APSC 496

Critical Function Prototype Report

Praxim - Surgical Robot

Submitted on: 30-NOV-2009

Submitted by: Nicholas Adams

Davy Chiu

Ibrahim Gadala

David Mountford

Erica Wodzak

# 1.0 Introduction

# Once the critical function of allowing the user to move the device freely in 3D space was determined, steps were taken to test two prototypes that were being considered. These tests consisted of a basic experiment that tested each prototype's range of motion and a Matlab simulation that also investigated the range of motion of each prototype.

# Although there is still need for optimization, it appears that prototype 2 is the more feasible option.

# 2.0 Critical Function

The critical function evaluated is this document is the “cutting tool moves” function identified in the functional decomposition. The ability for the user to move the cutting tool in 3D space is essential to the ultimate project goal of generating a 3D hard surface and interacting with other functions that have significant impact on the design – these include the mechanism used to implement and move the hard surface, how the device is secured to the knee and the types of shapes that can be created. This function also has a direct impact on many of the design requirements. The mechanism selected will set limitations on the size and weight of the device due to the type of joints used in the design, as well as motor speed and accuracy requirements and how the user interacts with the device. As a result, it is important that these effects are considered early, before the interfacing functions are developed further.

From the analysis completed in this report we plan to finalize the mechanism concept that allows the cutting tool to move in 3D space and assess the impact of link lengths on the envelope of the device.

# 3.0 Description of Prototype

The prototype developed to assess the critical function is based on a previous prototype used to evaluate the feasibility of the entire concept in 2009. This existing prototype implements a three link mechanism consisting of two rotational joints and one linear joint. The drawback with this link arrangement is the size and weight of the linear joint, and as a result linear joints will not be considered for this mechanism.

The portion of the existing prototype used for the critical function prototype is shown in . The motor and encoders were removed from linkage 1 and linkage 2 leaving only the mechanism developed by Nikolai Hungr. The ideal lengths of linkage 1 and linkage 2 are 5 cm and 4.5 cm respectively, based on Hungr’s analysis. The critical function prototype developed, however, uses link lengths of 5.5 cm and 6.0 cm. This combination of link lengths will be used to evaluate the potential operating enveloped of the device, but further analysis is necessary before these lengths are finalized.

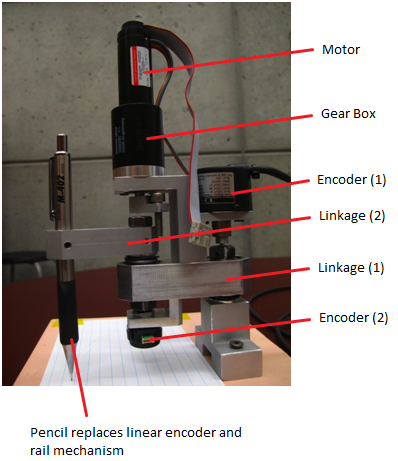


Figure – Dynamic Physical Constraint mechanism used in existing prototype

## Prototype 1

Prototype 1 (PT1) consists of three rotational joints at positions 1, 2 and 4. Bearings have been used at joints 1 and 2 while link 4 is allowed to rotate by a set screw held in place by a setscrew. Link 3 is a kept stationary by a setscrew and is not allowed to rotate. The link 3 has been designed to be repositioned depending on the experimental requirements. Link 4 has also been designed to vary in length from 3 cm to 6 cm in increments of 1 cm, but will be set at 6 cm for Prototype two. Figure 2 provides an overview the Prototype 1 mechanism. The figure highlights the relationship between Link 3 and Link 2 and the position of Link 4 and the cutting tool because they cannot rotate with respect to each other.

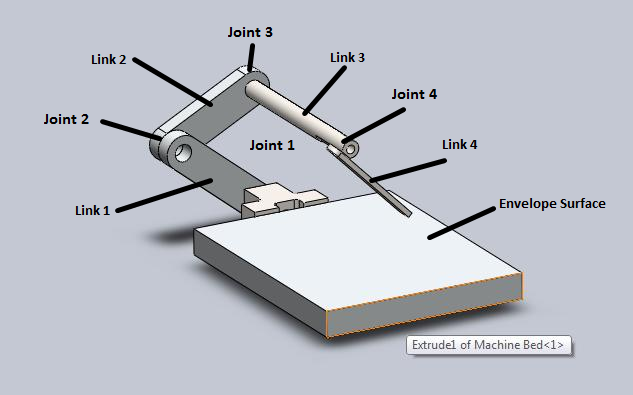


Figure – Model of prototype 1

Table – Prototype 1 link lengths

|  |  |
| --- | --- |
| **Link** | **Length [cm]** |
| Link 1 | 5.5 |
| Link 2 | 6.0 |
| Link 3 | Set to 10 |
| Link 4 | 6.0 |

## Prototype 2

Prototype 2 (PT2) consists of four rotational joints at positions 1, 2, 3 and 4. Bearings have been used at joints 1 and 2 while link 4 is allowed to rotate by a set screw held in place by a setscrew. Link 3 is a kept allowed to rotate by loosening the setscrew but a stopper is placed at the end to stop translation. Link 4 will be set at 6 cm for Prototype 2. In model of Prototype 2 – shown in Figure 3 – Link 3 can rotate with respect to Link 2 and Link 4 can always be oriented towards the envelope surface as a result.

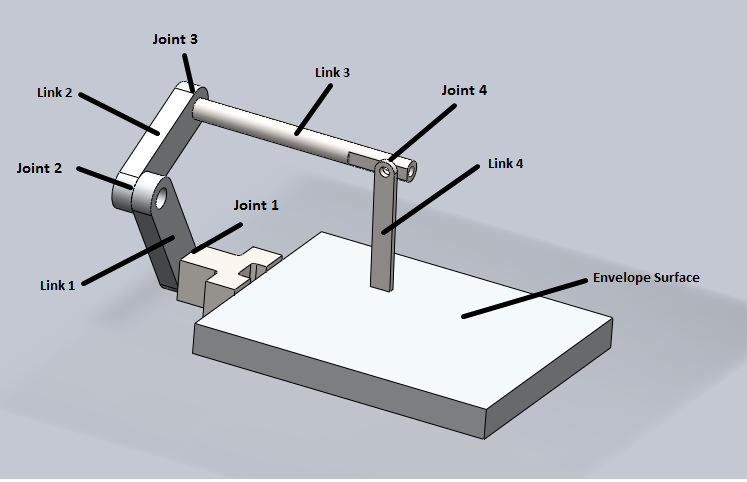


Figure – Model of Prototype 2

Table – Prototype 2 link lengths

|  |  |
| --- | --- |
| **Link** | **Length [cm]** |
| Link 1 | 5.5 |
| Link 2 | 6.0 |
| Link 3 | Set to 10 |
| Link 4 | 6.0 |

# 4.0 Experimental Method

The following experiment was formulated to examine two aspects of both prototypes being considered. The first aspect to be examined was the size of the workable machining envelope to see if it would be sufficient. The second aspect was to determine if there were any locations that the tool would be unable to reach.

**Testing Protocol**

The purpose of the testing protocol is to see if the desired range of motion is achieved with the rotation-rotation-rotation joint configuration design within the specified requirements. In order to complete Unicompartmental Knee Replacement Surgery the tool must have access to both femoral lobes. This can be achieved in two ways; (1) a linkage mechanism with a range of approximately 16 cm - equivalent to the entire knee - or (2) a linkage mechanism in which linkage 3 is adjustable to allow operation on individual lobes using the same mounting position.

