Stochastic Model Predictive Control

Seminar Paper

by

Denny Gert

18. September 2014

presented at the
Center of Computational Engineering Science (MathCCES)
RWTH Aachen

supervised by
Dr.-Ing. Thivaharan Albin
Institute of Automatic Control (IRT)
RWTH Aachen

Contents

1 Introduction

This seminar paper offers a brief overview about control methods for stochastic systems belonging to a class of control methods which is referred to as *Model Predictive Control*. *MPC* methods are based on numerical optimization and are capable of approximating future trajectories of a controlled system in order to determine an optimal control sequence, in which *optimal* is defined by an individual cost function and additional constraints. The trajectory approximation is realized by an internal model of the controlled system and is referred to as *prediction*. So called *certainty equivalent* approaches neglect stochastic influences, resulting in predictions provided by a deterministic prediction model. The need to respect safety constraints may cause a conservative controller design at the expense of control performance. Therefore *Stochastic MPC* approaches take account of stochastic influences in the numerical optimization. This leads to stochastic programs, that are rather difficult to solve and require a high computational effort in general. Several techniques have been developed, that enable a solution with reasonable computational effort. These techniques make certain assumptions on the controlled system and are therefore suited only for a subclass of the general problem formulation. This paper aims at serving as a shallow survey about major approaches for a linear continous system dynamics and hybrid systems. To this end, several *sMPC* techniques are presented in chapter 4, that use a system model from chapter 1 as prediction model. Chapter 5 presents an application example of the *Stochastic Hybrid Optimal Control*.

2 Modelling of stochastic systems

The subsequent chapter 2.1 presents a linear prediction model for input-output timeseries data in *state-space form*. It is suited as description for the continous state evolution of a system and therefore adresses so called *continous systems*. *Hybrid systems* are characterized by an additional discrete state evolution next to the continous and may be described as a *Discrete Hybrid Stochastic Automaton (DHSA)*, presented in chapter 2.2.

2.1 Linear State Space Model

For a *state vector* $x_k \in \mathbb{R}^n$ and *input vector* $u_k \in \mathbb{R}^p$, a system output $y_k \in \mathbb{R}^q$ can be predicted at timestep k as

$$x_{k+1} = A_k x_k + B_k u_k + \eta_k \tag{2.1.1a}$$

$$y_k = C_k x_k + D_k u_k + \xi_k, \tag{2.1.1b}$$

where the matrices A_k, B_k, C_k, D_k and the disturbances η_k, ξ_k depend on time, in case of a time-varying system, as well as on stochastic random variables. The magnitude of disturbances acting through matrices A_k, B_k, C_k, D_k depends on state or input quantities and are therefore labeled as *multiplicative uncertainties*, in contrast to *additive uncertainties* acting through η_k, ξ_k. Equation (2.1.1a) governs the dynamical behaviour of the system. Therefore it is conveniant to recast the primary goal of a control method of driving an output towards a desired setpoint $y \to y_\infty$ as driving the system states towards zero $x \to 0$. This can be achieved by suitable transformations of the system quantities and is straightforward in absence of additive uncertainties, if the structure of the system matrices satisfy certain feasibility conditions. In presence of additive uncertainties, further considerations must be made. In this case, $x \to 0$ can only be achieved in a statistical sense, and system matrices must satisfy certain *mean-square stabilizable* conditions [4].

2.2 Discrete Hybrid Stochastic Automaton

A DHSA consists of four components, namely a *Switched Affine System (SAS)*, *Event Generator (EG)*, *stochastic Finite State Machine (sFSM)* and a *Mode Selector (MS)*. The **Switched Affine System**

$$x_c(k+1) = A_{i(k)} x_c(k) + B_{i(k)} u_c(k) + \eta_{i(k)} \tag{2.2.1a}$$

$$y_c(k) = C_{i(k)} x_c(k) + D_{i(k)} u_c(k) + \xi_{i(k)} \tag{2.2.1b}$$

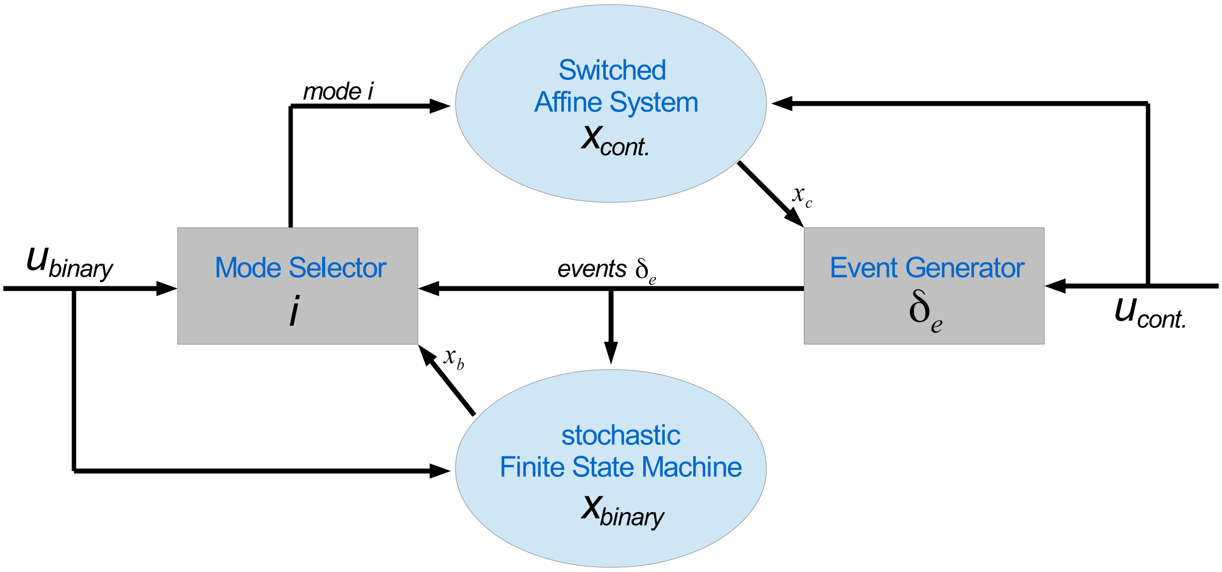

Figure 2.1: Shematic representation of DHSA dependencies

models the continous state evolution of the hybrid system, wherefore appearing quantities are provided with subscript c. It is a state-space model with an additional dependency introduced by *mode i(k)* $\in \mathbb{N}$. This operating mode is determined by the **Mode Selector** f_M for every timestep k as

$$i(k) = f_M(x_b(k), u_b(k), \delta_e(k)), \tag{2.2.2}$$

where $x_b \in \{0,1\}^{n_b}$ denotes binary state variables, which store the discrete state at time k, and $u_b \in \{0,1\}^{m_b}$ denotes binary system inputs. $\delta_e(k) \in \{0,1\}^{n_e}$ contains binary event signals, that might be triggered by the continous dynamics of the system and are therefore produced by the **Event Generator** f_E

$$\delta_e(k) = f_E(x_c(k), u_c(k), k). \tag{2.2.3}$$

The **stochastic Finite State Machine** f_B models the evolution of the binary states x_b. In presence of *enabled* stochastic transitions, the next binary state cannot be certainly predicted due to multiple possible transition paths from the current binary state. For a given binary state $\overline{x_b}$, the *sFSM* contains the probability of $\overline{x_b}$ becoming the active binary state in the next time step:

$$P[\, x_b(k+1) = \overline{x_b}\,] = f_B(x_b(k), u_b(k), \delta_e(k), \overline{x_b}). \tag{2.2.4}$$

