

# **PAROC DEMO User Manual**

Justin Katz, Baris Burnak, Iosif Pappas, Efstratios N. Pistikopoulos

Artie McFerrin Department of Chemical Engineering

Texas A&M University

College Station, TX, United States

The logo for PAROC, featuring the word "PAROC" in a bold, black, sans-serif font.

## Contents

1. INTRODUCTION .....	3
2. HIGH FIDELITY MODEL .....	4
3. MODEL APPROXIMATION .....	4
4. MULTIPARAMETRIC MPC BUILDER .....	5
5. CLOSED LOOP SIMULATION .....	6

# 1. INTRODUCTION

The presence of uncertainty in process systems is one of the key reasons for deviation from set operation policies. As these uncertainties realize themselves on different time scales such as on a control, scheduling or design level, an integrated, comprehensive approach to consider uncertainty is required.

For a detailed explanation of the PARAmetric Optimization and Control (PAROC) framework, the reader is directed to the full length paper<sup>1</sup> or the group's website<sup>2</sup>. This software enables the user to construct multiparametric model predictive controllers (mpMPC) from a given high fidelity model. The key steps are as follows:

- Introducing a user defined high fidelity model
- Model approximation via system identification
- Developing a Multiparametric Model Predictive Control
- Simulating closed loop dynamics

This software is developed in the MATLAB<sup>®</sup> environment, and requires the POP<sup>3</sup> and YALMIP<sup>4</sup> toolboxes to be installed on the user's local machine. These toolboxes can be downloaded from their respective research group's website. Any questions or comments can be sent to [paroc@tamu.edu](mailto:paroc@tamu.edu).

Downloading the PAROC software comes with a sample problem in the form of a nonisothermal CSTR.

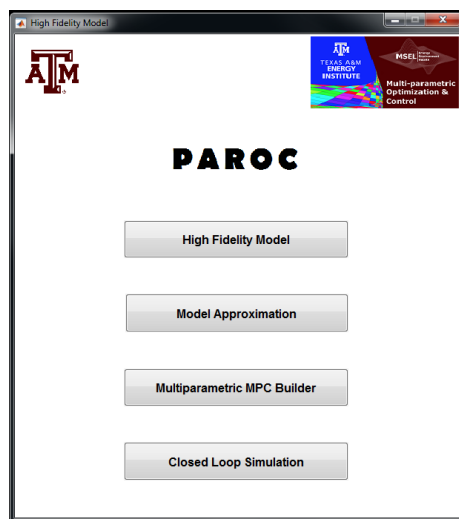


Figure 1. PAROC software interface.

---

<sup>1</sup> Pistikopoulos, E. N.; Diangelakis, N. A.; Oberdieck, R.; Papathanasiou, M. M.; Nascu, I.; Sun, M. PAROC-An integrated framework and software platform for the optimisation and advanced model-based control of process systems. Chemical Engineering Science 2015, 136, 115-138.

<sup>2</sup> [paroc.tamu.edu](http://paroc.tamu.edu)

<sup>3</sup> <http://parametric.tamu.edu/POP/>

<sup>4</sup> <https://yalmip.github.io/>

## 2. HIGH FIDELITY MODEL

In the first dialogue box, the user must enter the name of the function file associated with their high fidelity model. The function name should be present in the current working directory, and it must be in the format that MATLAB<sup>®</sup> ode functions accept.

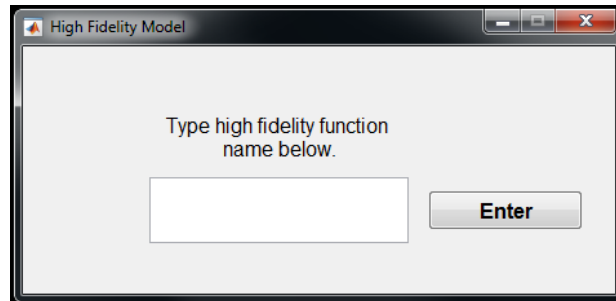


Figure 2. High fidelity model.

The user can use the provided high fidelity model, `cstrODE`, as a reference.

## 3. MODEL APPROXIMATION

After the high fidelity model has been introduced, the model approximation step requires a user defined sampling time, initial conditions, and number of inputs. After selecting the number of inputs from the drop-down menu, the user must define upper and lower bounds for each input. The software returns the normalized root mean squared error (NRMSE), which is an error criteria that MATLAB<sup>®</sup> uses to assess the quality of the approximate model with respect to each output, in a message box.

