell(j|k_1) = \ell(x_1(j|k_1), u_1(j|k_1), r_1(j))$. The penalty cost $\ell_p(k_1)$ is described in Section IV-A and is used to guarantee the performance of the hierarchical controller. In (6b), the model used by $\mathbf{C}_1$ assumes a piecewise constant control input over the time step Δt_1 and thus $A_1 = A^{\nu_1}$ and $B_1 = \sum_{j=0}^{\nu_1-1} A^j B$. In (6c), the outputs are constrained to the tightened output constraint set $\mathcal{Y}_1$, as discussed in Section II-B. In (6d), the terminal state $x_1(k_{1,F}|k_1)$ is constrained to the terminal set $\mathcal{T}$ from (3). Finally, (6e) provides $\mathbf{C}_1$ the choice of initial condition, $x_1(k_1|k_1)$, as either the current state of the system, $x(k)$, or the optimal state at this time step determined by $\mathbf{C}_1$ at the previous time step, $x_1^*(k_1|k_1-1)$. This choice of initial condition is important to guaranteeing recursive feasibility, as discussed in Section II-B.

The lower-level controller $\mathbf{C}_2$ updates at every time step of the system, by solving the constrained optimization problem $\mathbf{P}_2\,(x(k))$ defined as

$$J_2^*\,(x(k)) = \min_{U_2(k_2)} \sum_{j=k_2}^{k_2+N_2(k_2)-1} \ell(j|k_2) + \ell_T(k_2+N_2(k_2)), \quad (7a)$$

$$\text{s.t.}\ \forall j \in [k_2, k_2+N_2(k_2)-1]$$

$$x_2(j+1|k_2) = Ax_2(j|k_2) + Bu_2(j|k_2), \quad (7b)$$

$$y_2(j|k_2) = Cx_2(j|k_2) + Du_2(j|k_2) \in \mathcal{Y}, \quad (7c)$$

$$x_2(k_2+N_2(k_2)|k_2) \in \mathcal{S}_2(k_2+N_2(k_2)), \quad (7d)$$

$$x_2(k_2|k_2) = x(k). \quad (7e)$$

Note that $\mathbf{C}_2$ has a shrinking and resetting horizon. The prediction horizon length is defined as $N_2(k_2) \triangleq N_2 - (k_2 \mod N_2)$ where N_2 satisfies **Assumption 3ii**. This allows $\mathbf{C}_2$ to predict between the current time step and the time step of the next update of $\mathbf{C}_1$, at which point $(k_2 \mod N_2) = 0$ and prediction horizon resets back to $N_2(k_2) = N_2$. The input sequence $U_2(k_2)$ is defined similarly to $U_1(k_1)$. In (7a), the stage cost is a function of states, inputs, and references such that $\ell(j|k_2) = \ell\,(x_2(j|k_2), u_2(j|k_2), r_2(j))$. The terminal cost $\ell_T(k_2+N_2(k_2))$ is described in Section IV-B and represents operational costs beyond the prediction horizon of $\mathbf{C}_2$ to improve the performance of the hierarchical controller. In (7b), the lower-level controller has an exact model of the system. In (7c), the outputs are constrained to the output constraint set $\mathcal{Y}$ from (4). In (7d), the time-varying terminal constraint corresponds to the waysets used to coordinate between controllers $\mathbf{C}_2$ and $\mathbf{C}_1$. Finally, (7e) defines the initial condition as the current state of the system.

Definition 1. [4] *The wayset $\mathcal{S}_2(k) \subseteq \mathcal{X}$ denotes a set of states at time step k such that for any $x(k) \in \mathcal{S}_2(k)$ there exists a future input trajectory $\{u(k)\}_{k=k}^{k_F-1}$ and corresponding state trajectory $\{x(k)\}_{k=k}^{k_F}$ satisfying (1), (2), and (3).*

The two-level hierarchical controller is implemented throughout system operation based on **Algorithm 1**.

B. Constraint Satisfaction

As discussed in detail in [4], the controller formulations (6) and (7) are specifically designed to guarantee recursive feasibility for $\mathbf{P}_1\,(x(k))$ and $\mathbf{P}_2\,(x(k))$ when implementing the hierarchical controller using **Algorithm 1**. Furthermore, recursive feasibility guarantees the satisfaction of output constraints (4) and terminal constraints (3). This constraint satisfaction guarantee relies on i) output constraint tightening, ii) wayset formulation, and iii) initial condition options. With additional details provided in [4], these three features are summarized as follows.

