Vertical hierarchical MPC for constrained linear systems [⋆]

J. P. Koeln [a], V. Raghuraman [a], B. M. Hencey [b]

[a] *University of Texas at Dallas, Richardson, TX, 75080, United States*

[b] *Air Force Research Laboratory, Wright-Patterson Air Force Base, OH, 45433, United States*

Abstract

A hierarchical Model Predictive Control (MPC) formulation is presented for discrete-time linear systems with state and input constraints. A vertical hierarchical controller, with one controller per level, reduces the computational burden associated with the solution of centralized MPC having long prediction horizons and short time steps. To guarantee satisfaction of state and input constraints in the presence of both known and unknown disturbances, a robust MPC formulation is used at each level while waysets are used as a novel coordination mechanism between controllers at different levels. These waysets are implemented as terminal state constraints on lower-level controllers and are computed on-line based on the optimal state trajectory of upper-level controllers. To achieve the computational efficiency necessary for on-line calculation, waysets are represented as constrained zonotopes. State and input constraint satisfaction is proven for a hierarchical controller with an arbitrary number of levels and two numerical examples demonstrate the key features, performance, and scalability of the approach.

Key words: hierarchical control, robust model predictive control, set-based computing, zonotopes

1 Introduction

For the control of many complex systems, the ability to satisfy both input and state constraints is critical to maintaining safe and reliable system operation. Additionally, with increasing demand for performance and efficiency, optimal system operation is characterized by both transient and steady-state input and state trajectories that approach these constraints. Examples include the control of aircraft power systems [1, 2], on- and off-road hybrid vehicles [3–5], smart grids [6–8], and water distribution networks [9, 10].

For the control of input and state constrained systems, system operation is not always indefinite and the desired behavior is not always characterized by driving the system to steady-state or from one equlibrium to another. This idea is discussed in [11] for vehicle maneuvering problems where the notion of stability is replaced by the notion of completion. Similarly, this work focuses on the control of systems under finite operation, with the goal of guaranteeing state and input constraints during operation and terminal state constraints at the end of operation.

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. Feasibility of these constraints and stability of the closed-loop system are well understood for the case of a single centralized controller [12]. However, for systems that require fast control update rates and long prediction horizons, the time required to solve the resulting large optimization problem may prevent real-time implementation.

Alternatively, hierarchical MPC can be used to decompose control decisions across multiple levels of controllers [13]. Upper-level controllers use large time steps to achieve long prediction horizons with fewer discrete steps. Lower-level controllers with small time steps use short prediction horizons to minimize computational cost and enable real-time implementation. To handle the timescale separation between the system and actuator dynamics, several two-level hierarchical MPC formulations have been developed [14–20].

However, existing hierarchical formulations are not well suited to maximize the performance of a system subject

[⋆] Research supported by the United States Air Force Research Laboratory and cleared for public release on 3/27/2019, case number 88ABW-2019-1324. This paper was not presented at any IFAC meeting. Corresponding author J. P. Koeln Tel. +1 972-883-4649. Fax +1 972-883-4659.

Email addresses: justin.koeln@utdallas.edu (J. P. Koeln), vignesh.raghuraman@utdallas.edu (V. Raghuraman), brandon.hencey@us.af.mil (B. M. Hencey).

to input, state, and terminal constraints under finite operation. Most hierarchical MPC approaches are formulated with the goal of stabilizing the system to an equilibrium in the interior of state and input constraint sets. However, for systems with finite operation, such equilibrium might not exist as in the case of systems whose operation is based on the consumption of a finite resource (e.g. fuel in an aircraft [1, 2] or battery state of charge in an electric vehicle [3–5]). Moreover, existing approaches [14–20] are typically formulated where upper-level controllers are robust to the control decisions of lower-level controllers and overall control authority is divided among each control level. Existing reference tracking based coordination mechanisms require lower-level controllers to track state and input trajectories determined by upper-level controllers, preventing the hierarchical controller from utilizing the fast system dynamics to maximize system performance. Finally, while most hierarchical MPC formulations are designed to two controller levels, many systems have more than two timescales and an M-level hierarchical MPC would be more effective in controlling each timescale. To date, there does not exist a M-level hierarchical MPC framework that provides guaranteed state and input constraint satisfaction, even for linear systems.

To develop a constructive hierarchical MPC framework that guarantees input and state constraint satisfaction, this paper focuses on a vertical hierarchy, with one controller per level, for discrete-time linear systems. This work replaces the conventional reference tracking based coordination between controllers of the hierarchy with a novel coordination mechanism using *waysets*. A wayset defines 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. Within the proposed hierarchical MPC framework, waysets are computed based on optimal state trajectories determined by upper-level controllers and imposed as terminal constraints for lower-level controllers. Wayset-based coordination overcomes the limitations of existing hierarchical MPC frameworks by removing the need for constraint-feasible equlibrium, removing the conservatism that stems from requiring upper-level controllers to be robust to the lower-level control decisions, and allowing lower-level controllers to utilize the fast system dynamics to further improve system performance.

Similar coordination mechanisms have been used in the literature. For wastewater treatment systems, the hierarchical controller in [21] uses "interlayer targets" to achieve coordination between controllers at different levels. These interlayer targets inspired the use of *waypoints* as the coordination mechanism for a two-level hierarchy in [22]. These waypoints are imposed as terminal constraints on lower-level controllers within the hierarchy. For vehicle path-planning, waypoint tracking control is used in [23] to split long planning horizons into multiple shorter horizons by creating intermediate goals. This idea was extended to wayset tracking in [24], where waysets represent a region of the state space instead of a single point. However, in both cases the waypoint/wayset generation is performed off-line in a feed-forward fashion.

To enable on-line calculation of waysets, computational efficiency is vastly improved by representing the waysets as constrained zonotopes [25]. Zonotopes are widely used due to their computational efficiency in reach set calculations for hybrid system verification, estimation, and MPC [25–28]. As will be shown, the proposed wayset calculations are similar to the computation of reach sets and utilize linear transformation, Minkowski sum, and intersection operations.

To achieve guaranteed input and state constraint satisfaction, this paper develops a vertical hierarchical MPC framework with a novel wayset coordination mechanism. The specific contributions of this paper are 1) the development of an M-level hierarchical MPC framework that incorporates known disturbances and is robust to bounded unknown disturbances, 2) the definition and use of waysets to prove robust closed-loop constraint satisfaction, 3) the representation and calculation of waysets as constrained zonotopes to achieve efficient on-line calculation, and 4) the numerical demonstration of performance and scalability of the hierarchical approach. Note that the nominal version of this hierarchical MPC formulation without accounting for disturbances was initially presented in [29].

The remainder of the paper is organized as follows. Sections 2 and 3 present the class of constrained discrete-time linear systems and the proposed M-level hierarchical MPC formulation. Section 4 defines a robust output constraint tightening procedure and the wayset properties. Robust state and input constraint satisfaction is proved in Section 5. Section 6 details the calculation of waysets and the use of constrained zonotopes to achieve computational efficiency. Two numerical examples are provided in Section 7 to demonstrate the key features, performance, and scalability of the approach. Finally, Section 8 summarizes the conclusions of the paper.

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. The subscript i is used to denote the i^{th} controller in the hierarchy, $\mathbf{C}_i$, and $i^- = i - 1$ is shorthand used to reference the controller directly above, $\mathbf{C}_{i^-}$. 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 $\mathcal{Z} \cap \mathcal{W}$. The Pontryagin difference is defined as $\mathcal{Z} \ominus \mathcal{W} = \{z \in \mathbb{R}^n \mid z + w \in \mathcal{Z}, \forall w \in \mathcal{W}\}$. The Cartesian product is defined as $\mathcal{Z} \times \mathcal{Y} = \left\{[z^T \ y^T]^T \mid z \in \mathcal{Z}, y \in \mathcal{Y}\right\}$. The projection of the set $\mathcal{Y}$ on the first n dimensions is denoted $\pi_n(\mathcal{Y})$. The empty set is denoted as $\emptyset$.

2 Problem Formulation

Consider the discrete linear time-invariant system

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

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

Assumption 1 *With a fixed time step size 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 discrete time steps indexed by $k \in [0, k_F]$.*

Assumption 2 *The disturbance is $d(k) = \hat{d}(k) + \Delta d(k)$, where $\hat{d}(k)$ is known a priori for all $k \in [0, k_F]$ and $\Delta d(k)$ is unknown but bounded to a convex and compact set such that $\Delta d(k) \in \mathcal{D} \subset \mathbb{R}^n$.*

Starting from an initial condition $x(0)$, the goal is to develop a vertical hierarchical MPC approach that plans and executes an input trajectory $\{u(k)\}_{k=0}^{k_F-1}$ and corresponding state trajectory $\{x(k)\}_{k=0}^{k_F}$ which i) satisfies the system dynamics from (1); ii) satisfies the state and input constraints

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

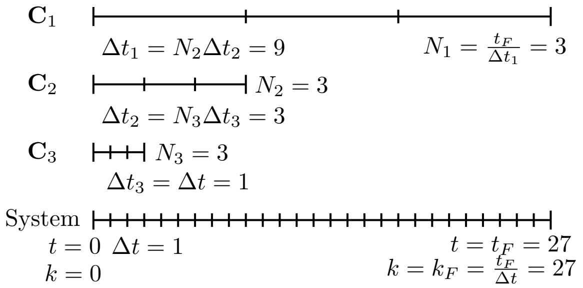
iii) satisfies the terminal constraint

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

and iv) minimizes the generic cost function

$$J^*(x(0)) = \min_{\{u(k)\}_{k=0}^{k_F-1}} \sum_{j=0}^{k_F} \ell\left(x(j), u(j), r(j)\right), \tag{4}$$

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



$\mathbf{C}_1$
$\Delta t_1 = N_2 \Delta t_2 = 9$
$N_1 = \frac{t_F}{\Delta t_1} = 3$
$\mathbf{C}_2$
$N_2 = 3$
$\Delta t_2 = N_3 \Delta t_3 = 3$
$\mathbf{C}_3$
$N_3 = 3$
$\Delta t_3 = \Delta t = 1$
System
$t = 0$ $\Delta t = 1$
$k = 0$
$t = t_F = 27$
$k = k_F = \frac{t_F}{\Delta t} = 27$

Fig. 1. The relationship between prediction horizons N_i and time step sizes $\Delta t_i, i \in [1, 3]$, for a three-level, $M = 3$, hierarchical controller operating a system for $t \in [0, 27]$.

