Hardware Synthesis of Model Predictive Control on a Programmable Logic Controller

Michael Ralea [1]; Dr. Ambrose Adegbege [2]
[1]The College of New Jersey, [2]NJ Space Grant Consortium

Abstract

Programmable Logic Controllers (PLC) are a wide-spread tool used in the control of industrial processes. Although they are associated with simple computations and PID control, the PLC has been pushed to the limits in recent research studying the application of implicit model predictive control (MPC). The PLC is pushed further by attempting to implement explicit MPC on the quad-tank process: a complex dynamical system. Functionality of explicit MPC is verified with a simple SISO system before the quad-tank process is observed.

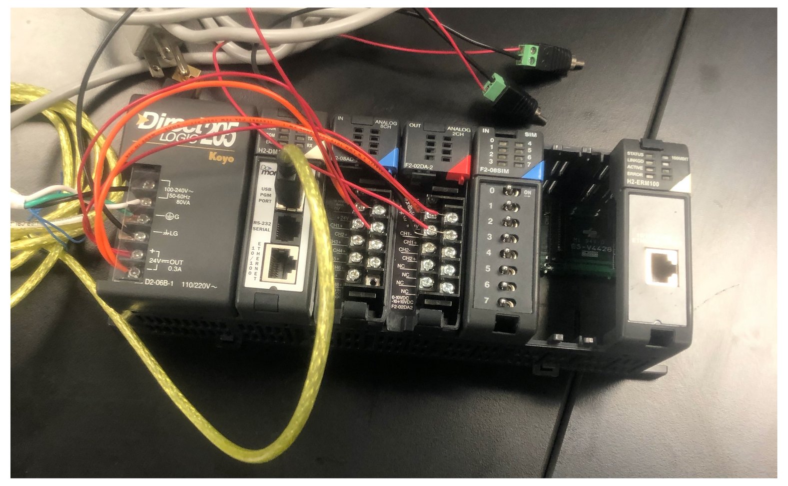


Figure 1. PLC Configuration

Introduction / Background

Model Predictive Control – An advanced method of process control built to handle complex dynamical systems. MPC predicts how dependent variables will change for future time steps, given changes in independent variables. It is an iterative method for finite-horizon optimization.

Explicit MPC - Rather than perform online computations as implicit MPC would, explicit uses offline computations to determine the optimal control output by a evaluating a linear function. This means the controller needs minimal run-time computations, increasing speed but also taking up significant storage.

Hardware Synthesis of Explicit MPC on a PLC – The PLC offers a robust interface to implement control schemes but has very limited storage. Given the popularity of the PLC in industry, the ability to implement explicit MPC might offer new control process capabilities that PID may not be suitable for.

