Trim, ballast and battery systems in a buoyancy engine for an underwater glider

Williams, S.

For the publisher’s version, please access the DOI link below./ Pour consulter la version de l’éditeur, utilisez le lien DOI ci-dessous.

Publisher’s version / Version de l'éditeur:
https://doi.org/10.4224/8895145
Student Report (National Research Council of Canada. Institute for Ocean Technology); no. SR-2005-16, 2005

NRC Publications Archive Record / Notice des Archives des publications du CNRC :
https://nrc-publications.canada.ca/eng/view/object/?id=a9260b87-0f6c-470d-bea2-446be292ecae
https://publications-cnrc.canada.ca/fra/voir/objet/?id=a9260b87-0f6c-470d-bea2-446be292ecae

REPORT NUMBER	NRC REPORT NUMBER	DATE		
SR-2005-16	---	August 2005		
REPORT SECURITY CLASSIFICATION		**DISTRIBUTION**		
Unclassified		Unlimited		
TITLE				
TRIM, BALLAST AND BATTERY SYSTEMS IN A BUOYANCY ENGINE FOR AN UNDERWATER GLIDER				
AUTHOR(S)				
Steven Williams				
CORPORATE AUTHOR(S)/PERFORMING AGENCY(S)				
National Research Council, Institute for Ocean Technology (NRC-IOT)				
PUBLICATION				

SPONSORING AGENCY(S)				
National Research Council, Institute for Ocean Technology (NRC-IOT)				
IMD PROJECT NUMBER		**NRC FILE NUMBER**		
42_2088_10		---		
KEY WORDS		**PAGES**	**FIGS.**	**TABLES**
buoyancy engine, BE, underwater glider, autonomous underwater vehicle, AUV		vii, 25, App. A-C	18	2

SUMMARY:

This report details the study into trimming and ballasting the buoyancy engine designed by NRC-IOT. The introduction discusses the basics of a glider, as well as current gliders already in operation around the world. Design considerations are then listed and their effects on the glider noted. The design of a nosepiece and tether cable prior to testing is described. The locations of the center of mass and center of buoyancy are calculated, are related to one another, and are discussed. Section 4 includes the implementation of a system of lead weights to test the buoyancy engine's ability to position itself horizontally and to rise and fall at acceptable glide angles. Section 5 details the concept selection of battery type, size, shape, weight, and other factors. MATLAB code for communicating with the serial data communications port is discussed and results of tests are given. Future work is then discussed along with a conclusion.

ADDRESS: National Research Council
Institute for Ocean Technology
Arctic Avenue, P. O. Box 12093
St. John's, NL A1B 3T5
Tel. (709) 772-5185 Fax: (709) 772-2462

National Research Council Canada | Conseil national de recherches Canada

Institute for Ocean Technology | Institut de technologies océaniques

TRIM, BALLAST AND BATTERY SYSTEMS IN A BUOYANCY ENGINE FOR AN UNDERWATER GLIDER

SR-2005-16

Steven Williams

August 2005

SUMMARY

This report details the study into trimming and ballasting the buoyancy engine designed by NRC-IOT. The introduction discusses the basics of a glider, as well as current gliders already in operation around the world. Design considerations are then listed and their effects on the glider noted. The design of a nosepiece and tether cable prior to testing is described. The locations of the center of mass and center of buoyancy are calculated, are related to one another, and are discussed. Section 4 includes the implementation of a system of lead weights to test the buoyancy engine's ability to position itself horizontally and to rise and fall at acceptable glide angles. MATLAB code for communicating with the serial data communications port is discussed and results of tests are given. Finally, section 5 details the concept selection of battery type, size, shape, weight, and other factors. Future work is then discussed along with a conclusion.

TABLE OF CONTENTS

1.0 INTRODUCTION	1
2.0 DESIGN CHALLENGE	3
2.1 Design Issues	3
3.0 MODIFICATIONS PRIOR TO TESTING	4
3.1 Nose Piece	4
3.2 Tether Cable	5
4.0 CENTER OF MASS AND CENTER OF BUOYANCY	7
4.1 Center of Mass Calculations	7
4.2 Center of Buoyancy Calculations	8
4.3 Relationship Between the COM and COB	9
5.0 BATTERY SYSTEM	12
5.1 Design Constraints	12
5.2 Battery Comparison	12
5.3 Analysis	18
6.0 MATLAB CODE	20
6.1 Communications	20
6.2 Controllers	20
7.0 CONCLUSIONS AND RECOMMENDATIONS	23
REFERENCES	24
APPENDIX A – CAD DRAWINGS	
APPENDIX B – MATLAB CODE	
APPENDIX C – SPECIFICATION SHEETS	

LIST OF TABLES

Table 1: COM of Buoyancy Engine 8
Table 2: Cost of batteries 19

LIST OF FIGURES

Figure 1: Seaglider's method of movement through water 1
Figure 2: Spray glider cross section 2
Figure 3: Nosepiece for BE 4
Figure 4: Tether Cable with Styrofoam for Neutral Buoyancy 6
Figure 5: Finding COM of the Buoyancy Engine 7
Figure 6: Position of COB as a result of diaphragm movement 9
Figure 7: COM and COB relationship 9
Figure 8: Change in COM and COB due to Moving Diaphragm 10
Figure 9: Horizontal position of Buoyancy Engine in the Trim Tank 11
Figure 10: Scale cross section of the BE 13
Figure 11: Packing Factor Inside Both Sections of the Buoyancy Engine 14
Figure 12 Battery Power in Electronics Section of BE 15
Figure 13: Power in the Motor Section of BE 16
Figure 14 Total Power of Battery System 16
Figure 15: Power due to maximum batteries for neutral buoyancy 17
Figure 16: Emoli Battery Layout 18
Figure 17: Depth and Position Using "if else" Controller 21
Figure 18: Depth and Position using PD controller 22

LIST OF ABBREVIATIONS AND SYMBOLS

AUV…………………………Autonomous Underwater Vehicle

ABS………………………….Acrylonitrile Butadiene Styrene

GPS………………………….Global Positioning System

NRC-IOT……………………National Research Council Institute for Ocean Technology

PVC…………………………Polyvinyl Chloride

GUI………………………….Graphical User Interface

COM………………………...Center of Mass

COB…………………………Center of Buoyancy

BE……………………..…….Buoyancy Engine

1.0 INTRODUCTION

AUVs, or Autonomous Underwater Vehicles are a new form of sub-sea technology emerging in today's race to explore one of the last mysteries left on earth, the Ocean. An AUV is simply a vessel that can explore water bodies without any direct human intervention. The AUV is given a preprogrammed path and then uses a combination of sensors and access to the GPS system in order to navigate. Recently, a new breed of the AUV has been developed and classed as an underwater glider.

The underwater glider is a very energy-efficient vessel that can operate without human contact for months at a time. It does this through the use of a buoyancy engine. This engine simply changes the volume inside the AUV to make it rise and fall in the water. It uses a small sliding mass to control its pitch attitude and moves forward using the lift force created on its wings.

