Visually Guided Reaching and Grasping Primitives for an Educational Robot

Senior Research Thesis
School of Computer Science, Carnegie Mellon University
March 2008

Zhengheng Gho
Advisor: Professor David S. Touretzky

Abstract

Background

Manipulation is not common in educational robots. Most of them do not include arms. Those that include an optional gripper, such as the Khepera, do not provide any sort of high level primitives for manipulating objects; they merely offer the user the ability to operate the gripper motors. While inexpensive 6 degree of freedom robot arms for educational use are commercially available, such as the Lynx Motion arm, they are sold as stand-alone products with no vision system. This limits them to blindly executing fixed trajectories pre-specified by the programmer.

Introduction and Goal of Project

The project aims to develop high level, visually-guided manipulation, primitives for an educational robot named Regis. The work will be done in Tekkotsu, an application development framework for mobile robots being developed at Carnegie Mellon.

Regis is a new prototype educational robot developed in the Tekkotsu lab at Carnegie Mellon. It consists of a Lynx Motion wheeled rover base, a 600 MHz Gumstix computer, and two Lynx Motion arms. One is a 6-dof arm with a gripper, called the "crab arm"; the other is a 4-dof arm with a webcam on the end, called the "goose neck". The goose neck design allows the robot to view both its surroundings and its own body, including the crab arm and gripper. This makes it possible to control the crab arm by visual servoing and generate novel trajectories based on currently perceived object locations and orientations.

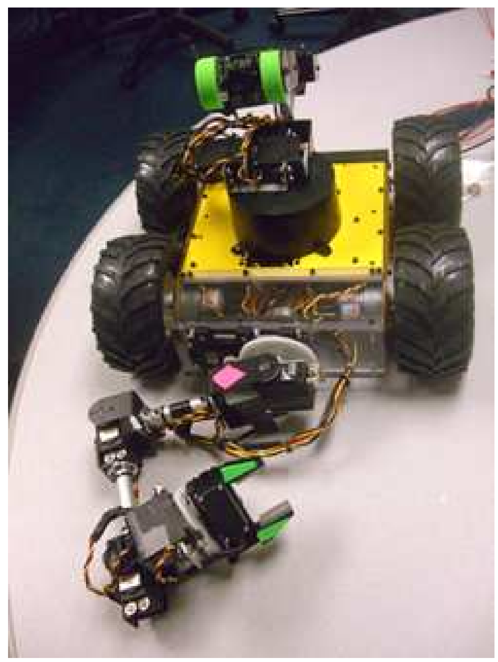


Figure 1 : Regis

“High level primitives” are the desired result of the project, replacing low-level commands to control individual joints. Examples of high level primitives include grasping an object, pushing an object (such as brick) from one location to another, toppling an object (such as a cylinder), and changing the orientation of an object.

There are several reasons why visual guidance is an essential component for the manipulation to be performed successfully. First, object location and orientation relative to the robot cannot be determined with pinpoint accuracy due to errors in determining the camera pose. These errors are part of the system due to inexpensive servos built with flimsy plastic and aluminum components. Coupled with the effect of gravity, the combined errors make it impossible to compensate for just with basic kinematics. Second, the crab arm is also subject to the same limitations. It will be near impossible to interact with an object with purely open loop control. Third, vision is necessary to detect errors when they occur. For example, the crab arm might miss gripping the object and a recovery strategy can be performed.

Test of Concept – Basic Grasping with Error Detection and Recovery

A basic demonstration of Regis grasping a cylinder is currently working. The demonstration relies on vision for determining the object position and setting up the approach path to bring in the gripper.

Basic Algorithm Explanation

Instead of approaching the target directly, we set up an approach gate which is close to the intended target but leaving enough space for manipulation of the gripper such that we will not knock down the target. The approach gate could be anywhere around the target but the approach path must be such that the target is in the direction the fingers are pointing to, such that there is a direct path from the gripper to the target. By minimizing the distance of the approach path, we will reduce the error that might occur when approaching the target (ie, since the gripper travel a smaller distance, the chance of it veering off course / knocking down the target is also lesser).

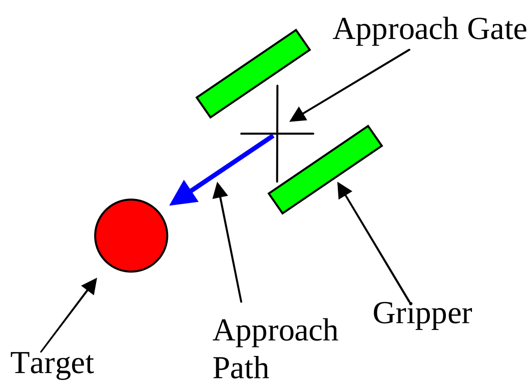


Figure 2: First Target

After the gripper reaches the approach gate, the gripper is rotated such that there is a direct approach path to the target. The final step will be moving the gripper along the approach path until the center of the gripper coordinate is as close as possible to the target before grasping the target.