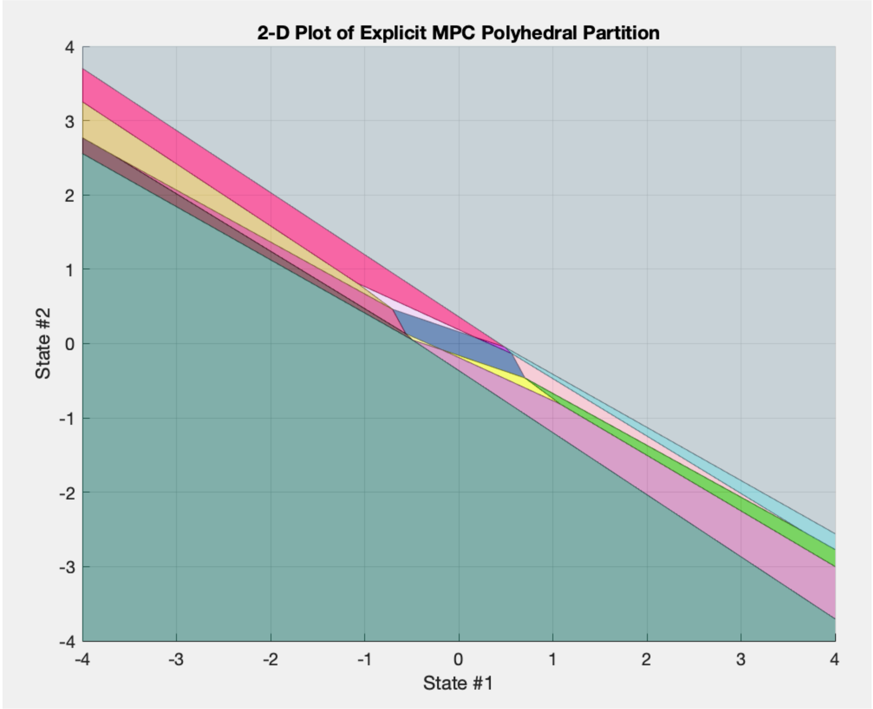


Figure 2. Polyhedral Partition of Double Integrator Model

Methodology

Offline		Online
System Modeling and Explicit MPC	Programmable Logic Controller	Search Algorithm
- State Space Representation ẋ = Ax + Bu y = Cx + Du - Use MATLAB MPC Toolbox to generate implicit object and convert to explicit solution - Simplify solution by combing regions that share common gains or who's union is a convex set	- Using Ladder Logic, create subroutines for each region to check if state falls within polyhedral partition - Test with simulation - Import values for linear function evaluation from explicit solution as a CSV into R memory - Generate State Observer (if needed)	- Read in states from system (or compute with state observer) - Check each region until the correct partition is found - Compute optimal gain(s)

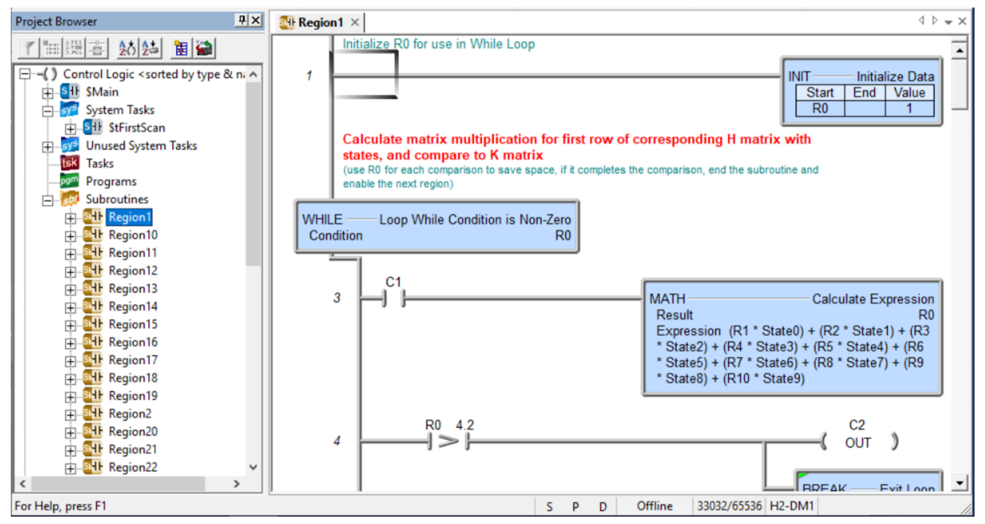


Figure 3. Ladder Logic Implementation + Methodology

Explicit MPC Verification

Using MATLAB's Model Predictive Control toolbox, the simple double integrator system is modeled and solved for. The solution yielded 15 polyhedral regions after combining regions who's union was a convex set. The PLC is then programmed to measure the states and search for the corresponding pre-computed control move by checking the inequality constraints for each region using Eq 1. The ladder logic is verified through exhaustive test-benching using the Do-More Designer's built in simulator (v 2.7.4).

$$H_i\, x(k) \le K_i$$
$$u(k) = F_i x(k) + G_i$$

$x(k)$ = state vector including state variables, output reference values and plant disturbance signals
$u(k)$ = corresponding control law

