Stochastic disturbance rejection in model predictive control by randomized algorithms

Ivo Batina*  Anton A. Stoorvogel*  †  Siep Weiland‡

Abstract

In this paper we consider model predictive control with stochastic disturbances and input constraints. We present an algorithm which can solve this problem approximately but with arbitrary high accuracy. The optimization at each time step is a closed loop optimization and therefore takes into account the effect of disturbances over the horizon in the optimization. Via an example it is shown that this gives a clear improvement of performance although at the expense of a large computational effort.

1 Introduction

Model Predictive Control (MPC) has been widely accepted in industry as an effective multivariable control design technique. Existing model predictive controllers are based on finite time horizon optimization. At each time instant, a finite horizon optimal control problem is solved, taking the current state of the plant as an initial condition and using a model for the prediction of the controlled plant behavior. Only the first control is applied to the plant and at the next time instant the optimization procedure is repeated based on the new measurements.

When disturbances are acting on the plant that one aims to control, it is desirable to include it in the control problem formulation. In MPC setting, there are three basic approaches for dealing with disturbances that have been suggested in the literature.

The first approach is to assume that the disturbance is known and either zero or constant over the optimization interval. This is known as the classical setting for which there exists a vast literature (see [4]) based on convex on-line optimization. First attempts have been made to obtain a closed-loop off-line solution (see [1]). In many real-world applications this approach is too "optimistic" since it ignores the effect that the disturbance may have on the performance of the controlled system.

The second approach, assumes the unknown disturbance to belong to a class of signals and optimization is based on a min-max approach where the minimization is performed over a set of input sequences and maximization over a set of disturbance sequences (see [3]). To be feasible, this approach requires the disturbance to be bounded. Since the min-max optimization looks for the worst possible disturbance realization this approach is generally too "pessimistic".

The third approach is a stochastic approach that generally yields a much more realistic view of the behavior of the system. The stochastic view on the disturbance in MPC could be traced back to Clarke's Generalized Predictive Control. Like in the many references that follow the same line of thought, the results are valid only when there are no constraints on the input and states. A modification of the open loop convex optimization is proposed in [2] for the case of an input constraint and a stochastic disturbance. The resulting control law, however, does not control the spread of the states influenced by an unknown disturbance and therefore often predicts constraint violations in cases where in reality constraint violation would not occur.

The stochastic approach is the approach adopted in this paper. The main difficulty with a stochastic disturbance in MPC is that when constraints on the input and states are present the minimization of the expected value of the cost function over the horizon yields a very difficult optimization problem. This is one of the reasons why MPC with a stochastic disturbance is still an open research problem. In this work we present an algorithm which can solve this problem approximately but with arbitrarily high accuracy. The optimization at each time step in the horizon is a closed loop optimization and expected value of the cost-to-go is computed approximately by a randomized algorithm. Although accurate, the algorithm has a drawback of a high computational complexity. The ambition of this paper is not to replace the existing MPC algorithms but to provide an useful tool for qualifying achievable performance of an MPC scheme and to evaluate the trade-off between computational complexity and the performance.

The paper is organized as follows. The problem definition is given in section 2. The background material on randomized algorithms is presented in section 3. The algorithm for solving the problem and a convergence proof of the result obtained by the algorithm are given in section 4, together
2 Problem formulation

In this paper, we consider the following linear time-invariant discrete-time system:

\[
\begin{align*}
x(t + 1) &= Ax(t) + Bu(t) + Ew(t) \\
z(t) &= Cz(t) + Dzu(t)
\end{align*}
\]  

(1)

The first equation describes a plant with the state \( x(t) \in \mathbb{R}^n \) and input \( u(t) \in U \subseteq \mathbb{R}^m \), where \( U \) is a compact, convex set which contains an open neighborhood of the origin. We assume that a disturbance \( w(t) \in W \subseteq \mathbb{R} \) is a white noise stochastic process taking values at each time \( t \) in the set \( W \) with some known probability distribution. The second equation describes the controlled output \( z(t) \in \mathbb{R}^p \). It is assumed that the state \( x(t) \) is measured.

The linear discrete-time system (1) is subject to amplitude constraints on the input. It is well known (see [5]) that such a system can be globally asymptotically stabilized via feedback, only in the case that all eigenvalues of the system matrix \( A \) lie on or inside the unit circle. This general result obviously applies to all types of model predictive controllers.

In this paper, we assume that the condition for global asymptotic stabilization is satisfied, i.e. all eigenvalues of \( A \) are located on or inside the unit circle.

