AIAA Guidance, Navigation, and Control Conference and Exhibit
15 - 18 August 2005, San Francisco, California

AIAA 2005-6079

Robust Constrained Receding Horizon Control for Trajectory Planning

Yoshiaki Kuwata*, Tom Schouwenaars†, Arthur Richards‡, and Jonathan How§

Department of Aeronautics and Astronautics
Massachusetts Institute of Technology

This paper presents a receding horizon controller (RHC) that can be used to design trajectories for an aerial vehicle operating in an environment with disturbances. Various constraints are imposed in the problem, such as turning rate limits and bounds on the vehicle speed, and target regions and no-fly zones are included in the environment. The proposed algorithm modifies these constraints to ensure that the on-line RHC optimization remains feasible even when the vehicle is acted upon by unknown, but bounded, disturbances. The approach uses a robust control invariant admissible set as a terminal set that does not need to be a target set of the overall guidance problem. This result extends previous work in two ways: the vehicle is guaranteed to remain safe under the influence of disturbances; and much longer robust trajectories can be constructed on-line. The full algorithm is demonstrated in several numerical simulations.

Keywords: Receding Horizon Control, Robust Constrained Optimization, Invariant set

I. Introduction

Model Predictive Control (MPC) or Receding Horizon Control (RHC) has recently drawn attention from a variety of fields because it can systematically handle constraints and multivariable systems.[1,2] The basic approach is to use a model of the system to predict the future behavior and generate control inputs that meet the constraints and optimize the performance objectives. This optimization is repeated on-line as the vehicle maneuvers and new measurements about the vehicle states and the environment are obtained. A key point is that these optimization based controllers can operate close to the constraint boundaries[3] to obtain better performance than traditional approaches. However, as a result, small disturbances could drive the system into an infeasible region, so it is important to account for external disturbances or inherent discrepancies between the model and the real process in a systematic manner.

Several Robust MPC approaches have been proposed in the past decade to account for these disturbances. Min-max based approaches[4–6] minimize a performance index while guaranteeing constraint satisfaction under worst-case disturbance. The main disadvantage of this approach is that it is computationally intense, and it is not suitable for on-line optimization. Computationally tractable (i.e., solvable in polynomial time) approaches have been proposed using linear matrix inequalities (LMIs).[7–9] The stability and the robustness of these LMI-based controllers have been proven using convex optimization techniques. Robust optimization[10] is also used to ensure that the solution is robust to any uncertain data realization. The *Constraint tightening* approach proposed in Ref. [11] is based on the idea of increasing the robustness of the controller by retaining a margin in the constraints for future feedback action, which becomes available to the MPC optimization as time progresses. Constraint tightening typically uses invariant sets, and the set operation tools[13,14] are

* Research Assistant, MIT Dept. of Aeronautics and Astronautics, kuwata@mit.edu
† Research Assistant, MIT Dept. of Aeronautics and Astronautics, toms@mit.edu
‡ Lecturer, University of Bristol, Dept. of Aerospace Engineering, Arthur.Richards@bristol.ac.uk
§ Associate Professor, MIT Dept. of Aeronautics and Astronautics, Associate Fellow of AIAA, jhow@mit.edu. Room 33-326, 77 Mass. Ave., Cambridge, MA 02139.

particularly helpful. The approach has been recently extended to reduce the conservatism of the controller by using the maximal robust control invariant admissible set as a terminal set.[15]

An extension is required when applying this approach to a class of problems, called "target reach," that are of particular interest for the guidance of unmanned aerial vehicles and the trajectory/activity planning for autonomous rovers. For target reach problems, it is required that a terminal target set be reached by the trajectory while meeting various constraints on the way. The algorithm presented in Ref. [15] assumes that the target set is reachable over the planning horizon, but it may not be computationally tractable to solve the large optimization problem on-line that is necessary to design a long trajectory. One approach is to terminate the optimization when a time limit is reached. When the algorithm has a feasible candidate solution, the optimization could be stopped at any time, but this will degrade the performance of the closed-loop system, and it is still necessary to find an initial feasible solution. If the controller is required to design a long trajectory from the initial states that are not known *a priori,* then finding this initial feasible solution on-line could be difficult.

This paper extends the constraint tightening approach[15] to address these computational issues. In particular, the new algorithm presented in this paper does not explicitly require that the system states be able to reach the target set over the planning horizon. Instead, the controller only requires that the states can be driven to a robust control invariant set, which can be updated as the system evolves. This approach also represents an extension of the concept of a *basis state* in which the system can safely remain for an indefinite period of time.[16–18] Note that the robust control invariant set used in this paper does not need to be associated directly with the target set, which allows for very short planning horizons. The approach is combined with a *cost-to-go* function, which can provide a good estimate of the path beyond the planning horizon to the goal.[21] As a result, the algorithm can be used to solve much longer robust paths than the approach in Ref. [15].

The paper is organized as follows. Section II extends the algorithm presented in Ref. [15] so that the target set is not necessarily reached. Section III presents several simulation results, and final remarks are presented in Section IV.

II. Robust Constrained RHC Algorithm

This section presents a robust constrained RHC algorithm that is based on tightening constraints.[11,15,19] The problem of interest has the overall goal of **reaching the target set** while robustly **maintaining feasibility**. However, in order to maintain the feasibility, reaching the overall goal could be aborted. The algorithm relaxes the constraints that the target set must be reached in the planning horizon, allowing the controller to use a short planning horizon. The computational burden then becomes less severe even when the target set is far. A cost-to-go function provides a good estimate of the remainder of the path to reach the target, even in a complicated environment. Additional constraints are added that require that *some* robust control invariant admissible set $\mathcal{R}$ is reachable over the short planning horizon. This ensures that the feasibility of the optimization at time k implies the feasibility at the next time $k+1$, resulting in the robust feasibility.

A. Problem Statement

The linear time invariant system dynamics are described as follows.

$$\begin{aligned} \boldsymbol{x}(k+1) &= A\boldsymbol{x}(k) + B\boldsymbol{u}(k) + \boldsymbol{w}(k) && (1)\\ \boldsymbol{y}(k) &= C\boldsymbol{x}(k) + D\boldsymbol{u}(k) \in \mathcal{Y} && (2)\\ \boldsymbol{w}(k) &\in \mathcal{W} && (3) \end{aligned}$$

where $\boldsymbol{x}(k) \in \mathcal{R}^{N_x}$ is the state vector, $\boldsymbol{u}(k) \in \mathcal{R}^{N_u}$ is the input vector, and $\boldsymbol{w}(k) \in \mathcal{R}^{N_x}$ is the disturbance vector. The disturbance $\boldsymbol{w}(k)$ is random but lies in the bounded set $\mathcal{W}$, which is assumed to be known. Eq. 2 captures the constraints on the states, the inputs, or combinations of both by using the general output vector $\boldsymbol{y}(k) \in \mathcal{R}^{N_y}$.

