2.2 Problem Statement

The aim is to control a linear system with discretized dynamics

$$\mathbf{x}(k+1) = \mathbf{A}\mathbf{x}(k) + \mathbf{B}\mathbf{u}(k) + \mathbf{w}(k) \tag{2.1}$$

where $\mathbf{x}(k) \in \Re^{N_x}$ is the state vector, $\mathbf{u}(k) \in \Re^{N_u}$ is the input, $\mathbf{w}(k) \in \Re^{N_x}$ is the disturbance vector. Assume the system $(\mathbf{A}, \mathbf{B})$ is controllable and the complete state $\mathbf{x}$ is accessible. The disturbance lies in a bounded set but is otherwise unknown

$$\forall k \ \mathbf{w}(k) \in \mathcal{W} \subset \Re^{N_x} \tag{2.2}$$

The assumptions that the complete state $\mathbf{x}$ and the bound $\mathcal{W}$ are known are relaxed in Chapter 3, where the problem is extended to an output feedback case and to derive the uncertainty bound from online measurements.

The control is required to keep an output $\mathbf{y}(k) \in \Re^{N_y}$ within a bounded set for all disturbances. The form of the output constraints

$$\begin{aligned} \mathbf{y}(k) &= \mathbf{C}\mathbf{x}(k) + \mathbf{D}\mathbf{u}(k) && (2.3) \\ \mathbf{y}(k) &\in \mathcal{Y} \subset \Re^{N_y}, \ \forall k && (2.4) \end{aligned}$$

can capture both input and state constraints, or mixtures thereof, such as limited control magnitude or error box limits [28]. The matrices $\mathbf{C}$ and $\mathbf{D}$ and the set $\mathcal{Y}$ are all chosen by the designer. The objective is to minimize the cost function

$$J = \sum_{k=0}^{\infty} \ell\left(\mathbf{u}(k), \mathbf{x}(k)\right) \tag{2.5}$$

where $\ell(\cdot)$ is a stage cost function. Typically, this would be a quadratic function, resulting in a quadratic program solution, or a convex piecewise linear function (*e.g.* $|\mathbf{u}| + |\mathbf{x}|$) that can be implemented with slack variables in a linear program [29]. The robust feasibility results require no assumptions concerning the nature of this cost, but assumptions are added in Section 2.4 to enable the robust convergence and completion results.

Definition (Robust Feasibility) Assume that the MPC optimization problem can be solved from the initial state of the system. Then the system is *robustly-feasible* if, for all future disturbances $\mathbf{w}(k) \in \mathcal{W}\ \forall k \geq 0$, the optimization problem can be solved at every subsequent step.

Remark 2.1. (Robust Feasibility and Constraint Satisfaction) Robust feasibility is a stronger condition than robust constraint satisfaction, *i.e.* the satisfaction of the constraints at all times for all disturbances within the chosen bound. Feasibility requires constraint satisfaction as the constraints are applied to the first step of the plan, hence robust feasibility implies robust constraint satisfaction. However, in some cases there may be states which satisfy the constraints but from which no feasible plan can be found, *e.g.* if a vehicle is going so fast it will violate its position constraint at the following time step. The formulations in this thesis all provide robust feasibility and therefore robust constraint satisfaction as well.

2.3 Robustly-Feasible MPC

The online optimization approximates the complete problem in Section 2.2 by solving it over a finite horizon of N steps. A control-invariant terminal set constraint is applied to ensure stability [11]. Predictions are made using the nominal system model, *i.e.* (2.1) and (2.3) without the disturbance term. The output constraints are tightened using the method presented in [14], retaining a margin based on a particular candidate policy.

Optimization Problem Define the MPC optimization problem $\mathsf{P}(\mathbf{x}(k), \mathcal{Y}, \mathcal{W})$

$$J^*(\mathbf{x}(k)) = \min_{\mathbf{u},\mathbf{x},\mathbf{y}} \sum_{j=0}^{N} \ell\left(\mathbf{u}(k+j|k), \mathbf{x}(k+j|k)\right) \tag{2.6}$$

subject to $\forall j \in \{0 \ldots N\}$

$$
\begin{aligned}
\mathbf{x}(k+j+1|k) &= \mathbf{A}\mathbf{x}(k+j|k) + \mathbf{B}\mathbf{u}(k+j|k) && \text{(2.7a)}\\
\mathbf{y}(k+j|k) &= \mathbf{C}\mathbf{x}(k+j|k) + \mathbf{D}\mathbf{u}(k+j|k) && \text{(2.7b)}\\
\mathbf{x}(k|k) &= \mathbf{x}(k) && \text{(2.7c)}\\
\mathbf{x}(k+N+1|k) &\in \mathcal{X}_F && \text{(2.7d)}\\
\mathbf{y}(k+j|k) &\in \mathcal{Y}(j) && \text{(2.7e)}
\end{aligned}
$$

where the double indices $(k+j|k)$ denote the prediction made at time k of a value at time $k+j$. The constraint sets are defined by the following recursion, which is the extension of the constraint form in Ref. [14] to include a time-varying $\mathbf{K}$

$$
\begin{aligned}
\mathcal{Y}(0) &= \mathcal{Y} && \text{(2.8a)}\\
\mathcal{Y}(j+1) &= \mathcal{Y}(j) \sim (\mathbf{C} + \mathbf{D}\mathbf{K}(j))\,\mathbf{L}(j)\mathcal{W}, \quad \forall j \in \{0 \ldots N-1\} && \text{(2.8b)}
\end{aligned}
$$

where $\mathbf{L}(j)$ is defined as the state transition matrix for the closed-loop system under a candidate control law $\mathbf{u}(j) = \mathbf{K}(j)\mathbf{x}(j)$ $j \in \{0 \ldots N-1\}$

$$
\begin{aligned}
\mathbf{L}(0) &= \mathbf{I} && \text{(2.9a)}\\
\mathbf{L}(j+1) &= (\mathbf{A} + \mathbf{B}\mathbf{K}(j))\,\mathbf{L}(j), \quad \forall j \in \{0 \ldots N-1\} && \text{(2.9b)}
\end{aligned}
$$

Remark 2.6 describes a method to compare different choices of $\mathbf{K}(j)$, and Remark 2.8 discusses the particular significance of choosing $\mathbf{K}(j)$ to render the system nilpotent. The operator '$\sim$' denotes the Pontryagin difference [28]

$$\mathcal{A} \sim \mathcal{B} \triangleq \{\mathbf{a} \mid \mathbf{a} + \mathbf{b} \in \mathcal{A}, \ \forall \mathbf{b} \in \mathcal{B}\} \qquad (2.10)$$

This definition leads to the following important property, which will be used in the proof of robust feasibility

$$\mathbf{a} \in (\mathcal{A} \sim \mathcal{B}), \mathbf{b} \in \mathcal{B} \Rightarrow (\mathbf{a} + \mathbf{b}) \in \mathcal{A} \qquad (2.11)$$

The matrix mapping of a set is defined such that

$$\mathbf{A}\mathcal{X} \triangleq \{\mathbf{z} \mid \exists \mathbf{x} \in \mathcal{X} : \mathbf{z} = \mathbf{A}\mathbf{x}\} \tag{2.12}$$

A Matlab™ toolbox for performing the necessary operations, in particular the calculation of the Pontryagin difference, for polyhedral sets is available [30]. The calculation of the sets in (2.8) can be done offline.