In the following, we will outline the main ingredients of the MPC approach used in this paper. We consider the control horizon \( T := [t, t + N] \) with length \( N > 0 \) where \( t \in \mathbb{Z} \) is a fixed time instant (representing the current time). Let a control and a disturbance sequence on this horizon be denoted by \( u : T \rightarrow U \) and \( w : T \rightarrow W \), respectively and let \( z : T \rightarrow \mathbb{R}^p \) be the controlled output of (1) subject to the initial condition \( x(t) \) at time \( t \), control \( u \) and disturbance \( w \).

Consider the standard MPC cost function of the form:

\[
J(x(t), u, w) := \sum_{k \in F} \|z(k)\|^2 + \|x(t + N + 1)\|^2 \quad (2)
\]

where \( Q \in \mathbb{R}^{p \times p} \) is a positive definite, symmetric matrix. The end point penalty is described as \( \|x\|^2 := < x, Qx > \).

The end point penalty is commonly used as a method for achieving asymptotic stability of a MPC scheme. Note that even if the system (1) is stable, the closed loop system with a MPC control law based on the minimization of (2) might not be. In general, the more the state at the end point is penalized in (2), the more likely it is that a model predictive control law will yield a stable closed loop system. For further details about the choice of \( Q \) we refer to [6].

We consider a MPC feedback law based on the following optimization

\[
V(x(t)) := \min_u \{ E_w J(x(t), u) \mid u(t) \in U, t \in T \} \quad (3)
\]

where \( E(.) \) denotes the conditional expectation with respect to \( (\cdot) \). If \( u_t \) is the state feedback law that minimizes \( E_w J(x(t), u) \) or, equivalently, the stochastic process that minimizes the same cost function, then \( u_t(t) \) is uniquely determined by \( x(t) \) and by setting \( u(t) = u_t(t) \). By repeating such a procedure for all \( t \) we obtain a receding horizon implementation for our control law.

Note that we do not restrict the optimization to be in open-loop, i.e. we allow that at every time instant \( k \in T \) the input \( u(k) \) is a function of \( x(k) \). Because of that, the input \( u(k) \) is a stochastic variable and therefore we do not have a simple finite-dimensional QP optimization problem as in the case of open loop optimization. The standard method for solving optimization problems like (3) is based on dynamic programming. The optimization (3) is equivalent to the dynamic program:

\[
V_k := \min_{u \in U} \left\{ \|Cz(k) + Dzu(k)\|^2 + E_w(k)V_{k+1} \right\} \quad (4)
\]

with a terminal condition:

\[
V_{t+N+1} := \|x(t + N + 1)\|^2_Q
\]

that has to be solved backwards from \( k = t + N \) to \( k = t \) to obtain the optimal feedback relation between \( u(k) \) and \( x(k) \). Here \( V_k \) is viewed as a function of the state, but we drop the argument for brevity.

It is well known that without constraints on the input the optimization (3) has a closed-form solution in the form of a state feedback law. When constraints on the input are present, however, the optimal "cost to go" \( V_k \) in (4) does not have a quadratic structure in general and therefore an analytic expression for its conditional mean is not straightforward. An additional difficulty is that we need to find an optimal input as a function of the state for every \( k, k \in T \) and this class of functions is infinite dimensional.

In this paper, we propose a method by which we can find an approximate solution of this problem. The solution is based on estimation of the conditional mean in (4) by a randomized algorithm. Although computationally very intensive, the method proposed here computes the solution with arbitrary high accuracy.

3 Empirical mean

An analytical computation of the expectation in (3) for a given input \( u \) based on the distribution of \( w \) is difficult. An alternative is to compute the empirical mean of the cost (2). The cost for a specific realization of the stochastic disturbance \( w \) is easily computed. The realizations are chosen
randomly, according to the distribution of \( w \). That is a motive for referring to an algorithm in which the empirical mean is used as a randomized algorithm. Let us formally define the empirical mean and recall the important Hoeffding’s inequality.