The general objective function takes the form

$$J = \sum_{k=0}^{\infty} l(\boldsymbol{u}(k), \boldsymbol{x}(k), \mathcal{X}_F) \quad (4)$$

where $l(\cdot)$ is a stage cost function and $\mathcal{X}_F$ is a target set into which the state $\boldsymbol{x}$ is to be driven. In the MPC framework, the optimization is performed over a finite horizon; then, the first control input is executed; the new state is measured and the new optimization is repeated until the system reaches the target set.

B. Algorithm

MPC solves the optimization problem on-line as the new information on the states becomes available. The control inputs are developed for a finite horizon of N steps, where the prediction of a value at time $(k+j)$ made at time k is denoted by indices $(k+j|k)$. The MPC optimization problem $\mathrm{P}(\boldsymbol{x}(k), \mathcal{Y}, \mathcal{W})$ at time k is defined as:

$$J^* = \min_{\boldsymbol{u}(\cdot), \mathcal{S}(k)} \left\{ \sum_{j=0}^{N} l\big(\boldsymbol{u}(k+j|k), \boldsymbol{x}(k+j|k), \mathcal{X}_F\big) + f\big(\boldsymbol{x}(k+N+1|k), \mathcal{X}_F\big) \right\} \tag{5}$$

subject to

$$\begin{aligned}
\boldsymbol{x}(k+j+1|k) &= A\boldsymbol{x}(k+j|k) + B\boldsymbol{u}(k+j|k), \quad j = 0, \ldots, N && (6)\\
\boldsymbol{y}(k+j|k) &= C\boldsymbol{x}(k+j|k) + D\boldsymbol{u}(k+j|k) \ \in \mathcal{Y}(j), \quad j = 0, \ldots, N && (7)\\
\boldsymbol{x}(k|k) &= \boldsymbol{x}(k) && (8)\\
\boldsymbol{x}(k+N+1|k) &\in \mathcal{S}(k) && (9)
\end{aligned}$$

The following subsections elaborate the output constraints $\mathcal{Y}$, the safety set $\mathcal{S}$, and the terminal penalty $f(\cdot)$.

1. Output Constraints

The predicted output $\boldsymbol{y}(k+j|k)$ lies in a different set $\mathcal{Y}(j)$ at each prediction step j. This set $\mathcal{Y}(j)$ is constructed by tightening the original set $\mathcal{Y}$ in the following manner.

At time $k+1$, a feasible solution $\boldsymbol{u}^*(k+j|k), \boldsymbol{x}^*(k+j|k), \boldsymbol{y}^*(k+j|k)$ to the problem $\mathrm{P}(\boldsymbol{x}(k), \mathcal{Y}, \mathcal{W})$ is available. Also, the measurement of the disturbance realization $\boldsymbol{w}(k)$ is available. Assume the candidate solution to the problem $\mathrm{P}(\boldsymbol{x}(k+1), \mathcal{Y}, \mathcal{W})$ takes the form

$$\begin{aligned}
\hat{\boldsymbol{u}}(k+j+1|k+1) &= \boldsymbol{u}^*(k+j+1|k) + K(j)L(j)\boldsymbol{w}(k), \quad j = 0, \ldots, N-1 \\
\hat{\boldsymbol{x}}(k+j+1|k+1) &= \boldsymbol{x}^*(k+j+1|k) + L(j)\boldsymbol{w}(k), \quad j = 0, \ldots, N && (10)\\
\hat{\boldsymbol{u}}(k+N+1|k+1) &= \kappa(\hat{\boldsymbol{x}}(k+N+1|k+1)) \\
\hat{\boldsymbol{x}}(k+N+2|k+1) &= A\boldsymbol{x}(k+N+1|k+1) + B\boldsymbol{u}(k+N+1|k+1) && (11)
\end{aligned}$$

The control policy $K(j)$ rejects the effect of the disturbance $\boldsymbol{w}(k)$. In order for $\hat{\boldsymbol{x}}$ and $\hat{\boldsymbol{u}}$ to satisfy the dynamics constraints Eq. 6,

$$\begin{aligned}
\hat{\boldsymbol{x}}(k+j+2|k+1) &= A\hat{\boldsymbol{x}}(k+j+1|k+1) + B\hat{\boldsymbol{u}}(k+j+1|k+1) \\
&= A\boldsymbol{x}^*(k+j+1|k) + B\boldsymbol{u}^*(k+j+1|k) \\
&\quad + (A+BK(j))L(j)\boldsymbol{w}(k) \\
&= \boldsymbol{x}^*(k+j+2|k) + L(j+1)\boldsymbol{w}(k)
\end{aligned}$$

Thus, it is required that

$$\begin{aligned}
L(0) &= I \\
L(j+1) &= (A+BK(j))L(j), \quad j = 0, \ldots, N-1 && (12)
\end{aligned}$$

Likewise, in order to satisfy the output constraints Eq. 7

$$\begin{aligned}
\hat{\boldsymbol{y}}(k+j+1|k+1) &= \boldsymbol{y}^*(k+j+1|k) + (C+DK(j))L(j)\boldsymbol{w}(k) \\
\boldsymbol{y}^*(k+j+1|k) &\in \mathcal{Y}(j+1) \\
\hat{\boldsymbol{y}}(k+j+1|k+1) &\in \mathcal{Y}(j)
\end{aligned}$$

it is sufficient to tighten the constraint set as

$$\begin{aligned} \mathcal{Y}(j+1) &= \mathcal{Y}(j) \sim (C + DK(j))L(j)\mathcal{W} \\ \mathcal{Y}(0) &= \mathcal{Y} \end{aligned} \tag{13}$$

where the operator $\sim$ denotes the Pontryagin difference that has the following property.[20]

$$\boldsymbol{a} \in (\mathcal{A} \sim \mathcal{B}),\, \boldsymbol{b} \in \mathcal{B} \quad \Rightarrow \quad \boldsymbol{a} + \boldsymbol{b} \in \mathcal{A}$$

