

Robotic Limb Calibration: Accelerometer Based Discovery of Kinematic Constants

Jeffrey R Croxell¹, Jerry B Weinberg³, Ryan W Krauss², Scott R Smith¹

Southern Illinois University Edwardsville

¹Department of Electrical and Computer Engineering

²Department of Mechanical Engineering

³Department of Computer Science

Abstract

Robot arms carry a unique set of challenges with setup for use with inverse kinematics. An automated calibration for such robots would prove beneficial. A solution is discussed utilizing accelerometer feedback on the tip of a robot arm and an automated calibration routine.

Introduction

Robotic arms have established their place as a useful tool in modern society. This includes industrial applications, space exploration and as research tools. Bipedal and humanoid robots are also gaining popularity as toys and as educational and research tools (Weinberg, et al. 2008). Limbed robots of any form carry some inherent difficulties not found in other platforms, such as wheeled or tracked robots. One specific area of difficulty is with setup. Robots with linkages typically require the use of careful measurements and complex kinematic equations for setup and calibration. Not only are these difficult, but if a robot is slightly different or changes, measurements and equations must be reworked. For many people, these setup requirements are not understood without college engineering courses, leaving this type of platform inaccessible to much of the population.

Running a simple startup calibration routine on the limbed robot to solve the complicated kinematic equations automatically would be ideal to make the platform more accessible. Expanding this further, such kinematic self-discovery could be combined with learning algorithms allowing bipedal robots to discover their limbs and take steps on their own much like how children learn to walk.

Inspiration

The typical way to interact with robotic arms is with inverse kinematics. This allows the user to specify a location in three dimensional space where they require the tip of the arm to be located. This makes the arm easy to use at a high level, otherwise users would find themselves controlling individual joints of the arm, and as the complexity of

the arm increases, so does this task. It is nice to have this high level control, unfortunately it is also difficult to setup. The more joints and links involved the more complicated the analysis and equations of the arm becomes.

It is the analysis process that makes using these arms difficult for some, but impossible for others. For example, at SIUE this material is taught in a robotics control class that is made available to seniors and graduate students. This makes the platform largely unavailable to students who have not reached this point in their academic career. The other problem comes into play with difficulty and setup time. Even those that know how to analyze these arms may find the process tedious and difficult. This is exacerbated by changes that can occur with these types of arms (part replacements, joint changes, limb changes, etc.) - that can require anywhere from small tweaks to complete reanalysis of the kinematic equations.

On walking style robots the problem is multiplied by the number of limbs you are dealing with on that robot. The unfortunate side effect is that there is an entire platform that is unreachable to a large portion of enthusiasts, educators, and students. If we can make the setup of these platforms completely painless, it would provide an additional platform for many people, and also improve the experience of others that use these kinematic equations on robots.

Chandrasekar and Bernstein (2008) used acceleration feedback in simulation to control a robotic arm. However, we were unable to find any related work on using accelerometer feedback for automated discovery of kinematics for robot limbs.

Method

To make setting up robotic arms easier, the obvious solution is to solve the kinematic equations automatically. These equations are made up of some basic physical attributes of the arm, specifically length of links, joint offsets, joint orientations and joint angles. For the typical arm, limb length, joint offsets and orientation all stay the same while some type of motor drives the revolute joints to various angles. This means that to solve the equations automatically we still need to find a set of known quantities.

For the scope of this paper we are focusing on finding the length of the links on the arm in question using the fewest known's possible. Arms containing only revolute joints use inverse kinematics to specify the joint angles to the arm based on the desired tip position in space. Thus the joints require position feedback in order to perform inverse kinematics, so we will use this to our advantage. With an additional piece of hardware, an accelerometer, located at the tip of the robot we can gain additional feedback needed to solve for the length of the arm.

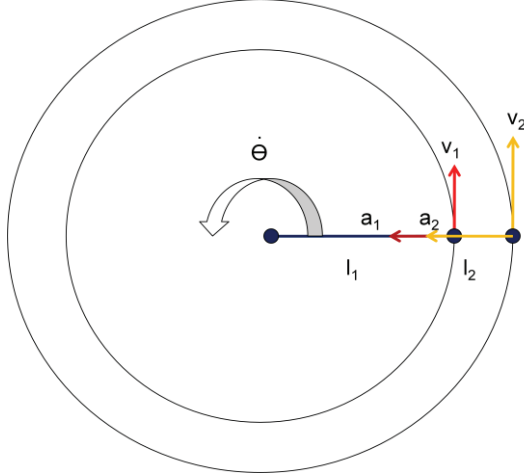


Figure 1: Dynamics occurring on a single link robot.