The *sFSM* can be reformulated as a deterministic *FSM*

$$x_b(k+1) = f_{B,det}(x_b(k), u_b(k), \delta_e(k), w(k)) \tag{2.2.5}$$

by introduction of *uncontrolled events* $w(k) \in \{0,1\}^l$, where l is the overall number of transitions. $w(k)$ indicates, which transition is taken from $x_b(k)$ to $x_b(k+1)$. For a simulation, w can be generated in every timestep as a vector of pseudo random numbers according to predefined discrete probability distributions. In order to obtain a well-posed problem, certain assumptions must be made on properties of the uncontrolled events, such as mutually exclusiveness and proper probability distributions summing up to one for a decision. A formulation of these assumptions is contained in [1].

There are several possiblities to obtain a computational model from the *DHSA* (2.2.1)-(2.2.4). A convenient choice is the *Mixed Logical Dynamical (MLD)* formulation, which consists of linear equations and a set of linear inequalities. A detailed introduction of the *MLD* formulation can be found in [3]. The principle is to contain the whole dynamics into one equation and realize logical constraints of the automaton by use of binary auxiliary variables and a set of inequality constraints. *MLD* systems are described by

$$x(k+1) = A_k x(k) + B_{1,k} u(k) + B_{2,k} \delta(k) + B_{3,k} z(k) \tag{2.2.6a}$$

$$y(k) = C_k x(k) + D_{1,k} u(k) + D_{2,k} \delta(k) + D_{3,k} z(k) \tag{2.2.6b}$$

$$E_{2k} \delta(k) + E_{3,k} z(k) \leq E_{1,k} u(k) + E_{4,k} x(k) + E_{5,k}. \tag{2.2.6c}$$

Vectors of system quantities u, x, y contain both, continous and binary parts, as for instance the state vector is composed as $x = [x_c \;\; x_b]^T$. δ and z denote binary auxiliary variables of suitable dimension, which are constrained to states and inputs by equation (2.2.6c). Matrices *A, B, C, D, E* can be generated automatically by software packages like HYSDEL[1] or YALMIP [2].

[1]Hybrid System DEscription Language, http://control.ee.ethz.ch/ hybrid/hysdel

[2]A free MATLAB Toolbox for rapid prototyping of optimization problems, http://users.isy.liu.se/johanl/yalmip

3 Optimization in presence of stochastic influence

The optimization performed for an MPC control for a continous system aims at determining an optimal control input sequence for a finite number of future sampling points in time. This interval is referred to as *control horizon* of length N. The resulting optimization problem is formulated as

$$\begin{aligned} \min_{u_{k=0}^{N-1}} \quad & E[J] \\ \text{s.t.} \quad & \text{system dynamics (2.1.1)} \\ & \text{constraints,} \end{aligned} \tag{3.0.1}$$

where the *performance cost* J and additional constraints still need to be defined. J is a function of the future system trajectory, for instance representing the distance of system states to the target state. Due to the stochastic nature of the system, the future system trajectory is not known. Thus, the optimization is performed with respect to the expected cost, as $E[\cdot]$ denotes the expectation value of its argument.

Model Predictive Controllers are embedded in *closed-loop* structures, meaning that control input sequences u are determined in dependency of the state history ψ, which, at a timestep k, contains all system states up to time k

$$\psi(k) = (x_0, \ldots, x_k).$$

Therefore, $u(k)$ can be stated as

$$u(k) = f(\psi(k)).$$

In case of independently distributed uncertainties, the dependency can be reduced to only to the actual state, such that

$$u(k) = f(x_k).$$

In any case, optimal control inputs u^* within the control horizon depend on future state trajectories, which are not known at optimization time due to the stochastic influence. This implies a significant difficulty for determining u^*, as the solution for the control problem (3.0.1) lies in the infinite function space $\mathbb{F}$. Thus, efficient solution methods determine the optimal control sequence for a subset of $\mathbb{F}$, for instance by choice of an ansatz function.

A further difficulty is the calculation of the expected cost, for that the distribution of the performance cost must be known. Generally, this can be only achieved by Monte-Carlo simulations. Once an optimal control input u_k^* is found and applied, a realization of the system trajectory might still cause a higher cost than the expected.

The stochastic optimization problem (3.0.1) enables the possibility to define *probability constraints*, which are one major reason for the performance gain of stochastic MPC approaches. These are limitations on system states, that only need to be respected with a certain probability. Such constraints may be appropriated, if for instance constraints on certain system quantities result from fatigue considerations. In this case, a violation of such a constraint is not crucial and may occur sometimes, but should not for most of the time. As a result of this tolerance, a lower performance cost can be achieved compared to the hard constrained problem.

