International Journal
Information Technology Applications (ITA)
Volume 10, Number 2, December 2021

# ROBUST EXPLICIT CONTROL OF COMPLEX HYBRID DISCRETE PROCESSES

**Štefan Kozák**

***Abstract:***

*The paper deals with design of simple end effective robust explicit control of complex hybrid systems. We consider discrete hybrid systems consisting of several discrete state-space models with parametrical uncertainties. The parametric uncertainties are considered in coefficients of state space model (A) and (B). The main goal of the paper is to be proposed and verified the robust explicit hybrid control algorithms such for stable and for unstable complex dynamical systems with respect to parameter model uncertainties. Proposed robust explicit controller method is based on the iterative algorithm's computation of control law parameters for the known target polytope with respect to parameters uncertainties and constraints on state, control and output variables. For demonstration ability of the proposed methods, we verified the several examples. An example is presented to illustrate the details of the proposed robust hybrid parametric controller design. In this paper we consider hybrid dynamical model consisting of 4 state-space model with 30% uncertainties of matrix (A) coefficients. We compare obtained numerical and graphical results with solution which are obtained from very known MPT toolbox. Our proposed method solution in all cases gives a stable solution while the MPT does not guarantee the stability for tested cases. The proposed method and algorithm is convenient for application for SISO and MIMO processes.*

***Keywords:***

*Hybrid control, explicit controller, robust control, robust hybrid control, parametrical uncertainties.*

***ACM Computing Classification System:***

*Information systems, Information systems applications, Process control systems.*

## Introduction

Robust model predictive control (MPC) of hybrid systems is important class of constraint model based control methods that can explicitly account presence of modelling uncertainties in the controlled process. Explicit MPC is control method where MPC optimization problem is solved off-line with multiparametric programming methods. Result of this optimization is partitioning of state space into disjunct partitions each of which is assigned to function of system state. Control input is than calculated in 2 steps. First step is location of current state into one of state space partitions. Second step is evaluation of corresponding function which gives current optimal input.

In this paper we design different type of explicit controller whose intent is to supply robust control input which is probably, instead of optimal one. Proposed controller consists of set of *(m+n)* dimensional (*m* – number of system inputs, *n* – number of system states) polytopes which represent state-input space. To get robust control input we first need to locate current system state into one of state-input space polytopes. Result of slicing this polytop at values of current state is *m*-dimensional polytope which defines range of robust system inputs.

## 1 Structure of Model

In this article we use discrete hybrid state space models. As it is hybrid model it contains more nominal submodels with defined area of operation. Following is structure of the model:

$$\begin{aligned} &\boldsymbol{x}(k+1) = A_{Ai}\boldsymbol{x}(k) + B_{Ai}\boldsymbol{u}(k) \\ &H_i x(k) \le K_i \end{aligned}$$

where $x$ is real vector representing state of model,

$x \in \bigcap R_i{}^n$

$u$ is real vector representing input of model,

$u \in \bigcap R_i{}^m$

$A_{Ai}$ is actual real matrix of proper dimension

$B_{Bi}$ is actual real matrix of proper dimension

$R_i^n$ is n-dimensional state space polytope

$R_i^m$ is m-dimensional input space polytope

$H_i, K_i$ are matrices defining area of operation of $i$-th submodel

As we would like to propose robust hybrid control we also use parametrical uncertainties on every model parameter. Actual model parameters are then calculated as follows:

$$\begin{aligned} A_{Ai} &= A_{Ni}(1 + A_{Ri}A_{Ui}) \\ B_{Ai} &= B_{Ni}(1 + B_{Ri}B_{Ui}) \end{aligned}$$

where $A_{Ni}$, $B_{Ni}$ are nominal matrices $A$, $B$

$A_{Ui}$, $B_{Ui}$ are matrices of uncertainties of $A_{Ni}$, $B_{Ni}$

$A_{Ri}$, $B_{Ri}$ are random matrices, $A_{Ri}, B_{Ri} \in \langle -1, 1 \rangle$

Matrix products in these equations should be interpreted piecewise.

Thus we have defined nominal model, its uncertainties and the way how actual model is calculated from nominal model and its uncertainties. $A_{Ui}$ and $B_{Ui}$ define maximum possible change of nominal values in $A_{Ni}$, $B_{Ni}$ respectively and $A_{Ri}$, $B_{Ri}$ determine actual change. This way we get $2^{(m+n)}$ of extreme actual models - each random coefficient is set either to 1 or -1.

## 2 Control Algorithm Design

As model itself contains parametrical uncertainties we are not able to bring it exactly to target point. It is always necessary to define some tolerance, $n$-dimensional polytope around the target, which is accurate enough to fulfil our requests. This target polytope can be of any size bigger than 0 and have to contain target point.

