43rd IEEE Conference on Decision and Control
December 14-17, 2004
Atlantis, Paradise Island, Bahamas

Robust Model Predictive Control through Adjustable Variables: an application to Path Planning

Alessandro Abate* and Laurent El Ghaoui*

***Abstract*— Robustness in *Model Predictive Control* (MPC) is the main focus of this work. After a definition of the conceptual framework and of the problem's setting, we will analyze how a technique developed for studying robustness in *Convex Optimization* can be applied to address the problem of robustness in the MPC case. Therefore, exploiting this relationship between Control and Optimization, we will tackle robustness issues for the first setting through methods developed in the second framework. Proofs for our results are included. As an application of this Robust MPC result, we shall consider a Path Planning problem and discuss some simulations thereabout.**

I. INTRODUCTION

Model Predictive Control is a methodology intended to devise, given an explicit model of a system, sequences of control inputs which could be dynamically updated as soon as new observations of the output may be available throughout time. These controls are obtained as the result of a constrained optimization problem over a certain time span, where the index to be minimized is a function of the outputs and the inputs of the system. In general the time horizon can be variable, but in this work we shall consider it fixed; the constraints act on the states and on the inputs of the system.

The problem of Robustness for MPC, both referred to model uncertainty and noise, is a fundamental question. As described in [1], some results have been attained in the linear, time-invariant (LTI) case. All of them have to deal with computational constraints, which often make the problem intractable or infinite.

Another way to look at the issue is the following: if tackling the problem from a control standpoint often clashes with the computational requirements needed by the inner optimization problem, why not addressing it through optimization techniques? This is the main idea of another approach for enforcing robustness, that of Minimax MPC [2]. Similar to this last one is the approach that we employ in this paper.

That of *Adjustable Robust Solutions* is a theory developed by Nemirovski et al, see [3], [4] . Simply stated, starting from a classical optimization problem with some deterministic uncertainty, the idea is to extend the number of decision variables if some of them can be adjusted to the uncertainty parameters of the problem, i.e. if they depend on them (it can be thought as a sort of feedback); the new variables will make this relation between the optimization domain and the uncertainty explicit. This brings first of all more flexibility to the feasibility set of the problem (which tends to be quite conservative in the robust counterpart case), and in some cases may as well result in a computationally tractable problem, or at least in a tractable approximation of it. This is obtained through the known semidefinite relaxation technique [5].

The outline of the paper is as follows: after a formal description of the MPC problem, we shall introduce the idea of *Adjustable Robust Variables*, and show how they can be used in an optimization procedure. Then, we will extend and apply these concepts to the MPC setting, and demonstrate how robustness can be achieved. Furthermore, we will describe a testbed that we developed for simulations, and talk over some results we attained. Extensions, possible applications, remarks and future work conclude the paper.

*Department of Electrical Engineering and Computer Sciences, University of California at Berkeley; Berkeley, CA 94720.
Email:{aabate,elghaoui}@eecs.berkeley.edu; Tel: +1(510)643-4867, Fax: +1(510)642-1341.

II. SETTING

Let the linear model of a system be described by

$$\sum : \begin{cases} x_{t+1} = A_t x_t + B_t u_t \\ y_{t+1} = C_{t+1} x_{t+1}, \quad x_t = x_0. \end{cases} \tag{1}$$

Here $x_k \in \mathbb{R}^n$, $u_k \in \mathbb{R}^m$, $y_k \in \mathbb{R}^p$ denote the state, control and output signals. We decide to deal with the general case of time-varying matrices. We make the hypothesis that the state is observable, and, for the moment, that the equations have no additive noise or uncertain terms.