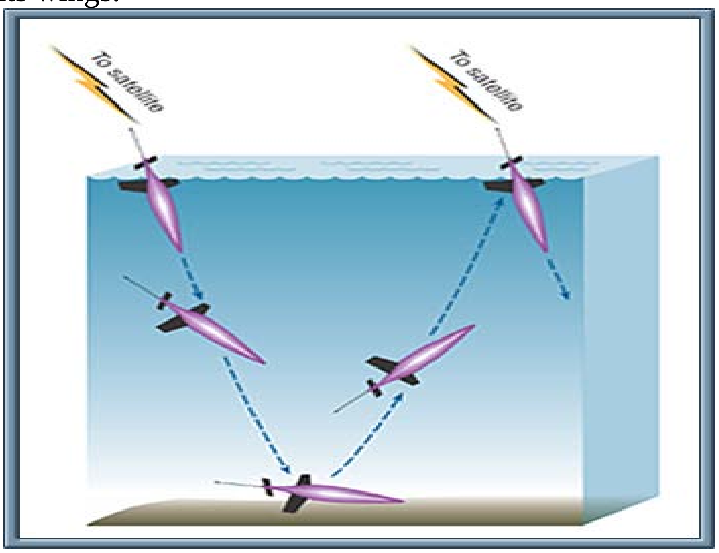


Figure 1: *Seaglider's method of movement through water*

The Seaglider is an underwater glider developed at the University of Washington and includes a very streamlined hull that matches the compressibility of water. In figure 1 you can see how it uses the upward and downward motions to move forward in the water. The antennae located on the back also permits it to connect to the GPS system when it surfaces, as well as receive and send instructions and data.

Besides the Seaglider, there are other gliders currently in use around the world. These include the Spray Glider developed by the Scripps Institution of Oceanography, and the Slocum Glider by Webb Research. The Slocum Glider has recently developed a thermal engine that permits it to carry out a mission for up to five years. It uses the temperature differences in the ocean as an energy source, which truly shows the advantages of the glider over other underwater research vessels. The Spray Glider is best know as the first glider to cross the Gulf Stream, moving at about 12 miles a day.

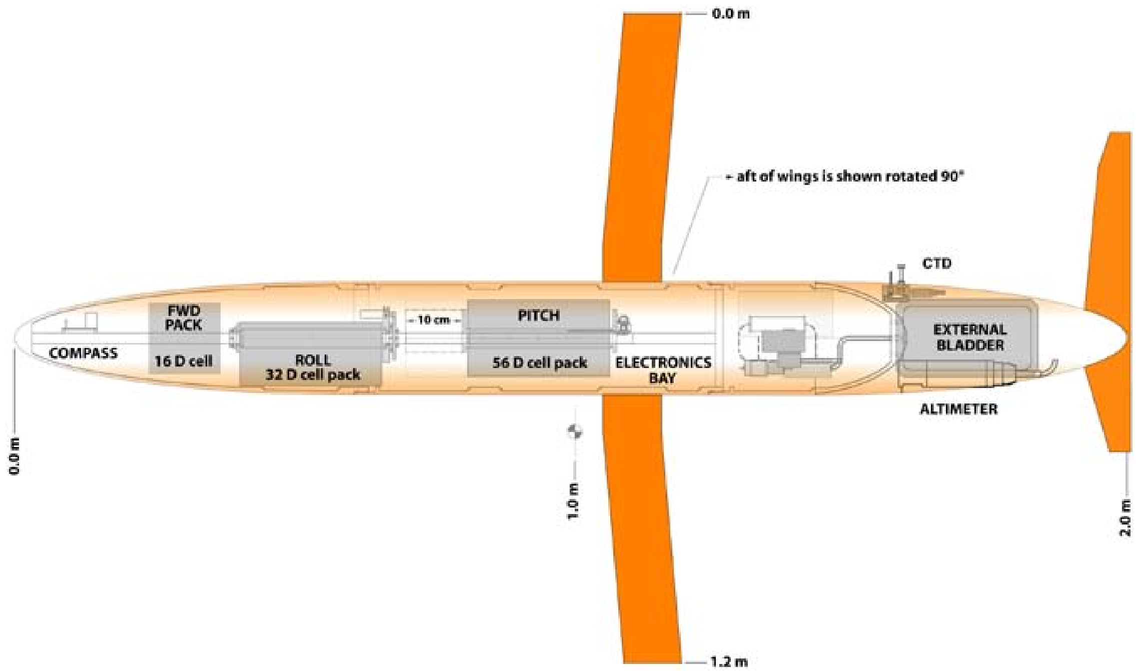


Figure 2: *Spray glider cross section*

Figure 2 shows the movement system in the Spray Glider as mentioned earlier. It uses a system of two bladders to change the volume located inside the hull. Its pitch and roll is controlled using two separate packs of moveable batteries.

2.0 DESIGN CHALLENGE

It is the goal of NRC-CNRC to produce a working Buoyancy Engine (BE) as a Test Platform for a flexible fin currently under development. This fin is made with rubber material capable of twisting its shape in order to control the BE's position and direction of motion. The BE uses a linear actuator and diaphragm to change its internal volume. Currently, it is operating using a tether cable connected to two 24V sources and the serial port. It also floats in a vertical position without pitch control.

In this phase of development, battery systems must be analyzed and concepts studied. Also, pitch control needs to be developed and the center of mass (COM) and center of buoyancy (COB) studied. As well, MATLAB code must be written that can collect the data from the serial port and plot instantaneously, leading to GUI development.

2.1 Design Issues

Weight – The BE cannot weigh more than its buoyant force including the mass of the batteries and the battery tray.

Batteries – The battery system needs to contain sufficient energy to power the BE and not create negative buoyancy. The batteries must fit into the BE without hindering any other aspect of the engine. Enough batteries must fit so that two 24V sources can be created. The batteries need to operate for at least one hour (average test time) without recharge.

Diaphragm – As the diaphragm extends past the end of the BE, it needs to be protected against collisions that could possibly damage and tear the rubber.

Pitch – When the piston moves, the locations of the COM and COB move. This can create a moment that will change the pitch attitude of the vehicle. This change needs to be controlled so that it can be used to an advantage. i.e. when BE increases its volume, the nose points upward and vice versa.

Tether Cable – The cable must be made neutrally buoyant in order to minimize the effects of its added mass on the end of the BE. Although this was not a big issue for the vertical testing, when the BE is placed horizontally, it will have a great impact on the validity of the data collected from the sensors.

3.0 MODIFICATIONS PRIOR TO TESTING

3.1 Nose Piece

As a result of the diaphragm being exposed, a nosepiece needed to be fabricated to protect the end of the Buoyancy Engine. Initially, a wire mesh was going to be used that would fit onto the nose via the holes in the end cap used during installation.
However, this would add a lot of extra mass to the front end of the BE which was very undesirable.
After concept selection, it was decided to fabricate the nose cone out of PVC that was left over from previous work. The PVC was then taken prior to machining and its volume and mass found. Its shape (cylindrical) allowed easy calculation of volume and mass and as a result, density (density = mass/volume). The density of the PVC was found to be 1.438 g/cm³.

***Figure 3**: Nosepiece for BE*

From the density the volume of the nose cap was determined:
Volume = Mass of Cap / Density of material
V = 549 g / 1.4378 cm/g³
V = 381.9 cm³

If the density of water is 1g /cm³ fresh water:
The added mass to the BE in water is 381.9 g and therefore must be included in the mass in water of the system.
It was also decided that the nose would be held in place via four screws therefore new holes had to be drilled into the end cap. These screws in total weigh 55.3 g and must also be included in the mass of the BE.