Assumption 3 *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

$$y(k) \triangleq \begin{bmatrix} x(k) \\ u(k) \end{bmatrix} = Cx(k) + Du(k) \in \mathcal{Y} \triangleq \mathcal{X} \times \mathcal{U}. \tag{5}$$

3 Robust Vertical Hierarchical Control

The proposed hierarchical control formulation has M levels of controllers $\mathbf{C}_i, i \in [1, M]$, each with a prediction horizon and time step size that satisfy the following assumptions.

Assumption 4 *For each controller $\mathbf{C}_i, i \in [1, M]$, the prediction horizon $N_i \in \mathbb{Z}_+$ and time step size $\Delta t_i > 0$ satisfy*

i) $\Delta t_M = \Delta t$;
ii) $\Delta t_{i^-} = N_i \Delta t_i$;
iii) $\Delta t_1 = \frac{t_F}{N_1}$.

These assumptions indicate i) the lowest-level controller $\mathbf{C}_M$ and the system (1) have the same time step size, ii) each controller $\mathbf{C}_i$ predicts state and input trajectories between consecutive updates of the controller $\mathbf{C}_{i-}$ directly above in the hierarchical controller, and iii) the highest-level controller $\mathbf{C}_1$ predicts to the end of system operation. Additionally, let $\nu_i \triangleq \frac{\Delta t_i}{\Delta t} \in \mathbb{Z}_+$, be defined as a time scaling factor for each controller. The time steps for $\mathbf{C}_i$ are indexed by k_i, where $k_i \triangleq \frac{k}{\nu_i}$ and $k_M = k$. Let $k_{i,F} \triangleq \frac{k_F}{\nu_i}$ such that $k_i \in [0, k_{i,F}]$. Fig. 1 shows how the conditions of **Assumption 4** determine the relationships between time step sizes and prediction horizons for a three-level hierarchical controller.

Each controller $\mathbf{C}_i$ updates only when $k = \nu_i k_i$ (i.e. when $k \mod \nu_i = 0$), by solving the constrained optimization problem $\mathbf{P}_i(x(k))$ defined as

$$J_i^*(x(k)) = \min_{\substack{\hat{x}_i(k_i|k_i)\\ \hat{U}_i(k_i)}} \sum_{j=k_i}^{k_i+N_i(k_i)} \ell\left(\hat{x}_i(j|k_i), \hat{u}_i(j|k_i), r_i(j)\right), \quad \text{(6a)}$$

$$\text{s.t.}\, \forall j \in [k_i, k_i + N_i(k_i)]$$

$$\hat{x}_i(j+1|k_i) = A_i\hat{x}_i(j|k_i) + B_i\hat{u}_i(j|k_i) + \hat{d}_i(j), \quad \text{(6b)}$$

$$\hat{y}_i(j|k_i) = C\hat{x}_i(j|k_i) + D\hat{u}_i(j|k_i) \in \hat{\mathcal{Y}}_i(j), \quad \text{(6c)}$$

$$\hat{x}_i(k_i + N_i(k_i)|k_i) \in \hat{\mathcal{S}}_i(k_i + N_i(k_i)), \quad \text{(6d)}$$

$$x(k) - \hat{x}_i(k_i|k_i) \in \mathcal{E} \,\vee\, \hat{x}_i(k_i|k_i) = \hat{x}_i^*(k_i|k_i - 1). \quad \text{(6e)}$$

First, note that $\mathbf{P}_i(x(k))$ has a *shrinking horizon*, based on the summation limits in (6a), with horizon length $N_i(k_i) \triangleq N_i - (k_i \mod N_i)$ where N_i satisfies **Assumption 4ii**. Thus, $\mathbf{C}_i$ predicts between the current time step and the time step of the next update of $\mathbf{C}_{i^-}$, at which point $(k_i \mod N_i) = 0$ and prediction horizon resets back to $N_i(k_i) = N_i$. The nominal input sequence over this horizon is defined as $\hat{U}_i(k_i) = \{\hat{u}_i(j|k_i)\}_{j=k_i}^{k_i+N_i(k_i)-1}$, with the optimal nominal sequence denoted as $\hat{U}_i^*(k_i)$. In (6b), the model used by $\mathbf{C}_i$ assumes a piecewise constant nominal control input over the time step size Δt_i and thus $A_i = A^{\nu_i}$ and $B_i = \sum_{j=0}^{\nu_i-1} A^j B$ (as in [17]). Since the known disturbance is time varying over the time step size Δt_i, the known disturbance used in (6b) is

$$\hat{d}_i(k_i) = \sum_{j=0}^{\nu_i-1} A^{\nu_i-1-j}\hat{d}(\nu_i k_i + j), \quad (7)$$

which ensures that $\hat{d}_i(k_i)$ captures the accumulated effect of the known time-varying disturbances, $\hat{d}(k), k \in [\nu_i k_i, \nu_i k_i + \nu_i - 1]$, during this slow time step. The nominal outputs $\hat{y}_i(j|k_i)$ in (6c) are constrained to the time-varying tightened output constraint set $\hat{\mathcal{Y}}_i(j)$, with details provided in Sections 4.1 and 4.2. The time-varying terminal state constraint in (6d) corresponds to the waysets $\hat{\mathcal{S}}_i(k_i + N_i(k_i))$ used as the only coordination mechanism between controllers $\mathbf{C}_i$ and $\mathbf{C}_{i^-}$. The properties of these waysets are defined in Section 4.3 and the calculation of the waysets is provided in Section 6.2. Finally, (6e) provides $\mathbf{C}_i$ the choice of nominal initial condition, $\hat{x}_i(k_i|k_i)$, which is a decision variable following the tube-based MPC formulation in [30]. The reasoning for this specific treatment of the initial condition is detailed in Section 4.4.

As shown in Fig. 2, coordination is achieved among the controllers through the use of waysets imposed as terminal constraints (6d). Within this hierarchical control framework, only the lowest level controller $\mathbf{C}_M$ directly affects the system. Once $\mathbf{C}_M$ has solved for the optimal nominal control input trajectory $\hat{U}_M^*(k_M)$ and optimal nominal initial condition $\hat{x}_M^*(k_M|k_M)$, the input to the system $u(k)$ is calculated based on the control law

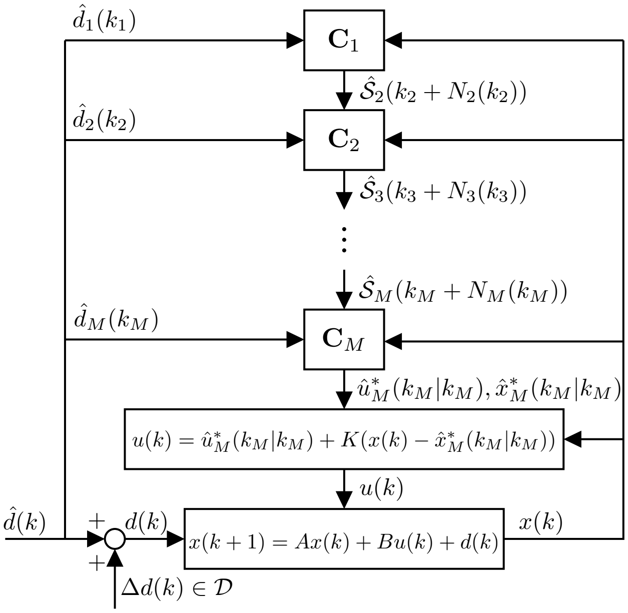

$\hat{d}_1(k_1)$
$\mathbf{C}_1$
$\hat{\mathcal{S}}_2(k_2 + N_2(k_2))$
$\hat{d}_2(k_2)$
$\mathbf{C}_2$
$\hat{\mathcal{S}}_3(k_3 + N_3(k_3))$
$\hat{\mathcal{S}}_M(k_M + N_M(k_M))$
$\hat{d}_M(k_M)$
$\mathbf{C}_M$
$\hat{u}_M^*(k_M|k_M), \hat{x}_M^*(k_M|k_M)$
$u(k) = \hat{u}_M^*(k_M|k_M) + K(x(k) - \hat{x}_M^*(k_M|k_M))$
$u(k)$
$\hat{d}(k)$
$+$
$d(k)$
$+$
$x(k+1) = Ax(k) + Bu(k) + d(k)$
$x(k)$
$\Delta d(k) \in \mathcal{D}$

Fig. 2. The hierarchical MPC structure with M levels, where controllers $\mathbf{C}_i, i \in [1, M]$ are formulated based on (6), the known disturbances $\hat{d}_i(k_i)$ are computed based on (7), the waysets $\hat{\mathcal{S}}_i(k_i + N_i(k_i)), i \in [2, M]$, are used to coordinate controllers $\mathbf{C}_i$ and $\mathbf{C}_{i^-}$, and the static feedback control law (8) bounds the effect of the unknown disturbance $\Delta d(k) \in \mathcal{D}$.

$$u(k) = \hat{u}_M^*(k_M|k_M) + K[x(k) - \hat{x}_M^*(k_M|k_M)], \quad (8)$$

where $K \in \mathbb{R}^{m \times n}$ is a static feedback control gain. This control law is used to bound the error between the nominal and true system state trajectories created by the unknown disturbances $\Delta d(k)$. Section 4.1 details the design of K and the resulting set $\mathcal{E}$ that bounds this error.

In summary, the M-level hierarchical controller is implemented based on **Algorithm 1**. The constrained optimization problem $\mathbf{P}_i(x(k))$ for each controller $\mathbf{C}_i, i \in [1, M]$ is specifically designed with the nominal model of (6b), the time-varying tightened output constraints of (6c), the wayset terminal constraints of (6d), and the initial state condition of (6e) to establish guaranteed robust satisfaction of output and terminal constraints as proven in Section 5.

Remark 1 *As discussed in [29], neither the references nor the exact formulation of cost function in* (6a) *affect the feasibility of any* $\mathbf{P}_i(x(k))$ *in the hierarchical controller.*

Remark 2 *While the focus of this paper is on hierarchical control for systems with finite operation per Assumption 1, indefinite system operation can be achieved by replacing the shrinking prediction horizon,* N_1, *of* C_1 *with a receding horizon of fixed length. To guarantee recursive*

Algorithm 1: M-level hierarchical MPC

```
initialize $k, k_i \leftarrow 0,\ \forall i \in [1, M]$
while $k < k_F$ do
    for $i = 1$ to $M-1$ do
        if $k \mod \nu_i = 0$ then
            solve $\mathbf{P}_i\,(x(k))$;
            calculate $\hat{\mathcal{S}}_i(k_i + N_i(k_i))$ and
              communicate to $\mathbf{P}_{i+1}\,(x(k))$;
            $k_i \leftarrow k_i + 1$;
        end
        solve $\mathbf{P}_M\,(x(k))$ and apply the input
          $u(k)$ to the system based on (8);
        $k_M \leftarrow k_M + 1$;
        $k \leftarrow k + 1$;
    end
end
```

feasibility, the terminal set $\mathcal{T}$ must be a Robust Positive Invariant (RPI) set based on pre-determined bounds of $d(k)$, as done in centralized MPC formulations [31].