Assuming to have a reference trajectory to track from time t over a span of N steps, $[y^d_{t+1}, y^d_{t+2}, \ldots, y^d_{t+N}]$, the MPC problem can be formulated at time t as the solution of the following open-loop optimization problem:

$$\begin{aligned} & min_{y_{t+n}, u_{t+n-1}, n=1,\ldots,N} \quad J(y_{t+1}, u_t, N) \\ & = \sum_{n=1}^{N} \{ (y_{t+n} - y^d_{t+n})^T Q_n (y_{t+n} - y^d_{t+n}) \\ & + u^T_{t+n-1} R_n u_{t+n-1} \}, \end{aligned} \tag{2}$$

subject to (these are general constraints on the state and the input):

$$\begin{aligned} & F_1 u_{t+n-1} \le g_1, \\ & E_2 y_{t+n} + F_2 u_{t+n-1} \le g_2, \quad \forall n = 1, \ldots, N. \end{aligned} \tag{3}$$

As previously stated, we shall consider that N is fixed. Moreover, the matrices Q_n and R_n in the weighted norms are assumed to be positive definite, for all n.

The optimization problem gives an optimal solution for the control u throughout the whole time span; nevertheless, just the first values of it will be applied, say $u_t, u_{t+1}, .., u_{t+m}$; then the observation y_{t+m+1} shall be collected and the optimum in the new time frame $[t+m, t+m+N]$ recomputed, and so on. Choosing suitably the objective function, as well as the constraints, it can be demonstrated that stability is attained.

It should be already quite clear how the *computational tractability* issue is fundamental for the actual application of the algorithm. In a robust framework, where the numerical burden is intrinsically more demanding, this need becomes even more stringent.

III. Robustness in MPC

There are two ways to introduce uncertainty in a model: stochastically, if we assume that the uncertainty can be described in probabilistic terms; deterministically, if on the contrary we can just define some set inside which the uncertain term is supposed to lie: we are in other words bounding this term. The two approaches imply completely different ways to solve the problem. In this paper we shall embrace the second view.

Furthermore, as said before, the uncertainty can affect the model in two ways: as an exogenous disturbance, or as a possible range for the parameters of the system. In general, the reader will see how the optimization method we are going to exploit can deal with both kinds of randomness. Nevertheless, as an application to the MPC setting, we will only develop the theory tailored to the first of the two kinds.

A. Introducing some uncertainty

Assume our model encompasses also an additional exogenous disturbance ξ in the state equation:

$$\sum_{\xi} : \begin{cases} x_{t+1} = A_t x_t + B_t u_t + E_t \xi_t \\ y_t = C_t x_t, \quad x_t = x_0; \end{cases} \tag{4}$$

The new signal $\xi_k \in \mathbb{R}^q$. We shall assume that the variable ξ_t is bounded within an ellipsoid Ξ_t,

$$\Xi_t = \{\xi_t | \xi_t^T S_t \xi_t \leq \rho^2, \ \rho > 0, S_t \succ 0\}.$$

Developing the state equation N times ahead, starting at time t and clustering the terms into properly sized vectors, we get to:

$$\begin{aligned} \underline{x} &= \underline{c} + \Delta \underline{u} + \Omega \underline{\xi}, \\ \underline{y} &= C \underline{x}; \end{aligned} \tag{5}$$

where:

$$\begin{aligned}
\underline{x} &= [x_{t+1}, \ldots, x_{t+N}]^T; \\
\underline{y} &= [y_{t+1}, \ldots, y_{t+N}]^T; \\
\underline{u} &= [u_t, u_{t+1}, \ldots, u_{t+N-1}]^T; \\
\underline{\xi} &= [\xi_t, \xi_{t+1}, \ldots, \xi_{t+N-1}]^T; \\
\underline{c} &= \begin{bmatrix} A_t \\ A_{t+1}A_t \\ A_{t+2}A_{t+1}A_t \\ \vdots \\ A_{t+N-1}A... \end{bmatrix} x_t; \\
C &= \begin{bmatrix} C_t & 0 & ... & 0 \\ 0 & C_{t+1} & 0 & ... \\ 0 & 0 & C_{t+2} & ... \\ \vdots & ... & ... & \vdots \\ 0 & ... & 0 & C_{t+N-1} \end{bmatrix}; \\
\Delta &= \begin{bmatrix} B_t & 0 & ... & ... & 0 \\ A_{t+1}B_t & B_{t+1} & 0 & ... & 0 \\ A_{t+2}A_{t+1}B_t & A_{t+2}B_{t+1} & B_{t+2} & 0 & ... \\ \vdots & ... & ... & ... & \vdots \\ A_{t+N-1}A... & ... & ... & ... & B_{t+N-1} \end{bmatrix}; \\
\Omega &= \begin{bmatrix} E_t & 0 & ... & ... & 0 \\ A_{t+1}E_t & E_{t+1} & 0 & ... & 0 \\ A_{t+2}A_{t+1}E_t & A_{t+2}E_{t+1} & E_{t+2} & 0 & ... \\ \vdots & ... & ... & ... & \vdots \\ A_{t+N-1}A... & ... & ... & ... & E_{t+N-1} \end{bmatrix}.
\end{aligned}$$