We would like to propose robust control which drives system state into defined target polytope. We do not care if the control is optimal or not. In this article we would like to design iterative method for computing robust control of hybrid systems. The simplified algoritm of this process can be described as follows:

| | |
|---|---|
| **Step 1** | Set the control target to defined target polytope |
| **Step 2** | Find state space and associated inputs space from which it is guaranteed to get to target polytope in one step if parametrical model uncertaities are met |
| **Step 3** | Set the target polytope to computed state space |
| **Step 4** | Go to Step 2 and repeat until explored state space is big enough |

The core of the algorithm is *Step 2* which we describe in next section:

**Step 2a** We process one vertex of target polytope so that we make backwards simulation of particular extreme actual model (model with maximal possible uncertainty) using all combinations of inputs extremes. If we have *m* inputs we deal with $2^m$ of input extremes combinations and after backward simulation we get $2^m$ states. Backward simulation is done using this formula:

$$x(k) = A_{Ai}^{-1}\left[x(k+1) - B_{Ai}u(k)\right]$$

If we combine these states with associated inputs we get *(m+n)* dimensional polytope. Important property of this polytope is that it contains all states from which it is possible to get to target polytop vertex in one step. Moreover if we cut this polytope at values of selected state we get accurately defined inputs which controls model into target polytope vertex. This property comes from linearity of model.

**Step 2b** We repeat *Step 2a* for all target polytope vertices. Thus we get new polytope which has similar property than previous one. If we cut it at values of selected state, we get range of inputs each of which controls model into target polytope. Example of such polytope for model with 2 states and one input is depicted in (Fig.1).

Now we know how to control actual model (which is derived from nominal model using one of extreme parametric uncertainties) into target polytope in one step. Solution to this problem is mentioned "states-inputs" polytope which defines states from which we are able to control model into target polytope in one step and also defines input range which can be used for every state.

**Step 2c** We repeat *Step 2b* for all possible parametrical uncertainties extremes and get set of similar "states-inputs" polytopes - one for each uncertainty extreme.

Conjunction of all these polytopes is again polytope (call it "partial solution" polytope) which defines all possible states which we can drive into target polytope if system meets defined model uncertainties. It also defines range of inputs which can be used to achieve this target. The mentioned polytopes conjunction is depicted in (Fig.2).

**Step 2d** As we are working with hybrid model which consists of linear submodels each of which is valid only on specified state space we have to trim "partial solution" polytope so that it does not exceed area of validity.

After repeating this procedure for every submodel of our hybrid model, final "solution" polytope array of *Step 2* is union of partial solution polytopes.

Very important condition is that original target polytope has to be a subset of "solution" polytope array which we get after first iteration. Otherwise we are not able to guarantee successful control. (Fig.3) shows final solution - polytopes array.

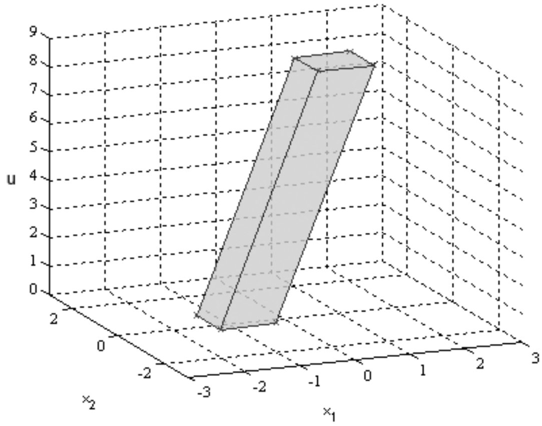



Fig.1. State-input polytope defining states from which we can control actual model into target polytope in one step. I also defines possible inputs range for particular states.

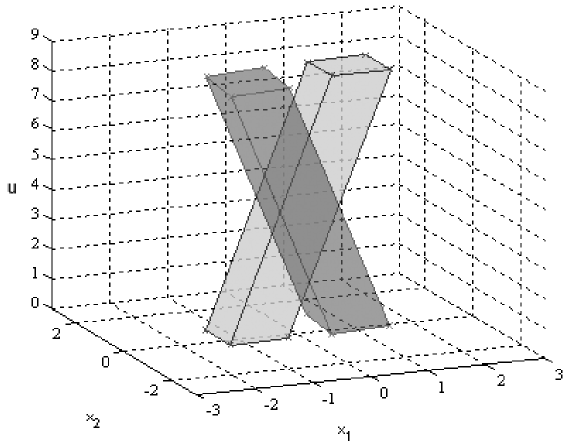



Fig.2. Conjunction of "states-inputs" polytopes.

