Generation of MPC-like Explicit Control Laws with Reinforcement Machine Learning

Patrik Valábek, Martin Klaučo

Institute of Information Engineering, Automation and Mathematics, Slovak University of Technology in Bratislava, Radlinského 9, 812 37 Bratislava, Slovakia, email: patrik@valabek.sk, martin.klauco@stuba.sk

Abstract

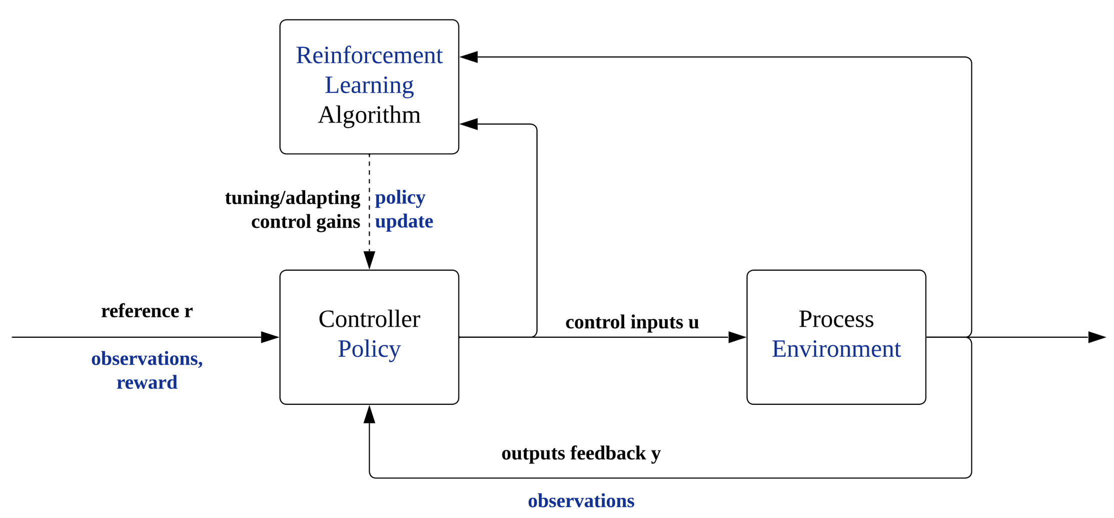
Reinforcement learning (RL) is a group of machine learning algorithms that do not require pre-collected data and learn solely from interactions with the process. We explore the application of RL techniques for generating MPC-like explicit control laws without knowing the model of the process. The proposed approach combines the principles of model predictive control design to derive control laws without prior knowledge of the process model. A series of simulations are performed using linear dynamical models to generate explicit control laws using RL. The performance of the RL-based approach is then compared to the one of the explicit model predictive control approach. With RL, we can stabilize and effectively control numerous processes regarding the process control input and output bounds.

Reinforcement Learning

RL is one of three basic machine learning paradigms alongside supervised learning and unsupervised learning. Compared to other two paradigms, RL does not require collected or labelled data. RL algorithm is often called Agent. Policy - controller is updated based on interaction with environment - process. The Agents goal is to learn a policy, which maximizes the cumulative value of objective function (reward)

$$\max -\sum_{k\in E} (y_k - r_k)^\intercal Q_\mathrm{y} (y_k - r_k) + \Delta u_k^\intercal Q_\mathrm{u} \Delta u_k,$$

where y_k is output from process, r_k is reference, Δu_k is difference of the controlled input, Q_y and$Q_{\text{-u}}$ are weighting matrices. To put RL in perspective with process control, we used block diagram below



our controller - policy, represented by neural network (NN) generates an control input to the environment - process. Inputs to the policy are called observables and contain every information about process available (outputs, reference, previous control inputs, etc.). RL algorithm uses evaluations of control inputs generated to maximize objective function and updates the policy until a desired level of control performance is achieved.

Training of Explicit Control Law

Training of explicit control law - policy - is done in simulation of the same length. At the beginning of each simulation the process is being reset with random initial condition and controlled to random reference or to steady states. Every time step in simulation the policy is updated by one step of gradient ascent calculated based on current and previously collected data from interactions with the process.