The objective function then can be expressed as:

$$J(y_{k+1}, u_k, N) = (\underline{y} - \underline{y}^d)^T \mathbf{Q} (\underline{y} - \underline{y}^d) + \underline{u}^T \mathbf{R} \underline{u}, \tag{6}$$

where we clustered the aforementioned matrices Q_n and R_n into the new **Q** and **R**.

B. Adjustable Robust Counterpart

Nemirovski and coworkers [3], have proposed the following technique for dealing with uncertain linear programs. Consider the problem:

$$\{min_x \{c^T x : Ax \leq b\}\}_{\zeta \equiv [A,b,c] \in \mathcal{Z}},$$

where $\mathcal{Z}$ is a given nonempty compact set inside which the parameters of the problem lie; as a general method, we can associate to it its *Robust Counterpart*,

$$min_x \{sup_{\zeta \equiv [A,b,c] \in \mathcal{Z}} c^T x : Ax - b \leq 0, \forall \zeta \equiv [A, b, c] \in \mathcal{Z}\}.$$

Here all possible realizations of ζ have to be satisfied; therefore, it is possibly an infinite optimization problem and as such computationally intractable. In the *Robust Counterpart (RC)* approach, all the variables represent decisions that must be made before the realization of the unknown parameters: they are *"here and now decisions"*.

In some cases, though, part of the variables might tune themselves to the varying data, or at least to a part of them: they're *"wait and see decisions"*. In general, every variable

x_i might have its own "information basis", i.e. depend on a prescribed portion ζ_i of the past realized data ζ.

Let us now split the vector into two parts: a "non-adjustable" one and an "adjustable" one, $x = (u, v)^T$; furthermore, assume that the objective is normalized w.r.t. the non-adjustable variables[1]:

$$\{min_{u,v}\{c^T u : Uu + Vv \leq b\}\}_{\zeta \equiv [U,V,b] \in \mathcal{Z}},$$

We can then define the *(RC)* and the so called *Adjustable Robust Counterpart, (ARC)*:

$$min_u\big\{c^T u : \exists v \forall (\zeta \equiv [U,V,b] \in \mathcal{Z}) : Uu + Vv \leq b\big\};$$

$$min_u\big\{c^T u : \forall (\zeta \equiv [U,V,b] \in \mathcal{Z}) \exists v : Uu + Vv \leq b\big\}.$$

It is straightforward to realize that the *ARC* is *more flexible* than *RC*, i.e. it has a larger robust feasible set, enabling better optimal values while still satisfying all possible realizations of the constraints.

Unfortunately, it turns out that in some cases the *ARC* can be computationally harder than the *RC*. This spurs the introduction of a simplification on how the adjustable variables can tune themselves upon the uncertain data: we shall then consider *affine dependence* between the adjustable variables and the uncertain parameters. Posing $v = w + W\zeta$, we obtain the *Affinely Adjustable Robust Counterpart (AARC)* of the uncertain LP:

$$\begin{aligned} min_{u,w,W}\big\{c^T u : &\quad Uu + V(w + W\zeta) \leq b, \forall (\zeta \equiv [U,V,b] \in \mathcal{Z})\big\} \\ \equiv min_u\big\{c^T u : &\quad \forall (\zeta \equiv [U,V,b] \in \mathcal{Z}) \exists (v, W) : \\ &\quad Uu + V(w + W\zeta) \leq b, \big\}. \end{aligned} \tag{7}$$

C. AARC for MPC

Having introduced the robust MPC setting with an exogenous disturbance, we shall adapt the aforementioned theory to this case considering an affine dependence of the input on the past realized data: these, being realized, are not uncertain. We will assume a "full" information basis, that is we shall exploit all the past available data.