There is some flexibility in the choice of terminal constraint set $\mathcal{X}_F$. It is found by the following Pontryagin difference

$$\mathcal{X}_F = \mathcal{R} \sim \mathbf{L}(N)\mathcal{W} \tag{2.13}$$

where $\mathcal{R}$ is a robust control invariant admissible set [31] *i.e.* there exists a control law $\kappa(\mathbf{x})$ satisfying the following

$$\forall \mathbf{x} \in \mathcal{R} \quad \mathbf{A}\mathbf{x} + \mathbf{B}\kappa(\mathbf{x}) + \mathbf{L}(N)\mathbf{w} \in \mathcal{R}, \ \forall \mathbf{w} \in \mathcal{W} \tag{2.14a}$$

$$\mathbf{C}\mathbf{x} + \mathbf{D}\kappa(\mathbf{x}) \in \mathcal{Y}(N) \tag{2.14b}$$

Remark 2.2. (Choice of Terminal Constraint Set) A variety of methods can be employed to calculate a suitable terminal set $\mathcal{X}_F$, based on identifying different sets $\mathcal{R}$ satisfying (2.14). For the greatest feasible set, use the Maximal Robust Control Invariant set [31], the largest set for which conditions (2.14) can be satisfied by any nonlinear feedback κ. If it is desired to converge to a particular target control, use the Maximal Robust Output Admissible set [28], the largest set satisfying (2.14) for a particular choice of controller $\kappa(\mathbf{x}) = \mathbf{K}\mathbf{x}$. Remark 2.8 discusses additional possibilities that arise if $\mathbf{K}(j)$ is chosen to render the system nilpotent.

Remark 2.3. (Comparison with Ref. [14]) In Ref. [14], the terminal constraint set is required to be invariant under the same constant, stabilizing LTI control law used to perform the constraint tightening. This is equivalent to restricting $\kappa(\mathbf{x}) = \mathbf{K}\mathbf{x}$ and $\mathbf{K}(j) = \mathbf{K}$ for some stabilizing $\mathbf{K}$.

This completes the description of the optimization problem and its additional design parameters. The following algorithm summarizes its implementation.

Algorithm 2.1. (Robustly Feasible MPC)

1. Solve problem $\mathsf{P}(\mathbf{x}(k), \mathcal{Y}, \mathcal{W})$

2. Apply control $\mathbf{u}(k) = \mathbf{u}^*(k|k)$ from the optimal sequence

3. Increment k. Go to Step 1

Theorem 2.1. ***(Robust Feasibility)*** *If* $\mathsf{P}\left(\mathbf{x}(0), \mathcal{Y}, \mathcal{W}\right)$ *has a feasible solution then the system (2.1), subjected to disturbances obeying (2.2) and controlled using Algorithm 2.1, is robustly-feasible.*

Proof: It is sufficient to prove that if at time k_0 the problem $\mathsf{P}(\mathbf{x}(k_0), \mathcal{Y}, \mathcal{W})$ is feasible for some state $\mathbf{x}(k_0)$ and control $\mathbf{u}^*(k_0|k_0)$ is applied then the next problem $\mathsf{P}(\mathbf{x}(k_0 + 1), \mathcal{Y}, \mathcal{W})$ is feasible for all disturbances $\mathbf{w}(k_0) \in \mathcal{W}$. The theorem then follows by recursion: if feasibility at time k_0 implies feasibility at time $k_0 + 1$ then feasibility at time 0 implies feasibility at all subsequent steps. A feasible solution is assumed for time k_0 and used to construct a candidate solution for time $k_0 + 1$, which is then shown to satisfy the constraints for time $k_0 + 1$.

Assume $\mathsf{P}(\mathbf{x}(k_0), \mathcal{Y}, \mathcal{W})$ is feasible. Then it has a feasible (not necessarily optimal) solution, denoted by *, with states $\mathbf{x}^*(k_0 + j|k_0),\ j \in \{0, \ldots, N + 1\}$, inputs $\mathbf{u}^*(k_0 + j|k_0),\ j \in \{0, \ldots, N\}$ and outputs $\mathbf{y}^*(k_0 + j|k_0),\ j \in \{0, \ldots, N\}$ satisfying all of the constraints (2.7). Now, to prove feasibility of the subsequent optimization, a candidate solution is constructed and then shown to satisfy the constraints. Consider the following candidate solution, denoted by $\hat{\cdot}$, for problem $\mathsf{P}(\mathbf{x}(k_0 + 1), \mathcal{Y}, \mathcal{W})$

$$\begin{aligned} \hat{\mathbf{u}}(k_0 + j + 1|k_0 + 1) &= \mathbf{u}^*(k_0 + j + 1|k_0) \\ &+ \mathbf{K}(j)\mathbf{L}(j)\mathbf{w}(k_0), \quad \forall j \in \{0 \ldots N - 1\} \end{aligned} \tag{2.15a}$$

$$\hat{\mathbf{u}}(k_0 + N + 1|k_0 + 1) = \kappa\left(\hat{\mathbf{x}}(k_0 + N + 1|k_0 + 1)\right) \tag{2.15b}$$

$$\hat{\mathbf{x}}(k_0 + j + 1|k_0 + 1) = \mathbf{x}^*(k_0 + j + 1|k_0) \tag{2.15c}$$

$$
\begin{aligned}
& + \quad \mathbf{L}(j)\mathbf{w}(k_0), \quad \forall j \in \{0 \ldots N\} \\
\hat{\mathbf{x}}(k_0+N+2|k_0+1) &= \mathbf{A}\hat{\mathbf{x}}(k_0+N+1|k_0+1) \qquad (2.15d) \\
& + \quad \mathbf{B}\kappa(\hat{\mathbf{x}}(k_0+N+1|k_0+1)) \\
\hat{\mathbf{y}}(k_0+j+1|k_0+1) &= \mathbf{C}\hat{\mathbf{x}}(k_0+j+1|k_0+1) \qquad (2.15e) \\
& + \quad \mathbf{D}\hat{\mathbf{u}}(k_0+j+1|k_0+1), \quad \forall j \in \{0 \ldots N\}
\end{aligned}
$$

This solution is formed by shifting the previous solution by one step, *i.e.* removing the first step, adding one step of the invariant control law κ at the end, and adding perturbations representing the rejection of the disturbance by the candidate controller $\mathbf{K}$ and the associated state transition matrices $\mathbf{L}$. The construction of this candidate solution also illustrates the generalization compared to Ref. [14]: the controller $\mathbf{K}(j)$ may be time varying and the terminal law $\kappa(\mathbf{x})$ is a general nonlinear law, known to exist, according to (2.14) but not necessarily known explicitly.