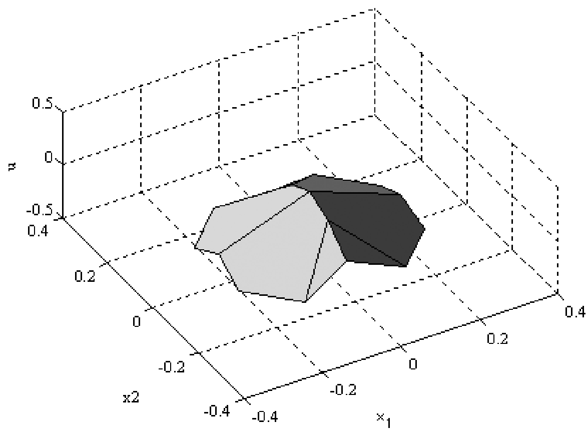



Fig.3. Solution polytopes array describing all states and corresponding inputs from which we can control model into final polytope in one step.

### 2.1 Control algorithm

Result of controler design is array of polytopes arrays which were created during iterations through control design algoritm. These polytopes define subset of state space for which we have defined control inputs. Let´s index these arrays based on order of iteration they were created.

First step of control is to define polytope that contains current state. We start searching from lower polytope index to higher and use first positive result. First polytope that contains current state defines also inputs range that guarantees that model state will be controlled towards target. As here we do not care about control performance we can select and implement input which meets following inequality:

$$H_i \begin{bmatrix} x_k \\ u_k \end{bmatrix} \leq K_i$$

where $x_k$ is current state of model,

$u_k$ is real current input of model,

$H_i, K_i$ are matrices defining area of operation of $i$-th controller polytope

In every next step we repeat this procedure.

## 3 Case Study

In this case study we design explicit controller for discrete hybrid model consisting of 4 discrete linear submodels each of which has 2 states and 1 input. The goal of control is to get system state into predefined surround of coordinate system.

Model used in the simulation is defined by following matrices:

$$A_{N1} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}, B_{N1} = \begin{bmatrix} 1 \\ 0.5 \end{bmatrix}, H_1 = \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix}, K_1 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$A_{N2} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}, B_{N2} = \begin{bmatrix} -1 \\ -0.5 \end{bmatrix}, H_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, K_2 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$A_{N3} = \begin{bmatrix} 1 & -1 \\ 0 & 1 \end{bmatrix}, B_{N3} = \begin{bmatrix} -1 \\ 0.5 \end{bmatrix}, H_3 = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}, K_3 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$A_{N4} = \begin{bmatrix} 1 & -1 \\ 0 & 1 \end{bmatrix}, B_{N4} = \begin{bmatrix} 1 \\ -0.5 \end{bmatrix}, H_4 = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}, K_4 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

Uncertainty matrices for all submodels are defined as follows:

$$A_U = \begin{bmatrix} 0.3 & 0.3 \\ 0.3 & 0.3 \end{bmatrix}, B_U = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

System input constraints were defined as follows:

$$-10 \le u \le 10$$

Target polytope is defined as follows:

$$\begin{bmatrix} 0 & -1 \\ 1 & 0 \\ 0 & 1 \\ -1 & 0 \end{bmatrix} x \le \begin{bmatrix} 0.1 \\ 0.1 \\ 0.1 \\ 0.1 \end{bmatrix}$$

After algorithm passed 20 iterations we obtained robust controler which is shown in (Fig.4).

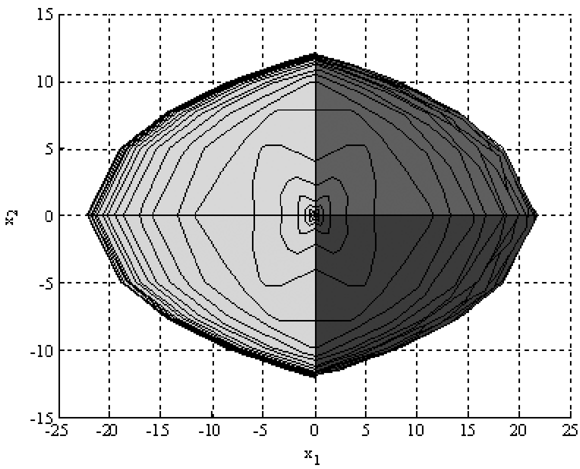



Fig.4. Partitioning of state space by explicit hybrid controller.

To simulate system control for every simulation step we have chosen different actual model based on model uncertainties. (Fig.5) shows different state trajectories based on what actual model is used in simulation, however every time system state ends in target polytope. (Fig.6) shows corresponding control inputs.

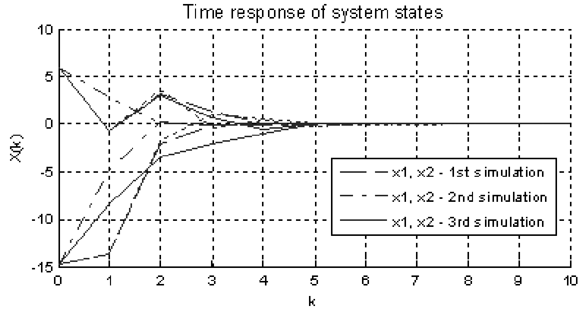



Fig.5. Different state trajectories based on actual model.

