APSC 496

Technical Analysis Report

Praxim - Surgical Robot

Version 1.0

Submitted on: 28-JAN-2010

Submitted by: Nicholas Adams

Davy Chiu

Ibrahim Gadala

David Mountford

Erica Wodzak

# Abstract

The purpose of this report is to look at the design variables and perform analysis to provide a basis for the design decisions of first prototype. Structural analysis examined the optimal link lengths to achieve a desired workspace based on arbitrary lengths. Gravity compensation analysis looked at the viability of mechanisms to neutralize the additional weight of the tool perceived by the user. Solutions include passive compensation with springs or active compensation with motors. Motor analysis verified the Maxon EC Max motor would be suitable for our design based on analyses on motor power and speed. Control system analysis found a suitable class of processing units that satisfied the required processing power for the computation of the tool position and hard constraint. The details of the results are presented in this report.

Table of Contents

[Abstract 2](#_Toc252473920)

[List of Figures 5](#_Toc252473921)

[Introduction 7](#_Toc252473922)

[Structural 7](#_Toc252473923)

[Gravity Compensation 9](#_Toc252473924)

[Goal 9](#_Toc252473925)

[Background 10](#_Toc252473926)

[Spring 10](#_Toc252473927)

[Motor Counterweight 10](#_Toc252473928)

[General Design 11](#_Toc252473929)

[Technical Analysis 11](#_Toc252473930)

[Gravity Compensation Results 12](#_Toc252473931)

[Commentary 16](#_Toc252473932)

[Gravity Compensation Options 17](#_Toc252473933)

[Motors 20](#_Toc252473934)

[Torque 20](#_Toc252473935)

[Speed 23](#_Toc252473936)

[Control System 25](#_Toc252473937)

[Speed 25](#_Toc252473938)

[Conclusion 29](#_Toc252473939)

[Appendix 30](#_Toc252473940)

# List of Figures

[Figure 1 – Workable area 7](#_Toc252474145)

[Figure 2 – Variables defined on linkages 7](#_Toc252474146)

[Figure 3: Diagram of potential gravity compensation measures 10](#_Toc252474147)

[Figure 4 – Plot of torque vs. theta R angle for unsymmetrical design where theta 12 = 90 deg 12](#_Toc252474148)

[Figure 5 - Plot of torque vs. theta 12 angle for unsymmetrical design where theta R = 0 deg 12](#_Toc252474149)

[Figure 6 - Plot of torque vs. theta 12 angle for unsymmetrical design where theta R = 45 deg 12](#_Toc252474150)

[Figure 7 - Plot of torque vs. theta 54 angle for unsymmetrical design where theta R = 0 deg 13](#_Toc252474151)

[Figure 8 - Plot of torque vs. theta 54 angle for unsymmetrical design where theta R = 45 deg 13](#_Toc252474152)

[Figure 9 - Plot of torque vs. theta R angle for symmetrical design where theta 12 = 90 deg 13](#_Toc252474153)

[Figure 10 - Plot of torque vs. theta 12 angle for symmetrical design where theta R = 0 deg 14](#_Toc252474154)

[Figure 11 - Plot of torque vs. theta 12 angle for symmetrical design where theta R = 45 deg 14](#_Toc252474155)

[Figure 12 - Plot of torque vs. theta 54 angle for symmetrical design where theta R = 0 deg 14](#_Toc252474156)

[Figure 13 - Plot of torque vs. theta 54 angle for symmetrical design where theta R = 45 deg 15](#_Toc252474157)

[Figure 14 – Linkage design 20](#_Toc252474158)

[Figure 15 – Torque vs. Theta 21](#_Toc252474159)

[Figure 16 - stuff 22](#_Toc252474160)

[Figure 17 - Maxon EC Max motor 23](#_Toc252474161)

[Figure 18 - Processing unit options 24](#_Toc252474162)

[Figure 19 - Block diagram depicting processing blocks in green 25](#_Toc252474163)

[Figure 20 - Code excerpt from Matlab that calculates tool position 26](file:///H:\davychiu\My%20Documents\School\UBC\apsc496\documents\technical%20analysis%20report\technical%20analysis%20report%20v1.docx#_Toc252474164)

[Figure 21 - Floating point benchmark on AT90CAN128 27](#_Toc252474165)

[Figure 22 - Frequency estimate 27](#_Toc252474166)

# Introduction

This document provides an overview of the technical analysis completed for the Praxibot to date. The analysis focuses on assigning physical values to the overall structure of the device and developing mechanisms to implement functions critical to the overall physical layout.