2. Safety Set

The set $\mathcal{S}(k)$ in Eq. 9 is an invariant set, and is called a *safety* set that meets the following conditions. At the terminal step, the candidate solution in Eq. 11 has a relation

$$\hat{\boldsymbol{x}}(k+N+2|k+1) = A\hat{\boldsymbol{x}}(k+N+1|k+1) + B\kappa(\hat{\boldsymbol{x}}(k+N+1|k+1))$$

Let $\mathcal{R}$ denote a robust control invariant admissible set,[13] *i.e.,*

$$\begin{aligned} & \boldsymbol{x} \in \mathcal{R} \\ \Rightarrow \quad & \exists \kappa(\boldsymbol{x}) \ \text{ s.t. } A\boldsymbol{x} + B\kappa(\boldsymbol{x}) + L(N)\boldsymbol{w} \in \mathcal{R}, \ \ \forall \boldsymbol{w} \in \mathcal{W} \\ & \qquad\qquad C\boldsymbol{x} + D\kappa(\boldsymbol{x}) \in \mathcal{Y}(N) \end{aligned}$$

Then,

$$\begin{aligned} & \boldsymbol{x}^*(k+N+1|k) \in \mathcal{R} \sim L(N)\mathcal{W} \\ \Rightarrow \quad & \hat{\boldsymbol{x}}(k+N+1|k+1) \in \mathcal{R} \\ \Rightarrow \quad & \hat{\boldsymbol{x}}(k+N+2|k+1) + L(N)\boldsymbol{w} \in \mathcal{R}, \ \forall \boldsymbol{w} \in \mathcal{W} \\ \Rightarrow \quad & \hat{\boldsymbol{x}}(k+N+2|k+1) \in \mathcal{R} \sim L(N)\mathcal{W} \end{aligned}$$

Therefore, if we define the safety set $\mathcal{S}$ as

$$\mathcal{S} \equiv \mathcal{R} \sim L(N)\mathcal{W}$$

then, $\boldsymbol{x}^*(k+N+1|k) \in \mathcal{S}$ at time k implies $\hat{\boldsymbol{x}}(k+N+2|k+1) \in \mathcal{S}$ at time $k+1$, satisfying the terminal state constraint Eq. 9.

As shown in Eq. 5, the set $\mathcal{S}$ itself is a decision variable that the on-line optimization solves for. Theoretically, using the maximal robust control invariant admissible set $\mathcal{C}_\infty$ as a terminal set, where $\mathcal{C}_\infty$ is the greatest feasible invariant set under any nonlinear feedback $\kappa(\boldsymbol{x})$, provides a larger set of initial states from which the optimization has a feasible solution. However, the calculation of the maximal robust control invariant set could be very computationally intensive, even for off-line computation, unless the problem setup is very simple (*e.g.,* double integrator with a few constraints). It is usually not feasible to precalculate the exact maximal robust control invariant admissible set and use it in the on-line MPC optimization. In such cases, finding a good robust control invariant set on-line is crucial to maintaining feasibility and achieving a good performance.

The proposed approach parameterizes the invariant set and solves for a simple robust control invariant admissible set $\mathcal{R}(k)$ in the optimization at time k. The invariant set could be parameterized in several ways, depending on the dynamics of the vehicle, and Section III addresses this issue in more detail. The safety constraint is written as

$$\boldsymbol{x}^*(k+N+1|k) \in \mathcal{S}(k) \equiv \mathcal{R}(k) \sim L(N)\mathcal{W} \tag{14}$$

Note that $\boldsymbol{x}^*(k+N+1|k) \in \mathcal{S}(k)$ at time k implies $\hat{\boldsymbol{x}}(k+N+2|k+1) \in \mathcal{S}(k)$ at time $k+1$, with the same argument.

3. Terminal Penalty

The algorithm uses the target set $\mathcal{X}_F$ in the objective function, and there is no requirement that the target set is reached. The function $f(\cdot)$ is the terminal penalty that represents the distance from the safety set to the goal $\mathcal{X}_F$. Hence, minimization will tend to drive the system to the goal if possible. This function $f(\cdot)$ is called a cost-to-go function in the receding horizon framework, and one simple example is a two-norm distance between the safety set and the goal. However, choosing a good cost-to-go function could significantly enhance the performance of the planning system,[21] especially when the operating environment is complicated. Simulation results in Subsection B highlight the significance of this cost-to-go function.

4. Algorithm Summary

The algorithm is summarized as follows.

Robust Safe but Knowledgeable (RSBK) Algorithm:

0. Compute
   – the constraint sets $\mathcal{Y}(j)$ through Eqs. 12 and 13,
   – a cost map that can be used to evaluate the cost-to-go function $f(\cdot)$
1. At time k, solve the optimization problem Eqs. 5 to 8
2. Apply control $\boldsymbol{u}(k) = \boldsymbol{u}^*(k|k)$ from the optimal sequence to the system Eq. 1
3. Increment k. Go to Step 1

C. Theorem

Theorem. (*Robust Feasibility*) The RSBK algorithm keeps the states in a feasible region while satisfying all of the constraints under the action of a bounded disturbance (Eq. 3), if the first optimization is feasible.

Proof. The proof is based on recursion. Once a feasible solution to the problem $\mathrm{P}(\boldsymbol{x}(k), \mathcal{Y}, \mathcal{W})$ is obtained at time k, a candidate solution to $\mathrm{P}(\boldsymbol{x}(k+1), \mathcal{Y}, \mathcal{W})$ at time $k+1$ can be constructed from Eqs. 10 and 11. This candidate solution satisfies, under the bounded disturbance $\boldsymbol{w}(k) \in \mathcal{W}$, all the output constraints $\mathcal{Y}(j)$ at time $k+1$, and the terminal state $\hat{\boldsymbol{x}}(k+N+2|k+1)$ lies in the set $\mathcal{S}(k)$. Thus, feasibility at time k guarantees feasibility at time $k+1$, and if the first optimization $\mathrm{P}(\boldsymbol{x}(0), \mathcal{Y}, \mathcal{W})$ is feasible, then all the future optimizations will be feasible. ∎

