Simulating the Effectiveness of Physics-informed Machine Learning in Nonlinear Model Predictive Control for Robotic Manipulators

Asa Paparo

Abstract

Existing methods of robotic motion planning rely on numerical optimization to generate control actions within constraints. One of the most successful methods of control has been Model Predictive Control (MPC), which continuously replans a trajectory by solving an online optimization problem. This means that for high dimension dynamical systems, the computational cost of MPC can rapidly become prohibitive. Physics Informed Neural Networks (PINNs) allow for the data driven creation and solving of high dimensional differential equations. Existing research has made progress in combining them, but lacks rigor and code optimization. In this paper, I develop an efficient NMPC implementation that relies on a PINN to plan movements. Then, I set up and integrate the controller with a state of the art simulation to demonstrate its effectiveness for a 7 DoF robotic arm. This development presents a significant advancement for fields that employ mobile robotics, such as manufacturing and search/rescue.

1 Introduction

Robotic manipulation is an area of high importance as global manufacturing increasingly relies on automation to accomplish tasks that would typically be done by humans. Advances in robotics and machine learning promise to be integral in the fourth industrial revolution, improving efficiency and worker safety [9]. Specifically, the advancement and effectiveness of industrial robotic systems is central to a developed nation's prosperity and economy. Due to high domestic costs and restrictions, offshore manufacturing is utilized by companies seeking to minimize cost of production. To improve economic conditions without worsening worker conditions, the United States must rely on robotic manufacturing [7]. Additionally, search and rescue presents a major opportunity for robotics to improve by making it safer and more efficient [4]. For medicine, surgical robots can accomplish tasks with more precision and safety than their human counterparts [3].

A primary goal of the field of control theory is the efficient and constrained control of complex dynamical systems. A highly popular approach for this is Model Predictive Control, which relies on continuously replanning a trajectory over a specific horizon [6]. In situations where high processing power is available, such as on Boston Dyanmics' robots, this is a necessary tradeoff for the flexibility of nonlinear MPC [10]. However, despite significant development on improving this controller, nonlinear MPC still suffers from requiring "enourmous computational effort" [1]. To address this, several groups have successfully used physics-informed machine learning for optimization in MPC instead of traditional methods [11, 13]. A natural application of this is in control of high dimensionality robotic arms, which is made significantly more efficient by the neural network. These papers provide evidence in favor of PINN-based MPCs for controlling lower dimensional systems, so I hope to use a robotics code framework called Drake [8] to simulate the control of a high dimensional industrial arm.

2 Methods

2.1 Nonlinear Model Predictive Control (NMPC)

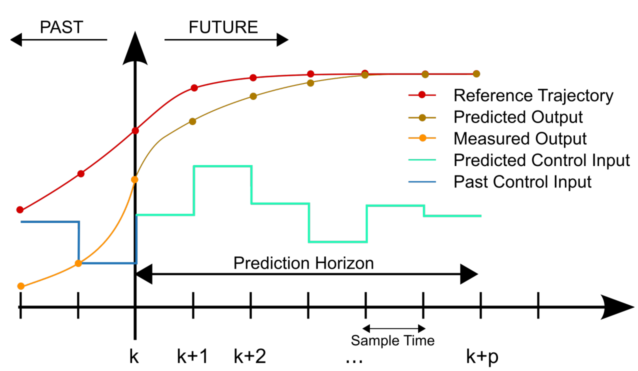

Figure 1: A diagram of model predictive control by Martin Behrendt [2]. It shows the planning of a new trajectory over time in comparison to the previously planned trajectory.

Model predictive control [5] and its nonlinear variation are an algorithm with several steps.

1. Create an optimized trajectory from current to target states over the specified horizon
2. Follow the first discrete state within the larger trajectory
3. Repeat with the new state

The trajectory calculated at each step is based on an optimal control law, which minimizes a cost function $J(x, u)$ [12].

$$J(x, u) = \int_0^\infty L(x(t), u(t), t)dt. \tag{1}$$

$$\begin{aligned} x^\star, u^\star &= \arg\min_{x,u} J(x, u) \\ \dot{x}(t) &= f(x(t), u(t)), \forall t \\ x(0) &= x_0 \end{aligned} \tag{2}$$

This would be solved by a PINN trained to the dynamics of the system, however, I was unable to successfully implement this on a deadline. For the remainder of this project, I use an "InverseDynamicsController," which uses the system's dynamics to convert PID control output into applicable voltages.