The structural section describes the process used to assign link lengths and describes the method used to determine the desired operating envelop for the device. The link lengths are vital for the design as they impact the structural design including bearing sizing and encoder placement, gravity compensation, and motor sizing and placement. Gravity compensation focuses primarily on developing the characteristics of the rotational gravity compensation problem and assesses the techniques considered to implement both rotational and vertical gravity compensation. The minimum motor size for hard constraint implementation has been determined based on the power requirements needed to resist the device weight and user force. Finally potential control systems are evaluated.

# 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):

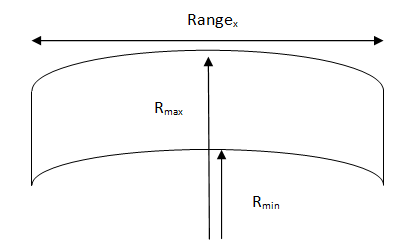


Figure – Workable area

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:

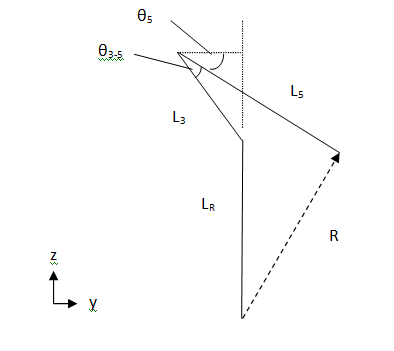


Figure – Variables defined on linkages

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 seen below.



# Gravity Compensation

This section discusses potential gravity compensation techniques and analyses their application to the device.

## Goal

Design a method to minimize the effect of unbalanced forces on the user experience of bone cutting using the four linkage uni-compartmental knee bone cutting device.

## Background

Gravity compensation is comprised of two components; (1) ability to stop device collapsing into the operating area and (2) ability stop device rotating about joint 1. The techniques considered utilize springs and counter weight designs and can be implemented either actively or passively. The active mechanisms are likely to be actuated by solenoids and motors located based on the device’s current orientation. The general orientation of the linkages can also be used to minimize the effect of unwanted forces, and counter weight designs are likely to use the motor as the counter weight to minimize the total weight of the device.

## Spring

Figure 3 provides an overview of the gravity compensation measures that may be implemented. The spring mechanism pictured can be implemented to stop the device collapsing into the operating area and to decrease the virtual weight experienced by the user while operating the device. In Figure 3 a compression spring is positioned along the radial constraint that restricts the motion of link 1 (L1) and link 2 (L2). The actuated stopper position can be varied actively or passively and will act to vary the length of the spring, thereby varying the force that counters the weight of the linkages and the tool. An active stopper could be actuated by a solenoid and would be positioned based on the final position of link 4. The stopper response will be in real time, but is not likely to be time critical and speed is therefore not a concern.

## Motor Counterweight

The motor required for the hard constraint can be positioned to reduce unwanted rotation about joint 1. If positioned in the same plane as joint 1 the motor weight will have no effect on the moment of the device about joint 1. Any other orientation will either increase the moment generated by the device as it rotates or resist the force applied by the user. The use of any counterweight can be optimized to minimize the moment about joint 1, especially in critical positions, but must be consider any implications to the overall weight of the device the overall.

Figure : Diagram of potential gravity compensation measures

## General Design

Figure 3 shows an overview of the potential gravity compensation mechanisms that can be implemented. Joint 1 is located at the lower end of the L1 and two designs are considered – (1) Unsymmetrical: rotation into and out of the page (2) Symmetrical: rotation to the left and right of page. Link 4, not pictured, extends horizontally into page and is connected to Link 5.

## Technical Analysis

In order to determine the required torque needed to implement effective gravity compensation, the torque produced at different angles has been determined for the two designs described above. Table 1 lists the input variables used in the analysis. The link lengths used fulfill all envelope requirements. The link masses are based on a sample link taken from the previous design, and will likely vary considerably from the final design link mass. As a result this analysis provides only a sensitivity analysis.

|  |  |
| --- | --- |
| **INPUT VARIABLES** | |
| **LINK LENGTHS** | |
| length link 1 [m] | 0.07 |
| length link 2 [m] | 0.07 |
| length link 3 [m] | 0 |
| length link 4 [m] | 0.05 |
| length link 5 [m] | 0.07 |
| **LINK MASS** | |
| mass link 1 [kg] | 0.1 |
| mass link 2 [kg] | 0.1 |
| mass link 3 [kg] | 0.1 |
| mass link 4 [kg] | 0.1 |
| mass link 5 [kg] | 0.1 |
| **CONSTANTS** | |
| Gravity [m/s^2] | 9.81 |
| PI | 3.141593 |
| **LINK ANGLES** | |
| Theta 3R [deg] | 0 |
| Theta 34 [deg] | 0 |
| **MASS TOOL** | |
| mass tool [kg] | 0.635 |

Table – List of input variables used for moment analysis

### Gravity Compensation Results

The plots below show torque required for gravity compensation at different angles of three major variables – (1) ϴR the angle between the radial link and vertical plane, (2) ϴ12 the internal angle between L1 and L2 and (3) ϴ54 the angle between L5 and L4. Positive torque as shown in the plots below is a torque required to counteract a counterclockwise moment.