To prove that the candidate solution in (2.15) is feasible, it is necessary to show that it satisfies all of the constraints (2.7a)–(2.7e) at timestep $k = k_0 + 1$ for any disturbance $\mathbf{w}(k_0) \in \mathcal{W}$.

Dynamics constraints (2.7a): Since the previous plan satisfied (2.7a), we know

$$
\mathbf{x}^*(k_0+j+2|k_0) = \mathbf{A}\mathbf{x}^*(k_0+j+1|k_0)+\mathbf{B}\mathbf{u}^*(k_0+j+1|k_0), \ \forall j \in \{0 \ldots N-1\} \qquad (2.16)
$$

Substituting on the both sides for $\mathbf{x}^*$ and $\mathbf{u}^*$ from the definitions of the candidate solution (2.15a) and (2.15c)

$$
\begin{aligned}
[\hat{\mathbf{x}}(k_0+j+2|k_0+1) - \mathbf{L}(j+1)\mathbf{w}(k_0)] &= \mathbf{A}[\hat{\mathbf{x}}(k_0+j+1|k_0+1) - \mathbf{L}(j)\mathbf{w}(k_0)] \\
& + \quad \mathbf{B}[\hat{\mathbf{u}}(k_0+j+1|k_0+1) - \mathbf{K}(j)\mathbf{L}(j)\mathbf{w}(k_0)] \\
&= \mathbf{A}\hat{\mathbf{x}}(k_0+j+1|k_0+1) \qquad (2.17) \\
& + \quad \mathbf{B}\hat{\mathbf{u}}(k_0+j+1|k_0+1) \\
& - \quad (\mathbf{A}+\mathbf{B}\mathbf{K}(j))\mathbf{L}(j)\mathbf{w}(k_0)
\end{aligned}
$$

Then using (2.9b), the definition of the state transition matrices $\mathbf{L}$, the last term on each side cancels leaving

$$\hat{\mathbf{x}}(k_0+j+2|k_0+1) = \mathbf{A}\hat{\mathbf{x}}(k_0+j+1|k_0+1)+\mathbf{B}\hat{\mathbf{u}}(k_0+j+1|k_0+1),\ \forall j \in \{0\ldots N-1\} \tag{2.18}$$

identical to the dynamics constraint (2.7a) for steps $j \in \{0\ldots N-1\}$. The final step of the candidate plan (2.15b) and (2.15d) satisfy the dynamics model by construction, hence (2.7a) is satisfied for steps $j \in \{0\ldots N\}$.

System constraints (2.7b): the candidate outputs (2.15e) are constructed using the output constraints (2.7b) and therefore satisfy them by construction.

Initial constraint (2.7c): The true state at time k_0+1 is found by applying control $\mathbf{u}^*(k_0|k_0)$ and disturbance $\mathbf{w}(k_0)$ to the dynamics (2.1)

$$\mathbf{x}(k_0+1) = \mathbf{A}\mathbf{x}(k_0) + \mathbf{B}\mathbf{u}^*(k_0|k_0) + \mathbf{w}(k_0) \tag{2.19}$$

Compare this equation with the constraints (2.7a) for step $j=0$ at time $k=k_0$

$$\begin{aligned} \mathbf{x}^*(k_0+1|k_0) &= \mathbf{A}\mathbf{x}(k_0|k_0) + \mathbf{B}\mathbf{u}^*(k_0|k_0) \\ &= \mathbf{A}\mathbf{x}(k_0) + \mathbf{B}\mathbf{u}^*(k_0|k_0) \end{aligned} \tag{2.20}$$

Then subtracting (2.20) from (2.19) shows that the new state can be expressed as a perturbation from the planned next state

$$\mathbf{x}(k_0+1) = \mathbf{x}^*(k_0+1|k_0) + \mathbf{w}(k_0) \tag{2.21}$$

Substituting $\mathbf{L}(0) = \mathbf{I}$ from (2.9a) into (2.15c) with $j=0$ gives

$$\hat{\mathbf{x}}(k_0+1|k_0+1) = \mathbf{x}^*(k_0+1|k_0) + \mathbf{w}(k_0) \tag{2.22}$$

From (2.21) and (2.22) we see that $\hat{\mathbf{x}}(k_0+1|k_0+1) = \mathbf{x}(k_0+1)$, satisfying (2.7c).

Terminal constraint (2.7d): substituting into (2.15c) for $j = N$ gives

$$\hat{\mathbf{x}}(k_0 + N + 1|k_0 + 1) = \mathbf{x}^*(k_0 + N + 1|k_0) + \mathbf{L}(N)\mathbf{w}(k)$$

Feasibility at time k_0 requires $\mathbf{x}^*(k_0+N+1|k_0) \in \mathcal{X}_F$ according to (2.7d) and therefore, using the property 2.11 of the Pontryagin difference (2.13) defining $\mathcal{X}_F$, we have

$$\hat{\mathbf{x}}(k_0 + N + 1|k_0 + 1) \in \mathcal{R} \tag{2.23}$$

The invariance condition (2.14a) ensures

$$\mathbf{A}\hat{\mathbf{x}}(k_0 + N + 1|k_0 + 1) + \mathbf{B}\kappa(\hat{\mathbf{x}}(k_0 + N + 1|k_0 + 1)) + \mathbf{L}(N)\mathbf{w} \in \mathcal{R},\ \forall \mathbf{w} \in \mathcal{W} \tag{2.24}$$

which from (2.15d) implies

$$\hat{\mathbf{x}}(k_0 + N + 2|k_0 + 1) + \mathbf{L}(N)\mathbf{w} \in \mathcal{R},\ \forall \mathbf{w} \in \mathcal{W} \tag{2.25}$$

Using the definition of the Pontryagin difference 2.10 and the terminal set (2.13) this shows

$$\hat{\mathbf{x}}(k_0 + N + 2|k_0 + 1) \in \mathcal{X}_F \tag{2.26}$$

which satisfies the terminal constraint (2.7d) for time $k = k_0 + 1$.

