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

Contents

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[1] or the group's website[2]. 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® environment, and requires the POP[3] and YALMIP[4] 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.

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

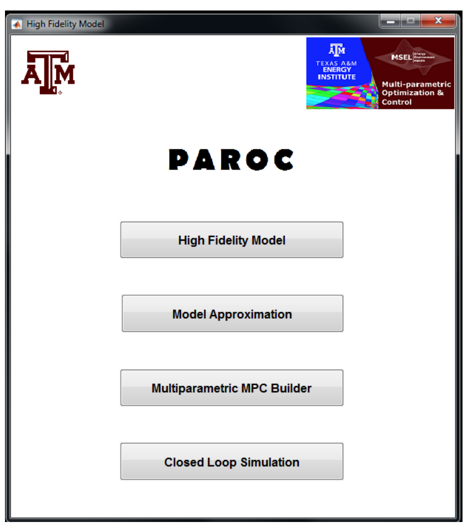

Figure 1. PAROC software interface.

---

1 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.

2 paroc.tamu.edu

3 http://parametric.tamu.edu/POP/

4 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® `ode` functions accept.

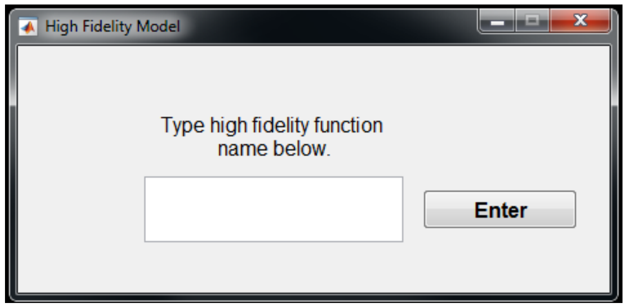

Figure 2. High fidelity model.

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® uses to assess the quality of the approximate model with respect to each output, in a message box.

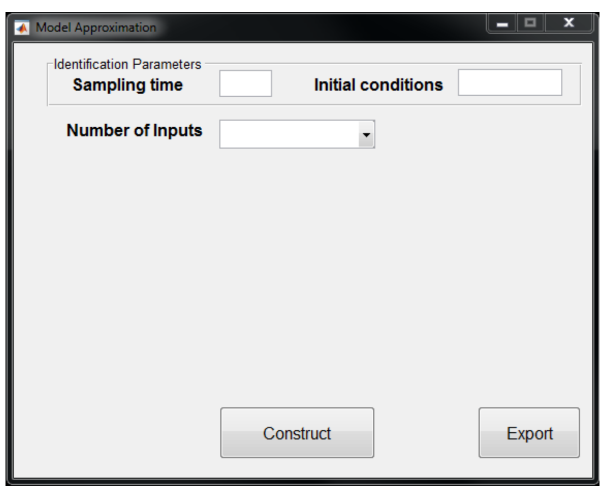

Figure 3. Model approximation.

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.

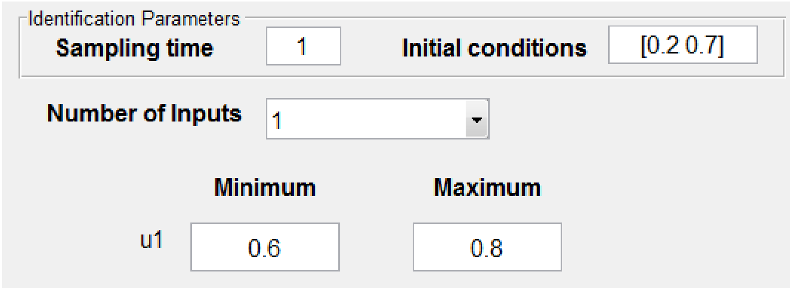

Figure 4. Example user inputs for model approximation

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.

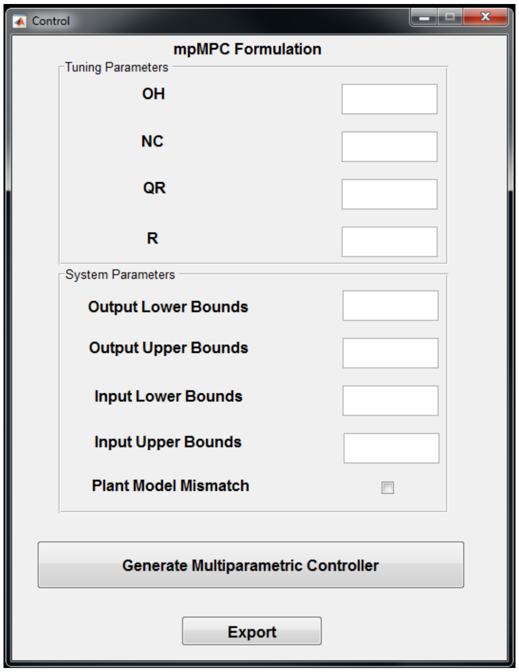

Figure 5. mpMPC generation.