Algorithm 1: Two-level hierarchical MPC.

```
initialize k, k_1, k_2 ← 0
while k < k_F do
    if k mod ν_1 = 0 then
        calculate ℓ_p(k_1);
        solve P_1 (x(k));
        calculate ℓ_T(k_2 + N_2(k_2)), S_2(k_2 + N_2(k_2))
          and communicate to P_2 (x(k));
        k_1 ← k_1 + 1;
    end
    solve P_2 (x(k)) for U_2^*(k_2) and apply the first
      input in the sequence, u_2^*(k_2|k_2), to the system;
    k_2 ← k_2 + 1;
    k ← k + 1;
end
```

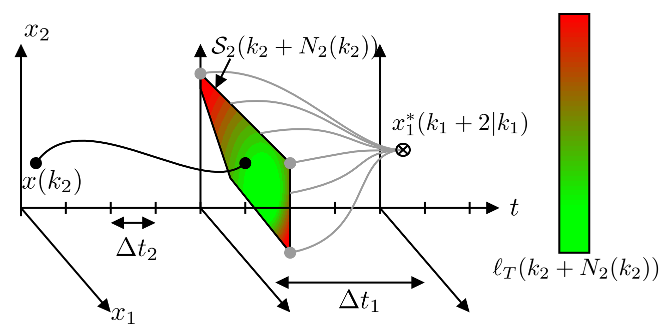

Fig. 1: Notional example of the combined use of waysets and terminal costs for coordination between controllers at different levels of the hierarchy.

While the lower-level controller $\mathbf{C}_2$ is formulated with the original output constraint set $\mathcal{Y}$, the upper-level controller $\mathbf{C}_1$ requires the tightened output constraint set $\mathcal{Y}_1 \subseteq \mathcal{Y}$. This constraint tightening prevents $\mathbf{C}_1$ from computing state trajectories that violate state constraints during the inter-sample time steps between the slow updates of $\mathbf{C}_1$. As a result, any optimal state and input trajectory determined by $\mathbf{C}_1$ is a feasible solution for $\mathbf{C}_2$.

As shown in Fig. 1, the wayset $\mathcal{S}_2(k_2+N_2(k_2))$, used as a terminal constraint in the formulation of $\mathbf{C}_2$, represents a backward reach set starting from the point $x_1^*(k_1+2|k_1)$ along the optimal state trajectory determined by $\mathbf{C}_1$. Thus, for any state within this wayset, there exists feasible state and input trajectories that steer the system back to the trajectory determined by $\mathbf{C}_1$. The wayset provides $\mathbf{C}_2$ the flexibility to deviate from the trajectories determined by $\mathbf{C}_1$. This flexibility allows $\mathbf{C}_2$ to further minimize short-term operational cost, by using its faster update rate, while guaranteeing long-term constraint satisfaction. **Algorithm 2** outlines the backward reach set computation used to define the wayset.

Algorithm 2: Wayset $\mathcal{S}_2(k_2 + N_2(k_2))$ computation at time step $k = \nu_1 k_1$.

```
initialize j ← N_2
if k_1 ≥ k_{1,F} − 1 then
    S_2(k_2 + N_2(k_2)) = T;
else
    S_2(j) = x_1^*(k_1 + 2|k_1);
    while j ≥ 1 do
        S̃_2(j − 1) = A^{-1} S_2(j) ⊕ (−A^{-1} B) U ;
        S_2(j − 1) = S̃_2(j − 1) ∩ X;
        j ← j − 1;
    end
    S_2(k_2 + N_2(k_2)) = S_2(j)
end
```

Since the wayset $\mathcal{S}_2(k_2 + N_2(k_2))$ is time-varying and depends on the state $x_1^*(k_1+2|k_1)$, each wayset needs to be computed on-line immediately following each update of $\mathbf{C}_1$. Constrained zonotopes [7] are used to provide the computational efficiency necessary to perform these set computations on-line. Assuming zonotopic input and state sets,

$$\mathcal{X} = \{G_x \xi_x + c_x \mid \|\xi_x\|_\infty \leq 1\},$$
$$\mathcal{U} = \{G_u \xi_u + c_u \mid \|\xi_u\|_\infty \leq 1\},$$

where $G_x \in \mathbb{R}^{n\times n}$, $G_u \in \mathbb{R}^{m\times m}$, $c_x \in \mathbb{R}^n$, and $c_u \in \mathbb{R}^m$. By applying **Algorithm 2** with Minkowski sum and intersection operations defined in [7], the wayset is a constrained zonotope such that

$$\mathcal{S}_2(k_2 + N_2(k_2)) = \{G\xi + c \mid \|\xi\|_\infty \leq 1, A\xi = b\}, \quad (8)$$

where $G \in \mathbb{R}^{n\times(n+m)N_2}$, $c \in \mathbb{R}^n$, $A \in \mathbb{R}^{nN_2\times(n+m)N_2}$ and $b \in \mathbb{R}^{nN_2}$.

Finally, the initial condition option (6e) is used to ensure recursive feasibility of $\mathbf{C}_1$. This initial condition option is similar to that used in robust MPC [9]. Due to the use of waysets, $\mathbf{C}_2$ may drive the system to a point in the wayset that is not a feasible initial condition for $\mathbf{C}_1$. Thus $\mathbf{C}_1$ is given the option to start its state trajectory from the current state of the system, if a feasible solution exists, or from the state along the optimal trajectory determined by $\mathbf{C}_1$ at the previous time step. While this first option in not guaranteed to be feasible, the second option always provides a feasible solution to both $\mathbf{C}_1$ and $\mathbf{C}_2$ [4].

