IB HL Exploration: Modeling a Robotic Arm

Date: 27 February 2014

**Introduction**

My motivation for choosing this topic was a problem that I had to solve as a programmer in Robotics Club last year. Our robot design involved a two-jointed robotic arm that we could manipulate to grip various objects. Programming the arm involved understanding the kinematics of it on a mathematical level. However, I did not have the sufficient mathematical background at the time to manage the problem, and the original arm design was eventually discarded for something simpler. Now, with greater knowledge of math, I am curious as to how this problem would be solved, which is why I have picked it as my exploration topic.

In the field of robotics, robotic arms are generally constructed to perform various tasks like gripping, picking up and transporting objects, and welding. A robotic arm consists of *links* connected by *joints* which allow for rotation, ultimately moving the *end effecter*, or terminus of the arm (equivalent to a human hand), to the desired position. In order to design, build, and program such arms, it is necessary to create mathematical models that can represent certain aspects of the arm such as the relationship between variables like joint position and end effecter position, as well as the rates of change of those variables.

*Figure 1: An example of a robotic arm*

The aim of this investigation is to construct a mathematical model for the motion of a two-joint robotic arm in a way that would allow me to control the position and velocity of the end effecter by programmatically manipulating the angles and angular velocity of the joints. Thus, I must find general equations that relate the angular velocities of the joints (how quickly they are rotating) and the linear velocities of the coordinates of the end effecter. In mathematical terms, for end effecter coordinates  and joint angles  and , I am aiming to find and , which are the angular velocities of the joints, in terms of  and , which are the linear velocities of the x- and y-components of the end effecter position.

**Investigation**

*Forward kinematic problem*

****

*Figure 2: Diagram of robotic arm*

(Source: MIT Introduction to Robotics course notes)

In Figure 2 is an illustration of the kind of robotic arm that I will be modeling for this exploration. There are two joints, whose angles are denoted with  and , which connect the links whose lengths are denoted with  and . Note that  is measured relative from Link 1. For simplicity, I will assume that the constants  and are both equal to 1. I will denote the x- and y-coordinates of the end effecter as .

Before I can begin finding equations relating the linear velocity of the end effecter and the angular velocities of the joint angles, I must find equations that relate the position of the end effecter and the joint angles. One way to mathematically represent robot arms is exemplified by a classical problem in robotics: the *forward kinematic problem*. In the forward kinematic problem, one is given the lengths of each link and the angles of each joint in the robotic arm, and must calculate the final position of the end effecter. Solving this problem will give equations for the coordinates of the end effecter as a function of the angles of each joint in the arm.

Solving the forward kinematic problem involves basic vector addition. If we define  as a vector with a magnitude of  and directional angle of , and  as a vector with a magnitude of  and directional angle of  measured from , then it is evident from Figure 2 that:

  (1)

 Having derived expressions for the coordinates of the end effecter, *x* and *y*, using the joint angles  and , I can now find equations describing how these variables vary with time; that is, model the motion of the arm.

*Differential motion*

If I wish to program the rotation of the joints of the arm in order to move the end effecter at some desired horizontal velocity and vertical velocity, then it is necessary to derive relationships between the angular velocity at the joints and the linear velocity of the end effecter.

 Consider again the equations derived as a solution to the forward kinematic problem, modified according to the new definition of . Also note that *x* and *y* have been explicitly written as functions of  and , as it will be important to remember that *x* and *y* are both dependent on multiple variables.

  (2)

 As a starting point to find the relationship between the rates of change of the end effecter coordinates and the joint angles, I will take the differential of the equations. For some single-variable function , the differential  is defined as:

  (3)

 When taking the differential of a multivariable equation, however, one must take the sum of the differentials of the partial derivatives of each variable in the function. A *partial derivative* is the derivative of a multivariable function where all variables are held constant except for some variable *x*; in that case, it is said to be the partial derivative taken with respect to *x*. Given some multivariable function , the partial derivative taken with respect to *x* can be denoted with . In order to find , one would take the derivative of normally with respect to *x*, but assume that *y* was equivalent to a constant.

Therefore, for some multivariable function , the differential could be written as:

  (4)

With this in mind, I will take the differentials of the *x* and *y* equations as shown below.

  (5)

Here, I will introduce the concept of the *Jacobian matrix*, denoted by **J**, which will be useful for finding the relationships that I am aiming to derive. The Jacobian matrix is the matrix of all of the first-order partial derivatives of a vector-valued function. A vector-valued function is a function where the input is one or more variables and the output is a vector. In mathematical notation, the Jacobian matrix for the vector-valued function  would be written as:

  (6)

In this problem, the Jacobian matrix “represents the differential relationship between the joint displacements and the resulting end-effecter motion” (Asada). As I will demonstrate, the Jacobian matrix will be very useful in this problem, as it relates changes in *x* and *y* to changes in  and .