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.

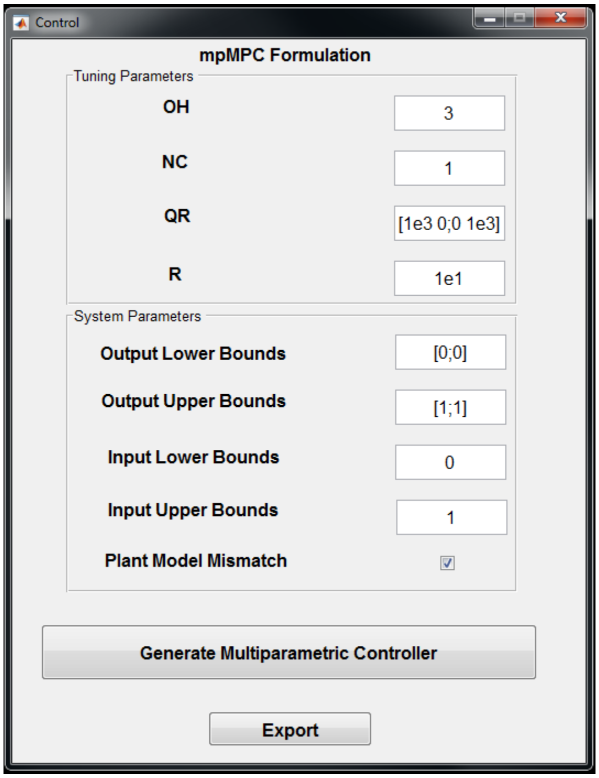

Figure 6: Example user inputs for mpMPC generation.

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 ouputs and reference points for the inputs. The user must also specify initial conditions for the system along with total run time for the system.

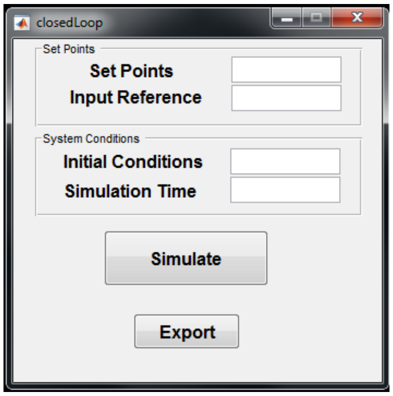

Figure 7. Closed loop simulation.

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

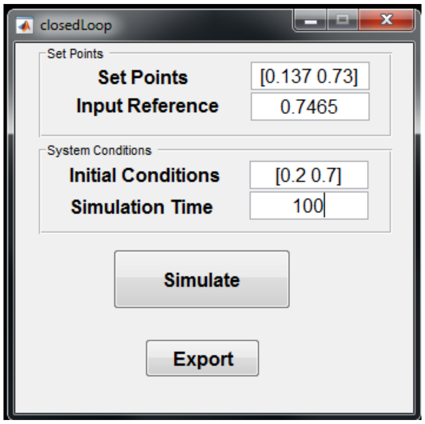

Figure 8. Example user inputs for closed loop simulation.

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.

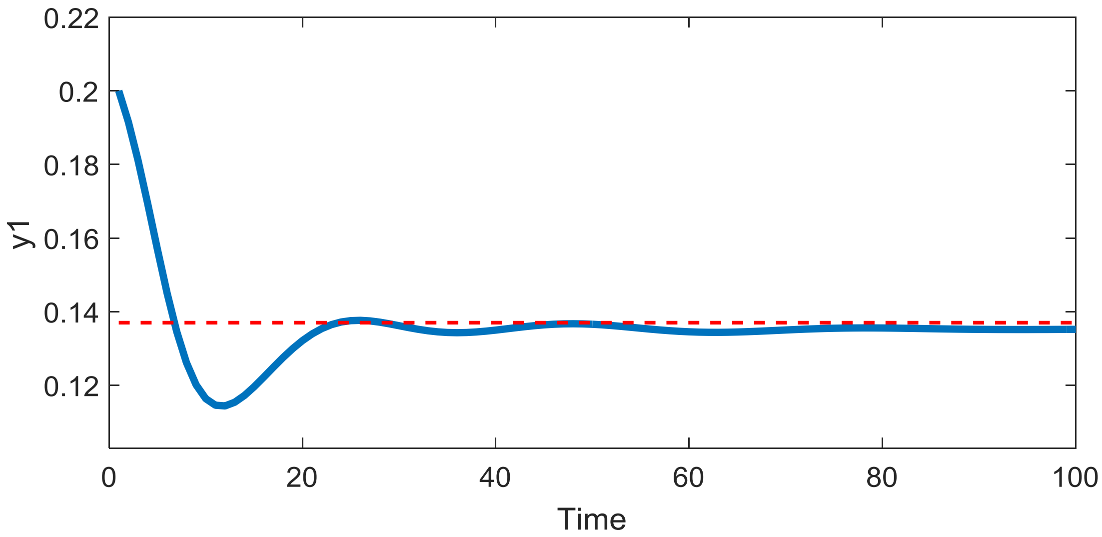

Figure 9. Output profile of the closed loop response for the first output.