Using output constraint tightening, waysets, and the initial condition option, the following theorem from [4] establishes guaranteed constraint satisfaction for the hierarchical controller.

Theorem 1. [4] *Following **Algorithm 1** for a two-level hierarchy, both the upper- and lower-level controller problems,* $\mathbf{P}_1(x(k))$ *and* $\mathbf{P}_2(x(k))$*, are feasible when solved at* $k \mod \nu_i = 0$*, resulting in system state and input trajectories satisfying constraints* (2) *and* (3).

III. Control Performance

With constraint satisfaction established, this section presents the main results of this paper by focusing on the control performance of the hierarchical controller.

In this paper, control performance is quantified by the cost function (5), which captures the sum of system operation costs at each discrete time step starting from $t = 0$ and ending at $t = t_F = k_F \Delta t$. With a time step of Δt and an initial prediction horizon of $N = k_F$, a centralized MPC approach produces the optimal (minimal) cost $J^*(x(0))$, as defined in (5). Since this paper assumes an exact model of the system (1) without disturbances and finite operation, the optimal solution determined by the centralized MPC controller any time $k \in [0, k_F - 1]$ is simply the tail of the optimal solution determined at time $k = 0$ [10].

With the optimal system operation cost denoted $J^*(x(0))$, any other control formulation results in a cost $J(x(0)) \geq J^*(x(0))$. First, consider the case where only the upper-level controller of the hierarchy is applied to the system. In this case, the inputs determined by $\mathbf{C}_1$ are applied directly to the system with a slow update period of $\Delta t_1 = N_2 \Delta t$. With an initial prediction horizon of $N_1 = k_{1,F}$, the minimal operational cost for this controller is denoted as $J^{up}(x(0))$, which is the sum of the $N_1 - 1$ coarse stage costs where

$$J^{up}(x(0)) = \sum_{k_1=0}^{N_1-1} J^{up}(k_1|k_1).$$

The double index notation $J^{up}(k_1|k_1)$ is used to denote the optimal operation cost at coarse time step k_1 determined by $\mathbf{C}_1$ at time k_1. As with the centralized MPC controller, if only the upper-level controller is applied directly to the system, then the optimal solution for $\mathbf{C}_1$ at any time step $k_1 \in [0, N_1-1]$ is the tail of the optimal solution determined at time $k = 0$. Therefore, $J^{up}(k_1|k_1) = J^{up}(k_1|0),\ \forall k_1$.

For a hierarchical controller with multiple levels, it is natural to expect the lower-level controllers to further reduce the total operation cost. Thus, denoting the minimal operating cost for the hierarchical controller as $J^h(x(0))$, it is expected that

$$J^*(x(0)) \leq J^h(x(0)) \leq J^{up}(x(0)). \quad (9)$$

However guaranteeing (9) requires effective coordination between the controller at different levels of the hierarchy.

An initial coordination strategy introduced in [10] utilized the notion of *waypoints*. A waypoint corresponds to a point along the optimal state trajectory determined by $\mathbf{C}_1$. This waypoint is treated as a terminal constraint in the formulation of the optimization problem for $\mathbf{C}_2$. Therefore, the lower-level controller has the flexibility to further improve system performance over the fast time steps between slow updates of the upper-level controller. Due to the waypoint constraint, operating costs for the waypoint-based hierarchy and the upper-level only controller can be directly compared at each coarse time step, where

$$J^h(k_1|k_1) \leq J^{up}(k_1|k_1),\ \forall k_1 \in [0, N_1 - 1].$$

Therefore, $J^h(x(0)) \leq J^{up}(x(0))$, which guarantees that the lower-level controller can only help to improve the control performance of the hierarchy.

An improved coordination strategy introduced in [4] expanded the idea of coordination via terminal constraints through the use of *waysets*. As shown in Fig. 1, a wayset represents a backward-reachable set from a point along the optimal state trajectory determined by $\mathbf{C}_1$. Waysets provide even greater flexibility to the lower-level controller while still guaranteeing controller feasibility and system constraint satisfaction. However, this additional flexibility introduces the potential for greedy behavior, where the $\mathbf{C}_2$ minimizes its own cost function over its short horizon while unknowingly increasing the long-term operational cost beyond its prediction horizon. This greedy behavior could lead to an increase in total operation cost where $J^h(x(0)) \geq J^{up}(x(0))$. In this case, the lower-level controller actually degrades the control performance. In Section V, a numerical example shows how a wayset-based hierarchy can greedily utilize a finite resource too quickly, leading to significant performance degradation during later system operation.

