APSC 496

Technical Analysis Report

Praxim - Surgical Robot

(Draft Version 1)

Submitted on: 28-JAN-2010

Submitted by: Nicholas Adams

Davy Chiu

Ibrahim Gadala

David Mountford

Erica Wodzak

# Abstract

Table of Contents

[Abstract 2](#_Toc252463897)

[Introduction 4](#_Toc252463898)

[Structural 4](#_Toc252463899)

[Gravity Compensation 6](#_Toc252463900)

[Motors 6](#_Toc252463901)

[Torque 6](#_Toc252463902)

[Speed 9](#_Toc252463903)

[Control System 11](#_Toc252463904)

[Choosing a Control System 11](#_Toc252463905)

[Determining the PID Gains 13](#_Toc252463906)

[Determining the Accuracy 13](#_Toc252463907)

[Conclusion 13](#_Toc252463908)

[Appendix 14](#_Toc252463909)

# Introduction

# Structural

An essential calculation which had to be completed before moving on to other analyses is that of the linkage lengths (most importantly Link1 and Link 2) and their corresponding “workable” area. In this analysis, we try to optimize the link lengths to maximize the workable area which can be sculpted at the tool bit. However, the link sizes should not be too large in order to minimize size and weight of the overall device. Thus, a balance must be stuck between optimum link sizes and maximum workable area.

At first, we must define how we express the workable area at the tool bit. From the *Matlab* plots results discussed in the CFP, we already know the general shape of these areas. They can be *approximated* as the area between two circular boundaries, one boundary of which is larger than the other. Thus, we define the outer or larger curved boundary as Rmax and the inner our smaller curved boundary as Rmin. The side boundaries of the workable area are approximated as straight vertical lines, and the distance between them is called Rangex. Thus, the workable area looks like the following (note: in reality, the origins of the radii should be different, but this is a very close approximation):

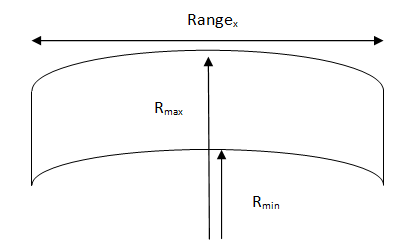


Figure - stuff

With the workspace coordinates defined, we must also define the variables on the linkage design in order to optimize the linkage sizes. A number of variables are incredibly important to this analysis, and are identified and illustrated below:

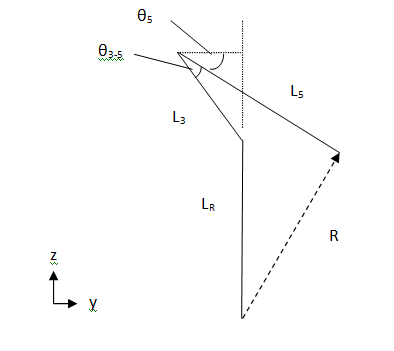


Figure - stuff

From the setup above, we can write the following geometric relationships between distances and angles:

(note: L1 , L2 and θ12 are not shown in the diagram)

We will create a virtual length called L’5:

L’5 = L5cosθ5

With that new virtual length, we can define the “reachable” length in the “z” and “y” directions stated above:

90 - )

90 - )

Since the “z” and “y” lengths described above are components of the R vector, we can compute the length of R simply as such:

Inserting all these variables and equations in a spreadsheet and optimizing the workable area based on links 1 and 2, the most reasonable option which gives us a large workable area while not overly extending the length of the links, happens when both links 1 and 2 are equal in length and measure 7 cm. The excel spreadsheet used in this analysis can be found in the Appendix.

# Gravity Compensation

# Motors

After winnowing down the motor types we can use to DC and stepper, we needed to perform a technical analysis on quantitative requirements for the hard-constraint system. In order to choose the correct motor, a power rating had to be specified. This power rating depends on the torque the motor must be able to provide and the speed the motor must be able to operate at.

## Torque

According to last year’s group, the torque the motor needed to supply was 7.5 Nm: “After researching various motor and motor controller combinations, the Maxon EC-Max 25W Brushless DC motor (Figure 47) matched with a 66:1 gearbox was selected. It was selected because it provided sufficient nominal torque, acceleration, and maximum velocity for the experimentally derived expected moment of 7.5 Nm (50 N force at 15cm) discussed in Section 3.” We preformed our own analysis on the newer linkage design and came up with a substantially smaller value of around 1 Nm. This result if most likely due to the fact that the newer linkage design reduces the amount of force needed to be applied at the hard constraint, due to the action of friction on the vertical rod and the partial load-carrying of the rod and linkages. The analysis was preformed with the following linkage design:

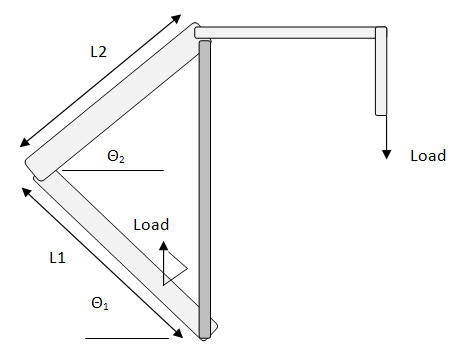


Figure - stuff

A spreadsheet was made to compute the value of the hard-constraint force that is needed at Link 1 (L1) with varying θs. Knowing the moment arm o the motor at that point, we can figure out the motor torque required. This spreadsheet proved to be very useful, as we could easily change the link lengths, the load, and the motor arm and immediately get the corresponding data. With link lengths of L1 = 15 cm and L2 = 10 cm, a load of 15 kg (hand force and the weight of the system), and a motor moment arm of 1 cm, we got back the following torque-theta curve:

Figure - stuff

\****θ1 (deg) is does not go below 48 degrees due to the different link lengths***

With the same load and moment values, but equal link length of 7 cm, the following curve is obtained (with a max torque of 1.5 Nm). Since the analysis of an optimized size for the workspace has yielded a result of equal linkage length of 7 cm, the maximum torque the motor must be able to counter is confirmed to be 1.5 Nm. The spreadsheet with all the values of thetas, distances and forces can be found in Appendix.

Figure - stuff

## Speed

An excel spreadsheet was made for the analysis of the change in link angles for movements of 0.5mm (up or down) of the tool attached to link 5. The link design figure used for the torque vs. theta relationship also illustrates the different links and how they move relative to the load. When the load moves up or down at an increment of 0.5mm, the angle θ changes. However, depending on the initial and final position of the load (and thus the length of the vertical link, shown in dark grey), the angle θ changes by different increments.

∆θ sees its largest value when the load is moving between it’s highest and second highest points (essentially near 14cm relative height); this corresponds to a 2˚ ∆θ. The in-depth calculations can be seen in the excel spreadsheet in the Appendix attached to this report. Two methods were used:

1. A manual change in increments and then evaluation of the corresponding ∆θ values.
2. Using *Excel’s* built-in solver to maximize the ∆θ cell value by changing the height cell value. This quickly and automatically finds the largest ∆θ for the setup.

Working based upon the largest ∆θ value, 2o, work backwards to find the minimum required speed of the hard-constraint movement. Approximating the surgeon’s motion at a speed of 10cm/s, the angle θ will change at approximately 40˚/s. This translates to a minimum of 6.67 rpm for the hard-constraint motion. Adding a sufficient safety factor to ensure the correct performance of the motor, we thus found the minimum speed of the hard constraint to be 30 rpm.

Knowing the torque and speed required of the motor, we can compute the power rating of the motor. Compared to the design used by last year’s group, the new linkage design has the overall advantage of requiring a lower torque and comparable speeds – thus, the power rating of the motor we need will be less. Last year used a pricey 25 Watt Maxon EC Max precision Motor in set with a 66:1 planetary gear head and a basic optical encoder. Since our desired power rating is considerably less than 25 Watts, and since the motor already present is functional, we choose to re-use the same motor as last year. We have proved, through calculations, our required torque and speed, and the current motor we have meets our functional requirements.



# Control System

## Choosing a Control System

In choosing a processing unit for the control system, the criteria most concerning the device are speed, programming difficulty and cost. The table below highlights the options available in a general sense.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Programming  Difficulty | Speed | Power | Cost | Display |
| 8/ 16 bit microcontroller | Moderate | Fast  ~20MHz | Low  <1W | ~$50 | Difficult to add |
| 32 bit microcontroller | Moderate | Very fast  ~600MHz | Low  1-5W | ~$150-400 | Possible |
| 32 bit x86 processor (PC) | Easy/moderate | Slow/fast  (depending on OS & h/w) | High  <50W | ~$150-500 | Easy to add |
| FPGA | Moderate  -difficult | Very fast  ~100MHz (no overhead) | Low  <5W | ~$100-$800 | Difficult to add |

Table - Processing Unit Options

How fast of a processing unit do we need? We know that the system as a whole must update at a rate of at minimum 1 kHz based on the results from Nikolai’s thesis. There are two main tasks the processing unit must perform. It must first read the values from the encoders/sensors and calculate the position of the end-effecter relative to the virtual and physical surface. Then based on that position, it must calculate the position of the hard restraint mechanism. In other words, it does a forward kinematics calculation and then a reverse kinematics calculation.

(block diagram)

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) ;

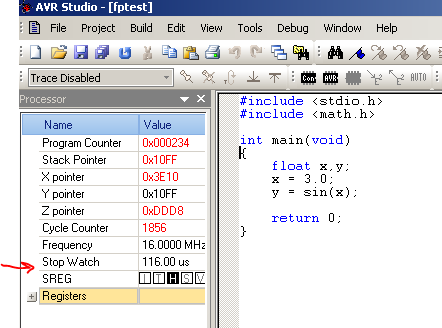
Theta\_12\_star = Theta\_12 + Theta\_star ;

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) ;

Figure - Code excerpt from Matlab that calculates tool position

Lookup table?



## Determining the PID Gains

## Determining the Accuracy

The accuracy of the robot is determined by the amount of error produced by the control system from the input to the output. The following diagram depicts all the possible errors involved.

encoder

Compute

position

Error 1

Compute

blocker

Error 4

Error 3

Error 2

motor

Motor controller

# Conclusion

# Appendix