4 Set Definitions

4.1 Output Constraint Tightening for $\mathbf{C}_M$

The proposed hierarchical MPC framework is robust to unknown bounded disturbances using the tube-based MPC formulation developed in [30]. For the lowest level controller $\mathbf{C}_M$, the only difference between the true system (1) and the model used for control (6b) is the unknown bounded disturbance $\Delta d(k) \in \mathcal{D}$. Using the control law (8), and comparing (1) and (6b) for $i = M$, the error $e(k) = x(k) - \hat{x}^*_M(k_M|k_M)$ satisfies

$$e(k+1) = (A + BK)e(k) + \Delta d(k). \tag{9}$$

Assuming K is designed to stabilize $A+BK$ and $\mathcal{E} \in \mathbb{R}^n$ is a disturbance invariant set for (9), then

$$(A + BK)\mathcal{E} \oplus \mathcal{D} \subseteq \mathcal{E}. \tag{10}$$

Thus, if $e(k) \in \mathcal{E}$, then $e(k+1) \in \mathcal{E}$ for all $\Delta d(k) \in \mathcal{D}$. As in [30], the constraint (6e) on the initial condition allows $\mathbf{C}_M$ to choose $\hat{x}_M(k_M|k_M)$ such that $e(k) \in \mathcal{E}$. The minimal disturbance invariant set [32] is

$$\mathcal{E} = \bigoplus_{i=0}^{\infty}(A + BK)^i\mathcal{D}, \tag{11}$$

and should be as small as possible to reduce conservatism of the controller. However, due to the infinite sum, computing $\mathcal{E}$ is difficult and a outer approximation $\tilde{\mathcal{E}}$ is typically used where $\mathcal{E} \subseteq \tilde{\mathcal{E}}$, $\tilde{\mathcal{E}}$ satisfies (10), and $\tilde{\mathcal{E}}$ can be represented as a polytope [31]. For the remainder of the paper $\mathcal{E}$ and $\tilde{\mathcal{E}}$ are used interchangeably.

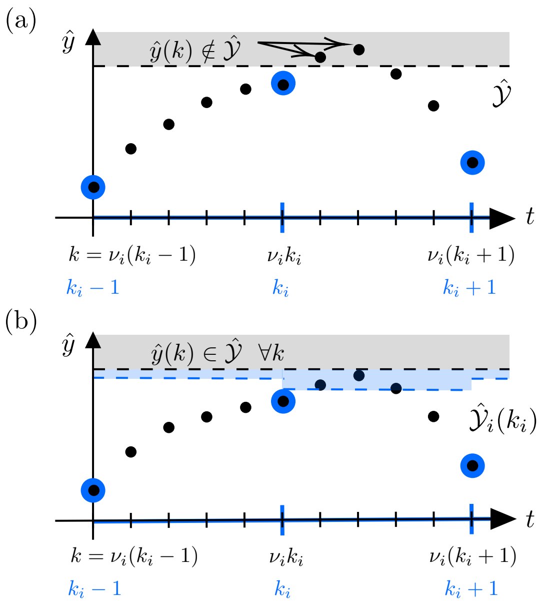


Fig. 3. (a) Controller $\mathbf{C}_i$ plans a feasible state trajectory (large blue dots) at the slow time step k_i but the resulting trajectory (small black dots) violates output constraints (dashed black line) at the inter-sample system time steps. (b) Tightening the output constraint set (dashed blue line) for $\mathbf{C}_i$ ensures that any trajectory at the slow time index is also feasible at the faster time indices.

From [30], when designing a robust MPC controller for $\mathbf{C}_M$ with the nominal system model (6b), tightened state, input, and terminal constraint sets are used where

$$\hat{\mathcal{X}} \triangleq \mathcal{X} \ominus \mathcal{E}, \quad \hat{\mathcal{U}} \triangleq \mathcal{U} \ominus K\mathcal{E}, \quad \hat{\mathcal{T}} \triangleq \mathcal{T} \ominus \mathcal{E}. \tag{12}$$

It is assumed that $\mathcal{D}$, and thus $\mathcal{E}$, are small relative to the state and input constraint sets such that $\hat{\mathcal{X}}$,$\hat{\mathcal{U}}$,$\hat{\mathcal{T}} \neq \emptyset$. Based on (5), the tightened output constraint set is

$$\hat{\mathcal{Y}} \triangleq \mathcal{Y} \ominus (\mathcal{E} \times K\mathcal{E}). \tag{13}$$

Since $\mathbf{C}_M$ has the same time step size as the system, i.e. $\nu_M = 1$, $\hat{\mathcal{Y}}_M(k_M) = \hat{\mathcal{Y}}$ for all k_M. However, for $\mathbf{C}_i, i \in [1, M-1]$, additional time-varying constraint tightening is required to account for inter-sample behavior between the slow updates of these upper-level controllers.

4.2 Output Constraint Tightening for $\mathbf{C}_i, i \in [1, M-1]$

As shown in Fig. 3(a), constraining the slow nominal output trajectory $\hat{y}_i(k_i) \in \hat{\mathcal{Y}}$ planned by $\mathbf{C}_i, i \in [1, M-1]$ does not guarantee that $\hat{y}(k) \in \hat{\mathcal{Y}}$ during the inter-sample updates when $k = [\nu_i k_i + 1, \nu_i(k_i + 1) - 1]$. Thus, it is important to further tighten the constraint set $\hat{\mathcal{Y}}$ to account for transient state trajectories between the slow updates, as shown in Fig. 3(b).

Definition 1 *The time-varying tightened nominal output constraint set $\hat{\mathcal{Y}}_i(k_i)$ is the set of all initial nominal state and nominal input combinations that results in an output trajectory satisfying* (13) *if the input is held constant for ν_i steps, i.e.*

$$\hat{\mathcal{Y}}_i(k_i) \triangleq \Big\{\hat{y}_i(k_i) = \left[\begin{smallmatrix}\hat{x}_i(k_i)\\ \hat{u}_i(k_i)\end{smallmatrix}\right] \Big| \, \forall k \in [\nu_i k_i, \nu_i(k_i+1)-1]\,,$$
$$\hat{y}(k) = \left[\begin{smallmatrix}\hat{x}(k)\\ \hat{u}(k)\end{smallmatrix}\right] \in \hat{\mathcal{Y}},\ \hat{u}(k) = \hat{u}_i(k_i),$$
$$\hat{x}(k+1) = A\hat{x}(k) + B\hat{u}(k) + \hat{d}(k),$$
$$\hat{x}(\nu_i k_i) = \hat{x}_i(k_i)\Big\}.$$

Similar to the procedure presented in [29], $\hat{\mathcal{Y}}_i(k_i)$ is calculated based on the tightened output constraint set $\hat{\mathcal{Y}}$ represented in H-Rep as

$$\hat{\mathcal{Y}} = \Big\{(\hat{x}(k), \hat{u}(k)) \Big| \hat{P}\,[C\ D] \left[\begin{smallmatrix}\hat{x}(k)\\ \hat{u}(k)\end{smallmatrix}\right] \leq \hat{q}\Big\}. \tag{14}$$

For $i \in [1, M-1]$, $\hat{\mathcal{Y}}_i(k_i)$ is time-varying due to the dependence on $\hat{d}_i(k_i)$ and may be computed as

$$\hat{\mathcal{Y}}_i(k_i) = \hat{\mathcal{Y}} \cap \hat{\mathcal{Y}}(\nu_i k_i + 1) \cap \cdots \cap \hat{\mathcal{Y}}(\nu_i k_i + \nu_i - 1), \tag{15}$$

where, for all $j \in [1, \nu_i - 1]$,

$$\hat{\mathcal{Y}}(\nu_i k_i + j) = \Big\{(\hat{x}(k), \hat{u}(k)) \Big| \hat{P}\,[C_j\ D_j] \left[\begin{smallmatrix}\hat{x}(k)\\ \hat{u}(k)\end{smallmatrix}\right] \leq \hat{q}(\nu_i k_i + j)\Big\}, \tag{16}$$

with $C_j = CA^j$, $D_j = D + C\sum_{l=0}^{j-1} A^l B$, and

$$\hat{q}(\nu_i k_i + j) = \hat{q} - \hat{P}C\sum_{l=0}^{j-1} A^{j-1-l}\hat{d}(\nu_i k_i + l). \tag{17}$$

Lemma 1 *For all $i < j$, $i, j \in [1, M]$, at time step $k = \nu_i k_i = \nu_j k_j$, $\hat{\mathcal{Y}}_i(k_i) \subseteq \hat{\mathcal{Y}}_j(k_j) \subseteq \hat{\mathcal{Y}}$.*

PROOF. See [29]. □

4.3 Waysets

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

In this paper, the waysets imposed as terminal constraints in (6d) are the sole coordination mechanism between levels of the hierarchical controller. Since (6d) imposes a constraint on the nominal predicted state, nominal waysets are used and denoted as $\hat{\mathcal{S}}$. Since the waysets are used to guarantee feasibility of constraints beyond the prediction horizon of lower-level controllers, each controller $\mathbf{C}_i$ has a different wayset denoted as $\hat{\mathcal{S}}_i$. Finally, the waysets are time-varying, denoted as $\hat{\mathcal{S}}_i(k_i + N_i(k_i))$, where the time step always corresponds to the time step of the next update of $\mathbf{C}_{i^-}$. Within the context of **Definition 2**, there are many possible ways to formulate $\hat{\mathcal{S}}_i(k_i + N_i(k_i))$. In this paper, waysets are formulated to satisfy the following assumptions in order to prove feasibility of the M-level hierarchical controller.

