What is the robot:

The delta robot, invented in the early 1980s, is a parallel robot consisting of three arms connected to universal joints at the base of the end effector. It utilizes parallelograms in the arms in order to maintain the orientation of the end effector in an X, Y, and Z direction with no rotation. The original purpose of this robot was to manipulate light and small objects at high speeds but has now become a very popular design in the 3D printer realm. Other uses of this robot can be found in the packaging industry and medical and pharmaceutical industries.

What were we tasked to do:

Develop the forward and inverse position and velocity kinematic equations for the Delta Robot.

Develop trajectory planning for the Delta Robot.

Develop modeling/simulation to test the kinematics and trajectory planning developed for the Delta Robot.

Develop C code that will successfully control the actual Delta Robot using the developed kinematics and trajectory planning.

Our goals vs. Kevin Harrington's goals:

Our Goals:

Design forward/inverse position and velocity kinematic equations and trajectory planning for the delta robot and successfully model this design in Matlab to prove the design is correct. Develop C code from our control design to successfully control the Delta Robot.

Kevin Harrington's Goals:

Develop forward/inverse position and velocity kinematic equations and implement these in C code such that the Delta Robot can be controlled successfully.

Delta Robot Kinematics:

Determining the kinematics of the delta robot was done by reasonably simple geometry. First, the robot columns need to be simplified so that the six columns that the robot carriages traveled on vertically (two for each carriage) were instead geometrical interpreted as 3 columns, one for each carriage. The three individual columns, that we will call A, B, and C, were geometrically positioned at the center of each of the actual two column pairs. In **Figure 1**, the position of the "Column Point" shows where the simplified placement of a column is placed. This simplified geometry of the delta robot greatly reduces the complexity of the kinematic analysis and provides a much simpler way of thinking about the movement of the robot. Some assumptions were made about the newly defined columns, such that they create an equilateral triangle so that the math and the placement on the coordinate system can be simplified.