**Range of Workable Area**

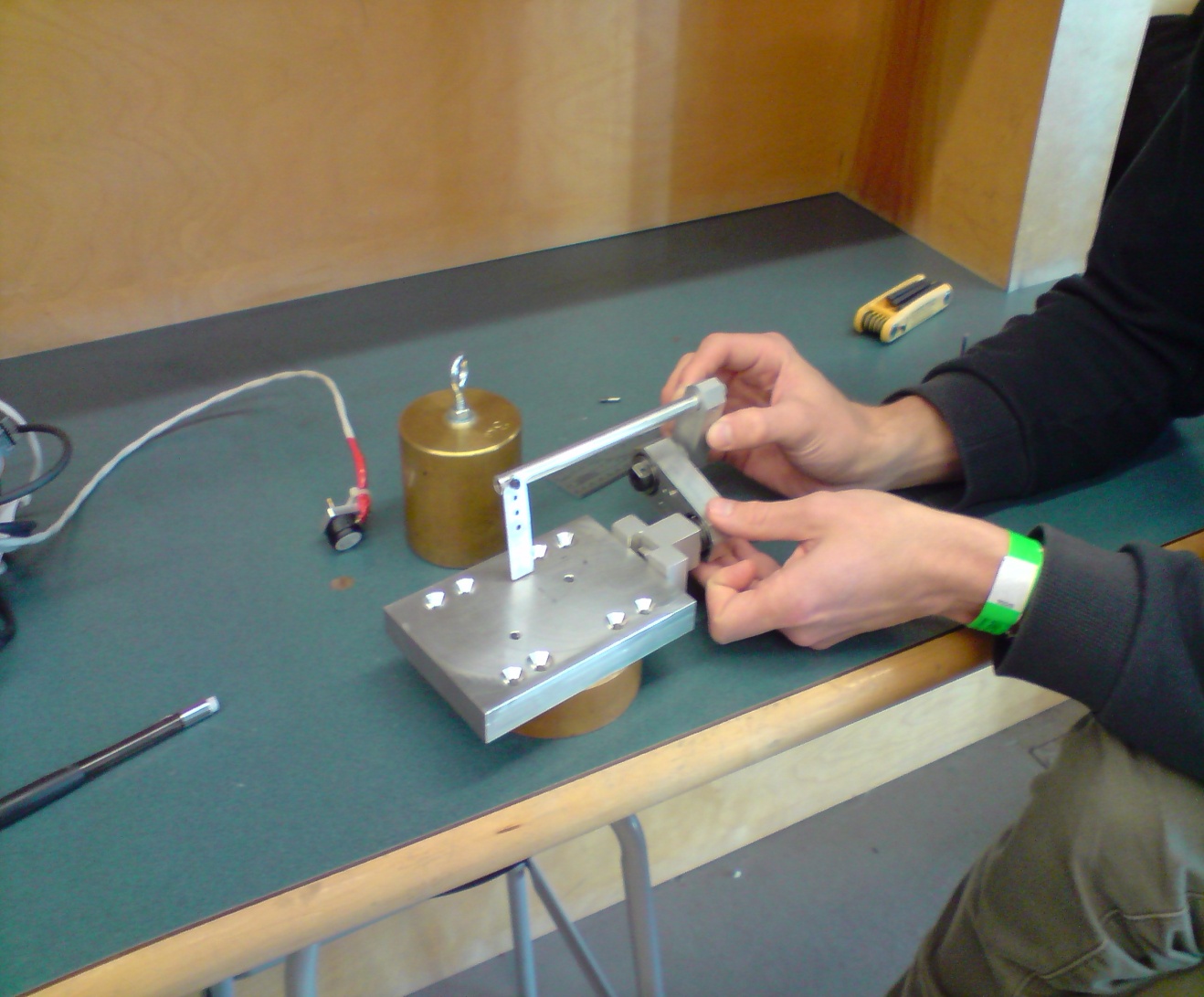
This test is to determine the feasibility of two design types; (PT1) four link mechanism with rotational joints at positions 1, 2 and 4 and (PT2) four link mechanism with rotational joints and positions 1, 2, 3 and 4. Steps will be taken to access the effectiveness of both designs with respect to the entire knee from one position design and the individual lobes design discussed above. Figure 4 shows the test set up.

Figure - Test Setup

Assumptions/Requirements:

1. Device is rigidly fixed to femur (no play in the bone mount)
2. Link 2 will never be perpendicular to the generated surface
3. Link 4 has a maximum angle of 45 degrees from the vertical position
4. Link 4 cannot come into contact with the support structure
5. Entire knee envelope has a radius of 8cm
6. Single lobe envelope must be 16cm by 8cm
7. Lateral deflection cannot occur
8. Linkages cannot transfer between right and left orientation
9. Link 4 length is 6cm

Method:

1. Setup up mechanism as PT1
2. Position mechanism in right orientation
3. Set reference point to the position where link 4 is perpendicular to surface and in line with the axis of the mechanism
4. Position tool (end of link 4) along the surface of the support structure at a edge of the envelope – on the verge of lateral deflection
5. Moving the tool along the surface and maintaining near lateral deflection position, map out the envelope of the current setup moving link 4 from +45 degree to -45 degree position
6. Maintain -45 degree position keeping tool on the surface, map out the envelope generated until link 1 is perpendicular to the surface
7. Moving the tool along the surface and maintaining link 1 perpendicular to surface position, map out the envelope of the current setup moving link 4 from -45 degree to +45 degree position
8. Maintain +45 degree position keeping tool on the surface, map out the envelope generated until mechanism reaches the lateral deflection position
9. Repeat for all steps for PT2

**MATLAB Simulation**

In order to have a more visual representation of the workable area, a Matlab script was written to plot the possible motion of both concepts. The script generates points at various combinations of link angles and plots these points out to create an envelope volume. The script, however does not take later deflection into account and no final conclusions can be drawn from the generated envelope. Later in the design process the script can be adapted to optimize link lengths and find the optimal range of angular deflection for each link.

# 5.0 Analysis of Results

After completing a physical prototype of the robot’s new linkage system, the critical function was preliminarily examined with a series of basic tests as described in the previous section. The main feature we tested for is the range of workable area the linkage allows for.

Prototype 1:

At first, our shaft link (3rd link from the bone mount) simply acted as an offset for the system, as it was rigidly fixed against movement with a set screw in its joint with the second link (PT1). All three other links were allowed to rotate in their respective planes. Although this set-up provides an easier task for hard-constraint calculations and control, since you need 1 less encoder and a less intensive mathematical calculation, the workable range provided is greatly lacking. Effectively, with the link sizes of the prototype we tested (15 cm shaft and 6 cm tool link), the area was a triangle with a height of merely 2 cm and a base length of about 3 cm, as seen in Figure 5. Errors in this analysis include possible slippage in the set screw fixation, flexibility in the link structure (final prototype will be designed more for sturdiness and rigidity), and the inconsideration of the tool bit orientation and size. For the latter, it was assumed that the point of contact of the final, 4th link with the surface will provide sufficient tool penetration for cutting, and that the tool was oriented at an angle to promote better machining conditions. Thus, it could be stated that the resulting limitation of the final link's motion would not allow the prototype to touch the work surface; resulting in poor bone cutting conditions.



Figure - Range of Motion for PT1

Prototype 2:

We decided to test whether the range from the previous set-up would be improved if the 3rd shaft link was allowed to rotate about its axis (PT2). The change was immediate and obvious. The rotation of the shaft link effectively allowed the projection of its rotation angle to be added to the range of the workable area, thus greatly increasing it. With the same link sizes used for the first testing set-up, this added degree-of-freedom to the third link allowed us to work on a rectangle with side lengths of 9 cm and 5 cm as seen in Figure 6. That being said though, implementing this extra DOF requires an additional encoder and a more intensive mathematical and computing process for hard surface control. Inherently, this analysis has less possible errors than the analysis of prototype 1, since slippage of the shaft link is allowed, and the tool is allowed to be oriented at many more angles that with the shaft secured in place. However, errors in this analysis still arise from linkage system flexibility and possible slippage of the shaft link *axially*, which would not occur in a real-life scenario. Thus, the main assumption for the analysis of prototype 2 is that, while taking work area measurements, the shaft did not slip axially as to provide a larger final work area. This means that a major limitation of this analysis is that the work area measured is the upper value of the range possible, since the shaft could not have possibly stayed fixed at the joint, yet we assume it did so we make a obtain a less conservative result. In essence, the undisputed result of this analysis is that prototype 2 has a significantly larger work area than prototype 1.