D. Remarks

1. The algorithm does not require that the target region $\mathcal{X}_F$ is reachable over the planning horizon N. The horizon N could be very short, resulting in a computationally fast algorithm.
2. In order to recursively prove the robust feasibility, the algorithm requires the existence of the initial feasible solution. However, because the horizon length N is much shorter than in previous algorithms, this approach can find an initial feasible solution much faster.
3. In contrast to the nominal safety approach,[16] the algorithm presented here never fails to find a feasible solution under bounded disturbances.
4. Invariance of $\mathcal{S}(k)$ does not mean that all the states over the planning horizon are in $\mathcal{S}(k)$: the system could be driven to $\mathcal{S}(k)$ in N steps. It is generally driven towards the target set $\mathcal{X}_F$ by the cost function.
5. If the candidate control $K(j)$ is nilpotent so that $L(N) = 0$, then the set $\mathcal{R}(k) = \mathcal{S}(k)$ is a *nominal control invariant set*.
6. The number of control variables is the same as the nominal MPC algorithm. Furthermore, by over-bounding the Pontryagin difference operation in Eqs. 13 and 14, the algorithm will have the same number of constraints.
7. This is an **anytime algorithm**, that is, the optimization can be stopped at anytime. This follows because a candidate feasible solution can be always constructed from the previous feasible (not necessarily optimal) solution.

III. Simulation Results

This section presents several simulation results that highlight the extensions in this new RHC algorithm. The simulation uses a rotorcraft UAV, but the algorithm easily extends to other vehicles, such as fixed-wing UAVs.

A. Dynamics model

The rotorcraft dynamics can be approximated by a double integrator with constraints on speed and acceleration.

$$\begin{bmatrix} \boldsymbol{r}(k+1) \\ \boldsymbol{v}(k+1) \end{bmatrix} = A \begin{bmatrix} \boldsymbol{r}(k) \\ \boldsymbol{v}(k) \end{bmatrix} + B\boldsymbol{a}(k) + \boldsymbol{n}(k) \tag{15}$$

$$\text{with} \qquad A = \begin{bmatrix} I_2 & \Delta t\, I_2 \\ O_2 & I_2 \end{bmatrix}, \qquad \text{and} \qquad B = \begin{bmatrix} \frac{(\Delta t)^2}{2} I_2 \\ \Delta t I_2 \end{bmatrix} \tag{16}$$

where $\boldsymbol{r}$, $\boldsymbol{v}$, and $\boldsymbol{a}$ are the position, velocity, and acceleration vector respectively. I_2 and O_2 express an identity matrix and a zero matrix of size 2 respectively. The disturbance $\boldsymbol{n}(k)$ enters as a disturbance on the acceleration, and can be written as

$$\begin{aligned} \boldsymbol{n}(k) &= B\boldsymbol{w}(k) \\ \boldsymbol{w}(k) &\in \mathcal{W} = \left\{ w \in \mathcal{R}^2 \,|\, \|\boldsymbol{w}\|_\infty \leq w_{\max} \right\} \end{aligned} \tag{17}$$

1. *Output constraints*

The general output constraints include the obstacle avoidance

$$[I_2,\ O_2]\boldsymbol{x}(k+j|k) \notin \mathcal{O}$$

where $\mathcal{O} \subset \mathcal{R}^2$ expresses no-fly zones that the vehicle is not allowed to enter, the maximum speed

$$\|[O_2,\ I_2]\boldsymbol{x}(k+j|k)\|_2 \leq v_{\max}, \tag{18}$$

and the maximum acceleration command

$$\|\boldsymbol{a}(k+j|k)\|_2 \leq a_{\max} \tag{19}$$

where $v_{\max}$ and $a_{\max}$ are the maximum speed and acceleration for the rotorcraft. Because the system (A, B) is a combination of two independent double integrators, a two-step nilpotent controller K for this system is analytically calculated from $(A+BK)^2 = 0$, which produces

$$K = \left[-\frac{1}{\Delta t^2} I_2,\ \ -\frac{3}{2\Delta t} I_2 \right]. \tag{20}$$

Performing constraint tightening (Eq. 13) gives the following constraint set[15,22]

$$[I_2,\ O_2]\boldsymbol{x}(k+j|k) \notin \mathcal{O} \oplus \alpha(j)\mathcal{W} \tag{21}$$

$$\|[O_2,\ I_2]\boldsymbol{x}(k+j|k)\|_2 \leq v_{\max} - \beta(j) \tag{22}$$

$$\|\boldsymbol{a}(k+j|k)\|_2 \leq a_{\max} - \gamma(j) \tag{23}$$

where

$$\begin{aligned} \alpha(0) &= 0 \\ \alpha(1) &= \|[1\ 0\ 0\ 0]B\|_1 \\ \alpha(j) &= \alpha(1) + \|[1\ 0\ 0\ 0]LB\|_1, \quad j \geq 2 \\ \beta(0) &= 0 \\ \beta(1) &= \sqrt{2}\,\|[0\ 0\ 1\ 0]B\|_1 w_{\max} \\ \beta(j) &= \beta(1) + \sqrt{2}\,\|[0\ 0\ 1\ 0]LB\|_1 w_{\max}, \quad j \geq 2 \\ \gamma(0) &= 0 \\ \gamma(1) &= \sqrt{2}\,\|[1\ 0]KB\|_1 w_{\max} \\ \gamma(j) &= \gamma(1) + \sqrt{2}\,\|[1\ 0]KLB\|_1 w_{\max}, \quad j \geq 2 \end{aligned} \tag{24}$$

and the operator $\oplus$ denotes the Minkowski summation that is defined as follows.

$$\mathcal{A} \oplus \mathcal{B} = \{a + b \mid a \in \mathcal{A} \text{ and } b \in \mathcal{B}\}$$

Note that Eq. 21 expands the no-fly zones to guarantee robust feasibility. These non-convex constraints are implemented using Mixed-integer Linear Programming. The coefficient $\sqrt{2}$ appears when performing the Pontryagin difference between a two-norm bounded set (Eq. 18 or 19) and the infinite-norm bounded disturbance set $\mathcal{W}$ (Eq. 17). This is because $\mathcal{W}$ has the maximum magnitude of the length $\sqrt{2}w_{\max}$ in the diagonal directions.