The accelerometer gives us the acceleration at the tip of the robot. The acceleration in particular that we are interested in is the normal acceleration, or centripetal acceleration. If we then utilize angular velocity($\dot{\theta}$) as our second known, we can find the radius, or in this case the length of the robot limb. In this type of circular motion, the acceleration is proportional to the length of the arm for a given angular velocity. This can be seen in Figure 1, if a limb is turned at a constant $\dot{\theta}$ angular velocity, the normal acceleration (a_1 and a_2) will depend on the length of the arm. This simple use of physics gives us the equation $r = \dot{\theta}^2 / a_n$ where r is the length of the limb in question. This equation gives us a continuous reading of the length of the link in question, so regardless of the speed, direction or amount of movement, the algorithm will attempt to solve the length.

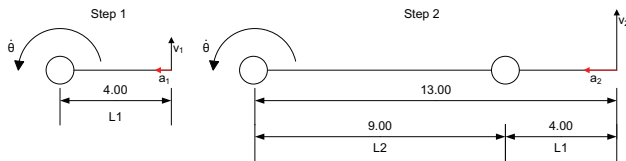


Figure 2: Calibration routine of a two link arm.

In order to solve for a multi jointed arm we must analyze one link/joint pair at a time. The calibration involves the structured movement of the limb(s) and the data collection as well as calculation. Moving one joint at a time results in a velocity at the tip proportional to the length of the limb.

For example in Figure 2, moving L2 while keeping L1 static results in the length of both links together. Then, moving L1 individually results in the length of that individual link. Working backwards the algorithm can then deduce the length of L2 as well.

Physical Implementation

The physical implementation includes a very simple two link arm with limbs 18 inches in length, and DC gearhead motors for the joints using optical encoders for feedback. To actually extract velocity information from these positional encoders we must differentiate the positions over time to achieve velocity data. This can result in a somewhat noisy signal, so it is put through a low pass filter to clean it up. A VTI SCA3000-D02 3-axis accelerometer was utilized on the tip of the arm for acceleration data, chosen for its high resolution and ease of use. It provides an I²C digital serial interface making it versatile and easy to connect with almost any microcontroller. It also means that filtering is done on board the accelerometer leaving less processing requirements for the user. Both joints are controlled via a Cypress PSoC 29466 8-bit microcontroller sending outputs to an H-Bridge for both joints and encoder data looping back to that same PSoC. All motor control is done at this level, only angles are sent to this microcontroller, at the same time it is continually checking the current velocity. Data from the accelerometer is captured using another Cypress PSoC 29466 microcontroller which polls the joint control PSoC for the current joint velocities. At this point both pieces of data needed to check the length are on the PSoC and the length can be calculated or relayed back to the PC for further processing through a serial link. The accelerometer PSoC is set as the central location for both data sets for synchronization issues. This being an acceleration feedback based system, it is sensitive to disturbances. Care was taken to keep the base of the arm level and anchored to the operating surface. The gear trains also needed to be smooth and clean. Control of the arm required a shaped input to reduce disturbances as much as possible, giving it gentle acceleration as well as a gentle deceleration to the stopped position regardless of the angle or speed desired.

Results

Measurements using the physical setup listed above have proven very encouraging. On a single link robot with even minimal movements we are able to measure the length of the arm. For test cases we utilized movements of 45 degrees and preliminary results show accuracy at about ± 1 cm on a 30cm long link. As expected this accuracy decreases as the link gets shorter because accelerations get smaller and are more susceptible to disturbances.

For testing purposes we use several angles at different speeds to see how this affected the measured length. For this algorithm it should compensate for all of these factors

and ideally show the same length. As we see in Figure 3, we ran the arm from 0 to 90 degrees and back several times, then from 0 to 45 degrees and back. In the dashed line is theta-dot in rad/sec which shows the angular velocity of the joint during each of the movements. In the dotted line the normal acceleration that the rotation causes at the tip of the arm, which is measured in G-forces. The algorithm then uses these two pieces of data in real time calculating the length of the limb, which is displayed by the solid line, measured in meters. We can see that even changing the angle traveled to as well as the angular velocity has a minimal affect on the measured length calculated continuously across the graph.