2.2 Simulation

To create an effective simulation for the system, I use Drake. The specific system I control is a 7 degrees of freedom KUKA iiwa arm, with documentation from MIT Manipulation [14]. Since I could not immediately get a PINN working

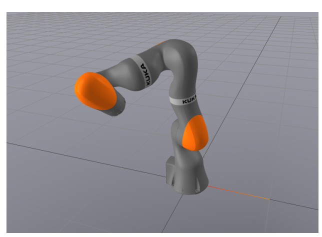

Figure 2: A visualization of the simulation, using Meshcat and the iiwa14 urdf file.

```
meshcat = StartMeshcat()
builder = DiagramBuilder()

# Adds both MultibodyPlant and the SceneGraph, and wires them together.
plant, scene_graph = AddMultibodyPlantSceneGraph(builder, time_step=1e-4)
# Note that we parse into both the plant and the scene_graph here.
iiwa_model = Parser(plant, scene_graph).AddModelsFromUrl("urdf url here")[0]
plant.WeldFrames(plant.world_frame(), plant.GetFrameByName("iiwa_link_0"))
```

```
plant.Finalize()

# Set up visualizer
visualizer = MeshcatVisualizer.AddToBuilder(builder, scene_graph, meshcat)

# PID controller
n = plant.num_positions()
kp = [100] * n
ki = [1] * n
kd = [20] * n
iiwa_controller = builder.AddSystem(
    InverseDynamicsController(plant, kp, ki, kd, False)
)
iiwa_controller.set_name("iiwa_controller")

# Wire simulation
builder.Connect(
    plant.get_state_output_port(iiwa_model),
    iiwa_controller.get_input_port_estimated_state()
)
builder.Connect(
    iiwa_controller.get_output_port_control(),
    plant.get_actuation_input_port()
)
diagram = builder.Build()
diagram.set_name("iiwa diagram")

context = diagram.CreateDefaultContext()
plant_context = plant.GetMyMutableContextFromRoot(context)

# Arbitrary starting config
q0 = np.array([-1.5, 0.1, 0, -1.2, 0, 1.6, 2])
w0 = 2 * q0
x0 = np.hstack((q0, w0))
plant.SetPositions(plant_context, q0)
iiwa_controller.GetInputPort("desired_state").FixValue(
    iiwa_controller.GetMyMutableContextFromRoot(context), x0
)

# Simulator
simulator = Simulator(diagram, context)
simulator.set_target_realtime_rate(1.0)

meshcat.StartRecording()
simulator.AdvanceTo(10.0)
meshcat.StopRecording()
meshcat.PublishRecording()
```

This code sets up a simulation, with a structure shown in figure 3. Notably,

3 Results

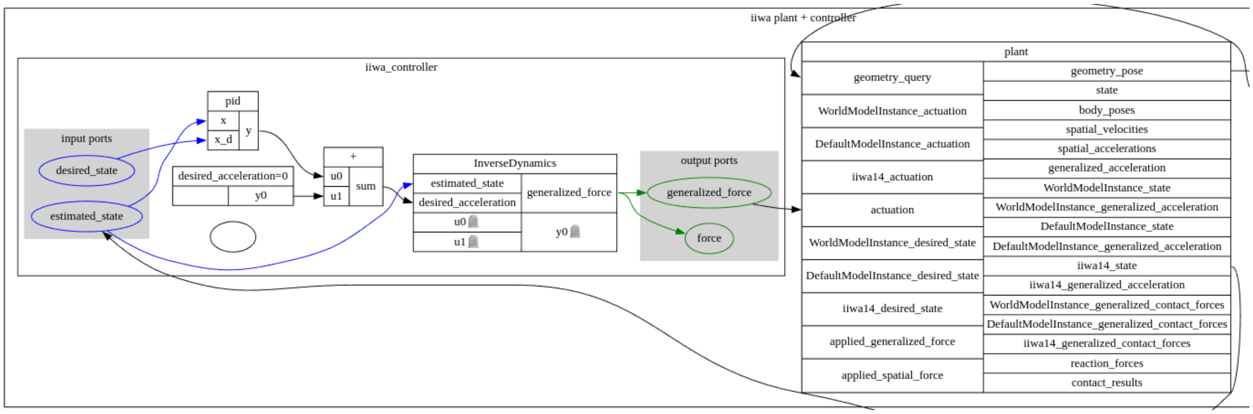

Figure 3: A flow input/output diagram of a PID controller connected to the iiwa system, generated by Drake.

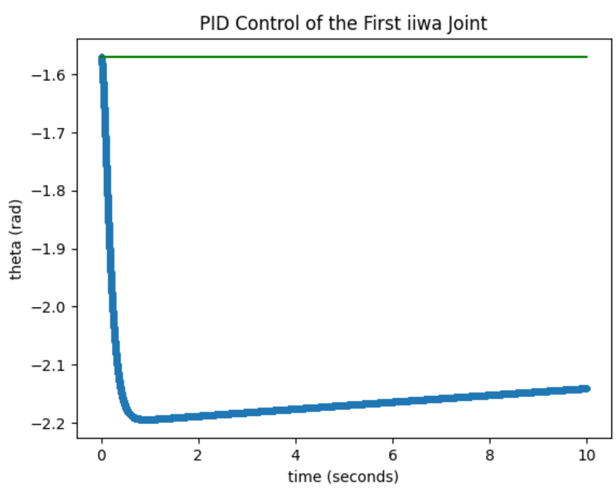

Figure 4: A graph of the base joint of the iiwa arm under inversion PID control. As you can see, because the arm is fully actuated, the position very quickly approaches its setpoint.