2. Terminal constraints

In order to reduce the computation load of the on-line optimization, it is desirable that a simple set is used as $\mathcal{S}(k)$. For a rotorcraft UAV, we use hovering states[16] as a robust control invariant admissible set $\mathcal{R}(k)$:

$$\begin{aligned} & \boldsymbol{x} \in \mathcal{R}(k) \\ \Rightarrow \quad & \exists \kappa(\boldsymbol{x}) \ \text{ s.t. } A\boldsymbol{x} + B\kappa(\boldsymbol{x}) + L(N)\boldsymbol{w} \in \mathcal{R}(k), \quad \forall \boldsymbol{w} \in \mathcal{W} \\ & \qquad\qquad C\boldsymbol{x} + D\kappa(\boldsymbol{x}) \in \mathcal{Y}(N) \end{aligned}$$

Note that in this example, $L(N) = 0$ when $N \geq 2$, due to the nilpotency of the controller K. This allows us to use a nominal control invariant set as $\mathcal{S}(k)$ with a tightened constraint set, leading to a very simple parametrization. The tightened constraint set $\mathcal{Y}(N)$ corresponds to $\alpha(N), \beta(N), \gamma(N)$ in Eqs. 24. The invariance of the set $\mathcal{S}(k)$ is guaranteed by imposing the following hovering constraints in the optimization.

$$\boldsymbol{x}(k + N + 2|k) = \boldsymbol{x}(k + N + 1|k) \notin \mathcal{O}$$

Fixed wing aircraft cannot use this hovering states as a terminal set due to its minimum speed bound. In such case, one simple parameterized invariant set that can be used to generate the terminal constraints is a loitering circle with the minimum turning radius.[16]

B. Results

The following values are used in the simulation.

- $\Delta t = 2.6\,\text{second}$
- $N = 6$
- $v_{\max} = 0.5\,\text{m/s}$
- $a_{\max} = 0.17\,\text{m/s}^2$

1. Comparison with Nominal Safety Approach

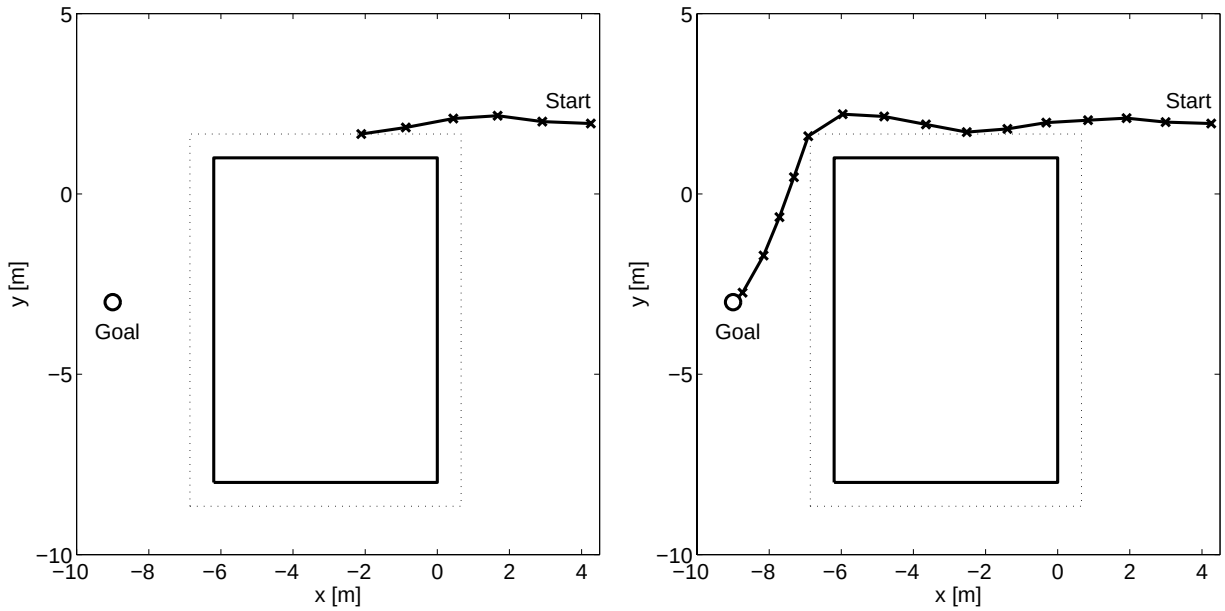
Figure 1 compares the two approaches that maintain feasibility of the MPC optimization by using an invariant set approach. The first approach uses nominal MPC with a nominal invariant set.[16] In this approach, a feasible solution to the optimization problem at time $k + 1$ is constructed by combining: 1) the trajectory portion constructed at time k that was not executed; and 2) an extra time step in the nominal invariant set in which the previous trajectory ends. However, the optimization could go infeasible if the vehicle does not reach the state that was predicted in the previous time step due to disturbance actions.

Figure 1(a) shows a trajectory generated with the nominal safety approach. The fifth optimization generates a trajectory that lies near the obstacle boundary. At the next time step, the vehicle was pushed into the infeasible region, and the optimization cannot return a feasible solution.

This issue is successfully resolved by the robust approach presented in this paper. Figure 1(b) shows that the vehicle can reach the goal in spite of the disturbance acting on the system. The constraints were tightened in such a way that no matter where the vehicle reaches after one time step, the feedback correction is possible to maintain feasibility of the MPC optimizations.

2. Change in the Environment

The next example demonstrates the performance improvement achieved by the different choice of the terminal penalty function $f(\cdot)$. We have previously developed an algorithm[21,23,24] that uses an intelligent cost-to-go



(a) Nominal trajectory design – becomes infeasible

(b) Robust trajectory design – maintains feasibility

Figure 1: Trajectories generated by the nominal controller and the robust controller. The vehicle starts in the right, and the goal is marked with ∘.

