Robust Model Predictive Control of Unmanned Aerial Vehicles using Waysets

Rohan C. Shekhar[1]
The University of Melbourne, Victoria, 3010, Australia.

Michael Kearney[2]
The University of Queensland, St Lucia, Queensland, 4072, Australia

Iman Shames[3]
The University of Melbourne, Victoria, 3010, Australia.

This paper introduces a new formulation of model predictive control (MPC) for robust trajectory guidance of unmanned aerial vehicles (UAVs). It generalises the ubiquitous concept of waypoints to *waysets*, in order to provide robustness to bounded state disturbances in the presence of obstacles. Using a variable horizon formulation of MPC, it shows how wayset guidance combined with constraint tightening can guarantee robust recursive feasibility and finite-time completion of a control maneuver. Simulations on a point mass fixed-wing UAV model moving through a field of obstacles with wind disturbances demonstrate significant computational benefits from using waysets when compared to existing mixed-integer optimization methods that employ long prediction horizons. Using the controller's robustness to mitigate linearization error, an additional example implements the strategy on a simulated quadrotor, demonstrating how waysets can be utilized to control more complex nonlinear systems.

[1] Research Fellow, Department of Mechanical Engineering, rshekhar@unimelb.edu.au.
[2] Lecturer, School of Mechanical and Mining Engineering, m.kearney@uq.edu.au
[3] Senior Lecturer, Department of Electrical and Electronic Engineering, iman.shames@unimelb.edu.au

Nomenclature

x	$=$ state vector
u	$=$ input vector
y	$=$ output vector
d	$=$ disturbance vector
w_i	$=$ ith waypoint
p	$=$ number of waypoints
o	$=$ number of unsafe regions (obstacles)
N	$=$ horizon length
$\bar{N}$	$=$ overall completion time
$\mathcal{S}$	$=$ wayset shape
$\mathcal{W}_i$	$=$ ith wayset
$\mathcal{D}$	$=$ disturbance set
$\mathcal{Y}$	$=$ output constraint set
$\mathcal{O}_i$	$=$ ith unsafe region (obstacle)
$\mathcal{C}_i$	$=$ corridor connecting wayset i and $i+1$
$\widetilde{\mathcal{Q}}(j)$	$=$ jth tightening of set $\mathcal{Q}$
$\ominus$	$=$ Pontryagin set difference operator
$\text{co}\{\cdot\}$	$=$ convex hull
$\mathcal{A}^S$	$=$ symmetric of set $\mathcal{A}$
$\mathcal{A}^C$	$=$ complement of set $\mathcal{A}$
$\lfloor x \rfloor$	$=$ greatest integer not exceeding x
$\lVert v \rVert$	$=$ 2-norm (Euclidean norm) of vector v
$\mathbb{Z}_{[\alpha,\beta]}$	$=$ $\{\alpha, \alpha+1, \ldots, \beta\}$ for integer numbers $\alpha < \beta$
$\mathbf{1}_n$	$=$ Column vector of ones having length n.

I. Introduction

The capability to plan and execute trajectories that safely take an unmanned aerial vehicle (UAV) from a starting point to a target region is essential to its autonomous operation. To provide this capability, planned trajectories must be executable by the vehicle (i.e. dynamically feasible), avoid obstacles, other vehicles and designated no-fly areas, have a sufficient level of optimality (in terms of fuel usage and/or travel time), and be robust to disturbances that are inherent in UAV operations, such as wind and turbulence. Additionally, this trajectory must be computed, recomputed, and executed in real-time to be able to handle changes to the UAV's target location and environment. Due to the importance of establishing this capability to future UAV operations, a wide variety of approaches have been applied to this problem including [1–6], and it has been the topic of a number of survey papers [7–9]. Among these approaches, model predictive control (MPC) [10] - also known as receding horizon control [11, Pages 7-8] - is able to satisfy the requirements for effective UAV trajectory planning, which incorporates constraint handling and robustness to bounded disturbances, within a number of settings [2, 12–14]. However, its key limitation is the computational effort required to calculate a trajectory. This effort primarily stems from long prediction horizons and the requirement to solve a mixed-integer program to capture the non-convex obstacle avoidance constraints [1]. Computational effort is a significant limitation when real-time controller implementation is desired. This is especially the case for a high sampling rate, where MPC is used for controlling fast system dynamics, or when guarantees of inter-sample constraint satisfaction necessitate such a sample rate.

Within the existing literature, the predominant strategy to reduce the computational cost of solving the mixed-integer program is to reduce the prediction (or planning) horizon of the predictive controller. This is achieved by either approximating the remainder of the trajectory beyond this horizon to the target via a *cost to go* [2, 12], or by travelling to the target via a number of intermediate waypoints [15]. The former suffers from the lack of a guaranteed recursively feasible path to the target, whilst the latter does not provide guaranteed robustness, since it is impossible to control the UAV to a point in the presence of persistent disturbances. Another approach taken to reduce the computational cost of mixed-integer MPC-based UAV trajectory planning is the *tunnel-MILP* approach presented in [16, 17]. This involves planning a path between the start point and the target, and creating a tunnel around this path made up of a sequence of obstacle-free convex polytopes. The obstacle avoidance constraints are replaced by constraining the UAV to remain within the tunnel. This approach, while shown to be less computationally intensive for trajectories that need to avoid many obstacles [16], still requires integer variables to index the convex polytopes that form the tunnel. The tunnel-MILP approach has been combined with the waypoint approach in [18], resulting in further computational reductions. However, there are no robustness guarantees, and providing such guarantees for tunnel-MILP approaches is yet to be considered in the literature.

The novel approach presented in this paper does not rely on mixed-integer programming for obstacle avoidance, whilst still guaranteeing robust constraint satisfaction in the presence of additive state disturbances, such as turbulence or gusts. Firstly, a sequence of *waysets* from the start point to the target are characterized and calculated. Furthermore, the relationship between the more general notion of waysets and the well-known concept of waypoints is elucidated. Following this, convex constraints are enforced to ensure that the UAV can be robustly driven between two successive waysets using variable-horizon MPC [14]. In particular, reachability conditions are developed to place the sequence of waysets so that the UAV can be robustly guided through their connecting corridor within a fixed number of time-steps. A robust MPC scheme is then formulated to travel between pairs of consecutive waysets until the target is reached. The approach is shown to guarantee finite-time arrival at the target for appropriately chosen input cost weightings. Finally, the proposed scheme is applied to two simulated examples, namely navigation of a simplified fixed-wing UAV in the presence of wind dusturbance and wayset guidance of a nonlinear quadrotor model. The robustness of the controller is used to handle the wind disturbance and minimum-speed constraints for the fixed-wing UAV, and linearization error for the quadrotor.

The structure of this paper is as follows: Section II provides a problem formulation for the robust UAV trajectory planning problem. Section III presents the underlying robust MPC formulation from [19], and uses it to develop the corridor constraints and wayset placement conditions. Section IV analyses the theoretical results of the developed approach and Section V presents the simulation study. Conclusions are drawn and future work is elucidated in Section VI.

II. Problem Formulation

A. Definitions

The following defnitions are needed to develop the results presented in this paper. For a set $\mathcal{A} \in \mathbb{R}^n$, define its *symmetric* and *complement*, respectively, by

$$\mathcal{A}^S := \{z \in \mathbb{R}^n \mid -z \in \mathcal{A}\}$$

$$\mathcal{A}^C := \{z \in \mathbb{R}^n \mid z \notin \mathcal{A}\}.$$

For sets $\mathcal{A} \in \mathbb{R}^n$ and $\mathcal{B} \in \mathbb{R}^n$, the Minkowski sum and Pontryagin difference operations are respectively defined by

$$\mathcal{A} \oplus \mathcal{B} := \{z \mid \exists a \in \mathcal{A}, b \in \mathcal{B}, \text{ s.t. } z = a + b\}$$

$$\mathcal{A} \ominus \mathcal{B} := \{z \mid z + b \in \mathcal{A}, \forall b \in \mathcal{B}\}.$$

These operations are more commonly known as *dilation* and *erosion* respectively in the field of mathematical morphology, allowing the properties detailed in [20] to be utilised.

B. Control Problem

Consider a UAV modelled by the linear system

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

where $x(k) \in \mathbb{R}^n$ is the state, $u(k) \in \mathbb{R}^m$ is the input, $d(k) \in \mathcal{D}$ are unknown disturbances, and $\mathcal{D} \subset \mathbb{R}^n$ is a compact disturbance set that contains the origin. Additionally, it is assumed that the system is controllable and all states are measurable. For a UAV, the inputs will typically represent thrust and and the states will comprise positions, velocities and possibly the fuel level.