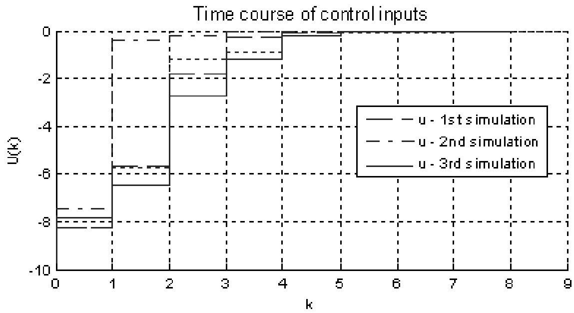



Fig.6. Different course of control inputs based on actual model.

Finally we have created optimal controller using MPT toolbox. We have created it so that it does not take into account any model uncertainties. To simulate this controller we used model with uncertainties. As expected this controller was not able to control the system from every initial state. To show difference between optimal and robust controller we selected some initial states which optimal controller was not able to control or not enough good, and made control simulation with optimal and robust controller. Results of this comparison are shown in (Fig.7) and (Fig.8).

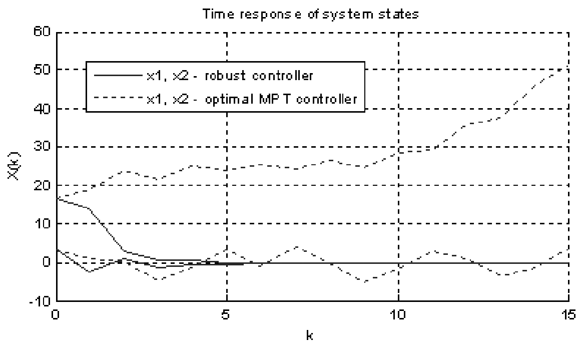



Fig.7. Comparison of state trajectories of optimal and robust controler.

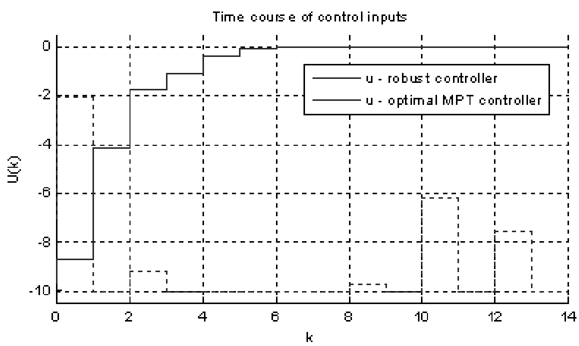



Fig.8. Comparison of control inputs of optimal and robust controler.

## Conclusion

This paper deals with design of robust control for discrete hybrid systems. The main result is a proposal of explicit hybrid controller which is able to control even non-stable hybrid system to desired state with arbitrary precision bigger than 0.

## References


[1] Bemporad. A. (2004). *Hybrid toolbox – user's guide*. http://www.ing.unitn.it/~bemporad/hybrid/toolbox

[2] Bemporad. A. (2007). *Model Predictive Control of Hybrid Systems*. http://www.dii.unisi.it/hybrid/school07/pdf/08.Bemporad.pdf

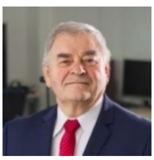
[3] Bemporad. A. (2009). Modeling and control of hybrid dynamical systems: The Hybrid toolbox for MATLAB. in Proc. MATHMOD Conference, I. Troch and F. Breitenecker, Eds, *ARGESIM Reports*, 82–100. Vienna, Austria

[4] Kvasnica. M., Grieder. P., and Baotic. M. (2004). *Multi-parametric toolbox (MPT).* http://control.ee.ethz.ch/~mpt/

[5] Kvasnica. M. (2009). *Real-time model predictive control via multi-parametric programming: theory and tools*. VDM Verlag, Saarbruecken

[6] Makýš. M., Kozák. Š. (2011). *Effective method for design of traffic lights control*. In IFAC 18th World Congress: Proceedings; Milano, Italy; 29 August - 2 September 2011 (2011), 14934-14939. Milano, Italy


## Authors




**prof. Ing. Štefan Kozák, PhD.**
Faculty of Informatics, Pan-European University, Bratislava, Slovakia
stefan.kozak@paneurouni.com

Currently at the Institute of Applied Informatics at the Faculty of Informatics, Pan-European University in Bratislava. His research interests include system theory, linear and nonlinear control methods, numerical methods and software for modeling, control, signal processing, IoT, IIoT and embedded intelligent systems for digital factory in automotive industry.