Consider the following relations:

$$\begin{aligned} u_t &= u_t^t; \\ u_{t+1} &= u_{t+1}^{t+1} + \xi_t u_{t+1}^t; \\ u_{t+2} &= u_{t+2}^{t+2} + \xi_{t+1} u_{t+2}^{t+1} + \xi_t u_{t+2}^t; \\ &\vdots \end{aligned} \tag{8}$$

We can compact them as:

$$\underline{u} = \underline{v} + W\underline{\xi}, \tag{9}$$

[1]This does not affect the generality of the problem, as we can always restate it in epigraphic form.

This way the new optimization variables are contained in the vector $\underline{v}$ and in the matrix W. The output equation becomes:

$$\underline{y} = C\underline{c} + C\Delta(\underline{v} + W\underline{\xi}) + C\Omega\underline{\xi}. \tag{10}$$

Plugging this term into the objective function at time t, we end up with:

$$\begin{gathered} J(y_{t+1}, u_t, N) = \\ (C\underline{c} + C\Delta(\underline{v} + W\underline{\xi}) + C\Omega\underline{\xi} - \underline{y}^d)^T \mathbf{Q} (C\underline{c} + C\Delta(\underline{v} + W\underline{\xi}) + C\Omega\underline{\xi} - \underline{y}^d) \\ +(\underline{v} + W\underline{\xi})^T \mathbf{R} (\underline{v} + W\underline{\xi}). \end{gathered} \tag{11}$$

Define the following three quantities, depending on the degree of their dependence on the uncertainties' vector:

$$\begin{aligned} \alpha &= \underline{c}^T C^T \mathbf{Q} C\underline{c} + \underline{c}^T C^T \mathbf{Q} C\Delta\underline{v} - \underline{c}^T C^T \mathbf{Q}\underline{y}^d + (\underline{y}^d)^T \mathbf{Q}\underline{y}^d - \\ &- (\underline{y}^d)^T \mathbf{Q} C\underline{c} - (\underline{y}^d)^T \mathbf{Q} C\Delta\underline{v} + \underline{v}^T \Delta^T C^T \mathbf{Q} C\Delta\underline{v} + \\ &+ \underline{v}^T \Delta^T C^T \mathbf{Q} C\underline{c} - \underline{v}^T \Delta^T C^T \mathbf{Q}\underline{y}^d + \underline{v}^T \mathbf{R}\underline{v}; \\ \beta &= \underline{W^T \mathbf{R}\underline{v}} + \underline{W}^T \Delta^T C^T \mathbf{Q} C\underline{c} + \underline{W}^T \Delta^T C^T \mathbf{Q} C\Delta\underline{v} + \\ &+ \Omega^T C^T \mathbf{Q} C\underline{c} + \Omega^T C^T \mathbf{Q}\Delta C\underline{v} - W^T \Delta^T C^T \mathbf{Q}\underline{y}^d - \Omega^T C^T \mathbf{Q}\underline{y}^d; \\ \gamma &= W^T \Delta^T C^T \mathbf{Q} C\Delta W + W^T \Delta^T C^T \mathbf{Q} C\Omega + \Omega^T C^T \mathbf{Q} C\Delta W + \\ &+ \Omega^T C^T \mathbf{Q} C\Omega + W^T \mathbf{R} W. \end{aligned} \tag{12}$$

Recasting the MPC problem at time t, with a time horizon of N steps, exploiting the quantities above and the *epigraphic* form, we obtain the following:

$$\begin{aligned} min &\quad t, \\ s.t. &\quad t - J(y_{t+1}, u_t, N) \geq 0, \quad \forall \xi \in \Xi_t, \end{aligned}$$

that is

$$\begin{aligned} min &\quad t, \\ s.t. &\quad t - \alpha - \beta^T \underline{\xi} - \underline{\xi}^T \beta - \underline{\xi}^T \gamma \underline{\xi} \geq 0, \quad \forall \xi \in \Xi_t. \end{aligned} \tag{13}$$