The system is subject to polytopal state and input constraints, which are compactly represented in the form

$$y(k) \in \mathcal{Y} := \{y \mid Ey \leq f\}, \tag{2}$$

where an output

$$y(k) := Cx(k) + Du(k)$$

is defined to allow the application of coupled constraints between the states and inputs of the system.

Problem 1. *Given a vehicle governed by* (1) *with an initial state* x_0*, a target set* $\mathcal{T}$*, and* o *known unsafe regions* $\mathcal{O}_l$*,* $l = 1, \dots, o$*, where* $\mathcal{T} \cap \mathcal{O}_l = \emptyset$*, it is desired to compute an* N*-step sequence of control inputs*

$$\boldsymbol{u}_N \coloneqq \{u(0), \dots, u(N-1)\},$$

for some completion time $N \in \mathbb{Z}_{[1,\bar{N}]}$ *that minimises the cost function*

$$\sum_{k=0}^{N-1} (1 + \alpha \, \|u(k)\|), \tag{3}$$

subject to the constraints (2)*, whilst ensuring that* $x(N) \in \mathcal{T}$ *and* $x(k) \notin \mathcal{O}_l$*,* $l = 1, \dots, o$ *for all* $k < N$ *and* $\bar{N}$ *is a design constant.*

The cost function (3) penalises a weighted sum of the completion time to $\mathcal{T}$ and the control effort applied, where the weighting is determined by some $\alpha > 0$. It is appropriate to a UAV scenario, where the control problem often has a minimum-time objective, with the tuning parameter α used to improve fuel economy or produce smooth trajectories. Later in the paper an algorithm for choosing $\bar{N}$ is introduced.

III. Controller Formulation

Problem 1 is addressed by decomposing the overall control problem into a series of set-to-set subproblems, to which robust MPC using constraint tightening is applied. In order to achieve this, some intermediary sets $\mathcal{W}_i$, $i \in \mathbb{Z}_{[0,p]}$, denoted *waysets*, are computed. Then, the system is driven from the initial state x_0 sequentially through the waysets $\mathcal{W}_1$ to $\mathcal{W}_p$, where $\mathcal{W}_p \in \mathcal{T}$. Furthermore, the sets are designed in such a way as to guarantee obstacle-free robust reachability between adjacent sets, that is, $x(k) \notin \mathcal{O}_i$, $i \in \mathbb{Z}_{[1,o]}, \forall d(k) \in \mathcal{D}$, while travelling between $\mathcal{W}_i$ and $\mathcal{W}_{i+1}$. This is formalised in what follows.

A. Wayset Definition

Define a sequence of waysets by

$$\mathcal{W}_i \coloneqq w_i \oplus \mathcal{S} = \{w_i + w \mid w \in \mathcal{S}\}, \forall i \in \mathbb{Z}_{[0,p]}, \tag{4}$$

where

$$\mathcal{S} \coloneqq \{x \mid Gx \leq h\},$$

is a fixed convex polytope defining the wayset shape, and w_i, $i = 0, \ldots, p$ are a sequence of *waypoints*, with $w_0 \coloneqq x_0$. These waypoints parameterize the wayset locations. In order to use the waysets for decomposing Problem 1, it is required that

$$\begin{aligned} &\mathcal{W}_i \subseteq \bigcup_{j=0}^{N} \mathcal{R}_j(\mathcal{W}_{i+1}), \forall i \in \mathbb{Z}_{[0,p-1]} \\ &\mathcal{W}_p \subseteq \mathcal{T}, \end{aligned} \tag{5}$$

where the set-valued map $\mathcal{R}_j(\cdot)$ denotes the j-step robust controllable set, defined by

$$\mathcal{R}_j(\mathcal{X}) \coloneqq \left\{ x \in \mathbb{R}^n \;\middle|\; \begin{array}{c} \exists \boldsymbol{u}_j \coloneqq \{u(0), u(1), \ldots, u(j-1)\} : \\ x(k+1) = Ax(k) + Bu(k) + d(k) \\ y(k) = Cx(k) + Du(k) \\ x(0) = x, \quad y(k) \in \mathcal{Y}, \quad x(k) \notin \mathcal{O}_i, \quad x(j) \in \mathcal{X} \\ \forall d(k) \in \mathcal{D}, \quad \forall k \in \mathbb{Z}_{[0,j-1]}, \quad \forall i \in \mathbb{Z}_{[0,p]} \end{array} \right\}.$$

This is the set of all states for which there exists a j-step sequence of control inputs that robustly drives the state of the system into $\mathcal{X}$, whilst avoiding all of the unsafe regions. If each adjacent pair of waysets satisfies (5) for $j \leq N$, where N is some horizon length, then by using an appropriate cost function and controller, Problem 1 can be solved in at most pN steps.

Note that, for robustness, it is necessary that $\mathcal{D} \subseteq \mathcal{S}$, since if this were not the case, it would be impossible to guarantee that the system state could terminate within a wayset in the presence of the disturbance.

B. Inter-Wayset Control with Robust MPC

Having defined the notion of waysets (4) and specifying their required properties (5), the sub-problem for a given pair of adjacent waysets $\mathcal{W}_i$ and $\mathcal{W}_{i+1}$ can be formulated with variable-horizon MPC, using constraint tightening for robustness [14, 21]. The disturbance feedback form of the tightening is used, as motivated by [22] and elucidated by [19]. Define the nominal prediction model by

$$\begin{aligned}\bar{x}(k+j+1|k) &= A\bar{x}(k+j|k) + B\bar{u}(k+j|k)\\ \bar{y}(k) &= C\bar{x}(k+j|k) + D\bar{u}(k+j|k)\\ \bar{x}(k|k) &= x(k),\end{aligned}$$

where the notation $\bar{x}(k+j|k)$ is used to denote the prediction of state $x(k)$ made j steps into the future from the current time k. A similar notation is adoped for the inputs and outputs, using prediction variables $\bar{u}$ and $\bar{y}$ respectively. The tightened constraints are then calculated using the matrices [23]

$$\begin{aligned}L(0) &= I\\ L(j+1) &= AL(j) + BP(j)\\ Q(j) &= CL(j) + DP(j),\end{aligned}$$

which are used to define the tightened sets

$$\begin{aligned}\widetilde{\mathcal{Y}}(j+1) &:= \widetilde{\mathcal{Y}}(j) \ominus Q(j)\mathcal{D}\\ \widetilde{\mathcal{S}}(j+1) &:= \widetilde{\mathcal{S}}(j) \ominus L(j)\mathcal{D}\\ \widetilde{\mathcal{O}}_l(j+1) &:= \widetilde{\mathcal{O}}_l(j) \oplus L(j)\mathcal{D}^S, \forall l \in \mathbb{Z}_{[1,o]}\\ \widetilde{\mathcal{Y}}(0) := \mathcal{Y}, \quad \widetilde{\mathcal{T}}(0) &:= \mathcal{T}, \quad \widetilde{\mathcal{S}}(0) := \mathcal{S},\end{aligned} \tag{6}$$

for all $j \in \mathbb{Z}_{[0,N-1]}$, where N is the horizon length. The matrices $P(j)$ represent a disturbance-feedback constraint tightening policy, which is assumed to be designed in advance.

For a given $x(k) \in \mathcal{W}_i$, a predictive controller to steer the state of the system to $\mathcal{W}_{i+1}$ can be designed by finding the sequence of predicted inputs

$$\bar{\boldsymbol{u}}_N^*(x(k)) := \{u^*(k|k), u^*(k+1|k), \ldots, u^*(k+N-1|k)\} \tag{7}$$

and a horizon length N^* such that

$$(\boldsymbol{u}_N^*(x(k)), N^*) = \underset{\bar{\boldsymbol{u}}_N, N}{\arg\min}\, J(x(k), \bar{\boldsymbol{u}}_N, N), \tag{8}$$

where

$$\bar{\boldsymbol{u}}_N(k) := \{\bar{u}(k|k), \bar{u}(k+1|k), \ldots, \bar{u}(k+N-1|k)\}, \tag{9}$$

for some horizon length dependent cost function $J(\cdot,\cdot,\cdot)$, subject to the constraints

$$\begin{aligned}
\bar{y}(k+j|k) &\in \widetilde{\mathcal{Y}}(j) \\
\bar{x}(k+j|k) &\notin \widetilde{\mathcal{O}}_l(j), \forall l \in \mathbb{Z}_{[1,o]} \\
\bar{x}(k+N|k) &\in \widetilde{W}_{i+1}(j),
\end{aligned}$$

where

$$\widetilde{W}_{i+1}(j) \coloneqq w_{i+1} \oplus \widetilde{\mathcal{S}}(j).$$

Using the optimal predicted input sequence and horizon length, the variable-horizon MPC algorithm of [14] is applied as the control law to guide the state to $\mathcal{W}_{i+1}$, namely

$$u(k) = \bar{u}^*(k|k).$$

As proven in [14], this controller guarantees robust recursive feasibility and finite-time completion for an appropriate cost function and initial feasible solution, which allows the controllers to steer the state of the system starting from any point in $\mathcal{W}_i$ to a point in $\mathcal{W}_{i+1}$ if this pair of waysets satisfy (5). Note that this condition also guarantees the existence of an initial feasible solution.