3.2 Tether Cable

The cable needed to be made neutrally buoyant so it would not hinder the vehicle during test.

At 10 meters long and 466 grams, it was decided the best way to do this was with Styrofoam placed at specific intervals along the cable.

At 1 g/cm³ in fresh water, the volume of the cable is equal to its mass in water

Assuming the cable is a cylinder,
$V = pi * r^2 * h$

where r = radius (cm) and h = length of cable (cm)

$V = (3.14159)\ (0.28)^2\ (1066.8)$
V = 262.8 cc

Therefore, for neutral buoyancy:
466 g – 262.8 g = 203.2g
203.2 g must be removed from the cable.

This can be achieved using the buoyant effects of the Styrofoam. However first the density of the foam had to be found. Using the density equation, it was found to be 0.0587 g/cm³.

To find the total volume and mass of Styrofoam needed:

Eq (1) density of foam = mass / volume
Eq (2) volume (1 cc = 1 g) – mass = 203.2 g

therefore:
Volume = 215.87 cc
Mass = 12.67 g

If a foam block is placed every 30 cm, 35 blocks would be needed, therefore the volume of each block must be 6.17 cc.

If blocks have a height of 2 cm and a hole for the cable:
$6.17\ cc = (L) * (L) * (H) – (0.28)^2(pi)(H)$

where: L = length and width of block
H = height of block = 2 cm

$6.17 = L^2 * (2) – 0.2375 * (2)$
L = 1.82 cm

It was then decided to cut these blocks in half in order to easily install them onto the cable. A small piece of tape was used to hold them together which does not significantly affect the buoyancy of the cable.

Figure 4*: Tether Cable with Styrofoam for Neutral Buoyancy*

4.0 CENTER OF MASS AND CENTER OF BUOYANCY

4.1 Center of Mass Calculations

The location of the center of mass of the BE is one of the major factors affecting the attitude of the BE. The COM must line up with the COB in order for the BE to maintain horizontal positioning. As the body is non-uniform it is very difficult to calculate the exact position of the COM. Experimentally, it can be found by hanging the Buoyancy engine from linear hanging scales and taking the mass readings as well as their position to a datum point on the BE.A MATLAB code was written to determine the center of mass (with and without batteries).

Figure 5: *Finding COM of the Buoyancy Engine*

The center of mass can be found using the following:

$x_c = (m_1*x_1 + m_2*x_2)/(m_1 + m_2)$
where:
x_c = center of mass
m_1 and m_2 = readings on scales 1 and 2 respectively
x_1 and x_2 = distances to scales 1 and 2 respectively

Initially, we wanted to find the COM without any added mass inside the hull.

A series of test were performed and the average of the test taken as the experimental center of mass:

Test #	Mass Scale 1 (kg)	Mass scale 2 (kg)	Distance to scale 1 (cm)	Distance to scale 2 (cm)	Center of Mass (cm)
1	4.72	3.52	33	92.5	58.41
2	2.5	6.15	15	74.5	57.30
3	4.32	3.78	23	97.5	57.77
4	4	4.32	24	89.2	57.78

Ave COM = 57.8 cm from electronics end cap

Table 1*: COM of Buoyancy Engine*

This value agreed with the rough measurement of 57.5 cm taken by placing the BE on a wedge.

4.2 Center of Buoyancy Calculations

The buoyant force is the upward force on the engine as a result of the volume of water it displaces. The COB is located at the geometric center of the displaced volume. Since the BE is composed of two simple shapes, the volume and center of both the end cap and hull can easily be calculated.

The cap was divided into two sections, the circular top and the legs. The hull was then divided into the outside cylinder, and the moveable inside cylinder (diaphragm).

The COB was then found by adding up the volumes of each section and the product of their individual centers and volumes. Summing these products and dividing by the total volume gives the location of center of buoyancy.
The equation is as follows:

$$x = [\ (x1 * V1) + (x2 * V2) + (xn * Vn)\ldots\ ] / V$$

where x = center of mass overall
V = total volume overall
Xn = center of mass of each section
Vn = Volume of each section

A MATLAB code was written and it was found that when the piston was in the neutral position, the COB was located at 51.25 cm from the electronics end cap.
The following graph shows the position of the COB when the diaphragm moves away from the neutral position:

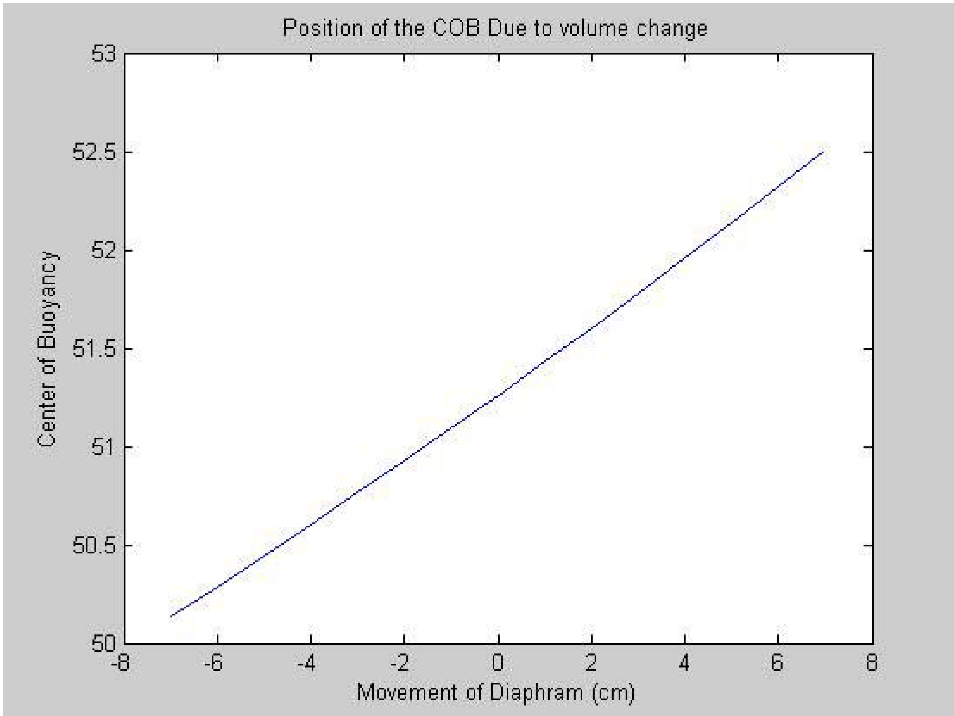


Figure 6: *Position of COB as a result of diaphragm movement*

4.3 Relationship Between the COM and COB

The COM and COB are closely related in controlling the pitch and roll attitude of the device. When they are located at different points in the BE, there is a moment created that will cause the BE to change its pitch attitude. This can be seen in the following diagram:

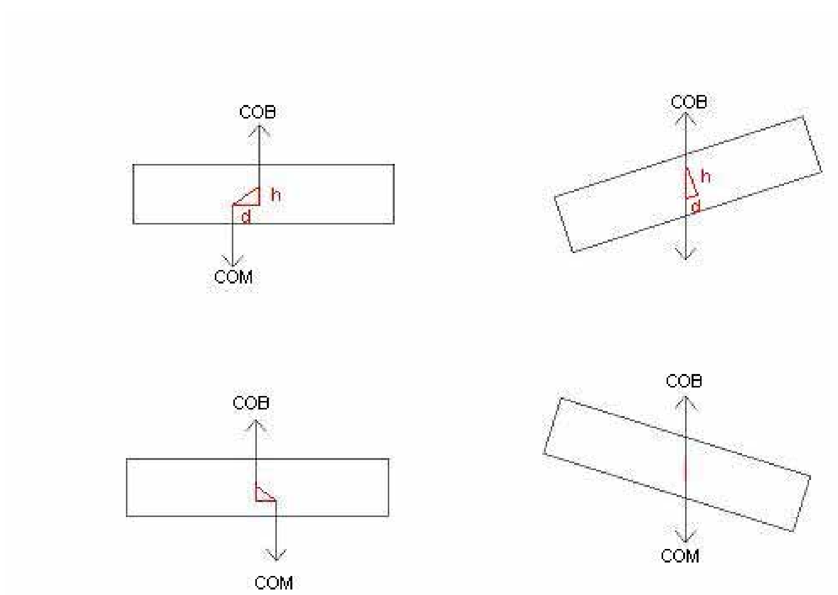


Figure 7*: COM and COB relationship*

The COM is located in the bottom of the engine for stability reasons. When the COB is located in front of the COM, the engine will tip upward. This is a direct result of their distance apart (d) and height difference (h).

The angle that the BE will tip can be determined easily when the ‘d’ and ‘h’ values are known. It is:

$$\theta = 90 - (\tan^{-1} (h/d))$$

where θ is the angle of the BE relative to the horizontal axis.

Using MATLAB, a program was written to determine the change in the positions of the COM and COB as a result of the piston movement (assuming that in the neutral position, the battery mass would allow the COM and COB to line up). The following results were obtained:

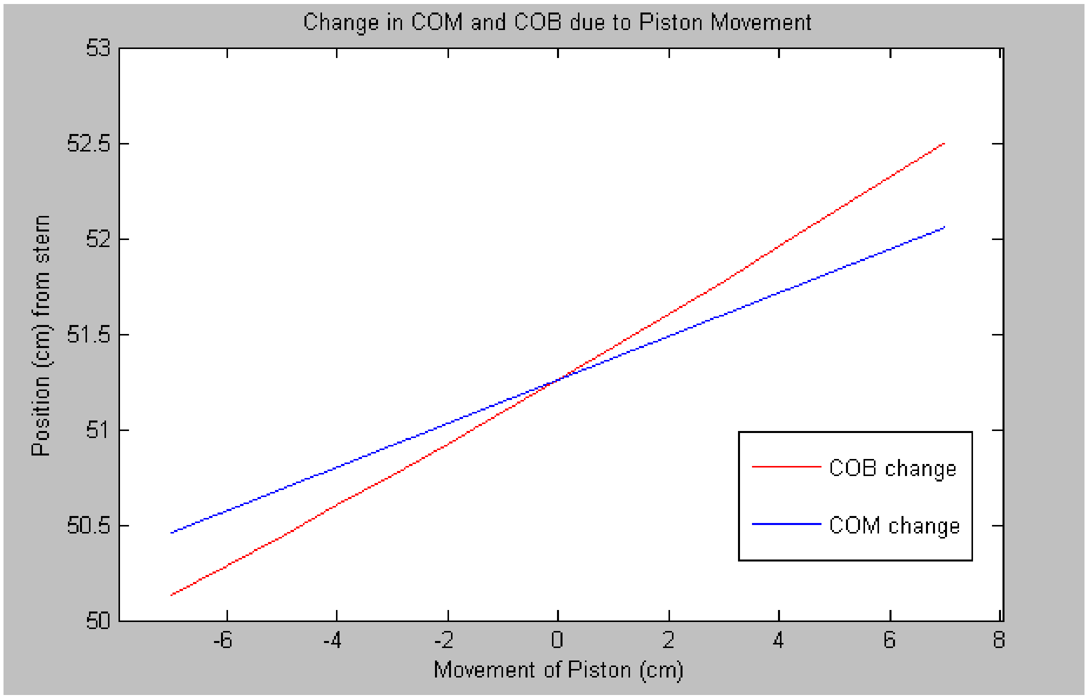


***Figure 8**: Change in COM and COB due to Moving Diaphragm*

From this graph it can be seen that the COB is located at about 51.25 cm when the piston is in the neutral position. Also, the COB changes by about 1.24 cm when the piston moves 7 cm.

After performing a second series of linear hanging scale tests, it was found that by moving the piston while the BE was hanging by the scales, the center of mass changed by 0.8 cm with a piston movement of 7 cm.

This situation will allow the COB to be located behind the COM when the piston is in an aft ward position and vice versa when the piston is in a forward position. As a result the nose will point down when the BE is diving, and up when the BE is surfacing.

The center of mass should be able to be changed by a moving battery tray that can shift the internal weight of the batteries towards the nose of the AUV. Until a battery system and layout are decided on, the exact changes in the center of mass cannot be calculated. (Battery systems are discussed in a separate section of the report).

The mathematical model was then tested though a variety of test conditions in the Trim Tank. Lead masses were used in place of the battery system. The MATLAB code was used in order to find the proper locations of the lead masses in order to achieve a level attitude in the neutral position.

Lead masses of 1.6 kg and 0.28 kg were placed at 31.5 cm and 5 cm respectively from the end cap. Also, a mass of 0.39 kg was taped to the outside of the nose cone (107.5 cm). With this setup, the BE was horizontal and neutral when the piston was moved –1.8 (aft ward) from its starting position. This small positive buoyancy is desirable in case of malfunction. The engine's nosepiece did float up on ascent as expected. On descent, the engine moved down horizontally, indicating that with a small movement of batteries, this can be changed so that the nosepiece points downward.

Figure 9: *Horizontal position of Buoyancy Engine in the Trim Tank*

5.0 BATTERY SYSTEM

In designing a battery system for the AUV, there are many limits and restrictions that must all be incorporated into the final design. As all the constraints affect one another, it is difficult to find a system that will work.

We ultimately would like to design a battery system that uses two separate 24 V battery packs; one that will power the electronics and one to power the motor. According to Chris Warren's specification sheets [5], the motor will draw approximately 5 Amps and the electronics will draw 83 milliamps. For testing purposes, we would like to see the battery system last for at least an hour, as most testing will not take longer than this.

Some of the constraints we have to be concerned about are:

Type – Different battery compositions offer unique advantages. For example, Lithium ion batteries can be recharged, hold current well, and can tolerate higher current draw. However, they are costly, affected by temperature, and react violently with water. Alkaline batteries typically have a high current rating, and great lifetime, however, cannot be recharged.

Physical space – As the ABS pipe is only 10 cm in diameter and one metre long, there is not much physical space for batteries. Once the piston and electronics are in place, the amount of usable space is greatly decreased.

Shape – As a result of the space limitations, only certain shapes and sizes of batteries will fit into the BE, greatly reducing our selection.

Weight – The total weight of the BE must be equal to its volume in order to achieve neutral buoyancy. This means that a total mass of about 2.19 kg can be added in batteries and their mounting mechanisms.

