Simple Hands Project : Learning to Regrasp

RISS : Yudha Prawira Pane

Advisor : Prof. Matthew T. Mason , Alberto Rodriguez , Robert Paolini

ABSTRACT

In order to perform successful manipulation task, a robotic arm must know the current state of the object that it will manipulate and whether the object is in stable pose for that particular action.

Our system has proven a capability of doing grasping action quite well. But when it comes to execute a post-grasp manipulation , there are some uncertainties in the object's pose that must be able to accomodate.

One solution to reduce the amount of uncertainties hence improving the confidence of the robot about its state is by doing regrasp action. Similar to human , after grasping an object we can do some actions (e.g. flipping , rotating , etc) to reach more stable pose. The problem now is choosing the 'right' action that would eventually lead to more stable and predictable object's pose.

This poster describes the method that we use to quantify the system delay and finally the machine learning techniques to learn the likelihood of choosing the best regrasp action.

CONT ACT

Yudha Prawira Pane
Robotics Institute , CMU
Email: yudhapane@yahoo.com
Website: learningbot.wordpress.com

SYSTEM ARCHITECTURE

This project is implemented with an ABB manipulator and ROS (Robot Operating System) as the online software architecture. For the offline learning process, we use the image from an IEEE 1394 camera as the ground truth and calculated in MATLAB. Finally we designed and manufactured our own custom gripper with three fingers compliantly coupled to a single actuator. The object that the robot is trying to manipulate is a marker.