By imposing terminal costs, denoted as $\ell_T(k_2 + N_2(k_2))$ for the formulation of $\mathbf{C}_2$ in (7), the lower-level controller can only improve control performance, resulting in $J^h(x(0)) \leq J^{up}(x(0))$. As shown in Fig. 1, this terminal cost represents the constrained state transition cost from the terminal state $x_2(k_2 + N_2(k_2)|k_2)$ to the optimal state $x_1^*(k_1+2|k_1)$ determined by the upper-level controller. Note that this optimal state is exactly the state used to define the wayset in **Algorithm 1**. Therefore, while the wayset constraint (7d) guarantees that there is a feasible trajectory from $x_2(k_2 + N_2(k_2)|k_2)$ to $x_1^*(k_1 + 2|k_1)$, the terminal cost now represents the exact operational cost for this state transition. Using this terminal cost at coarse time step k_1, $\mathbf{C}_2$ now minimizes the cost function $J^h(k_1|k_1) + J^h(k_1+1|k_1)$, effectively doubling the prediction horizon of the lower-level controller.

While Section IV demonstrates how to compute the terminal cost $\ell_T(k_2 + N_2(k_2))$, the following theorem states that using terminal costs results in a lower-level controller that can only improve the performance of the system.

Theorem 2. *Following* ***Algorithm 1*** *for a two-level hierarchy with penalty cost $\ell_p(k_1)$ and terminal cost $\ell_T(k_2 + N_2(k_2))$ as computed in Section IV, the hierarchical controller results in a reduced operational cost compared to only applying the upper-level controller such that $J^h(x(0)) \leq J^{up}(x(0))$.*

Proof. See Appendix. □

IV. Terminal Cost Computation

A. Upper-level Controller

From (6a), the cost function for the upper-level controller consists of stage costs, a terminal cost, and a penalty cost. Since $\mathbf{C}_1$ predicts to the end of system operation, the terminal cost is the same as that used in (5). The penalty cost is used to ensure that the hierarchical controller provides improved control performance as stated in **Theorem 2**. Specifically,

$$\ell_p(k_1) = \begin{cases} 0 & \text{if } x_1(k_1|k_1) = x_1^*(k_1|k_1-1), \\ \Delta J(k_1-1|k_1-1) & \text{if } x_1(k_1|k_1) = x(k), \end{cases}$$

where $\Delta J(j|j) \triangleq \max(0, J^h(j|j) - J^{up}(j|j))$. Therefore, $\ell_p(k_1)$ penalizes $\mathbf{C}_1$ only when using the current state as its initial condition and this penalty is based on the difference between operational costs determined by $\mathbf{C}_2$ and $\mathbf{C}_1$. If $\mathbf{C}_2$ chose a higher operating cost at the previous coarse time step $k_1 - 1$, then $\Delta J(k_1 - 1|k_1 - 1) \geq 0$. This formulation of the penalty cost ensures that if $\mathbf{C}_1$ choses to start at $x(k)$ then the corresponding optimal trajectory has an operating cost at least $\Delta J(k_1 - 1|k_1 - 1)$ less than the optimal trajectory starting at $x_1^*(k_1|k_1 - 1)$. This property is used in the proof of **Theorem 2** in the Appendix.

B. Lower-level Controller

From (7a), the cost function for the lower-level controller consists of stage costs and a time-varying terminal cost. Traditionally, terminal costs in MPC are formulated as a function of the terminal state $x_2(k_2 + N_2(k_2)|k_2)$. However, the proposed terminal cost represents the state transition cost from $x_2(k_2 + N_2(k_2)|k_2)$ to $x_1^*(k_1 + 2|k_1)$ subject to linear output constraints (4). If the cost function in (5) is quadratic, the resulting terminal cost would be piecewise-quadratic, a known result from the field of explicit MPC [8]. However, this cost is dependent on $x_1^*(k_1 + 2|k_1)$ as well as references $r(j)$. Therefore, the terminal cost is time-varying and would be very difficult to parameterize with respect to both $x_1^*(k_1 + 2|k_1)$ and references $r(j)$. For these reasons, formulating the terminal cost as a function of the terminal state is impractical.

Alternatively, the terminal cost can be formulated in terms of ξ, the variables used to define the wayset as a constrained zonotope in (8). First note, if $k_2 + N_2(k_2) = k_F$, $\mathbf{C}_2$ predicts to the end of system operation and therefore $\ell_T(k_2 + N_2(k_2)) = \ell_T(k_F)$. Otherwise, for all $k_2 + N_2(k_2) < k_F$,

$$\ell_T(k_2 + N_2(k_2)) = \sum_{j=k_2+N_2(k_2)}^{k_2+2N_2(k_2)-1} \ell(j|k_2). \qquad (10)$$

If the stage costs are in the form of a weighted quadratic function, then let

$$\ell(j|k_2) = \|r(j) - z(j|k_2)\|_\Lambda^2, \qquad (11)$$

where $z(j|k_2) = Ex(j|k_2) + Fu(j|k_2)$.

Theorem 3. *The terminal cost* (10) *with the quadratic stage costs* (11) *is the constrained state transition cost from $x_2(k_2 + N_2(k_2)|k_2)$ to $x_1^*(k_1 + 2|k_1)$ and can be exactly represented as a quadratic function of ξ from* (8) *where*

$$\ell_T(k_2 + N_2(k_2)) = \xi^T P_T \xi + 2q_T \xi + r_T, \qquad (12)$$

and P_T is time-invariant while q_T and r_T are time-varying due to dependence on references $r(j)$ and state $x_1^*(k_1+2|k_1)$.

