

Real-Time Model Predictive Control of a Laboratory Liquid Tanks System

I. Rauová, M. Kvasnica, L. Čírka, and M. Fikar

Institute of Information Engineering, Automation and Mathematics,
Slovak University of Technology in Bratislava,
81237 Bratislava, Slovakia

Abstract

This paper deals with real-time implementation of Model Predictive Control (MPC) of a laboratory liquid tanks system using the dSPACE platform. The MPC problem is solved using parametric programming techniques, which allow closed-form solution to the underlying optimization problem to be obtained off-line in a form of a look-up table. Once such a table is calculated, the subsequent implementation reduces to a simple set-membership test, which can be performed very efficiently on-line. In the paper we present a step-by-step description of all steps leading towards the derivation and implementation of such a controller for a laboratory device.

1 Introduction

This paper deals with real-time implementation of Model Predictive Control (MPC) of a laboratory liquid tanks system. In MPC [4] the control objectives are translated into an optimization problem, which is formulated over a finite prediction horizon. The result of the optimization is a sequence of optimal control moves which drives the system states (or outputs) towards a given reference while respecting system constraints (such as upper and lower limits on the tanks' levels) and minimizing a selected performance criterion (e.g. the regulation error). Traditionally MPC is implemented in a so-called Receding Horizon fashion, where the optimal control problem is re-solved at every time instance for a new value of the measured initial conditions. This induces a significant computational load at each sampling time, which might be prohibitive if not enough computational power is available, or if the sampling time is too short. To ease the computational demands at each step we show that if the underlying optimization problem is solved using parametric programming techniques [1, 2], the resulting optimal MPC feedback law takes a form of a look-up table. Implementation of such a table can be done very efficiently on-line, as the evaluation of the feedback law involves only matrix multiplications, additions, and logic comparisons. Therefore the implementation can be done much faster compared to traditional on-line MPC techniques.

Motivated by these upsides, in this paper we show how parametric MPC can be synthesized and applied to control a laboratory liquid tanks system. The plant consists of two interconnected tanks. The control objective is manipulate the pump in such a way that the level of the lower tank tracks a prescribed reference signal. In the paper we show how to synthesize a parametric solution to the MPC problem using the Multi-Parametric Toolbox (MPT) [3]. First, we introduce the plant itself and derive its corresponding mathematical model. Then, we show which commands have to be used to set up the MPC problem and solve it using MPT. We show that once the parametric MPC controller is calculated in Matlab, it can be directly embedded into a Simulink scheme where it acts as a C-coded S-function. The controller code is written in a way such that the Real-Time Workshop can be used to compile and download the controller evaluation code directly into a control platform. In our setup we have used the dSPACE setup which provides input/output interfaces to our laboratory plant and is also capable of running the controller code in real-time. A state observer is used to compensate the lacks of measurements of certain signals.

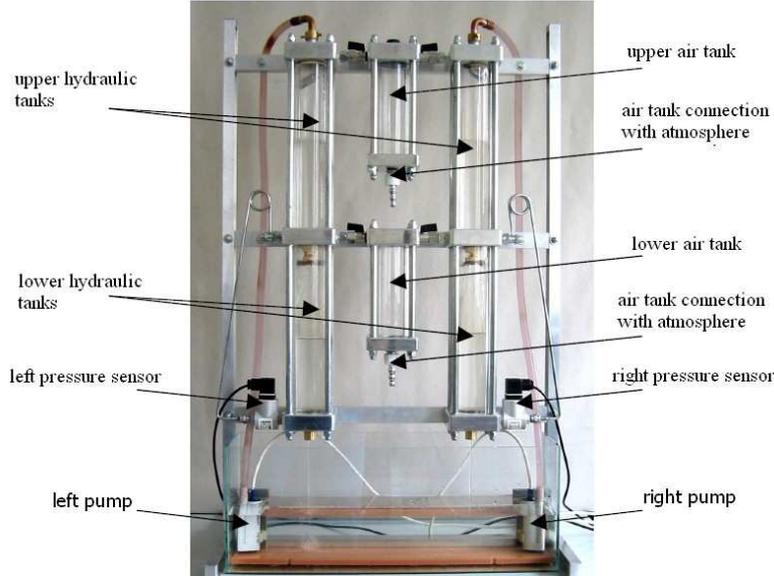


Figure 1: The laboratory liquid tanks equipment.