Output constraints (2.7e): Begin by testing steps $j = 0 \dots N-1$ of the candidate solution. Substituting the state and control perturbations (2.15c) and (2.15a) into the output definition (2.15e) gives

$$\hat{\mathbf{y}}((k_0+1)+j|k_0+1) = \mathbf{y}^*(k_0+(j+1)|k_0)+(\mathbf{C}+\mathbf{D}\mathbf{K}(j))\mathbf{L}(j)\mathbf{w}(k_0),\ \ \forall j \in \{0 \dots N-1\} \tag{2.27}$$

Also, given feasibility at time k_0, we know

$$\mathbf{y}^*(k_0 + (j + 1)|k_0) \in \mathcal{Y}(j + 1),\ \forall j \in \{0 \dots N - 1\} \tag{2.28}$$

Recall the definition of the constraint sets (2.8b)

$$\mathcal{Y}(j+1) = \mathcal{Y}(j) \sim (\mathbf{C} + \mathbf{DK}(j))\,\mathbf{L}(j)\mathcal{W}, \quad \forall j \in \{0 \dots N-1\}$$

Now the property of the Pontryagin difference (2.11) can be applied to (2.27) and (2.28)

$$\begin{gathered} \mathbf{y}^*(k_0+j+1|k_0) \in \mathcal{Y}(j+1) \Rightarrow \\ \hat{\mathbf{y}}(k_0+j+1|k_0+1) = \mathbf{y}^*(k_0+j+1|k_0) + (\mathbf{C}+\mathbf{DK}(j))\mathbf{L}(j)\mathbf{w}(k_0) \in \mathcal{Y}(j), \\ \forall \mathbf{w}(k_0) \in \mathcal{W},\ \forall j \in \{0 \dots N-1\} \end{gathered}$$

proving that the first N steps of the candidate outputs (2.15e) satisfy the constraints (2.7e) at time $k = k_0 + 1$. Also, it follows from (2.23) and the admissibility requirement (2.14b) that the final control step using $\kappa(\hat{\mathbf{x}}(k{+}N{+}1|k{+}1))$ is admissible for set $\mathcal{Y}(N)$ according to (2.14b).

Having shown that the candidate solution (2.15) satisfies the constraints at time k_0+1, given that a solution for time k_0 exists, then feasibility at time k_0 must imply feasibility at time k_0+1, and hence at all future times, by recursion. $\square$

Fig. 2-1 illustrates for a simple example how the constraint tightening in (2.8b) allows margin for feedback compensation in response to the disturbance. Here, an output y is constrained to be less than or equal to $Y_{\max}$ at all times. A two-step nilpotent control policy has been chosen as $\mathbf{K}(j)$ and the plan constraints are tightened after the first and second steps accordingly. At time k, the planned output signal $y(\cdot|k)$ must remain beneath the constraint line marked for time k. Suppose the plan at time k is right against this boundary. The first step is executed and the same optimization is solved at time $k+1$, but now subject to the constraint boundary marked for time $k+1$. The upward arrows show the margin returned to the optimization at step $k+1$: the nilpotent control policy can be added to the previous plan (the line of constraint for time k) resulting in a new plan that would satisfy the new constraints *i.e.* below the $k+1$ constraint line. The downward arrows show the margin retained for later steps.

The decision variables for the robust problem are a single sequence of controls,

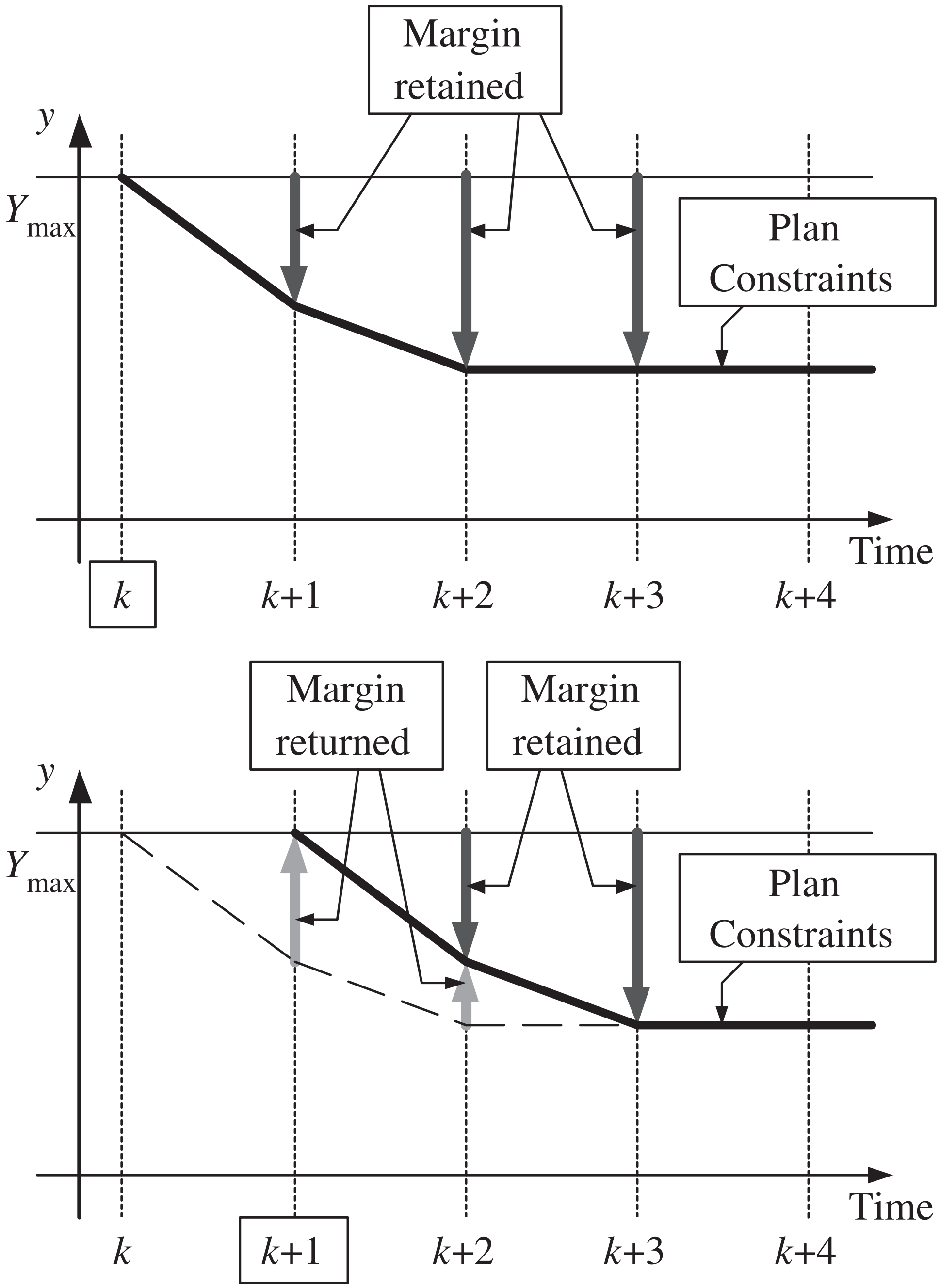


Fig. 2-1: Illustration of Constraint Tightening for Robustness

states and outputs extending over the horizon. The number of variables grows linearly with horizon length. If the original constraint set $\mathcal{Y}$ and the disturbance set $\mathcal{W}$ are polyhedra then the modified constraint sets $\mathcal{Y}(j)$ are also polyhedra [28, 31] and the number of constraints required to specify each set $\mathcal{Y}(j)$ is the same as that required to specify the nominal set $\mathcal{Y}$ [31]. Therefore the type and size of the optimization is unchanged by the constraint modifications. For example, if the nominal optimal control problem, ignoring uncertainty, is a linear program (LP), the robust MPC optimization is also an LP of the same size.

