Model Predictive Control of Vehicle Maneuvers with Guaranteed Completion Time and Robust Feasibility

Arthur Richards [1] and Jonathan P. How [2]

Abstract

A formulation for Model Predictive Control is presented for application to vehicle maneuvering problems in which the target regions need not contain equilibrium points. Examples include a spacecraft rendezvous approach to a radial separation from the target and a UAV required to fly through several waypoints. Previous forms of MPC are not applicable to this class of problems because they are tailored to the control of plants about steady-state conditions. Mixed-Integer Linear Programming is used to solve the trajectory optimizations, allowing the inclusion of non-convex avoidance constraints. Analytical proofs are given to show that the problem will always be completed in finite time and that, subject to initial feasibility, the optimization solved at each step will always be feasible in the presence of a bounded disturbance. The formulation is demonstrated in several simulations, including both aircraft and spacecraft, with extension to multiple vehicle problems.

1 Introduction

This paper extends Model Predictive Control (MPC) to applications in vehicle maneuvering problems. MPC is a feedback control scheme in which a trajectory optimization is solved at each time step [5]. The first control input of the optimal sequence is applied and the optimization is repeated at each subsequent step. Because the on-line optimization explicitly includes the operating constraints, MPC can operate closer to constraint boundaries than traditional control schemes [5]. The resulting efficiency gains have made MPC popular for process control. In the field of aerospace control, MPC has been used to stabilize an aerodynamic system [13] and for spacecraft maneuvers [14]. Other work [1] has used the combination of MILP and MPC to stabilize general hybrid systems around equilibrium points.

The first innovation of this work is the formulation of MPC with general terminal constraints. When MPC is used for steady-state control, terminal constraints are applied to ensure stability. It has been shown [6] that stability depends on the terminal state lying in an *invariant set*. That is, a feasible feedback control law may be identified that would keep the state within that set once it had been entered. For maneuvering problems, it is natural that the terminal set be the target region for the maneuver. However, it is restrictive to enforce this set to be invariant. For example, a spacecraft rendezvous may require the chaser vehicle to be at a certain position relative to the target, moving towards it with a certain velocity. All of these quantities are specified with some tolerance, forming a target region in the state space. When the chaser enters that region, a physical connection would be made and the maneuver is complete. However, since the target velocity is non-zero, the region is not invariant. The formulation presented in this paper replaces the notion of stability with *completion*: given an arbitrary target set in the state space, the system will reach the set in finite time.

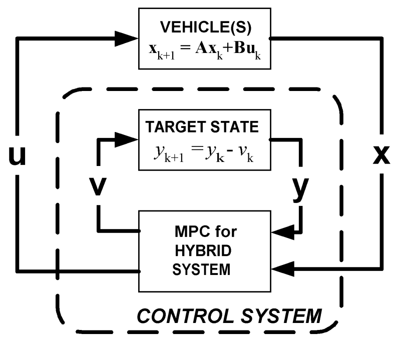

Fig. 1: Control System Overview

Fig. 1 shows an overview of the control scheme proposed in this work. As well as the dynamic states $\mathbf{x}$ and inputs $\mathbf{u}$, a finite state machine is included in the system model, with discrete state y and input v. The state $y = 0$ implies that the maneuver has been completed, $y = 1$ otherwise. In the model, constraints on v couple the continuous and discrete systems such that the transition $y \rightarrow 0$ can only occur when the vehicle states $\mathbf{x}$ are in the prescribed target region. MPC acts upon the combined, hybrid system to drive the combined state ($\mathbf{x}$, y) to the terminal set $y = 0$ in finite time. The method can be extended to include multiple vehicles and multiple targets, allowing the scheme to be used for UAV maneuvers combining trajectory control and waypoint assignment [10].

The new formulation extends existing MPC methods. Previous work [16] has shown that allowing the horizon length to vary leads to finite-time completion. For the

[1]Space Systems Laboratory, Massachusetts Institute of Technology, Cambridge MA 02139 arthurr@mit.edu
[2]Associate Professor, MIT jhow@mit.edu

single-vehicle, single-target case, the formulation in this paper provides a way of encoding this variation. Other work has shown that MPC can reach an invariant set in finite time [11]. In this paper, the set $y = 0$ is invariant by construction. By coupling entry to this set with presence in the target region, finite-time arrival in an arbitrary set is accomplished.

The inputs to the discrete state machine v_k are binary variables in a Mixed-Integer Linear Program (MILP). The use of MILP optimization offers significant advantages beyond the inclusion of the discrete state model. Previous work has shown that MILP can design trajectories for complex maneuvers, including non-convex constraints such as collision avoidance and waypoint assignment [2, 9]. Although a relatively complicated optimization form, MILP problems can be solved by a branch-and-bound algorithm [15] and efficient software [4] exists for this purpose. In addition, problem-specific techniques can be used to solve trajectory design MILPs rapidly [9].