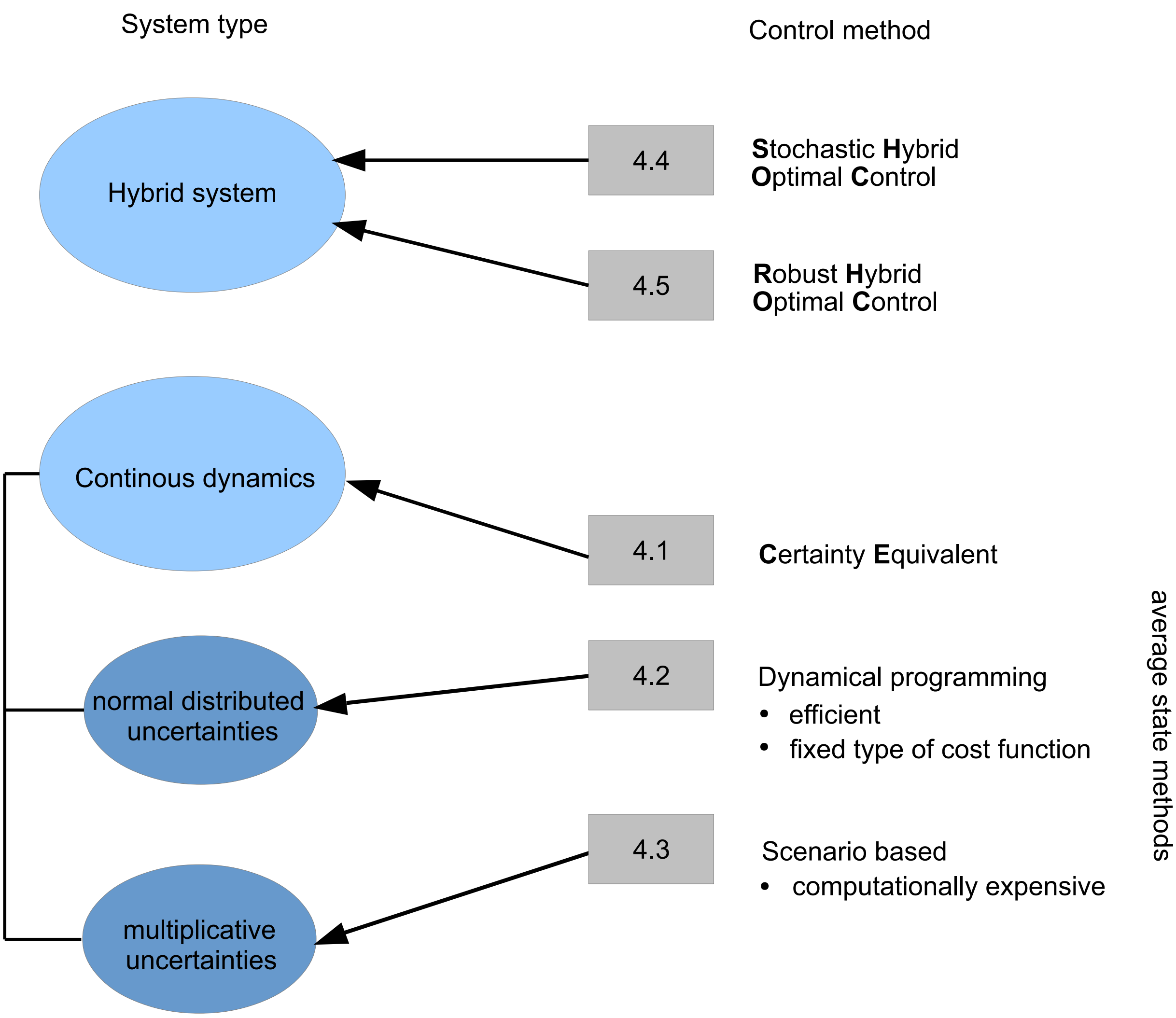

Figure 3.1: Assignment of presented control methods to system model types (numbers denote chapter references)

4 Control methods for stochastic systems

Besides the choice of an ansatz function for the optimal sequence of control inputs, the subsequently presented control methods make on the hand further assumptions on the system and on the other hand further assumptions on the cost function, in order to derive efficient solution methods for the optimization problem (3.0.1). As a result, these methods are only suited for a respective subcategory of systems. Figure 3.1 contains an assignment of these methods to the respective system category.

Assumptions on the cost function lead to two different categories of control methods. The so called *average state control methods* aim at determining a control sequence, which is minimizing a cost function that considers *every possible outcome* of future system trajectories. This leads to the minimization of an expected cost, as stated in chapter 3. The second category makes a certain choice for a future trajectory, that is decisive for determining the control sequence. Methods from the second category like the *Stochastic Hybrid Optimal Control (SHOC)* and the *Robust Hybrid Optimal Control (RHOC)* presented in chapters 4.4 and 4.5, promise a better control performance, if system trajectories are widely spread and do not lie close to an average state.

Assumptions on the system concern the character of the stochastic uncertainties. The underlying probability distribution can be continous or discrete. In a discrete case, the number of possible system trajectories is finite. The optimization with respect to a finite number of possible outcomes lead to *scenario based* approaches, also referred to as *scenario enumeration* methods.

Continous probability densities typically result from parameter identification, disturbances caused by installed devices, or system quantities whose evolution cannot precisely predicted. Uncertainties from this kind typically cause only small deviations from a nominal model and might be described with stationary gaussian distributions. In such a case, statistical properties, e.g. mean and variance, might be exploited to derive a solution method. Arbritrary continous probability distributions can only be tackled by a discretization of probability and a subsequent application of a scenario based method.

4.1 Certainty Equivalent MPC

Certainty equivalent approaches exclude the stochastics from the optimization by usage of a prediction model for uncertainties. Consider for example a time-invariant system with only additive disturbances

$$x_{k+1} = Ax_k + Bu_k + \eta_k.$$

A reasonable prediction model for η_k might be its most probable realization, leading to a replacement with its expectation value $E[\eta_k]$. As a result, the expectation value for the performance cost in (3.0.1) might be omitted, and the optimization problem can be solved efficiently. Once the optimal sequence of control inputs over the control horizon $u_0^*, \dots, u_{N-1}^*$ has been obtained, only the first control move u_0^* is applied, and the horizon is shifted one timestep ahead. Subsequently, a new control sequence is determined by optimization. This strategy is referred to as *receding horizon control* and is inherent to Model Predictive methods. The performance cost J might be calculated with respect to a time interval, which does not correspond to the control horizon. In this case, one speaks of a *prediction horizon*.

4.2 Stochastic MPC for systems with normal distributed multiplicative and additive uncertainties

The approach presented in this chapter is based on [4] and is restricted to uncertainties with known stationary Gaussian distributions. In this case, a linear expansion over m scalar random variables $q^1, \dots, q^m$, uncorrelated and normal distributed with zero mean, can be found with a suitable basis $[A^{(i)}\ B^{(i)}\ d^{(i)}]$ and a nominal model $[\bar{A}\ \ \bar{B}\ \ 0]$:

$$[A_k\ B_k\ \eta_k] = [\bar{A}\ \ \bar{B}\ \ 0] + \sum_{i=1}^{m} [A^{(i)}\ B^{(i)}\ \eta^{(i)}]\ q_k^{(i)}. \tag{4.2.1}$$

Based on this expansion, formulas can be derived, that enable the determination of an optimal control sequence very efficiently, which is a major advantage compared to alternative approaches. Since these derivations involve a special formulation for the performance cost, J is restricted to

$$\begin{aligned} J(x_k, u_k) = &\sum_{i=0}^{\infty}(||E[x_{k+i|k}]||_Q^2 + ||E[u_{k+i|k}]||_R^2) \\ &+ \kappa^2 \sum_{i=0}^{\infty} E\left\{[||x_{k+i|k} - E[x_{k+i|k}]||_Q^2 + ||u_{k+i|k} - E[u_{k+i|k}]||_R^2] - \Theta\right\}, \end{aligned} \tag{4.2.2}$$

where J is already representing the expected cost, and quantities subscripted by $k+i|k$ denote the $k+i-th$ prediction at timestep k. $E[\ \cdot\ _{k+i|k}]$ is meant as the prediction of a quantity by the nominal model used in the expansion (4.2.1). The first term of the expected cost J represents a weighted cost on the average states and inputs with positive definite weighting matrices Q and R. The second term corresponds to a weighting of the expected variance, extended by

$$\Theta = \underset{i \to \infty}{lim} E[||x_{k+i|k}||_Q^2 + ||u_{k+i|k}||_R^2],$$

which is added to ensure that J converges to zero. For $\Theta = 0$, J generally converges to a non-zero value due to the additive disturbances η_k, which do not vanish if $x \to 0$ is achieved. κ is a constant parameter representing a relative weighting of mean and variance terms.