Proof. See Appendix. □

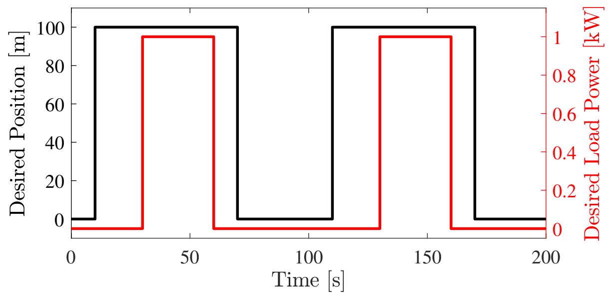

Fig. 2: References for position and load power.

V. Numerical Example

To demonstrate the benefits of using terminal costs for hierarchical MPC coordination, this section develops a two-level hierarchy with terminal costs for the simplified vehicle system from [4]. The system model is

$$x(k+1) = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} x(k) + \begin{bmatrix} 0 & 0 & 0 \\ 1 & -1 & 0 \\ -1 & -1 & -1 \end{bmatrix} u(k),$$

where the three states represent position, velocity, and on-board stored energy, the three inputs represent acceleration, deceleration, and power to an on-board load, all of which deplete the stored energy. The system and lower-level controller have time step sizes of $\Delta t = \Delta t_2 = 1$ second. Finite operation is defined for $k_F = 200$ seconds. Choosing $\Delta t_1 = 10$ seconds results in $\nu_1 = 10$ and prediction horizons of $N_1 = 20$ and $N_2 = 10$ steps.

The desired operation, defined by $\{r(k)\}_{k=0}^{k_F}$, is shown in Fig. 2 for the first state (position), and third input (load power). References for the first and second inputs (acceleration and deceleration) are 0 for the entire mission, and are not shown in Fig. 2. These references are used to define (5) as the weighted quadratic cost function from (11) where $z(j) = \begin{bmatrix} [1\ 0\ 0]\, x(j) \\ u(j) \end{bmatrix}$ and $\Lambda = \text{diag}\left([10^{-2}\ 10^{0}\ 10^{0}\ 10^{2}]\right)$.

Given an initial on-board stored energy, $E(0)$, the output and terminal constraints defining $\mathcal{Y}$ and $\mathcal{T}$ are

$$\begin{bmatrix} -1 \\ -20 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \leq y(k) \leq \begin{bmatrix} 105 \\ 20 \\ E(0) \\ 1 \\ 1 \\ 1 \end{bmatrix}, \quad \begin{bmatrix} -1 \\ -1 \\ 0 \end{bmatrix} \leq x(k_F) \leq \begin{bmatrix} 1 \\ 1 \\ E(0) \end{bmatrix}.$$

For two different initial conditions, $x(0) = [0\ 0\ 150]^T$ and $x(0) = [0\ 0\ 100]^T$, Fig. 3 shows simulation results using the proposed hierarchical MPC with terminal costs (Hier-T) compared to centralized MPC (Cent), hierarchical MPC without terminal costs (Hier-NT), and hierarchical MPC with only the upper-level controller (Hier-Up). The three subplots show the position reference tracking, the depletion of on-board energy, and the load power reference tracking.

First, note that all controllers satisfy the output and terminal constraints constraints. For the hierarchical controllers, this is achieved using constraint tightening and wayset terminal constraints as discussed in Section II-B. While Fig. 3 shows that the four controllers result in qualitatively similar performance, the quantitative differences in control performance are clearly shown in Fig. 4. For both initial conditions, the centralized controller provides the lowest cost (best performance), as expected. Guaranteed by **Theorem 2**, the hierarchy with terminal costs results in an operating cost less than the upper-level only controller. The performance of the wayset-based hierarchical controller without terminal costs is significantly different for the two different initial conditions. When viewed as a resource distribution problem, an initial on-board stored energy of $E(0) = 150$kJ represents a case where there is enough energy to operate the system as desired. In this case, there is enough energy to support the short-sighted, greedy behavior of a hierarchy without terminal costs. However, if the initial on-board stored energy is reduced to $E(0) = 100$kJ, there is insufficient energy to operate as desired and an intelligent controller must ration this resource throughout operation. Now, the greedy behavior of the lower-level controller results in an operating cost greater than if only the upper-level controller was used. Fig. 3 shows that the majority of this increase in operating costs comes from the inability to closely track the load power reference trajectory (input 3).

Using a desktop computer with a 3.2 Ghz i7 processor and 16 GB of RAM, all controllers were formulated and solved with YALMIP [11] and Gurobi [12]. While the addition of terminal costs does not increase the number of decision variables for wayset-based hierarchical MPC, the simulation results shows a modest increase in average computation time for the lower-level controller from $\Delta t_{calc} = 0.066$ seconds without terminal costs to $\Delta t_{calc} = 0.106$ seconds. Additionally, the use of constrained zonotopes allows the waysets and terminal costs to be computed very quickly with wayset and terminal cost calculations averaging 2 milliseconds and 0.4 milliseconds respectively.