The particular implementation of the variable horizon presented in [14] uses integer variables to allow the completion time to be quantified within the optimization problem. However, it is also possible to solve the optimization problem over all fixed horizon lengths up to some maximum, and choose the solution with the lowest cost. The former approach will be used in the sections to follow, since the worst case complexity of using MILP for the variable horizon is equivalent to solving for all horizon lengths, but is potentially much faster in practice.

C. Wayset Reachability using Corridor Constraints

Placing a general pair of waysets such that (5) is satisfied is nontrivial, since the union over controllable sets combined with the obstacle-avoidance constraints is highly non-convex. In addition, even with constraint tightening, (6) requires a large number of enlarged obstacle sets to be calculated and stored. Instead *sufficient* conditions can derived to ensure satisfaction of (5) with purely convex constraints. These conditions involve more stringent fixed-horizon *corridor reachability* constraints,

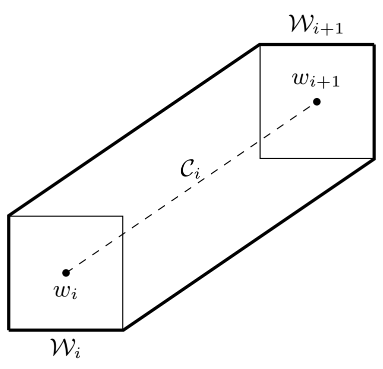


Fig. 1 Illustration of corridor constraints for adjacent waysets.

which require the existence of trajectories that lie in a corridor defined by the convex hull of the two waysets, which is chosen to be obstacle free. This is illustrated by Fig. 1 for simple square waysets. The thick outer line depicts the convex hull. Note that all of the sets are defined in the full state-space, so will have velocity components that cannot be depicted in two dimensions. Given that conditions can be established to robustly keep the trajectory between successive waysets within the corridor, obstacles avoidance is guaranteed by design of the corridors.

The corridor is then defined as

$$\mathcal{C}_i := \text{co}\{\mathcal{W}_i, \mathcal{W}_{i+1}\}.$$

Applying constraint tightening for robustness, the corridor constraints for waysets $\mathcal{W}_i$ and $\mathcal{W}_{i+1}$ are given by

$$\widetilde{\mathcal{C}}_i(0) = \mathcal{C}_i$$

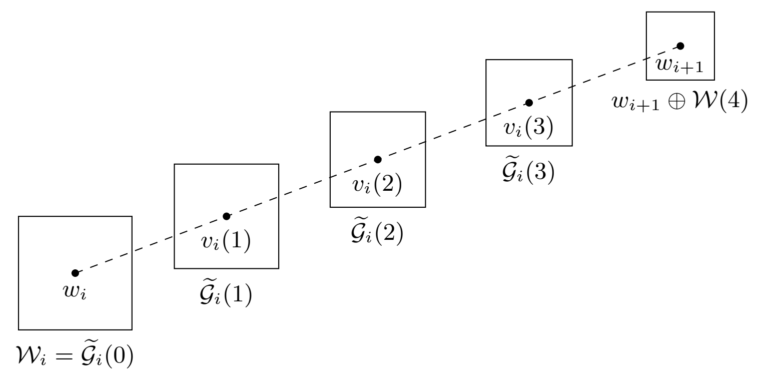
$$\widetilde{\mathcal{C}}_i(j+1) = \widetilde{\mathcal{C}}_i(j) \ominus L(j)\mathcal{D}.$$

It can be shown that the constraint $x(k+j|k) \in \widetilde{\mathcal{C}}_i(j)$ is bilinear in $x(k+j|k)$, w_i and w_{i+1}. Therefore, to retain convexity, an approximation to the true corridor is used, which is sufficient to guarantee that the predicted states lie within the corridor. Assuming a fixed-horizon arrival time of N steps between the waysets, the approximated corridor is defined by the sequence of sets

$$\widetilde{\mathcal{G}}_i(j) := v_i(j) \oplus \widetilde{\mathcal{S}}(j), \quad \forall j \in \mathbb{Z}_{[0,N-1]},$$

where

$$v_i(j) := (w_i + \gamma(j)(w_{i+1} - w_i)),$$



w_{i+1}
$w_{i+1} \oplus \mathcal{W}(4)$
$v_i(3)$
$\widetilde{\mathcal{G}}_i(3)$
$v_i(2)$
$\widetilde{\mathcal{G}}_i(2)$
$v_i(1)$
$\widetilde{\mathcal{G}}_i(1)$
w_i
$\mathcal{W}_i = \widetilde{\mathcal{G}}_i(0)$

Fig. 2 Tightened corridor approximation.

for $j \in \mathbb{Z}_{[1,N]}$, with prespecified weights $\gamma(j), \gamma(0) = 0$ that satisfy

$$0 \leq \gamma(1) \leq \gamma(2) \leq \cdots \leq \gamma(N-1) \leq 1.$$

These new constraints are shown in Fig. 2. The next result demonstrates the sufficiency of these new corridor constraints.

Theorem 1. *For any $x(0) \in \mathcal{W}_i$, there exists an N-step predicted trajectory to the tightened set $\widetilde{\mathcal{W}}_{i+1}(N)$ that satisfies the corridor constraints*

$$x(j|0) \in \widetilde{\mathcal{C}}_i(j), \tag{10}$$

for all $j \in \mathbb{Z}_{[1,N]}$, providing that

$$x(j|0) \in \widetilde{\mathcal{G}}_i(j), \tag{11}$$

for all $j \in \mathbb{Z}_{[1,N]}$.

Proof. If $x(j|0) \in \widetilde{\mathcal{C}}_i(j)$, then it must lie within a copy of $\widetilde{\mathcal{S}}(j)$ translated somewhere along the line connecting the waypoints w_i and w_{i+1}. Hence, there exists some $\gamma(j)$ such that

$$x \in (w_i + \gamma(j)(w_{i+1} - w_i)) \oplus \widetilde{\mathcal{S}}(j). \tag{12}$$

The sufficient conditions merely fix the values of $\gamma(j)$ so that the resulting problem is convex. ■

Note that, these sufficient conditions will not actually be applied when controlling the UAV. Instead, given that the wayset locations will be fixed before applying the controller, the corridor constraints (10) will be explicitly enforced. However, appropriate selection of the weights $\gamma(j)$ is required to minimize conservatism in placement of the waysets.

D. Waypoint Placement with Guaranteed Reachability

It is now possible to place waypoints in such a way as to ensure the existence of fixed N-step robustly safe trajectories between the waysets. To begin with, the N-step controllable set of wayset $\mathcal{W}_{i+1}$ through a corridor from $\mathcal{W}_i$, subject to the tightened output constraints (11) is defined as

$$\bar{\mathcal{R}}_N(w_i, w_{i+1}) := \left\{ x \in \mathbb{R}^n \;\middle|\; \begin{array}{c} \exists \bar{\boldsymbol{u}}(0)_N := \{\bar{u}(0|0), \bar{u}(1|0), \ldots \bar{u}(N-1|0)\} : \\ \bar{x}(j+1|0) = A\bar{x}(j|0) + B\bar{u}(j|0) \\ \bar{y}(j|0) = C\bar{x}(j|0) + D\bar{u}(j|0) \\ \bar{y}(j|0) \in \mathcal{Y}(j), \quad \bar{x}(j|0) \in \widetilde{\mathcal{G}}_i(j) \\ \bar{x}(0|0) = x \qquad \bar{x}(N|0) \in w_{i+1} \oplus \widetilde{\mathcal{S}}(N) \\ \forall j \in \mathbb{Z}_{[0,N-1]}. \end{array} \right\}. \tag{13}$$

This is the set of all x such that there exists an initial nominal predicted trajectory ($k = 0$) satisfying the tightened constraints and entering the succesive wayset in N steps. Note that for simplicity, time is indexed from $k = 0$ for each subproblem, which is why the trajectory defining the reachable set (13) is written as a prediction made from time zero. Since all of the constraints are polytopal, $\bar{\mathcal{R}}_N(\cdot, \cdot)$ itself is a convex polytope. It can be calculated explicitly by projection, leaving a set affinely parameterized by the waypoint positions w_i and w_{i+1}. Let this set be represented in half-space form by

$$\bar{\mathcal{R}}_N(w_i, w_{i+1}) = \{x \mid \Gamma_N x \leq \theta_N - S_N w_i - T_N w_{i+1}\}, \tag{14}$$

for some appropriately dimensioned matrices Γ_N, θ_N, S_N and T_N. The projection can be calculated using algorithms like Fourier-Motzkin elimination [24] or equality set projection [25]. The operation only needs to be performed once, since the controllable set is parameterized in terms of the waypoint position. An alternative to using projection is to explicitly ensure the existence of feasible trajectories from each vertex of $\mathcal{W}_i$ to anywhere within $\mathcal{W}_{i+1}$.