Assume a set \( \Theta \) and a probability measure \( P \) on \( \Theta \) are given. Let \( f : \Theta \to \Omega, \Omega \) an interval on \( \mathbb{R} \) possibly equal to \( \mathbb{R} \) be a scalar-valued function measurable with respect to \( P \). The expectation of \( f \) can be expressed as:

\[
E f = \int_{\Theta} f(\theta) dP. \tag{5}
\]

Our aim is to approximate (5) by drawing \( m \) independent, identically distributed (i.i.d) samples \( \theta_1, \cdots, \theta_m \) from \( \Theta \) in accordance with \( P \) and compute the empirical mean:

\[
\hat{E} f : = \frac{1}{m} \sum_{j=1}^{m} f(\theta_j) \tag{6}
\]

The empirical mean (6) is a function of a randomly chosen multisample \( \theta \) and it is obviously stochastic. Such an estimate is useful only if we have an insight in the error given by \( |E f - \hat{E} f| \). Since (6) is stochastic, the error is expressed in a probabilistic confidence interval rather than in the form of a strict bound. We have confidence \( \delta \) in the approximation (6) if \( |E f - \hat{E} f| < \varepsilon \) with a probability of at least \( \delta \).

An upper bound for the confidence \( \delta \) can be derived by Hoeffding’s inequality [7] for the case \( \Omega \subset \mathbb{R} \). Hoeffding’s inequality is an inequality that applies to the sum of independent zero-mean random variables with bounded range. Suppose \( y_1, \cdots, y_m \) are independent random variables with \( E y_i = 0 \) for each \( y_i \), and that \( a_i \leq y_i \leq b_i \) for each \( i \). Hoeffding’s inequality then states:

\[
\text{Prob} \left\{ \sum_{i=1}^{m} y_i \geq \alpha \right\} \leq e^{-\frac{\alpha^2}{2m}} \tag{7}
\]

where \text{Prob} indicates probability.

Now, differences \( f(\theta_1) - E f, \cdots, f(\theta_m) - E f \), for some multisample \( \theta \), are all bounded range, zero mean random variables to which Hoeffding’s inequality can be applied. This leads to the following:

\[
\text{Prob} \left\{ \left| \sum_{i=1}^{m} f(\theta_i) - E f \right| < \varepsilon \right\} \geq 1 - 2e^{-\frac{2\varepsilon^2}{m}} \tag{8}
\]

where \( |\Omega| \) denotes the length of \( \Omega \). Thus, the confidence that one has in the empirical mean (6) is at most \( 1 - 2e^{-\frac{2\varepsilon^2}{m}} \).

Expression (7) implies that, given \( \varepsilon \), the desired confidence \( \delta \) is achieved if the number of samples \( m \) satisfies:

\[
m \geq \frac{|\Omega|^2}{2\varepsilon^2} \ln \frac{2}{1 - \delta} \tag{8}
\]

An important property of (8) is that the lower bound for the number of samples is independent of the dimension of the underlying stochastic process, i.e. the dimension of the set \( \Theta \).

From (7) follows:

\[
\text{Prob} \left\{ \left| \hat{E} f - E f \right| < \varepsilon \right\} \to 1 \text{ as } m \to \infty,
\]

for all \( \varepsilon > 0 \). We say that the empirical mean (6) converges in probability to the expectation \( E f \).

4 Algorithm

In the algorithm for approximate solution of the optimization problem (3) we use a minimization of an empirical mean instead of the true mean in (4). The empirical mean is calculated by generating a number of samples for the disturbance over the horizon.

Note that at time \( p = t + N \) we have:

\[
V_p = \min_{u \in U} \left\{ \|z(p)\|^2 + \|Ax(p) + Bu\|^2 \right\} \tag{9}
\]

so \( w(t+N) \) is immaterial for the optimization (3) and therefore set to 0.

The algorithm is based on a standard dynamic programming approach. To express an estimated “cost to go” we define the disturbance and control sequences restricted to the time interval \( T_s = [t + s, t + N] \) as \( w_s : [t + s, t + N - 1] \to \mathbb{W}, w(t+N) = 0 \) and \( u_s : [t + s, t + N] \to \mathbb{U} \). The system (1) is time invariant which makes the cost (2) and the optimization problem (3) independent of the current time \( t \). So, without loss of generality, we can assume that \( t = 0 \).

Suppose that we take \( \kappa \) samples of the disturbance \( w(0) \) at \( t = 0 \). With that, there are \( \kappa \) possible states \( x(1) \) for the initial condition \( x(0) \) and the input \( u(0) \). For each one of these possible futures we generate \( \kappa \) samples of the disturbance \( w(1) \) which establishes \( \kappa^2 \) possible future states \( x(2) \). Continuing in this way, by the persistent sampling of the disturbance up to time \( N - 1 \) the number of samples of \( w \) is \( \kappa^N \). The number of samples of the restricted disturbance sequence \( w_s \) is \( \kappa^{N-s} \). The number of samples of \( w \) grows exponentially with the horizon.