2 Physical Setup

The laboratory liquid tanks device, as depicted in Figure 1, is comprised of two pairs of interconnected tanks. Each pair consists of two glass tanks of circular cross-section, with liquid inlet situated on the top of the upper tank. The upper and lower tanks are connected by a small opening, which allows the liquid from the upper tank to flow into the lower tank by forces of gravity. Another opening is located at the bottom of the lower tanks and the outflowing liquid is captured in a reservoir at the bottom of the device. A pump then remits the liquid back to the top of the upper tanks. The height of the tanks is 310 mm, the tank diameter is 50 mm, and the diameter of each opening is 4 mm. In this work we only consider control of the left pair of the tanks.

The liquid levels in the lower tanks is measured by pressure sensors, which automatically convert the measurements into a voltage signal. The pumps can also be controlled by a voltage signal, which is proportional to the throughput of the pumps. The input and output signals are wired to a dSPACE DS 1104 I/O card, which provides an interface between the plant and Matlab. The voltage signals are connected to the I/O card by means of BNC connectors.

As the experimental device provides voltage signals in the range of $[-10 \text{ V}, 10 \text{ V}]$ for each of the physical quantities, the following relation could be used to convert the pump voltage signal $q_{[V]}$, i.e. the value in volts, into the corresponding quantity expressed in cm^3s^{-1} units:

$$q_{[\text{cm}^3\text{s}^{-1}]} = 0.0114q_{[V]}^3 - 0.4327q_{[V]}^2 + 5.3468q_{[V]} - 0.5778 \quad (1)$$

Similar formula relates the actual tank level $h_{[\text{cm}]}$, expressed in centimeters, and the voltage signal from the level sensor $h_{[V]}$:

$$h_{[V]} = 0.3676h_{[\text{cm}]} - 0.8941 \quad (2)$$

The mathematical model of the left tank pair can be captured by a set of two differential equation of the following form:

$$F \frac{dh_1}{dt} = q_0 - k\sqrt{h_1}, \quad (3)$$

$$F \frac{dh_2}{dt} = k_1\sqrt{h_1} - k\sqrt{h_2}. \quad (4)$$

Here, $F = 12.566 \text{ cm}^2$ denotes the cross-sectional area of the tanks, h_1 and h_2 denote, respectively, the liquid levels in the upper and lower tank, q_0 is the liquid inflow, and $k = 3.667 \text{ cm}^2 \cdot \text{s}^{-1}$ is a constant representing the resistance of the openings connecting the tanks. By linearizing (3)–(4) around steady-state values $h_1^s = h_2^s = 21.44 \text{ cm}$ and $q_0^s = 16.9811 \text{ cm}^3 \cdot \text{s}^{-1}$, the following state-space model can be derived:

$$\dot{x} = Ax + Bu, \quad (5)$$

$$y = Cx, \quad (6)$$

where $x = [(h_1 - h_1^s), (h_2 - h_2^s)]^T$ is the state vector, $u = q_0 - q_0^s$ is the input, and $y = h_2 - h_2^s$ is the measured output. Based on the steady state values mentioned above, the numerical values of matrices A and B are given by

$$A = \begin{pmatrix} -\frac{k}{2F\sqrt{h_1^s}} & 0 \\ \frac{k}{2F\sqrt{h_1^s}} & \frac{k}{2F\sqrt{h_2^s}} \end{pmatrix}, \quad B = \begin{pmatrix} \frac{1}{F} \\ 0 \end{pmatrix} \quad (7)$$

This linear state-space representation can then be used to find a closed-form representation of the MPC feedback law by using techniques of parametric programming as described in the next section.