For the moment, we disregard the input and state constraints. As explained in [3], we can relax the constraint in the optimization problem before obtaining a semidefinite program; the following lemma and theorem show how to do this:

Lemma 1: : For every $\underline{v}$, W, the implication

$$\begin{gathered} \forall s, \zeta : s^2 \leq 1 \quad , \quad \zeta^T(\rho^{-2} S_l)\zeta \leq 1, \; l = 1, \ldots, L \\ \Downarrow \\ \alpha \quad + \quad 2\zeta^T \beta s \quad + \quad \zeta^T \gamma \zeta \quad \geq \quad 0 \end{gathered}$$

is valid iff there's feasibility for the constraint:

$$\forall \zeta \in \Xi, \quad \alpha + 2\zeta^T \beta + \zeta^T \gamma \zeta \geq 0.$$

Manipulating this lemma (refer to [3] for further details), we get to the following result:

Theorem 1: : The explicit semidefinite program

$$\min_{\underline{\lambda},\underline{v},W,t} \quad t,$$
$$s.t. \quad \begin{pmatrix} \gamma + \rho^{-2}\sum_{l=1}^{L}\lambda_l S_l & \beta \\ \beta^T & \alpha - \sum_{l=1}^{L}\lambda_l \end{pmatrix} \succeq 0$$
$$\underline{\lambda} \geq 0. \quad (14)$$

is an approximation to the Affinely Adjustable Robust Counterpart (in the case L=1, there's equivalence between the two).

For specifications about the quality of the approximation, please refer to the cited literature. As said, the advantage attained through this new expression is twofold: the new formulation of the optimization problem has first a more flexible feasibility set (as already shown before), then it can be codified into a computationally tractable program.

Exploiting these two results, and after proper manipulations, we can recast the optimization problem as an LMI.

D. Main Result

Theorem 2: : The explicit semidefinite program

$$\begin{bmatrix} \mathbf{P} & \begin{pmatrix} W^T & I & \underline{0} \\ \underline{v}^T & \underline{0}^T & 1 \end{pmatrix} \\ \begin{pmatrix} W & \underline{v} \\ I & \underline{0} \\ \underline{0}^T & 1 \end{pmatrix} & \mathbf{M}^{-1} \end{bmatrix} \succ 0, \qquad (15)$$

where the matrices **P** and **M** will be defined in the proof, is an approximation to the *Affinely Adjustable Robust Counterpart* (in the case L=1, there's even equivalence between the two) and describe one of the constraints of the optimization problem (13).

Proof: To begin with, assume the uncertainty set is actually an intersection of spheres, rather than ellipsoids; in other words, consider $S_l = I_{N-1\times N-1}$. This assumption doesn't affect the generality of the results. As sated before, we shall use all the past realized data (we have "full" information basis), therefore for this particular instance $L = N - 1$.

Starting from equation (13), exploiting the trick in Lemma (1) and recasting the problem in semidefinite form, as Theorem (1) shows, we obtain that the optimization problem can be expressed as:

$$\min_{\underline{\lambda},\underline{v},W,t} \quad t,$$
$$s.t. \begin{pmatrix} t - \gamma + \rho^{-2}\sum_{l=1}^{L}\lambda_l S_l & -\beta \\ -\beta^T & -\alpha - \sum_{l=1}^{L}\lambda_l \end{pmatrix} \succeq 0$$
$$\underline{\lambda} \geq 0. \quad (16)$$