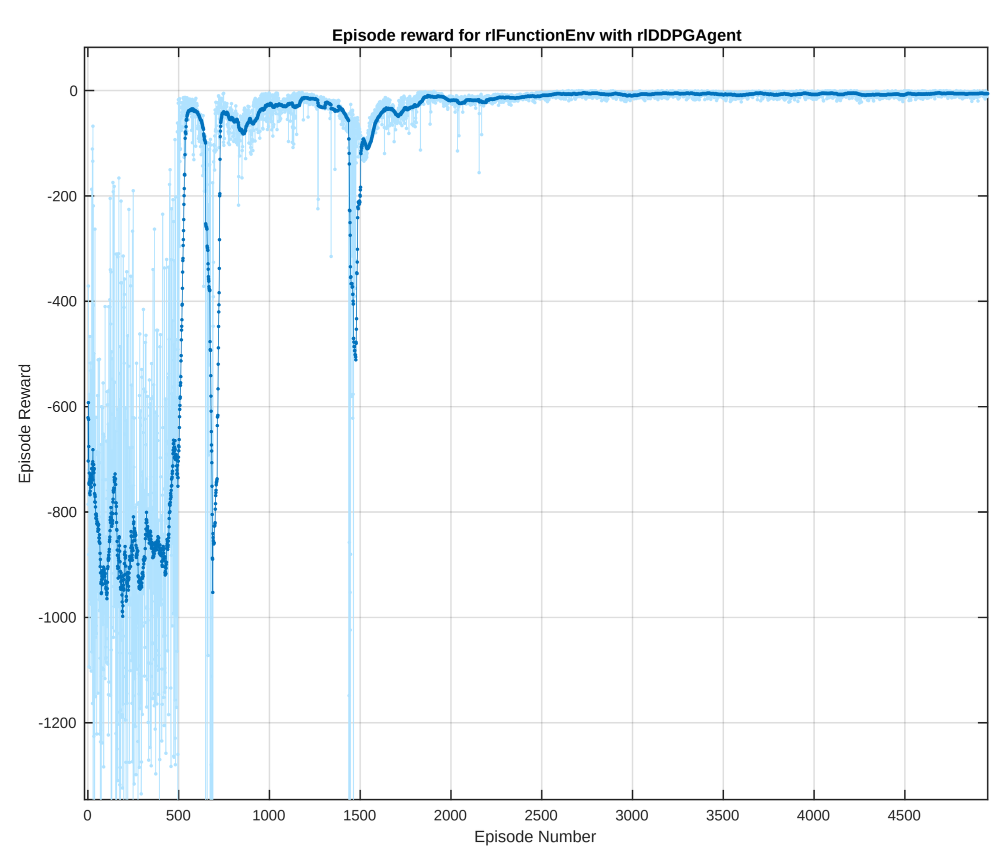


Figure 1:
Training of MIMO system. Light blue indicates cumulative value of objective function in one simulation and dark blue mean of last 40 simulations.

Results

After successful training, we obtained our explicit control law in the form of a NN. To test the performance of this law, we simply replace the controller in the closed loop with the trained policy. In the following figures, we can see comparisons of the trained control laws with the MPC control.

We demonstrated control on three processes: a double integrator (DI) (Figure 2), a MIMO process (Figure 3), and a stable SISO system of second order with oscillations (Figure 4). The explicit control law effectively stabilized and controlled each process from random initial conditions to desired states.

We implemented control input bounds by selecting appropriate activation functions. Also, for the SISO system of second order, we successfully implemented output bounds. By penalizing the crossing of soft bounds, the RL algorithm was able to train an effective explicit control law that did not cross the bounds.

All of this results were achieved without any information about model, just by the RL algorithm interacting with the process. The comparison of the Cumulative values of objective function can be seen in Table 1.

Table 1: Comparing objective function values in controlled systems.