This is likely due to how the time-varying q_T and r_T terms from (12) are implemented in YALMIP. For systems with long prediction horizons and a large number of states and inputs, the number of ξ variables required to represent the waysets and terminal costs might pose challenges to real-time control execution. Therefore, future work will explore the use of reduced-order inner-approximations of waysets and the corresponding approximations of terminal costs to provide greater scalability of the proposed approach.

VI. Conclusions

A two-level hierarchical MPC formulation was presented with coordination between controllers through the use of both terminal constraints and terminal costs. Wayset-based terminal constraints guaranteed constraint satisfaction while terminal costs guaranteed hierarchical control performance. The terminal costs were specifically formulated to balance both short- and long-term control performance without the need for controller tuning. As a result, the hierarchical controller was proven to provide better control performance compared to only applying the upper-level controller. A numerical example demonstrated the merits of including of terminal costs as a coordination mechanism for hierarchical MPC. Future work will focus on the efficient calculation of

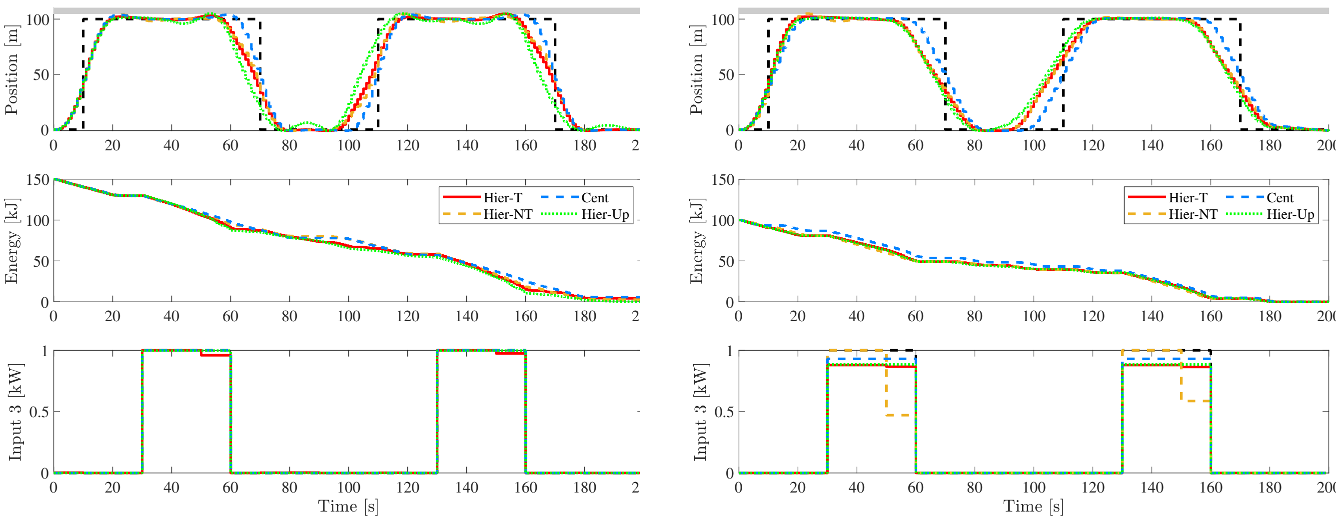

Fig. 3: Simulation results for initial conditions $x(0) = [0\ 0\ 150]^T$ (left) and $x(0) = [0\ 0\ 100]^T$ (right) comparing the proposed hierarchical MPC with terminal costs (Hier-T) to centralized MPC (Cent), hierarchical MPC without terminal costs (Hier-NT), and hierarchical MPC with only the upper-level controller (Hier-Up).

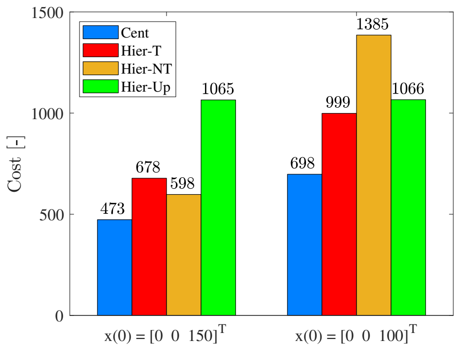

Fig. 4: Comparison of system operation cost for both initial conditions and each of the four controllers.

lower complexity inner-approximations of the waysets and corresponding terminal cost approximations for improved scalability.

APPENDIX

Proof of ***Theorem 2.*** Achieving the desired reduction in operational cost is equivalent to

$$\sum_{k_1=0}^{N_1-1} J^h(k_1|k_1) \leq \sum_{k_1=0}^{N_1-1} J^{up}(k_1|0), \tag{13}$$

where the left-hand side represent the operational cost of the two-level hierarchy as the sum of operational costs over every coarse time step.