Assumption 5 *The waysets $\hat{\mathcal{S}}_i(k_i+N_i(k_i))$ in* (6d) *satisfy the following:*

(1) for $\mathbf{C}_1$, $\hat{\mathcal{S}}_1(k_1 + N_1(k_1)) = \hat{\mathcal{T}}$,
(2) for $\mathbf{C}_i$, $i \in [2, M]$,
  - *(a) $\hat{\mathcal{S}}_i(k_i+N_i(k_i))$ is only recomputed at updates of $\mathbf{C}_{i^-}$ per **Algorithm 1** ,*
  - *(b) $\hat{x}^*_{i^-}(k_{i^-}+1|k_{i^-}) \in \hat{\mathcal{S}}_i(k_i + N_i(k_i))$,*
  - *(c) if $\hat{x}^*_{i^-}(k_{i^-}+2|k_{i^-})$ exists, for each state in $\hat{\mathcal{S}}_i(k_i + N_i(k_i))$ there exists a trajectory satisfying* (6b) *and* (6c) *that drives the system to $\hat{x}^*_{i^-}(k_{i^-}+2|k_{i^-})$,*
  - *(d) if $\hat{x}^*_{i^-}(k_{i^-}+2|k_{i^-})$ does not exist, $\hat{\mathcal{S}}_i(k_i + N_i(k_i)) = \hat{\mathcal{S}}_{i^-}(k_{i^-} + N_{i^-}(k_{i^-}))$,*
  - *(e) if $k_i + N_i(k_i) = k_{i,F}$, $\hat{\mathcal{S}}_i(k_i + N_i(k_i)) = \hat{\mathcal{T}}$.*

Conceptually, these assumptions state: 1) for the highest level controller, the wayset equals the terminal constraint set, since $\mathbf{C}_1$ always predicts to the final time step per **Assumption 4iii**; 2a) since the wayset for $\mathbf{C}_i$ depends on the state trajectory determined by $\mathbf{C}_{i^-}$, waysets for $\mathbf{C}_i$ are only recomputed when $\mathbf{C}_{i^-}$ updates; 2b) noting that the time index of the wayset for $\mathbf{C}_i$ corresponds to the time index for the second optimal nominal state in the trajectory determined by $\mathbf{C}_{i^-}$, this optimal state exists in the wayset; 2c) since the prediction horizon for each controller shrinks over time, if the third optimal nominal state in the trajectory determined by $\mathbf{C}_{i^-}$ exists, then the wayset for $\mathbf{C}_i$ is defined as all the nominal states such that there exists feasible nominal input and state trajectories that drive the nominal system to this state; 2d) if the third optimal nominal state in the trajectory determined by $\mathbf{C}_{i^-}$ does not exist, then the wayset for $\mathbf{C}_i$ is set equal to the wayset for $\mathbf{C}_{i^-}$; and 2e) if $\mathbf{C}_i$ predicts to the final time step, the wayset is set equal to the tightened terminal constraint set.

4.4 Initial Conditions

In the formulation of $\mathbf{P}_i\,(x(k))$, (6e) provides $\mathbf{C}_i$ with two options for the choice of nominal initial condition, $\hat{x}_i(k_i|k_i)$. The first option, $x(k) - \hat{x}_i(k_i|k_i) \in \mathcal{E}$, comes from the tube-based MPC formulation presented in [30]. The second option, $\hat{x}_i(k_i|k_i) = \hat{x}^*_i(k_i|k_i-1)$, similar to [33] allows the nominal initial condition to equal the optimal nominal state for this time step determined by $\mathbf{C}_i$ at the previous time step. To understand the role

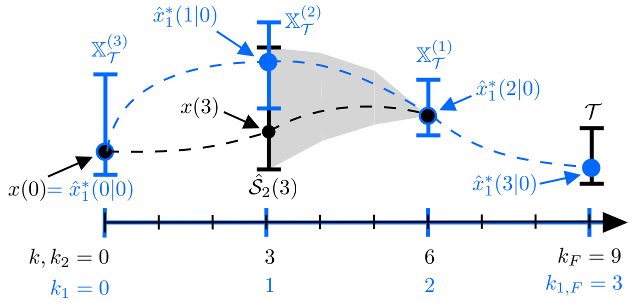


Fig. 4. Schematic showing the need for the initial condition option in (6e) where $x(3)$ satisfies the wayset constraint $x(3) \in \hat{\mathcal{S}}_2(3)$ but $x(3) \notin \mathbb{X}_{\mathcal{T}}^{(2)}$.

of this initial condition option, consider the following definition and assumption.

Definition 3 [34] *The feasible set $\mathbb{X}_{\mathcal{T}}^{(N_1)} \subset \mathcal{X}$ denotes the set of states such that $\mathbf{P}_1(x(k))$ admits a solution, with prediction horizon N_1 and terminal set $\mathcal{T}$.*

Assumption 6 *The initial condition satisfies $x(0) \in \mathbb{X}_{\mathcal{T}}^{(N_1)}$ and thus there exists a feasible solution to $\mathbf{P}_1(x(0))$ at time step $k = k_1 = 0$.*

For clarity of exposition, Fig. 4 demonstrates the need for the initial condition option in (6e) for the case where $M = 2$ and $\Delta d(k) = 0$. With a prediction horizon of $N_1 = 3$, Fig. 4 shows how $x(0) \in \mathbb{X}_{\mathcal{T}}^{(3)}$ by **Assumption 6**. Therefore, $\mathbf{C}_1$ has a feasible state trajectory $\{\hat{x}_1^*(j|0)\}_{j=0}^{3}$. The feasibility of this trajectory implies $\hat{x}_1^*(1|0) \in \mathbb{X}_{\mathcal{T}}^{(2)}$ and $\hat{x}_1^*(2|0) \in \mathbb{X}_{\mathcal{T}}^{(1)}$. Per **Assumption 5**, $\hat{\mathcal{S}}_2(3)$ denotes the set of states such that there exists a trajectory satisfying (6b) and (6c) which drives the system to $\hat{x}_1^*(2|0)$. As proven in the following section, the lower-level controller(s) will drive the system from $x(0)$ to $x(3) \in \hat{\mathcal{S}}_2(3)$, however there is no guarantee that $x(3) \in \mathbb{X}_{\mathcal{T}}^{(2)}$. If $x(3) \notin \mathbb{X}_{\mathcal{T}}^{(2)}$, then the second initial condition option in (6e) is required to maintain feasibility of $\mathbf{P}_1(x(3))$. The following section details how the initial condition option and properties of the waysets defined in **Assumption 5** establish feasibility of all controllers.

5 Hierarchical Control Feasibility

The following lemmas establish feasibility of individual controllers within the hierarchy starting with the highest-level controller $\mathbf{C}_1$.

Lemma 2 *If $\mathbf{P}_1(x(k))$ is feasible at $k = \nu_1 k_1$, then $\mathbf{P}_1(x(k))$ is feasible at $k = \nu_1(k_1 + 1)$.*

PROOF. As discussed in [29], the optimal solution at $k = \nu_1(k_1 + 1)$ is the tail of trajectories determined at the previous time step $k = \nu_1 k_1$. To show that this candidate solution satisfies the constraints in (6), first note that the nominal system model is time-invariant and that while $\hat{d}_1(k_1)$ is time-varying, the trajectory of $\hat{d}_1(k_1)$ is known per **Assumption 2** and does not change during system operation. Thus, the candidate solution satisfies (6b). Similarly, while $\hat{\mathcal{Y}}_1(k_1)$ varies with k_1, $\hat{\mathcal{Y}}_1(k_1)$ at a particular k_1 remains constant since the time step dependency only comes from the dependence of $\hat{\mathcal{Y}}_1(k_1)$ on $\hat{d}(k)$ per (15)-(17). Thus, the candidate solution satisfies (6c). Since $\hat{\mathcal{S}}_1(k_1 + N_1(k_1)) = \hat{\mathcal{T}}$ is time-invariant, the candidate solution satisfies (6d). Finally, (6e) provides the option to let $\hat{x}_1(k_1|k_1) = \hat{x}_1^*(k_1|k_1 - 1)$. Note, this candidate solution at $k = \nu_1(k_1 + 1)$ is optimal if $x(k) - \hat{x}_1(k_1 + 1|k_1 + 1) \in \mathcal{E}$ does not admit a feasible solution. □

Next, if $\mathbf{C}_i$ is feasible at the time step of the upper-level controller $\mathbf{C}_{i^-}$ update, then $\mathbf{C}_i$ remains feasible for all time steps until the next update of $\mathbf{C}_{i^-}$.

Lemma 3 *If $\mathbf{P}_i(x(k))$ is feasible at $k = \nu_i k_i$, where k mod $\nu_{i^-} = 0$ (i.e at the time of a $\mathbf{C}_{i^-}$ update), then $\mathbf{P}_i(x(k))$ is feasible at each time step $k = \nu_i(k_i + 1)$ through $k = \nu_i(k_i + N_i - 1)$.*

PROOF. See [29]. □

Finally, at the time step of the upper-level controller $\mathbf{C}_{i^-}$ update, feasibility of $\mathbf{C}_{i^-}$ guarantees feasibility of $\mathbf{C}_i$.

Lemma 4 *If $\mathbf{P}_{i^-}(x(k))$ has a feasible solution at $k = \nu_{i^-} k_{i^-}$ and $\mathbf{P}_M(x(k-1))$ had a feasible solution at the previous time step $k - 1$, then $\mathbf{P}_i(x(k))$ has a feasible solution at this time step.*

PROOF. The proof for the robust case presented in this paper is similar to the nominal case presented in [29] where two cases must be considered based on

1) $x(k) - \hat{x}_{i^-}^*(k_{i^-}|k_{i^-}) \in \mathcal{E}$ or,
2) $x_{i^-}^*(k_{i^-}|k_{i^-}) = x_{i^-}^*(k_{i^-}|k_{i^-} - 1)$.

For 1), the feasible solutions to $\mathbf{P}_i(x(k))$, comes directly from the feasible solutions determined by $\mathbf{P}_{i^-}(x(k))$ as shown in [29].

For 2), a feasible solution to $\mathbf{P}_i(x(k))$ exists with a nominal state trajectory satisfying $x(k) - \hat{x}_i(k_i|k_i) \in \mathcal{E}$. If $\mathbf{P}_M(x(k-1))$ had a feasible solution at the previous time step $k-1$, then $x(k-1) - \hat{x}_M^*(k_M - 1|k_M - 1) \in \mathcal{E}$ due to (6e) and $x(k) - \hat{x}_M^*(k_M|k_M - 1) \in \mathcal{E}$ due to the invariance of $\mathcal{E}$ under control law (8).Thus, $\hat{x}_M^*$ is a feasible

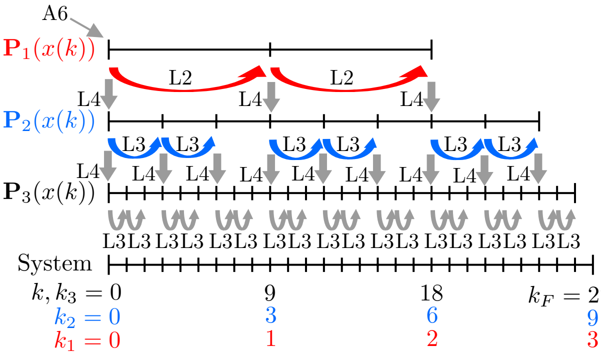


Fig. 5. Schematic showing how **Assumption 6** and **Lemmas 2-4** are used to establish feasibility of each controller at every time step for a three-level hierarchical controller.

initial condition for $\mathbf{P}_i\,(x(k))$ that satisfies constraints (6b)-(6e) as shown in [29]. □

The main results of this paper guarantees constraint satisfaction for the M-level hierarchical controller.

