Applications of Physics-informed Machine Learning for Automatic Adaptation to Varying Loads in Robotic Manipulators

Asa Paparo

October 19, 2023

Introduction

- us government priorities: - semi automated production - make national production economically feasible - well educated laborors, robots work with them

- developed a new framework that is applicable

- antibiotics production - very important production - microchips, taiwan, etc - focus on relevant industries

- cutting edge of robotics surgery is nano robotic surgery - nanorobots swarm etc - in order to exploit specifics, it would be highly beneficial to modify architecture in X ways - claim adaptive ml algorithms are needed - adaptive algorithms for smart manufacturing and adaptive ml

Current methods of robotic manipulation rely on predefined systems and control algorithms to function effectively. [9] This is a effective in most scenarios. However, it requires retuning for any change to the system. In consumer or high risk situations, it may be impractical to retune systems periodically to account for changes to the physical hardware. In this paper, I propose an alternative paradigm using physics-informed machine learning to correct for changes to the system on the fly.

The goal of this experiment is to examine the effectiveness ¡develop a new generation¿ of PINN (Physics Informed Neural Networks) in adapting the model of a system on the fly. This would enable the dynamic updating of control algorithms, given little prior knowledge about external conditions. For example, a manipulator should be able to grasp any arbitrary load and continue to plan trajectories effectively. In order to test this, I derive the governing equations of motion for a simple 2-DOF arm system and use drake [11] to run discrete simulations of MPC and trajectory planning for this arm. To verify my hypothesis, I modify the load attatched to the manipulator midway through the simulation and record the new effectiveness of controlling it.

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 machine learning promise to be as impactful as an industrial revolution, with potential to deliver widespread benefits to civilization.[1] Specifically, advances in robotic automation for manufacturing and medicine have potential to reshape their industries for the better. In the case of manufacturing, this allows for significantly greater productivity and worker safety. For medicine, surgical robots can accomplish tasks with more precision and safety than their human counterparts. [3, 4, 5, 6] Automation in manufacturing is also critical for the prosperity and improvement of our modern society. For example, relevance in the global economy is viewed as core to American national security, and automation is an essential part in that [7]. Such importance is further exacerbated by declining birth rates in industrialized nations, which means a smaller workforce to support a growing aging population [8]. Because of this, advances in robotic automation are essential in overcoming global challenges and improving the standard of living around the world.

Abstract

Current methods of robotic manipulation rely on predefined systems and control algorithms to function effectively. [9] This is a effective in most scenarios. However, it requires retuning for any change to the system. In consumer or high risk situations, it may be impractical to retune systems periodically to account for changes to the physical hardware. In this paper, I propose an alternative paradigm using physics-informed machine learning to correct for changes to the system on the fly.

The goal of this experiment is to examine the effectiveness of PINN (Physics Informed Neural Networks) in updating the model of a system on the fly. This would enable the dynamic updating of control algorithms, given little prior knowledge about external conditions. For example, a manipulator should be able to grasp any arbitrary load and continue to plan trajectories effectively. In order to test this, I derive the governing equations of motion for a simple 2-DOF arm system and use drake [11] to run discrete simulations of MPC and trajectory planning for this arm. To verify my hypothesis, I modify the load attatched to the manipulator midway through the simulation and record the new effectiveness of controlling it.

analytical dynamics

Methodology

Derivation

To derive the dynamics for a double jointed arm system, I follow the derivation for an acrobot in Underactuated [9]. This is a fairly simple derivation using the euler-lagrange equation. - talk about what was done in methodologies before me -

$$\begin{bmatrix} l_{c1}s_1 \\ -l_{c1}c_1 \end{bmatrix}, \quad \begin{bmatrix} l_1 s_1 + l_{c2}s_{1+2} \\ -l_1 c_1 - l_{c2}c_{1+2} \end{bmatrix}, \tag{1}$$

$$T = T_1 + T_2, \quad T_1 = \frac{1}{2} I_1 \dot{q}_1^2 \tag{2}$$

$$T_2 = \frac{1}{2}(m_2 l_1^2 + I_2 + 2 m_2 l_1 l_{c2} c_2)\dot{q}_1^2 + \frac{1}{2} I_2 \dot{q}_2^2 + (I_2 + m_2 l_1 l_{c2} c_2)\dot{q}_1 \dot{q}_2 \tag{3}$$

$$U = -m_1 g l_{c1} c_1 - m_2 g (l_1 c_1 + l_{c2} c_{1+2}) \tag{4}$$

$$(I_1 + I_2 + m_2 l_1^2 + 2 m_2 l_1 l_{c2} c_2)\ddot{q}_1 + (I_2 + m_2 l_1 l_{c2} c_2)\ddot{q}_2 - 2 m_2 l_1 l_{c2} s_2 \dot{q}_1 \dot{q}_2 \tag{5}$$

$$-m_2 l_1 l_{c2} s_2 \dot{q}_2^2 + m_1 g l_{c1} s_1 + m_2 g (l_1 s_1 + l_{c2} s_{1+2}) = 0 \tag{6}$$

$$(I_2 + m_2 l_1 l_{c2} c_2)\ddot{q}_1 + I_2 \ddot{q}_2 + m_2 l_1 l_{c2} s_2 \dot{q}_1^2 + m_2 g l_{c2} s_{1+2} = \tau \tag{7}$$

Control

To control this system, I use Model Predictive Control.

Optimization

```
    builder = DiagramBuilder()

    pendulum = builder.AddNamedSystem("pendulum", PendulumPlant())

    controller = builder.AddNamedSystem("controller",
                                        PidController(kp=[10.], ki=[1.], kd=[1.]))

    builder.Connect(pendulum.get_state_output_port(),
                    controller.get_input_port_estimated_state())
    builder.Connect(controller.get_output_port_control(), pendulum.get_input_port())

    builder.ExportInput(controller.get_input_port_desired_state())
    builder.ExportOutput(pendulum.get_state_output_port())

    logger = LogVectorOutput(pendulum.get_state_output_port(), builder)
    logger.set_name("logger")

    diagram = builder.Build()
    diagram.set_name("diagram")

    display(SVG(pydot.graph_from_dot_data(
        diagram.GetGraphvizString(max_depth=2))[0].create_svg()))

```

Physics Informed Neural Network (PINN)

Conclusions

References

[1] https://arxiv.org/pdf/2011.03044.pdf

[2] Roach, N., Venkadesan, M., Rainbow, M. et al. Elastic energy storage in the shoulder and the evolution of high-speed throwing in Homo . Nature 498, 483–486 (2013). https://doi.org/10.1038/nature12267

[3] https://nvlpubs.nist.gov/nistpubs/eab/nist.eab.1.pdf

[4] https://www.tandfonline.com/doi/abs/10.1080/10408363.2018.1561640

[5] https://onlinelibrary.wiley.com/doi/abs/10.1002/rcs.408

[6] https://www.mckinsey.de/ /media/McKinsey/Business

[7] https://escholarship.org/content/qt0tw0h6kg/qt0tw0h6kg.pdf

[8] https://www.tandfonline.com/doi/full/10.1080/09513590701718364

[9] cite underactuated

[10] https://towardsdatascience.com/physics-informed-neural-networks-pinns-an-intuitive-guide-fff138069563

[11] cite drake