For determining the control sequence, the ansatz

$$\begin{aligned} u_{k+i|k} &= K x_{k+i|k} + f_{i|k}, & i &= 0, 1, \ldots \\ f_{i|k} &= 0, & i &= N, N+1, \ldots \end{aligned} \tag{4.2.3}$$

is chosen. For $f \equiv 0$, (4.2.3) represents the optimal solution minimizing the cost (4.2.2) subject to no constraints, if a constant matrix K_{opt} is found and applied. Due to the quadratic formulation of the cost function, the *optimal feedback gain* K_{opt} can efficiently determined by *dynamical programming recursions* [4]. Therefore, a numerical solution for K_{opt} can be obtained by performing a sufficiently large number of iterations.

For an optimization subject to constraints, the unconstraint optimal solution is augmented by $f_{i|k}$ over the control horizon. Since f does not depend on the state history, it may be determined by solving a standard quadratic program with respect to the linear constraint

$$F \cdot E[x_{k+i|k}] + G \cdot E[u_{k+i|k}] \leq h, \;\; i = 0, 1, \ldots \tag{4.2.4}$$

The hard limit h can be (offline) computed, such that the state remains in a set, for that a desired probability constraint is satisfied. Further, an optimization subject to (4.2.4) might be not feasible. The feasibility can be determined in dependency on the last measured state and is therefore known prior to the optimization. In a non-feasible case, an alternative optimization can be performed, that minimizes the maximum constraint violation. Details can be found in [4], as well as a convergence analysis.

4.3 Scenario based optimization approach

Scenario based approaches require a discrete description of probability distrubtions, given as a set of probabilities $p(k) = [p_1(k), \ldots, p_s(k)]^T$ for s disturbance realizations. A method for discretization of a continous distribution is presented in chapter 4.3.2. It is convenient, to illustrate the set of possible system trajectories graphically as an *optimization tree*, where every node represents a possible system state. The root node corresponds to the measured state at optimization time. Branches of a node represent possible state transitions. Every branch has an assigned realization probability known from the discrete probability distribution. The complexity of the optimization tree grows exponentially with the problem size and the horizon length. Every path of the optimization tree will contribute to the cost function in case of an average-state control, such that scenario based methods require a high computational effort. The size of the optimization tree can be reduced by simply ommitting paths with a low realization probability, or by merging several paths by application of clustering methods (cf. ch. 4.3.2). Once an optimization tree has been constructed, the cost function can be formulated according to the tree description. To this end, let χ denote the set of tree nodes $\chi = \{\chi_1, \chi_2, \dots, \chi_n\}$. Every node has an assigned system state, and a probability π_i to be reached from the root node, such that $\chi_i = \{x_i, \pi_i\}$. Since a state x_i is computed from a continous system model, like for instance the state-space model presented in chapter 2.1, one could also state, that every node has an assigned realization of the system matrices of a state-space model

$$x_i = A_i x_{pre(i)} + B_i u_{pre(i)}, \tag{4.3.1}$$

as well as an assigned control input u_i, if i denotes the numbering of the tree nodes excluding the root node, and *pre(i)* denotes the predecessor of node i. Additive disturbances have not been used here, for reasons adressed in the next chapter. Let further Ω denote the set of leaf nodes. Then a suitable cost function is

$$J = \sum_{i \in \chi \setminus \{\chi_1\}} \pi_i \pi_i x_i^T Q x_i + \sum_{j \in \chi \setminus \Omega} \pi_j u_j^T R u_j \tag{4.3.2}$$

with positive definite weighting matrices Q,R. This corresponds to a weighted sum of all system trajectories. The weighting factors π_i, π_j impose a weighting of a trajectory according to its realization probability. In order to make the resulting optimization problem practically feasible, the infinite function solution space $\mathbb{F}$ must be restricted to a subset, which is adressed in the next chapter.

4.3.1 Offline control law computation with Lyapunov Functions

The prior goal of a control input sequence is to drive the system states to zero, at least in a statistical sense, leading to the mean square stability condition

$$\lim_{k \to \infty} E[\, x_k^T x_k \,] = 0. \tag{4.3.3}$$

It can be shown [2], that condition (4.3.3) is satisfied if the inequality

$$E[V_x(k+1|k)] - V_x(k|k) \leq -x(k|k)^T L x(k|k) \tag{4.3.4}$$

is satisfied for a Lyapunov function $V_x : \mathbb{R}^{n_x} \to \mathbb{R}$, and a strictly positive definite matrix L, which represents a decay rate on the states. With the choice $V_x = x^T P x$ with strictly positive definite P, the expection value can be formulated as

$$E[V_x(k+1|k)] = \sum_{j=1}^{s} p_j(k) x(k+1|k,j)^T P x(k+1|k,j),$$

where $x(k+1|k,j)$ denotes a system state predicted at time k according to the j-th disturbance realization. Making the ansatz

$$u(k|k) = Kx(k|k) \tag{4.3.5}$$

and the substitutions

$$P = Q^{-1}, L = W^{-1}, K = YQ^{-1}$$

with strictly s.p.d matrices Q, W, condition (4.3.4) can be transformed into a *Linear Matrix Inequality (LMI)* problem [2], which can be stated as: find a matrix $\Xi(Q, W, Y, p(k))$ such that Ξ is positive definite, where Ξ is a block-structured matrix, composed of the matrices Q, W, Y and the disturbance realization probabilities $p(k)$. If this LMI problem is solved for every $p(k)$, condition (4.3.4) is satisfied and a control input sequence is obtained (cf. eq. 4.3.5), that guarantees mean square stability on the states. The LMI problem does not depend on the actual state and can be computed offline. Therefore, a control sequence can be computed offline, that is stabilizing the system, but is not minimizing the cost (4.3.2), thus is not the desired optimal solution. The so obtained control sequence might be used as a backup solution, if the online optimization, formulated as follows, cannot be accomplished within a sampling interval. Further, the transformation to an LMI problem is only possible for a system without additive disturbance. Therefore, the control method is restricted to multiplicative disturbances.

In order to define the online optimization problem, the stability condition (4.3.4) is imposed as additional constraint

$$\begin{aligned} \underset{u}{min} \quad & J \\ \text{s.t.} \quad & x_1 = x(k|k) \\ & x_i = A_i x_{pre(i)} + B_i u_{pre(i)}, \;\; \forall i \in \chi \setminus \{\chi_1\} \\ & \sum_{j=1}^{s} p_j(k)(A_j x_1 + B_j u_1)^T P^* (A_j x_1 + B_j u_1) \leq x_1^T (P^* - L^*) x_1 \end{aligned} \tag{4.3.6}$$

where matrices P^*, L^* are known from the offline solution of the LMI problem. A possible choice for the cost J might be eq. (4.3.2).