3 Model Predictive Control

In model predictive control, the optimal control actions are found by optimizing for the predicted plant behavior while taking process constraints into account. This is usually achieved by formulating and solving an optimization problem where a given performance criterion is minimized subject to the constraints. A model of the plant is employed as an additional constraint to capture the process evolution in time. If the plant model is given as a discrete-time linear state-space model, the following Constrained Finite Time Optimal Control (CFTOC) problem could be used to solve for the optimal control inputs:

$$\min_{\Delta u_0, \dots, \Delta u_{N-1}} \sum_{k=0}^{N-1} \|R\Delta u_k\|_p + \|Q(y_k - y_{\text{ref}})\|_p \quad (8a)$$

$$\text{s.t. } x_0 = x(0) \quad (8b)$$

$$x_{k+1} = Ax_k + Bu_k \quad (8c)$$

$$y_k = Cx_k \quad (8d)$$

$$x_k \in \mathcal{X} \quad (8e)$$

$$y_k \in \mathcal{Y} \quad (8f)$$

$$u_k \in \mathcal{U}. \quad (8g)$$

Here, p denotes a matrix norm (either $p = 1$, $p = 2$, or $p = \infty$), the integer $N < \infty$ represents the prediction horizon, and R and Q are weighting matrices used to tune performance of the MPC controller. The linear model in (8c) serves to predict the future states based on the knowledge of the initial state $x(0)$, which is assumed to be measured. The optimization is performed over the increments Δu_k to provide an offset-free tracking of the reference trajectory y_{ref} . The state, output, and input constraints are represented, respectively, by the polyhedral sets \mathcal{X} , \mathcal{Y} , and \mathcal{U} .

MPC is usually implemented in the so-called *receding horizon* fashion. Here, the optimal solution to the CFTOC problem (8) is found for a particular value of $x(0)$, which results into the optimal sequence $[\Delta u_0^*, \dots, \Delta u_{N-1}^*]$. Out of this sequence, only the first element (i.e. Δu_0^*) is actually implemented to the plant and the rest is discarded. At the next time instance, a new initial state measurements $x(0)$ is obtained and the whole procedure is repeated. This repetitive optimization is performed in order to introduce feedback into the whole procedure and to deal with possible disturbances and plant-model mismatches.

If the initial state $x(0)$ and the value of the reference signal y_{ref} are both known, the CFTOC problem (8) can be solved either as a quadratic program (QP) for $p = 2$, or as a linear program (LP) for $p = 1$ and $p = \infty$. Even though efficient polynomial-time algorithms exist to solve both types of problems, the time needed to perform the optimization can be prohibitive if the sampling time is too short, or if the implementation hardware is very simple and thus less capable. To address this issue, in their seminal work [1] the authors have shown (for a quadratic type of performance indices) how to solve the CFTOC problem (8) *parametrically* for all admissible initial conditions $x(0)$ by employing techniques of *parametric programming*. In this approach the optimal solution to (8) is found as an explicit state-feedback law parameterized in the initial condition $x(0)$. The advantage of the parametric solutions is that the optimal control input can be obtained in real-time by simply evaluating a look-up table. The main result of the parametric approach is summarized by the following theorem.

Theorem 3.1 (Explicit solution to the CFTOC problem (8) [2]) *The optimal solution to the CFTOC problem (8) is a piecewise affine function of the initial state $x(0)$*

$$\Delta u_0^* = F_r x(0) + G_r \quad \text{if } x(0) \in \mathcal{R}_r \quad (9)$$

where $\mathcal{R}_r = \{x(0) \mid H_r x(0) \leq K_r\}$ is a set of polytopic regions, and F_r and G_r are the matrices of the affine state-feedback law active in the r -th region.