The battery packs must be designed so that enough batteries can be connected together to make two 24 V sources with a high enough capacity as to be able to power the system for an hour.

Initially, I began to look at typical C and D cell alkaline batteries. Also, two different types of Li-ion batteries were researched. The first was a rectangular model made by Saft Batteries. These batteries are part of the MP series and come in three different sizes. The second was a cylindrical battery made by Moli Energy.

The shape of the batteries greatly affect how the system must be laid out for maximum packing efficiency. For this reason an AutoCAD model of the engine was constructed that would allow for scale models of batteries to be virtually placed in the engine.

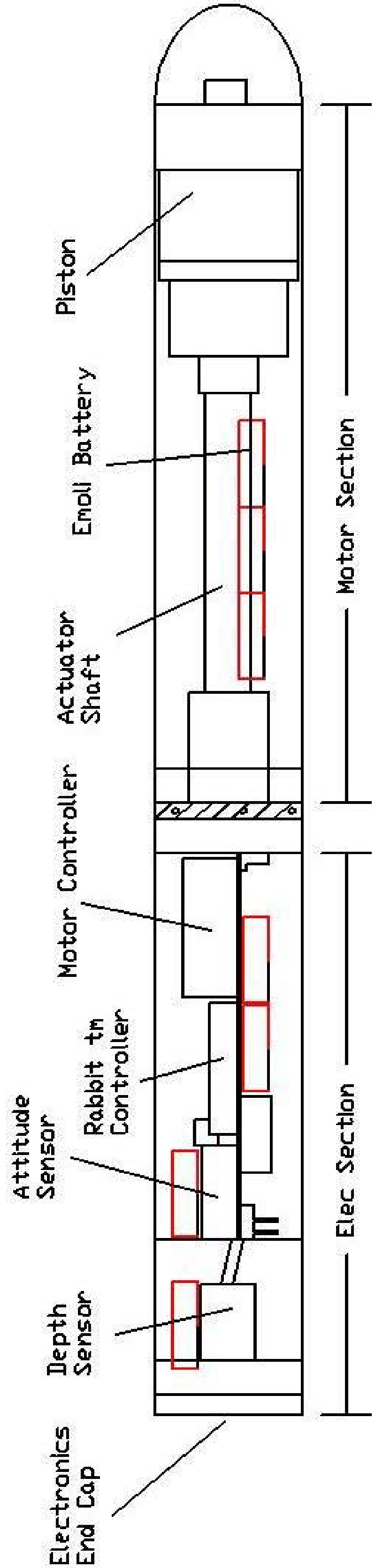
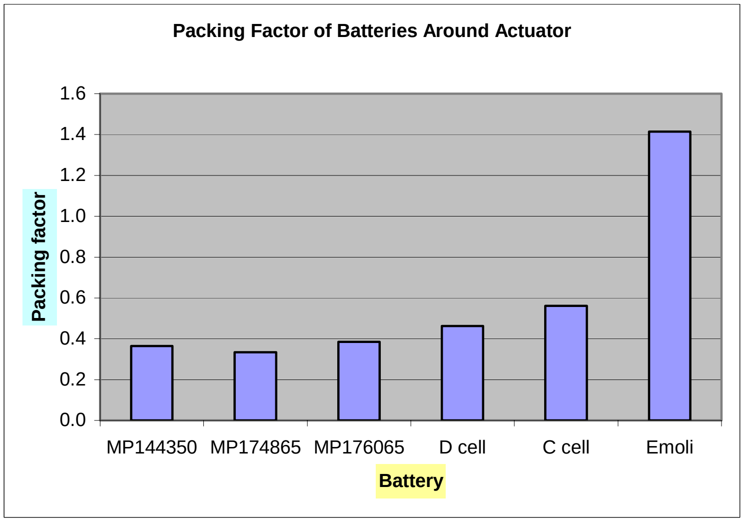


Figure 10: *Scale cross section of the BE*

The first design constraint considered was size. For this, the packing factor inside the BE was found by calculating the approximate internal volume of the electronics section and the motor section separately. A model of each battery was then placed into the AutoCAD model (see figure 10). The number of batteries that would fit into each section was determined as well as how many would have to be placed in the other section to achieve neutral buoyancy. The best battery layout, weight, and power were not involved in these initial calculations.

Battery Type	Volume (cm³)	# in engine	# left over for neutral buoyancy	Tot Volum of Batteries	Packing factor
MP144350	25.9	24	8	621.6	0.364
MP174865	47.5	12	5	570	0.334
MP176065	73	9	5	657	0.385
D cell	52.6	15	0	789	0.463
C cell	29.9	32	0	956.8	0.561
Emoli	67	36	9	2412	1.414

Area inside motor section (cm³) =	1705.8



Battery Type	Volume (cm³)	# around electronics	# left over for neutral buoyancy	Tot Vol of Batteries	Packing factor
MP144350	25.9	27	5	699.3	0.283
MP174865	47.5	10	7	475	0.192
MP176065	73	7	7	511	0.206
D cell	52.6	4	11	210.4	0.085
C cell	29.9	23	9	687.7	0.278
Emoli	67	22	23	1474	0.596

Area inside elec section (cm³) =	2475

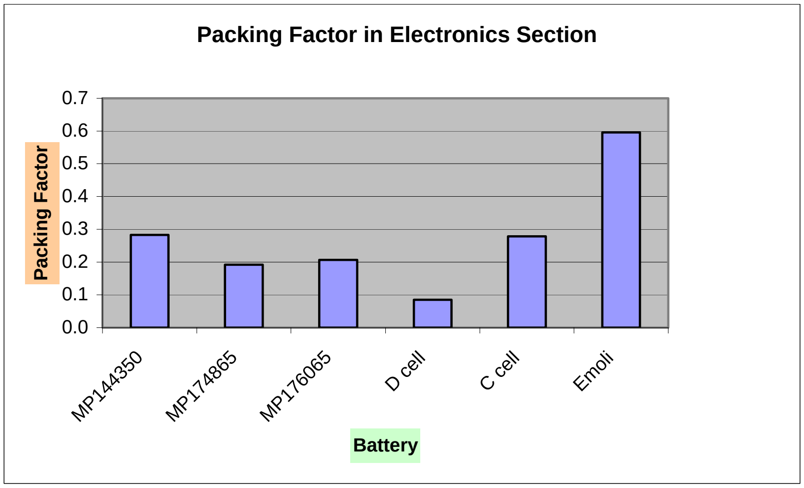


***Figure 11**: Packing Factor Inside Both Sections of the Buoyancy Engine*

From these results, it can be seen that the smaller sized batteries, as expected, generated the largest packing factor, particularly the Emoli batteries, which are about the same diameter as an AA battery. It can easily be seen that the D cell and the two largest of the MP series are not well suited to the application in regards to their size.