Algorithm	J
$\mathrm{RL}_{\mathrm{DI}}$	$-3.092 \cdot 10^2$
$\mathrm{MPC}_{\mathrm{DI}}$	$-3.044 \cdot 10^2$
$\mathrm{RL}_{\mathrm{MIMO}}$	-4.341
$\mathrm{MPC}_{\mathrm{MIMO}}$	-4.110
$\mathrm{RL}_{\mathrm{SISO2}}$	$-5.604 \cdot 10^2$
$\mathrm{MPC}_{\mathrm{SISO2}}$	$-5.373 \cdot 10^2$

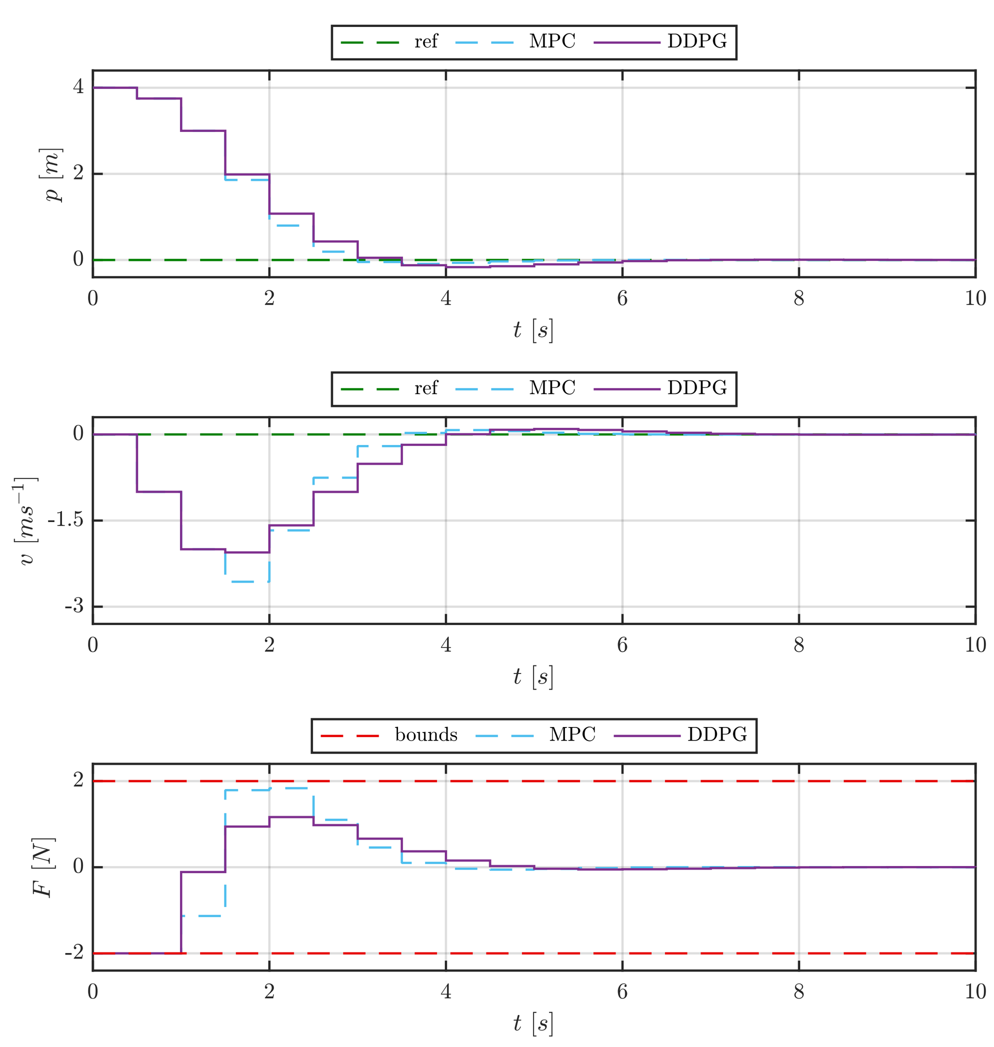


Figure 2: Control of the double integrator system. Upper two graphs show outputs, position p and velocity v. Bottom one shows control input, force applied F. Percentage error between MPC and RL is only 2.09%.