Figure 2. (right) Close- up of the robot hand

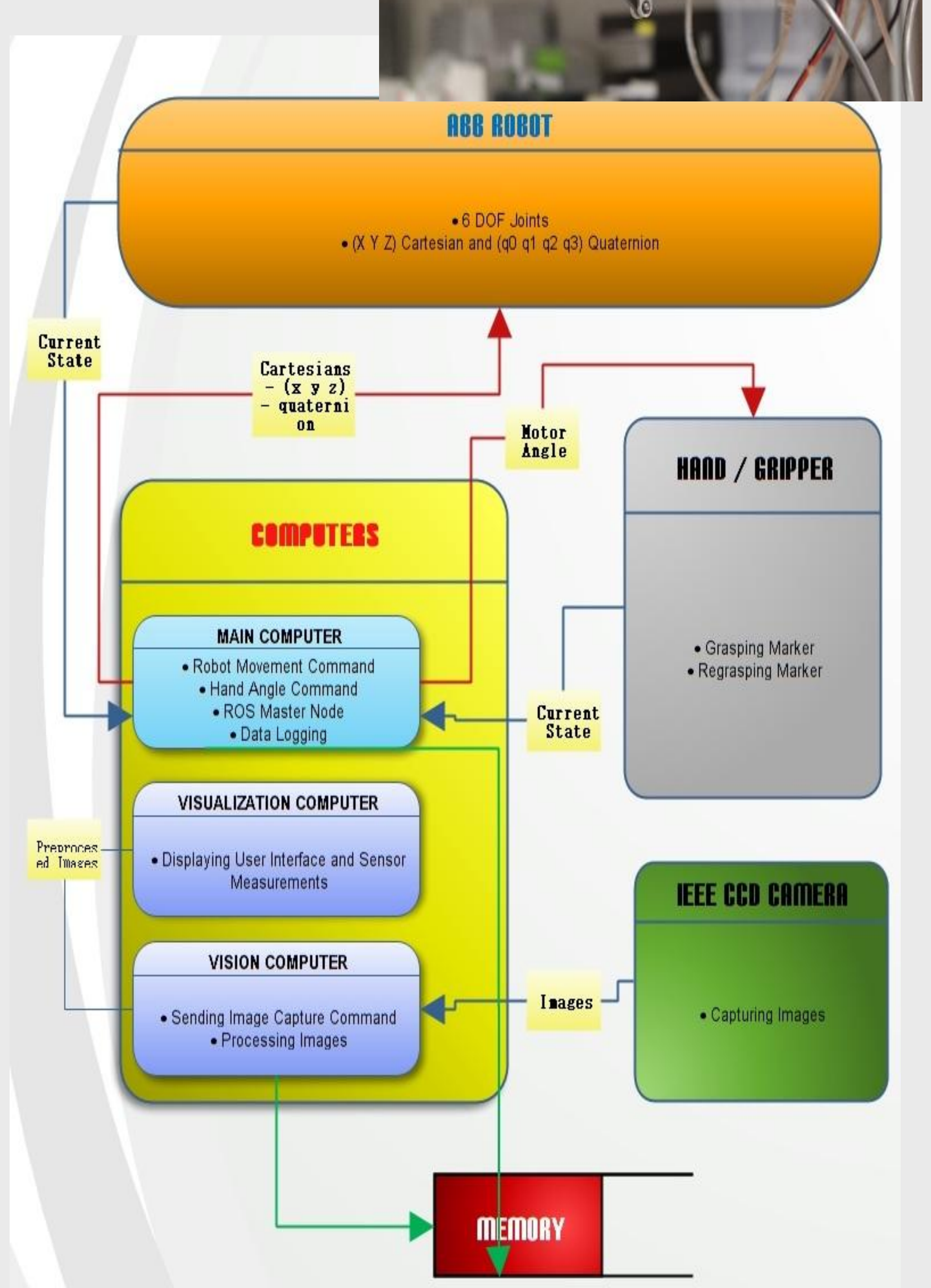

Figure 3. Data Flow Diagram of the System Implementation

METHODS AND MATERIALS

QUANTIFYING SYSTEM DELAY BASED ON SINUSOID TRAJECTORY

Our system is suffering from a delay that comes from the vision system processing. This makes synchronization task become intractable. One solution for this problem is by quantifying the exact amount of delay that the system currently face and manipulate the ROS timestamps based on that value. Concretely, the robot is programmed to follow sinusoidal trajectory while the computer keeps querying images. The timestamps for these parallel processes are saved in a logger file. After running an object detecting algorithm for each image, we finally get the position of the hand and the corresponding timestamp. Displaying sinusoid graph generated from the images and robot log side by side gives us the amount of delay in ms.

GAUSSIAN PROCESS FOR LEARNING THE REGRASP FUNCTION

One of the best way to model uncertainty is using gaussian distribution. In this project, we learn the function model that maps from pre-regrasp marker pose (distance and orientation) to post-regrasp pose. First we collect as much as 450 datasets (this will be much larger in future work) , and then regress a non-linear function using gaussian process. Gaussian process (GP) is a practical, probabilistic, non-parametric statistical learning method using kernel machines. This supervised-learning algorithm particularly works great for regression and classification. In the implementation, we need to set several properties of GP including

1. Mean function using constant mean
2. Covariance function
3. Likelihood function using gaussian
4. Hyperparameters of those above.

Finally we run the GP for each of the pose element (distance and orientation), and generate 10.000 test sets for visualization part.

RESULTS

The result of the sinusoidal comparison between images and robot state from the logger is depicted in the plot below.

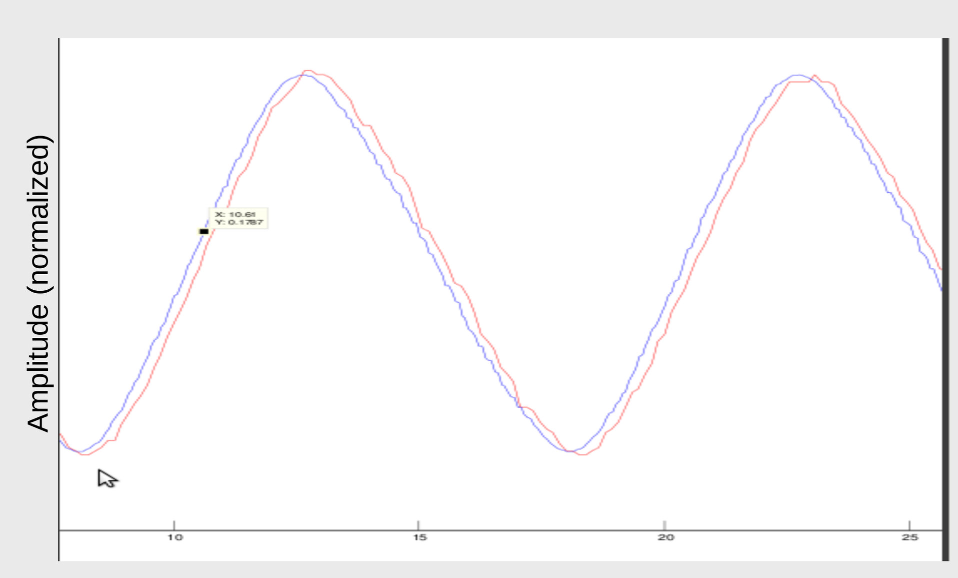

Figure 3. Sinusoid Trajectory Comparison

Based on the plot above, we successfully quantify the delay that occurs in the system (180 ms).

The result for the regrasp function regression is shown as follow:

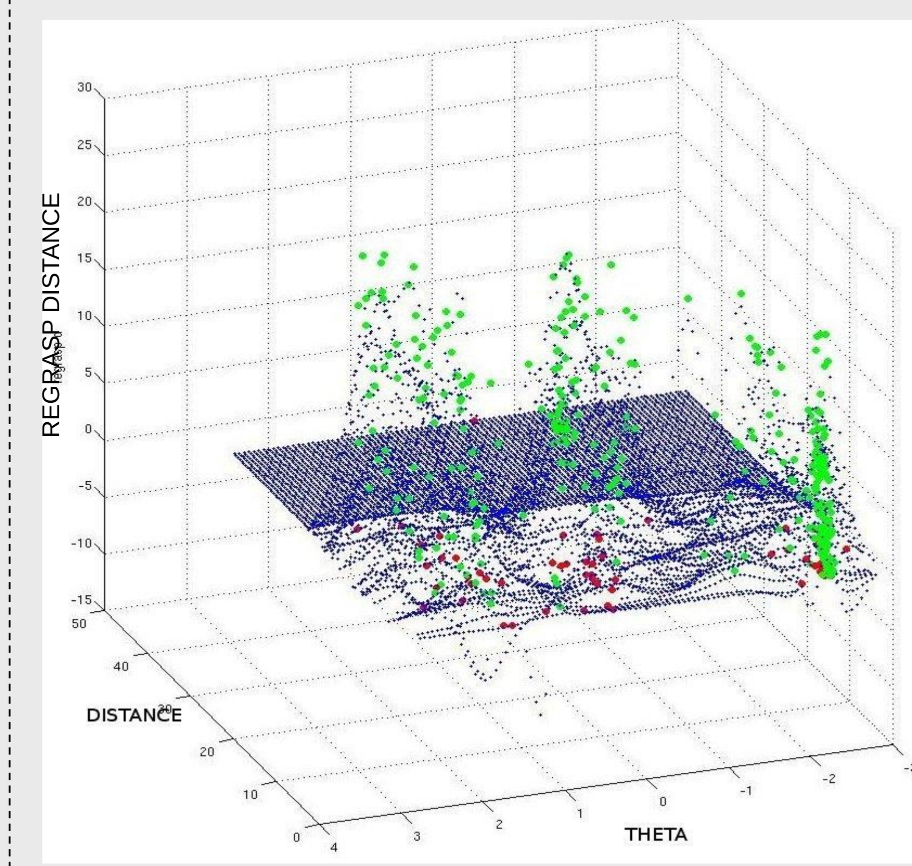

Figure 3. Gaussian Process regression of regrasp distance function. Green dots, blue dots , and red dots represent training set, test set, and failed regrasping respectively.

The gaussian process regress the function quite well, showing that there are 3 major clusters that the objects pose are likely to fall into.

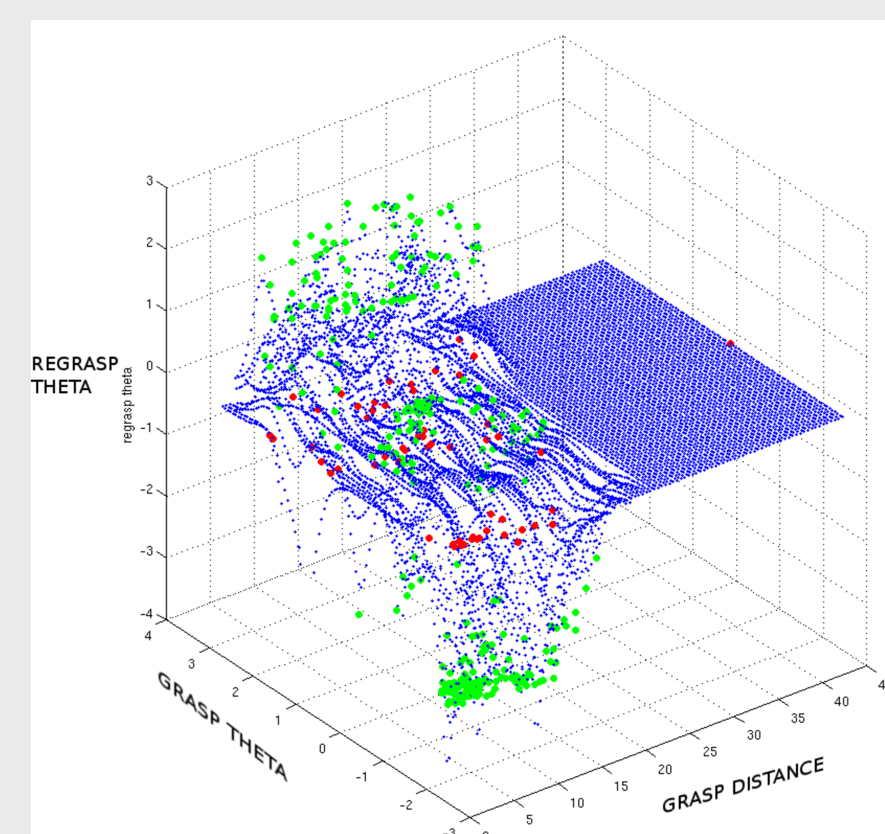

Figure 3. Gaussian Process regression of regrasp theta function. Green dots, blue dots , and red dots represent training set, test set, and failed regrasping respectively.

CONCLUSIONS AND FURTHER WORKS

The non-parametric machine learning method such as gaussian process has proven a quite satisfying result in regression problem. Though the cluster seems a bit tenuous, which likely because of relatively small amount of dataset, at least we get a forward view about the tendencies of the object pose. Further work will include building a multimodal probability distribution for each grasping pose and implementing parameterized regrasp function as the outcome of regrasp function.

REFERENCES

1. Paolini , R., Rodriguez , A., Srinivasa, S.S., Mason, M.T. : A Data-Driven Stastitical Framework for Post-Grasp Manipulation. In: ISER 2012
2. Rasmussen , C.E., Williams, C.K.I.: Gaussian Processes for Machine Learning. Press, MIT (2006)
3. Rodriguez , A., Mason, M.T., Srinivasa , S.S., Bernstein, M., Zirbel, A. : Abort and Retry in Grasping. In: IEEE IROS 2011.