As shown in Fig. 5a, at time step $k_1 = 0$, $\mathbf{C}_2$ plans a trajectory such that the total operational cost satisfies

$$J^h(0|0) + J^h(1|0) + \sum_{j=2}^{N_1-1} J^{up}(j|0) \leq \sum_{j=0}^{N_1-1} J^{up}(j|0), \tag{14}$$

where $J^h(0|0)$ and $J^h(1|0)$ are the stage and terminal costs for $\mathbf{C}_2$. Note that (14) holds since the trajectory determined by $\mathbf{C}_1$ is always a feasible trajectory for $\mathbf{C}_2$.

At each time step $k_1 \in [1, N_1 - 2]$, the upper-level controller has a choice of initial condition from (6e) such that $x_1(k_1|k_1) = x(k) \vee x_1^*(k_1|k_1-1)$. As shown in Fig. 5b, if $\mathbf{C}_1$ chooses $x_1(k_1|k_1) = x_1^*(k_1|k_1-1)$, then there exists a feasible trajectory for $\mathbf{C}_2$ such that

$$J^h(k_1|k_1) + J^h(k_1+1|k_1) + \sum_{j=k_1+2}^{N_1-1} J^{up}(j|k_1) \leq J^h(k_1|k_1-1) + \sum_{j=k_1+1}^{N_1-1} J^{up}(j|k_1-1). \tag{15}$$

Due to the penalty cost imposed on $\mathbf{C}_1$ from Section IV-A, if $\mathbf{C}_1$ chooses $x_1(k_1|k_1) = x(k)$, then there exists a feasible trajectory for $\mathbf{C}_2$, as shown in Fig. 5c, such that

$$J^h(k_1|k_1) + J^h(k_1+1|k_1) + \sum_{j=k_1+2}^{N_1-1} J^{up}(j|k_1) \leq \sum_{j=k_1}^{N_1-1} J^{up}(j|k_1-1) - \Delta J(k_1-1|k_1-1). \tag{16}$$

At the time step $k_1 = N_1 - 1$, there exists a feasible trajectory $\mathbf{C}_2$, as shown in Fig. 5d, such that

$$J^h(N_1-1|N_1-1) \leq J^h(N_1-1|N_1-2). \tag{17}$$

From the definition of $\Delta J(j|j)$ and combination of (14)-(16), it can be shown that there exists feasible trajectories at each time step $k_1 \in [1, N_1 - 2]$ such that

$$\sum_{j=0}^{k_1} J^h(j|j) + J^h(k_1+1|k_1) + \sum_{j=k_1+2}^{N_1-1} J^{up}(j|k_1) \leq \sum_{j=0}^{N_1-1} J^{up}(j|0). \tag{18}$$

Adding (18) for $k_1 = N_1 - 2$ and (17) results in the desired operational cost relationship from (13), proving the theorem.

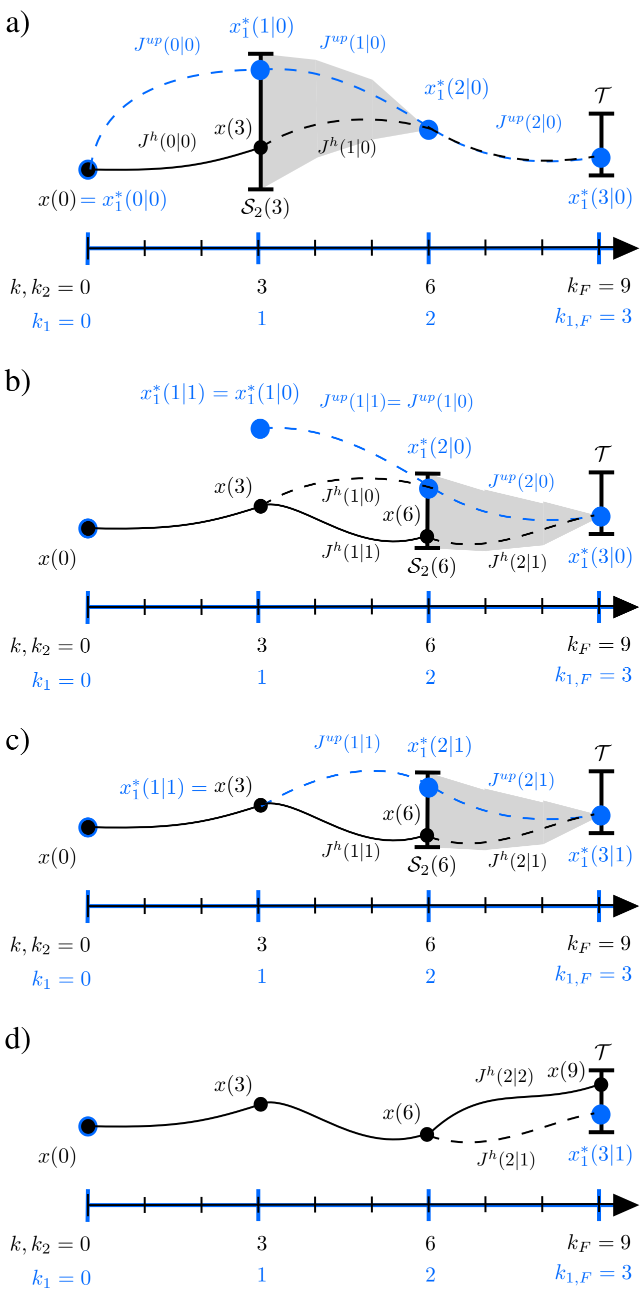

Fig. 5: Notional state trajectories used to demonstrate the operating cost relationships among feasible trajectories.