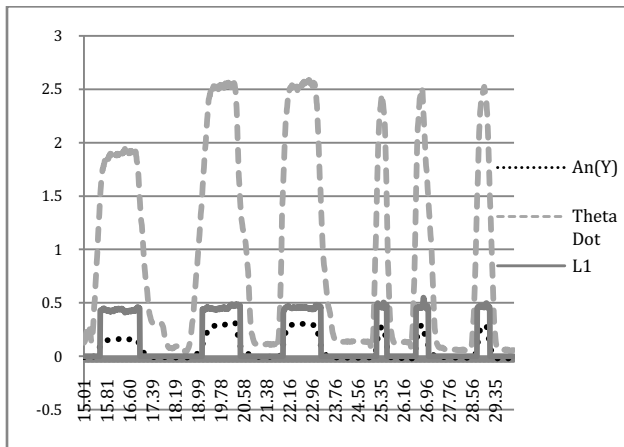


Figure3: Measurement results from 6 movements.

This shows that we can calculate the length of the arm fairly well in real time. While this is nice, it is probably rarely necessary. We can instead run a calibration routine beforehand, remove outlying points, and average all of the calculated lengths together. This gives us a much more accurate reading on the length of the arm.

When adding another link, maintaining accuracy becomes much more difficult. Locking the second joint and moving the first joint we can still maintain fairly good results. The more difficult part comes when continuing the calibration routine, locking the first joint and moving the second. Without changing parameters or treating the joint any differently, it is difficult for it to counteract the force of the second joint and link. The point at which the first joint rebounds and counteracts those forces greatly affects the measurements being made on the length.

Future Work

The most immediate work to be completed is work on the control of the first and second joints to allow us to make a more acceptable measurement of a two link arm. This should be achievable without more prior knowledge of the arm, or more accelerometers. If better control of these joints cannot solve this problem, possibly a learning algo-

rithm to allow the machine to understand the weight and force of the second link, allowing it to counter act those forces autonomously.

Assuming the additional features to improve a two link robot, a more complete solving of the constants for the kinematic equations would be a logical short term step. This would allow for a full realization of the project: turn the robot on, automatic calibration, specify coordinates in space and the robot is able to find them automatically.

Once this is complete on a single robot arm, the ideal platform to continue this work on would be multi limbed robots, such as a humanoid. With accelerometers on each of the hands and each of the feet, a humanoid robot could potentially calibrate itself. Having a robot of this nature with the kinematics analyzed automatically would be convenient, but the potential of it being able to learn to walk by itself or even learn different gates if it finds that its limbs have changed would be absolutely amazing.

There is always the ability to expand the original work on to other practical resources such as industrial robots. With the ubiquity of industrial robots, automatic calibration and setup as well as continuous feedback of this type could potentially find a place in this field.

Conclusion

Robot arms have been a large part of the robotic industry for some time. They are finding their way into factories, universities, and even homes, yet it still takes an expert to set one up. It would be nice to have these details taken care of automatically. We have shown that with one additional source of feedback this is a real possibility. Simple calibration routines can yield the length of the links on a robot arm. Filling in these constants leads us to our kinematic equations, giving us high level use of the arm.

The impact of this research could traverse multiple industries and fields of study. Industrial robots would benefit through the simple setup and constant feedback. Bipedal research robots would have interesting possibilities in self-discovery using this feedback to learn how to use their limbs. Educators would benefit from the instant, no setup necessary use of legged robots as an additional robotic platform to teach.

References

- Craig, J. J. (2005). Introduction to Robotics Mechanics and Control. Upper Saddle River, NJ: Pearson Education.
- Chandrasekar, J & Bernstein, D (2008).Position Control Using Acceleration-Based Identification and Feedback With Unknown Measurement Bias.Journal of Dynamic Systems, Measurement, and Control. 130, 1-9.
- Weinberg, J.B., W. Yu, K. Wheeler-Smith, R. Knight, R. Mead, I. Bernstein, J. Croxell, & D. Webster

(2008). Making Intelligent Walking Robots Accessible to Educators: A Brain and Sensor Pack for Legged Mobile Robots. Technical Report of The 2008 Association for the Advancement of Artificial Intelligence (AAAI-08) Workshop on AI Education. Chicago, IL. 84-89.