Term Paper: Robot Arm Modeling

Akul Penugonda

December 10, 2014

1 Abstract

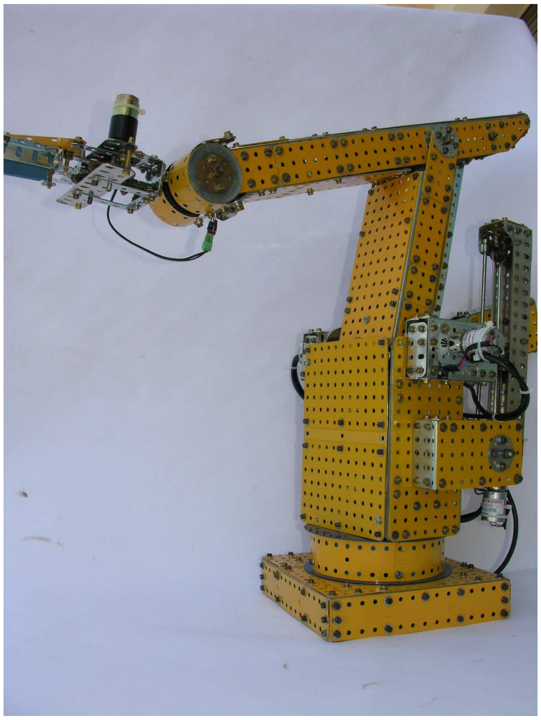
This project attempts to model and verify the motion of a robot arm. The two joints used in robot arms - prismatic and rotational. Both will be used in this project. There are several stages to proving/modelling the motion of a 3-dimensional robot arm. These are the simplest, one-dimensional case of a prismatic joint, the one-arm case of the prismatic and rotational joint in two dimensions, then the two-arm case of the prismatic and rotational joints. This paper will discuss the methodology by which these models were constructed, then discuss how these models were proven. I found that the lower dimension models were easier to prove, but the higher dimensions were far more difficult. This project succeeded in describing a one-link rotational arm, and in proving a two- and one-dimensional prismatic arm.

2 Introduction

2.1 Background

Robot arms are used nowadays in many scenarios, from healthcare, to manufacturing, to the International Space Station. Typically, their job is to perform repetitive tasks that humans are either incapable of performing repeatedly with a high precision or that humans simply do not want to do [1]. They are also used in humanoid robots - as arms, go figure. As the cost of these arms decreases in relation to that of human labor, their use will become more widespread than it already is. Robot arms have been in existence as early as 1937. The first robot arm was built with rudimentary solenoids and only one electric motor (in contrast to the practice today of an actuated motor on each joint). In the classic style of earlier computers, the arm 'read' instructions off of a paper tape with hole punches. [2]

A major development in robotic arms was the development of the Stanford Arm at Stanford University. This arm had the ability to move with six degrees of freedom - it could move in 3-dimensional space, and it's wrist could rotate about all 3 axes. This enabled the arm to have the ability to move in a manner analogous to a human arm, paving the path for a robot that could replace some of what humans were doing in factories. The first industrial arm was released a short time later, in early 1974. So began the widespread use of robot arms in factories worldwide, which continues today.



The structure of robot arms can be discussed in a general fashion. Robot arms are made up of links and joints. The joints may either be prismatic, meaning that they extend forward in the direction of the joint it is attached to, or rotational, meaning that they change the angle between the links they are connected to. By combining these in different arrangements, it is possible to create any number of robot arms. For example, the Stanford arm contained 3 rotational joints.

Typically, robots are taught the paths they need to move through the operator waving a pendant - a control path. This gives end-effector positions to the robot. The arm should be able to choose velocities that will enable it to avoid obstacles in its path. In this proposal, I will model such a procedure - meaning that I will model the motion of the end-effector of the robot. Each joint will either move at a constant velocity or be at rest.

3 Related Work

There has been some work done on verifying the safety of robot arms. Most industrial safety measures involve the use of panic switches and detectors.