The next component looked at was the power output by each cell. The total power was calculated for each cell assuming a one-hour test time. As the electronics section and motor section use separate power sources, the power for each section was calculated separately and then added together. This can be seen in the following data sheet:

Battery System for Electronics Section of BE

* calculations assuming a 1 hour test time

Battery Type	# around sensor	# under circuit	Voltage (V)	Ah	current (Amps)	Power of 1 batt (Watt)	Power tot (Watt)
MP144350	7	7	3.6	2.3	2.3	8.28	115.92
MP174865	3	3	3.6	4.6	4.6	16.56	99.36
MP176065	2	2	3.75	6.1	6.1	22.875	91.5
C	7	6	1.5	6	6	9	117
D	0	4	1.5	12	12	18	72
Emoli	12	10	3.75	2.4	2.4	9	198

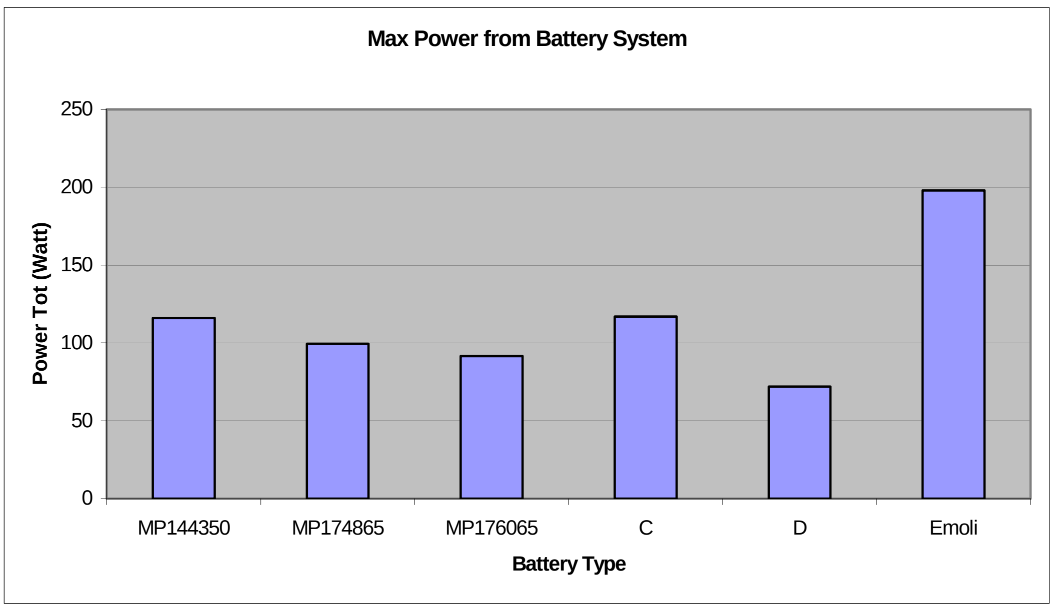


Figure 12 *Battery Power in Electronics Section of BE*

Battery System for Motor Section of BE *asuming 1 hour test time

Battery Type	# around actuator	Voltage (V)	Ah	Current (Amps)	Power of 1 batt (Watt)	Power tot (Watt)
MP144350	24	3.6	2.3	2.3	8.3	198.7
MP174865	12	3.6	4.6	4.6	16.6	198.7
MP176065	9	3.75	6.1	6.1	22.9	205.9
C	32	1.5	6	6	9	288
D	15	1.5	12	12	18	270
Emoli	36	3.75	2.4	2.4	9	324

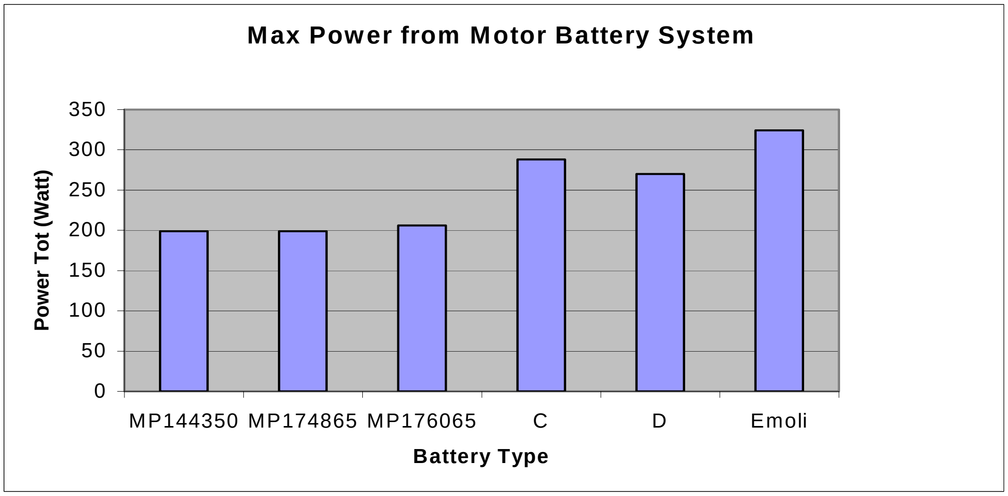


***Figure 13**: Power in the Motor Section of BE*

Total Power from Each Battery Type

Battery Type	Total Batteries	Batt needed for neutral buoyancy	Power in actuator	Power in Elecronics	Total Power
MP144350	38	32	198.7	115.9	314.6
MP174865	18	17	198.7	99.4	298.1
MP176065	13	14	205.9	91.5	297.4
C	45	32	288	117	405
D	19	15	270	72	342
Emoli	58	45	324	198	522

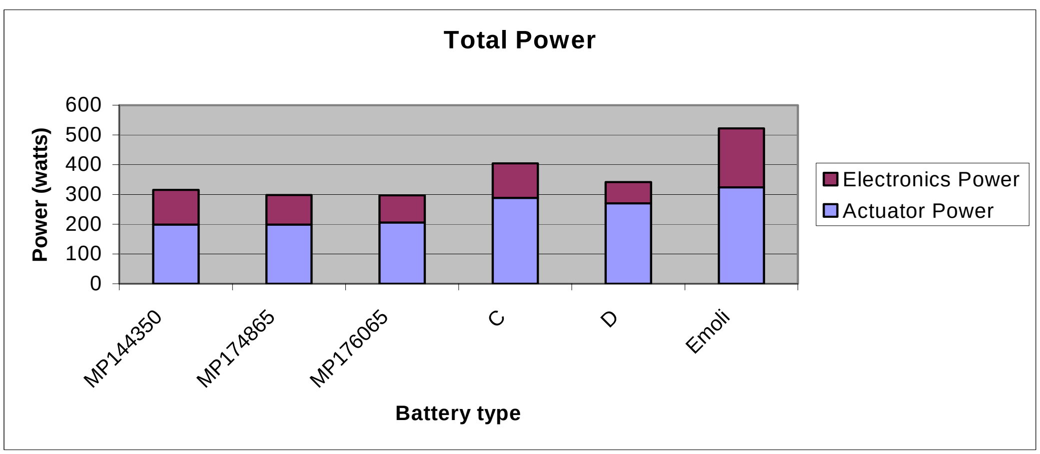


***Figure 14** Total Power of battery system*

From these data it was easily seen that the Emoli batteries would perform best, followed by the C cell battery. As the alkaline battery cannot be recharged, it is not an ideal solution. The MP Series batteries were very close in performance, however the smaller model produces the most power as a result of a higher battery concentration in the BE.