Theorem 1 *Following* ***Algorithm 1*** *for an M-level hierarchical controller, all control problems $\mathbf{P}_i\,(x(k))\,,\ i \in [1, M]$ are feasible when solved at $k \mod \nu_i = 0$, resulting in system state and input trajectories satisfying constraints* (2) *and* (3)*.*

PROOF. Using **Assumption 6** and **Lemmas 2-4**, Fig. 5 shows how feasibility is established for each $\mathbf{C}_i$, $i \in [1, M]$. Due to the constraint tightening presented in Section 4.1, feasibility of $\mathbf{P}_M\,(x(k))$ and the use of control law (8) guarantees the satisfaction of (2). By the wayset properties defined in **Assumption 5.2e**, $\hat{\mathcal{S}}_i(k_i + N_i(k_i)) = \hat{\mathcal{T}}$ once $k_i + N_i(k_i) = k_{i,F}$ and thus feasibility of $\mathbf{P}_M\,(x(k))$ also guarantees the satisfaction of (3). □

Remark 3 *With all states and inputs constrained in* (2)*, the constraint satisfaction established in Theorem 1 also provides bounded input bounded output (BIBO) stability. For many applications, BIBO stability is preferred over asymptotic stability so that the dynamics of the system can be used to maximize performance as in the case of completion-based MPC [11].*

6 Set Computations

6.1 Tightened Output Constraints

The tightened output constraints are calculated as in (15) and can be computed off-line prior to system operation. Thus, computational cost is not a primary concern. However, redundant constraints should be removed to minimize the number of constraints, n_h.

6.2 Waysets

Inspired by the iterative approach to feasible set calculation in [34], **Algorithm 2** generates waysets that satisfy **Assumption 5**. First, the if statement handles the case when $x^*_{i^-}(k_{i^-} + 2|k_{i^-})$ does not exist, due to the shrinking horizon, by setting the wayset for $\mathbf{C}_i$ equal to wayset for $\mathbf{C}_{i^-}$, satisfying **Assumption 5.2d**. If $x^*_{i^-}(k_{i^-} + 2|k_{i^-})$ does exist, then the iterative approach computes the wayset by starting at this optimal nominal state $x^*_{i^-}(k_{i^-} + 2|k_{i^-})$ at time step $k_i + 2N_i$ and working backwards to compute sets of states and inputs at previous time steps that drive the nominal system to this optimal state. The iterations end once the wayset is calculated for the time step $k_i + N_i$. The intersection operation is used to ensure that these state and input trajectories also satisfy the tightened output constraints from (6c). Finally, since waysets define a set of states, the projection operation is used to project the calculated output wayset into the first n dimensions.

Algorithm 2: Wayset $\hat{\mathcal{S}}_i(k_i + N_i(k_i))$ computation for $i \in [2, M]$ at time step $k = \nu_{i^-} k_{i^-}$.

```
1  initialize $j \leftarrow N_i$
2  if $N_{i^-}(k_{i^-}) < 2$ then
3      $\hat{\mathcal{S}}_i(k_i + N_i(k_i)) = \hat{\mathcal{S}}_{i^-}(k_{i^-} + N_{i^-}(k_{i^-}))$;
4  else
5      $\hat{\mathcal{S}}_i(j) = \hat{x}^*_{i^-}(k_{i^-} + 2|k_{i^-})$;
6      while $j \geq 1$ do
7          $\tilde{\mathcal{Y}}_i^{\hat{\mathcal{S}}_i(j)} = \Big\{(\hat{x}_i, \hat{u}_i) \mid \hat{x}_i^+ \in \hat{\mathcal{S}}_i(j),$
             $\hat{x}_i^+ = A_i\hat{x}_i + B_i\hat{u}_i + \hat{d}_i(k_i + N_i + j - 1)\Big\}$;
8          $\hat{\mathcal{S}}_i(j-1) = \pi_n(\tilde{\mathcal{Y}}_i^{\hat{\mathcal{S}}_i(j)} \cap_I \hat{\mathcal{Y}}_i(k_i + N_i + j - 1))$;
9          $j \leftarrow j - 1$;
10     end
11     $\hat{\mathcal{S}}_i(k_i + N_i(k_i)) = \hat{\mathcal{S}}_i(j)$
12 end
```

While the steps in **Algorithm 2** conceptually define the wayset calculations, these steps can be simplified using the notion of generalized intersection. Note that line 7 is equivalent to

$$[A_i\ B_i]\,\hat{y}_i \in \hat{\mathcal{S}}_i(j) \ominus \hat{d}_i(k_i + N_i + j - 1), \tag{18}$$

where the Pontryagin difference simply shifts the center of $\hat{\mathcal{S}}_i(j)$ since $\hat{d}_i(k_i + N_i + j - 1)$ is a vector and not a set. This condition and the tightened output constraint condition from (6c), $\hat{y}_i \in \hat{\mathcal{Y}}_i(k_i + N_i + j - 1))$ must be satisfied. The generalized intersection can be used to enforce both conditions as

$$\hat{y}_i \in \hat{\mathcal{Y}}_i(k_i + N_i + j - 1))\cap_{[A_i\ B_i]} \Big(\hat{\mathcal{S}}_i(j) \ominus \hat{d}_i(k_i + N_i + j - 1)\Big)\,. \tag{19}$$

Projection can then be used to transform this output constraint set to a state constraint set.

As discussed in [34], the iterative approach provides improved computational efficiency compared to projection-based methods. However, when polytopic constraint sets and waysets are represented in H-Rep or as a convex hull of vertices (V-Rep), on-line wayset calculation may still be limited by computational cost. As discussed in [25] and the references therein, the worst-case complexity of linear transformation, Minkowski sum, and generalized intersection scales exponentially in the set dimension. Therefore, **Algorithm 2** is likely to be very computationally expensive and potentially numerically unstable for n, m greater than about five (potentially less than 5 if the number of halfspaces or vertices defining the polytopes is large). In fact, the numerical examples in [34] are restricted to $n \leq 4$, $m \leq 2$, and less than ten iteration steps.

Under **Algorithm 1 1**, $\hat{\mathcal{S}}_i(k_i + N_i(k_i))$ is recomputed on-line at every update of $\mathbf{C}_{i^-}$. Thus, efficient set computations are critical to the wayset-based hierarchical control. The following section demonstrates how zonotopes can be used to significantly reduce the cost of computing waysets, enabling the proposed approach.

6.3 Zonotope-based Set Calculation

A *zonotope* is the Minkowski sum of a finite set of line segments or, equivalently, the image of a hypercube under an affine transformation [35, 36] . Using the generator-representation (G-Rep), the zonotope $\mathcal{Z} \subset \mathbb{R}^n$ is defined by its center $c \in \mathbb{R}^n$ and n_g generators g_i that form the columns of $G \in \mathbb{R}^{n \times n_g}$, such that $\mathcal{Z} = \{G\xi + c \mid \|\xi\|_\infty \leq 1\}$. The complexity of a zonotope is captured by its order, $o = \frac{n_g}{n}$.

Zonotopes have been widely used due to their computational efficiency in reach set calculations for hybrid system verification, estimation, and MPC [25–28]. As with the iterative algorithm in [34], computing these reach sets utilizes linear transformation and Minkowski sum operations. Zonotopes are closed under these operations (i.e. the Minkowski sum of two zonotopes is a zonotope) and the number of generators grows linearly with the number of Minkowski sum operations, compared to the potential exponential growth with H-Rep. Unfortunately, zonotopes in general are not closed under intersection and the conversion from G-Rep to H-Rep for intersection operations is inefficient.

Constrained zonotopes were developed in [25] to overcome this limitation. Using the constrained generator-representation (CG-Rep), the constrained zonotope has n_c equality constraints on ξ such that $\mathcal{Z} = \{G\xi + c \mid \|\xi\|_\infty \leq 1, A\xi = b\}$, where $A \in \mathbb{R}^{n_c \times n_g}$ and $b \in \mathbb{R}^{n_c}$. The complexity of a constrained zonotope is captured by the degrees-of-freedom order, $o_d = \frac{n_g - n_c}{n}$.

Using the shorthand $\mathcal{Z} = \{G, c, A, b\}$, [25] shows that constrained zonotopes are closed under linear transformation, Minkowski sum, and generalized intersection where

$$R\mathcal{Z} = \{RG_z, Rc_z, A_z, b_z\}, \tag{20}$$

$$\mathcal{Z} \oplus \mathcal{W} = \left\{ [G_z \; G_w], c_z + c_w, \begin{bmatrix} A_z & 0 \\ 0 & A_w \end{bmatrix}, \begin{bmatrix} b_z \\ b_w \end{bmatrix} \right\}, \tag{21}$$

$$\mathcal{Z} \cap_R \mathcal{Y} = \left\{ [G_z \; 0], c_z, \begin{bmatrix} A_z & 0 \\ 0 & A_y \\ RG_z & -G_y \end{bmatrix}, \begin{bmatrix} b_z \\ b_y \\ c_y - Rc_z \end{bmatrix} \right\}. \tag{22}$$

To perform the wayset calculations using zonotopes, it is necessary to convert sets from H-Rep to CG-Rep. For a set defined via box constraints such as $\mathcal{Z} = \{z \in \mathbb{R}^n \mid \underline{z} \leq z \leq \bar{z}\}$, the corresponding CG-Rep is

$$\mathcal{Z} = \left\{ diag\left(\frac{\bar{z} - \underline{z}}{2}\right), \frac{\bar{z} + \underline{z}}{2}, [\,], [\,] \right\}. \tag{23}$$

More generally, any convex polytope defined in H-Rep can be represented in CG-Rep using the following procedure from [25]. If $\mathcal{Z} = \left\{z \in \mathbb{R}^n \mid Hz \leq \bar{f}\right\}$ is a convex polytope, there exists a bounding zonotope $\mathcal{Z}_0 = \{G, c\} \subset \mathbb{R}^n$ such that $\mathcal{Z} \subset \mathcal{Z}_0$. Additionally, there exists $\underline{f} \in \mathbb{R}^n$ such that $Hz \in \left[\underline{f}, \bar{f}\right]$ for all $z \in \mathcal{Z}$. Note that the set $\mathcal{F} = \left[\underline{f}, \bar{f}\right]$ is defined via box constraints and can be written in CG-Rep based on (23). Thus, $\mathcal{Z}$ is defined as the generalized intersection of two zonotopes $\mathcal{Z} = \mathcal{Z}_0 \cap_H \mathcal{F}$ with CG-Rep from (22).