In 1985, Linger was one of the first to propose that robot safety should be addressed with "high knowledge about the process", regarding the task they were to perform [4]. Safety systems were developed that would analyze the working area of a robot in real-time to verify that they did not collide with anything in their motion. Dhillon and Yang in 2001 were the first to pioneer the use of modeling techniques to verify the safety of a robot with a build-in state machine. Their method treated the robot as a state machine, and modeled it's uptime/downtime. [3]

These all give rise to the question, what does it mean to be safe? In 2006 Kulic and Croft created a strategy for robot-human interaction by which the robot modulates its velocity in response to its surrounding conditions, with the goal that if it hits something, it will not hit it hard enough to cause damage, and tries to be safe. [5]

Some work on logical temporal analysis was done by Rahimi and Xiadong for robot software verification. They used a high-level temporal logic system (different from dTL) to analyze robot software, and verify that actions met some desired pre or post condition. Their verification system used a high-level logic system that directly simulated the reception of sensors and other information, and was somewhat akin to a programming language. An effort at Carnegie Mellon led by Reid Simmons is also progressing. [6]

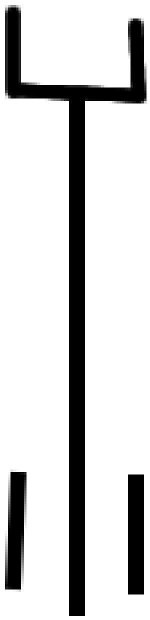
These methods differ from the aim of this project. Approaches relating to active safety systems involve finding safety violations after the creation of the robot, in practice. This is great, since it avoids actual failures, but it also fails to account for why the violations are actually occurring, nor try to prevent them. The temporal analysis described above is high-level, involving such predicates as BEFORE, AFTER, and DURING. These are higher-level constructs used to prove systems that may not reflect the same properties as dTL, since similar forms of reasoning of differential equations and other such things are not present.

In addition, these projects are general-purpose robot proofs, not targeted at robot arms in particular. Robot arms are different from rovers, roombas, and other similar bots since they do not move on wheels across a plane, but rather rely on fine joint motions. It is a different sort of idea.

Currently, robotic arms are modeled using the Denavit-Hartenberg method. This method provides a transformation matrix for each link, transforming one endpoint on the link to the next. Composing these matrices with each other yields a matrix that gives the position of the end effector in terms of the angle positions of the joints. The end-effector positions may be reverse-engineered (in most cases, depending on the arm) to yield the joint angles in terms of the end-effector position. These calculations are used everywhere it is necessary to command a robot arm.

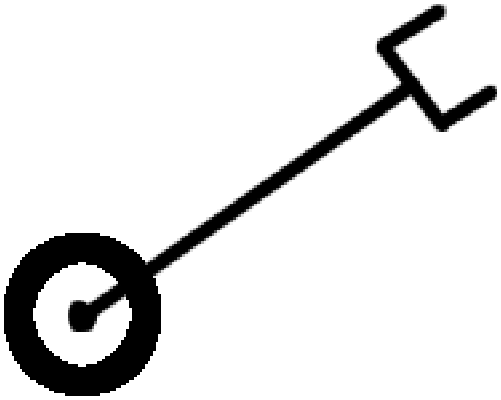
4 Approach

The key component of modeling this system was safety. I wanted the joint/arm to be able to move without actually hitting an obstacle with a given position and size. This obstacle could represent a physical barrier or a human.



For the 1-dimensional prismatic arm, this was easy - the input obstacle must be a y-coordinate. Since the origin of the robot was defined at (0,0) and it extended in the positive y direction, it was evident that for a collision to occur, the y position of the end-effector must be greater than that of the obstacle. Thus, the velocity of the arm chosen every time T is limited such that the arm's length cannot be greater than this y position or less than 0.

For the 2-dimensional prismatic arm, the obstacle was modeled as a sphere. Since the joint could only move out in a straight line, it's possible to take the projection of the position vector of the obstacle onto that line and then check it's magnitude to determine whether a collision is occurring. This point sets a natural upper bound for the distance the prismatic joint can extend. At this point, the problem can be reduced to the 1-dimensional version, so the velocity safety condition is the same.