The second innovation is a new method of achieving *robust feasibility*. This is a concern when using MPC, since it tends to operate the systems close to the constraint boundaries. While this is beneficial for performance, it means that small disturbances could 'push' the state beyond the feasible region. If this happens, the optimization becomes infeasible, and no control is defined. To avoid this situation, modifications are derived such that the trajectory optimization is always feasible, provided the disturbance remains within known bounds. Previous appraches to MPC robustness have involved min-max problems [16] or invariant set methods [7]. The new approach restricts control effort at later steps in the plan, equivalent to planning conservatively for the far future, in preparation for compensation by future feedback action. This guarantees robust feasibility without increasing the size of the problem, which is a drawback of many existing methods [5].

Three simulation examples are included. The first involves a spacecraft rendezvous with a space station, requiring approach to a non-equilibrium target. This demonstrates the finite-time completion for general maneuver problems. The second example involves a single aircraft traversing an obstacle field to a fixed target region. It illustrates the robust feasibility in the presence of unmodeled disturbance. The final example extends the problem to two aircraft, demonstrating the application to multiple vehicles.

2 Controller Formulation

This section presents the formulation of the controller and its proof of finite-time completion. For clarity, the single vehicle, single objective case is shown first. The extension to multiple vehicles and targets is then demonstrated.

2.1 Single Vehicle Case

Problem Statement The aim is to control a vehicle, with discretized dynamics

$$\mathbf{x}_{\mathrm{k+1}} = \mathbf{A}\mathbf{x}_{\mathrm{k}} + \mathbf{B}\mathbf{u}_{\mathrm{k}} \tag{1}$$

where $\mathbf{u} \in \Re^m$ and $\mathbf{x} \in \Re^n$, such that it passes through the target region at some time step

$$\exists k \ \mathbf{P}\mathbf{x}_{\mathrm{k}} \leq \mathbf{q} \tag{2}$$

where $\mathbf{P} \in \Re^{(l \times n)}$ and $\mathbf{q} \in \Re^l$ are given parameters. For example, a UAV problem might require the vehicle to pass within 10 units of the origin at any altitude, requiring four constraints ($l = 4$). A spacecraft rendezvous problem might require additional constraints on the velocity, which can also be written in this form.

Throughout the maneuver, the system is required to obey various operating constraints, both convex and non-convex, applied to the states and control. Examples include control limits and obstacle avoidance [2, 9]. All such constraints can be expressed in the following linear constraint form

$$\mathbf{C}_1\mathbf{x} + \mathbf{C}_2\mathbf{u} + \mathbf{C}_3\delta + \mathbf{C}_4 \geq \mathbf{0} \tag{3}$$

where δ is a vector of auxiliary binary variables and $\mathbf{0}$ is a suitably-sized vector whose entries are zero. The objective function is a combination of time, counted in units of time steps, and the one norm of the input (*i.e.* the fuel use, for spacecraft), weighted by α

$$J = \sum_{\mathrm{k}=0}^{K_F} (1 + \alpha|\mathbf{u}_{\mathrm{k}}|) \tag{4}$$

where K_F is the final step of the maneuver.

MPC Formulation At each time step k, the model predictive control problem is to design an input sequence $\{\mathbf{u}_{k|k} \ldots \mathbf{u}_{k|(k+N)}\}$ for the horizon N, where $\mathbf{u}_{k|j}$ denotes the control designed at time k for application at time j. Then the first element of that sequence is applied. The optimization includes a model of the dynamics

$$\forall j \in [k \ldots (k+N)] \ \mathbf{x}_{\mathrm{k|(j+1)}} = \mathbf{A}\mathbf{x}_{\mathrm{k|j}} + \mathbf{B}\mathbf{u}_{\mathrm{k|j}} \tag{5}$$

In the optimization model, the system state is augmented with a single discrete state, $y \in \{0, 1\}$, where $y = 0$ implies, by definition, that the target has been reached at the current step or earlier. The dynamics of y are described by a discrete-time state-space model

$$\forall j \in [k \ldots (k+N)] \ y_{\mathrm{k|(j+1)}} = y_{\mathrm{k|j}} - v_{\mathrm{k|j}} \tag{6}$$

The discrete input sequence $v_{\mathrm{k|j}} \in \{0, 1\} \ \forall j \in [k \ldots (k+N)]$ is an additional decision variable in the optimization. Since these variables are binary, integer optimization is required. The following constraint couples the discrete input v to the continuous states, thereby linking the discrete state machine to the continuous system.

$$\begin{aligned} &\forall j \in [k \dots (k+N)] \\ &\mathbf{P}(\mathbf{A}\mathbf{x}_{\mathrm{k|j}} + \mathbf{B}\mathbf{u}_{\mathrm{k|j}}) \leq \mathbf{q} + \mathbf{1}M(1 - v_{\mathrm{k|j}}) \end{aligned} \tag{7}$$

where $\mathbf{1}$ is a vector of suitable size whose entries are one and M is a large positive integer. If $v_{\mathrm{k|j}} = 1$, the target constraint defined by (2) is met at the step $j+1$, otherwise the constraint is relaxed. Also, if $v_{\mathrm{k|j}} = 1$, the discrete state y makes the transition from 1 to 0 between steps j and $j+1$, according to (6). Therefore, the model of the discrete state is consistent with its definition: $y = 0$ implies that the target has been reached. Since v is restricted to take only integer values, the continuous constraints $0 \leq y_{\mathrm{k|j}} \leq 1$ are sufficient to ensure that y takes only values 0 or 1. It is not necessary to specify y as an additional integer decision variable in the problem.