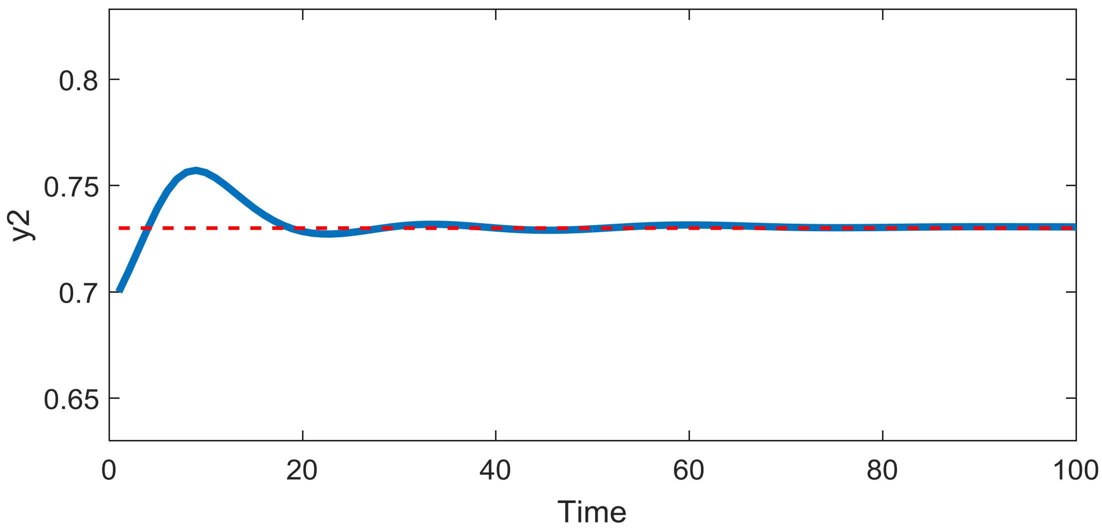

Figure 10. Output profile of the closed loop response for the second output.

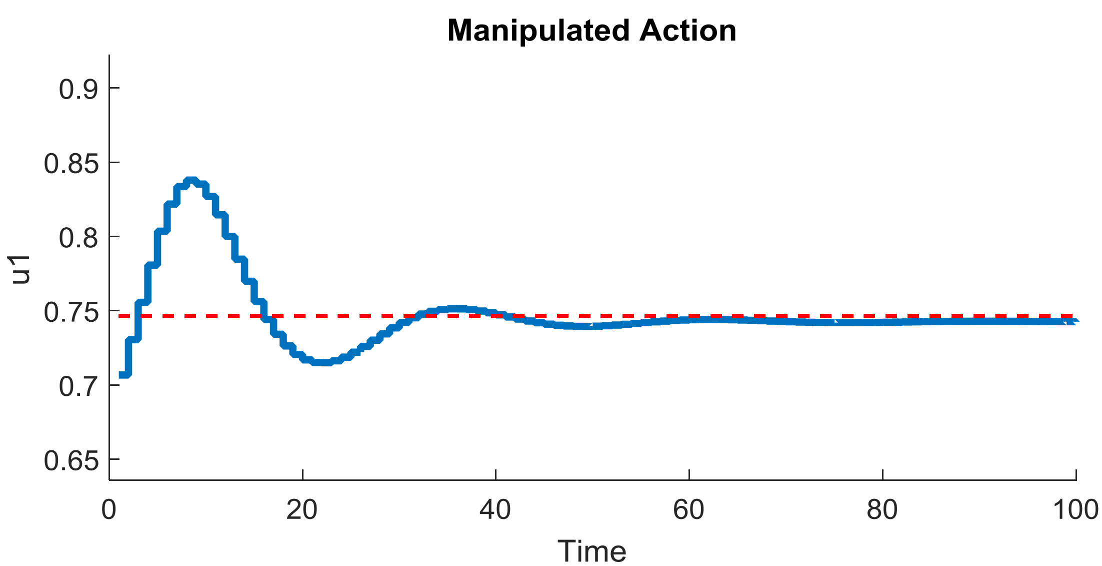

Figure 9. Input profile of the closed loop response for the manipulated action.