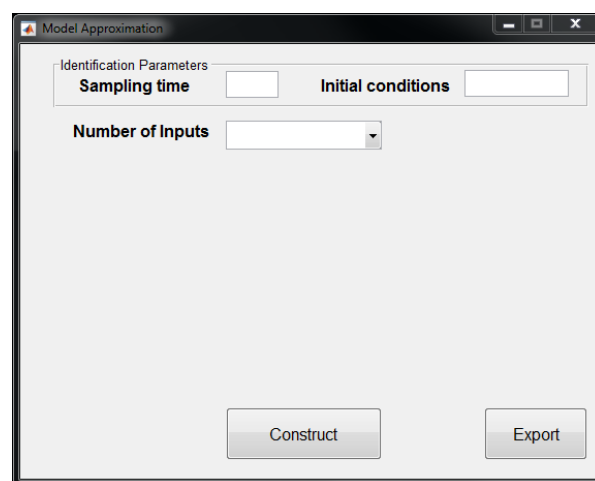
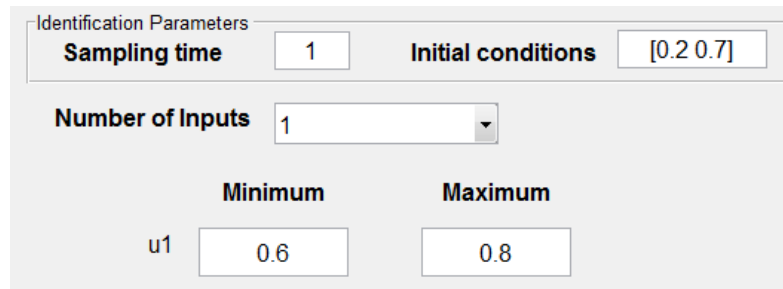


Figure 3. Model approximation.

In this demo version, the maximum allowable number of inputs is 4. Also, the initial conditions must be provided as a vector inputs of the appropriate size as seen in Figure 4.



Identification Parameters

Sampling time: 1      Initial conditions: [0.2 0.7]

Number of Inputs: 1

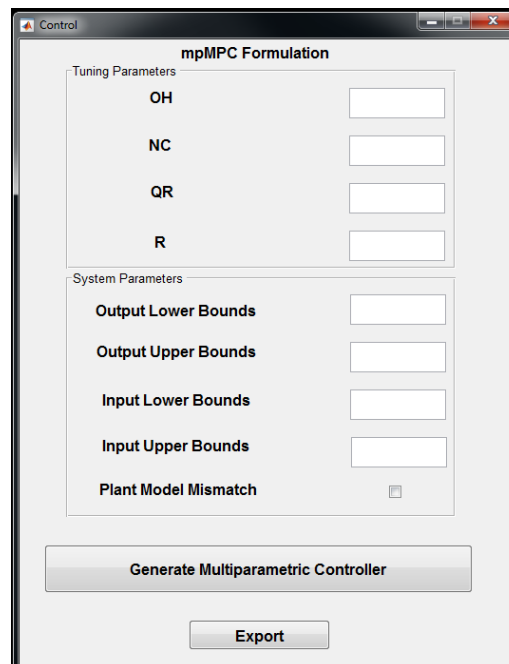
Minimum: 0.6      Maximum: 0.8

u1

Figure 4. Example user inputs for model approximation

## 4. MULTIPARAMETRIC MPC BUILDER

With the construction of the approximate model, the necessary mpMPC parameters are required to develop the controller, as seen in Figure 5. The user is asked to provide the following: Output horizon (OH), control horizon (NC), set point deviation penalty matrix (QR), input deviation penalty matrix (R), and upper and lower bounds for inputs and outputs. As an addition, the user may select if there is plant model mismatch to include an error term in the control formulation that accounts for the expected output vs. the measured output.



Control

mpMPC Formulation

Tuning Parameters

OH: [ ]

NC: [ ]

QR: [ ]

R: [ ]

System Parameters

Output Lower Bounds: [ ]

Output Upper Bounds: [ ]

Input Lower Bounds: [ ]

Input Upper Bounds: [ ]

Plant Model Mismatch: ☐

Generate Multiparametric Controller

Export

Figure 5. mpMPC generation.

The output horizon must be greater than or equal to the control horizon, and they must be scalar quantities. The penalty matrices, QR and R, are square matrices that must have the standard

properties associated with MPC formulations. The upper and lower bounds must be expressed as vector quantities of the appropriate size. These guidelines can be seen as follows, in Figure 6.

The figure shows a software window titled "Control" with a sub-header "mpMPC Formulation". It is divided into two main sections: "Tuning Parameters" and "System Parameters".