The operating constraints (3) are represented in the optimization in the following form

$$\begin{aligned} &\forall j \in [k \dots (k+N)] \\ &\mathbf{C}_1\mathbf{x}_{\mathrm{k|j}} + \mathbf{C}_2\mathbf{u}_{\mathrm{k|j}} + \mathbf{C}_3\delta_{\mathrm{k|j}} + \mathbf{1}M(1 - y_{\mathrm{k|j}}) \geq \mathbf{0} \end{aligned} \tag{8}$$

in which the constraints are relaxed if $y = 0$. This relaxation of the constraints after completion of the problem means the plan incurs no cost after passing through the target. This is an important property for the proof of completion, shown later in this section.

The initial conditions are taken from the current state values. The discrete state y_{k} is propagated outside the controller according to its dynamics model (6) (see Fig. 1)

$$\mathbf{x}_{\mathrm{k|k}} = \mathbf{x}_{\mathrm{k}} \tag{9}$$

$$y_{\mathrm{k|k}} = y_{\mathrm{k}} \tag{10}$$

The terminal constraint is that the discrete state $y_{\mathrm{k}} = 0$

$$y_{\mathrm{k|k+N+1}} = 0 \tag{11}$$

The objective function (4) is represented in the optimization by the following

$$J^* = \min_{\{\mathbf{x},\mathbf{u},y,v\}} \sum_{\mathrm{j=k}}^{(k+N)} (\alpha|\mathbf{u}_{\mathrm{k|j}}| + y_{\mathrm{k|j}}) \tag{12}$$

in which the one-norm $|\mathbf{u}|$ can be found using slack variables and the inclusion of $yk|j$ in the summation represents the penalty on time, since $y_{\mathrm{k}} = 1$ before completion and 0 after.

MPC Algorithm At each time step:

1. Solve the minimization of (12) subject to constraints (5)–(11);
2. Apply the first element of the continuous-system control sequence $\mathbf{u}_{\mathrm{k|k}}$ to the vehicle;
3. Propagate discrete state y_{k} by applying the first element of the discrete control sequence $v_{\mathrm{k|k}}$ to its dynamics model (6);
4. Repeat from 1 until $y_{\mathrm{k}} = 0$.

Proposition: The control resulting from the algorithm above drives the system to the target region (2) in finite time.

Proof: Assume the system is in state $(\mathbf{x}_{\mathrm{k}}, y_{\mathrm{k}})$ and the optimization solved at this point yields the following optimal sequences

$$\begin{array}{lll} \{ & \mathbf{u}_{k|k} \dots \mathbf{u}_{k|(k+N)} & \} \\ \{ & v_{k|k} \dots v_{k|(k+N)} & \} \\ \{ & \mathbf{x}_{k|k} \dots \mathbf{x}_{k|(k+N)} & \mathbf{x}_{k|(k+N+1)} \} \\ \{ & y_{k|k} \dots y_{k|(k+N)} & y_{k|(k+N+1)} \} \end{array}$$

where $\mathbf{x}_{k|k} = \mathbf{x}_k$ and $y_{k|k} = y_k$, according to the initial conditions of the optimization, and $y_{k|(k+N+1)} = 0$ to suit the terminal constraints. Let the cost of this sequence be J^*_k. After the applying the first elements of this sequence, the system state equals the second state of the sequence $(\mathbf{x}_{\mathrm{k|(k+1)}}, y_{\mathrm{k|(k+1)}})$. Then the following sequence is feasible for the next optimization, at step $k+1$. It consists of the completion of the previous problem, followed by zero control input on the final step $k+N+1$. The relaxation of the constraints (8) after completion ensures that this new final step is admissible.

$$\begin{array}{llll} \{ & \mathbf{u}_{\mathrm{k|(k+1)}} \dots \mathbf{u}_{\mathrm{k|(k+N)}} & \mathbf{0}\} & \\ \{ & v_{\mathrm{k|(k+1)}} \dots v_{\mathrm{k|(k+N)}} & 0\} & \\ \{ & \mathbf{x}_{\mathrm{k|(k+1)}} \dots \mathbf{x}_{\mathrm{k|(k+N)}} & \mathbf{x}_{\mathrm{k|(k+N+1)}} & \mathbf{A}\mathbf{x}_{\mathrm{k|(k+N+1)}}\} \\ \{ & y_{\mathrm{k|(k+1)}} \dots y_{\mathrm{k|(k+N)}} & y_{\mathrm{k|(k+N+1)}} & 0\} \end{array}$$

The cost value of this candidate sequence is given by $\hat{J}_{(k+1)} = J^*_k - \alpha|\mathbf{u}_{k|k}| - y_{k|k}$, since the additional steps at the end of the sequence incur no cost. It is an upper bound on the cost of the optimal solution, hence

$$J^*_{\mathrm{(k+1)}} - J^*_{\mathrm{k}} \leq -\alpha|\mathbf{u}_{\mathrm{k}}| - y_{\mathrm{k}} \tag{13}$$