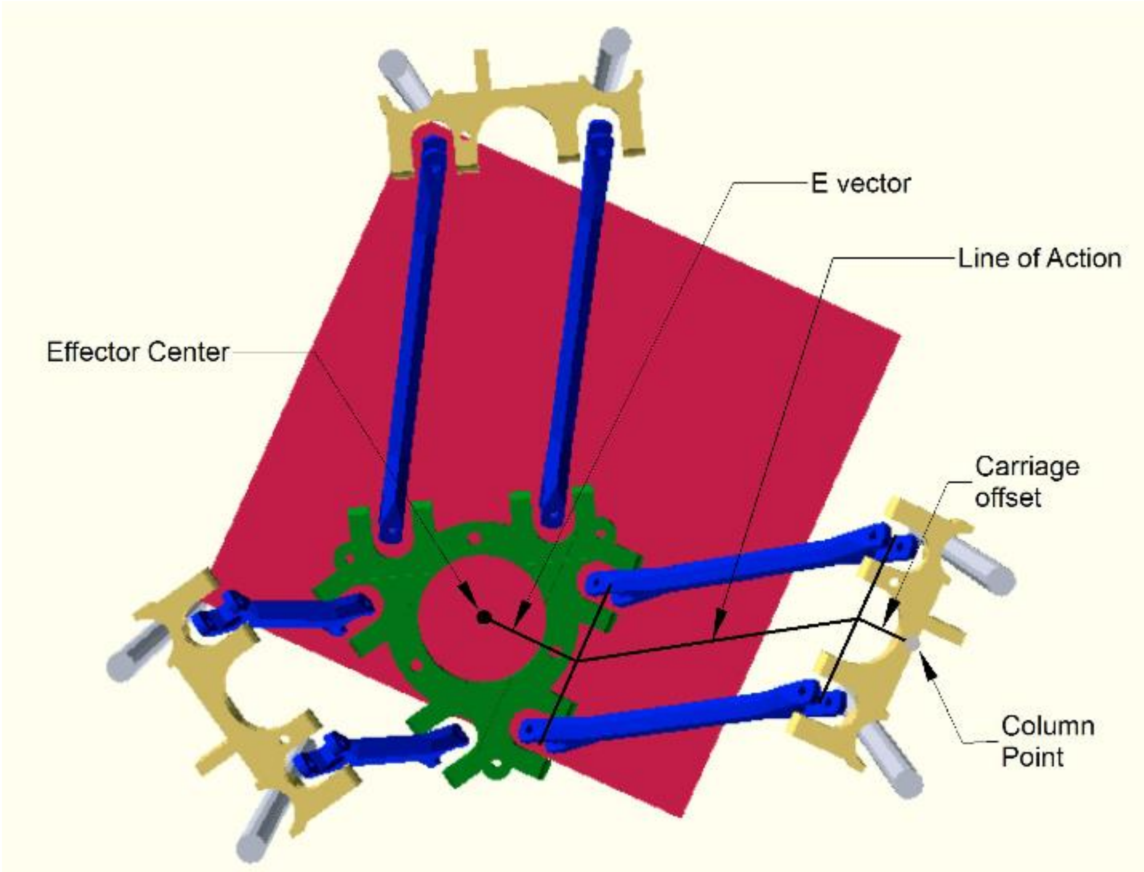

Figure 1 – Top view of Delta Kinematic Robot Model

The heights of each carriage must be realized and defined such that the height of the carriage above the end effector platform (Az, Bz, Cz) is realized as well as the distance the head extends below the end effector platform (Hcz) and the height that the head is located above the bed (z) (see Figure 2). The height of the head above the bed is essentially the distance between the location of the head and the lowest point to which the head can extend, also known as the 'bed'. At this point, height equations are able to be derived using the aforementioned parameters to describe the vertical movement of the carriages and the head.

Az = Z + Acz + Hcz
Bz = Z + Bcz + Hcz
Cz = Z + Ccz + Hcz

Solving for Z
Z = Az - Acz - Hcz
Z = Bz - Bcz - Hcz

Z = Cz - Ccz – Hcz

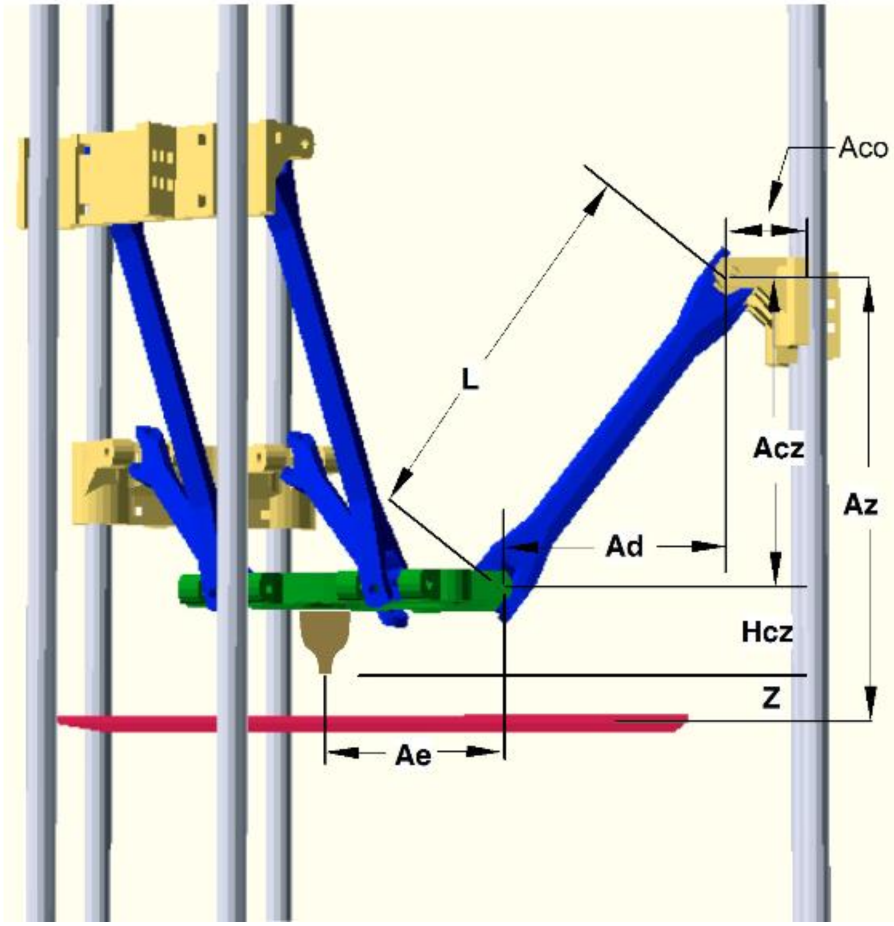

Figure 2 – Side view of Delta Robot with Designated Dimensions

An understanding of how the end effector of the robot will move with respect to a coordinate system is needed next. It is assumed that the middle point of the effector platform, the head, is located directly in the middle of the effector platform (see Figure 3, Center of the green platform). The distance between the edge of the platform and the center of the head is designated as (Ae, Be, Ce), the distance from the edge of the platform to the column bracket that holds the robot arm designated as (Ad, Bd, Cd), and the distance between where the arm and the bracket meet and the columns (A, B, or C) is (Aco), which gives us the total distance between a column and the head, as (e.g. Ad + Ae + Aco). Then by designating the length of each arm connected to the platform as (L) we are able to realize a simplex set of formulas utilizing the Pythagorean Theorem (see Figure 2 for a visual understanding of the aforementioned distance designations).