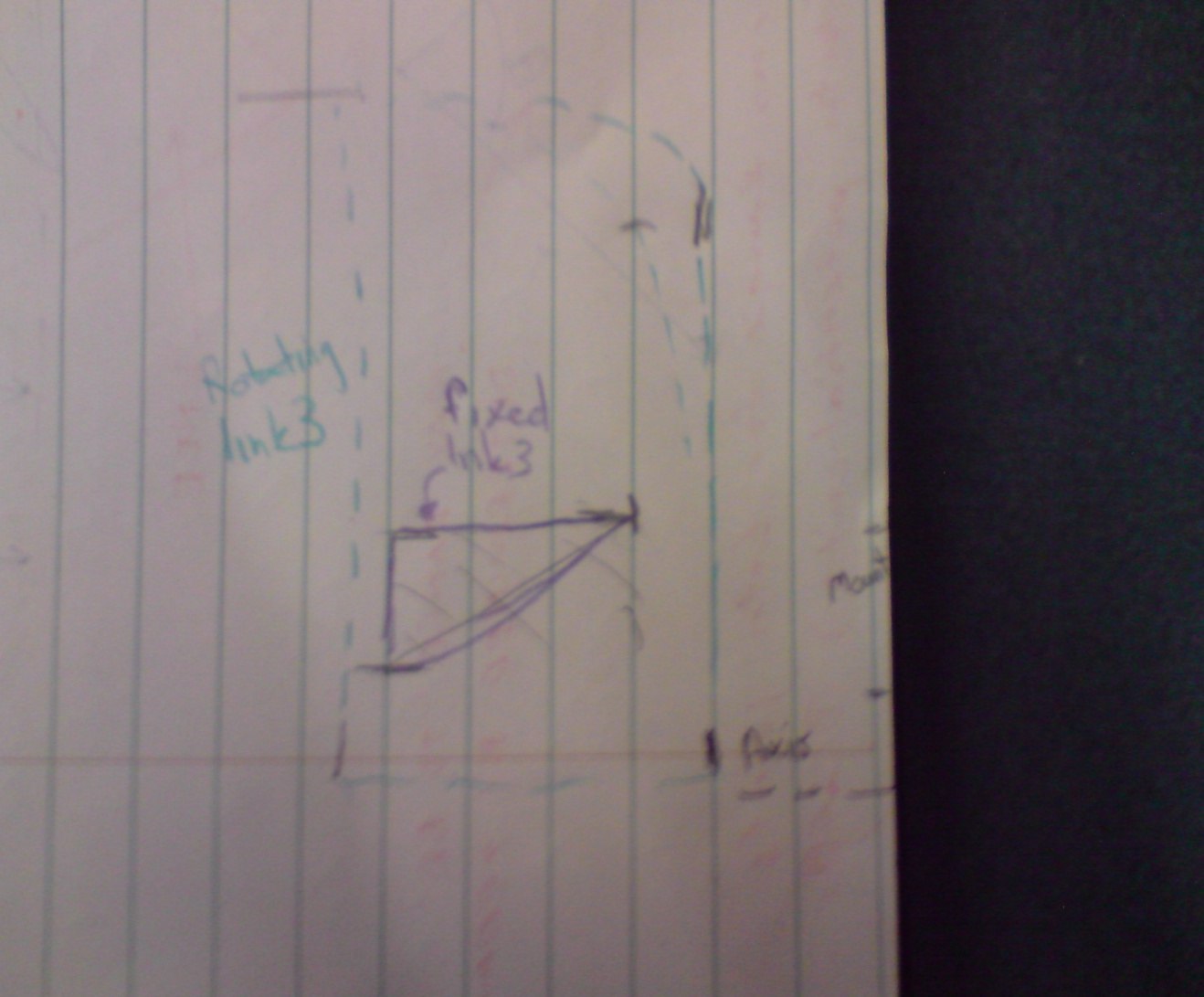


Figure - Range of Motion for PT1 and PT2

**MATLAB Simulation Results**

The MATLAB simulation has been used to verify the results generated from PT1 and PT2 and provide an estimate of each prototypes workable machining envelope.

Prototype 1:

Figure 7shows the enveloped generated for PT1 using the link lengths from Table 1. In the MATLAB script Link 4 has been offset by 30 degrees resulting in the orientation shown in Figure 7. In the x-y plane, horizontal to the envelope surface , PT1 has a range of approximately 10 cm by 15 cm. These results are somewhat deceiving as there are many locations within this volume that cannot be reached and the cutting tool is constrained to specific paths when trying to move in a single plane. This restricted motion can be seen by the significant holes in the generated operation space. Figure 7 also suggests that the vertical envelope is reduced to the point that the user will not have freedom of motion away from the surface in some areas.

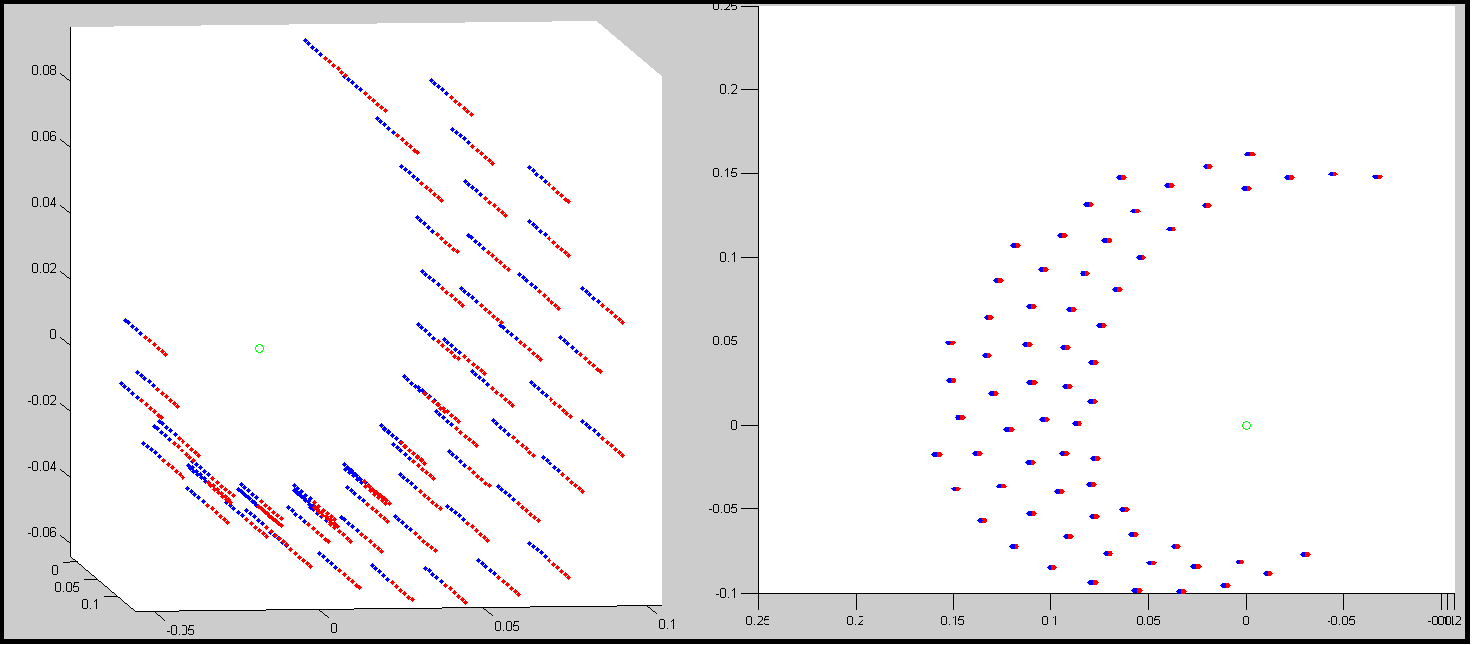
****

Figure – Prototype 1 MATLAB range of motion simulation

Prototype 2:

The analysis of PT2 has been conducted using the link lengths described in and the generated spatial plot is shown in . The x-y plane enveloped created by PT2 is slightly larger than PT1 at approximately 18 cm by 10 cm. More significantly the envelope plot of PT2 does not have the same vertical restrictions. The result indicates that PT2 in its current setup can have a vertical range of approximately 7cm while maintain a 15 cm by 10 cm envelope. The user will have far more freedom away from the imposed constraint and a greater variety of hard constraints can be imposed as a result.