This shows that the cost J^*_{k} decreases by at least one unit per time step while $y_{\mathrm{k}} = 1$. Since J^*_{k} must be positive by construction, then $y_{\mathrm{k}} \to 0$ must occur before $k \to \bar{k}$, where $\bar{k}$ is the smallest integer larger than the initial cost J^*_0. Due to the coupling (7), $y_{\mathrm{k}} \to 0$ implies that the target region has been reached. Therefore, the maneuver must be completed in fewer than $\bar{k}$ steps. □

2.2 Multi-Vehicle, Multi-Target Case

The formulation in the previous section naturally extends to multiple vehicles and targets. Let there be N_T targets and N_V vehicles. Each vehicle p has input $\mathbf{u}_{\mathrm{pk}}$ and state $\mathbf{x}_{\mathrm{pk}}$, with its own dynamics model and constraints in the optimization, similar to (1) and (3). Each target q is defined by an equation similar to (2) with its own parameters $\mathbf{P}_{\mathrm{q}}$ and $\mathbf{q}_{\mathrm{q}}$. There is an objective state y_{q} for each target. The modified objective dynamics are

$$y_{\mathrm{qk|(j+1)}} = y_{\mathrm{qk|j}} - \sum_p v_{\mathrm{pqk|j}} \tag{14}$$

while the coupling constraints become

$$\mathbf{P}_\mathrm{q}(\mathbf{A}_\mathrm{p}\mathbf{x}_\mathrm{pk|j} + \mathbf{B}_\mathrm{p}\mathbf{u}_\mathrm{pk|j}) \leq \mathbf{q}_\mathrm{q} + \mathbf{1}M(1 - v_\mathrm{pqk|j}) \quad (15)$$

where $v_\mathrm{pqk|j} = 1$ implies that vehicle p visits target q at step $j+1$ in the plan made at step k. From (14), this means that y_q changes from 1 to 0 at that step.

Capability constraints [10] can also be included in this framework. An entry in the capability matrix $K_\mathrm{pq} = 1$ implies that vehicle p can be assigned to target q. The additional constraints are

$$\forall p, q, j \ v_\mathrm{pqk|j} \leq K_\mathrm{pq} \quad (16)$$

The terminal set for the optimization is now "all objectives met". This corresponds to having all elements of $\mathbf{y}$ set to zero. The infinity-norm of $\mathbf{y}$ is found by the constraint

$$\forall q \ \hat{z}_\mathrm{k|j} \geq y_\mathrm{qk|j} \quad (17)$$

and the terminal constraint is therefore $\hat{z} = 0$, implying $\mathbf{y} = \mathbf{0}$. Using a proof very similar to that in the previous section, it can be shown that $\hat{z} \to 0$ in finite time, implying that all the targets are visited.

3 Robust Feasibility

The on-line optimization in MPC leads to state trajectories at the very limits of the operating constraints. While this offers improved efficiency, it also raises concerns over robustness: a disturbance could 'push' the state outside the operating region, making the optimization infeasible. The property of *robust feasibility* [5] is achieved if, given a solution from the initial condition, all subsequent optimizations will be feasible. In this section, the formulation is modified to achieve this property under the action of an unknown but bounded disturbance.

The analysis below identifies expressions for the corrections to the disturbance at each time step, such that the new plan rejoins the previous plan after two time steps. The two-step correction is chosen because it can be exactly calculated for second-order systems, which are typically used for vehicles. It is not necessarily desirable to correct in two steps, but the existence of this solution ensures that the optimization is feasible. While the exact disturbance is not known, the disturbance bounds can be used to derive control bounds such that the corrections are always feasible. Therefore, there is always at least one feasible solution at the new time step. Note that the immediate two-step correction is not necessarily implemented: the optimization will usually find a less costly correction spread over a longer period.

Assume the system is in state $\mathbf{x}_\mathrm{k}$. An optimization is solved with the following dynamic constraints

$$\begin{array}{rcll} \mathbf{x}_\mathrm{k|(k+1)} & = & \mathbf{A}\mathbf{x}_\mathrm{k} & +\mathbf{B}_1\mathbf{u}_\mathrm{k|(k} \\ \mathbf{x}_\mathrm{k|(k+2)} & = & \mathbf{A}\mathbf{x}_\mathrm{k|(k+1)} & +\mathbf{B}_1\mathbf{u}_\mathrm{k|(k+1)} \\ \mathbf{x}_\mathrm{k|(k+3)} & = & \mathbf{A}\mathbf{x}_\mathrm{k|(k+2)} & +\mathbf{B}_1\mathbf{u}_\mathrm{k|(k+2)} \end{array} \quad (18)$$

The first control step $\mathbf{u}_\mathrm{k|(k}$ is executed and a random disturbance $\mathbf{w}_k$ acts, moving the system to the new state

$$\mathbf{x}_\mathrm{(k+1)} = \mathbf{A}\mathbf{x}_\mathrm{k} + \mathbf{B}_1\mathbf{u}_\mathrm{k|k} + \mathbf{B}_2\mathbf{w}_\mathrm{k} \quad (19)$$