Another factor that had to be looked at was the total weight of the system.
The weight of each battery was calculated, and the maximum number of batteries for neutral buoyancy recorded. From this, another chart was made up comparing the cells:

Power for max Batteries due to Neutral Buoyancy

Battery Type	Total Batteries**	Batt Needed for Neutral Buoyancy	Power of 1 battery	Total Power
MP144350	32	32	8.28	264.96
MP174865	17	17	16.56	281.52
MP176065	13	14	22.875	297.375
C	32	32	9	288
D	15	15	18	270
Emoli	45	45	9	405

** For this graph, the battery total is limited by the amount allowed for neutral bouyancy.

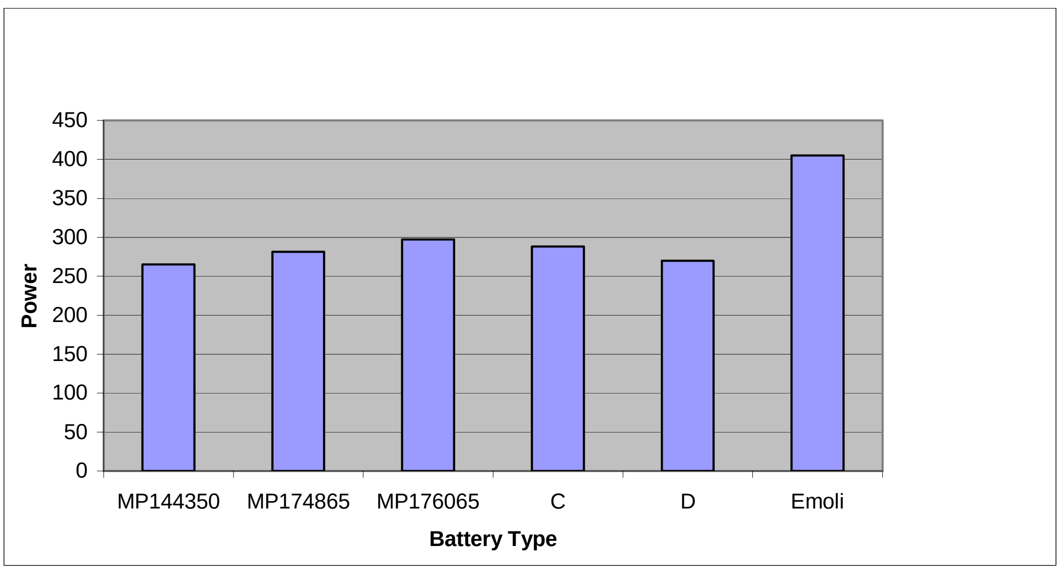


Figure 15: *Power due to max batteries for neutral buoyancy*

After analyzing the graphs, it was easy to see that the smaller batteries ultimately preformed well. Also, a few of the batteries could be completely ruled out:

a. The D cell battery would not fit into the electronics section, and with a battery tray would not fit into the motor section either.

b. The MP 176065 had problems in that a 24 V source could not be created with the batteries in the electronics section without having to use batteries from the motor section. This would hinder the sliding battery tray and add more weight to the nose.

The COM and COB of the best battery systems were then examined. AutoCAD models of the MP 174865 and the Emoli cells positioning in the BE were built. These positions were then used in the COM MATLAB program.

Ideally, the COM and COB should sit 51.25 cm from the electronics end cap when the piston is in the neutral position. Using the MP series, it was found that without adding extra weight or floats, the COM could not be moved behind the COB.

Using the Emoli batteries, it is possible to place both the COB and COM at 51.25 cm (see AutoCAD appendix). When the piston moves to +7cm (forward) , the COB moves to 52.5 and COM to 51.3. This will create a moment that will turn the BE upward which is desirable. If the batteries around the actuator are then slid forward, when the piston moves to – 7 cm (aft ward), the COB moves to 50.14 and the COM to 52.87. This would allow the BE to dive nose down at an angle dependant on the positioning of the batteries.

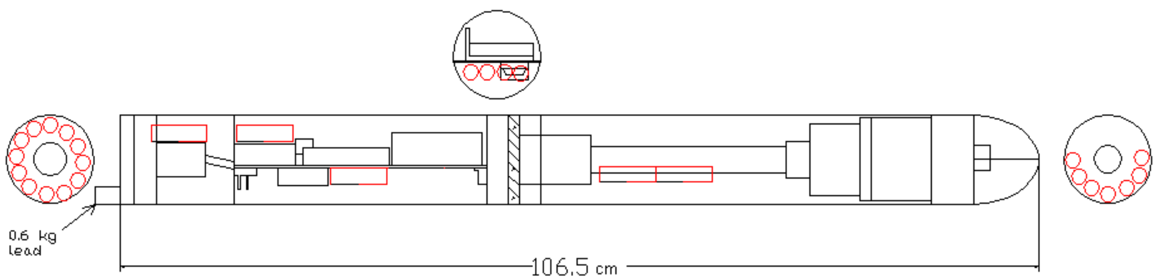


Figure 16: *Emoli Battery Layout*

From the above, it can be seen that the Emoli batteries are best suited to the application. As Emoli Canada will usually only sell direct to battery pack manufactures and the like, it may be difficult to obtain the cells. However, after searching for similar shaped batteries, a model from www.batteryspace.com, the ICR18650 was found with the same dimensions.

Battery Type	Cost per unit $	Amount Needed	Total Cost $
MP 144350	44.50 CAD	32	1424 CAD
MP 174865	62.50 CAD	17	1062.50 CAD
MP 176065	76.20 CAD	13	990.60 CAD
Emoli 18650J	4.00 US	32	128 US
Emoli 26700	15.00 US	n/a	n/a
Battery Space 18650	5.00 US	32	160 US

Table 2*: Cost of batteries*

As can be seen in this table, the MP series are very expensive and do not provide a power advantage. However, they do perform better in lower temperatures and for this reason could be considered in a final product. Some modifications of the internal layout and size would also be required.

For this engine, the industrial Li-Ion batteries from Emoli and Battery Space would provide the best price to performance ratio.

A few things to consider in the future about the battery system:

The current drawn by the motor depends on its depth. This is a result of the increased pressure pushing against the piston. A current probe is needed to determine the amount of power the motor will draw.

The maximum discharge rate of the battery cannot be exceeded. For this reason, the motor may have to use a battery with a higher discharge rate such as the Emoli 26700.

6.0 MATLAB CODE

6.1 Communications

It was desirable to communicate with the buoyancy engine via MATLAB as it would permit the ability to collect and create more graphs to analyze the data. As well, it will lead to the development of a GUI.

It was decided that the code would run externally to the BE. That is, the Dynamic C code programmed by Chris Warren would not be altered but controlled automatically by MATLAB instead of manually in HyperTerminal. This provided a problem as it was not clear initially if MATLAB's serial library would be able to do this.

The first issue was the text instructions and constants printed by the engine. These data had to be stored in a separate structure from the sensor data. A loop was set up to collect these data and it was found that it took exactly 40 cycles. The constants were then pulled out of this text, converted to doubles, and saved as variables.

The sensor data are collected in a loop and split up in a structure according to the data type. The raw data are then converted to readable data used in analysis.

6.2 Controllers

A controller for the piston was then programmed. A target depth is given to the engine and the piston must move until the target depth is reached. The vehicle's actual depth is measured by a pressure sensor. Initially, an "if else" statement was used to control the piston. However, this model could only move the piston by a constant amount each cycle. This resulted in the BE overshooting its target depth by a very large amount.