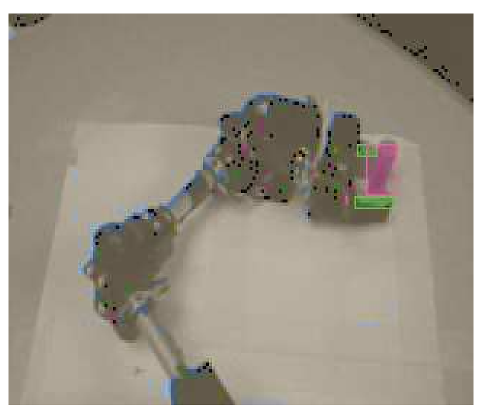


Figure 3:Image from Camera

Error Detection and Recovery
Error detection is performed by detecting if the target is within the centre of both grippers. Yet there are times when the target occludes the gripper from the sight of the camera or the object might be in the centre of the gripper but there is no firm grip on the object. A basic movement of the gripper can solve the problem as there will be a change in the coordinate of the gripper but the object will not follow the gripper.

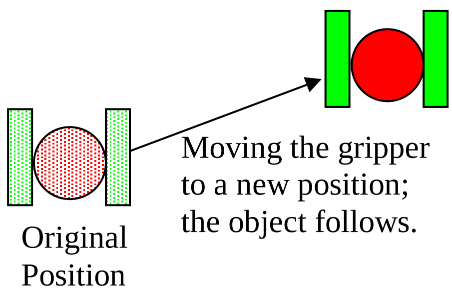


Figure 4: Successful Grasp

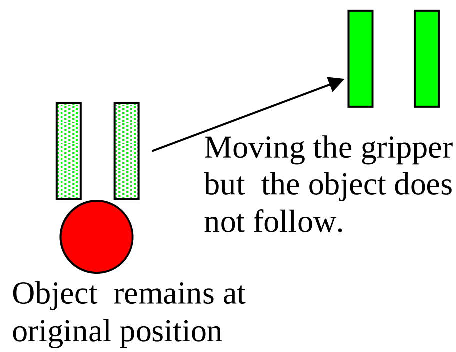


Figure 5: Unsuccessful Grasp

The current recovery process is to perform the approach and grasp again.

Conclusion and Future Work
The simple demonstration performs well with decent accuracy within the limited workspace defined for it. It is able to detect if the object is within the gripper and perform recovery. Future work will broaden the workspace, improve the setting and changing of the approach gate, and apply basic kinematics to make the motion more fluid and accurate.

Goose Neck Calibration

One of the efforts that is currently in progress is the calibration of the goose neck.

After testing, we concluded that the base and the pitch could be calibrated relatively independent of the other joints of the goose neck. On the other hand, we found out that shoulder and elbow can be calibrated relatively independent of the pitch and the base but they are dependent on the position of each other, hence they must be calibrated jointly.

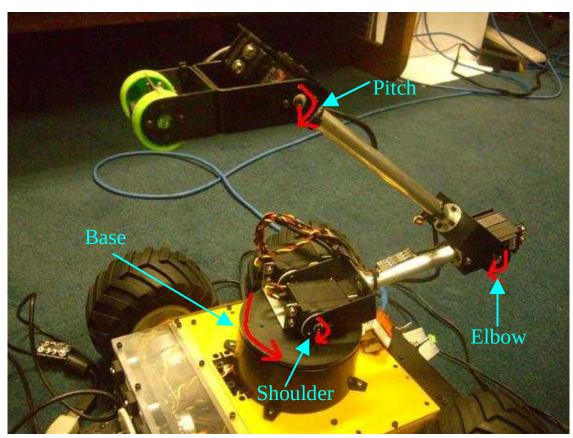
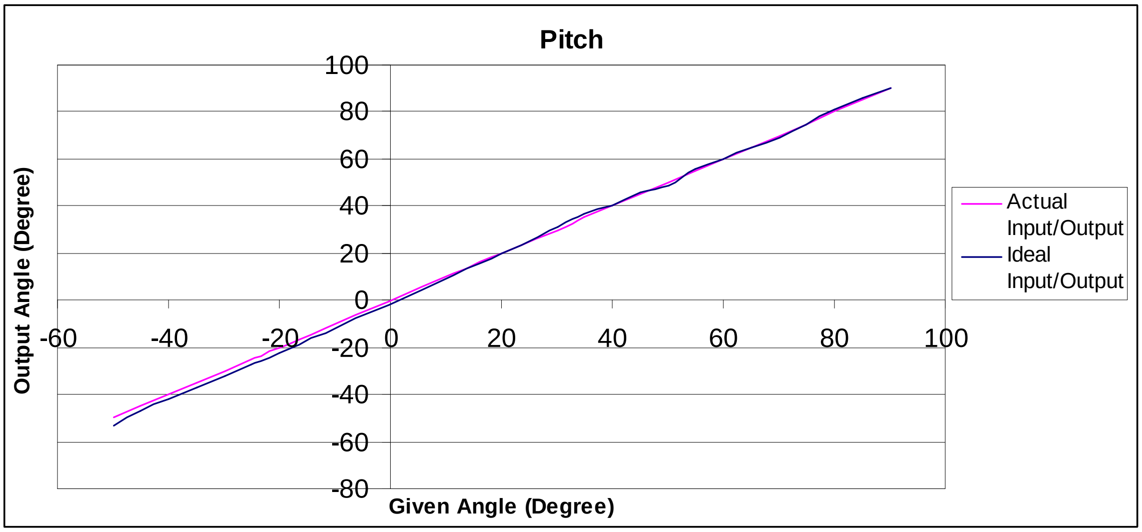