For this project, while I was able to make progress with the simulation and nonlinear model predictive controller, I was unable to successfully finish a python implementation of the neural network on time. This limited my ability to collect data on every component working as a whole. However, as shown in figure 3, I was successful in creating the closed loop control simulation based on the loaded arm dynamics. Overall, the tools I employed for the simulation of this robot were very effective, and will likely see significantly more prevalence going forward. Lastly, as shown in 4, I demonstrated the effectiveness of seperate PID control for individual components of the arm.

4 Discussion

My results are rather inconclusive regarding the effectiveness of PINN-based NMPC, however they convey the utility of simulation in the robotics field. It is important to note, that for these controllers to reach widespread adoption, a well documented and high performance implementation will have to exist, considering the technical barrier to implementing it from scratch.

References

[1] N. Blet et al. "NONLINEAR MPC VERSUS MPC USING ON-LINE LINEARISATION — A COMPARATIVE STUDY". In: *IFAC Proceedings Volumes* 35.1 (2002). 15th IFAC World Congress, pp. 147–152. ISSN: 1474-6670. DOI: https://doi.org/10.3182/20020721-6-ES-1901.00593. URL: https://www.sciencedirect.com/science/article/pii/S1474667015390145.

[2] Martin Behrendt. *A basic working principle of Model Predictive Control*. 2009.

[3] G. P. Moustris et al. "Evolution of autonomous and semi-autonomous robotic surgical systems: a review of the literature". In: *The International Journal of Medical Robotics and Computer Assisted Surgery* 7.4 (2011), pp. 375–392. DOI: https://doi.org/10.1002/rcs.408. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/rcs.408. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/rcs.408.

[4] Yugang Liu and Goldie Nejat. *Robotic Urban Search and Rescue: A Survey from the Control Perspective*. 2013.

[5] Yuval Tassa, Nicolas Mansard, and Emo Todorov. "Control-limited differential dynamic programming". In: *2014 IEEE International Conference on Robotics and Automation (ICRA)*. 2014, pp. 1168–1175. DOI: 10.1109/ICRA.2014.6907001.

[6] Michael G. Forbes et al. "Model Predictive Control in Industry: Challenges and Opportunities". In: *IFAC-PapersOnLine* 48.8 (2015). 9th IFAC Symposium on Advanced Control of Chemical Processes ADCHEM 2015, pp. 531–538. ISSN: 2405-8963. DOI: https://doi.org/10.1016/j.ifacol.2015.09.022. URL: https://www.sciencedirect.com/science/article/pii/S2405896315011039.

[7] Sridhar Kota and Thomas C. Mahoney. *Manufacturing Prosperity: A Bold Strategy for National Wealth and Security*. Tech. rep. MForesight: Alliance for Manufacturing Foresight, 2018. URL: https://deepblue.lib.umich.edu/handle/2027.42/145156.

[8] Russ Tedrake and the Drake Development Team. *Drake: Model-based design and verification for robotics*. 2019. URL: https://drake.mit.edu.

[9] Mohd Javaid et al. "Substantial capabilities of robotics in enhancing industry 4.0 implementation". In: *Cognitive Robotics* 1 (2021), pp. 58–75. ISSN: 2667-2413. DOI: https://doi.org/10.1016/j.cogr.2021.06.001. URL: https://www.sciencedirect.com/science/article/pii/S2667241321000057.

[10] Boston Dynamics. *Picking Up Momentum*. 2022.

[11] Jonas Nicodemus et al. "Physics-informed Neural Networks-based Model Predictive Control for Multi-link Manipulators". In: *IFAC-PapersOnLine* 55.20 (2022). 10th Vienna International Conference on Mathematical Modelling MATHMOD 2022, pp. 331–336. ISSN: 2405-8963. DOI: https://doi.org/10.1016/j.ifacol.2022.09.117. URL: https://www.sciencedirect.com/science/article/pii/S2405896322013118.

[12] Kris Hauser. *Robotic Systems*. University of Illinois at Urbana-Champaign, 2023.

[13] Jingyue Liu, Pablo Borja, and Cosimo Della Santina. *Physics-informed Neural Networks to Model and Control Robots: a Theoretical and Experimental Investigation*. 2023. arXiv: 2305.05375 [cs.RO].

[14] Russ Tedrake. *Robotic Manipulation. Perception, Planning, and Control*. 2023. URL: http://manipulation.mit.edu.