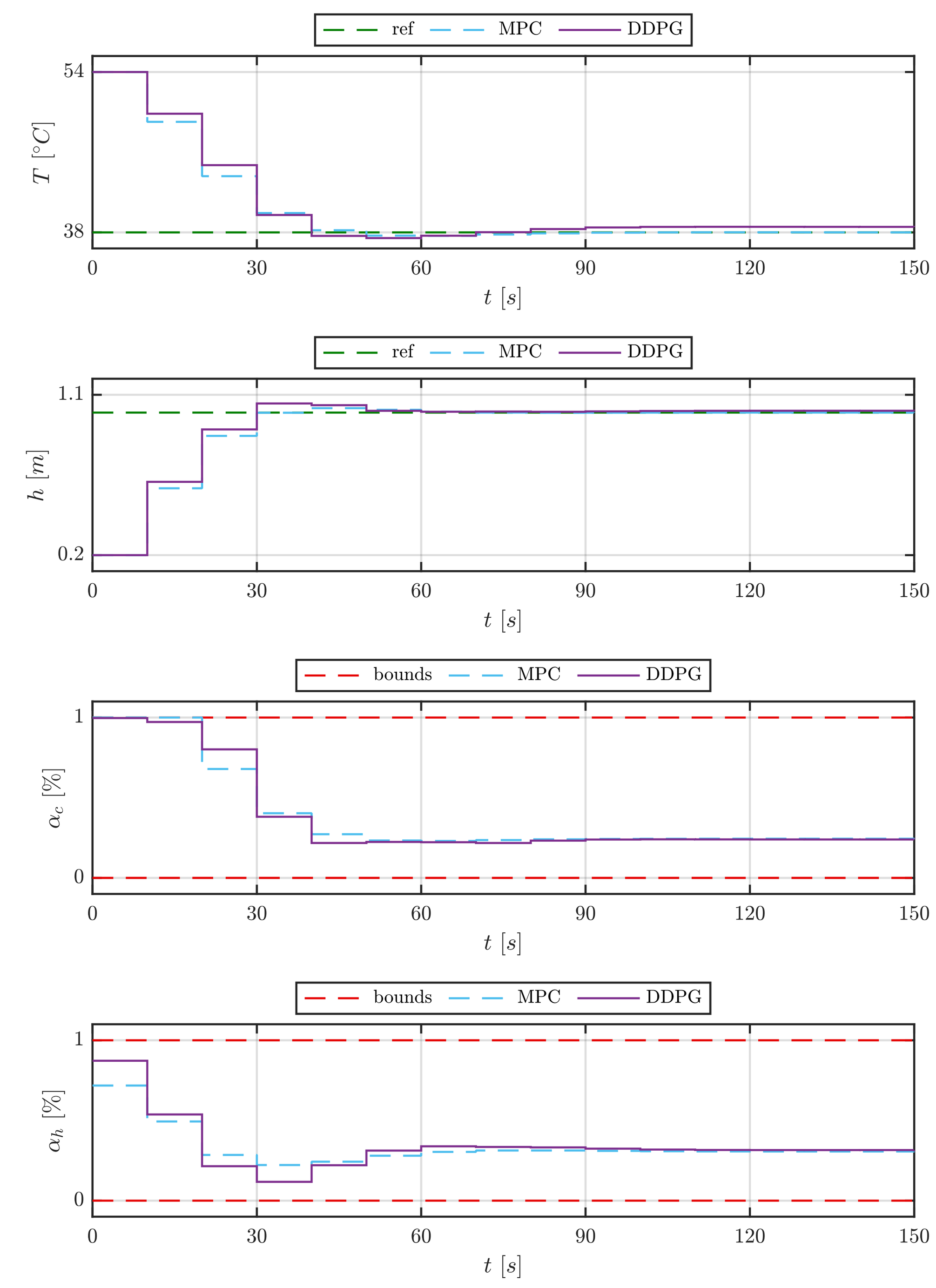


Figure 3: Control of the MIMO system with high interactions. Controlled system is a tank in which cold and hot inlet streams are mixed. Upper two graphs show outputs, temperature T and tank level h and bottom two shows control inputs, percentage openings for cold α_c and hot α_h inlet steams. Percentage error between MPC and RL is only 5.62%, but small steady state error could be observed.