Note the following details when executing **Algorithm 2** using constrained zonotopes. Line 5 initializes the wayset as a point from the optimal nominal state trajectory determined by $\mathbf{C}_{i^-}$ as $\hat{\mathcal{S}}_i(j) = \left\{[\,], \hat{x}^*_{i^-}(k_{i^-} + 2|k_{i^-}), [\,], [\,]\right\}$. As discussed in Section 6.2, Line 7 and the intersection operation in Line 8 are expressed as the generalized intersection in (19). The projection operation in Line 8 is equivalent to a linear transformation in CG-Rep where R from (20) is $R = [I_n \; 0_{n \times m}]$.

Based on **Algorithm 2**, the number of generators n_g and number of constraints n_c required to represent $\hat{\mathcal{S}}_i(k_i + N_i(k_i))$ grows linearly with the prediction horizon N_i. Assuming n states and that the CG-Rep of $\hat{\mathcal{Y}}_i(k_i + N_i + j - 1))$ has $n_{g,\hat{\mathcal{Y}}_i}$ generators and $n_{c,\hat{\mathcal{Y}}_i}$ constraints, the generalized intersection in (19) adds $n_{g,\hat{\mathcal{Y}}_i}$ generators and $n + n_{c,\hat{\mathcal{Y}}_i}$ constraints to the CG-Rep of $\hat{\mathcal{S}}_i(j)$. Thus, for a prediction horizon of N_i, the wayset $\hat{\mathcal{S}}_i(k_i + N_i(k_i))$ has $n_{g,\hat{\mathcal{Y}}_i} N_i$ generators and $(n + n_{c,\hat{\mathcal{Y}}_i})N_i$ constraints.

As shown in the following numerical examples, CG-Rep reduces the time required to compute waysets by several orders-of-magnitude compared to H-Rep.

7 Numerical Examples

To demonstrate the formulation and use of waysets in hierarchical MPC, this section presents two numerical examples. The first example is the same vehicle system from [22, 29] and highlights the wayset and tightened output constraint set calculations, the resulting robust constraint satisfaction, and the overall closed-loop performance and computational cost of the two-level hierarchy compared to centralized MPC. The second example presents a three-level hierarchy for a linearized thermal system to demonstrate the scalability of the approach. All results were generated using MATLAB on a desktop computer with a 3.6 GHz i7 processor and 16 GB of RAM and all MPC optimization problems were formulated and solved with YALMIP [37] and Gurobi [38].

7.1 Vehicle Example

Consider the simplified vehicle system model

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

where the states $x(k) \in \mathbb{R}^3$ represent position, velocity, and on-board stored energy, and the inputs $u(k) \in \mathbb{R}^3$ represent acceleration, deceleration, and power to an on-board load, all of which deplete the stored energy. Per **Assumption 2**, the disturbances $d(k) \in \mathbb{R}^3$ with $d(k) = \hat{d}(k) + \Delta d(k)$.

The system and lowest level controller have time step sizes of $\Delta t = \Delta t_M = 1$ second. Finite operation is defined for 100 seconds, thus $k_F = 100$. Choosing $\Delta t_1 = 10$ seconds results in $\nu_1 = 10$ and maximum prediction horizons of $N_1 = N_2 = 10$ steps. The output constraints defining $\mathcal{Y}$ and $\mathcal{T}$ are

$$\begin{bmatrix} -1 \\ -20 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \leq y(k) \leq \begin{bmatrix} 105 \\ 20 \\ 100 \\ 1 \\ 1 \\ 1 \end{bmatrix}, \quad \begin{bmatrix} -1 \\ -1 \\ 0 \end{bmatrix} \leq x(k_F) \leq \begin{bmatrix} 1 \\ 1 \\ 100 \end{bmatrix}.$$

Given an initial state of $x(0) = [0\ 0\ 100]^T$, the desired operation, defined by $\{r(k)\}_{k=0}^{k_F}$, is shown in Fig. 6 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 operation, and thus are not shown in Fig. 6. These references are used to define (4) as the weighted quadratic cost function

$$\ell\left(x(j), u(j), r(j)\right) = \|r(j) - y_r(j)\|_\Lambda^2, \qquad (24)$$

where $y_r(j) = \begin{bmatrix} [1\ 0\ 0]x(j) \\ u(j) \end{bmatrix}$, $\Lambda = \text{diag}\left([10^2\ 10^0\ 10^0\ 10^2]\right)$.

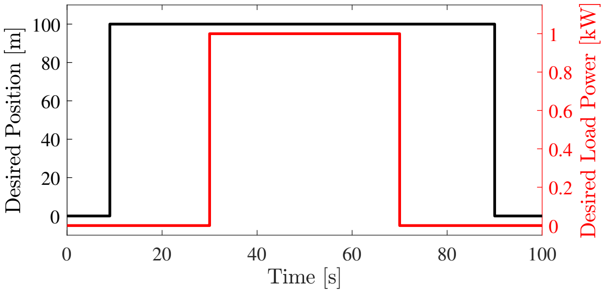


Fig. 6. References for state and load power.

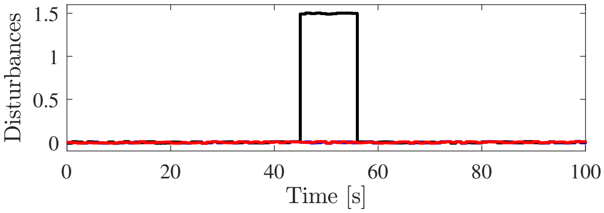


Fig. 7. Disturbance profile consisting of a large known pulse and small unknown deviations.

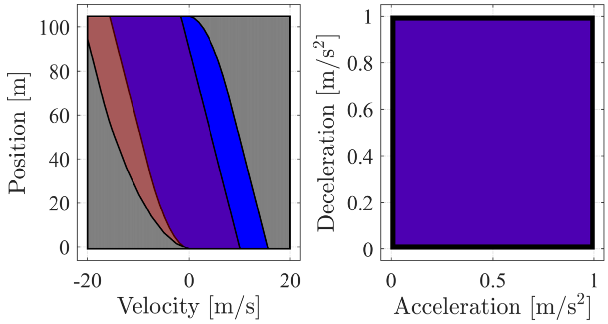


Fig. 8. Example of constraint tightening with projections of various output constraint sets on the position and velocity states to the left and the acceleration and deceleration inputs to the right. Black denotes the original output constraints $\mathcal{Y}$ while gray denotes the robust output constraints $\hat{\mathcal{Y}} = \hat{\mathcal{Y}}_2(k_2), \forall k_2$. Blue denotes the time-varying tightened output constraint set $\hat{\mathcal{Y}}_1(k_1), \forall k_1 \neq 5$, while semi-transparent red denotes $\hat{\mathcal{Y}}_1(k_1), k_1 = 5$.

The disturbances $d(k)$ are shown in Fig. 7, which consist of a known pulse to the second state of magnitude 1.5 from 45 to 55 seconds. Note that the known disturbances are permitted to change between updates of the upper level controller. This unique feature of the proposed approach is enabled by calculating the equivalent known disturbances in (7) and the time-varying constraint tightening in (15). The unknown disturbances are independently generated from a uniformly distributed random signal bounded such that $\|\Delta d(k)\|_\infty \leq 0.01$. The static feedback control gain $K \in \mathbb{R}^{m \times n}$ from (8) compensates for these unknown disturbances and was designed as a discrete-time linear-quadratic regulator with weighting matrices $Q = I_n, R = I_m$.

Fig. 8 provides an example of the constraint tightening used by the two controllers of the hierarchy. Given

Table 1
Complexity and Computation Time of Waysets

Complexity			
Wayset	CG-Rep	H-Rep	min H-Rep
	$n_c \times n_g$	n_h	n_h
Maximum	30×60	6913	197
Mean	30×60	3120	108
Minimum	30×60	38	21
Approximate Computation Time (seconds)			
	CG-Rep	H-Rep	min H-Rep
Maximum	0.006	393	353
Mean	0.002	120	114
Minimum	0.0006	0.51	0.38

the output constraint set $\mathcal{Y}$, the time-invariant tightened output constraint set $\hat{\mathcal{Y}} = \hat{\mathcal{Y}}_2(k_2), \forall k_2$, used by the lower-level controller $\mathbf{C}_2$, is calculated based on (13), where an outer approximation of $\mathcal{E}$ is calculated using the results from [31]. Two examples of $\hat{\mathcal{Y}}_1(k_1)$ are shown to demonstrate **Lemma 1** and the dependency of $\hat{\mathcal{Y}}_1(k_1)$ on the time-varying known disturbance.

Fig. 9 shows simulation results using the proposed wayset-based hierarchical controller compared to a shrinking horizon centralized controller that predicts to the end of system operation and a receding horizon centralized controller (Cent Short) with a short prediction horizon of 10 time steps. Since tracking a desired position of 100 meters from 10 to 90 seconds is a major objective for the operation of the system, the first subplot in Fig. 9 highlights this part of operation. Both the centralized and hierarchical controllers track the desired reference while compensating for the known and unknown disturbances and satisfy the output constraints, shown here by keeping position below 105 meters. Alternatively, the optimization problem for the 10-step receding horizon centralized controller becomes infeasible at $t = 11$ seconds when the velocity of the vehicle is too high to avoid violating the position constraint. The second and third subplots show that the wayset-based hierarchy satisfies the terminal constraint while maintaining a positive amount of stored energy. Finally, the fourth subplot shows the trajectory for input 3, from 30 to 70 seconds. Due to the limited amount of on-board energy, neither controller is able to track the desired load power and must shed some of the load to satisfy the terminal constraint. Centralized MPC evenly distributes this load shedding while $\mathbf{C}_2$ of the hierarchy is more greedy and only load sheds once it is required to satisfy the wayset constraint determined by $\mathbf{C}_1$.