In a same way as above, a LMI problem can be formulated for a constrained problem, resulting in additional constraints for the problem setup (4.3.6). Details can be found in [2]. Further, the scenario approach enables the possibility to treat probability constraints in a special way. Assume an optimal control sequence $\bar{u}^*$ has been obtained by the optimization problem (4.3.6) subject to a set of additional probability constraints, where each constraint corresponds to a respective system trajectory. This is a standard situation, as generally formulated (nonlinear) probability constraints are typically decomposed to a set of probability constraints, and assigned to a certain system outcome in order to obtain linear inequalities. A scenario tree based on $\bar{u}^*$ can then be constructed, and the relative number of constraint violations can be counted. If the relative number of constraint violations does not meet the maximum tolerance imposed by the probability constraints, the optimization could be repeated with a smaller number of constraints in order to obtain a control sequence u^*, which causes a lower cost but is still respecting the probability limit. There exist several strategies to decide which constraint, and how many constraints might be ommitted for a repeated optimization.

4.3.2 Discretization of continous probability distributions

Continous probability distributions can be discretized in order to employ a scenario based method. This can be done naively, by choice of suitable supporting points. Generally, such a choice is more or less arbitrary and might influence the optimality of the resulting control sequences strongly. A more sophisticated approach is the use of clustering algorithms, as for instance *k-means*. In a first step, a scenario tree is constructed, where the number of branches, respectively the number of trajectories, approaches infinity. This can be realized by Monto-Carlo methods and results in a high number of nodes. The goal of a *k-means* algorithm is to reduce the number of nodes to k, such that every obtained node represents the mean of a cluster of nodes, such that the squared sum of the distances w.r.t corresponding cluster nodes is minimized.

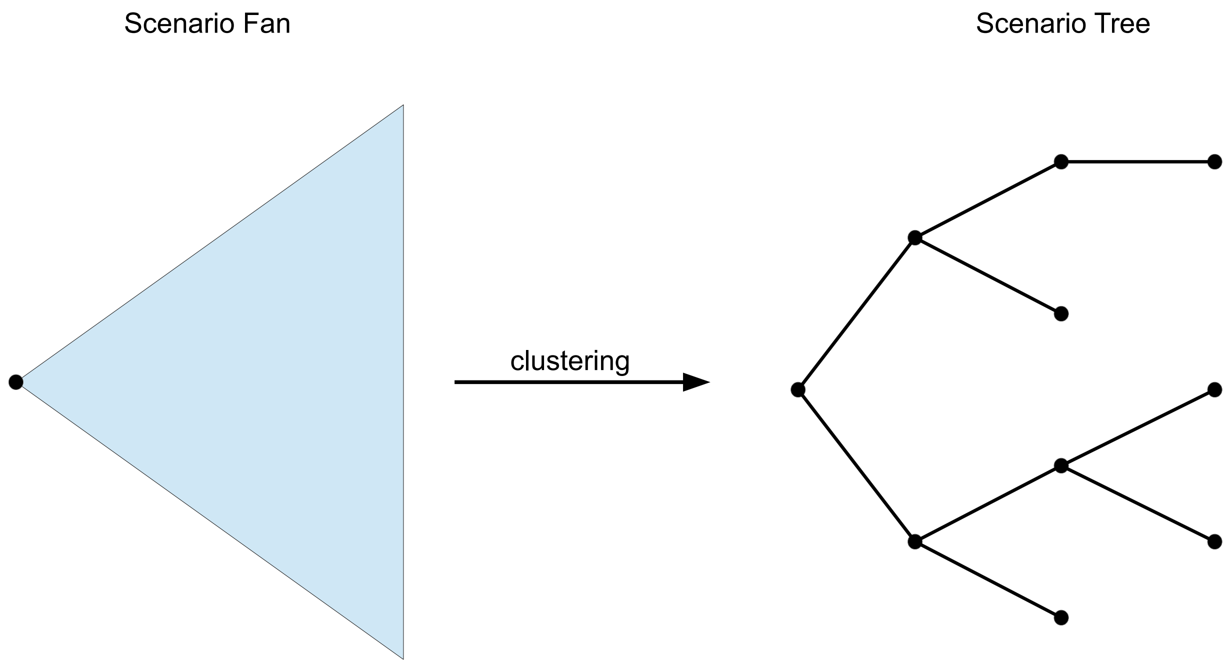

Figure 4.1: Scenario tree construction by clustering

4.4 Stochastic Hybrid Optimal Control

The control methods presented in this chapter and in the subsequent chapter are adressing hybrid systems. As stated in chapter 2.2, uncontrolled events $w(k)$ of a DHSA (cf. eq. 2.2.5) determine which transition leads to the next binary state and therefore represent the stochastic component of the system. The SHOC assumes, that uncontrolled events can be decided. Depending on the choice of the performance cost J, this leads to a *Mixed-Integer Linear Program (MILP)* or *Mixed-Integer Quadratic Program (MIQP)* formulated as

$$\begin{aligned} \min_{\{w(k),u(k)\}_{k=0}^{N-1}} \quad & J + \mu\, J_{prob} \\ \text{s.t.} \quad & \text{DHSA dynamics (2.2.1)} - \text{(2.2.4)} \\ & \text{chance constraint (4.1.3)}, \end{aligned} \tag{4.4.1}$$

where J_{prob} is an additional cost function penalizing unlikely choices of $w(k)$. For linearity, the DHSA dynamics needs to be reformulated in MLD form (eq. 2.2.6). With the probability $\pi = f(w(0), \ldots, w(N-1))$ for a sequence of transitions over the control horizon, the *probability cost* is formulated as

$$J_{prob} = -\, ln\, \pi \;=\; -\sum_{k=0}^{N-1} \sum_{i=1}^{l} w_i(k)\; ln(p_i), \tag{4.4.2}$$

where p_i denotes the probability of transition i at timestep k. Therefore, in case of a weighting factor $\mu = 0$, the control action taken by a SHOC controller is determined as if the stochastic system would behave in an optimal way, where *optimal* is defined only by the performance cost J. For $\mu \to \infty$, the most likely trajectory is decisive. In case of equivalent probabilities for different trajectories, the most benefitial among these will determine the control action. A chance constraint

$$\pi \geq p_{lim} \tag{4.4.3}$$

with *probability limit* p_{lim} might be imposed, to enforce a realization probability on the decisive trajectory. If a time-dependend probability limit $\tilde{p}$(k) is used instead of the fixed value p_{lim} and updated according to

$$\tilde{p}(k+1) = \begin{cases} \frac{\tilde{p}(k)}{P(w^*(k))} & \text{if } \; x_b(k+1) = \hat{x}(k+1) \\ p_{lim} & \text{if } \; x_b(k+1) \neq \hat{x}(k+1), \end{cases}$$

asymptotical convergence in probability to the objective state can be expected under a couple of reasonable assumptions to the system [1]. $P(w^*(k))$ denotes the probability for the decisive transitions $w^*(k)$ and is known from the optimization result.