It was quite difficult to choose a metric for obstacle avoidance that did not unnecessarily restrict the motion of the arm. It would be possible to make it easy and restrict it's x and y motion, but that is an unreal constraint on a robot arm's behavior. So, there are two different cases - if the obstacle is counterclockwise or clockwise to the end effector. Then, if the desired end-point is the opposite in relation to the obstacle, the velocity should be in the direction of the obstacle. Otherwise, it should be the opposite. This produces 4 different cases. A safe velocity is restricted by that direction, and also by whether it will produce the correct endpoint.

The differential equations for each of these arms can be derived from the position of the end-effector in terms of the joint parameter. For prismatic joints, they are the same, so the chosen velocity will be the velocity of the end-effector. For the rotational joint, the velocity is the angular velocity of the joint, which when multiplied by the length of the joint becomes the position of the end-effector.

5 Model Assumptions

For the prismatic arm in 1 dimension, the initial position must be less than that of the obstacle, since otherwise it would start out colliding! the position should also be greater than zero, since a prismatic joint can't go negative by nature. T, a time constant, is greater than 0 since time intervals must be greater than 0.

For the prismatic arm in 2 dimensions, the initial length starts out greater than 0, a reasonable assumption. If it started at zero, the obstacle could be placed immediately on top of it, preventing it from moving at all. The radius of the obstacle must be greater than zero, otherwise it would not make sense realistically. The length of the joint must be less than the projected distance of the obstacle plus the radius - otherwise a collision would occur. Also, the magnitude of the direction vector of the joint is 1, as is custom for direction vectors.

For the rotational arm in 2 dimensions, we first make sure that both the start point and end point are the length of the link away from the origin. Also, neither should collide with the obstacle at the outset, since otherwise it would not be possible to move from one to the other without a collision. The obstacle is also stipulated to be within a link's length from the origin, since otherwise no collision would ever occur.

6 Proof Strategies

The first two instances were easily proven, so I did not need to come up with any advanced strategies for them. I was unable to prove the rotation of the 1d arm as safe. I attempted several strategies. One possibility was allowing the arm to collide with an obstacle momentarily, but then it would halt and reverse. Another was to cut in distance from the end-effector to the obstacle sphere. Neither were effective - the first created too many new cases, and the second did not hold.

7 Lessons Learned

This problem is very difficult - it's quite evident why there is no literature directly ascribing to this topic. The complexity increases very fast from prismatic to rotational joints, and even more highly when combining them. In every case, the safety decision regarding which velocity to pick becomes more and more difficult, since the range of motion of the robot increases during the differential equation, and more cases need to be accounted for.

Something else that dramatically increases difficulty with this problem is the inability of KeyMaera to reason effectively with trigonometric functions. It is comparatively easy to come up with representations for distance and intersection using these functions - without them, it is a much greater struggle.

The goals of this project initially were to model a 3d robot arm, with two links. This was not met for the reasons above - I failed to account for the sheer complexity that adding more joints would create, which is why I have only submitted models with one joint.

8 Deliverables

OneArmRot.key contains the model for a one link arm with a rotational joint. OneArm2d.key contains the model for a one line arm with a prismatic joint in 2 dimensions, and Planar1dprismatic.key contains the model for a prismatic joint in one dimension. Corresponding proof files are present for the latter two models.

References

[1] Section iv, chapter 4 - industrial robots and robot system safety. *OSHA Technical Manual (OTM).*

[2] An automatic block-setting crane. *Meccano Magazine*, 23, 1938.

[3] N. Yang B. S. Dhillon. Formulas for analyzing a redundant robot configuration with a built-in safety system. *Microelectronics Reliability*, 37, 1997.

[4] K. L. Liu B.S. Dillon, A. R. M. Fashandi. Robot systems reliability and safety: a review. *Journal of Quality in Maintenance Engineering*, 8, 2002.

[5] E.A. Croft D. Kulic. Real-time safety for human-robot interaction. *Robotics and Autonomous Systems*, 54, 2006.

[6] X Xiadong M Rahimi. A framework for software safety verification of industrial robot operations. 1991.