Figure – Plot of torque vs. theta R angle for unsymmetrical design where theta 12 = 90 deg

Figure - Plot of torque vs. theta 12 angle for unsymmetrical design where theta R = 0 deg

Figure - Plot of torque vs. theta 12 angle for unsymmetrical design where theta R = 45 deg

Figure - Plot of torque vs. theta 54 angle for unsymmetrical design where theta R = 0 deg

Figure - Plot of torque vs. theta 54 angle for unsymmetrical design where theta R = 45 deg

Figure - Plot of torque vs. theta R angle for symmetrical design where theta 12 = 90 deg

Figure - Plot of torque vs. theta 12 angle for symmetrical design where theta R = 0 deg

Figure - Plot of torque vs. theta 12 angle for symmetrical design where theta R = 45 deg

Figure - Plot of torque vs. theta 54 angle for symmetrical design where theta R = 0 deg

Figure - Plot of torque vs. theta 54 angle for symmetrical design where theta R = 45 deg

### Commentary

This analysis serves both as a means to evaluate two specific design orientations (unsymmetrical vs. symmetrical) and also to evaluate how potential gravity compensation mechanisms can be implemented to reduce the moment in critical areas. The desired link motion ranges used in the operating envelop and link sizing are shown in Table 2.

|  |  |
| --- | --- |
| Theta\_R | -45 to 45 |
| Theta\_12\_min | 30 |
| Theta\_12\_max | 120 |
| Theta\_5 | -45 to 45 |
| Theta\_3 | 20 |
| Theta\_3 | -15 |

Table – Table of maximum and minimum link ranges

As expected, the symmetrical design has requires no torque when the ϴR is equal to zero. For both designs the moment increases largely linearly away from zero, but overall range between the maximum and minimum moments for each design are considerably different. Using the same input parameters the unsymmetrical design torque varies by over 1 Nm between the ϴR limits, while the symmetrical design varies by approximately half that. The reason for this dramatic change may be due to the orientation, but this has not been confirmed yet.

In general the impact of changes in the other two angles considered has far less impact on the total torque required for gravity compensation, but is more significant when ϴR approaches its limits. ϴ12 has a considerable impact on the position of a large portion of the device’s mass and on the torque required as a result. While ϴ54 positions the mass of the tool, the total motion of the link is far smaller and the total impact on the torque required is considerably less as a result.

## Gravity Compensation Options

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Option** | **Description** | **Active/Passive** | **Implementation** | **Implications** | **Conclusion** |
| **Compression Spring** | A compression spring positioned along the radial linkage acts to resist downward force of tool and linkages towards operating area | Passive | The spring used must be sized according to the overall mass of the structure, and the force produced will increase linearly as the user approaches the operating surface. The tool will position itself away from the operating surface when released. The spring also has the potential to restrict the minimum operating radius. | The size or the spring and the resistance felt by the user will be proportional to the mass of the device that the spring must support – overall device mass becomes critical. If the tool is released while operating away from ϴR = 0 position the design has the potential to negatively affect rotational gravity compensation. | Feasible |
| **Extension Spring** | Two compression springs are positioned along L1 and L2 and will extend as ϴ12 decreases. This acts to resist downward force of tool and linkages towards operating area. | Passive | The spring used must be sized according to the overall mass of the structure, and the force produced will increase linearly as the user approaches the operating surface. The tool will position itself away from the operating surface when released | The size or the spring and the resistance felt by the user will be proportional to the mass of the device that the spring must support – overall device mass becomes critical. If the tool is released while operating away from ϴR = 0 position the design has the potential to negatively affect rotational gravity compensation. | Feasible |
| **Compression Spring and linear servo** | A compression spring positioned along the radial linkage acts to resist downward force of tool and linkages towards operating area. The spring repositioned based on the orientation of the device to minimize unwanted forces | Active | A linear servo positioned at joint 1 will reposition the compression spring and provide and active constraint. The servo step length will determine the number of potential spring positions available and the servo will require control and power. | The mass of the device will have far less impact on resistance felt by the user. The linear servo will need to be mounted at joint 1 where space is limited. Extra processing capacity maybe required. | Will only be considered if mass is too great for passive spring options |
| **Motor Counterweight/ Counterweight** | Motor is attached to joint 1 below the centre of gravity to resist moment of the rest of the device at joint 1 | Passive | Motor can be attached below joint 1 either vertically or horizontally. Both cases require some sort of transmission to actuate the hard constraint. The counterweight size and position can be optimized to counter the moment due to changes in ϴR at a specific ϴ12. | Transmission will add complexity and cost to the design. The counterweight option has the potential to effectively track moments for a specific angle of ϴ12 to minimize moment at operating surface. | Feasible |
| **Torsion spring** | Torsion spring is implemented at joint 1 to resist moment in one direction. | Passive | A torsion spring will be attached to the radial constraint and the bone mount. The force generated by the spring will increase linearly as user passes bit of operating area. | Linkage setup will need to change to ensure the moment about joint 1 is only ever in on direction. Resistance to sideways motion is likely to reduce user experience. Fine tuning spring position and size is likely to be difficult | Not feasible |
| **Rotational Motor** | Motor positioned at joint one provides gravity compensation | Active | By plotting out moment at different positions a controller can be programmed to provide a counter torque to resist motion in that direction | Unlike the active spring mechanism described above the rotational motor is likely to be a fixed constraint and will significantly impact user input force required to operate the device. The motor must also be positioned at joint 1 where there is limited space, and will add weight to the design. | No Feasible |