Let us focus on the constraint of the problem in the LMI form; it can be rewritten explicitly in terms of the decision variables as:

$$\begin{pmatrix} t + \rho^{-2}\sum_{l=1}^{L}\lambda_l S_l & 0 \\ 0 & -\sum_{l=1}^{L}\lambda_l \end{pmatrix} -$$
$$\begin{bmatrix} W^T & I & \underline{0} \\ \underline{v}^T & \underline{0}^T & 1 \end{bmatrix}$$
$$\begin{bmatrix} \mathbf{R} + \Delta^T C^T \mathbf{Q} C\Delta & \Delta^T C^T \mathbf{Q} C\Omega & \Delta^T C^T \mathbf{Q}(C\underline{c} - \underline{y}^d) \\ \Omega^T C^T \mathbf{Q} C\Delta & \Omega^T C^T \mathbf{Q} C\Omega & \Omega^T C^T \mathbf{Q}(C\underline{c} - \underline{y}^d) \\ (C\underline{c} - \underline{y}^d)^T \mathbf{Q} C\Delta & (C\underline{c} - \underline{y}^d)^T \mathbf{Q} C\Omega & (C\underline{c} - \underline{y}^d)^T \mathbf{Q}(C\underline{c} - \underline{y}^d) \end{bmatrix}$$
$$\begin{bmatrix} W & \underline{v} \\ I & \underline{0} \\ \underline{0}^T & 1 \end{bmatrix} \succeq 0.$$

Unfortunately, this whole term is non-linear, which makes the problem difficult to handle computationally. Nevertheless, the multiplicative matrix in the second term can be expressed as:

$$\begin{bmatrix} \Delta^T C^T \mathbf{Q} C\Delta & \Delta^T C^T \mathbf{Q} C\Omega & \Delta^T C^T \mathbf{Q}(C\underline{c} - \underline{y}^d) \\ \Omega^T C^T \mathbf{Q} C\Delta & \Omega^T C^T \mathbf{Q} C\Omega & \Omega^T C^T \mathbf{Q}(C\underline{c} - \underline{y}^d) \\ (C\underline{c} - \underline{y}^d)^T \mathbf{Q} C\Delta & (C\underline{c} - \underline{y}^d)^T \mathbf{Q} C\Omega & (C\underline{c} - \underline{y}^d)^T \mathbf{Q}(C\underline{c} - \underline{y}^d) \end{bmatrix} +$$
$$+ \begin{bmatrix} \mathbf{R} & \underline{0} & \underline{0} \\ \underline{0} & \underline{0} & \underline{0} \\ \underline{0} & \underline{0} & 0 \end{bmatrix} = \begin{bmatrix} \Delta^T C^T \\ \Omega^T C^T \\ (C\underline{c} - \underline{y}^d)^T \end{bmatrix} \mathbf{Q} \begin{bmatrix} C\Delta & C\Omega & (C\underline{c} - \underline{y}^d) \end{bmatrix}$$
$$+ \begin{bmatrix} \mathbf{R} & \underline{0} & \underline{0} \\ \underline{0} & \underline{0} & \underline{0} \\ \underline{0} & \underline{0} & 0 \end{bmatrix}.$$

This fact, along with the observation that all the matrices are full rank, and that Q is by assumption positive definite, enables us to infer that the first term is positive definite; therefore, the multiplicative matrix as a whole is also positive definite, being the sum of a positive definite matrix, and of a non-negative definite one (the second additive matrix). Hereinafter, we shall refer to this multiplicative matrix as **M**.

It is then possible to apply the *Schur Complement*, (refer to [6] for more details) to the whole expression; after naming $\begin{pmatrix} t + \rho^{-2}\sum_{l=1}^{L}\lambda_l S_l & 0 \\ 0 & -\sum_{l=1}^{L}\lambda_l \end{pmatrix} = \mathbf{P}$, we obtain:

$$\begin{bmatrix} \mathbf{P} & \begin{pmatrix} W^T & I & \underline{0} \\ \underline{v}^T & \underline{0}^T & 1 \end{pmatrix} \\ \begin{pmatrix} W & \underline{v} \\ I & \underline{0} \\ \underline{0}^T & 1 \end{pmatrix} & \mathbf{M}^{-1} \end{bmatrix} \succ 0.$$

■

This last expression is an LMI for a constraint in our optimization problem; therefore, it can be enforced in a computationally tractable way through simulation toolboxes like SeDuMi. We have demonstrated that the feasibility set of the possibly infinite robust optimization problem can be reduced, under the hypothesis of having an ellipsoidal uncertainty set, to a finite-dimensional semidefinite program involving LMI's. Therefore, the robust optimization is computationally tractable.