Remark 2.4. (Anytime Computation) The result in Theorem 2.1 depends only on finding *feasible* solutions to each optimization, not necessarily *optimal* solutions. Furthermore, the robust feasibility result follows from the ability to construct a feasible solution to each problem before starting the optimization. This brings two advantages, first that this solution may be used to initialize a search procedure, and second that the optimization can be terminated at any time knowing that a feasible solution exists and can be used for control. Then, for example, an interior point method [33] can be employed, initialized with the feasible candidate solution.

Remark 2.5. (Reparameterizing the Problem) Some optimal control approaches [32] simplify the computation by reparameterizing the control in terms of basis functions, reducing the number of degrees of freedom involved. It is possible to combine this basis function approach with constrait tightening for robustness as developed in this Chapter. Robust feasibility of the reparameterized optimization can be assured by expressing the control sequence as a perturbation from the candidate solution. This is equivalent to substituting the following expression for the control sequence

$$\mathbf{u}(k+j|k) = \hat{\mathbf{u}}(k+j|k) + \sum_{n=1}^{N_b} \alpha_n b_n(j)$$

where $\hat{\mathbf{u}}(k+j|k)$ is the candidate solution from (2.15), $b_n(j)$ denotes a set of N_b basis functions, defined over the horizon $j = 0 \dots N$ and chosen by the designer, and α_n are the new decision variables in the problem. Since the candidate solution is known to be feasible, $\alpha_n = 0\ \forall n$ is a feasible solution to the revised problem.

Remark 2.6. (Comparing Candidate Policies) The choice of the candidate control policy $\mathbf{K}$ is an important parameter in the robustness scheme. One way of comparing possible choices of $\mathbf{K}$ is to evaluate the largest disturbance that can be accommodated by the scheme using that candidate policy. Assuming the disturbance set (2.2) is a mapped unit hypercube $\mathcal{G} = \{\mathbf{v} \mid \|\mathbf{v}\|_\infty \leq 1\}$ with a variable scaling δ

$$\mathcal{W} = \delta \mathbf{E} \mathcal{G} \tag{2.29}$$

then it is possible to calculate the largest value of δ for which the set of feasible initial states is non-empty by finding the largest δ for which the smallest constraint set $\mathcal{Y}(N)$ is non-empty. Given the property (2.11) of the Pontryagin difference, this can be found by solving the following optimization

$$\begin{array}{rclrcl} \delta_{\max} & = & \max\limits_{\delta,\mathbf{p}} \delta & & & \\ \text{s.t.} \mathbf{p} + \delta\mathbf{q} & \in & \mathcal{Y} & \forall \mathbf{q} & \in & (\mathbf{C}+\mathbf{DK}(0))\mathbf{L}(0)\mathbf{E}\mathcal{G} \oplus \\ & & & & & (\mathbf{C}+\mathbf{DK}(1))\mathbf{L}(1)\mathbf{E}\mathcal{G} \oplus \ldots \oplus \\ & & & & & (\mathbf{C}+\mathbf{DK}(N-1))\mathbf{L}(N-1)\mathbf{E}\mathcal{G} \end{array} \tag{2.30}$$

where $\oplus$ denotes the Minkowski or vector sum of two sets. Since the vertices of $\mathcal{G}$ are known, the vertices of the Minkowski sum can be easily found as the sum of all cominbations of vertices of the individual sets [31]. The number of constraints can be large but this calculation is performed offline for controller analysis, so solution time is not critical. This calculation is demonstrated for the examples in Section 2.5 and shown to enable comparison of the "size" of the sets of feasible initial conditions when using different candidate controllers.

Remark 2.7. (Approximate Constraint Sets) In some cases it may be preferable to use approximations of the constraint sets (2.8) rather than calculating the Pontryagin difference exactly. The proof of robust feasibility in Theorem 1 depends on the property (2.11) of the Pontryagin difference. This property is a weaker condition than the definition of the Pontryagin difference (2.10), which is the *largest* set obeying (2.11). Therefore, the constraint set recursion in (2.8) can be replaced by any sequence of sets for which the following property holds

$$\mathbf{y} + (\mathbf{C} + \mathbf{D}\mathbf{K}(j))\,\mathbf{L}(j)\mathbf{w} \in \mathcal{Y}(j),\ \forall \mathbf{y} \in \mathcal{Y}(j+1),\ \forall \mathbf{w} \in \mathcal{W},\ \forall j \in \{0 \dots N-1\}$$

This test can be accomplished using norm bounds, for example.

Remark 2.8. (Significance of Nilpotency) If the candidate control is restricted to be nilpotent, then the final state transition matrix $\mathbf{L}(N) = \mathbf{0}$ according to (2.9). Referring back to the requirements of the terminal set (2.14a), (2.14b) and (2.13), this means that the set $\mathcal{R}$ can be a *nominally* control invariant set and $\mathcal{X}_F = \mathcal{R}$. Nominal control invariance is a weaker condition than robust control invariance, and in some cases, a nominally-invariant set can be found when no robustly-invariant set exists. For example, a nominal control invariant set can have no volume, but this is impossible for a robust control invariant set. In the spacecraft control example in Section 2.5.2, a coasting ellipse or "passive aperture" is a particularly attractive terminal set and is shown to offer good performance. However, a passive aperture is only a nominally invariant set, not robustly invariant, so the candidate control must be nilpotent if the passive aperture terminal constraint is used.

Remark 2.9. (Nilpotent Controllers) Constant nilpotent controllers can be synthesized by using pole-placement techniques to put the closed-loop poles at the origin. For a controllable system of order M, this generates a constant controller $\mathbf{K}(j) = \mathbf{K}\ \forall j$ that guarantees convergence of the state to the origin in at most N_x steps, *i.e.* $\mathbf{L}(N_x) = \mathbf{0}$. This satisfies the requirement that $\mathbf{L}(N) = \mathbf{0}$ provided $N \geq N_x$. Greater flexibility in the choice of $\mathbf{K}(j)$ can be achieved by using a time-varying controller. A finite-horizon (M-step) LQR policy with an infinite terminal cost is suitable, found by solving

$$\begin{aligned}
\mathbf{P}(M) &= \infty\mathbf{I} \\
\forall j \in \{1 \dots M\}\ \mathbf{P}(j-1) &= \mathbf{Q} + \mathbf{A}^T\mathbf{P}(j)\mathbf{A} - \mathbf{A}^T\mathbf{P}(j)\mathbf{B}(\mathbf{R} + \mathbf{B}^T\mathbf{P}(j)\mathbf{B})^{-1}\mathbf{B}^T\mathbf{P}(j)\mathbf{A} \\
\mathbf{K}_L(j-1) &= -(\mathbf{R} + \mathbf{B}^T\mathbf{P}(j)\mathbf{B})^{-1}\mathbf{B}^T\mathbf{P}(j)\mathbf{A}
\end{aligned}$$

Most generally, a general M-step nilpotent policy, applying control at some steps

$\mathcal{J} \subset \{0 \ldots M-1\}$ with $N_x \leq M \leq N$, can be found by solving the state transition equation for the inputs $\mathbf{u}(j)$ $j \in \mathcal{J}$

$$\mathbf{0} = \mathbf{A}^M \mathbf{x}(0) + \sum_{j \in \mathcal{J}} \mathbf{A}^{(M-j-1)} \mathbf{B} \mathbf{u}(j) \tag{2.31}$$

If the system is controllable, this equation can be solved for a set of controls as a function of the initial state $\mathbf{u}(j) = \mathbf{H}(j)\mathbf{x}(0)$. This can be re-arranged into the desired form $\mathbf{u}(j) = \mathbf{K}(j)\mathbf{x}(j)$.