4.5 Robust Hybrid Optimal Control

The SHOC does not take safety constraint violations into account, that might occur if the system trajectory does not meet the assumed optimal trajectory w^*. Therefore, the RHOC, formulated in [1] in detail, extends the optimization problem formulation (4.1.1) of the SHOC by an additional constraint

$$h(u, w, \delta, z) \leq 0, \; \forall w_i, \;\; \text{s.t } P(w_0 \ldots w_{N-1}) \; \geq \; p_s, \;\; i = 0, \ldots, N-1, \tag{4.5.1}$$

where δ and z are binary auxiliary variables resulting from the reformulation of the DHSA in MLD form (cf. eq. 2.2.6). With the additional constraint, the obtained optimization problem cannot be directly formulated as a MIP, but h can be reformulated in a set of constraints, where each realization of stochastic events determines one constraint. This again referrs to a scenario enumeration approach, which can be avoided by determining optimal control inputs $u^* \in S$ only for a subset $S \subseteq V$, where V denotes the feasible set of control inputs from the SHOC. This leads to following strategy:

1. Obtain $\bar{u}^*$ from SHOC
2. Obtain constraint violations for $\tilde{u}^*$ by solving a reachability problem
   $\exists k : \; h(x(k), \bar{u}_i^*, w_i) > 0$
3. In case of violations, optimize again w.r.t violated constraints and repeat

5 Application Example

Combustion processes are governed by complex fluid dynamics inside the combustion chamber. An extensive prediction can only be realized by time-consuming CFD[1] methods. In order to apply a Model Predictive Control, a simplified prediction model must be found. One possibility is a prediction in terms of a linear model (eq. 2.1.1), which is certainly not capable of accounting the complex nonlinear behaviour of a combustion process, but might be a decent approximation within a small neighbourhood of a stationary operating point. A linear model can be obtained by *parameter identification* from measured input-output timeseries data. Measurements from combustion engines, recorded in the scope of various research projects at the IRT[2], have revealed, that a single linear model is not capable of predicting the captured data in a satisfying manner, eventhough trajectories stay close to the respective operating point. It has been observed, that a better approximation can be obtained by usage of a small number of different linear models, since measurements suggest a system behaviour switching between different *modes* of similar response characteristics. Since the underlying reason for alternations between these modes might only be accessible by CFD computations, a simple approach is the consideration of transitions between system modes in terms of a stochastic model, respectively in terms of a *Markov Chain*. The inherent *Markov Property*

$$P(x_{b,k+1} = \overline{x_{b}}_{,k+1} \mid x_{b,k} = \overline{x_{b,k}} \ \dots \ x_{b,0} = \overline{x_{b}}_{,0}) = P(x_{b,k+1} = \overline{x_{b}}_{,k+1} \mid x_{b,k} = \overline{x_{b}}_{,k})$$

states, that transition probabilities do only depend on the current state of the system, not on the former state history. A comparison with the stochastic Finite State Machine (eq. 2.2.4) yields, that a suitable system model for the above setting can be obtained by application of a DHSA as system model. In order to set up a first implementation, a Stochastic Hybrid Optimal Control has been applied on an imaginary SISO[3] system with transitions between system modes according to a Markov Chain. The resulting DHSA can be defined in the open source Matlab Toolbox YALMIP[4] to automatically setup and solve the optimization problem (4.4.1).

[1] Computational Fluid Dynamics
[2] Institute of Automatic Control, RWTH Aachen
[3] Single-Input-Single-Output
[4] http://users.isy.liu.se

5.1 System model

The controlled system is assumed to operate in 3 alternating modes. The corresponding continous dynamics is governed by models $S1$, $S2$, $S3$, which are constructed according to the specifications presented in Tab. 5.1. Step Responses for the obtained models are pictured in fig. 5.1. $S1$, $S2$, $S3$, formulated in state-space form according to (2.2.1), serve as Switched Affine System.

Model name	Type	Static gain	Damping	Time constant
S1	PT2	1	0.1	1
S2	PT2	1	1	1
S3	Double Integrator	1	-	-