**Tuning Parameters:**

- OH:** Input field contains "3".
- NC:** Input field contains "1".
- QR:** Input field contains "[1e3 0; 0 1e3]".
- R:** Input field contains "1e1".

**System Parameters:**

- Output Lower Bounds:** Input field contains "[0;0]".
- Output Upper Bounds:** Input field contains "[1;1]".
- Input Lower Bounds:** Input field contains "0".
- Input Upper Bounds:** Input field contains "1".
- Plant Model Mismatch:** A checkbox that is checked.

At the bottom of the window, there are two buttons: "Generate Multiparametric Controller" and "Export".

Figure 6: Example user inputs for mpMPC generation.

## 5. CLOSED LOOP SIMULATION

To ensure quality results coming from the mpMPC that has been generated, a closed loop simulation can be performed. The user must define set points for the outputs and reference points for the inputs. The user must also specify initial conditions for the system along with total run time for the system.

The figure shows a software window titled "closedLoop". It is divided into two main sections: "Set Points" and "System Conditions".

**Set Points:**

- Set Points:** Input field is empty.
- Input Reference:** Input field is empty.

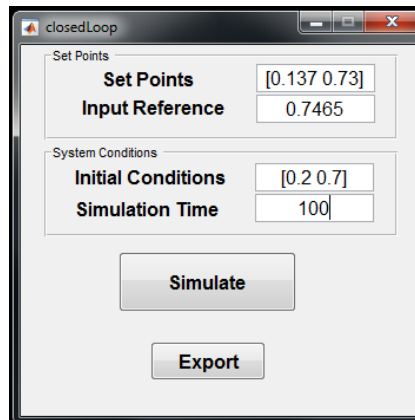
**System Conditions:**

- Initial Conditions:** Input field is empty.
- Simulation Time:** Input field is empty.

At the bottom of the window, there are two buttons: "Simulate" and "Export".

Figure 7. Closed loop simulation.

The set points, input reference, and initial conditions must be provided as vector quantities of the appropriate size, as seen in Figure 8.



The screenshot shows a MATLAB/Simulink-style window titled 'closedLoop'. It contains two main sections: 'Set Points' and 'System Conditions'. In the 'Set Points' section, 'Set Points' is set to '[0.137 0.73]' and 'Input Reference' is set to '0.7465'. In the 'System Conditions' section, 'Initial Conditions' is set to '[0.2 0.7]' and 'Simulation Time' is set to '100'. At the bottom, there are two buttons: 'Simulate' and 'Export'.

Section	Parameter	Value
Set Points	Set Points	[0.137 0.73]
	Input Reference	0.7465
System Conditions	Initial Conditions	[0.2 0.7]
	Simulation Time	100

Figure 8. Example user inputs for closed loop simulation.

This step outputs the closed loop dynamics of the high fidelity model for a given simulation time. A figure is generated for each output and input, seen as follows.

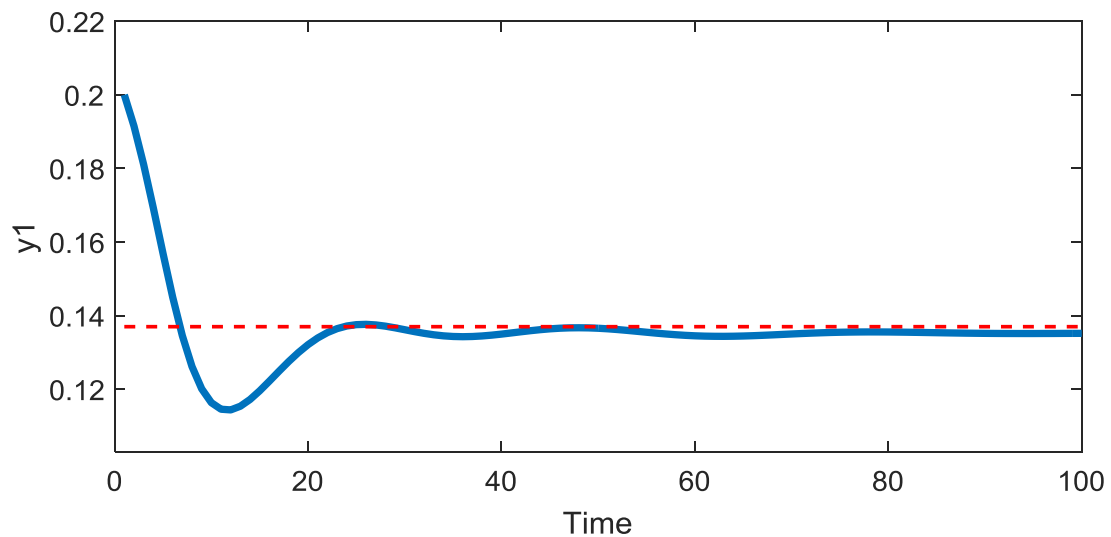


Figure 9. Output profile of the closed loop response for the first output.