For all \( s \in [0, N] \) and for each of the \( \kappa^{N-s} \) samples \( w_{i,j}^{s} \) of \( w_s, i \in [1, \kappa^{N-s}] \) we write the cost function as (recall that \( t = 0 \)):

\[
J^s(x(s), u_s, w_{i,j}^{s}) = \sum_{k=s}^{N} \|z(k)\|^2 + \|x(N+1)\|^2 \tag{10}
\]

The empirical conditional mean of the cost function in \( x(s) \) given a restricted input sequence \( u_s \) is then calculated as:

\[
\hat{E}_{u_s} J(x(s), u_s) := \frac{1}{\kappa^{N-s}} \sum_{i=1}^{\kappa^{N-s}} J(x(s), u_s, w_{i,j}^{s}) \tag{10}
\]
Minimization of \((10)\) over a given set of restricted input sequences will yield a suboptimal input sequence. This optimization problem at time \(s\) is defined as:
\[
\hat{V}_s := \min_{u_s} \left\{ E_{w_s} J(x(s), u_s) \mid u_s(\tau) \in U, \tau \in T_s \right\}
\]  
(11)
We call \(\hat{V}_s(x(s))\) an empirical optimal "cost to go". Since we have to minimize \((10)\) for all possible states at \(s\) that are determined by all samples of \(w(\tau), \tau \in [0, s - 1]\) we need a sufficient number of samples for \(w(\tau), \tau \in [s, N]\). This yields exponential growth in the number of required samples as a function of the horizon. Although the idea just described seems simple, there are a few important issues that have to be addressed before its implementation in the form of an algorithm can be presented. As already mentioned, the optimal "cost to go" in the optimization problem \((4)\) does not have a quadratic structure and that obviously also applies to the empirical optimal "cost to go". Moreover, \(\hat{V}_s(x(s))\) is stochastic because it is parameterized by a randomly chosen sample of the disturbance \(w_s\). The empirical optimal "cost to go" needs, in principle, to be computed for all possible states \(x(s)\). Instead we look at all of our sampled past disturbances and predict \(x(s)\). That yields a grid of the state space at time \(x(s)\) but not a uniform grid but a grid biased towards those states which are "likely", given past disturbances. Another issue is convexity of \((10)\) which is a crucial property from the optimization point of view since it guarantees convergence of our algorithm.

An approximation of the optimization problem \((3)\) can now be written according to the notation above as:
\[
\hat{V}_0(x(0)) = \min_{u_0} \left\{ E_{w_0} J(x(0), u_0) \mid u_0(\tau) \in U, \tau \in [S, N] \right\}
\]  
(12)
The following lemma gives a key for the construction of an algorithm for optimization problem \((11)\) and the confidence that one can have in it:

**Lemma 1** The empirical optimal cost \((11)\) is a strictly convex function in \(x(s)\) for all \(s\).

**Proof:** To prove the statement first we have to show that the set of the control sequences \(u_s\) is a convex set. With the element-wise addition on \(u_s\), that follows trivially from the definition of convex sets since \(u_s(\tau) \in U, \tau \in T_s, U\) is a convex set by definition.

Secondly, the cost \(J^s\) is a strictly convex function of \(x(s)\) and \(u_s\) for each \(w_s\). Because the estimated conditional mean \((10)\) is defined via operations that preserve convexity it is also a strictly convex function in \(x(s)\) and \(u_s\).

Because \((10)\) is strictly convex in \(u_s\) and the set of the sequences \(u_s\) is a convex set, the optimization problem \((11)\) is a convex optimization problem and the minimizing sequence, denoted as \(\tilde{u}_s\) is unique.

Finally, to prove the statement of the lemma, consider \(x^a(s), x^b(s) \in \mathbb{R}^s, x^a(s) \neq x^b(s)\). The corresponding minimizing control sequences are denoted as \(\tilde{u}^a_s\) and \(\tilde{u}^b_s\), respectively.

Strict convexity of \((10)\) then implies:
\[
\hat{V}(\lambda x^a(s) + (1 - \lambda)x^b(s)) \leq \hat{V}(\lambda \tilde{u}^a_s(x^a(s)) + (1 - \lambda)\tilde{u}^b_s(x^b(s)) = \lambda \hat{V}(\tilde{u}^a_s(x^a(s))) + (1 - \lambda)\hat{V}(\tilde{u}^b_s(x^b(s)))
\]  
for all \(x^a(s), x^b(s) \in \mathbb{R}^s\) and \(\lambda \in (0, 1)\).

