Final Year Project Proposal:

A Data-Driven Self-Awareness Model for Robotics Systems

Jeremiah M. Via

Supervised by Nick Hawes

7 October 2011

Abstract

Fault-detection in robotics systems is a difficult task. They are continually becoming more complex and thus subtle errors are hard to diagnose. Traditional approaches have relied on explicitly modeling component faults in code, but this technique does not scale to complex robots operating in dynamic environments. A new technique involves making the robot self-aware to the internal state of its various components. The aim of this project is to measure the efficacy of a probabilistic self-awareness model for robotics systems on two robotics middlewares, ROS and CAST, and if time allows, extend the model using a neural network, and then compare the two approaches. During this project, I will be collaborating with the creator of the original approach.

Description

This is a project with a heavy research bias. It involves implementing software to connect the CAST and ROS robotics middlewares to state-of-the-art software for data-driven fault detection and then measuring how well this fault-detection system performs on these middlewares. It also involves extending this technique with a different machine learning approach, neural networks. My hypothesis is that neural networks should generalize better than the current approach.

This software has multiple uses. It can be used during development as a debugging tool for the developer. They can see if their software is causing the robot to enter faulty states caused by any number of things. More interestingly, this information can be fed back to the robot's planning system. This is what truly creates that idea of self-awareness. A robot which can determine if its components are not running correctly can take action to overcome these faults, and is thus much more robust. This is of critical importance when robots are working around humans.

User Specification

This software is intended for robotics researchers and possibly those developing industrial robots. It will require a minimal understanding of machine learning to successfully incorporate into a robotics system. As such, this software will target expert computer users at the cutting-edge of robotics. I hope to make the software as intuitive as possible, but given that this is a machine learning system, it might be difficult.

Functional Specification

In order to measure the efficacy of the probabilistic self-awareness model, a bridge will have to be written to connect the two middlewares to the system which implements the probabilistic model. This is absolutely essential. Additionally, I will have to create a bridge to connect the system I write.

If time allows, I hope to create a neural network which can perform the task. It does not matter if it performs well, but I would like to know that is it possible. With enough time it would then be possible to create an optimized neural network which performed well at the task. That being said, I truly do hope I can implement a well-functioning neural network as I think it would be very interesting to see.

Non-Functional Specification

In order to complete this project, access to robots will be necessary to verify results gathered in simulation. After data has been gathered in simulation, a few data points can be collected on real robots, although this is not so important since it is the messages passing through the system which are of interest. Other than that, no additional materials need to be provided by the school. All the software required is freely available.

Approach

I am taking 70 credits in the first term as well as applying for PhD positions in the US, so my output will be less than the second term. My goals for the first term are to have implemented the connection to both middlewares and to have collect enough data to analyze the efficacy of the probabilistic model. Also, since I am taking the neural computation module, I will use this term to get a strong enough background in neural networks to be able to implement them.

Second term will be markedly lighter. I will use this time to implement the neural network for the data-driven technique. Because it can be difficult to create neural networks, I will start with the simplest kind appropriate for the task. As I get results, I will try to make improvements on the neural network architecture. This approach means I will always have a working system which can classify the system state. As I get working neural networks, I will be able to collect results to determine their efficacy.

Additionally, towards the end of second term, I will create a presentation. I will have to determine what is the best approach for this. Fault-detection is inherently not very visual, so I may present graphs of the system state over time and run a quick example of a robot having a fault in simulation. This is idea will become more refined as the presentation approaches.

During the Spring holiday, I plan to finish the project. I will write my report from the detailed notes I plan to take. This technique worked very well for team Java and it allowed me to create a nice report.