E. Introduction of the Constraints

At this point, we want to enforce some constraints of the form (3). For easiness we shall consider only the first constraint:

$$F_1 u_{t+n-1} \leq g_1, \quad n = 1, \dots, N.$$

Vectorizing the inputs and clustering the matrices, we obtain:

$$\mathbf{F}_1 \underline{u} \leq \mathbf{g}_1.$$

After substituting the inputs:

$$\mathbf{F}_1(\underline{v} + W\underline{\xi}) \leq \mathbf{g}_1.$$

In the optimization problem, we shall refer to the $(N-1)$ projections of this vector of inequalities. We will show that these constraints can be expressed in the same way as the former one. Define:

$$\begin{aligned} \alpha_i^u &= \underline{g}_{1_i} - (\mathbf{F}_1 \underline{v})_i; \\ \beta_i^u &= -(\mathbf{F}_1 W)_i; \\ \gamma_i^u &= 0; \quad i = 1, \dots, N-1. \end{aligned}$$

Then, the former inequality can be expressed as:

$$\begin{pmatrix} \gamma^u + \rho^{-2} \sum_{l=1}^{L} \lambda_{l,i} S_{l,i} & \beta^u \\ \beta^{u^T} & \alpha^u - \sum_{l=1}^{L} \lambda_{l,i} \end{pmatrix} \succeq 0$$
$$\underline{\lambda}_i \geq 0, \quad i = 1, \dots, N-1. \quad (17)$$

All these $(N-1)$ constraints can be incorporated into the big optimization program exploiting the fact that, given two matrices A and B, $A \succeq 0$ and $B \succeq 0 \Leftrightarrow \begin{pmatrix} A & 0 \\ 0 & B \end{pmatrix} \succeq 0$.

IV. Simulations

We have implemented the methodology through increasingly complex simulations in MATLAB. The special toolbox that we made use of is SeDuMi [7]. As shown, the method is in general valid for any linear, time-varying model; we restricted the simulation to an LTI setting though. As usual in MPC, the weighting matrices $\mathbf{Q}$ and $\mathbf{R}$ have been selected empirically, depending on the magnitude of the two terms in the objective function, as well as on the relative weight and bias we desired to wield to our control action. Figure (1) shows a three dimensional example, where we wish to track a trajectory that ends up in the origin. In this particular instance, we have kept the uncertainty rather small; in fact, it has a bound on its magnitude of 10^{-4}. The time horizon has been set to $N = 25$. Depending on the magnitude of the eigenvalues of the system and output matrices, we can possibly increase the time horizon and be able to achieve a satisfactory converging behavior.

It is interesting to check how "robust" our control action can be. To do so, we have benchmarked, with different uncertainty levels (magnitude of the exogenous input, compared to that of the control input), the optimal control obtained through the AARC approach, with the "ideal" optimal control computed with the knowledge of the disturbances. The comparison has been made evaluating the average of the outcomes of n experiments over the value of the objective function. Table (1) shows the results using 50 realizations for the uncertainties. It can be observed that a dimensional increase in the magnitude of ρ (compared in percentage with the magnitude of the input signals) does not correspond to a comparable increase in the error ("price") between the results attained with the ideal control and those referring to the AARC one, even though the objective function depends quadratically on the disturbances. As a matter of fact, figure (2) shows that the results with a larger ρ, though less precise, are not particularly deteriorated.

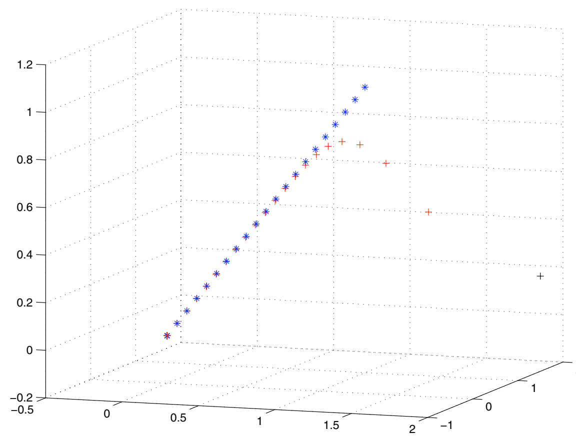

Fig. 1. $N = 25$; $\|\rho\| \leq 10^{-4}$; $y_0 = [2; 1.5; 0.2]$.