At the next time step, a new plan is sought to satisfy

$$\begin{array}{rcll} \mathbf{x}_\mathrm{(k+1)|(k+2)} & = & \mathbf{A}\mathbf{x}_\mathrm{(k+1)} & +\mathbf{B}_1\mathbf{u}_\mathrm{(k+1)|(k+1)} \\ \mathbf{x}_\mathrm{(k+1)|(k+3)} & = & \mathbf{A}\mathbf{x}_\mathrm{(k+1)|(k+2)} & +\mathbf{B}_1\mathbf{u}_\mathrm{(k+1)|(k+2)} \end{array} \quad (20)$$

We seek the solution to the new planning problem such that the new plan rejoins the old plan at the third step, satisfying

$$\mathbf{x}_\mathrm{(k+1)|(k+3)} = \mathbf{x}_\mathrm{k|(k+3)} \quad (21)$$

Combining (18) - (21), the new plan can be expressed as perturbations to the control terms of the previous plan

$$\begin{bmatrix} \mathbf{u}_\mathrm{(k+1)|(k+1)} - \mathbf{u}_\mathrm{k|(k+1)} \\ \mathbf{u}_\mathrm{(k+1)|(k+2)} - \mathbf{u}_\mathrm{k|(k+2)} \end{bmatrix} = -[\mathbf{A}\mathbf{B}_1 \ \mathbf{B}_1]^{-1}\mathbf{A}^2\mathbf{B}_2\mathbf{w}_k \quad (22)$$

Note that the invertibility of the matrix $[\mathbf{A}\mathbf{B}_1 \ \mathbf{B}_1]$ follows from the two-step controllability of the system, and illustrates the choice of the two-step correction. Given a norm bound on the disturbance $\|\mathbf{w}\| \leq \bar{W}$, the following bounds on the control perturbations can be found

$$\begin{array}{rcl} \|\mathbf{u}_\mathrm{(k+1)|(k+1)} - \mathbf{u}_\mathrm{k|(k+1)}\| & \leq & \beta_1 \\ \|\mathbf{u}_\mathrm{(k+1)|(k+2)} - \mathbf{u}_\mathrm{k|(k+2)}\| & \leq & \beta_2 \end{array} \quad (23)$$

where the associated induced norms are used to calculated the β values as follows

$$\begin{array}{rcl} \beta_1 & = & \|[\mathbf{I} \ \mathbf{0}][\mathbf{A}\mathbf{B}_1 \ \mathbf{B}_1]^{-1}\mathbf{A}^2\mathbf{B}_2\| \ \bar{W} \\ \beta_2 & = & \|[\mathbf{0} \ \mathbf{I}][\mathbf{A}\mathbf{B}_1 \ \mathbf{B}_1]^{-1}\mathbf{A}^2\mathbf{B}_2\| \ \bar{W} \end{array} \quad (24)$$

These norm bounds are used to restrict control actions on later plan steps as follows

$$\begin{array}{rcl} \forall k: \ \|\mathbf{u}_\mathrm{k|k}\| & \leq & \bar{U} \\ \|\mathbf{u}_\mathrm{k|(k+1)}\| & \leq & \bar{U} - \beta_1 \\ \|\mathbf{u}_\mathrm{k|j}\| & \leq & \bar{U} - \beta_1 - \beta_2 \ \forall j > k+1 \end{array} \quad (25)$$

The effect of these constraints is illustrated in Fig. 2. When planning at step k, the control for step $k+1$ is restricted to be β_1 below the maximum available $\bar{U}$. Therefore, at the next step, the new plan may increase the control at $k+1$ by up to β_1 units, even if the previous plan used the maximum available control for that step. A similar addition of up to β_2 is allowed at step $k+2$. Therefore, it is always feasible to add the perturbations in (22) and rejoin the previous plan in two steps. This solution is likely to be very expensive, and may not be the solution chosen by the optimization, but its existence guarantees that the problem is feasible. Therefore, if a feasible solution exists at step k, this implies feasibility at step $k+1$ for all disturbances $\mathbf{w}_\mathrm{k}$ with $\|\mathbf{w}_\mathrm{k}\| \leq \bar{W}$.

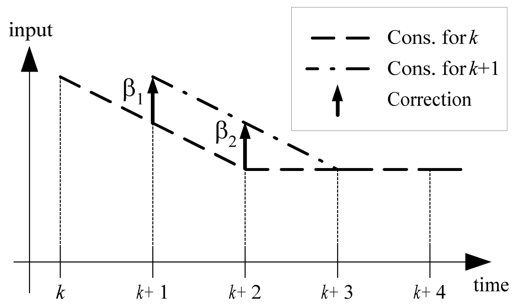

Fig. 2: Illustration of Robustness Technique

The states at the first two steps are also perturbed by the action of the disturbance and the resulting correction. Since the initial state is fixed in the optimization (9), the constraints on this first step may be completely relaxed. The state perturbation to the second plan step can be expressed and bounded by

$$\mathbf{x}_{(k+1)|(k+2)} - \mathbf{x}_{k|(k+2)} = \{\mathbf{I} - \mathbf{B}_1[\mathbf{I}\ \mathbf{0}][\mathbf{AB}_1\ \mathbf{B}_1]^{-1}\mathbf{A}\}\mathbf{AB}_2\mathbf{w}_k \tag{26}$$

hence

$$\|\mathbf{x}_{(k+1)|(k+2)} - \mathbf{x}_{k|(k+2)}\| \leq \alpha \tag{27}$$

$$\alpha = \|\{\mathbf{I} - \mathbf{B}_1[\mathbf{I}\ \mathbf{0}][\mathbf{AB}_1\ \mathbf{B}_1]^{-1}\mathbf{A}\}\mathbf{AB}_2\|\ \bar{W} \tag{28}$$