Proposition 2.1. ***(Repeating Trajectories as Terminal Constraints)*** *For any system, an admissible trajectory that repeats after some period, defined by a set*

$$\mathcal{X}_F = \{\mathbf{x} \mid \mathbf{A}^{N_R}\mathbf{x} = \mathbf{x},\ \mathbf{C}\mathbf{A}^j\mathbf{x} \in \mathcal{Y}(N),\ \forall j = 0 \ldots (N_R - 1)\} \tag{2.32}$$

where N_R is the chosen period of repetition, is nominally control invariant, i.e. *satisfies (2.14b) and (2.14a), under the policy $\kappa(\mathbf{x}) = \mathbf{0}$ $\forall \mathbf{x}$*

Proof: see Section 2.A.1.

2.4 Robust Convergence

For some control problems, robust feasibility and constraint satisfaction are sufficient to meet the objectives. This section provides an additional, stronger result, showing convergence to a smaller bounded set within the constraints. A typical application would be dual-mode MPC [16], in which MPC is used to drive the state into a certain set, obeying constraints during this transient period, before switching to another, steady-state stabilizing controller.

The result in this section proves that the state is guaranteed to enter a set $\mathcal{X}_C$. This set is not the same as the terminal constraint set $\mathcal{X}_F$. Nor does the result guarantee that the set $\mathcal{X}_C$ in invariant, *i.e.* the state may leave this set, but it cannot remain outside it for all time. Some other forms of robust MPC *e.g.* [26, 12] guarantee invariance of their terminal constraint sets, but they have stronger restrictions on the choice of terminal constraints. The variable-horizon MPC formulation in Section 2.6 also offers a robust set arrival result. In that case, the target set $\mathcal{Q}$ is chosen freely by the user, and not restricted to the form of $\mathcal{X}_C$ dictated below, but the variable horizon optimization is more complicated to implement.

First, an assumption on the form of the cost function is necessary. This is not restrictive in practice because the assumed form is a common standard.

Assumption (Stage Cost) The stage cost function $\ell(\cdot)$ in (2.6) is of the form

$$\ell(\mathbf{u}, \mathbf{x}) = \|\mathbf{u}\|_{\mathbf{R}} + \|\mathbf{x}\|_{\mathbf{Q}} \tag{2.33}$$

where $\|\cdot\|_{\mathbf{R}}$ and $\|\cdot\|_{\mathbf{Q}}$ denote weighted norms. Typical choices would be quadratic forms $\mathbf{u}^T\mathbf{R}\mathbf{u}$ or weighted one-norms $|\mathbf{R}\mathbf{u}|$.

Theorem 2.2. ***(Robust Convergence)*** *If* $\mathsf{P}\left(\mathbf{x}(0), \mathcal{Y}, \mathcal{W}\right)$ *has a feasible solution, the optimal solution is found at each time step, and the objective function is of the form (2.33), then the state is guaranteed to enter the following set*

$$\mathcal{X}_C = \{\mathbf{x} \in \Re^{N_x} \mid \|\mathbf{x}\|_{\mathbf{Q}} \leq \alpha + \beta\} \tag{2.34}$$

where α *and* β *are given by*

$$\alpha = \max_{\mathbf{w} \in \mathcal{W}} \sum_{i=0}^{N} \left(\|\mathbf{K}(i)\mathbf{L}(i)\mathbf{w}\|_{\mathbf{R}} + \|\mathbf{L}(i)\mathbf{w}\|_{\mathbf{Q}}\right) \tag{2.35}$$

$$\beta = \max_{\mathbf{x} \in \mathcal{X}_F} \left(\|\mathbf{x}\|_{\mathbf{Q}} + \|\kappa(\mathbf{x})\|_{\mathbf{R}}\right) \tag{2.36}$$

These represent, respectively, the maximum cost of the correction policy for any disturbance $\mathbf{w} \in \mathcal{W}$ and the maximum cost of remaining in the terminal set for one step from any state $\mathbf{x} \in \mathcal{X}_F$. Note that the common, albeit restrictive, choice of the origin as the terminal set $\mathcal{X}_F = \{\mathbf{0}\}$ yields $\beta = 0$.

Proof: Theorem 2.1 showed that if a solution exists for problem $\mathsf{P}(\mathbf{x}(k_0), \mathcal{Y}, \mathcal{W})$ then a particular candidate solution (2.15) is feasible for the subsequent problem $\mathsf{P}(\mathbf{x}(k_0 + 1), \mathcal{Y}, \mathcal{W})$. The cost of this candidate solution $\hat{J}(\mathbf{x}(k_0 + 1))$ can be used to bound the optimal cost at a particular time step $J^*(\mathbf{x}(k_0+1))$ relative to the optimal cost at the preceding time step $J^*(\mathbf{x}(k_0))$. A Lyapunov-like argument based on this function will then be used to prove entry to the target set $\mathcal{X}_C$. Since the theorem requires optimal solutions to be found at each step, redefine the superscript * to denote the optimal solution, giving the following cost for the optimal solution at time k_0

$$J^*(\mathbf{x}(k_0)) = \sum_{i=0}^{N} \{\|\mathbf{u}^*(k_0+j|k_0)\|_{\mathbf{R}} + \|\mathbf{x}^*(k_0+j|k_0)\|_{\mathbf{Q}}\} \tag{2.37}$$

and the cost of the candidate solution in (2.15)

$$\begin{aligned}
\hat{J}(\mathbf{x}(k_0+1)) &= \sum_{j=0}^{N} \{\|\hat{\mathbf{u}}(k_0+j+1|k_0+1)\|_{\mathbf{R}} + \|\hat{\mathbf{x}}(k_0+j+1|k_0+1)\|_{\mathbf{Q}}\} && (2.38)\\
&= \sum_{j=0}^{N-1} \{\|\mathbf{u}^*(k_0+j+1|k_0) + \mathbf{K}(j)\mathbf{L}(j)\mathbf{w}(k_0)\|_{\mathbf{R}} + \\
&\qquad \|\mathbf{x}^*(k_0+j+1|k_0) + \mathbf{L}(j)\mathbf{w}(k_0)\|_{\mathbf{Q}}\} + && (2.39)\\
&\qquad \|\mathbf{x}^*(k_0+N+1|k_0)\|_{\mathbf{Q}} + \|\kappa(\mathbf{x}^*(k_0+N+1|k_0))\|_{\mathbf{R}}
\end{aligned}$$