Table 5.1: Continous model specifications

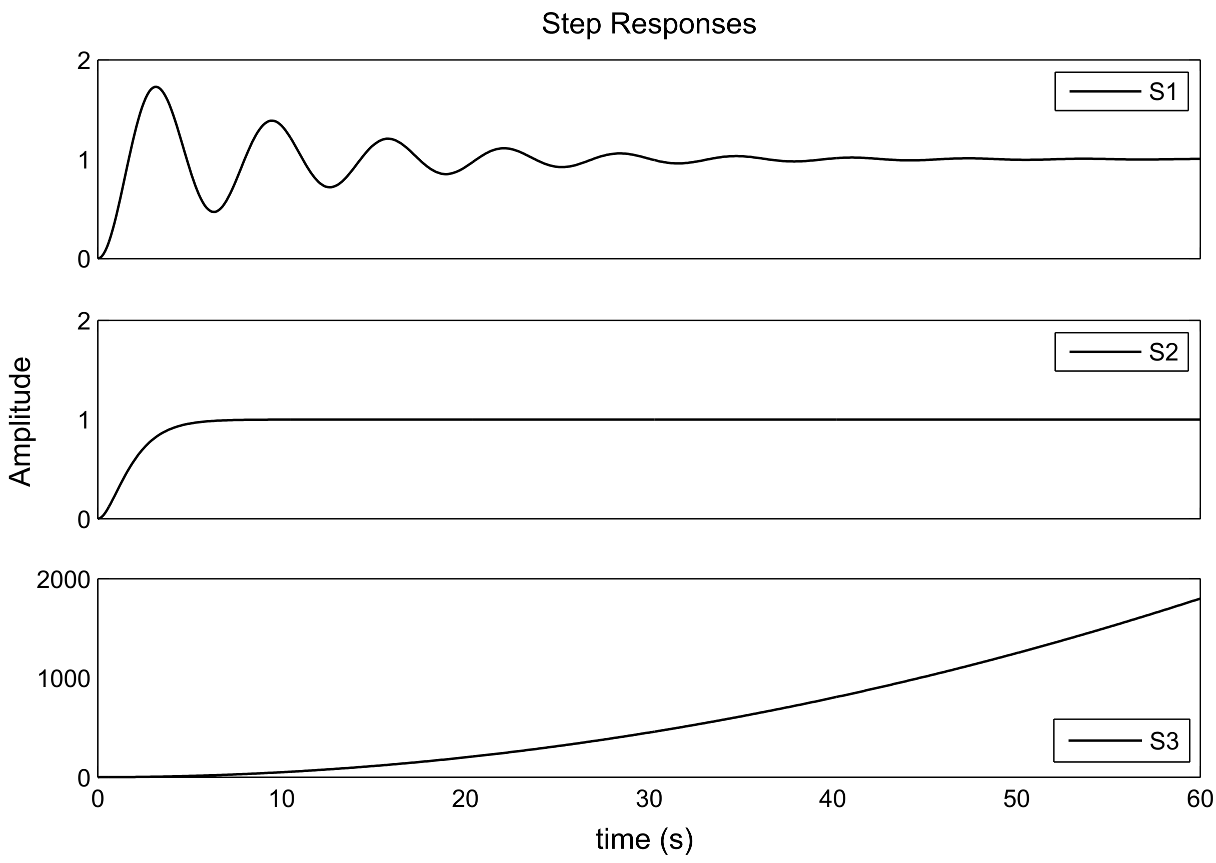

Figure 5.1: Step Responses of continous models

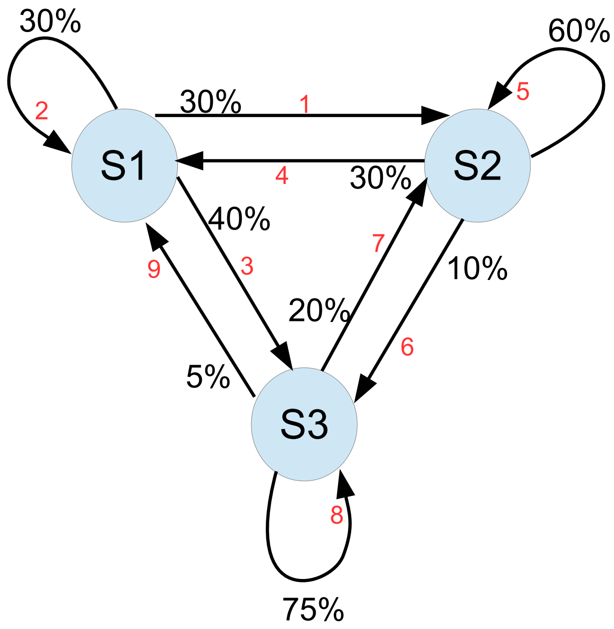

Figure 5.2: Markov Chain model for state transitions (red numbers indicate the numeration of transitions)

Transitions between system modes *S1, S2, S3* happen stochastically according to the Markov Chain model illustrated in fig. 5.2. Given a binary state vector $x_b(k) \in \{0,1\}^3$, indicating which system mode is active at timestep k, transition probabilities at timestep k can be obtained by a matrix vector multiplication

$$p_t(k) = \Theta \cdot x_b(k) = \begin{pmatrix} 0.3 & 0.3 & 0.05 \\ 0.3 & 0.6 & 0.2 \\ 0.4 & 0.1 & 0.75 \end{pmatrix} \cdot x_b(k), \tag{5.1.1}$$

where $\Theta_{i,j}$ is the probability for a transition from the *j-th* mode to the *i-th* mode in one timestep. $p_t(k) \in \mathbb{R}^3$ then contains the probabilites for transition into each state $[1\ 0\ 0]^T, [0\ 1\ 0]^T, [0\ 0\ 1]^T$. For a well-posed setup, column entries of Θ are required to sum up to one and active binary states need to be mutually exclusive. Equation (5.1.1) represents the stochastic Finite State Machine. For an application of the SHOC method, a reformulation as deterministic FSM (2.2.5) is required. To this end, $p_t(k)$ can be used to generate uncontrolled events. Given $x_b(k)$ and a vector of uncontrolled events $w(k) \in \{0,1\}^9$, indicating which transition is taken at timestep k, the next binary state can be obtained according to

$$x_b(k+1) = x_b(k) + N \cdot w(k) = x_b(k) + \begin{pmatrix} -1 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & -1 & 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & -1 & 0 & -1 \end{pmatrix} \cdot w(k). \tag{5.1.2}$$

The entries $N_{i,j}$ of the *Netmatrix* N indicate, if transition j causes mode i to become active or inactive by containing 1 or -1 respectively. For a well-posed setup, the active transition must be mutually exclusive. Equation (5.1.2) serves as deterministic FSM. A definition of the Mode Selector is trivial in this case, as a mode is defined once a transition has been set by generation of the uncontrolled events *w(k)*. By setting

$$w(k) = \delta_e(x), \tag{5.1.3}$$

the generation of the uncontrolled events is realized by the Event Generator.

5.2 Controller design

In order to design a controller according to chapter 4.4, the Event Generator is reformulated as

$$\delta_e(x) = u_b(x). \tag{5.2.1}$$

This states, that uncontrolled events (cf. eq. 5.1.3) are not determined by generation of pseudo random numbers but are obtained as exergeneous system inputs. Using this modification, the optimization problem

$$\begin{aligned} \min_{\{w(k),u(k)\}_{k=0}^{N-1}} \quad & \sum_{k=0}^{N-1} \left(\lambda \, ||y_k - y_{ref}||_2 + \rho \, ||u_k||_2 + \mu \sum_{i=1}^{9} w_i(k) \, ln(p_i) \right) \\ \text{s.t.} \quad & \text{DHSA dynamics} \\ & -30 \leq x_c \leq 30 \\ & -9 \leq u_c \leq 9 \end{aligned} \tag{5.2.2}$$

is formulated. The obtained controller is aware of the exact continous system models and of the exact transition probabilities for computation of the probability cost. For performance comparison, three certainty equivalent controller are designed, that use $S1$, $S2$, $S3$ as single prediction models respectively. Parameters for each controller are presented in Tab 5.2.

Controller name	N	λ	ρ	μ
SHOC	10	50	0.2	$5{\cdot}10^5$
CE-S1	10	50	0.2	-
CE-S2	10	50	0.2	-
CE-S3	10	50	0.2	-

Table 5.2: Control parameter settings

5.3 Performance results

For each controller, 100 simulations have been done with a simulation time of 60 seconds and a sampling time of 0.1 seconds. The initial states are set to zero and the output setpoint is set to $y_\infty = 10$.

For a comparison of results, the *root mean square deviation (rmsd)*

$$rmsd = \sqrt{\frac{\sum_{i=1}^{n}(y_i - y_\infty)^2}{n}}$$

has been calculated for each ensemble, for a time interval $10s \leq t \leq 60s$. As Figure 5.4 indicates, this time interval corresponds to trajectories, for that the setpoint has been already met. The *rmsd* values for each ensemble are presented in Figure 5.3. Among the certainty equivalent controllers, the usage of system model *S2* obtains the best result in average, while the SHOC controller is able to reduce the mean *rmsd* about roughly 50%.