Since the state constraints may be non-convex, for example enforcing collision avoidance, simple norm bounds of the form (25) cannot be used. Instead, the perturbation is included as a design variable and its norm is limited. If the nominal state constraints are

$$\mathbf{C}_1\mathbf{x}_{k|j} + \mathbf{C}_3\delta_{k|j} + \mathbf{C}_4 \geq \mathbf{0} \tag{29}$$

where $\delta_{k|j}$ are auxiliary binaries for avoidance, then the constraints are revised to include a perturbation vector $\mathbf{\Delta}$ as follows

$$\mathbf{C}_1\mathbf{x}_{k|j} + \mathbf{C}_1\mathbf{\Delta}_{k|j} + \mathbf{C}_3\delta_{k|j} + \mathbf{C}_4 \geq \mathbf{0} \tag{30}$$

and the norm of the perturbation is limited to suit the robustness constraints.

$$\begin{aligned} \forall k:\ \|\mathbf{\Delta}_{k|k}\| &\leq M \\ \|\mathbf{\Delta}_{k|(k+1)}\| &\leq \alpha \\ \|\mathbf{\Delta}_{k|j}\| &\leq 0\ \forall j > k+1 \end{aligned} \tag{31}$$

Note that this represents a relaxation of the nominal state constraints. Depending on the problem, it might be necessary to alter the parameters $\mathbf{C}_i$ to ensure this relaxation does not violate the real constraints on the system. Also note that the large number M is used as the constraint for the first state perturbation, since the initial state constraints are completely relaxed.

4 Implementation

MILP optimizations are solved using CPLEX software [4]. The problem is translated into the AMPL modeling language [3]. Simulation is performed in MATLAB, along with the data interface to AMPL. The constraint and dynamics matrices are written to data files at the start of each simulation. At each step, the initial conditions are written to separate data files, and an AMPL script is invoked to combine the relevant model and data, invoke CPLEX and return the result to MATLAB.

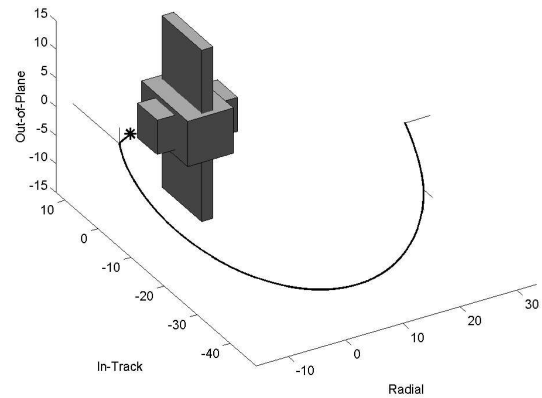

Fig. 3: Spacecraft Rendezvous with ISS under Maneuvering MPC

5 Examples

5.1 ISS Rendezvous

This section demonstrates the application of the described MPC technique to the problem of autonomous rendezvous of a spacecraft with a space station. Fig. 3 shows a spacecraft maneuvering under MPC from its given starting point to a target state $\mathbf{x}_T$ on the radial axis, marked by a star. The station was modeled as the boxes shown in the figure. The control time step was 90 s, with a horizon length of 30 steps. The spacecraft reached the target after just over 2800 seconds maneuvering time.

Fig. 4 shows the optimization costs from the solution at each control step. Analysis predicts that the cost should decrease by at least one unit per control step. The dashed line in the figure shows this rate of decrease starting from the initial cost, representing the upper bound on convergence. Not only is the cost function below this line, but its gradient is always the same or steeper, as predicted by the analysis.

Fig. 5 shows the same simulation using MPC with the simple terminal constraint $\mathbf{x}_{k|(k+N)} = \mathbf{x}_T$, instead of the discrete state model described in this paper. Had $\mathbf{x}_T$ been an equilibrium, the control would have been stabilizing, as proven in Ref. [1]. In this case, after 9000 seconds of simulated time, the chaser craft has not reached the target and does not appear to be converging. This demonstrates the significance of the MPC formulation presented in this paper: the target region need not be invariant.

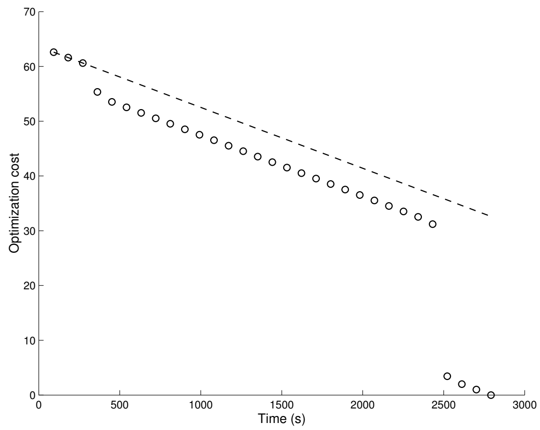

Fig. 4: Optimization Costs during Maneuver. The dashed line shows the upper bound on convergence from the first step.