Having calculated the projection, (14) can be used to express the set membership constraint

$$\mathcal{W}_i \subseteq \bar{\mathcal{R}}_N(w_i, w_{i+1})$$

in half-space form by observing that

$$\Gamma_N x \leq \theta_N - S_N w_i - T_N w_{i+1}, \forall x \in \mathcal{W}_i$$

$$\iff \max_{x \in \mathcal{W}_i} \Gamma_N x \leq \theta_N - S_N w_i - T_N w_{i+1}$$

$$\iff \max_{x \in \mathcal{W}} \Gamma_N x \leq \theta_N - (\Gamma_N + S_N) w_i - T_N w_{i+1},$$

where the maximisation is taken row-wise and the fact that $\mathcal{W}_i = w_i \oplus \mathcal{S}$ has been exploited. Recognizing that this maximization defines a linear program, the dualization procedure of [22] can be used to write the constraint in the equivalent form

$$\mathcal{Z}(w_i, w_{i+1}, N) \neq \emptyset, \quad \forall i \in \mathbb{Z}_{[0,p-1]} \tag{15}$$

where

$$\mathcal{Z}(w_i, w_{i+1}, N) \coloneqq \{Z \mid Z^T h \leq \theta_N - (\Gamma_N + S_N) w_i - T_N w_{i+1},\ Z^T G = \Gamma_N\} \tag{16}$$

Any waypoint placement algorithm can now be augmented with the additional convex constraint (15) to ensure robust inter-wayset reachability. It is straightforward to enforce the constraints by constraining the visibility graph of the candidate waypoints with the reachability conditions. Problem 2 describes how this can be achieved.

Problem 2 (Waypoint Placement)**.** *Place p waypoints such that*

$$w_i + \gamma(w_{i+1} - w_i) \notin O_l \oplus \mathcal{S}^S, \forall \gamma \in [0, 1]. \tag{17}$$

$$\exists N \in \mathbb{Z}_{[1,\bar{N}]} : \mathcal{Z}(w_i, w_{i+1}, N) \neq \emptyset \tag{18}$$

for all $i \in \mathbb{Z}_{[0,p-1]}$ and $l \in \mathbb{Z}_{[1,o]}$.

It is worthwhile to note that this placement problem can be solved with many different off-the-shelf waypoint algorithms that ensure line-of-sight visibility along connecting lines. The only required modification is to strengthen the definition of visibility to include the reachability condition (17)-(18).

To solve Problem 2, it is first assumed that each waypoint w_i can be written as $w_i = [s_i^\top \ \ \sigma_i^\top]^\top$ where s_i is the spatial component of w_i. Furthermore, it is assumed that the other component is identical for all waypoints $\sigma_1 = \cdots = \sigma_p = \sigma$, where σ is the center of the wayset in the other non-spatial states. Thus each waypoint can be written as $w_i = \chi(s_i) = [s_i^\top \ \ \sigma^\top]^\top$.

There are many possible path planning methods that return a sequence of points that form a collision-free path (see for example, [26, Chapters 6-7] and references therein). Let the sequence $\{\bar{c}_i\}_{i=1}^{\bar{p}}$ be obtained from an arbitrary planning method. Algorithm 1 (see appendix) returns a sequence of waypoints based on the path $\{\bar{c}_i\}_{i=1}^{\bar{p}}$. This algorithm is proposed to solve Problem 2. However, for an arbitrary input sequence $\{\bar{c}_i\}_{i=1}^{\bar{p}}$, Algorithm 1 is not guaranteed to return a sequence of feasible waypoints that satisfy (17)-(18). In what follows, guarantees are provided such that Algorithm 1 returns a sequence of feasible waypoints satisfying (17)-(18) for the case where the path sequence is calculated via grid-based methods.

1. Waypoint Placement for Paths Obtained from Grid-Based Methods

Consider the case where the sequence $\{\bar{c}_i\}_{i=1}^{\bar{p}}$ is generated via methods that generate a grid map of the environment and perform a graph search. The A^* search or the D^* search are famous examples for such methods [26, Appendix H]. Assume that the environment is discretised into square *cells* of arbitrary side length $\rho > 0$. Two cells are called adjacent if they share a side. Let c be the center of a cell, then this cell is labelled *free* if there is no point s that simultanously $\|c-s\|_\infty \leq \rho/2$ and $[s^T \ \bar{\sigma}^T]^T \in O_l \oplus \mathcal{S}$ for some $\bar{\sigma}$. Consider a graph, $\mathcal{P}_{free}$ where the free cells are its vertices and an edge is incident at two vertices if the corresponding cells are adjacent. Let the sequence $\{\bar{c}_i\}_{i=1}^{\bar{p}}$ be obtained from applying the search algorithm to $\mathcal{P}_{free}$. Note that the path planning is done independent of the velocity constraints and as a result the dimensions of the search space are kept small.

Note that in general it is not guaranteed that Algorithm 1 returns a set of waypoints that satisfy (17)-(18) and terminate at the destination. Particularly consider the case where $w_i = \chi(\bar{c}_j)$ for some i and j. Now if $\mathcal{Z}(\chi(c_i), \chi(c_j), N) = \emptyset$ for all $N \in \mathbb{Z}_{[1,\bar{N}]}$, then the algorithm will not return a set of waypoints that satisfy (17)-(18) and terminate at the destination. However, it is possible to provide guarantees under some assumptions. Specifically, the following assumption is made.

Assumption 1. *Let c_i and c_j be the centers of two arbitrary adjacent cells. Then, for a given cell* *size ρ, and at least an $N \in \mathbb{Z}_{[1,\bar{N}]}$, $\mathcal{Z}(\chi(c_i), \chi(c_j), N) \neq \emptyset$ and $\mathcal{Z}(\chi(c_j), \chi(c_j), N) \neq \emptyset$.*

The following result holds for Algorithm 1.

Proposition 1. *Under Assumption 1 and provided that a collision free sequence of cell centers $\{\bar{c}_i\}_{i=1}^{\bar{p}}$ is given, Algorithm 1 returns a sequence of waypoints, $\{w_i\}_{i=1}^{p}$, $p \leq \bar{p}$, where each pair of consecutive waypoints w_i and w_{i+1} satisfy* (17)-(18).

Proof. Due to Assumption 1, in the worst case scenario $p = \bar{p}$ and for all w_i and w_{i+1} (17)-(18) are satisfied. ■

E. Inter-wayset controller

Once the waypoints have been placed, the inter-wayset optimization problem can be defined, applying constraint tightening and corridor constraints for robustly safe trajectories. Problem 3 is then applied recursively and used to implement MPC.

Problem 3.

$$J_i^*(x(k)) = \min_{\bar{\boldsymbol{u}}_N(k), N, \gamma(j)} \sum_{k=0}^{N-1} (1 + \alpha \left\| \bar{u}(k+j|k) \right\|), \tag{19}$$

subject to

$$\begin{aligned} \bar{x}(0|k) &= x(k) \\ \bar{y}(k+j|k) &\in \mathcal{Y}(j), \forall j \in \mathbb{Z}_{[0,N-1]} \\ \bar{x}(k+j|k) &\in \widetilde{\mathcal{C}}_i(j), \forall j \in \mathbb{Z}_{[1,N-1]} \\ \bar{x}(k+N|k) &\in w_{i+1} \oplus \widetilde{\mathcal{S}}(N) \\ N &\in \mathbb{Z}_{[1,\bar{N}]} \\ 0 \leq \gamma(j) &\leq 1, \forall j \in \mathbb{Z}_{[1,N-1]}, \end{aligned} \tag{20}$$

where

$$\widetilde{\mathcal{C}}_i(j) = (w_i + \gamma(j)(w_i - w_{i+1})) \oplus \widetilde{\mathcal{S}}(j)$$

Since the waypoint positions are now fixed, the weights $\gamma(j)$ become decision variables in the optimization problem. Then, the constraint (20) ensures that $x(k+j|k)$ lies within some translated copy of $\widetilde{\mathcal{S}}(j)$ located along the line connecting the waypoints. This explicitly ensures that the predicted state is contained within the tightened convex hull. Algorithm 2, as outlined in the appendix, can be applied for choosing the smallest maximum horizon length $\bar{N}$.