Theorem 3.1 shows that the optimal solution to the CFTOC problem (8) can be found as a *look-up* table consisting of r components. Therefore, once the table is calculated, MPC can be implemented in real time by simply evaluating the table for the actual measurements of $x(0)$. The table can be calculated efficiently using e.g. the Multi-Parametric Toolbox [3]. Performance of the MPC scheme can be tuned by appropriately adjusting the weighting matrices Q and R , and by a suitable choice of the prediction horizon N .

4 Results

In this section we show how MPC could be used for control of the laboratory liquid tanks equipment described in Section 3. The control objective is to drive the level in the lower tank to a time varying reference signal y_{ref} while respecting flow constraints $0 \leq q_0 \leq 20 \text{ cm}^3\text{s}^{-1}$ and level bounds $0 \leq h_{1,2} \leq 31 \text{ cm}$. The control synthesis using the Multi-Parametric Toolbox begins with a definition of the prediction model:

```
>> A = [-0.0315, 0; 0.0315, -0.0315]; B = [0.0769; 0]; C = [1, 0]; D = 0;
>> tanks = ss(A, B, C, D);
>> Ts = 5;
>> model = mpt_sys(tanks, Ts);
```

Here, we have first defined the linear state-space model (5)–(6) as a state-space object using the Control Toolbox. Subsequently, the `ss` object was converted to MPT's native discrete-time format using the sampling time $T_s = 5$ seconds. Input, output, and state constraints can now be directly added to the `model` variable:

```
>> q0max = 20; q0min = 0; q0s = 16.9811;
>> h1max = 31; h1min = 0; h1s = 21.44;
>> h2max = 31; h2min = 0; h2s = 21.44;
>> model.umax = q0max - q0s;
>> model.umin = q0min - q0s;
>> model.xmax = [h1max - h1s; h2max - h2s];
>> model.xmin = [h1min - h1s; h2min - h2s];
>> model.ymax = h2max - h2s;
>> model.ymin = h2min - h2s;
```

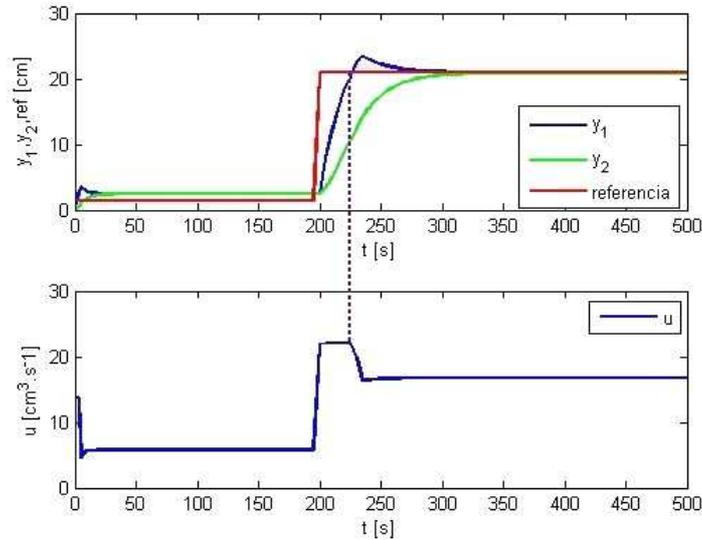



Figure 3: Simulation results.

are utilized by the MPC scheme. Specifically, notice that the controller starts to decrease the control effort even before the reference signal is reached around the time of 230 seconds.

The real-time implementation of the resulting explicit MPC feedback law can then be done by a simple single-click action, where the Simulink diagram is compiled and automatically downloaded to the dSPACE card for execution. The DSP processor located on the card then executes the look-up table (9) in real-time as a sequence of search operations. In Figure 4 we present the real data measurements obtained during a real-time experiment. Again, it can be nicely seen from Fig. 4 that the MPC controller utilizes the predictions to change the value of the input signal in advance, such that output is steered towards the time-varying reference. The reference tracking error is attributed to two reasons. First, the MPC design is based on a linear model of the plant, which is only accurate in a close neighborhood of the linearization point. Another reason being that a state observer is utilized to get estimates of the state vector.