Eq 1. Explicit MPC Equations for finding region and computing gain

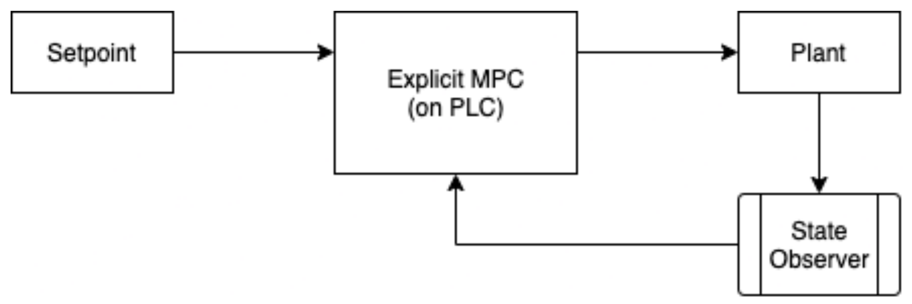


Figure 4. Control Scheme Model

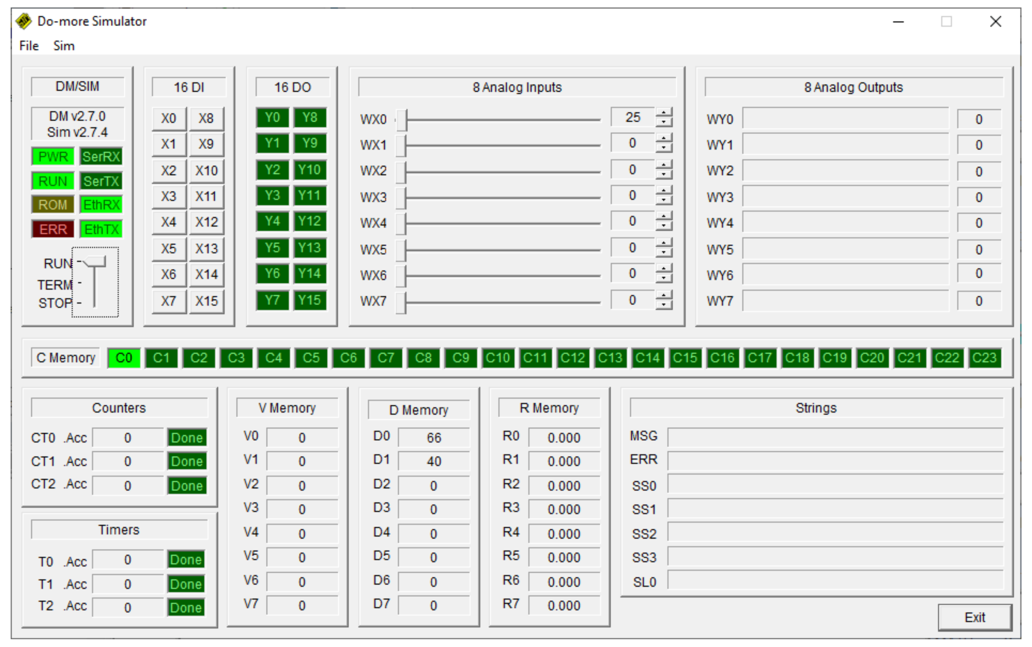
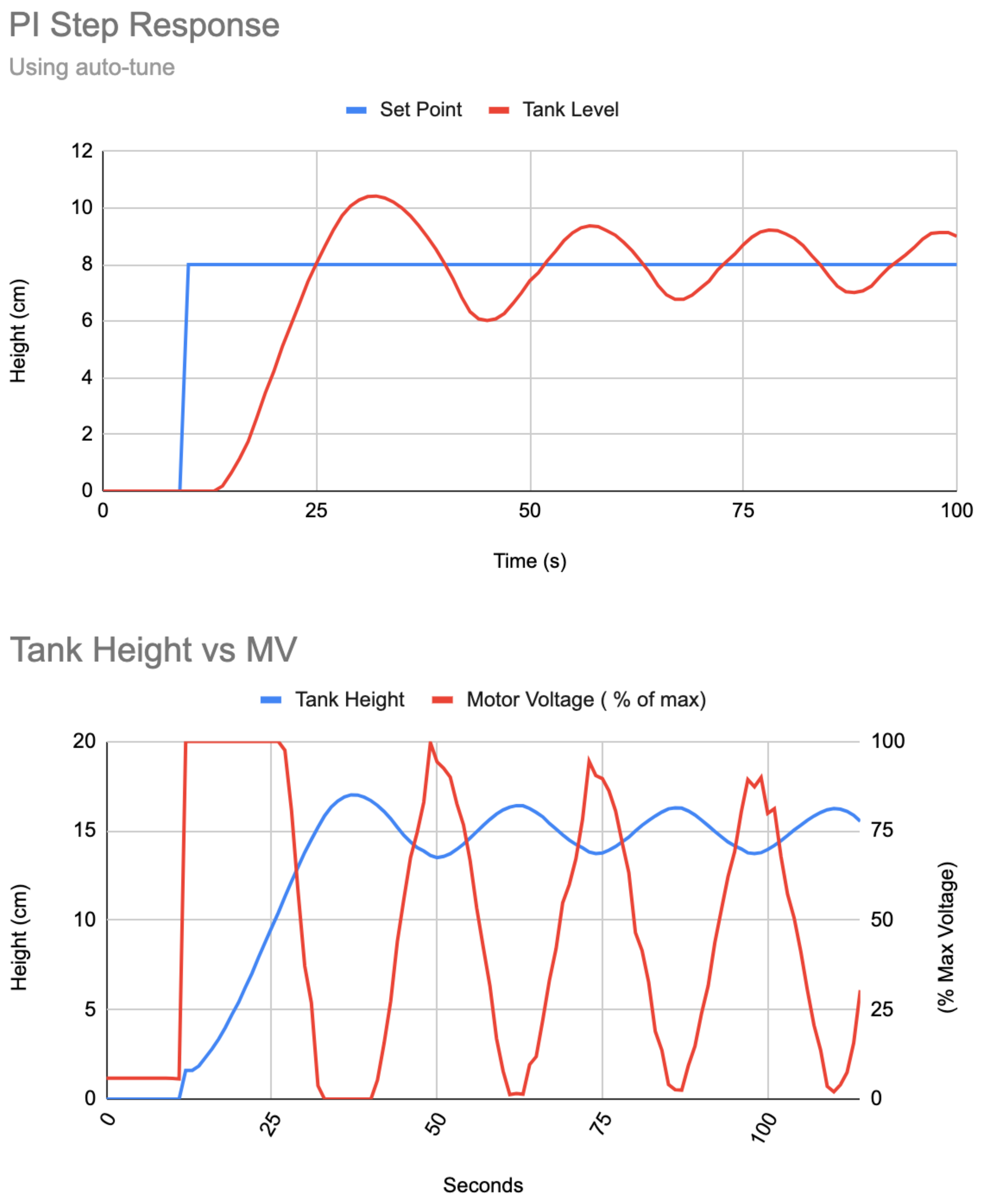