Ad^2 + Acz^2 = L^2
Bd^2 + Bcz^2 = L^2

Cd^2 + Ccz^2 = L^2

Next, the coordinate system is defined for the robot such that it is located on a cartesian plane where the A column is located on the Y axis of the coordinate system and the B and C columns are located 120 degrees in each direction from the A column (see Figure 3). Since the assumption was made that the columns are equidistant from each other, this simplifies the design such that each column (A, B, C) is equal in distance from the origin of the coordinate system.

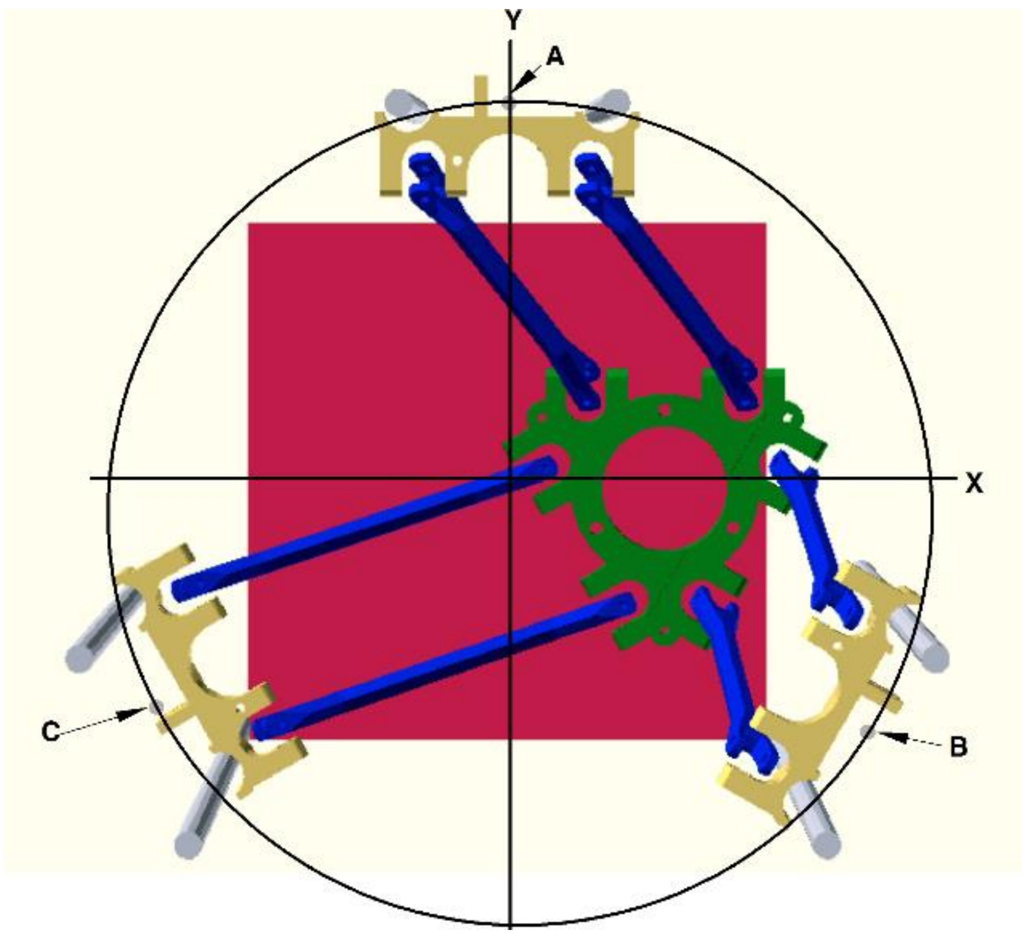

Figure 3 - Top view of Delta Kinematic Robot Model in Cartesian Coordinate System

A physical location of the columns in the X Y plane is then designated by (Ax, Ay, Bx, By, Cx, Cy). Observation shows that the pivot points are the points exactly below where each line of action terminates on its carriage and these can be calculated by subtracting a vector from each column that represents the carriage offset. These pivot locations are designated as (Apx, Apy, Bpx, Bpy, Cpx, Cpy). The coordinates where the lines of action meet the effector platform are designated as (Acx, Acy, Bcx, Bcy, Ccx, Ccy) and the (Ae, Be, Ce) vectors are able to be broken down and designated as (Aex, Aey, Bex, Bey, Cex, Cey) because we can represent a vector by a delta x and delta y. At this point equations can be derived to relate (Ae, Be, and Ce) to the X Y plane, while holding (z) constant.

$$X = Acx - Aex = Bcx - Bex = Ccx - Cex$$
$$Y = Acy - Aey = Bcy - Bey = Ccy - Cey$$