Figure 6: Components of Goose Neck

Base Calibration
When given an input angle, the base rotates to the given input angle with fairly decent accuracy (± 1 degree of error), thus no calibration is performed.

Pitch
The pitch is pretty accurate for the positive angles that are given to it as shown in graph 1. There are some deviations from the ideal angle towards the negative part of the graph and a simple linear regression is performed to correct the problem

Calibration for angles more than zero:
Return the given angle
Calibration for angles less than zero:
Expected Angle = 0.00769998+0.955882x where x is the given angle



Graph 1

Calibration of Elbow and Shoulder
The elbow has an inherent problem with its zero position being off about 8.5 degrees; this is due mainly to the gears. The interlocking teeth of the both gears make it impossible for a physical correction. Instead we corrected the problem by adding an offset to every position. This solves the problem pretty well.

On the other hand, each position of the shoulder depends on the elbow. If the elbow is extended further away (horizontally) from the base, the torque exerted by the elbow on the shoulder is greater. Therefore a greater correction factor is required. By taking data points at 10 degree intervals for each position for the shoulder and the elbow, we hope that by applying a linear function, we can alleviate the problem. The formula for the elbow will utilize equal weights when considering the values as the elbow does not really depend on the shoulder. The shoulder will utilize a weighted formula depending on which elbow position it is closer to.

Example:

Shoulder Desired Angle (SDA)	Shoudler Command Required (SCR)	Elbow Desired Angle (EDA)	Elbow Command Required (ECR)
-50	-55.58	-40	-30.37
-60	-64.46	-40	-30.37
-50	-55.00	-50	-40.68
-60	-63.03	-50	-40.68

If the desired angle for the shoulder is -57 (USA) and the desired angle for the elbow is -47 (UEA), we will search for the four nearest values for the input angle that is needed to reach the desired angle (for the edge cases, we will interpolate from the previous four values).

Hence the formula for input angle require for the desired angle will be:

Elbow (Equal Weights):

((ECR when (SDA = -60 & EDA = -50) + ECR when (SDA = -50 & EDA = -50)) / 2
– (ECR when (SDA = -60 & EDA = -40) + ECR when (SDA = -50 and EDA = -40)) / 2)
* abs(UEA – (-40)) / 10 +
(ECR when (SDA = -60 & EDA = -40) + ECR when (SDA = -50 and EDA = -40)) / 2
= ((-40.68 + -40.68) / 2 – (-30.37 + -30.37)) / 2 * (7 / 10) + (-30.37 + -30.37) / 2
= -37.587

Shoulder (Weighted):

(((SCR when (SDA = -60 & EDA = -50) * abs(UEA – (-40))
+ (SCR when (SDA = -60 & EDA = -40) * abs(UEA – (-50)))
- ((SCR when (SDA = -50 & EDA = -50) * abs(UEA – (-40))
+ (SCR when (SDA = -50 & EDA = -40) * abs(UEA – (-50))))
* abs(USA – (-50)) / 10 +
((SCR when (SDA = -50 & EDA = -50) * abs(UEA – (-40))
+ (SCR when (SDA = -50 & EDA = -40) * abs(UEA – (-50)))
= (((-63.03 * (7/10)) + (-64.46 * (3/10))) – ((-55.00 * (7/10)) + (-55.58 * (3/10)))) * 7/10
+ ((-55.00 * (7/10)) + (-55.58 * (3/10)))
= ((-44.121 + -19.338) – (-38.5 + -16.674)) * 7 / 10 + (-38.5 + -16.674)
= (-63.459 – (55.174)) * 7/10 + (55.174)
= 60.7935

Conclusion and Future Work

While testing and calibrating Regis, we noticed that the initial position and the speed of approach might also affect the final position. Hence we would have to test how accurate is our calibration as a whole system after integration of the code.

Acknowledgement

Thanks to Ethan J. Tira-Thompson for guidance through the project.
Supported by NSF 0717705