Figure 5. Do-More Simulator

PLC / Quad-Tank PI

Before attempting to solve the explicit control scheme for the quad-tank process, the ability to interface with the PLC was verified. Using a simpler coupled-tank system and the Do-More Designer's built in Auto-Tune feature, a PI controller is implemented to mediocre results. The step response of the controller gives a clear example of why simple PID may not be enough for processes with strict constraints. The water level is shown to fluctuate quite significantly and overshoot by a massive margin. Although the PI constants were likely not optimal, the response is still indicative of a need for a more complex control scheme.



Graph 1. PI Step Response

Figure 6. Quad-Tank System

Conclusions / Future Work

The implementation of explicit MPC on a PLC poses a laborious task for the programmer. Although the solution is relatively simple to obtain, the amount of coding increases rapidly as the complexity of the system increases. For the quad-tank, the explicit solution yields 28 regions, averaging 20 comparisons per region. This means there are 560 comparisons to be typed by hand, each with 10 operations within. The comparison portion of the code for the quad-tank system is completed but uses a 6 state representation of the model. Because there are only 4 measurable states of the system, a state observer needs to be implemented on the PLC. Ultimately, the explicit MPC will be tested against existing work on the quad-tank system and the results of other MUSE students.

References

1. A. Bemporad, F. Borrelli and M. Morari, "Robust model predictive control: Piecewise linear explicit solution," 2001 European Control Conference (ECC), Porto, 2001, pp. 939-944, doi: 10.23919/ECC.2001.7076033.
2. Richard M. Levenson, Zachary E. Nelson and Ambrose A. Adegbege, "Programmable Logic Controller for Embedded Implementation of Input-Constrained Systems"
3. T. A. Johansen, W. Jackson, R. Schreiber and P. Tondel, "Hardware Synthesis of Explicit Model Predictive Controllers," in IEEE Transactions on Control Systems Technology, vol. 15, no. 1, pp. 191-197, Jan. 2007, doi: 10.1109/TCST.2006.883206.

Contact

Michael Ralea
Email – raleam1@tcnj.edu
Cell – (862) 309 - 2818

Dr. Ambrose Adegbege
Email – adegbega@tcnj.edu