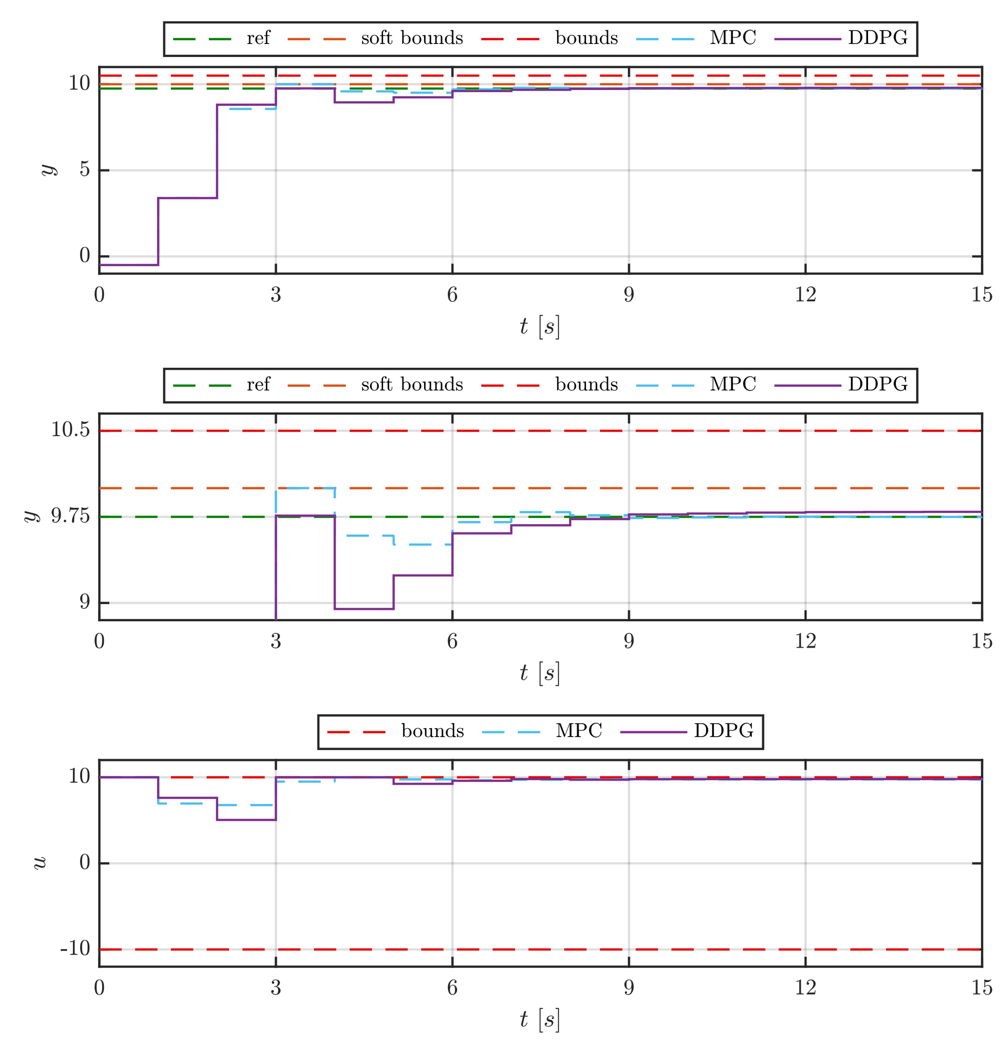


Figure 4: Control of the second order SISO system with oscillations. The upper two graphs show output y with the middle one zoomed to the area of bounds. The control input u is shown on the bottom graph. Percentage error between MPC and RL is only 4.30%, but small steady state error could be observed.

Acknowledgments

This research is funded by the European Commission under the grant no. 101079342 (Fostering Opportunities Towards Slovak Excellence in Advanced Control for Smart Industries). The authors gratefully acknowledge the contribution of the Scientific Grant Agency of the Slovak Republic under the grants 1/0545/20, 1/0490/23, and the Slovak Research and Development Agency under the project APVV-20-0261 and APVV-21-0019.