Table 1 shows the maximum, mean, and minimum complexity of the waysets for this example simulation using CG-Rep and H-Rep along with the computation time. Both H-Rep and min H-Rep calculations were performed using the Multi-Parametric Toolbox (MPT) [39]. For CG-Rep and H-Rep, no attempt was made to remove redundant constraints. Overall, the CG-Rep achieves three to five orders-of-magnitude reduction in computation time, enabling on-line calculation of waysets. Note that 60 generators and 30 equality constraints were used to represent each wayset in CG-Rep regardless of the shape of each wayset. For scalability of the CG-Rep wayset calculations, it will be important to maintain a desired degree-of-freedom order, $o_d \leq o_d^{des}$. In [25], generator and constraint reduction techniques are presented that result in outer-approximations. However, for wayset calculations, complexity reduction must create inner-approximations. The development of these techniques is the focus of future work.

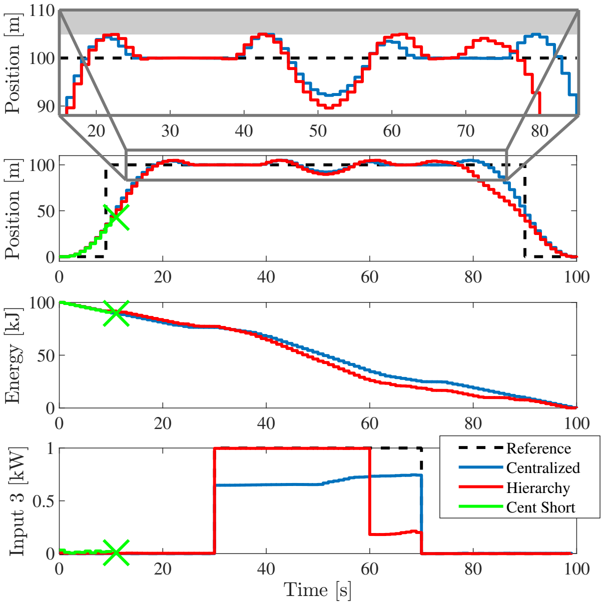


Fig. 9. Simulation results comparing the shrinking horizon centralized controller, receding horizon centralized controller with a short prediction horizon, and a two-level hierarchical controller.

7.2 *Thermal Example*

To evaluate the scalability of the hierarchical approach, consider the thermal system shown in Fig. 10, where $T_i, i \in [1, n]$ are the temperatures of n thermal elements arranged in a chain, each with a thermal capacitance C_i. Heat transfer, Q_i between thermal elements T_i and T_{i+1} is controlled by the coolant mass flow rate $\dot{m}_i$ resulting in $Q_i = \dot{m}_i c_p (T_i - T_{i+1})$, where c_p is the specific heat of the coolant. Disturbances consist of the heat input Q_0 and the ambient temperature T_∞. From conservation of energy, the nonlinear, continuous-time dynamics are $C_i \dot{T}_i = Q_{i-1} - Q_i, \forall i \in [1, n]$. For the following results, $C_i = 10^4$ J/K and $c_p = 4181$ J/(kg K) are assumed.

To represent this system in the form of (1), these dynamics are linearized about a nominal mass flow rate $\dot{m}_i = \dot{m}^o = 0.25$ kg/sec and temperature difference

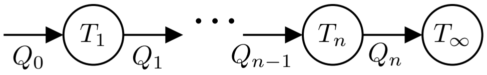


Fig. 10. Thermal system comprised of a disturbance heat input Q_0 and controllable heat transfer $Q_i, i \in [1, n]$, between n thermal elements of temperature T_i and ambient surroundings of temperature T_∞.

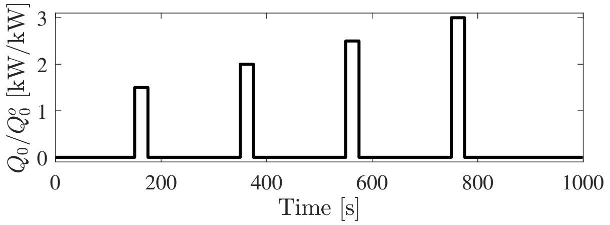


Fig. 11. Known disturbance profile for the heat input Q_0.

$T_i - T_{i+1} = \Delta T^o = 50$K and discretized with a time step size of 1 second. The corresponding steady-state heat input is $Q_0^o = \dot{m}^o c_p \Delta T^o$. From this linearization, the states $x(k) \in \mathbb{R}^n$ represent temperature deviations from nominal and the inputs $u(k) \in \mathbb{R}^n$ represent mass flow rate deviations from nominal. The disturbances $d(k) = \hat{d}(k) + \Delta d(k) \in \mathbb{R}^2$ represent deviations from the nominal heat input and nominal ambient temperature, satisfying **Assumption 2**. Finite operation is defined for 1000 seconds ($k_F = 1000$) with the known ambient temperature remaining at nominal and the known heat input consisting of four pulses as shown in Fig. 11. The unknown disturbances are independently generated from a uniformly distributed random signal bounded such that $|\Delta d(k)| \leq [0.1 Q_0^o \quad 5]^T$. As with the previous vehicle example, the static feedback control gain K from (8) was designed as a discrete-time linear-quadratic regulator with weighting matrices $Q = R = I_n$.

For a three-level hierarchical controller, the system and lowest level controller have time step sizes of $\Delta t = \Delta t_M = 1$ second while the middle and upper level controllers have time step sizes of $\Delta t_2 = 5$ and $\Delta t_1 = 40$ seconds, respectively. As a result, $\nu_1 = 40$, $\nu_2 = 5$, and the maximum prediction horizons are $N_1 = 25$, $N_2 = 8$, and $N_3 = 5$ steps. The output constraints $\mathcal{Y}$ are defined such that $\|x(k)\|_\infty \leq 100$ and $\|u(k)\|_\infty \leq 0.25$, $\forall k \in [0, k_F - 1]$. The terminal constraint simply enforces $\|x(k_F)\|_\infty \leq 100$. Given an initial state of $x(0) = 0$, the desired operation is to satisfy the output and terminal constraints while minimizing the control inputs to their lower bounds ($u(k) = -0.25$). A quadratic cost function is used as in (24) with $\Lambda = I_n$.

Fig. 12 shows the state trajectories for a three element system ($n = 3$) under the proposed wayset-based hierarchical controller compared to a shrinking horizon centralized controller that predicts to the end of system operation and a receding horizon centralized controller with a short prediction horizon of 5 time steps. For the disturbances shown in Fig. 11, the heat pulses are so large that there does not exist a steady state under these heat loads that satisfies the specified output constraints. Thus, the controller needs to pre-cool each thermal element to take advantage of the thermal capacitance in the system. Both the shrinking horizon controller and the hierarchical controller achieve this pre-cooling effectively, demonstrating that waysets can effectively constrain short-term system operation to guarantee long-term output constraint satisfaction beyond the prediction horizon of the lower-level controllers. Alternatively, the receding horizon centralized controller becomes infeasible at $t = 157$ seconds where, given the short prediction horizon, the controller cannot pre-cool the system enough prior to the heat load to avoid violating the temperature constraint.

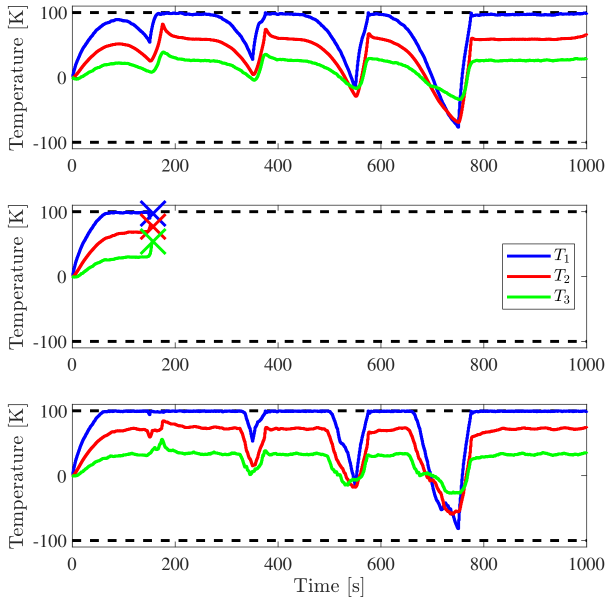


Fig. 12. Simulation results comparing the shrinking horizon centralized controller (top), receding horizon centralized controller with a short prediction horizon (middle) and a three-level hierarchical controller (bottom).

The value of the proposed hierarchical control approach is the scalability with respect to prediction horizon and system order. For the results shown in Fig. 12 with $n = 3$ states and $k_F = 1000$ steps, the average computation time is 5.96 seconds for the centralized controller while the average computation times, including wayset calculations, for the hierarchical controller C_1, C_2 and C_3 are 0.12, 0.03, and 0.02 seconds, respectively. To demonstrate scalability with respect to prediction horizon, Fig. 13 shows average computation times for the centralized and hierarchical controllers for prediction horizons ranging from $k_F = 200$ to $k_F = 1000$ steps for a three element system ($n = 3$). As indicated by the dashed line, the centralized controller is able to maintain real-time calculation speed, $t_{calc} \leq \Delta t$, for prediction horizons of $k_F \leq 360$ steps. In these cases, a centralized approach is practical and preferable to the hierarchical controller. However, for longer prediction horizons, $k_F > 360$, a centralized approach is no longer viable under the available

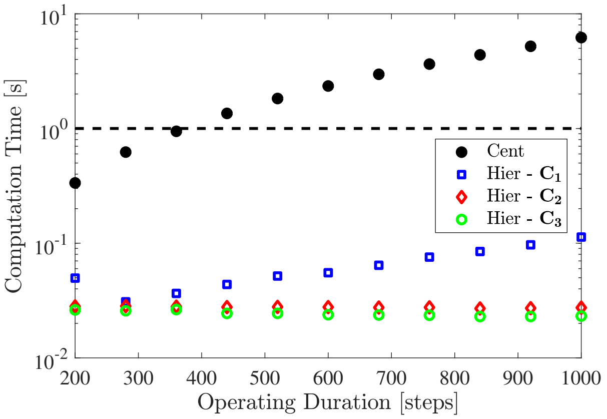


Fig. 13. Average computation time for the centralized and hierarchical controller as a function of operating duration.

computational resources, warranting the proposed hierarchical approach with two orders-of-magnitude faster computation times. For these results, only the prediction horizon N_1 of the upper level controller was varied to accommodate the change in k_F. For the hierarchical MPC formulation, real-time execution only requires $t_{calc,i} \leq \Delta t_i$ for each controller, and thus the upper-level controllers are allotted more time to solve their optimization problems. Extending the proposed theoretical hierarchical MPC formulation to directly account for computational delay, as in [40], is the focus of future work.