The result presented in Lemma 1 makes it possible to obtain an efficient algorithm using convex optimization, for example a bisection algorithm to minimize the empirical means in \((11)\).

The algorithm for solving \((3)\) approximately can now be derived following the dynamic programming principle \((4)\), with the empirical mean instead of the true mean stated as:
\[
\hat{V}_s := \min_{u_e} \left\{ \|C_s x(s) + D_s u\|^2 + E_{w(s)} \hat{V}_{s+1} \right\}
\]  
(13)
with the terminal condition:
\[
\hat{V}_{N+1} := \|x(N + 1)\|^2_Q
\]
that has to be solved backwards from \(s = N\) to \(s = 0\) i.e. from time instant \(t + N\) to \(t\). Before presenting the algorithm a digression about notation is necessary.

With the disturbance sampled as described and with some input sequence \(u_s\), at each time instant in the horizon \(s = 0, \ldots, N - 1\) there are \(k^s\) possible states denoted as \(x^i, i = 1, \ldots, k^s\). The estimated optimal cost in \(x^i\) is denoted as \(\hat{V}_s(x^i)\).

We can now present our algorithm:

**Algorithm 1**

**Step 1** Set \(\hat{u}(s) = 0\) for \(s = 0, 1, \ldots, N\) and draw \(\kappa^N\) samples of \(w\) as described before.

Set \(V_0 = \infty\). Set accuracy parameter \(\lambda\). Set \(s = N\).

**Step 2** Determine a new \(\hat{u}(s), s = N\) using \((9)\) for each \(x^i, i = 1, \ldots, k^N\). Compute \(\hat{V}_N(x^i)\) for each \(i\). Set \(s = N - 1\).

**Step 3** Determine a new \(\hat{u}(s)\) using \((13)\) for each \(x^i, i = 1, \ldots, k^s\). Compute \(\hat{V}_s(x^i)\) for each \(i\).

If \(s = 0\) go to step 4, otherwise set \(s = s - 1\) and go to step 3.

**Step 4** If \(|\hat{V}_0 - V_0| < \lambda\) stop. Otherwise set \(V_0 = \hat{V}_0\) and go to step 2.
The following theorem states the main result of the paper:

**Theorem 1** The input sequence \( \hat{u} \) and the cost function \( \hat{V}_0 \) obtained from the Algorithm 1 converge in probability to the optimal input sequence and the optimal cost \( V_0 \) for the optimization problem (4) if \( \lambda \rightarrow 0 \) and \( \kappa \rightarrow \infty \).

**Proof:** As \( \lambda \rightarrow 0 \) the cost function obtained from the algorithm 1 converges to the empirical optimal cost \( \hat{V}_0 \). Let \( u_0 \) be a minimizing sequence for the optimization problem (3) and \( \hat{u}_0 \) be a minimizing sequence for the optimization problem (12).

Note that as a consequence of the result presented in Lemma 1 we have:

\[
\hat{E}_w J (x(0), \hat{u}_0) \leq \hat{E}_w J (x(0), u_0) \tag{14}
\]

From convexity of the optimization problem (3) follows:

\[
E_w J (x(0), \hat{u}_0) \geq E_w J (x(0), u_0) \tag{15}
\]

Since from (7) we have:

\[
\text{Prob} \left( \left| \hat{E}_w J (x(0), u_0) - E_w J (x(0), u_0) \right| < \varepsilon \right) \rightarrow 1
\]

and

\[
\text{Prob} \left( \left| \hat{E}_w J (x(0), \hat{u}_0) - E_w J (x(0), \hat{u}_0) \right| < \varepsilon \right) \rightarrow 1
\]

for all \( \varepsilon > 0 \) as \( \kappa \rightarrow \infty \), equations (14) and (15) imply that \( \hat{u}_0 \) converges to \( u_0 \) as \( \kappa \rightarrow \infty \) and \( \lambda \rightarrow 0 \).

Now, we can write:

\[
\text{Prob} \left( \left| \hat{E}_w J (x(t), \hat{u}_0) - E_w J (x(t), u_0) \right| < \varepsilon \right) \rightarrow 1
\]

for all \( \varepsilon > 0 \) as \( \kappa \rightarrow \infty \), and \( \lambda \rightarrow 0 \).