IV. Robustness Guarantees

The following theorems describe the robustness benefits provided by appling the approach detailed in this paper.

Theorem 2 (Corridor safety)**.** *If the waypoints w_i and w_{i+1} are placed with line-of-sight visibility to the enalrged unsafe regions $\mathcal{O}_l \oplus \mathcal{S}^S$, then*

$$\mathrm{co}\{\mathcal{W}_i, \mathcal{W}_{i+1}\} \cap \mathcal{O}_l = \emptyset,$$

for all $i \in \mathbb{Z}_{[0,p-1]}$ and $l \in \mathbb{Z}_{[1,o]}$.

Proof. Line of sight visibility implies that for all $l \in \mathbb{Z}_{[1,o]}$

$$\ell(w_i, w_{i+1}) \cap (\mathcal{O}_l \oplus \mathcal{S}^S) = \emptyset,$$

where

$$\ell(w_i, w_{i+1}) = \{x \mid x = w_i + \gamma(w_{i+1} - w_i), \forall \gamma \in [0,1]\}$$

is the set of all states that lie on the line segment connecting the waypoints. Using the duality property of the Pontryagin difference [20], this means that

$$\begin{aligned}\ell(w_i, w_{i+1}) &\subseteq (\mathcal{O}_l \oplus \mathcal{S}^S)^C \\ &= \mathcal{O}_l^C \ominus \mathcal{S},\end{aligned}$$

which implies that

$$\ell(w_i, w_{i+1}) \oplus \mathcal{S} \subseteq \mathcal{O}_l^C.$$

Noting that the LHS defines the corridor connecting the two waysets, the result follows directly. ■

Theorem 3 (Robust Feasibility)**.** *If the waypoint placement problem has a feasible solution, then there exists a feasible trajectory from the initial state x_0 to $\mathcal{T}$ that passes through all of the waysets.*

Proof. From Problem 3 and [14], it is clear that the constraint-tightened inter-wayset problem is recursively feasible, given an initial feasible solution. Additionally, the waysets have been placed to guarantee the existence of such a feasible trajectory between any adjacent pair of waysets. By induction therefore, it is evident that there exists a feasible trajectory from the initial state to the target. ∎

Theorem 4 (Robust Finite-Time Completion)**.** *If the input weighting α is chosen such that*

$$H = 1 - \alpha \max_{d \in \mathcal{D}} \sum_{j=0}^{N-1} \|P(j)d\| > 0,$$

where $\mathcal{D}$ is the disturbance set, $P(j)$ is the disturbance-feedback constraint-tightening policy described in Section IIIB, and N is the maximum horizon length for the sub problem, the overall control problem is guaranteed to reach completion in $\mathcal{T}$ in at most

$$\sum_{i=1}^{p-1} \left\lfloor \max_{\zeta \in \mathcal{W}_i} \frac{J_i^*(\zeta)}{H} \right\rfloor$$

steps, where $J_i^*(\zeta)$ is the value function (19) *for an N-step trajectory commencing at $\zeta \in \mathcal{W}_i$ and terminating in $\mathcal{W}_{i+1}$.*

Proof. This is a straightforward extension of the finite-time completion result outlined in [14] and extended in [27] to handle disturbance feedback constraint tightening. ∎

V. Numerical Examples

Firstly, this section considers the problem of controlling a fixed-wing UAV modelled by a unit point mass moving in two dimensions subject to minimum and maximum speed constraints, in the presence of a wind disturbance and obstacles. Whilst minimum-speed constraints are non-convex in Cartesian coordinates, the robustness procedure developed in this paper allows a linear model to be used with tightening applied to compensate for model mismatch. Secondly, the problem of motion control of a quadrotor with nonlinear dynamics is considered. A linearized model is used to calculate the control inputs to ensure a safe motion between consecutive waysets. The robustness procedure introduced in this paper is used to handle the linearisation errors. Code used for simulations can be found at `https://dl.dropboxusercontent.com/u/18202212/Waysetcode.zip`.

A. Fixed-wing UAV Control

A fixed-wing UAV may be modelled by as a point mass moving in two dimensions with minimum speed constraints. Considering the dynamics of the point mass, its 1 Hz zero-order-hold discretized dynamics are represented by

$$\begin{bmatrix} s(k+1) \\ v(k+1) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} s(k) \\ v(k) \end{bmatrix} + \begin{bmatrix} 0.5 & 0 \\ 0 & 0.5 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} u(k) + d(k),$$

where $s(k) \in \mathbb{R}^2$ is the Cartesian position vector of the vehicle, $v(k) \in \mathbb{R}^2$ is the velocity, and the applied force vector is $u(k)$. A wind disturbance $d(k)$ of maximum magnitude $d_{\max}$ acts only on the velocity states of the UAV. The applied force and velocity must satisfy the constraints $\|u(k)\| \leq u_{\max}$ and $v_{\min} \leq \|v(k)\| \leq v_{\max}$ in the presence of the wind distubance. By considering the centripetal acceleration induced by the input force, it can be shown that this minimum velocity and maximum input constraints combine to enforce a minimum turning radius of

$$R_{\min} := \frac{v_{\min}^2}{u_{\max}}, \tag{21}$$

on the original continous-time system

Since a linear model with convex constraints will be used for control, it is necessary to modify the output of the controller to ensure that the minimum-velocity constraints are satisfied. Given the commanded input force $u(k)$, it is clear that $\bar{v}(k+1|k) = \bar{v}(k) + u(k)$. Then, if $\|\bar{v}(k+1|k)\| < v_{\min} + d_{\max}$, there is the chance of the UAV dropping below the minimum speed at the next time step with this input. To guarantee satisfaction of the minimum-speed constraints, it is necessary to instead apply a scaled version of the input

$$\hat{u} = \xi u, \tag{22}$$

which is illustrated in Fig. 3. The thick arrows on the figure show the scaled input and predicted future velocity respectively. The scaling factor ξ is chosen such that $\|v + \xi u\| = v_{\min} + d_{\max}$. This is a quadratic in ξ, which has solutions

$$\xi_i := \|u\|^{-2} \left(-u^T v + (-1)^i \sqrt{(u^T v)^2 - \|u\|^2 \left(\|v\|^2 - (v_{\min} + d_{\max})^2\right)} \right),$$

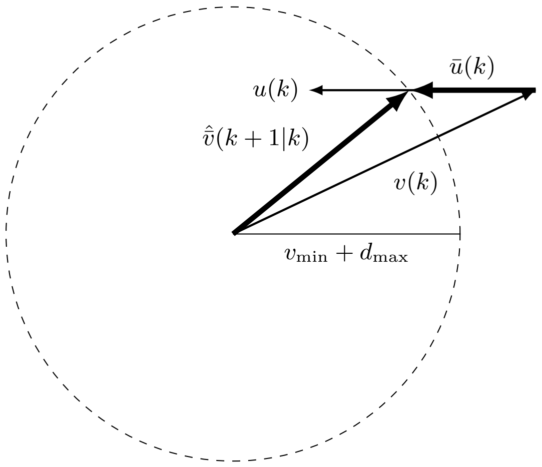


Fig. 3 Minimum velocity compensation

for $i \in \mathbb{Z}_{[0,1]}$. Then, choosing $\xi = \xi_{i^*}$, where

$$i^* := \arg\min_i |\xi_i|,$$

minimizes the anticipated change in velocity induced by (22). In the case where the $|v(k)| < v_{\min} + d_{\max}$ and $\|u\| = 0$, the modified input is selected to be

$$\bar{u} = (v_{\min} + d_{\max} - \|v\|)v/\|v\|.$$

In both cases, the resulting velocity prediction $\hat{\bar{v}}(k+1|k) = v(k) + \hat{u}(k)$ satisfies $\left\|\hat{\bar{v}}(k+1|k)\right\| = v_{\min} + d_{\max}$.

Modifying the input in this way introduces a disturbance into the system of maximum magnitude $v_{\min}+d_{\max}$, resulting in an overall disturbance magnitude of $v_{\min}+2d_{\max}$ when the wind disturbance is also considered.