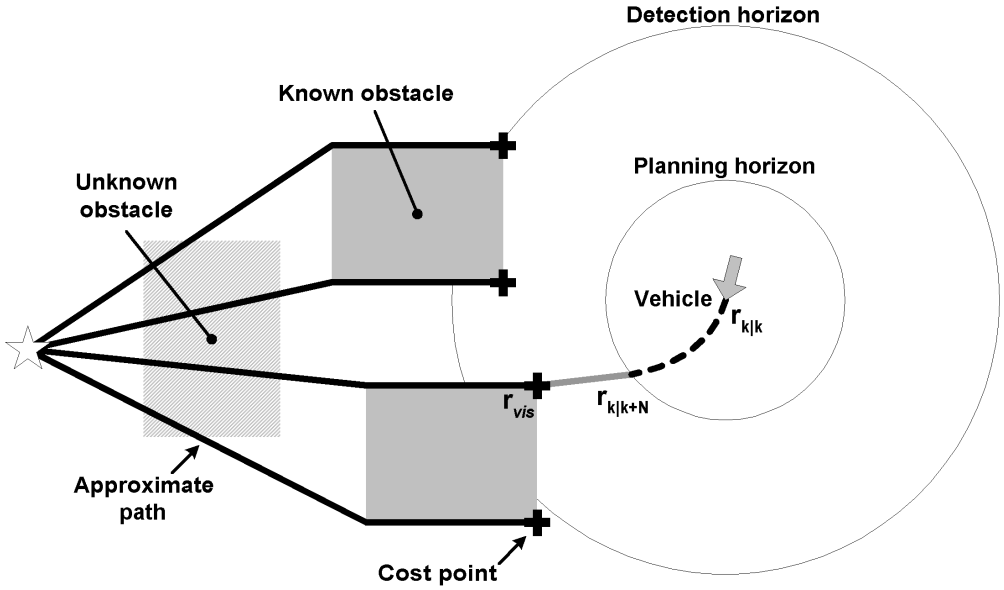
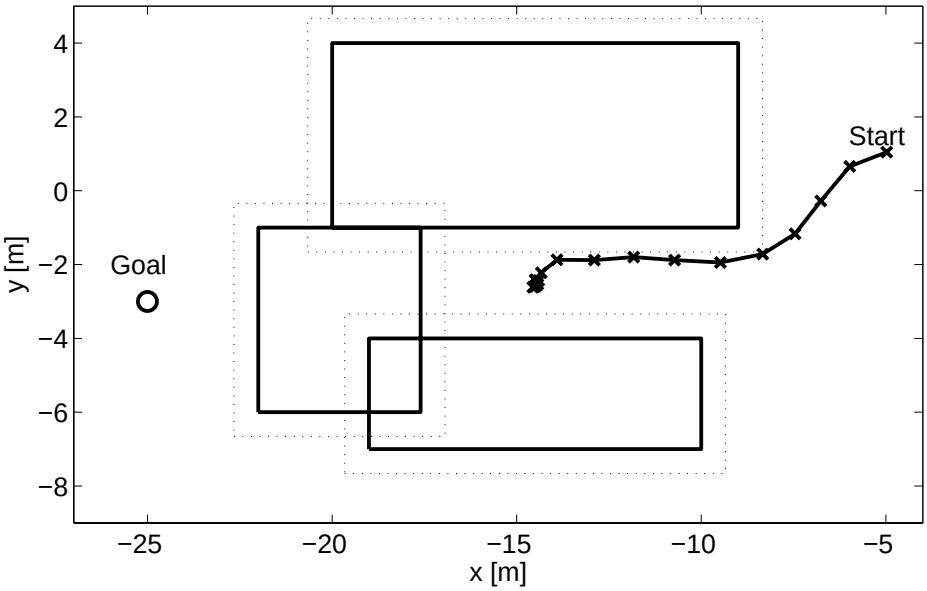
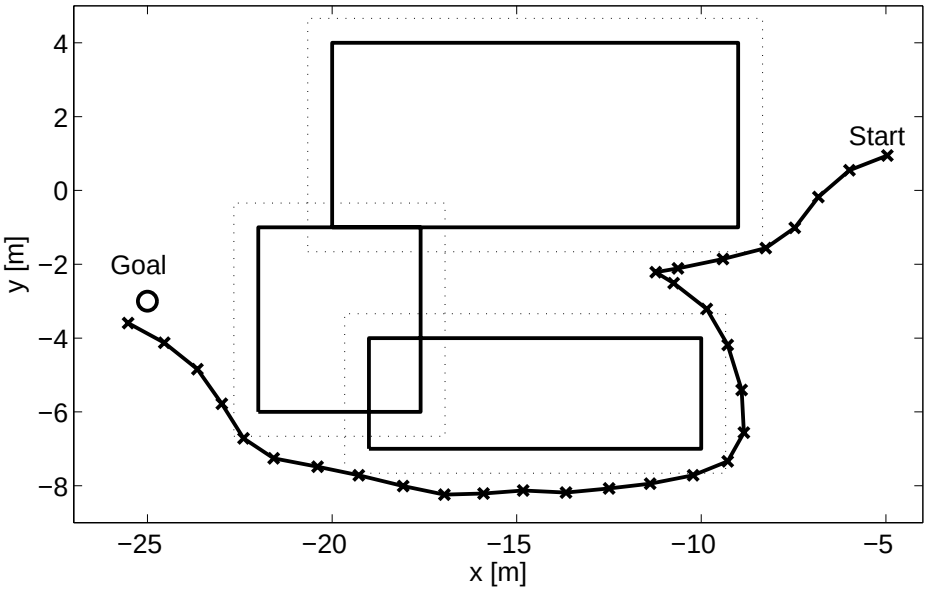


Figure 2: Representation of a cost-to-go function[23]



(a) Simple terminal penalty – feasible but trapped



(b) Intelligent terminal penalty – feasible and not trapped

Figure 3: Comparison of two robust safe trajectory planning approaches. Two trajectories are generated with simple terminal penalty and intelligent terminal penalty. The vehicle starts in the right, and the goal is marked with ∘. The disturbance level is 10% of the control magnitude.

function that represents the future maneuver of the aircraft operated in an obstacle rich field. For UAV trajectory planning problems with no-fly zones, a cost-to-go evaluated at the terminal state is based on the estimate of the collision free path length from the terminal state to the target region as shown in Figure 2. The cost-to-go estimate is based on the best knowledge of the world, and the new information could become available beyond the detection range. The combination of the detailed trajectory over the planning horizon and the simple trajectory beyond it produced a computationally efficient trajectory optimization algorithm with near-optimal performance.[21]

Figure 3 compares trajectories generated with different terminal penalty functions. The obstacle in the left is not initially known. The optimal route is to go through a narrow passage between the other two obstacles. The vehicle finds a new obstacle when it comes within the sensor detection range. The sensor detection range is 8 meters and is larger than the planning horizon ($N = 5$ in this example only). The safety set $\mathcal{S}(k)$ is not affected by the new information of the environment and hence is invariant after the obstacle discovery, as long as $\mathcal{S}(k)$ is within the detection range. Figure (a) shows that the vehicle remains safe against the pop-up obstacle under persistent disturbance. In this case, a simple two-norm distance to the goal is used as a cost-to-go function, and the vehicle is trapped in the concave region. In Figure (b), an intelligent cost-to-go function brings the vehicle out of the entrapment successfully guiding it to the target.