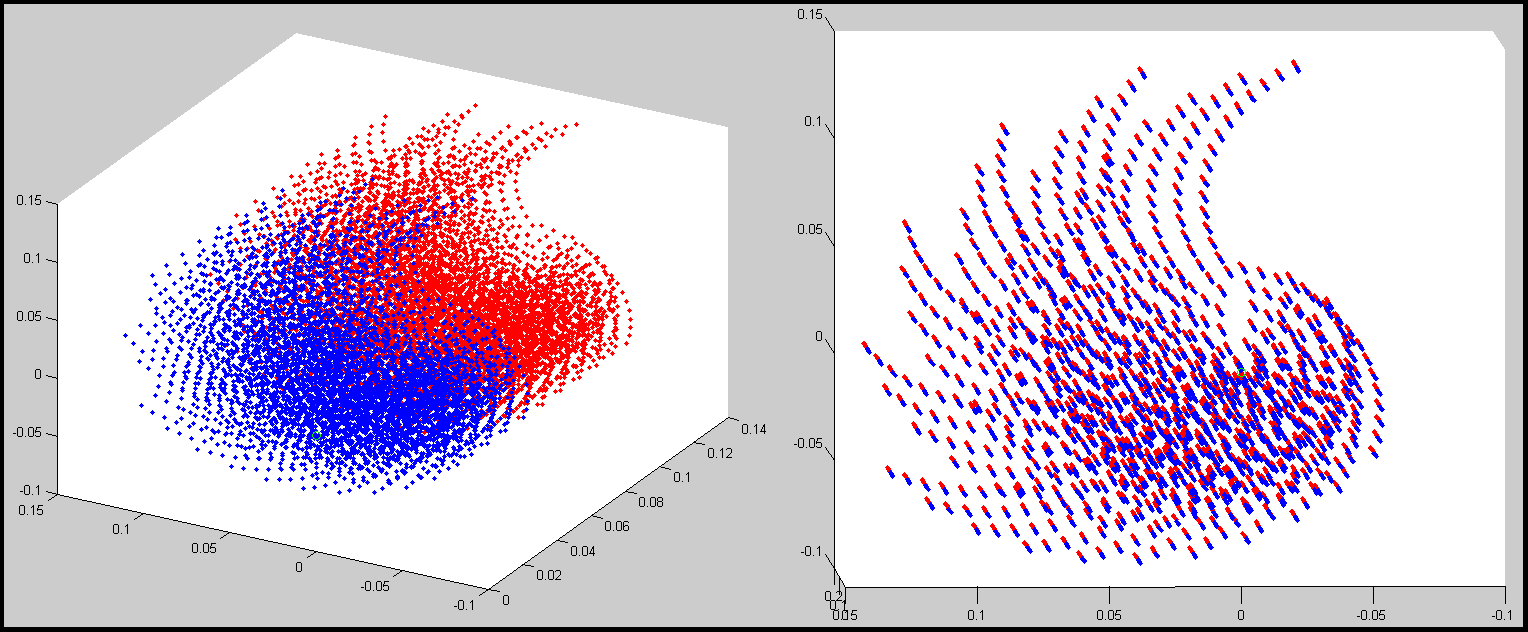


Figure – Prototype 2 MATLAB range of motion simulation

Table 3 summarizes the results for both the experimental and Matlab simulated results.

Table – Summary of critical function prototype analysis

|  |  |  |
| --- | --- | --- |
| **Test** | **Prototype 1** | **Prototype 2** |
| Physical test envelope in x [cm] | 8 | 8 |
| Physical test envelope in y [cm] | 6 | 15 |
| MATLAB max x envelope [cm] | 10 | 10 |
| MATLAB max y envelope [cm] | 7 | 18 |
| MATLAB usable z envelope [cm] | 2 | 7 |

# 6.0 Future Steps and Updated Timeline

A better understanding of the geometry and the installation of unicompartmental knee implants will be required in order to determine if the device's workable area is sufficient to do both implants with one setup, or if the length of link 3 should be adjustable to allow setup for individual implants.

Also, in order to finalize the design of this concept we need to develop the concepts of the other functions that interact with this function further. Developing these concepts will aid in finalizing specifics of the this critical function.

A new timeline has been developed to better reflect the our current status and goals for the next month, as well as a rough timeline estimate for tasks further in the future. This updated timeline can be found in Appendix C.

# 7.0 Conclusion

The limited range of motion of prototype 1 appears to make it not a viable option, while the range of prototype 2 is promising, so it will be the concept that we will continue to be developed.

With this conclusion, the work plan has been updated to allow for research into the use and shape of intercompartmental knee implants, as well as the further development of concepts for other functions before finalizing the details of this concept.

# Appendix A - Supplementary Simulations

Appendix C includes MATLAB analysis of two additional mechanisms. After the analysis of Prototype 1 and 2 it was noted that PT1, as depicted in Figure 2, significantly limits the cutting tool range of motion, and that two hard constraints would be required to successfully implement PT2. Of the mechanisms presented here, Prototype 3 (PT3) is another mechanism that will be considered, and Prototype 4 (PT4) is a redesign version of PT1.

As with PT1 and PT2, these simulations have been developed to provide an approximate range of motion for each prototype and no optimization has been performed on the geometry of any of the prototypes. The lengths and angles used here are based on those chosen for PT1 and PT2 and the generated envelope sizes can be compared without additional analysis.

## Prototype 3:

Prototype 3, shown in Figure 9, implements the hard constraint at joint 2 that will set the minimum length of link 2. This prototype has examined using 2 different setups, PT3a and PT3b.

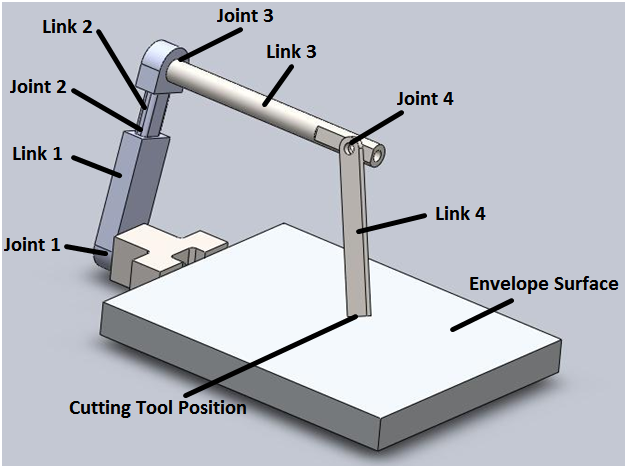


Figure – Model of Prototype 3

**PT3a Mechanism**

In PT3a link 1 is free to rotate about joint 1, while link 3 is held in place with respect to link 2 and joint 3. Link 4 can rotate about joint 4.

**PT3a Results**

The PT3a configuration yielded results similar to PT1. This is expected as PT3a replaces the rotational joint used to provide freedom in the vertical direction at joint 2 with a linear joint, without making any other changes to the mechanism. The approximate envelop for this mechanism and setup has been determined to be 10cm by 10cm by 2 cm – see Table 4 for a comparison of the envelopes of all prototypes.

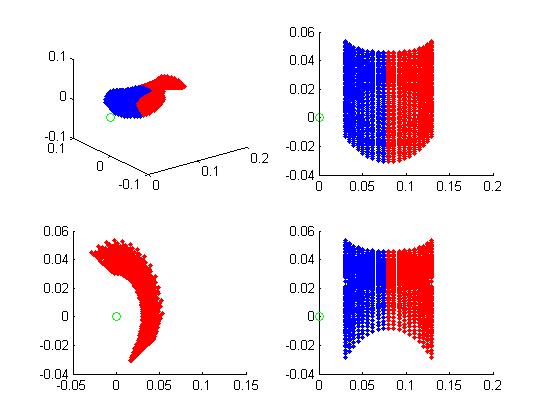


Figure – MATLAB plot of Prototype 3a envelope

**PT3b Mechanism**

For the PT3b mechanism link 1 can be positioned at setup, but is fixed during the surgery. As a result, for the PT3b MATLAB model link 1 is fixed about joint 1 while link 3 was free to rotate about joint 3 with. Link 4 can rotate about joint 4.

**PT3b Results**

Of the mechanisms that only require one hard constraint, the PT3b configuration appears to provide the largest range of motion and has significant vertical freedom.. The approximate envelop for this mechanism and setup has been determined to be 10cm by 8cm by 6cm.

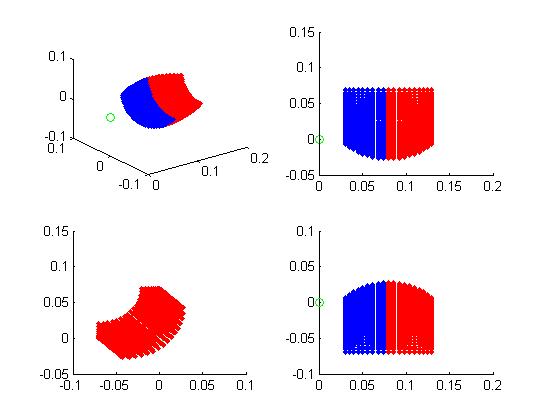


Figure – MATLAB plot of Prototype 3b envelope

## Prototype 4:

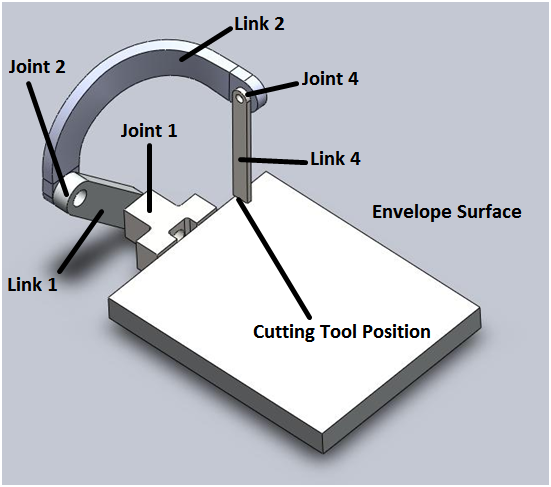
The PT4 mechanism repositions joint 4 6 cm higher than the position of joint 4 in PT1. This setup maximizes the possible range of motion of the tool position by maximizing the number of orientations in which link 4 will be approximately perpendicular to link 2 while link 1 is also perpendicular to link 2 – in effect maintaining the angle between links 1 and 2 as far away from any singularities as possible. As shown in Figure 12, link 3 has been completely removed and link 2 is connection to link 4 at joint 4. 