For numerical simulation, the input limit $u_{\max} = 3$, as well as the velocity limits $v_{\max} = 4$ and $v_{\min} = 0.8$, are enforced. A wind disturbance is added to the system, with $\|d\| \leq 0.2$. The model UAV is controlled to a rectangular terminal set through a simple obstacle field using both the method developed in the previous sections in addition to a more complex mixed-integer controller using binary variables for collision avoidance and minimum-speed enforcement from [14]. The MPC cost function in both cases is given by

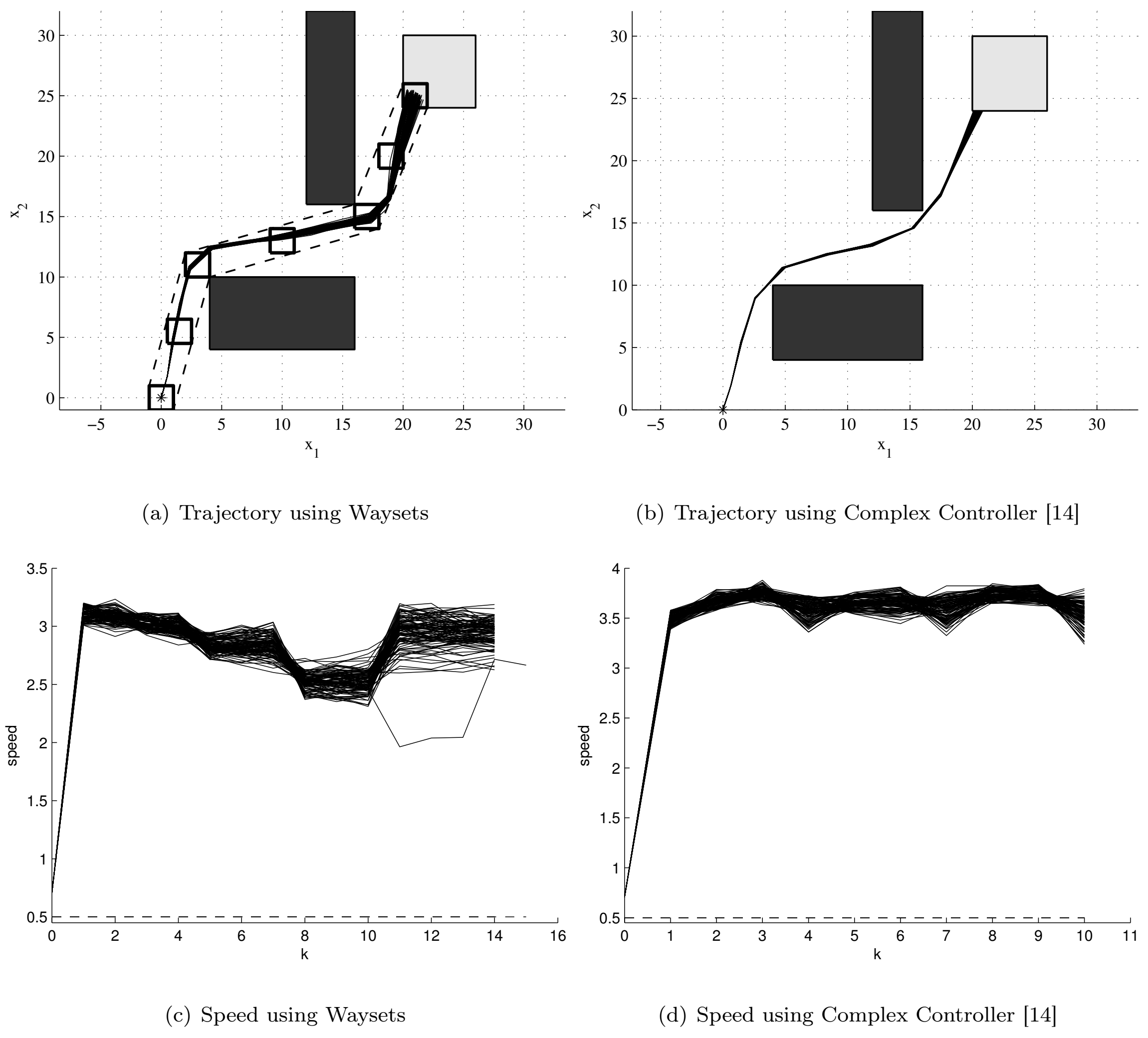
$$J(x(k), N) := \sum_{j=0}^{N-1} 1 + 0.01\,\|\bar{u}(k+j|k)\|,$$

where a 2-norm of the input is used to cost fuel consumption, assuming the actual UAV can be modelled as having a single vectored actuator. This results in a Mixed-Integer Second-Order Cone Program (MISOCP) to be solved at each MPC iteration, where integer variables are used solely to encode the horizon length and not for obstacle avoidance.

Seven square waysets are placed en route to the terminal set (six intermediate, one initial and one terminal) to guarantee robust four-step reachability according to Algorithm 1. A maximum prediction horizon of $N = 4$ steps is subsequently enforced for each inter-wayset problem. For the complex controller, obstacle avoidance is encoded explicitly with integer variables, over a prediction horizon of 32 steps. The minimum-velocity compensation is only needed for the wayset approach, since the MILP formulation encodes them explicitly. All norm constraints are approximated with 12-sided inner or outer polygon approximations as appropriate. A simplified optimal constraint tightening policy with a nilpotence constraint, as described in [23, 27], is used for each horizon length respectively to provide robustness. In addition, the obstacle sets need to be further enlarged for the complex controller, to ensure that the trajectory does not impinge on the corners of the obstacles.

Simulations over 100 disturbance realizations using the wayset-based predictive controller results in the trajectories and velocities shown in Figure 4. In Figs. 4(a)-4(b), obstacles are indicated as solid black boxes, the target set as a shaded box and the waysets are depicted by the white squares. The dashed line in Figs. 4(c)-4(d) represent the minimum velocity constraint. The comparison of computation times and costs are presented in Table 1. The terms $\overline{\Delta t_{\max}}$ and $(\Delta t_{\max})_{\max}$ represent the mean and maximum of the maximum solution time per timestep for each disturbance sequence. Simulations were performed using MATLAB, GUROBI, YALMIP [28] and the MPT Toolbox [29], on a 2.5 GHz iMac 12.1 running OS X 10.9.2.

Ths simulation results show that for this control problem, the wayset approach is on average approximately 301 times faster than the complex MILP approach. The contrast is even more stark when comparing the maximum times, where the wayset approach is 773 times faster than the MILP approach. The tradeoff is shown by the modest increase in average closed-loop costs, where the



(a) Trajectory using Waysets (b) Trajectory using Complex Controller [14]

(c) Speed using Waysets (d) Speed using Complex Controller [14]

Fig. 4 Fixed-Wing UAV simulation

Scheme	$\overline{\Delta t_{\max}}$	$(\Delta t_{\max})_{\max}$	Mean Cost
Wayset Controller	0.0054	0.0060	14.1052
Complex Controller [14]	1.6369	4.6506	10.1082

Table 1 Numerical Comparison

wayset approach is seen to cost approximately 1.4 times that of the complex MILP approach. This additional cost primarily arises from the extra highly-conservative tightening applied for minium-speed constraints; the cost increase is only 1.2 times that of the complex MILP approach when this tightening is relaxed, at the expense of robustness guarantees to the minimum-speed compensation. In either scenario, the computational advantage of the wayset approach clearly outweighs the extra cost incurred when considering real-time controller implementation. Indeed, the complex MILP controller could not be implemented in real time, since the sampling rate is 1 Hz. Finally, the closed loop cost may be improved by tuning the horizon length, number of waysets and wayset shape appropriately for a given control problem. This will form part of future investigations.