5 Conclusions

In this paper we have presented how to design an MPC strategy to control a laboratory liquid tanks model. The MPC optimization problem is solved parametrically using the Multi-Parametric Toolbox. The solution, which takes a form of a look-up table, can be easily implemented on the dSPACE real-time control platform.

Acknowledgment

The authors are pleased to acknowledge the financial support of the Scientific Grant Agency of the Slovak Republic under the grants 1/3081/06, 1/4055/07, and by the ESF project JPD 3 2005/NP1-047 PhD students for Modern Industrial Automation in SR, code No. 13120200115. This work was supported by the Slovak Research and Development Agency under the contracts No. VV-0029-07 and No. LPP-0092-07.

References

- [1] A. Bemporad, M. Morari, V. Dua, and E.N. Pistikopoulos. The Explicit Linear Quadratic Regulator for Constrained Systems. *Automatica*, 38(1):3–20, January 2002.

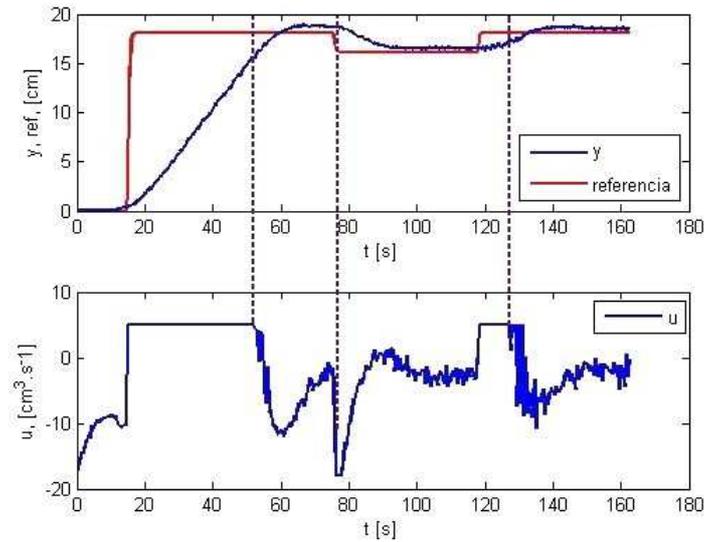


Figure 4: Real-time control results.

- [2] F. Borrelli. Constrained Optimal Control of Linear and Hybrid Systems. In *Lecture Notes in Control and Information Sciences*, volume 290. Springer, 2003.
- [3] M. Kvasnica, P. Grieder, M. Baotic, and M. Morari. Multi-Parametric Toolbox (MPT). In *Hybrid Systems: Computation and Control*, pages 448–462, March 2004. Available from <http://control.ee.ethz.ch/~mpt>.
- [4] J.M. Maciejowski. *Predictive Control with Constraints*. Prentice Hall, 2002.

Bc. Ivana Rauová

Institute of Information Engineering, Automation, and Mathematics
 Slovak University of Technology in Bratislava
 Radlinského 9, 812 37 Bratislava, Slovakia
 e-mail: iva.rauova@gmail.com

Ing. Michal Kvasnica, PhD.

Institute of Information Engineering, Automation, and Mathematics
 Slovak University of Technology in Bratislava
 Radlinského 9, 812 37 Bratislava, Slovakia
 e-mail: michal.kvasnica@stuba.sk

Ing. Ľuboš Čirka, PhD.

Institute of Information Engineering, Automation, and Mathematics
 Slovak University of Technology in Bratislava
 Radlinského 9, 812 37 Bratislava, Slovakia
 e-mail: lubos.cirka@stuba.sk

Prof. Dr. Ing. Miroslav Fikar

Institute of Information Engineering, Automation, and Mathematics
 Slovak University of Technology in Bratislava
 Radlinského 9, 812 37 Bratislava, Slovakia
 e-mail: miroslav.fikar@stuba.sk