The cost of the candidate solution can be bounded from above using the triangle inequality, applied to each term in the summation

$$\begin{aligned}
\hat{J}(\mathbf{x}(k_0+1)) &\leq \sum_{j=0}^{N-1} \{\|\mathbf{u}^*(k_0+j+1|k_0)\|_{\mathbf{R}} + \|\mathbf{K}(j)\mathbf{L}(j)\mathbf{w}(k_0)\|_{\mathbf{R}} + \\
&\qquad \|\mathbf{x}^*(k_0+j+1|k_0)\|_{\mathbf{Q}} + \|\mathbf{L}(j)\mathbf{w}(k_0)\|_{\mathbf{Q}}\} + && (2.40)\\
&\qquad \|\mathbf{x}(k_0+N+1|k_0)\|_{\mathbf{Q}} + \|\kappa(\mathbf{x}(k_0+N+1|k_0))\|_{\mathbf{R}}
\end{aligned}$$

This bound can now be expressed in terms of the previous optimal cost in (2.37)

$$\begin{aligned}
\hat{J}(\mathbf{x}(k_0+1)) &\leq J^*(\mathbf{x}(k_0)) - \|\mathbf{u}(k_0)\|_{\mathbf{R}} - \|\mathbf{x}(k_0)\|_{\mathbf{Q}} + \\
&\qquad \sum_{i=0}^{N-1} \{\|\mathbf{K}(i)\mathbf{L}(i)\mathbf{w}(k_0)\|_{\mathbf{R}} + \|\mathbf{L}(i)\mathbf{w}(k_0)\|_{\mathbf{Q}}\} + && (2.41)\\
&\qquad \|\mathbf{x}^*(k_0+N+1|k_0)\|_{\mathbf{Q}} + \|\kappa(\mathbf{x}^*(k_0+N+1|k_0))\|_{\mathbf{R}}
\end{aligned}$$

The summation term is clearly bounded by the quantity α from (2.35). Also, since the constraints require $\mathbf{x}^*(k_0 + N + 1|k_0) \in \mathcal{X}_F$, the final two terms are bounded by the maximum (2.36). Finally, using the non-negativity of $\|\mathbf{u}(k_0)\|_\mathbf{R}$, (2.41) can be rewritten as

$$\hat{J}(\mathbf{x}(k_0 + 1)) \leq J^*(\mathbf{x}(k_0)) - \|\mathbf{x}(k_0)\|_\mathbf{Q} + \alpha + \beta \tag{2.42}$$

and since the cost of the candidate solution forms an upper bound on the optimal cost $J^*(\mathbf{x}(k_0 + 1)) \leq \hat{J}(\mathbf{x}(k_0 + 1))$, this implies that

$$J^*(\mathbf{x}(k_0 + 1)) - J^*(\mathbf{x}(k_0)) \leq -\|\mathbf{x}(k_0)\|_\mathbf{Q} + \alpha + \beta \tag{2.43}$$

Recall from (2.34) the definition of the convergence set $\mathcal{X}_C = \{\mathbf{x} \in \Re^{N_x} \mid \|\mathbf{x}\|_\mathbf{Q} \leq \alpha + \beta\}$. Therefore if the state lies outside this set, $\mathbf{x}(k_0) \notin \mathcal{X}_C$, the right-hand side of (2.43) is negative and the optimal cost is strictly decreasing. However, the optimal cost $J^*(\cdot)$ is bounded below, by construction, so the state $\mathbf{x}(k_0)$ cannot remain outside $\mathcal{X}_C$ forever and must enter $\mathcal{X}_C$ at some time. □

The smallest region of convergence $\mathcal{X}_C$ for a given system is attained by setting the target set $\mathcal{X}_F$ to be the origin, yielding $\beta = 0$, and weighting the state much higher than the control. This would be desirable for a scenario to get close to a particular target state. The set $\mathcal{X}_C$ becomes larger as the control weighting is increased or as the terminal set is enlarged. An example in the following section demonstrates the use of the convergence property for transient control subject to constraints using a dual-mode approach. A final property of the convergence set, that it can be no smaller than the uncertainty set, is captured in the following proposition.

Proposition 2.2. *The convergence set contains the disturbance set, $\mathcal{X}_C \supseteq \mathcal{W}$.*

Proof: See Section 2.A.2

2.5 Numerical Examples

The following subsections demonstrate the properties of the new formulation using numerical examples. The first uses a simple system in simulation to illustrate the effectiveness of the robust controller in comparison to nominal MPC. The second example considers steady-state spacecraft control, showing in particular how the choice of terminal constraint sets can change performance. The third set of examples use set visualization to verify robust feasibility. The final example shows the exploitation of the convergence result to enlarge the region of attraction of a stabilizing control scheme, subject to constraints. In all examples, set calculations were performed using the Invariant Set Toolbox for Matlab™ [30].

2.5.1 Robust Feasibility Demonstration

This section shows a very simple example comparing nominal MPC with the robustly-feasible formulation of Section 2.3 in simulation, including random but bounded disturbances. The system is a point mass moving in one dimension with force actuation, of up to one unit in magnitude, and a disturbance force of up to 30% of the control mgnitude. The control is required to keep both the position and the velocity within ± 1 while minimizing the control energy. In the notation of Section 2.2

$$\begin{aligned} \mathbf{A} &= \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} & \mathbf{B} &= \begin{bmatrix} 0.5 \\ 1 \end{bmatrix} & \mathcal{W} &= \{\mathbf{w} = \mathbf{B}z \mid |z| \leq 0.3\} \\ \mathbf{C} &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} & \mathbf{D} &= \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} & \mathcal{Y} &= \{\mathbf{y} \in \Re^3 \mid \|\mathbf{y}\|_\infty \leq 1\} \end{aligned}$$

The horizon was set to $N = 10$ steps. The Matlab implementation of Ackermann's formula was used to design a nilpotent controller by placing both closed-loop poles at $s = 0$. The resulting candidate controller was

$$\mathbf{K} = [-1 \ \ -1.5]$$

Using this controller in the expression (2.8) for constraint tightening gave the following constraint sets

$$\begin{array}{rcllllllll}
\mathcal{Y}(0) & = & \{\mathbf{y} \in \Re^3 & \mid & |y_1| \leq & 1 & |y_2| \leq & 1 & |y_3| \leq & 1 \quad \} \\
\mathcal{Y}(1) & = & \{\mathbf{y} \in \Re^3 & \mid & |y_1| \leq & 0.85 & |y_2| \leq & 0.7 & |y_3| \leq & 0.4 \quad \} \\
\mathcal{Y}(j) & = & \{\mathbf{y} \in \Re^3 & \mid & |y_1| \leq & 0.7 & |y_2| \leq & 0.4 & |y_3| \leq & 0.1 \quad \},\ \forall j = \{2 \dots 10\}
\end{array}$$

