GAUSSIAN PROCESSES/ROBOT ARM DYNAMICS INTRODUCTION AND METHODS

Jacob Lister
Department of Electrical and Computer Engineerning
Department of Mathematical Sciences
Purdue University Fort Wayne
Fort Wayne, IN 46805, USA
aldrjt01@pfw.edu

ABSTRACT

In this work, I analyze the inverse dynamics model of a SARCOS seven joint robotic arm. The technique that will be used to perform this analysis was Guassian Processes Regression, as the functions that dictate the movement of robotic arms are complex and almost always extremely non-linear. In addition, the Projected Processes approximation will be used due to the large size of the dataset.

1 INTRODUCTION

In this section, I provide an overview of the inverse dynamics problem, a brief description of kinematics, and a brief explanation of the analysis that will be performed.

Inverse dynamics is used when working to model joint movement, often of a robotic arm or the arms and legs of an animal or person. This technique is used to determine the torque being applied to the joints in a system based on the position, velocity, and accelation of a body (or end effector in the case of a robotic arm). Using inverse dynamics, you can compute the torques needed to follow a certain trajectory, which provides the ability to simulate biological models or allows robotic arms to "plan" the path they will take when moving from one position to another.

The specific system I am looking at is that of a SARCOS robotic arm. This arm has 7 joints, which means 7 torques to find. Additionally, this means that there are 7 positions, velocities, and accelerations as outputs. Because we are working with inverse dynamics, the positions, velocities, and accelerations will be used as inputs, and the torques will be used as outputs. This means I will be creating a mapping of 21 inputs of 7 outputs.

Inverse dynamics can be performed in a variety of ways. The most straight-forward way is to analytically compute the torque functions of each joint in the system. This method requires full knowledge of the system and often becomes extremely complicated for many-jointed systems, so this method cannot always be used. There are also methods of estimating the torques using statistical methods. Due to the fact that the composition of torque functions is of trigonometric functions mainly, these functions end up being highly non-linear. To see more information about how the torque functions are composed, see Appendix A (will be implemented in final report). Due to the non-linearity of the torques, simple statistic methods such as linear regression are expected to perform poorly. Guassian Processes Regression is a more robust statistical method and will be the focus of this manuscript. Guassian Processes Regression will be used to model the torques of the SARCOS arm given the 21 inputs.

I chose to work on this project with the interest of learning about a method through the inverse dynamics of a robotic arm can be modeled non-analytically. As I have taken an undergraduate robotics course, and am currently taking a graduate level robotics course, I have worked with forward kinematics and a small amount of inverse dynamics. The inverse dynamics I have worked with has been solely analytic, working out each torque value in a system by hand. I have been curious about learning about non-analytic methods for a while now, and this was the perfect oppurtunity to do so.

The remaining part of this manuscript is organized as follows. Section 2 is dedicated to the methods that will be used.

2 METHODS

In this section, I describe the data source, Guassian Processes Regression (GPR), and Projected Processes (PP) Approximation.

2.1 DATA SOURCE AND SOFTWARE

The SARCOS dataset was sourced from the website for the textbook *Gaussian Processes for Machine Learning* Rasmussen & Williams (2006b). The analysis will be performed using the software Python 3.10 and the packages `numpy` and `scipy`. The dataset is publicly available and the code will be publicly available in a Github Repository (Lister).

2.2 GUASSIAN PROCESSES REGRESSION

To estimate the torque functions of the SARCOS arm, I will use a Guassian Processes Regression (GPR). GPR is a technique used in machine learning and more broadly in statistics. It is often used a supervised learning learning technique when working with non-linear systems. Because Guassian Processes (GPs) can interpreted as a distribution over a function space (Rasmussen & Williams, 2006a), we can use them to create a more robust method of regression. There are some additional advantages to working with guassian processes (scikit-learn), some of which are detailed below:

- The prediction is probabilistic (Gaussian) so that one can compute empirical confidence intervals and decide based on those if one should refit (online fitting, adaptive fitting) the prediction in some region of interest.
- Versatile: different kernels can be specified. Common kernels are provided, but it is also possible to specify custom kernels.

This versatility helps GPR work well in non-linear systems. There are some downsides to GPR, one being the fact that it uses all samples provided to perform its prediction, so the computational complexity becomes difficult to manage with large datasets; additionally, GPR loses efficiency in higher dimensional spaces (scikit-learn), but this should not be an issue as I am working with a dataset with a relatively small amount of dimensions. Because of the fact that the dataset I am working with is large, having over 40,000 samples, an approximation method for GPR will be used, which is detailed in the next subsection.

2.3 PROJECTED PROCESSES APPROXIMATION

The approximation method for GPR I will be using is the Projected Processes (PP) Approximation. This approximation method only uses a subset of the dataset to predict the function of the system by ”absorbing” the information of the entire dataset into the subset (Rasmussen & Williams, 2006a).

This method was chosen over the other approximation methods provided in the book for a number of reasons. This method was chosen over the Subset of Regressors Method due to PP’s higher performance and the Bayesian Committee Machine Method was not chosen due to it’s higher computational complexity for similar performance. The last method that was not chosen was the Subset of Dataset method, and while it does have the same computational complexity and similar performance as PP (Rasmussen & Williams, 2006a), it is based on a degenerate GP, so to maximize performance in predicting the torques, I chose PP over it. Overall, the Projected Processes Approximation provided a strong middle ground of the different approximation methods, leading to it being chosen.

REFERENCES

Jacob Lister. URL https://github.com/jacobtlister/stat512-project.

Carl Rasmussen and Christopher Williams. *Guassian Processes for Machine Learning*. MIT Press, 2006a. URL https://gaussianprocess.org/gpml/.

Carl Rasmussen and Christopher Williams. Guassian processes for machine learning - data, 2006b. URL https://gaussianprocess.org/gpml/data/.

scikit-learn. 1.7. guassian processes. URL https://scikit-learn.org/stable/modules/gaussian_process.html.