The proof of the theorem 1 is based on the inequality (7) which is derived from Hoeffding's inequality. In the form that is available in the literature, this inequality can be applied only to the random variables with bounded range. This limits the applicability to the case of bounded disturbance i.e. \( W \subset \mathbb{R} \). An extension to the general case is possible and is a topic for further research.

The optimization problem (3) can therefore be solved by the above algorithm with arbitrary accuracy. The number of required samples, however, grows exponentially with the horizon \( N \). Also the fact that the control input sequence \( \hat{u}_0 \) is computed via iterating several times over the horizon makes this algorithm computationally very intensive. There are various simplifications of the algorithm that lead to a reduction of the computational burden. For example, we do not restrict a relationship between the state at the time \( t + s \) and the input for the sake of the accuracy. Consider however that we impose additional structure such as \( \hat{u}(t + s) = Fx(t + s) \) where we force the controller to be of a linear feedback structure. This makes it possible to simplify the above algorithm considerably in the sense that the required increase in the number of samples required to preserve a certain accuracy with a growing horizon is polynomial instead of exponential as in the general case.

## 5 Numerical example

In this section we compare an MPC scheme utilizing the algorithm from section 4 and the classical MPC scheme. We consider the system of the form (1) with:

\[
A = \begin{bmatrix} 0.7326 & -0.0861 \\ 0.1722 & 0.9909 \end{bmatrix}, \quad B = \begin{bmatrix} 0.0609 \\ 0.0064 \end{bmatrix}
\]

\[
E = \begin{bmatrix} 0.5 \\ 0 \end{bmatrix}, \quad C = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \quad D = \begin{bmatrix} 0.1 \\ 0 \end{bmatrix}
\]

The stochastic disturbance \( w(t) \) is assumed to be uniformly distributed on the interval \([-\alpha, \alpha]\) where \( \alpha \) varies in experiments as \( \alpha = 0.5, 1, 1.5, 2 \).

Our aim is to regulate the system in the origin while fulfilling the following constraint on the input:

\[-2 \leq u(t) \leq 2\]

As a indication of the achieved level of disturbance rejection we use variance of the system state's norm i.e. \( \|x\| \). Simulations are performed over the 200 sec. time interval.

The classical MPC controller is based on the approximate solution to the optimization problem (3) as described in section 4 and optimization is repeated at each time step according to the receding horizon strategy. Because of the small control horizon, we have to sample only disturbance at \( t \), \( w(t) \) and we set \( w(t + 1) = 0 \) according to (9). The number of samples for \( w(t) \) is chosen to be \( \kappa = 20 \), resulting in the 20 samples of the disturbance sequence \( w \).
The simulation results are summarized in tables 1 and 2. Variance is larger when a classical MPC scheme is applied. This difference is expressed as the percentage of the value obtained by controlling the plant with the algorithm proposed in the paper. Note that the performance improvement is the greatest when $\alpha = 1$ and it is getting smaller as $\alpha$ increases. When the level of the disturbance acting on the system is small as in the case $\alpha = 0.5$, the performances are comparable. For small $\alpha$, the constraints are not dominating the performance and both algorithms yield approximately the same performance as the unconstrained problem. For large $\alpha$ the constraints are so restrictive that both algorithms yield approximately the same performance as the open loop performance ($u = 0$).

$$
\begin{array}{|c|c|c|c|c|}
\hline
$\alpha$ & 0.5 & 1 & 1.5 & 2 \\
\hline
\text{var}(\|x\|) & 0.0276 & 0.1472 & 0.4244 & 0.8709 \\
\hline
\end{array}
$$

Table 1: Variance of the system state; MPC by randomized algorithm

$$
\begin{array}{|c|c|c|c|c|}
\hline
$\alpha$ & 0.5 & 1 & 1.5 & 2 \\
\hline
\text{var}(\|x\|) & 0.0277 & 0.1604 & 0.4396 & 0.8797 \\
\hline
\text{p(\%)} & 0.4 \% & 9 \% & 3.5 \% & 1 \% \\
\hline
\end{array}
$$

Table 2: Percentage of the performance loss; standard MPC

6 Conclusion

This paper presents an algorithm to design model predictive controllers which incorporates the effect of (stochastic) disturbances. Although the algorithm is numerically intensive it can obtain the optimum with arbitrary accuracy. Future work is to incorporate simplifications which reduce the computational effort while remaining close to the optimal performance. But, to be able to evaluate the decay in performance caused by the simplifications an algorithm as presented in this paper is needed first.

Other aspects currently under investigation are the issue of state constraints and measurement feedback.

References