Furthermore, we are able to solve the equations for one column and determine the other two by similarity using the formula for a circle:

$$(X - CX)^2 + (Y - CY)^2 = CR^2$$

where Cx and Cy is the center of the circle and Cr is the radius. Such that when solving for the A column's Acx and Acy the following is obtained:

$$Acx = X - Aex$$
$$Acy = Y - Aey$$

Recognizing that the pivot points of the circle can be described as (Apx – Aex, Apy – Aey) the points can be designated as (Avx, Avy) and defined by the following set of equations:

$$Avx = Apx - Aex$$
$$Avy = Apy - Aey$$
$$Bvx = Bpx - Bex$$
$$Bvy = Bpy - Bey$$
$$Cvx = Cpx - Cex$$
$$Cvy = Cpy - Cey$$

Next the carriage height above the effector platform (Acz, Bcz, Ccz) (see Figure 2) can be calculated with the following set of equations providing the **inverse kinematics and forward kinematics** of the Delta Robot:

Inverse Kinematics:

Carriage height above the effector platform

$$Acz = \mathrm{sqrt}(L^2 - (X - Avx)^2 - (Y - Avy)^2)$$
$$Bcz = \mathrm{sqrt}(L^2 - (X - Bvx)^2 - (Y - Bvy)^2)$$
$$Ccz = \mathrm{sqrt}(L^2 - (X - Cvx)^2 - (Y - Cvy)^2)$$

Carriage Height above the Bed related to (z)

$$Az = Z + Acz + Hcz$$
$$Bz = Z + Bcz + Hcz$$
$$Cz = Z + Ccz + Hcz$$

Forward Kinematics:

$$(X - Avx)^2 + (Y - Avy)^2 + Acz^2 = L^2$$
$$(X - Bvx)^2 + (Y - Bvy)^2 + Bcz^2 = L^2$$
$$(X - Cvx)^2 + (Y - Cvy)^2 + Ccz^2 = L^2$$

Carriage Height above the Plane of the effector platform related to (z):

$Acz = Az - Z - Hcz$
$Bcz = Bz - Z - Hcz$
$Ccz = Cz - Z – Hcz$

Once the forward and inverse kinematics were determined the technique used to derive the forward velocity for the Delta Robot was to take the Jacobian of the Position Kinematics in order to create a Jacobian matrix that aids in deriving the forward velocity of the end effector. Since the Inverse Velocity Technique makes use of the Jacobian to determine the forward velocity of the Delta Robot, the Inverse Velocity was a straight forward technique known as the Derivation of the Pseudoinverse (Moore-Penrose Inverse). The reason this technique was used as opposed to simply taking the inverse of the Jacobian and multiplying by the velocity matrix of the end effector is because in our case we had a greater number of rows than columns (e.g. n>m) in our matrix.

Conclusion:

Our project development was planned and based in accordance with the material covered in RBE 501, Robot Dynamics. At the start of our project we had already been introduced to the position kinematics development approach of a serial robot. Since the Delta Robot is of a parallel design, we needed to expand on what we learned in the course in order to develop the position kinematics. To guide us, we relied on documentation of engineers/technicians who have worked with similar robots and who successfully developed parallel robot control design of this kind, both in a modeling environment and in real world practice. Developing the position kinematics was a large hurdle for our project and once this was accomplished we were able to rely primarily on the course material to guide us with the following design aspects of our project, such as the velocity kinematics and the trajectory planning. Due to the way the course was structured, it helped our team to tackle individual obstacles of our project one at a time which helped us from becoming overwhelmed by the size and complexity of the complete design needed in order to successfully accomplish our goals. In correlation of the design phase, team members worked to implement the design in matlab as a way of ensuring the robot control was working correctly. As always, modeling a complex system provided insight into areas of the design that needed to be “tweaked” and saved us time when it came to implementation. Modeling proved to be especially helpfully to our team as we are all remote students and did not have real access to the Delta Robot. Once modeling had proven our design successful, we were able to port the Matlab code over to C code, which did require a decent amount of effort from the more savvy coders on the team. This C code was then able to used to successfully control the microcontroller of the Delta Robot, and in turn, the robot manipulator.

This project provided us with challenges that spanned from design and implementation of the control of the Delta Robot to the challenges presented when working with team members from different backgrounds, skill sets, and even something as simple as personal scheduling conflicts. These conflicts are always present when a team of engineers work on any program in the real world and we are proud to say that we were able to overcome these challenges and finish successfully as a team of engineers should in any program.

NOTE: We must reference the following!!!!!

Johann C. Rocholl (Rostock) Style Delta Robot Kinematics

by Steve Graves

Also based on multiple existing sources:

http://en.wikipedia.org/wiki/Delta_robot

// modified from https://gist.github.com/kastner/5279172

//http://www.cutting.lv/fileadmin/user_upload/lindeltakins.c

//http://blog.machinekit.io/2013/07/linear-delta-kinematics.html