Note that the algorithm maintains safety as long as $\mathcal{S}(k)$ is invariant at time $(k+1)$. Other applications include planning under moving obstacles and multi-vehicle collision avoidance problems.[17]

C. Long Trajectory Design

The last example demonstrates that the approach developed in this paper can design a very long trajectory without computational issues. In this example, the target region is far from the current vehicle location. In order for the plan to reach the target set, it is required to make a plan as long as $\sim$30 steps. Designing one long trajectory that reaches this target set is not computationally tractable for real-time applications.

Figure 4 shows trajectories generated under three different disturbance levels.

- $w_{\max} = 0$
- $w_{\max} = 0.1\, a_{\max}$
- $w_{\max} = 0.2\, a_{\max}$

In all cases, the robust controller guided the vehicle to the target set, and the average computation time was less than 0.2 second.

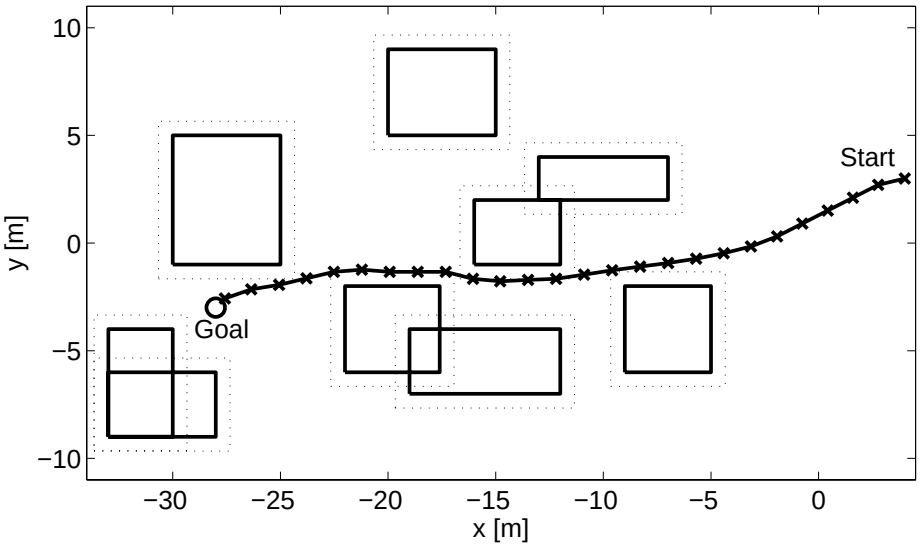
When the disturbance level is 10% of the control authority, the trajectory is similar to the one with no disturbance. However, when the disturbance level is raised to 20% of the control authority, the vehicle takes a different route because the passage in the middle of the figure used by the other plans is too narrow to pass through robustly. A cost-to-go calculation based on the robustified environment does not allow the vehicle to enter the narrow passage where the vehicle could violate the collision avoidance constraints due to a strong disturbance.

Note that the vehicle moves slowly when the disturbance is strong, as it is expected intuitively. Because more margin must be saved to reject a stronger disturbance, less control authority can be used when generating the trajectory. The hovering state used as a terminal invariant set requires the vehicle be able to stop at the end of each plan using the small control authority available in the prediction.

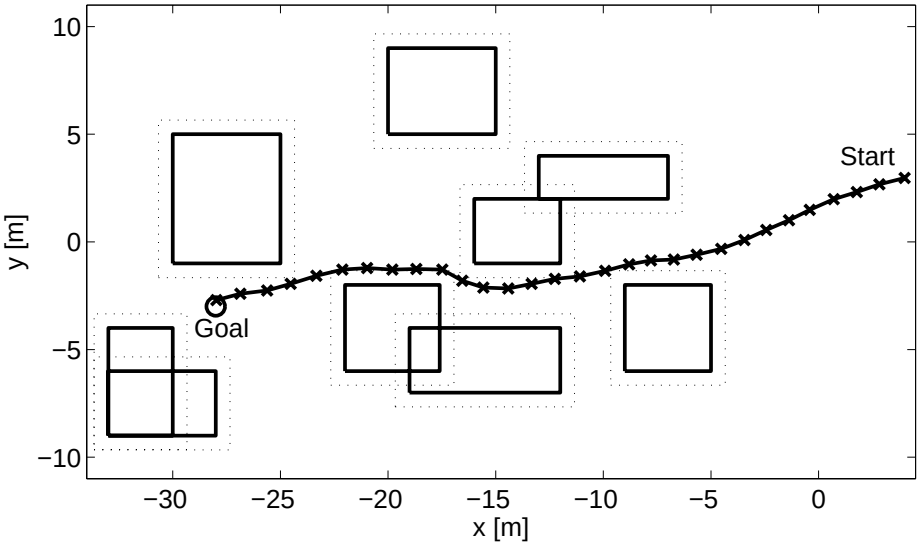
Table 1: Comparison of the performance for three different disturbance levels.

Disturbance level	Average speed	Steps
0 %	0.50 m/s	26
10 %	0.44 m/s	30
20 %	0.28 m/s	48