B. Quadrotor Motion Control With Nonlinear Dynamics

The quadrotor model used to generate the results in this subsection are based on the nonlinear model developed in [30, Section 4]. This model is linearized with respect to a hovering trim condition and discretized with the sampling rate of 5 Hz, giving a linear discrete-time system with 16 states and 4 inputs. The first three states correspond to x, y, and z positions of the quadrotor (m), the second three states are the linear velocities in the aforementioend directions (m/s). State variables 7, 8, and 9 are roll, pitch, and yaw angles (rad), respectvely, and states 10, 11, and 12 are the angular velocity of these angles (rad/s). States 13, 14, 15, and 16 are the revolutions of the propellers (rad/s). The control inputs are the pulse-amplitude modulated signal to the propeller motors, as described in [30, Section 4]. The linearization is carried at a hover position, i.e. all the states except for the propellers are zero with propellers rotating at 463.1 rad/s. The descritezed model obtained from a 5Hz sampling of the linearized system is given below

$$x(k+1) = Ax(k) + Bu(k) \tag{23}$$

where A and B are given in the appendix. The states and inputs are required to satisfy the operating constraints

$$\begin{aligned} |x| &\leq \begin{bmatrix} 5\mathbf{1}_3^T & 0.125\mathbf{1}_3^T & 0.0785 & 0.0785 & 3.1416 & 3.5 & 3.5 & 10 & 231.55\mathbf{1}_4^T \end{bmatrix}^T \\ |u| &\leq 50. \end{aligned}$$

The quadrotor is initialized at position $(0\,\text{m}, 0\,\text{m}, 1\,\text{m})$ in Cartesian coordinates, with all other states at the origin. The system is thus in a steady hover condition 1 m above the ground. Seven cubical waysets of side length 2.5 m are placed on a helical arc of radius 1 m and rise of 3.6×10^{-3} m per degree, traversing an angle of 45°. The system is required to transit through all waysets at a top speed of 0.25 m/s in each coordinate direction. The cost function used for each inter-wayset problem is chosen as

$$J_i(x(k), N) := \left[\sum_{j=0}^{N-1} 1 + 10^{-4} \|u(k+j|k)\|\right] + 500 \|x(k+N|k) - w_{i+1}\|,$$

which is a weighted sum of the time to completion and the 1-norm of the inputs (appropriate given the four independently actuated propellers), augmented with a term that enforces a strong weighting towards completion of each inter-wayset problem at the trim condition. The additional cost weighting is necessary in this case, since the linearization of the system is only valid for a small range of Euler angles about the trim condition. A maximum prediction horizon of 8 steps is chosen for each problem. Constraint tightening is added to further mitigate the impact of linearization error, treating this error as a disturbance. The disturbance polytope is given by

$$\mathcal{D} := \{d \mid |d| \leq \bar{d}\}$$

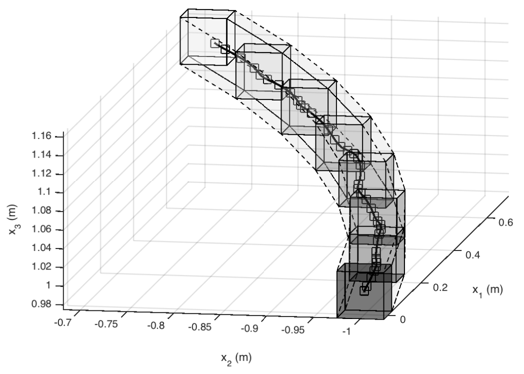
where

$$\bar{d} := \begin{bmatrix} 1 & 1 & 0 & 1 & 3 & 3 & 1 & 0 & 0 & 4 & 2 & 2 & 0 & 0 & 0 & 0 \end{bmatrix}^\top \times 10^{-3}.$$

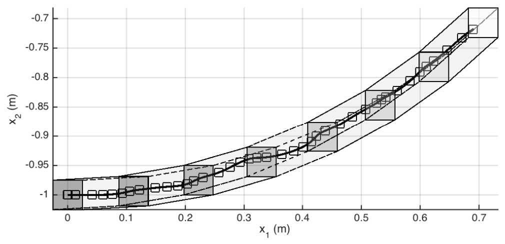
The control problem is simulated using the same computational platform as in the previous example. Fig. 5(a)-5(b) illustrates the resulting trajectory from two different viewpoints. The corridor is indicated by dashed lines. The Cartesian components of the quadrotor's velocity is shown in Fig. 5(c). It can be seen that the extra weighting added to the cost function causes the quadrotor to slow down slightly as it approaches each wayset. The wayset approach allows the quadrotor to be controlled in real time, since the maximum solution time per timestep for the MPC is 0.131 s, lower than the sampling period of 0.2 s.

VI. Conclusion

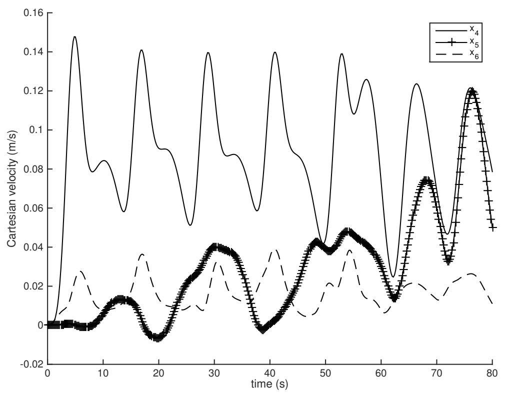
This paper has presented a novel approach using model predictive control with waysets for robust and safe motion control of unmanned aerial vehicles (UAVs) subject to operating constraints. The proposed scheme requires vastly fewer computational resources when compared to existing methods presented in the literature. Numerical simulations with a fixed wing UAV and Quadrotor have been used to demonstrate the applicability of the theoretical results presented in the paper.



(a) Trajectory in three dimensions



(b) Top view of trajectory



(c) Velocities

Fig. 5 Quadrotor simulation

Optimal wayset positioning and consideration of moving obstacles are immediate future research directions. Moreover, addressing the problem of robust and safe motion control of agents with nonliner motion models is of significance and will be addressed by extending the present work. In particular, it should be noted that the problem of travelling between each pair of consecutive waypoints is independent of the motion between other pairs. Thus, it is envisaged that different linearized models that approximate the motion between each consecutive pair of waysets can be considered. As a result, different MPC problems with different models can be solved to achieve a robust and safe motion of an agent with nonlinear dynamics towards the target.

Acknowledgments

The authors would like to thank Chris Manzie and Eric Kerrigan for their helpful suggestions. This research was supported by an Early-Career research grant from the School of Engineering at The University of Melbourne. I. Shames is supported by a McKenzie fellowship.

References

[1] Schouwenaars, T., Feron, E., de Moor, B., and How, J., "Mixed Integer Programming for Multi-vehicle Path Planning," in "Proceedings of the European Control Conference," , 2001, pp. 2603–2608.

[2] Bellingham, J., Richards, A., and How, J. P., "Receding horizon control of autonomous aerial vehicles," in "Proceedings of the American Control Conference," IEEE, Vol. 5, 2002, pp. 3741–3746, doi:10.1109/ACC.2002.1024509.

[3] Frazzoli, E., Dahleh, M. A., and Feron, E., "Real-time motion planning for agile autonomous vehicles," *Journal of Guidance, Control, and Dynamics*, Vol. 25, No. 1, 2002, pp. 116–129, doi:10.2514/2.4231.

[4] Yang, H. and Zhao, Y., "Trajectory planning for autonomous aerospace vehicles amid known obstacles and conflicts," *Journal of guidance, control, and dynamics*, Vol. 27, No. 6, 2004, pp. 997–1008, doi:10.2514/1.12514.

[5] Mattei, M. and Blasi, L., "Smooth flight trajectory planning in the presence of no-fly zones and obstacles," *Journal of guidance, control, and dynamics*, Vol. 33, No. 2, 2010, pp. 454–462, doi:10.2514/1.45161.

[6] Chamseddine, A., Zhang, Y., Rabbath, C. A., and Theilliol, D., "Trajectory Planning and Replanning Strategies Applied to a Quadrotor Unmanned Aerial Vehicle," *Journal of Guidance, Control, and Dynamics*, Vol. 35, No. 5, 2012, pp. 1667–1671, doi:10.2514/1.56606.

[7] Hoy, M., Matveev, A. S., and Savkin, A. V., "Algorithms for collision-free navigation of mobile robots in complex cluttered environments: a survey," *Robotica*, pp. 1–35, doi:10.1017/S0263574714000289.

[8] Goerzen, C., Kong, Z., and Mettler, B., "A survey of motion planning algorithms from the perspective of autonomous UAV guidance," *Journal of Intelligent and Robotic Systems*, Vol. 57, No. 1-4, 2010, pp. 65–100, doi:10.1007/s10846-009-9383-1.

[9] Betts, J. T., “Survey of numerical methods for trajectory optimization,” *Journal of guidance, control, and dynamics*, Vol. 21, No. 2, 1998, pp. 193–207.

[10] Mayne, D., Rawlings, J., Rao, C., and Scokaert, P., “Constrained model predictive control: Stability and optimality,” *Automatica*, Vol. 36, No. 6, 2000, pp. 789 – 814,
doi:10.1016/S0005-1098(99)00214-9.

[11] Maciejowski, J. M., *Predictive control with constraints*, Prentice Hall, Essex, England, 2002.