Figure – Model of Prototype 4

**PT4 Mechanism**

In PT4, the hard constraint is implemented at joint 2 and will maintain a minimum angle between link 1 and link 3. In general, link 1 is free to rotate about joint 1, link 2 in free to rotate about joint 2 while away from the hard constraint, and link 4 is free to rotate about joint 4.

**PT4 Results**

The simulated range of motion for this prototype is somewhat surprising. The area this prototype covers in the yz-plane is a very pointed arc, similar to the arc shown in Figure 7 for PT1, and does not provide the same vertical freedom seen in PT3b. The enveloped generated for PT4 is approximately 10cm by 10cm by 2cm. Further simulation could be run in the future to analyse the effect of modifying geometric parameters. The arrangement shown in Figure 12, does not seem to completely implement the desired 90 degree angle between link 1 and an imaginary link 2 (represented by PT1 link 2 in Figure 2), and this is likely to have had an effect on the results.

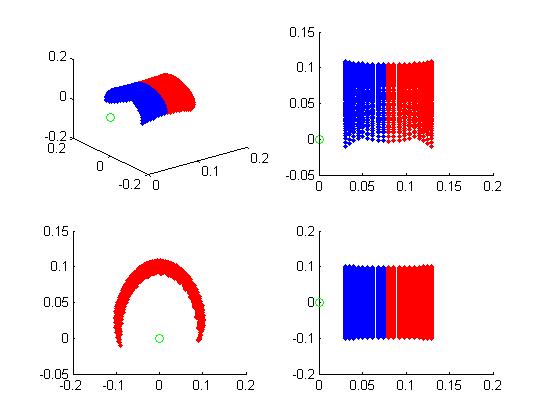


Figure – MATLAB plot of prototype 4 envelope

Table – Summary of Prototype 3 and 4 analysis

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test** | **PT1** | **PT2** | **PT3a** | **PT3b** | **PT4** |
| Physical test envelope in x [cm] | 8 | 8 | - | - | - |
| Physical test envelope in y [cm] | 6 | 15 | - | - | - |
| MATLAB max x envelope [cm] | 10 | 10 | 10 | 10 | 10 |
| MATLAB max y envelope [cm] | 7 | 18 | 10 | 8 | 10 |
| MATLAB usable z envelope [cm] | 2 | 7 | 2 | 6 | 2 |

# Appendix B - Linear hard constraint power screw/worm gear calculations

The following calculation has been preformed to obtain an understanding of the feasibility of implementing a power screw and worm gear system for hard constraint control in a linear displacement joint prototype. The main intended purpose of this calculation was to quantify the minimum motor speed needed to be able to adjust the hard surface constraint in sufficient time.

Main system parameters such as stroke, worm and worm gear pitches, and power screw pitch diameter, have been chosen based on the anticipated size of the linear displacement joint. The mass of the load is assumed to be 10 kg, based on the functional requirement of the device which states that it must resist, at minimum, a pushing force of 10 kg. The speed of the load (the pushing force in this case is assumed to be constant) is assumed to be 0.5 m/s, based on a distance of 0.5 mm (from functional requirements also) to be pushed in 990 microseconds (leaving the remaining time for sensing, transmission, and computation).

**Assumptions**

* Force on constraint = 10kg
* Cutting tool speed = 0.5 m/s
* Required cutting tool movement = 0.5mm
* Available time after position processing = 990 microseconds
* Number of thread on worm gear = 3

**Results**

The torque requirements of the gears and worm involved in the design were calculated, in order to properly spec out these components if needed. In general, the torque/force values are quite low, the majority of available gears will meet the requirement. The required speed of the motor was also found to be approximately 3000 rpm, assuming a triple thread (gain) worm. Although this value is within the realms of physical and mechanical possibility, it is a tough performance to achieve, especially in our high precision and immediate response application.

**Calculations**

Figure 14 and Figure 15 show scans of the calculations made for this analysis.