To demonstrate scalability with respect to system order, Fig. 14 shows average computation times for the centralized and hierarchical controller for system orders ranging from $n = 3$ to $n = 10$ with $k_F = 360$ steps. While the complexity of the waysets grows linearly with system order, the computation times of the controllers within the hierarchy remain significantly faster than that of the centralized controller. However, for the tenth order system, the waysets used by $\mathbf{C}_2$ in CG-Rep had 800 generators and 720 constraints, thus motivating future research in lower-complexity inner approximations of waysets for improved scalability of the proposed hierarchical approach.

8 CONCLUSIONS

A multi-level vertical hierarchical MPC formulation was presented for constrained linear systems. A robust MPC formulation was used for each controller to guarantee state and input constraint satisfaction in the presence of known and unknown disturbances. Waysets were developed as a novel coordination mechanism between controllers at different levels of the hierarchy. These waysets served as terminal constraints for lower-level controllers, providing flexibility in short-term operation of the system while guaranteeing long-term ability to satisfy output and terminal constraints. Using a constrained zonotope representation, waysets were efficiently computed on-line based on the state trajectories

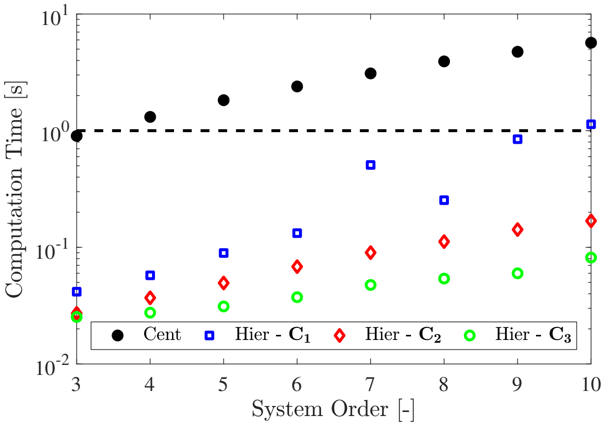


Fig. 14. Average computation time for the centralized and hierarchical controller as a function of system order.

determined by upper-level controllers. Numerical examples demonstrated the performance and scalability of the wayset-based hierarchy compared to centralized MPC approaches. Future work will focus on the efficient calculation of lower complexity inner-approximations of the waysets for improved scalability and the extension of this work to hybrid and nonlinear systems.

References

[1] J. Seok, I. Kolmanovsky, and A. Girard, "Coordinated Model Predictive Control of Aircraft Gas Turbine Engine and Power System," *Journal of Guidance, Control, and Dynamics*, vol. 40, no. 10, 2017.

[2] D. B. Doman, "Rapid Mission Planning for Aircraft Thermal Management," *AIAA Guidance, Navigation, and Control Conference*, 2015.

[3] B. Sampathnarayanan, S. Onori, and S. Yurkovich, "An optimal regulation strategy with disturbance rejection for energy management of hybrid electric vehicles," *Automatica*, vol. 50, pp. 128–140, 2014.

[4] M. Josevski and D. Abel, "Multi-time Scale Model Predictive Control Framework for Energy Management of Hybrid Electric Vehicles," *IEEE Conference on Decision and Control*, pp. 2523–2528, 2014.

[5] F. Wang, M. A. M. Zulkefli, Z. Sun, and K. A. Stelson, "Investigation on the Energy Management Strategy for Hydraulic Hybrid Wheel Loaders," *Dynamic Systems and Control Conference*, 2013.

[6] F. Kennel, D. Görges, and S. Liu, "Energy Management for Smart Grids With Electric Vehicles Based on Hierarchical MPC," vol. 9, no. 3, pp. 1528–1537, 2013.

[7] J. M. Guerrero, M. Chandorkar, T.-L. Lee, and P. C. Loh, "Advanced Control Architectures for Intelligent Microgrids - Part I: Decentralized and Hierarchical Control," *IEEE Transactions on Industrial Electronics*, 2013.

[8] R. Zamora and A. K. Srivastava, "Controls for microgrids with storage: Review, challenges, and research needs," *Renewable and Sustainable Energy Reviews*, 2010.

[9] R. Negenborn, A. Sahin, Z. Lukszo, B. De Schutter, and M. Morari, "A non-iterative cascaded predictive control approach for control of irrigation canals," *IEEE International Conference on Systems, Man and Cybernetics*, 2009.

[10] C. Ocampo-Martinez, D. Barcelli, V. Puig, and A. Bemporad, “Hierarchical and decentralised model predictive control of drinking water networks: application to Barcelona case study,” *IET Control Theory & Applications*, vol. 6, no. 1, pp. 62–71, 2012.

[11] A. Richards and J. P. How, “Model predictive control of vehicle maneuvers with guaranteed completion time and robust feasibility,” *American Control Conference*, 2003.

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

[13] R. Scattolini, “Architectures for distributed and hierarchical Model Predictive Control - A review,” *Journal of Process Control*, vol. 19, pp. 723–731, 2009.

[14] M. Farina, X. Zhang, and R. Scattolini, “A Hierarchical MPC Scheme for Coordination of Independent Systems With Shared Resources and Plug-and-Play Capabilities,” *IEEE Transactions on Control Systems Technology*, 2018.

[15] D. Barcelli, A. Bemporadz, and G. Ripaccioli, “Hierarchical multi-rate control design for constrained linear systems,” *Proceedings of the IEEE Conference on Decision and Control*, pp. 5216–5221, 2010.

[16] R. Scattolini, P. Colaneri, and D. D. Vito, “A switched MPC approach to hierarchical control,” *17th IFAC World Congress*, pp. 7790–7795, 2008.

[17] R. Scattolini and P. Colaneri, “Hierarchical model predictive control,” *IEEE Conference on Decision and Control*, pp. 4803–4808, 2007.

[18] D. Barcelli, A. Bemporad, and G. Ripaccioli, “Decentralized Hierarchical Multi-Rate Control of Constrained Linear Systems,” *18th IFAC World Congress*, pp. 277–283, 2011.

[19] D. Barcelli, A. Bemporadz, and G. Ripaccioli, “Hierarchical multi-rate control design for constrained linear systems,” *Proceedings of the IEEE Conference on Decision and Control*, pp. 5216–5221, 2010.

[20] C. Vermillion, A. Menezes, and I. Kolmanovsky, “Stable hierarchical model predictive control using an inner loop reference model and λ-contractive terminal constraint sets,” *Automatica*, vol. 50, pp. 92–99, 2014.

[21] M. Brdys, M. Grochowski, T. Gminski, K. Konarczak, and M. Drewa, “Hierarchical predictive control of integrated wastewater treatment systems,” *Control Engineering Practice*, vol. 16, pp. 751–767, 2008.

[22] J. P. Koeln and A. G. Alleyne, “Two-Level Hierarchical Mission-Based Model Predictive Control,” *American Control Conference*, pp. 2332–2337, 2018.

[23] R. J. M. Afonso, R. K. H. Galvao, and K. H. Kienitz, “Predictive control with trajectory planning in the presence of obstacles,” *UKACC International Conference on Control*, 2012.

[24] R. C. Shekhar, M. Kearney, and I. Shames, “Robust Model Predictive Control of Unmanned Aerial Vehicles Using Waysets,” *Journal of Guidance, Control, and Dynamics*, vol. 38, 2015.

[25] J. K. Scott, D. M. Raimondo, G. R. Marseglia, and R. D. Braatz, “Constrained zonotopes: A new tool for set-based estimation and fault detection,” *Automatica*, vol. 69, pp. 126–136, 2016.

[26] O. Maler, “Computing reachable sets: an introduction,” *Tech. Rep. French National Center of Scientific Research*, 2008.

[27] M. Althoff, O. Stursberg, and M. Buss, “Computing reachable sets of hybrid systems using a combination of zonotopes and polytopes,” *Nonlinear Analysis: Hybrid Systems*, vol. 4, no. 2, pp. 233–249, 2010.

[28] J. M. Bravo, T. Alamo, and E. F. Camacho, “Robust MPC of constrained discrete-time nonlinear systems based on approximated reachable sets,” *Automatica*, 2006.

[29] J. P. Koeln and B. M. Hencey, “Constrained Hierarchical MPC via Zonotopic Waysets,” *American Control Conference*, pp. 4237–4244, 2019.

[30] D. Mayne, M. Seron, and S. Raković, “Robust Model Predictive Control of Constrained Linear Systems with Bounded Disturbances,” *Automatica*, vol. 41, pp. 219–224, 2005.

[31] S. Raković, E. C. Kerrigan, K. I. Kouramas, and D. Q. Mayne, “Invariant Approximations of the Minimal Robust Positively Invariant Set,” *IEEE Transactions on Automatic Control*, vol. 50, no. 3, pp. 406–410, 2005.

[32] I. Kolmanovsky and E. G. Gilbert, “Theory and Computation of Disturbance Invariant Sets for Discrete-Time Linear Systems,” *Mathematical Problems in Engineering*, vol. 4, pp. 317–367, 1998.

[33] L. Hewing and M. N. Zeilinger, “Stochastic Model Predictive Control for Linear Systems Using Probabilistic Reachable Sets,” in *Proceedings of the IEEE Conference on Decision and Control*, 2019.

[34] F. Scibilia, S. Olaru, and M. Hovd, “On feasible sets for MPC and their approximations,” *Automatica*, vol. 47, pp. 133–139, 2011.

[35] L. J. Guibas, A. Nguyen, and L. Zhang, “Zonotopes as bounding volumes,” *Proceedings of the fourteenth annual ACM-SIAM symposium on discrete algorithms*, 2003.

[36] A. Girard, “Reachability of uncertain linear systems using zonotopes,” in *Hybrid Systems: Computation and Control*, M. Morari and L. Thiele, Eds. Springer, 2005, pp. 291–305.

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

[38] G. O. Inc., “Gurobi Optimizer Reference Manual,” 2016.

[39] M. Herceg, M. Kvasnica, C. Jones, and M. Morari, “Multi-Parametric Toolbox 3.0,” *European Control Conference*, 2013.

[40] Y. Su, K. K. Tan, and T. H. Lee, “Computation delay compensation for real time implementation of robust model predictive control,” *Journal of Process Control*, vol. 23, 2013.