[12] Kuwata, Y., *Real-time Trajectory Design for Unmanned Aerial Vehicles using Receding Horizon Control*, Master’s thesis, Massachusetts Institute of Technology, 2003.

[13] Richards, A. and How, J., “Model predictive control of vehicle maneuvers with guaranteed completion time and robust feasibility,” in “Proceedings of the American Control Conference,” IEEE, Vol. 5, 2003, pp. 4034–4040,
doi:10.1109/ACC.2003.1240467.

[14] Richards, A. and How, J. P., “Robust variable horizon model predictive control for vehicle maneuvering,” *International Journal of Robust & Nonlinear Control*, Vol. 16, No. 7, 2006, pp. 333–351,
doi:10.1002/rnc.1059.

[15] Afonso, R. J. M., Galvao, R. K. H., and Kienitz, K. H., “Predictive control with trajectory planning in the presence of obstacles,” in “Control (CONTROL), 2012 UKACC International Conference on,” IEEE, 2012, pp. 508–514,
doi:10.1109/CONTROL.2012.6334682.

[16] Vitus, M. P., Pradeep, V., Hoffmann, G., Waslander, S. L., and Tomlin, C. J., “Tunnel-MILP: Path planning with sequential convex polytopes,” in “AIAA Guidance, Navigation, and Control Conference,” AIAA, 2008, pp. 1–13,
doi:10.2514/6.2008-7132.

[17] Vitus, M. P., Waslander, S. L., and Tomlin, C. J., “Locally optimal decomposition for autonomous obstacle avoidance with the tunnel-MILP algorithm,” in “47th IEEE Conference on Decision and Control,” IEEE, 2008, pp. 540–545,
doi:10.1109/CDC.2008.4739394.

[18] Afonso, R. J. M., Galvao, R. K. H., and Kienitz, K. H., “Waypoint Trajectory Planning in the Presence of Obstacles with a Tunnel-MILP approach,” in “European Control Conference (ECC),” IEEE, 2013, pp. 1390–1397.

[19] Kuwata, Y., Richards, A., and How, J., “Robust Receding Horizon Control using Generalized Constraint Tightening,” in “Proceedings of the 26th American Control Conference,” IEEE, 2007, pp. 4482–4487,

doi:10.1109/ACC.2007.4283000.

[20] Haralick, R., Sternberg, S. R., and Zhuang, X., "Image Analysis Using Mathematical Morphology," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, Vol. PAMI-9, No. 4, 1987, pp. 532–550,
doi:10.1109/TPAMI.1987.4767941.

[21] Richards, A., "Robust Model Predictive Control for Time-Varying Systems," in "44th IEEE Conference on Decision and Control and 2005 European Control Conference," , 2005, pp. 3747–3752,
doi:10.1109/CDC.2005.1582745.

[22] Goulart, P. J., Kerrigan, E. C., and Maciejowski, J. M., "Optimization over state feedback policies for robust control with constraints," *Automatica*, Vol. 42, No. 4, 2006, pp. 523 – 533,
doi:10.1016/j.automatica.2005.08.023.

[23] Shekhar, R. C. and Maciejowski, J. M., "Optimal constraint tightening policies for robust variable horizon model predictive control," in "Proceedings of the 51st IEEE Conference on Decision and Control (CDC)," , 2012, pp. 5170–5175,
doi:10.1109/CDC.2012.6426710.

[24] Dantzig, G. B. and Eaves, B. C., "Fourier-Motzkin elimination and its dual," *Journal of Combinatorial Theory, Series A*, Vol. 14, No. 3, 1973, pp. 288 – 297,
doi:10.1016/0097-3165(73)90004-6.

[25] Jones, C. N., Kerrigan, E. C., and Maciejowski, J. M., "Equality Set Projection: A new algorithm for the projection of polytopes in halfspace representation," Tech. rep., Department of Engineering, University of Cambridge, 2004. CUED/F-INFENG/TR.463.

[26] Choset, H., Lynch, K., Hutchinson, S., Kantor, G., Burgard, W., Kavraki, L., and Thrun, S., *Principles of robot motion: theory, algorithms, and implementations.*, MIT Press, Boston, 2005.

[27] Shekhar, R. C., *Variable Horizon Model Predictive Control: Robustness & Optimality*, Ph.D. thesis, University of Cambridge, 2012.

[28] Löfberg, J., "YALMIP : A Toolbox for Modeling and Optimization in MATLAB," in "Proceedings of the CACSD Conference," Taipei, Taiwan, 2004, pp. 284 – 289,
doi:10.1109/CACSD.2004.1393890.

[29] Kvasnica, M., Grieder, P., and Baotić, M., "Multi-Parametric Toolbox (MPT)," , 2004.

[30] Jiřinec, T., *Stabilization and control of unmanned quadcopter*, Master's thesis, Czech Technical University in Prague, 2011.

APPENDIX

Waypoint Generation and Feasible Maximum Horizon Length Calculation

Algorithm 1: Waypoint Generation

1. Take a sequence of cell centers points, $\bar{c}_1, \ldots, \bar{c}_{\bar{p}}$, and a maximum horizon length $\bar{N}$ as inputs.
2. Set the first point in the sequence as the center of waypoint w_0.
3. Pick the next waypoint to be the farthest point in the cell center sequence that satisfies (17)-(18).
4. Repeat the previous step until
   (a) the last waypoint is chosen to be $\bar{c}_{\bar{p}}$, i.e. a sequence of feasible waypoints are generated, or
   (b) there is no point in the cell center sequence that (17)-(18) hold for a previously chosen waypoint.

Algorithm 2: Calculating the Smallest Feasible Maximum Horizon Length $\bar{N}$

1. Take a sequence of cell centers points, $\bar{c}_1, \ldots, \bar{c}_{\bar{p}}$, as input.
2. Set the $\bar{N}$ to be equal to 1.
3. Apply Algorithm 1 to $\bar{c}_1, \ldots, \bar{c}_{\bar{p}}$ and the candidate $\bar{N}$.
4. If a sequence of feasible waypoints is generated then return $\bar{N}$ as the smallest feasible maximum horizon length. Otherwise, increment $\bar{N}$ by one and repeat the previous step.

Quadrotor Dynamics Parameters

The A and B matrices of (23) are given by

$A =$

$$\begin{bmatrix}
1 & 0 & 0 & 0.2 & 0 & 0 & 0 & -0.1962 & 0 & 0 & -0.0131 & 0 & 0.0001 & 0 & -0.0001 & 0 \\
0 & 1 & 0 & 0 & 0.2 & 0 & 0.1962 & 0 & 0 & 0.0131 & 0 & 0 & 0 & 0.0001 & 0 & -0.0001 \\
0 & 0 & 1 & 0 & 0 & 0.2 & 0 & 0 & 0 & 0 & 0 & 0 & 0.0001 & -0.0001 & 0.0001 & -0.0001 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & -1.9620 & 0 & 0 & -0.1962 & 0 & 0.0028 & 0 & -0.0028 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 1.9620 & 0 & 0 & 0.1962 & 0 & 0 & 0 & 0.0027 & 0 & -0.0027 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0.0009 & -0.0009 & 0.0009 & -0.0009 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0.2 & 0 & 0 & 0 & 0.0037 & 0 & -0.0037 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0.2 & 0 & -0.0037 & 0 & 0.0037 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0.2 & 0.0034 & 0.0034 & 0.0034 & 0.0034 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0.0278 & 0 & -0.0278 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & -0.0281 & 0 & 0.0281 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0.0260 & 0.0260 & 0.0260 & 0.0260 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.1353 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.1353 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.1353 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.1353
\end{bmatrix}$$

and

$$B = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0.0001 & -0.0001 & 0.0001 & -0.0001 \\ 0.0010 & 0 & -0.0010 & 0 \\ 0 & 0.0010 & 0 & -0.0010 \\ 0.0008 & -0.0008 & 0.0008 & -0.0008 \\ 0 & 0.0019 & 0 & -0.0019 \\ -0.0020 & 0 & 0.0020 & 0 \\ 0.0018 & 0.0018 & 0.0018 & 0.0018 \\ 0 & 0.0256 & 0 & -0.0256 \\ -0.0259 & 0 & 0.0259 & 0 \\ 0.0239 & 0.0239 & 0.0239 & 0.0239 \\ 0.6053 & 0 & 0 & 0 \\ 0 & 0.6053 & 0 & 0 \\ 0 & 0 & 0.6053 & 0 \\ 0 & 0 & 0 & 0.6053 \end{bmatrix}.$$

The units in the B matrix have been omitted for brevity.