# 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:

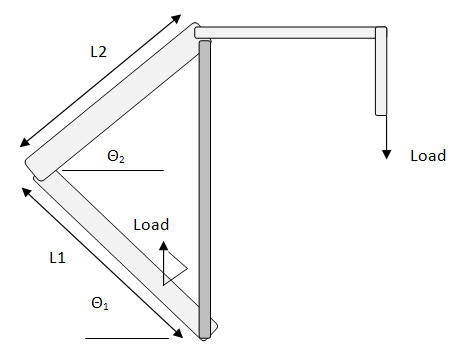


Figure – Linkage design

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 – Torque vs. Theta

\****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.



Figure - Maxon EC Max motor

# Control System

## Speed

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 |

Figure - 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.



Figure - Block diagram depicting processing blocks in green

Determining an exact metric for speed is difficult to do because speed depends on not only the frequency of the processing unit but also the architecture. However, if we assumed the architecture allowed that all instructions completed in one cycle, then the minimum speed can be determined from the code excerpt below.

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

With the assumption that floating point instructions complete in the same time as scalar instructions, all scalar instructions must be accounted for. It can be safely said to increase the number of instructions by an arbitrary factor of 200 to account for overhead and other instructions.

So a processor running at 5.6 MHz would be fast enough but only a general purpose processor like a Pentium processing core has the architecture – specifically a FPU (float point unit) – that would satisfy our assumption. However, Pentiums and alike have frequencies of 1GHz and above; so therefore any processor in that class would work for our purposes.

What about the worst case scenario? For this analysis, the focus is on the microcontroller used by last year’s group – the Atmel AVR 8-bit AT90CAN128 running at 16MHz. The code excerpt is used again as a comparison. Since this microcontroller does not have a floating point unit, all floating point instructions must be converted into integer math by the compiler. This has significant impact on performance, which can be seen in the benchmark below depicting a simulation for a “sine” instruction. If the code excerpt were to be implemented on this microcontroller, it would not satisfy the 1kHz speed requirement. However, simple floating point math like add and subtract have less of an impact on performance. In the code excerpt, 7 of the 28 functions are trigonometric functions. If we assume an average time of 25us and similar amount of instructions for the reverse kinematics calculation, the microcontroller is still not fast enough.

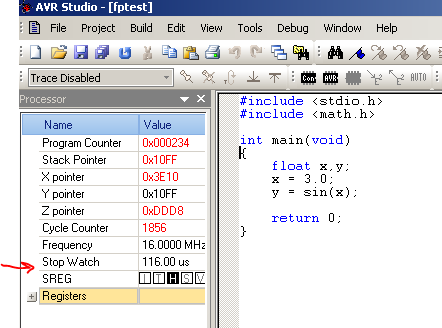


Figure - Floating point benchmark on AT90CAN128



Figure - Frequency estimate

There are several ways which the AT90CAN128 can be used. One method is to pre-compute all the possible values of the end-effecter position with corresponding block position and use a lookup table to search for the blocker position. This method is at the expense of resolution limited by the memory available.

From these results, it can be concluded that a faster 32-bit processor should be pursued if the cost is not an issue. Other factors like programming difficulty and real-time operating systems require further analysis.

# Conclusion