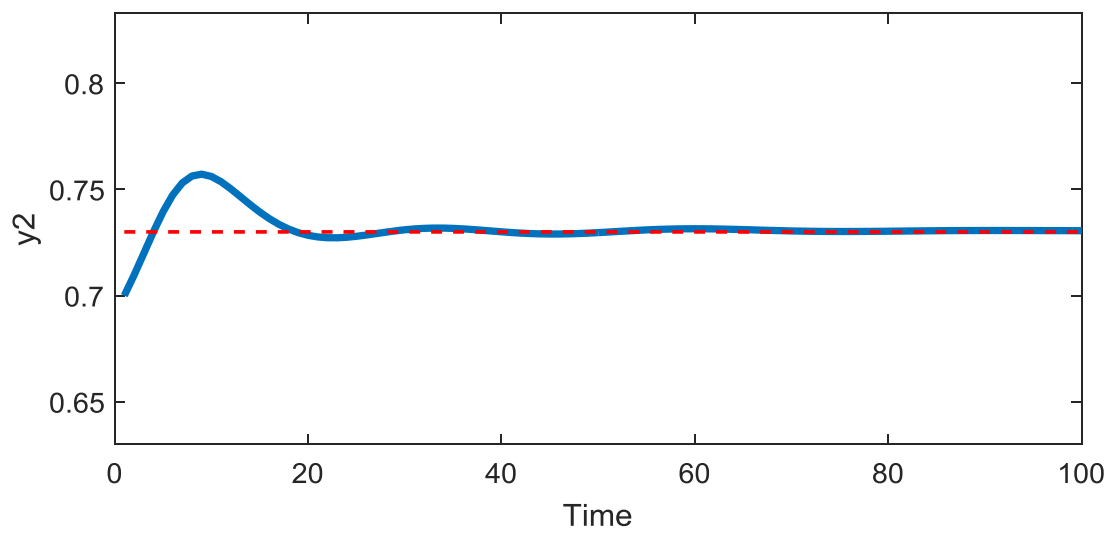


Figure 10. Output profile of the closed loop response for the second output.

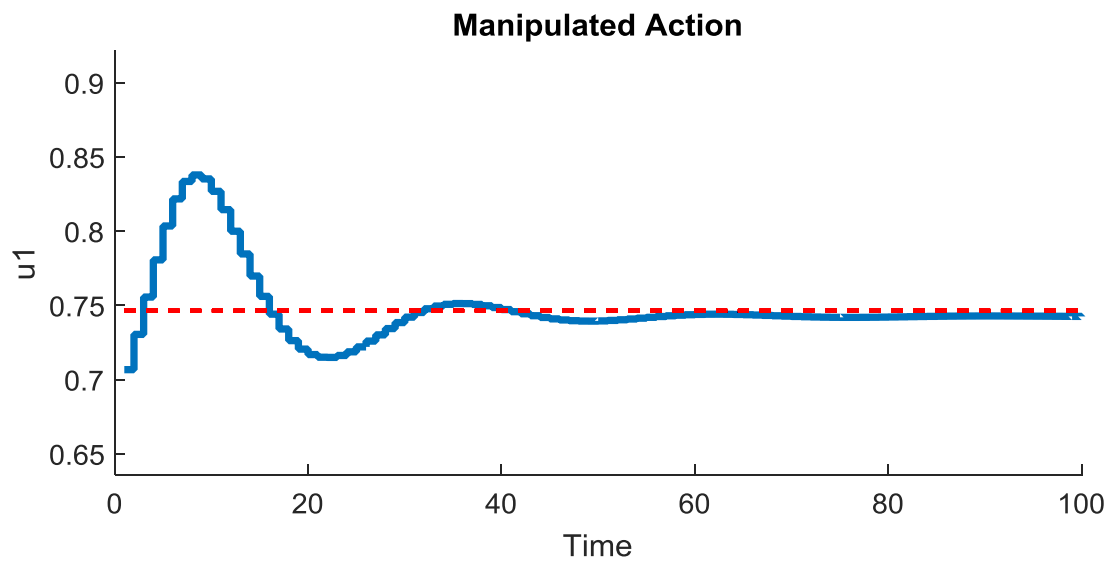


Figure 9. Input profile of the closed loop response for the manipulated action.