If I define  and , then the equations in Equation 5 may be written in vector form as:

  (7)

Substituting values, **J**, would be equal to:

  (8)

I am concerned, however, with the rates of change of the variables with respect to time. I will then define the following vectors:

   (9)

If I divide the expression  by *dt* on both sides, then I may obtain the equation:

  (10)

Which is equivalent to:

  (11)

 This equation is useful for finding the linear velocity of the end effecter given the angular velocities of the joints. However, in an actual robot arm, the motors control the rotation of the joints, which means that I need to find an equation that can produce the angular velocities necessary to produce a desired end effecter velocity. This can be done quite simply through use of the inverse of the Jacobian matrix, as shown below:

  (12)

 In order to obtain explicit equations relating the angular velocities and linear velocities, it is necessary to expand the above vector equation. This involves calculating the inverse of the Jacobian matrix. The following general formula is used to calculate the inverse of any 2 x 2 matrix:

  (13)

 Calculating the Jacobian determinant and assuming that  and  are both equal to 1 for simplicity,

  (14)

 Using the angle sum identities,

  (15)

I may simplify Equation 14:

  (16)

The adjugate matrix of the Jacobian matrix would be written as

  (17)

Multiplying the reciprocal of the determinant from Equation 16 and the adjugate matrix from equation 17,

  (18)

 The inverse of the Jacobian matrix is equal to: 

  (19)

 Substituting **J**-1 and  as defined in Equation 9 into the equation  and multiplying the two matrices on the right-hand side of the equation,

  (20)

 The following explicit equations may be obtained:

  (21)

Having obtained equations for and in terms of and , I have achieved my original aim. To apply these equations to the physical world, i.e. apply them to controlling a robot arm, it would be necessary to control and measure the angular velocity of the joints. Realistically, two motors would be used to rotate the joints; to manipulate the angular velocity of the joints, one would simply vary the voltage being sent to the motors, which would be done programmatically. The angular velocity of the joints could be measured using devices called rotary encoders. To program the end effecter to move at a desired velocity, then, I would substitute the x- and y-components of the target velocity for and  into the equations above, producing the values for and  that would be needed to achieve and . I would then programmatically set the motors controlling the joints to run so that they rotate the joints as the angular velocities and .

Note that the angular velocities depend not only on the desired end effecter velocities, but also the configuration of the arm, i.e. the angles of the joints themselves. If and are to be kept constant as the arm moves, then and must vary throughout the motion as the configuration of the arm would be continuously changing. Therefore, if I wished to program a robotic arm, it would be necessary for the program to check at regular intervals of time the values of and in order to calculate the values of and that would keep and at a constant value.

There arises a problem, then, of determining the time step needed for checking the angles and updating the angular velocity in order to maintain a certain degree of accuracy of the resultant linear velocities. The desired accuracy would depend on the scale of the robot arm. For an arm with links several meters long, an error of a few centimeters would be negligible; but if the links were more around half a meter long, the same error would be more significant. However, finding the solution to this problem is beyond the scope of this paper.

**Conclusion**

 The original aim of this exploration was to be able to model a robot arm in a way that would allow for the control of the end effecter by programming the joints of the arm to rotate at certain angular velocities. Therefore, I had to find the relationship between the linear velocity of the end effecter with respect to the angular velocities of the joints, and vice versa. Mathematically, this meant finding the angular velocities and  as functions of and, the desired linear velocities of the x- and y-coordinates of the end effecter. I accomplished this by first finding the spatial relationships between the end effecter position coordinates and the joint angles. Then, I applied a Jacobian matrix to produce the equations describing the relationship between these variables as they varied with time, i.e. the differential motion. My final equations were:



By finding these relationships, I gained mathematical knowledge that would be essential to the programming task. However, it is difficult to verify this knowledge for multiple reasons. For example, there is not currently a robot arm available for me to test on. More importantly, the angular velocities must be updated iteratively throughout the motion of the arm in order to keep the end effecter velocity constant, as the angular velocities depend on the joint angles—which vary as the arm moves. A clear extension of this exploration would be to investigate the time step needed for each iteration in order to maintain a certain degree of accuracy (which would depend on the scale of the robot and the arm) for the end effecter velocity.

Another possible extension for this topic would be to consider the movement of the robot arm in three dimensions, as in my investigation I only considered the x- and y-dimensions. Additionally, one could also consider a robot arm that has more than two joints. This would certainly add to the complexity of the problem, and involve more advanced mathematical methods such as fuzzy logic and approximating the “joint motions with a Taylor series expansion” (MathWorks; Lenarčič and Wenger 204).

Bibliography

"Modeling Inverse Kinematics in a Robotic Arm." *MathWorks Documentation Center*. The Mathworks, Inc., n.d. Web. 27 Feb. 2014.

Asada, H. Harry. *Chapter 5: Differential Motion*. Cambridge: Massachusetts Institute of Technology, n.d. PDF.

Dawkins, Paul. "Pauls Online Notes : Calculus III - Differentials." *Pauls Online Notes : Calculus III - Differentials*. Lamar University, n.d. Web. 17 Dec. 2013.

Dawkins, Paul. "Pauls Online Notes : Calculus III - Partial Derivatives." *Pauls Online Notes : Calculus III - Partial Derivatives*. Lamar University, n.d. Web. 17 Dec. 2013.

Lenarčič, J., and Philippe Wenger. *Advances in Robot Kinematics: Analysis and Design*. New York: Springer, 2008. Print.