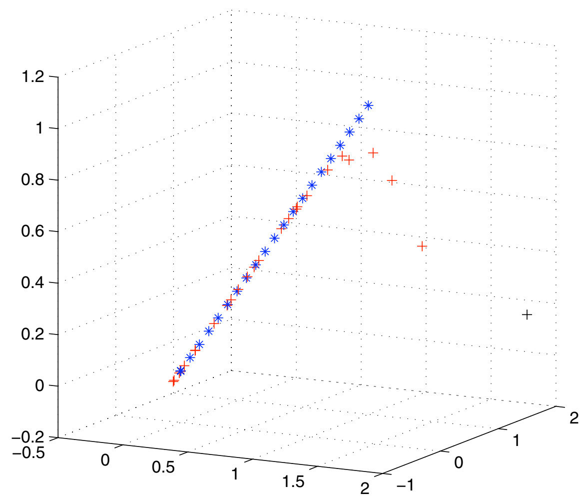

Fig. 2. $N = 25$; $\|\rho\| \leq 10^{-2}$; $y_0 = [2; 1.5; 0.2]$.

V. Conclusions and Future Work

This paper has been proposing a way to investigate robustness in an MPC framework, exploiting a relaxation technique employed in convex optimization. After a rigorous derivation of the results, simulations have shown the viability of the method. The authors are currently investigating the inclusion of convex state constraints in the framework. There are already some results in literature that

ρ	J_{av} w/ Ideal Control	J_{av} w/ AARC Control	Price (%)
$10^{-6}(10^{-3}\%)$	7.2071	7.2071	$10^{-8}\%$
$0.0001(0.1\%)$	7.2071	7.2071	$10^{-3}\%$
$0.001(1\%)$	7.2071	7.2073	0.01%
$0.01(10\%)$	7.2083	7.2244	0.5%
$0.05(20\%)$	7.2098	7.7330	6.5%

Table 1: Ideal optimal control, Optimal AARC control vs. uncertainty level.

could be a significative starting point, see [8]. Furthermore, a testbed has been developed in order to apply this results to the current Berkeley Aerial Robot (BEAR) research project.

VI. Acknowledgments

The work has been partially supported by the ARO-MURI ACCLIMATE, DAAD-19-02-1-0383 grant, and the NSF CCR-0225610 grant. The first author is recipient of the "Fondazione Aldo Gini" fellowship.

References

[1] A. Bemporad and M. Morari, "Robust model predictive control: A survey," *Hybrid Systems: Computation and Control, F.W. Vaandrager and J.H. van Schuppen*, vol. 1569, pp. 31–45, 1999, lecture Notes in Computer Science.

[2] J. Loefberg, "Minimax approaches to robust model predictive control," Ph.D. dissertation, Univ. of Linkoping, 2003.

[3] A. Ben-Tal, A. Goryashko, E. Guslitzer, and A. Nemirovski, "Adjustable solutions of uncertain linear programs."

[4] A. Ben-Tal and A. Nemirovski, "Robust convex optimization," vol. 23, 1998.

[5] L. Vandenberghe and S. Boyd, "Semidefinite programming," vol. 38(1), march 1996.

[6] S. Boyd, L. E. Ghaoui, E. Feron, and V. Balakrishnan, *Linear Matrix Inequalities in Systems and Control Theory*. SIAM, 1994.

[7] J. F. Sturm, "Using sedumi 1.0x, a matlab toolbox for optimization over symmetric cones," 1998. [Online]. Available: http://fewcal.kub.nl/sturm/software/sedumi.html

[8] T. Schouwenaars, B. DeMoor, E. Feron, and J. How, "Mixed integer linear programming for multi-vehicle path planning," in *Proc. ECC'01*, 2001.