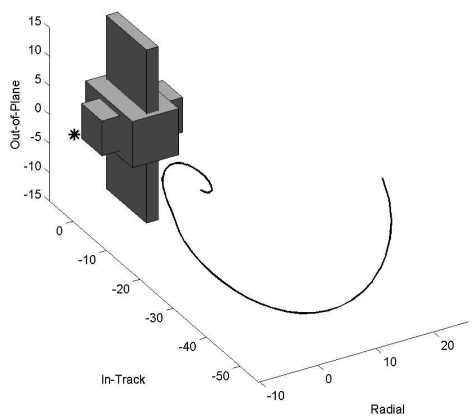

Fig. 5: Rendezvous Simulation as Fig. 3 using MPC with Simple Terminal Constraint

5.2 Robust Control of Aircraft

To demonstrate the effect of the robustness formulation, multiple simulations of the same maneuver were performed with random disturbances applied. The maneuver can be seen in Fig. 6, simulating control of a single aircraft traversing an obstacle field to a fixed target [12]. A random disturbance force, with magnitude limited to 15% of the turning force, was applied at each time step. First, twelve simulations were performed using the basic MILP/MPC formulation, without modifications for robustness. In every case, the system reached a state in which the optimization was infeasible, occurring between 3 and 11 time steps after the start. When the modifications for robustness were included, a further 12 simulated maneuvers were all completed successfully. Fig. 6 shows the trajectories from six of the simulations using the robustly-feasible method. The trajectories encroach upon the obstacles, due to the discrete-time enforcement of avoidance and the relaxation of the constraints for robustness. However, these incursions are bounded and the obstacles can be enlarged accordingly.

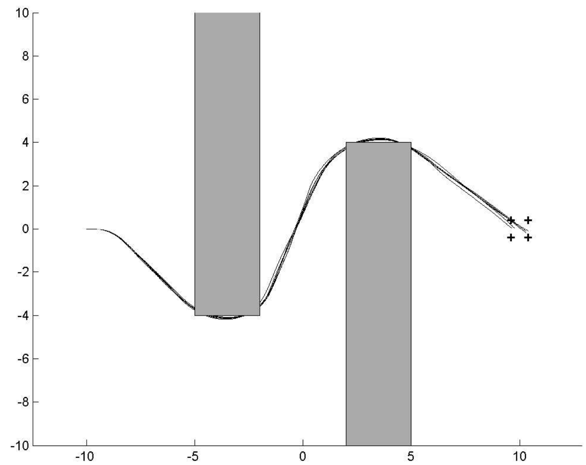

Fig. 6: Six Simulations using Robustly-Feasible Controller. Without the robustness modifications, the controller was unable to complete the maneuver without becoming infeasible.

5.3 Robust Collision Avoidance for Two Aircraft

This section shows the robust controller formulation applied to a multiple-vehicle problem. Two vehicles are required to visit two waypoints, as shown in Fig. 7. The capabilities are constrained such that the assignment is fixed to that shown, *i.e.* $\mathbf{K} = \mathbf{I}$ in (16). The dynamics and disturbances for both vehicles are the same as in the previous example. In the absence of collision avoidance constraints, the two vehicles would travel in straight lines to their assigned destinations, leading to a collision in the center.

Six simulations were performed using MPC without robustness modifications, including collision avoidance constraints. In very case, the optimization become infeasible after two or three steps. In some cases, feasibility was regained later in the simulation, but constraints had been violated and the paths were erratic. Fig. 7 shows the trajectories from six simulations with robustness modification included. All optimizations were feasible throughout. Both vehicles make diversions from the straight paths in order to avoid collision while minimizing overall flight time.

Fig. 8 shows the separation distance between the vehicles, expressed as the infinity-norm of the separation vector, through each of the six simulations. The constraints were set to require a minimum separation of four units, marked by the dashed line. The two vehicles encroach upon this region, since the constraints are applied at discrete intervals and the robustness formulation allows limited relaxation of the constraints. However, knowing the disturbance, maximum speed and time step length, this incursion can be shown to be less than 0.6 units. This level is shown by the dash-dot line in the figure. The predicted separation is maintained throughout.