Proof of ***Theorem 3.*** Let R, Z, and U denote trajectories such that, for $k = k_2 + N_2$,

$$\begin{aligned} R &= [r^T(k+N_2-1) \quad r^T(k+N_2-2) \quad \dots \quad r^T(k)]^T, \\ Z &= [z^T(k+N_2-1) \quad z^T(k+N_2-2) \quad \dots \quad z^T(k)]^T, \\ U &= [u^T(k+N_2-1) \quad u^T(k+N_2-2) \quad \dots \quad u^T(k)]^T. \end{aligned}$$

Given (11), (10) can now be re-written as

$$\ell_T(k_2 + N_2(k_2)) = (R - Z)^T \hat{\Lambda} (R - Z),$$

where $\hat{\Lambda}$ is block diagonal with Λ repeated N_2 times. There exist matrices $\hat{A}, \hat{B}$ where $Z = \hat{A}x_1^*(k_1 + 2|k_1) + \hat{B}U$. Furthermore, with ξ from (8), $U = \hat{c}_u + \hat{G}_u \hat{T} \xi$ where

$$\hat{c}_u = \begin{bmatrix} c_u \\ c_u \\ \vdots \\ c_u \end{bmatrix}, \hat{G}_u = \begin{bmatrix} G_u & 0 & \dots & 0 \\ 0 & G_u & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & G_u \end{bmatrix},$$

and

$$\hat{T} = \begin{bmatrix} [I\ 0] & 0 & \dots & 0 \\ 0 & [I\ 0] & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & [I\ 0] \end{bmatrix}.$$

Now, as in (12), the terminal cost is quadratic in ξ where

$$\begin{aligned} P_T &= (\hat{B}\hat{G}_u\hat{T})^T \hat{\Lambda} (\hat{B}\hat{G}_u\hat{T}), \\ q_T &= (\hat{A}x_1^*(k_1+2|k_1) + \hat{B}\hat{c}_u - R)^T \hat{\Lambda} (\hat{B}\hat{G}_u\hat{T}), \\ r_t &= \|\hat{A}x_1^*(k_1+2|k_1) + \hat{B}\hat{c}_u - R\|^2_{\hat{\Lambda}}. \end{aligned}$$

Note that both q_T and r_T are time-varying due to their dependence on references R and state $x_1^*(k_1 + 2|k_1)$ while P_T is time-invariant.

References

[1] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. M. Scokaert. Constrained model predictive control: Stability and optimality. *Automatica*, 36:789–814, 2000.

[2] D.Q. Mayne, M.M. Seron, and S.V. Raković. Robust Model Predictive Control of Constrained Linear Systems with Bounded Disturbances. *Automatica*, 41:219–224, 2005.

[3] Riccardo Scattolini. Architectures for distributed and hierarchical Model Predictive Control - A review. *Journal of Process Control*, 19:723–731, 2009.

[4] Justin P. Koeln and Brandon M. Hencey. Constrained Hierarchical MPC via Zonotopic Waysets. *American Control Conference*, pages 4237–4244, 2019.

[5] Justin Koeln, Vignesh Raghuraman, and Brandon Hencey. Vertical hierarchical MPC for constrained linear systems. *Automatica*, 2020.

[6] Rohan C. Shekhar, Michael Kearney, and Iman Shames. Robust Model Predictive Control of Unmanned Aerial Vehicles Using Waysets. *Journal of Guidance, Control, and Dynamics*, 38, 2015.

[7] Joseph K. Scott, Davide M. Raimondo, Giuseppe Roberto Marseglia, and Richard D. Braatz. Constrained zonotopes: A new tool for set-based estimation and fault detection. *Automatica*, 69:126–136, 2016.

[8] Alberto Bemporad, Manfred Morari, Vivek Dua, and Efstratios N. Pistikopoulos. The explicit linear quadratic regulator for constrained systems. *Automatica*, 2002.

[9] Lukas Hewing and Melanie N. Zeilinger. Stochastic Model Predictive Control for Linear Systems using Probabilistic Reachable Sets. *IEEE Conference on Decision and Control*, pages 5182–5188, 2018.

[10] Justin P. Koeln and Andrew G. Alleyne. Two-Level Hierarchical Mission-Based Model Predictive Control. *American Control Conference*, pages 2332–2337, 2018.

[11] Johan Lofberg. YALMIP: a toolbox for modeling and optimization in MATLAB. *2004 IEEE International Conference on Computer Aided Control Systems Design*, 2004.

[12] Gurobi Optimization Inc. Gurobi Optimizer Reference Manual, 2016.