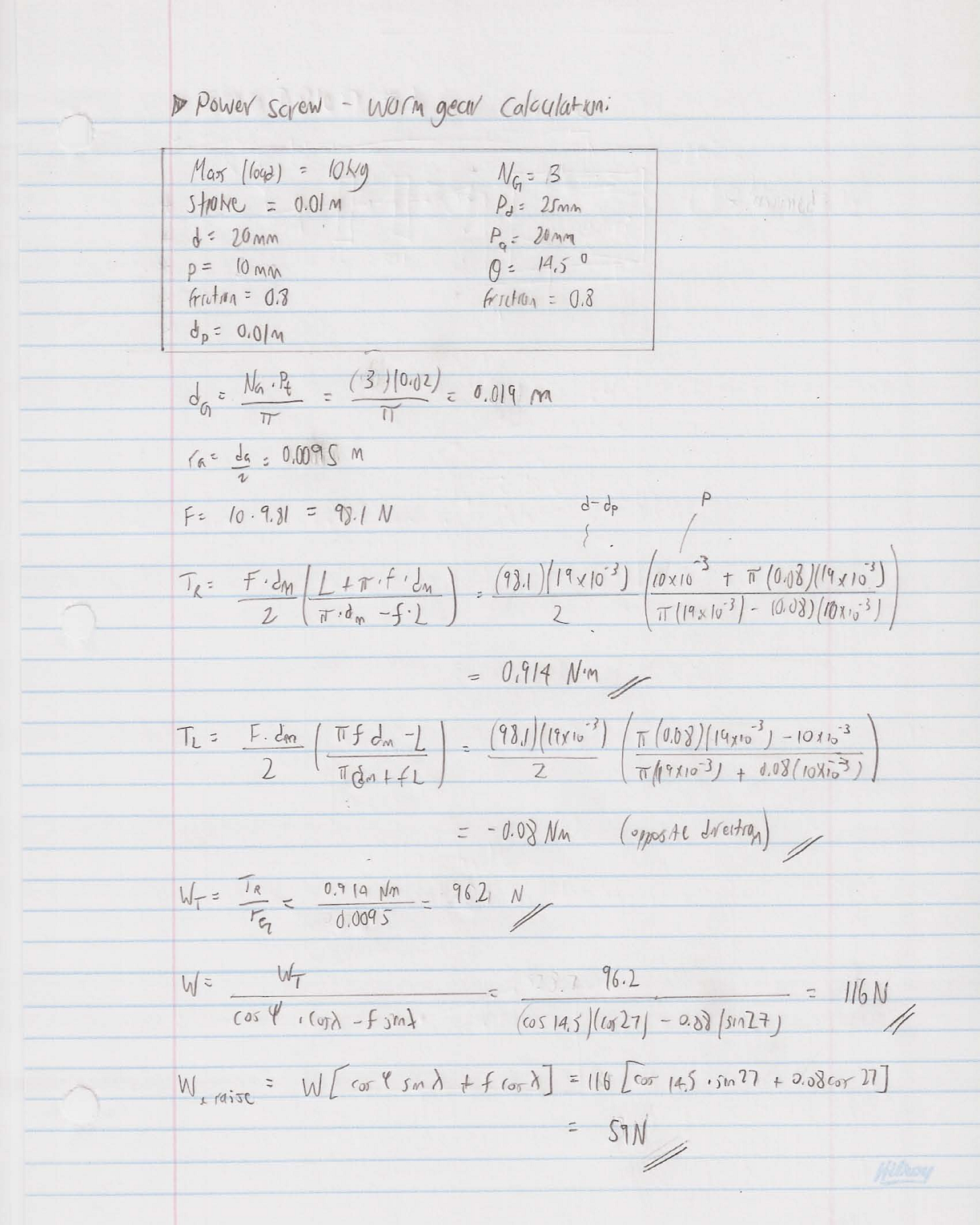


Figure – Power screw/worm gear calculation page 1

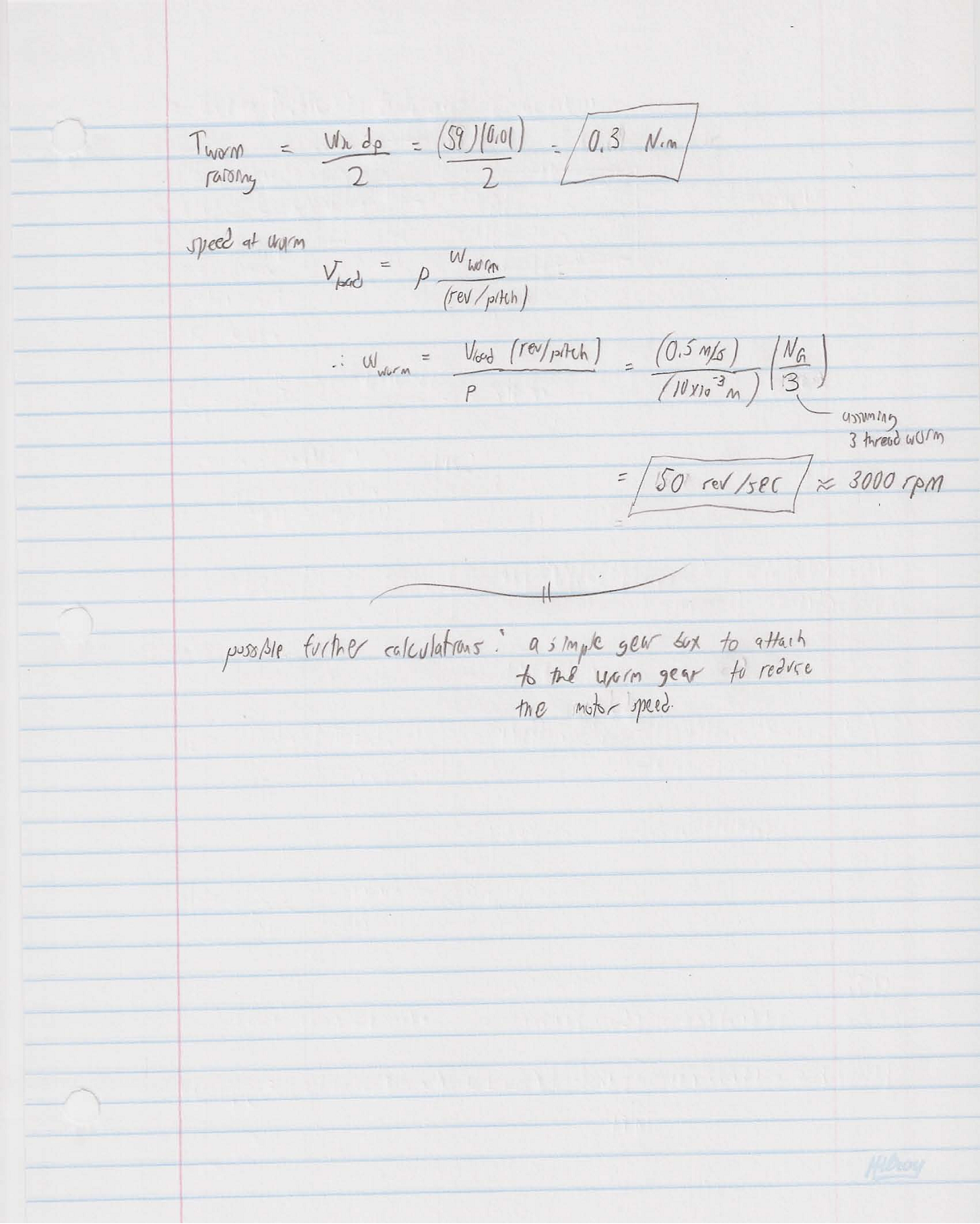


Figure – Power screw/worm gear calculation page 2

# Appendix C - MATLAB Code and Results for PT1

%Geometric Parameters

Length\_1 = 6e-2;

Ratio\_1to2 = 1.1;

Length\_3 = 8e-2;

Theta\_3\_max = 45;

Length\_4 = 7e-2;

Theta\_4\_max = 45;

Theta\_1\_max = 60;

Theta\_12\_min = 30;

Theta\_12\_max = 135;

Step = pi()/180;

%Calculated Lengths and Conversions to radians

Length\_2 = Length\_1 \* Ratio\_1to2;

Theta\_3\_max = Theta\_3\_max \* pi() / 180;

Theta\_4\_max = Theta\_4\_max \* pi() / 180;

Theta\_1\_max = Theta\_1\_max \* pi() / 180;

Theta\_1\_min = -Theta\_1\_max ;

Theta\_3\_min = -Theta\_3\_max ;

Theta\_12\_min = Theta\_12\_min \* pi() / 180 ;

Theta\_12\_max = Theta\_12\_max \* pi() / 180 ;

hold on

subplot(2,1,1), plot3(0,0,0,'go')

subplot(2,1,2), plot3(0,0,0,'go')

%Nested Loops

for Theta\_1 = Theta\_1\_max : -24\*Step : Theta\_1\_min

for Theta\_2 = Theta\_12\_min : 10.5\*Step : Theta\_12\_max

%for Theta\_3 = Theta\_3\_min : 6\*Step : Theta\_3\_max

Theta\_3 = -30;

for Theta\_4 = 0 : 9\*Step : Theta\_4\_max

%Tool Position

x\_position = Length\_3 - (Length\_4 \* sin(Theta\_4));

x\_position\_neg = Length\_3 + (Length\_4 \* sin(Theta\_4));

Theta\_offset = Theta\_2 + Theta\_1 - (pi()/2);

Theta\_3\_offset = Theta\_3 + Theta\_offset;

y\_position = Length\_2 \* cos(Theta\_offset) - Length\_1 \* sin(Theta\_1) + Length\_4 \* sin(Theta\_3\_offset);

z\_position = Length\_1 \* cos(Theta\_1) + Length\_2 \* sin(Theta\_offset) - Length\_4 \* cos(Theta\_3\_offset);

hold all

subplot(2,2,1), plot3(x\_position,y\_position,z\_position,'.'), view(3), hold on

subplot(2,2,1), plot3(x\_position\_neg,y\_position,z\_position,'r.'), view(3), hold on

subplot(2,2,2), plot3(x\_position,y\_position,z\_position,'.'), view(0,0), hold on

subplot(2,2,2), plot3(x\_position\_neg,y\_position,z\_position,'r.'), view(0,0), hold on

subplot(2,2,3), plot3(x\_position,y\_position,z\_position,'.'), view(90,0), hold on

subplot(2,2,3), plot3(x\_position\_neg,y\_position,z\_position,'r.'), view(90,0), hold on

subplot(2,2,4), plot3(x\_position,y\_position,z\_position,'.'), view(0,90), hold on

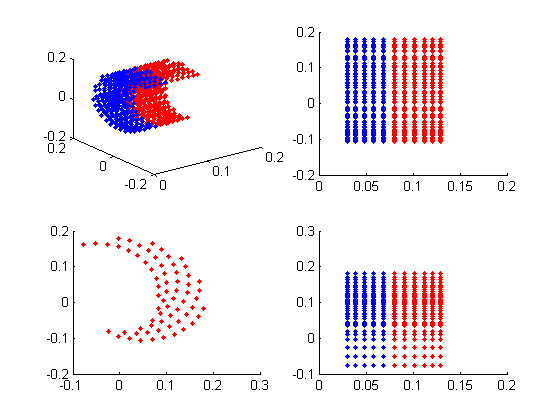
subplot(2,2,4), plot3(x\_position\_neg,y\_position,z\_position,'r.'), view(0,90), hold on

end

%end

end

end



%Geometric Parameters

Theta\_1\_max = 60 ;

Radius\_min = 6e-2 ;

Radius\_max = 9e-2 ;

Length\_3 = 8e-2 ;

Theta\_3 = 30 ;

Length\_4 = 7e-2 ;

Theta\_4\_max = 45 ;

Step = pi() / 180 ;

%Calculated Lengths and Conversions to radians

Theta\_1\_max = Theta\_1\_max \* pi() / 180 ;

Theta\_1\_min = -Theta\_1\_max ;

Theta\_3 = Theta\_3 \* pi() / 180 ;