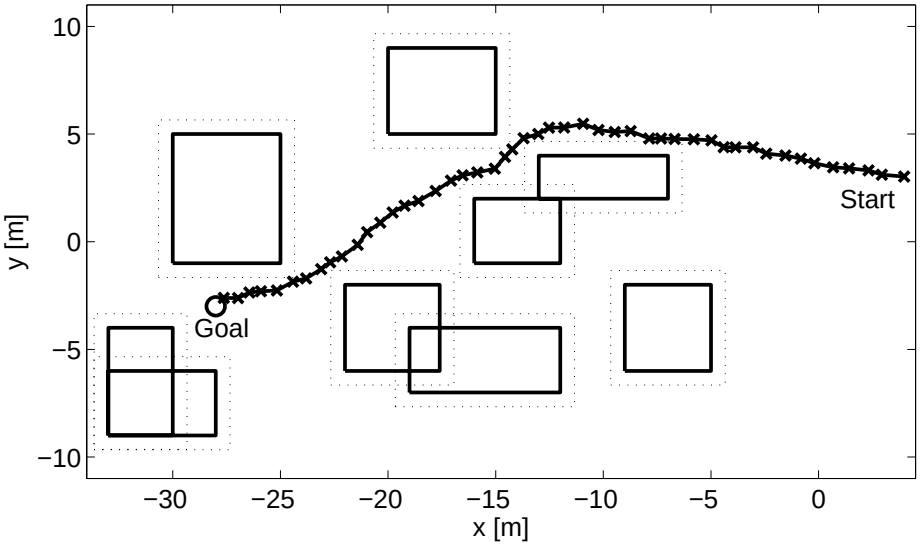
Table 1 summarizes this result. The average speed becomes significantly smaller when the disturbance level is increased from 10% to 20%. The number of steps it takes to reach the target set is significantly longer with the 20% disturbance level, partly because of the longer route it chooses, but mainly due to the reduced speed.



(a) No disturbance.



(b) Disturbance level 10%.



(c) Disturbance level 20%.

Figure 4: Trajectories generated by the robust controller. The vehicle starts in the right, and the goal is marked with ∘.

IV. Conclusions

This paper presented a computationally efficient robust constrained RHC algorithm. Our result extends previous algorithms to allow for much shorter plans that do not necessarily reach the target set. Instead, each plan segment is constrained to enter a robust control invariant admissible set that is constructed on-line. Simulation results for a rotorcraft showed that the proposed algorithm safely navigates the vehicle to the target under the action of an unknown but bounded disturbance. A good choice of the cost-to-go function was shown to significantly improve the performance while maintaining feasibility of the optimizations.

V. Acknowledgements

Research funded in part by AFOSR Grant # FA9550-04-1-0458.

References

[1]Maciejowski, J. M., *Predictive Control with Constraints*, Prentice Hall, 2002.

[2]Rossiter, J. A., *Model-Based Predictive Control: A Practical Approach*, CRC Press, 2003.

[3]Primbs, J. A., "The analysis of optimization based controllers," *Automatica*, Vol. 37, No. 6, 2001, pp. 933–938.

[4]Scokaert, P. O. M. and Mayne, D. Q., "Min-Max Feedback Model Predictive Control for Constrained Linear Systems," *IEEE Transactions on Automatic Control*, Vol. 8, No. 43, August 1998, pp. 1136–1142.

[5]Casavola, A., Giannelli, M., and Mosca, E., "Min-max predictive control strategies for input-saturated polytopic uncertain systems," *Automatica*, Vol. 36, 2000, pp. 125–133.

[6]Lofberg, J., *Minimax approaches to robust model predictive control*, Ph.D. thesis, Linkoping University, 2003.

[7]Kothare, M. V., Balakrishnan, V., and Morari, M., "Robust Constrained Model Predictive Control using Linear Matrix Inequalities," *Automatica*, Vol. 32, No. 10, 1996, pp. 1361–1379.

[8]Wan, Z. and Kothare, M. V., "Robust Output Feedback Model Predictive Control Using Off-line Linear Matrix Inequalities," *Proceedings of the IEEE American Control Conference*, June 2001.

[9]Cuzzolaa, F. A., Geromel, J. C., and Morari, M., "An improved approach for constrained robust model predictive control," *Automatica*, Vol. 38, No. 7, 2002, pp. 1183 1189.

[10]Abate, A. and Ghaoui, L. E., "Robust Model Predictive Control through Adjustable Variables: an application to Path Planning," *Proceedings of the IEEE Conference on Decision and Control*, December 2004.

[11]Gossner, J. R., Kouvaritakis, B., and Rossiter, J. A., "Stable Generalized Predictive Control with Constraints and Bounded Disturbances," *Automatica*, Vol. 33, No. 4, 1996, pp. 551–568.

[12]Michalska, H. and Mayne, D. Q., "Robust Receding Horizon Control of Constrained Nonlinear Systems," *IEEE Transactions on Automatic Control*, Vol. 38, No. 11, November 1993, pp. 1623–1633.

[13]Kerrigan, E. C., *Robust Constraint Satisfaction Invariant Sets and Predictive Control*, Ph.D. thesis, University of Cambridge, November 2000.

[14]Kvasnica, M., Grieder, P., Baotic, M., and Christophersen, F. J., *Multi-Parametric Toolbox (MPT)*, ETH, June 2004.

[15]Richards, A., *Robust Constrained Model Predictive Control*, Ph.D. thesis, MIT, December 2004.

[16]Schouwenaars, T., How, J., and Feron, E., "Receding Horizon Path Planning with Implicit Safety Guarantees," *Proceedings of the IEEE American Control Conference*, IEEE, Boston, MA, July 2004.

[17]Schouwenaars, T., How, J., and Feron, E., "Decentralized Cooperative Trajectory Planning of Multiple Aircraft with Hard Safety Guarantees," *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, August 2004.

[18]Schouwenaars, T., Valenti, M., Feron, E., and How, J. P., "Linear Programming and Language Processing for Human-UAV Team Mission Planning and Execution," *AIAA Journal of Guidance, Control, and Dynamics*, accepted June 2005.

[19]Chisci, L., Rossiter, J. A., and Zappa, G., "Systems with Persistent Disturbances: Predictive Control with Restricted Constraints," *Automatica*, Vol. 37, No. 7, 2001, pp. 1019–1028.

[20]Kolmanovsky, I. and Gilbert, E. G., "Maximal Output Admissible Sets for Discrete-Time Systems with Disturbance Inputs," *Proceedings of the IEEE American Control Conference*, Seattle WA, 1995.

[21]Bellingham, J., Richards, A., and How, J., "Receding Horizon Control of Autonomous Aerial Vehicles," *Proceedings of the IEEE American Control Conference*, Anchorage, AK, May 2002, pp. 3741–3746.

[22]Richards, A. and How, J., "Implementation of Robust Decentralized Model Predictive Control," *AIAA GNC*, 2005.

[23]Richards, A., Kuwata, Y., and How, J., "Experimental Demonstrations of Real-time MILP Control," *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, Austin, TX, Aug 2003.

[24]Kuwata, Y. and How, J., "Stable Trajectory Design for Highly Constrained Environments using Receding Horizon Control," *Proceedings of the IEEE American Control Conference*, IEEE, June 2004, pp. 902–907.