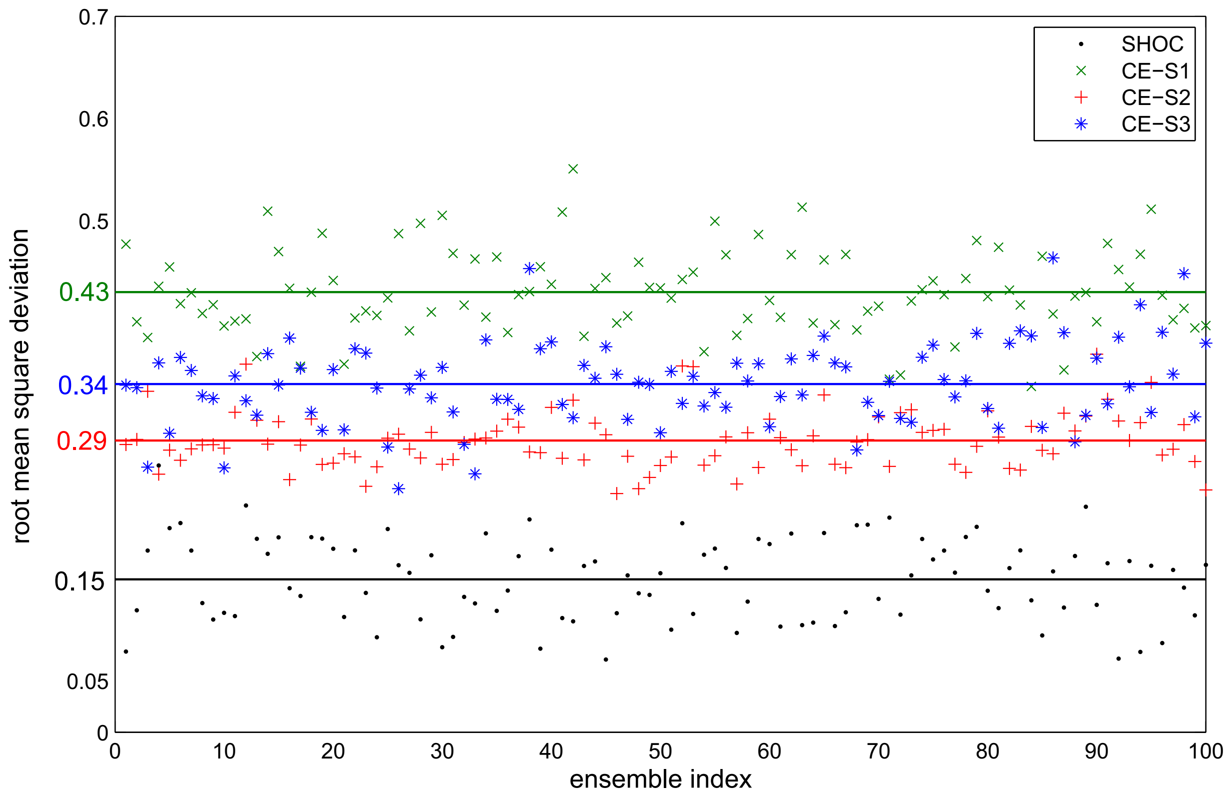

Figure 5.3: Root mean square deviations (straight lines denote mean values)

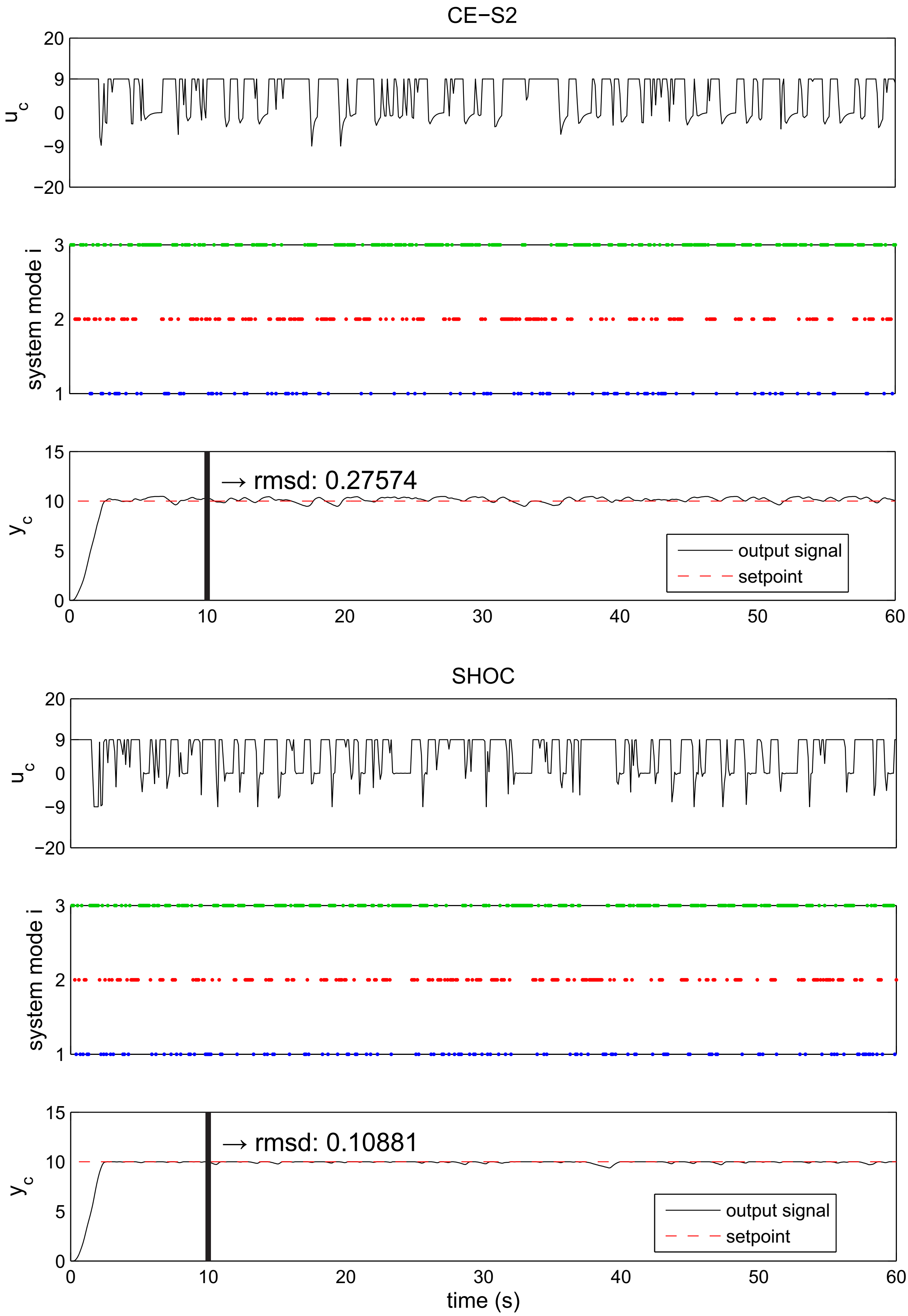

Figure 5.4: Simulation results of CE-S2 (top) and SHOC (bottom) for 2 ensembles

References

[1] A. Bemporad and S. D. Cairano. Optimal control of discrete hybrid stochastic automata. *Lecture Notes in Computer Science*, 3414:151–167, 2005.

[2] A. Bemporad and S. D. Cairano. Scenario-based model predictive control of stochastic constraint linear systems. In *Joint 48th IEEE Conference on Decision and Control and 28th Chinese Control Conference*, December 16–18, 2009.

[3] A. Bemporad and M. Morari. Control of systems integrating logic, dynamics, and constraints. *Automatica*, 35:407–427, 1999.

[4] M. Cannon. Stochastic model predictive control: State space methods. Lecture notes, http://users.ox.ac.uk, 2008.