Theta\_4\_max = Theta\_4\_max \* pi() / 180 ;

hold on

subplot(2,2,1), plot3(0,0,0,'go'), hold on

subplot(2,2,2), plot3(0,0,0,'go'), hold on

subplot(2,2,3), plot3(0,0,0,'go'), hold on

subplot(2,2,4), plot3(0,0,0,'go'), hold on

%Nested Loops

for Theta\_1 = Theta\_1\_max : -10\*Step : Theta\_1\_min

for Radius = Radius\_min : 3e-3 : Radius\_max

%for Theta\_3 = Theta\_3\_min : 6\*Step : Theta\_3\_max

for Theta\_4 = 0 : 5\*Step : Theta\_4\_max

%Tool Position

x\_position = Length\_3 - (Length\_4 \* sin(Theta\_4));

x\_position\_neg = Length\_3 + (Length\_4 \* sin(Theta\_4));

y\_position = Radius \* sin(Theta\_1) - Length\_4 \* cos(Theta\_4) \* sin(Theta\_1 - Theta\_3) ;

z\_position = Radius \* cos(Theta\_1) - Length\_4 \* cos(Theta\_4) \* cos(Theta\_1 - Theta\_3) ;

hold all

subplot(2,2,1), plot3(x\_position,y\_position,z\_position,'.'), view(3), hold on

subplot(2,2,1), plot3(x\_position\_neg,y\_position,z\_position,'r.'), view(3), hold on

subplot(2,2,2), plot3(x\_position,y\_position,z\_position,'.'), view(0,0), hold on

subplot(2,2,2), plot3(x\_position\_neg,y\_position,z\_position,'r.'), view(0,0), hold on

subplot(2,2,3), plot3(x\_position,y\_position,z\_position,'.'), view(90,0), hold on

subplot(2,2,3), plot3(x\_position\_neg,y\_position,z\_position,'r.'), view(90,0), hold on

subplot(2,2,4), plot3(x\_position,y\_position,z\_position,'.'), view(0,90), hold on

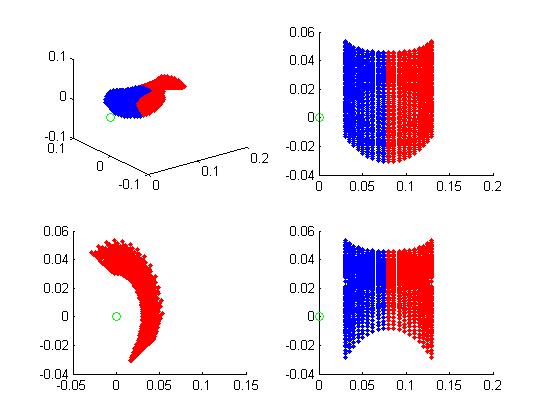
subplot(2,2,4), plot3(x\_position\_neg,y\_position,z\_position,'r.'), view(0,90), hold on

end

%end

end

end



%Geometric Parameters

Length\_1 = 6e-2;

Ratio\_1to2 = 1.1;

Length\_3 = 8e-2;

Theta\_3\_max = 45;

Length\_4 = 7e-2;

Theta\_4\_max = 45;

Theta\_1\_max = 60;

Theta\_12\_min = 30;

Theta\_12\_max = 135;

Step = pi()/180;

%Calculated Lengths and Conversions to radians

Length\_2 = Length\_1 \* Ratio\_1to2;

Theta\_3\_max = Theta\_3\_max \* pi() / 180;

Theta\_4\_max = Theta\_4\_max \* pi() / 180;

Theta\_1\_max = Theta\_1\_max \* pi() / 180;

Theta\_1\_min = -Theta\_1\_max ;

Theta\_3\_min = -Theta\_3\_max ;

Theta\_12\_min = Theta\_12\_min \* pi() / 180 ;

Theta\_12\_max = Theta\_12\_max \* pi() / 180 ;

hold on

subplot(2,1,1), plot3(0,0,0,'go')

subplot(2,1,2), plot3(0,0,0,'go')

%Nested Loops

for Theta\_1 = Theta\_1\_max : -24\*Step : Theta\_1\_min

for Theta\_2 = Theta\_12\_min : 10.5\*Step : Theta\_12\_max

for Theta\_3 = Theta\_3\_min : 6\*Step : Theta\_3\_max

%Theta\_3 = -30;

for Theta\_4 = 0 : 9\*Step : Theta\_4\_max

%Tool Position

x\_position = Length\_3 - (Length\_4 \* sin(Theta\_4));

x\_position\_neg = Length\_3 + (Length\_4 \* sin(Theta\_4));

Theta\_offset = Theta\_2 + Theta\_1 - (pi()/2);

Theta\_3\_offset = Theta\_3 + Theta\_offset;

y\_position = Length\_2 \* cos(Theta\_offset) - Length\_1 \* sin(Theta\_1) + Length\_4 \* sin(Theta\_3\_offset);

z\_position = Length\_1 \* cos(Theta\_1) + Length\_2 \* sin(Theta\_offset) - Length\_4 \* cos(Theta\_3\_offset);

hold all

subplot(2,2,1), plot3(x\_position,y\_position,z\_position,'.'), view(3), hold on

subplot(2,2,1), plot3(x\_position\_neg,y\_position,z\_position,'r.'), view(3), hold on

subplot(2,2,2), plot3(x\_position,y\_position,z\_position,'.'), view(0,0), hold on

subplot(2,2,2), plot3(x\_position\_neg,y\_position,z\_position,'r.'), view(0,0), hold on

subplot(2,2,3), plot3(x\_position,y\_position,z\_position,'.'), view(90,0), hold on

subplot(2,2,3), plot3(x\_position\_neg,y\_position,z\_position,'r.'), view(90,0), hold on

subplot(2,2,4), plot3(x\_position,y\_position,z\_position,'.'), view(0,90), hold on

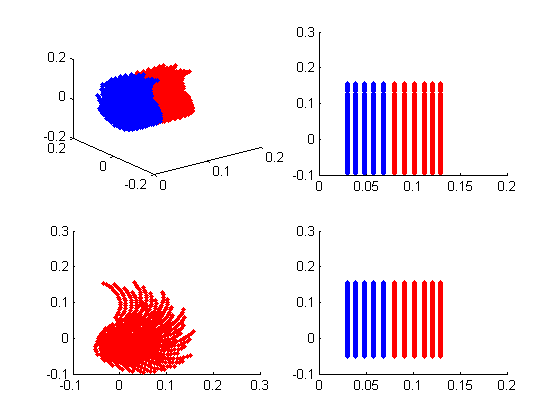
subplot(2,2,4), plot3(x\_position\_neg,y\_position,z\_position,'r.'), view(0,90), hold on

end

end

end

end



# Appendix E - MATLAB Code and Results for PT3a

%Geometric Parameters

Theta\_1\_max = 60 ;

Radius\_min = 6e-2 ;

Radius\_max = 9e-2 ;

Length\_3 = 8e-2 ;

Theta\_3 = 30 ;

Length\_4 = 7e-2 ;

Theta\_4\_max = 45 ;

Step = pi() / 180 ;

%Calculated Lengths and Conversions to radians

Theta\_1\_max = Theta\_1\_max \* pi() / 180 ;

Theta\_1\_min = -Theta\_1\_max ;

Theta\_3 = Theta\_3 \* pi() / 180 ;

Theta\_4\_max = Theta\_4\_max \* pi() / 180 ;

hold on

subplot(2,2,1), plot3(0,0,0,'go'), hold on

subplot(2,2,2), plot3(0,0,0,'go'), hold on

subplot(2,2,3), plot3(0,0,0,'go'), hold on

subplot(2,2,4), plot3(0,0,0,'go'), hold on

%Nested Loops

for Theta\_1 = Theta\_1\_max : -10\*Step : Theta\_1\_min

for Radius = Radius\_min : 3e-3 : Radius\_max

%for Theta\_3 = Theta\_3\_min : 6\*Step : Theta\_3\_max

for Theta\_4 = 0 : 5\*Step : Theta\_4\_max

%Tool Position

x\_position = Length\_3 - (Length\_4 \* sin(Theta\_4));

x\_position\_neg = Length\_3 + (Length\_4 \* sin(Theta\_4));

y\_position = Radius \* sin(Theta\_1) - Length\_4 \* cos(Theta\_4) \* sin(Theta\_1 - Theta\_3) ;

z\_position = Radius \* cos(Theta\_1) - Length\_4 \* cos(Theta\_4) \* cos(Theta\_1 - Theta\_3) ;

hold all

subplot(2,2,1), plot3(x\_position,y\_position,z\_position,'.'), view(3), hold on

subplot(2,2,1), plot3(x\_position\_neg,y\_position,z\_position,'r.'), view(3), hold on

subplot(2,2,2), plot3(x\_position,y\_position,z\_position,'.'), view(0,0), hold on

subplot(2,2,2), plot3(x\_position\_neg,y\_position,z\_position,'r.'), view(0,0), hold on

subplot(2,2,3), plot3(x\_position,y\_position,z\_position,'.'), view(90,0), hold on

subplot(2,2,3), plot3(x\_position\_neg,y\_position,z\_position,'r.'), view(90,0), hold on

subplot(2,2,4), plot3(x\_position,y\_position,z\_position,'.'), view(0,90), hold on