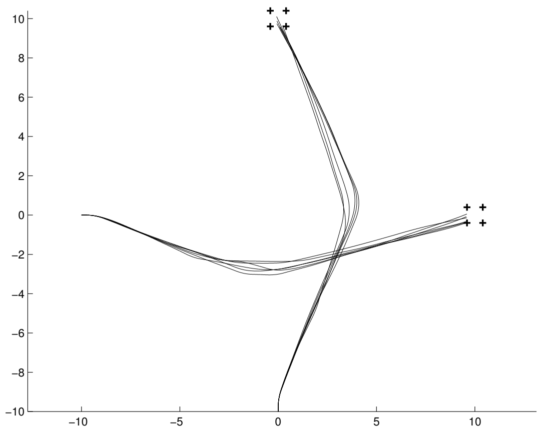

Fig. 7: Six Simulations using Robustly-Feasible Controller with Collision Avoidance

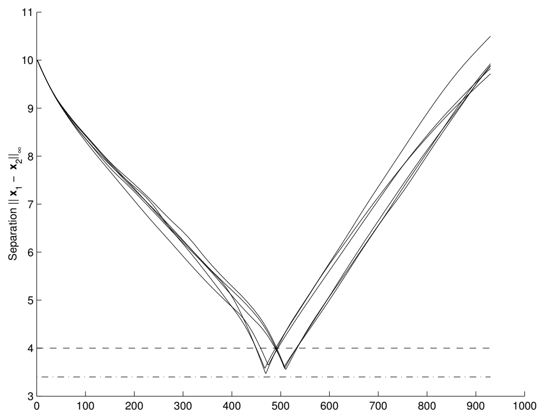

Fig. 8: Separation Distances (∞–norm) during Simulations in Fig. 7

Conclusions

A new formulation for Model Predictive Control has been presented that guarantees finite-time completion of vehicle maneuvers. The targets can be arbitrary regions in state-space, generalizing existing MPC formulations which require equilibrium target states for stability. The system model is augmented with a finite state machine, each state recording whether or not a particular target has been visited. The terminal constraints can then be expressed in terms of the discrete states. Modifications to the constraints also ensure that the controller is robustly feasible: if an initial solution can be found, and a bounded disturbance acts on the system, the optimization problems at subsequent steps are guaranteed to be feasible. The controller analysis is supported by simulations, involving aircraft and spacecraft examples. The combination of guaranteed completion and robust feasibility allow the benefits of MPC – feedback compensation while operating close to constraint boundaries – to be achieved in vehicle maneuvering applications.

Acknowledgements

The research was funded in part under DARPA contract (MICA) # N6601-01-C-8075.

References

[1] A. Bemporad and M. Morari, "Control of Systems Integrating Logic, Dynamics, and Constraints," in *Automatica,* Pergamon / Elsevier Science, New York NY, Vol. 35, 1999, pp. 407–427.

[2] T. Schouwenaars, B. DeMoor, E. Feron and J. How, "Mixed Integer Programming for Multi-Vehicle Path Planning," *European Control Conference,* Porto, Portugal, September 2001, pp. 2603-2608.

[3] R. Fourer, D. M. Gay, and B. W. Kernighar, *AMPL, A modeling language for mathematical programming*, Boyd & Fraser, Danvers MA, (originally published by The Scientific Press) 1993, pp 291–306.

[4] *ILOG AMPL CPLEX System Version 7.0 User's Guide*, ILOG, Incline Village, NV, 2000, pp. 17–53.

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

[6] D. Q. Mayne, J. B. Rawlings, C. V. Rao, P. O. M. Scokaert, "Constrained Model Predictive Control: Stability and Optimality," *Automatica*, 36(2000), Pergamon Press, UK, pp. 789–814.

[7] E. C. Kerrigan and J. M. Maciejowski, "Robust Feasibility in Model Predictive Control: Necessary and Sufficient Conditions," *40th IEEE CDC*, Orlando FL, December 2001, pp. 728–733.

[8] H. P. Rothwangl, "Numerical Synthesis of the Time Optimal Nonlinear State Controller via Mixed Integer Programming," *ACC*, 2001.

[9] A. G. Richards, T. Schouwenaars, J. How, E. Feron, "Spacecraft Trajectory Planning With Collision and Plume Avoidance Using Mixed-Integer Linear Programming," *AIAA JGCD*, Vol. 25 No. 4, July 2002.

[10] A. G. Richards, J. S. Bellingham, M. Tillerson and J. P. How, "Co-ordination and Control of Multiple UAVs", AIAA paper no. 2002-4588, *AIAA Guidance, Navigation, and Control Conference and Exhibit,* 2002.

[11] P. O. M. Scokaert, D. Q. Mayne and J. B. Rawlings, "Suboptimal Model Predictive Control (Feasibility Implies Stability)," *IEEE Trans. on Automatic Control*, Vol. 44 No. 3, 1999, p. 648.

[12] A. G. Richards, J. How, "Aircraft Trajectory Planning with Collision Avoidance using Mixed Integer Linear Programming," *ACC,* Anchorage AK, May 2002.

[13] R. Franz, M. Milam, J. Hauser, "Applied Receding Horizon Control of the Caltech Ducted Fan," *ACC,* Anchorage AK, May 2002.

[14] V. Manikonda, P. O. Arambel, M. Gopinathan, R. K. Mehra and F. Y. Hadaegh, "A Model Predictive Control-based Approach for Spacecraft Formation Keeping and Attitude Control," *ACC,* San Diego CA, June 1999.

[15] C. A. Floudas, *Nonlinear and Mixed-Integer Programming – Fundamentals and Applications*, Oxford University Press, 1995.

[16] P. O. M. Scokaert and D. Q. Mayne, "Min-Max Feedback Model Predictive Control for Constrained Linear Systems," *IEEE Trans. on Automatic Control*, Vol. 43, No. 8, Aug. 1998, p 1136.