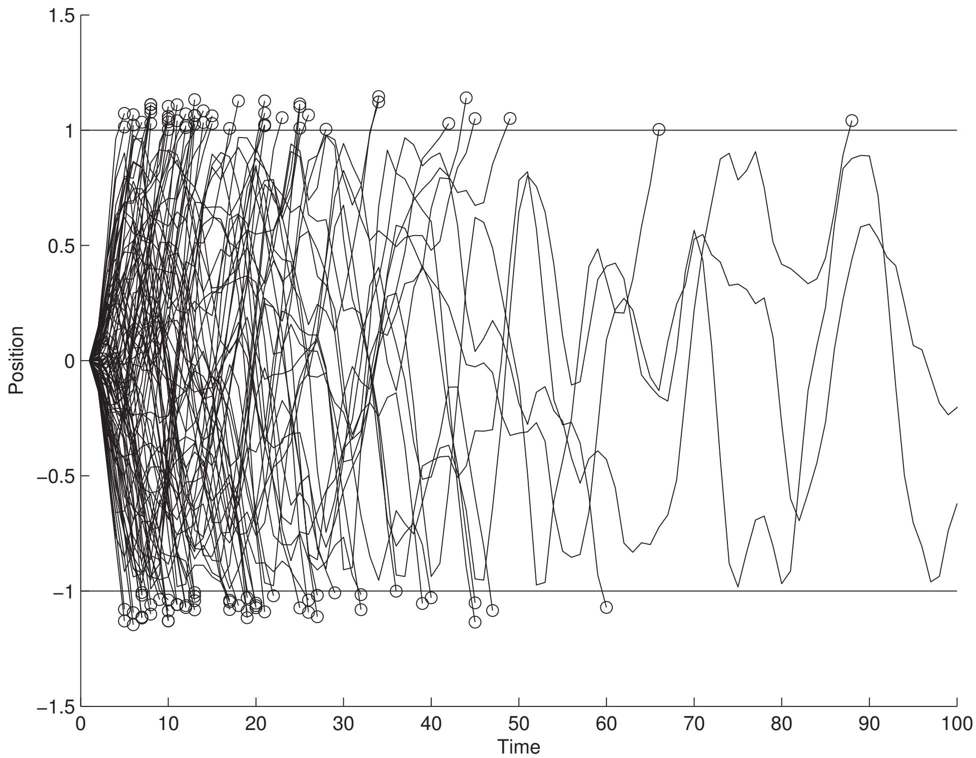
Note that since the controller is chosen to be nilpotent, the sets remain constant after a finite number of steps. The terminal constraint set was chosen to be the origin, $\mathcal{X}_F = \{\mathbf{0}\}$. The stage cost was quadratic, with the weighting heaviliy biased to the control

$$\ell(\mathbf{x}, \mathbf{u}) = \frac{1}{1000}\mathbf{x}^T\mathbf{x} + 100u^2$$

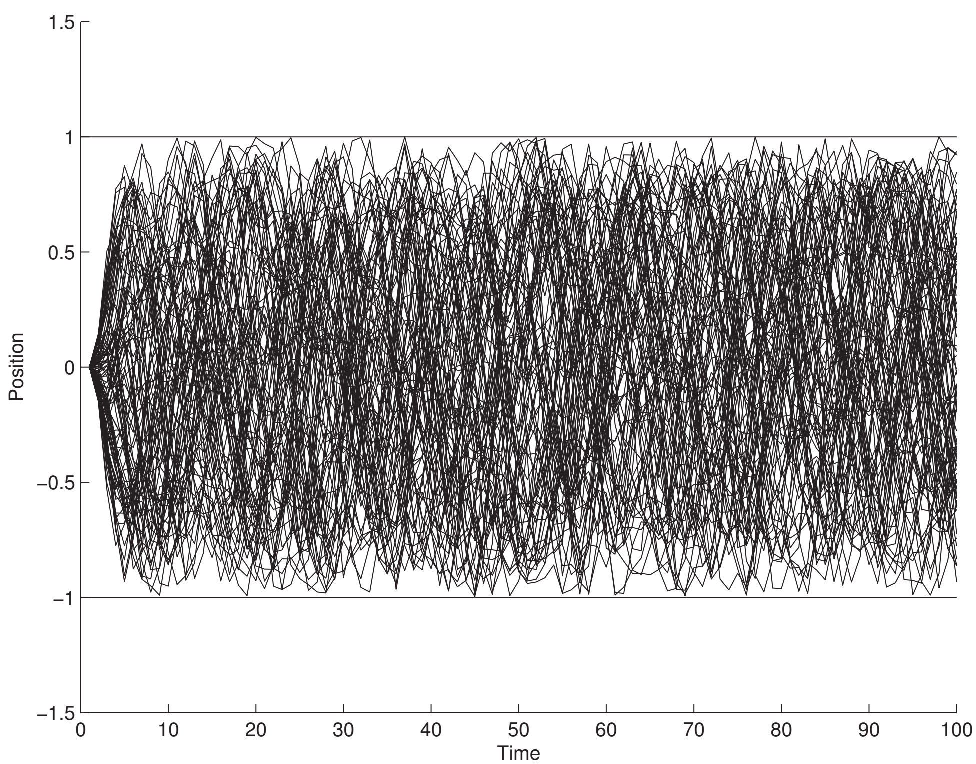
Fig. 2-2(a) shows the position time histories from 100 simulations, each with randomly-generated disturbances, using nominal MPC, *i.e.* without constraint tightening. The circles mark where a problem becomes infeasible. Of the 100 total simulations, 98 become infeasible before completion. Fig. 2-2(b) shows position time histories for a further 100 simulations using robustly-feasible MPC as in Section 2.3. Now all 100 simulations complete without infeasibility. Note that the mass goes all the way out to the position limits of ± 1, but never exceeds them. This is to be expected from a control-minimizing feedback law and shows that the controller is robust but still using all of the constraint space available, thus retaining the primary advantage of MPC.

2.5.2 Spacecraft Control

This section shows the application of robust MPC to precision spacecraft control. The scenario requires that a spacecraft remain within a 20m-sided cubic “error box” relative to a reference orbit using as little fuel as possible. In particular, this example is concerned with the choice of terminal constraint set for this scenario. Recall that the MPC optimization approximates the infinite horizon control problem by solving over a finite horizon and applying a terminal constraint to account for behavior beyond



(a) Nominal MPC



(b) Robust MPC

Figure 2-2: State Time Histories from 100 Simulations comparing Nominal MPC. 'o' denotes point at which problem becomes infeasible.