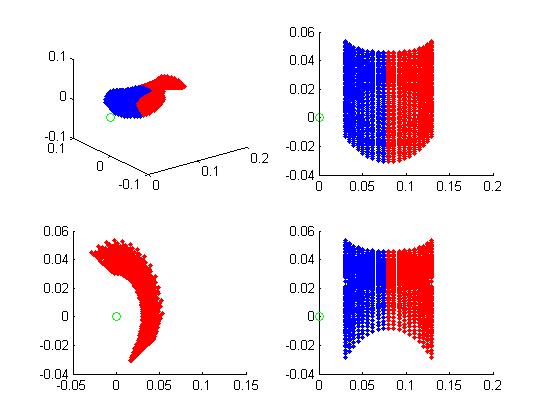
subplot(2,2,4), plot3(x\_position\_neg,y\_position,z\_position,'r.'), view(0,90), hold on

end

%end

end

end



# Appendix F - MATLAB Code and Results for PT13b

%Geometric Parameters

Theta\_3\_max = 45 ;

Radius\_min = 6e-2 ;

Radius\_max = 10e-2 ;

Length\_3 = 8e-2 ;

Theta\_1 = -45 ;

Length\_4 = 7e-2 ;

Theta\_4\_max = 45 ;

Step = pi() / 180 ;

%Calculated Lengths and Conversions to radians

Theta\_3\_max = Theta\_3\_max \* pi() / 180 ;

Theta\_3\_min = -Theta\_3\_max ;

Theta\_1 = Theta\_1 \* pi() / 180 ;

Theta\_4\_max = Theta\_4\_max \* pi() / 180 ;

hold on

subplot(2,2,1), plot3(0,0,0,'go'), hold on

subplot(2,2,2), plot3(0,0,0,'go'), hold on

subplot(2,2,3), plot3(0,0,0,'go'), hold on

subplot(2,2,4), plot3(0,0,0,'go'), hold on

%Nested Loops

%for Theta\_3 = Theta\_3\_max : -10\*Step : Theta\_3\_min

for Radius = Radius\_min : 3e-3 : Radius\_max

for Theta\_3 = Theta\_3\_min : 6\*Step : Theta\_3\_max

for Theta\_4 = 0 : 5\*Step : Theta\_4\_max

%Tool Position

x\_position = Length\_3 - (Length\_4 \* sin(Theta\_4));

x\_position\_neg = Length\_3 + (Length\_4 \* sin(Theta\_4));

y\_position = Radius \* sin(Theta\_1) - Length\_4 \* cos(Theta\_4) \* sin(Theta\_1 - Theta\_3) ;

z\_position = Radius \* cos(Theta\_1) - Length\_4 \* cos(Theta\_4) \* cos(Theta\_1 - Theta\_3) ;

hold all

subplot(2,2,1), plot3(x\_position,y\_position,z\_position,'.'), view(3), hold on

subplot(2,2,1), plot3(x\_position\_neg,y\_position,z\_position,'r.'), view(3), hold on

subplot(2,2,2), plot3(x\_position,y\_position,z\_position,'.'), view(0,0), hold on

subplot(2,2,2), plot3(x\_position\_neg,y\_position,z\_position,'r.'), view(0,0), hold on

subplot(2,2,3), plot3(x\_position,y\_position,z\_position,'.'), view(90,0), hold on

subplot(2,2,3), plot3(x\_position\_neg,y\_position,z\_position,'r.'), view(90,0), hold on

subplot(2,2,4), plot3(x\_position,y\_position,z\_position,'.'), view(0,90), hold on

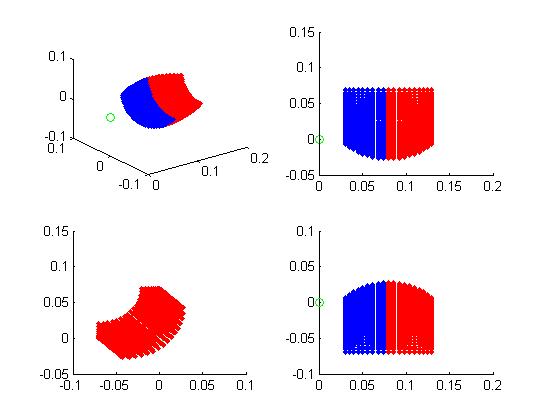
subplot(2,2,4), plot3(x\_position\_neg,y\_position,z\_position,'r.'), view(0,90), hold on

end

end

end

%end



# Appendix G - MATLAB Code and Results for PT4

%Geometric Parameters

Length\_1 = 5e-2;

Ratio\_1to2 = 1.1;

Length\_3 = 8e-2;

Theta\_3\_max = 45;

Length\_4 = 7e-2;

Theta\_4\_max = 45;

Theta\_1\_max = 60;

Theta\_12\_min = 30;

Theta\_12\_max = 135;

Step = pi()/180;

%Calculated Lengths and Conversions to radians

Length\_2 = Length\_1 \* Ratio\_1to2;

Theta\_3\_max = Theta\_3\_max \* pi() / 180;

Theta\_4\_max = Theta\_4\_max \* pi() / 180;

Theta\_1\_max = Theta\_1\_max \* pi() / 180;

Theta\_1\_min = -Theta\_1\_max ;

Theta\_3\_min = -Theta\_3\_max ;

Theta\_12\_min = Theta\_12\_min \* pi() / 180 ;

Theta\_12\_max = Theta\_12\_max \* pi() / 180 ;

hold on

subplot(2,2,1), plot3(0,0,0,'go'), hold on

subplot(2,2,2), plot3(0,0,0,'go'), hold on

subplot(2,2,3), plot3(0,0,0,'go'), hold on

subplot(2,2,4), plot3(0,0,0,'go'), hold on

%Nested Loops

for Theta\_1 = Theta\_1\_max : -10\*Step : Theta\_1\_min

for Theta\_12 = Theta\_12\_min : 7\*Step : Theta\_12\_max

%for Theta\_3 = Theta\_3\_min : 6\*Step : Theta\_3\_max

%Theta\_3 = 0;

for Theta\_4 = 0 : 5\*Step : Theta\_4\_max

%Tool Position

x\_position = Length\_3 - (Length\_4 \* sin(Theta\_4));

x\_position\_neg = Length\_3 + (Length\_4 \* sin(Theta\_4));

delta\_z = Length\_4 \* (1 - cos(Theta\_4));

Length\_2\_star = (Length\_2 ^2 + delta\_z ^2) ^.5;

Theta\_star = tan(delta\_z / Length\_2) ;

Theta\_12\_star = Theta\_12 + Theta\_star ;

y\_position = Length\_2\_star \* cos(Theta\_1 + Theta\_12\_star) - Length\_1 \* sin(Theta\_1) ;

z\_position = Length\_2\_star \* sin(Theta\_1 + Theta\_12\_star) + Length\_1 \* cos(Theta\_1) ;

hold all

subplot(2,2,1), plot3(x\_position,y\_position,z\_position,'.'), view(3), hold on

subplot(2,2,1), plot3(x\_position\_neg,y\_position,z\_position,'r.'), view(3), hold on

subplot(2,2,2), plot3(x\_position,y\_position,z\_position,'.'), view(0,0), hold on

subplot(2,2,2), plot3(x\_position\_neg,y\_position,z\_position,'r.'), view(0,0), hold on

subplot(2,2,3), plot3(x\_position,y\_position,z\_position,'.'), view(90,0), hold on

subplot(2,2,3), plot3(x\_position\_neg,y\_position,z\_position,'r.'), view(90,0), hold on

subplot(2,2,4), plot3(x\_position,y\_position,z\_position,'.'), view(0,90), hold on

subplot(2,2,4), plot3(x\_position\_neg,y\_position,z\_position,'r.'), view(0,90), hold on

end

%end

end

end