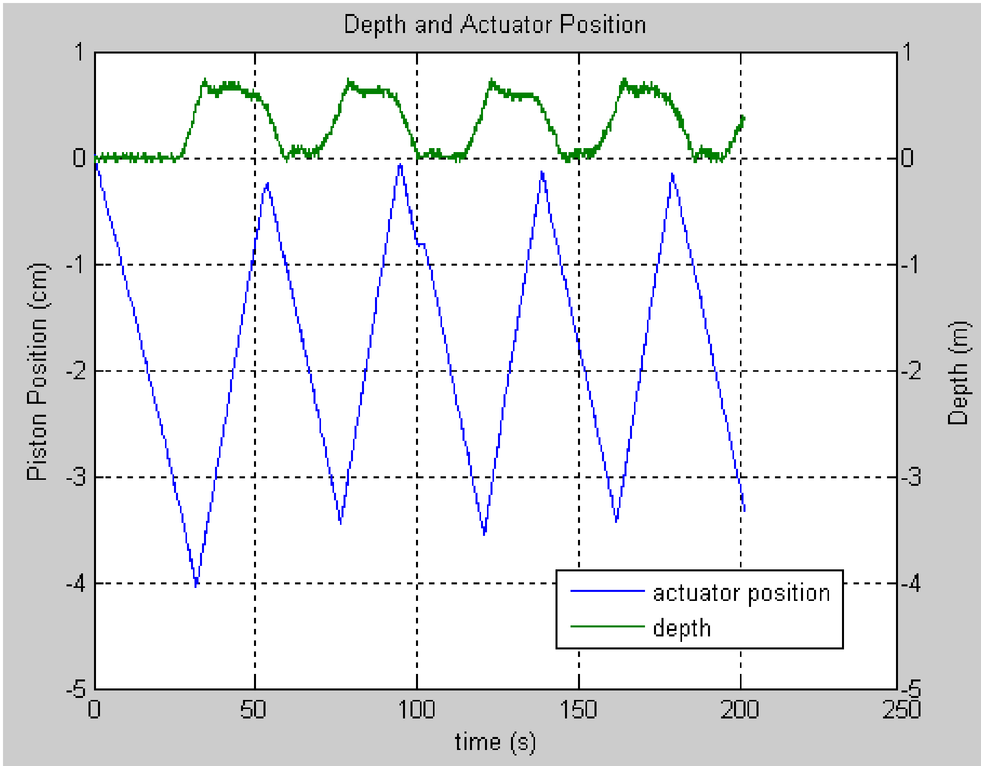


Figure 17*: Depth and Position Using "if else" Controller*

As seen in figure 14, the actuator moves in very sharp increments. It does not slow as target depth is reached (0.5 m). Therefore the depth will not stay at target. Rather, it moves from max depth (0.9 m) to the surface continuously.

A new controller was developed using the error between the target depth and measured depth as a means to dynamically change the piston position. To allow the BE to move even closer to the target depth, the derivative of the error is also included into the equation. This is commonly known as a proportional derivative, or 'PD' controller.

Piston distance = X_off – Kp * depth_error – Kd * d_depth_error

where: X_off = piston offset for neutral buoyancy
Kp = proportional error constant
Kd = derivative error constant
depth_error = target depth – actual depth + 1.075*sin(pitch*pi/180)) **
d_depth error = derivative of depth error (velocity).

** Note that the expression for the "depth error" converts the tilt condition of the vehicle to an equivalent level trim condition so the nose of the vehicle does not sink below the target depth as a result of the target depth being measured at the opposite end of the vehicle. 1.075 is the length of the BE in meters.

This controller allowed the BE to move quickly to the target depth and to remain close to this depth with small excursions. Changing and optimizing the kd and kp constants and offsets will improve the response and steady-state performance of the engine.

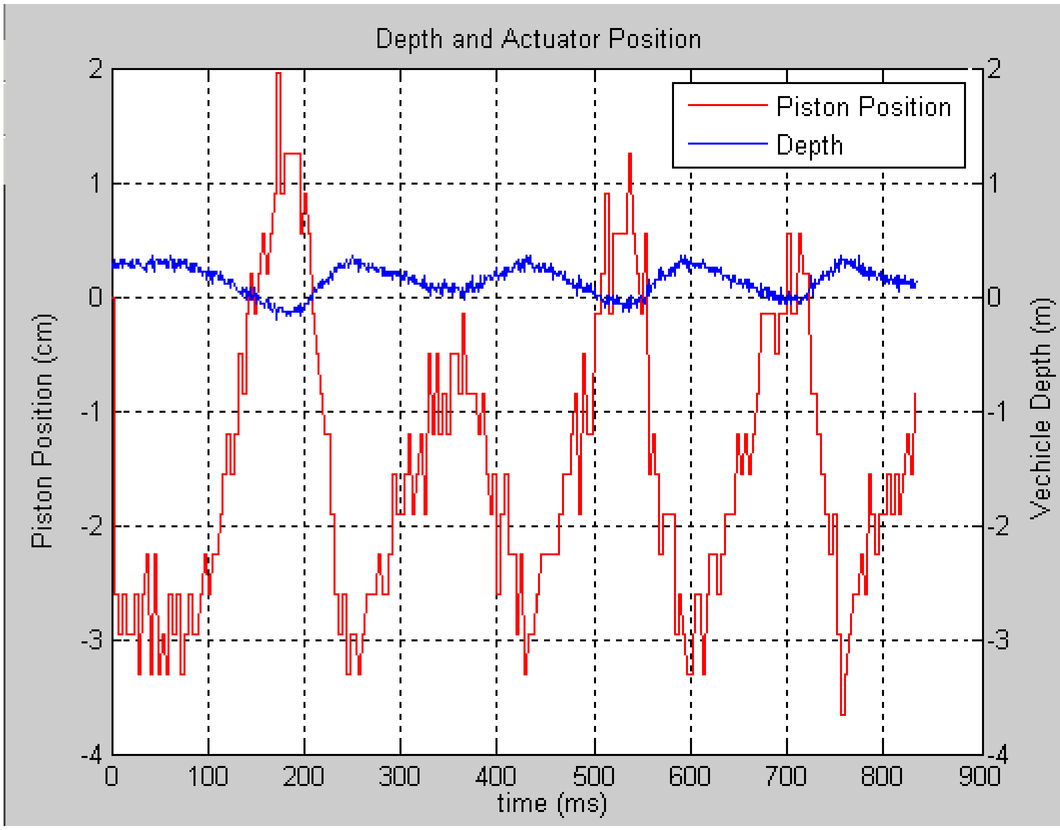


Figure 18*: Depth and Position using PD controller*

As can be seen in this figure, the piston position is dynamically changing and the depth is saying much closer to a target depth of 0.3 meters.

7.0 CONCLUSIONS AND RECOMMENDATIONS

This report has detailed and discussed the mechanics behind designing a trim and ballasting system for a Buoyancy Engine. Important Center of Mass and Center of Buoyancy relations have been described and results of experiments given. MATLAB code to calculate their position has been provided and tested. A number of battery candidates have been researched and the AA sized lithium ion batteries have proven to be the best selection. Finally a depth controller has been described and results of testing given.

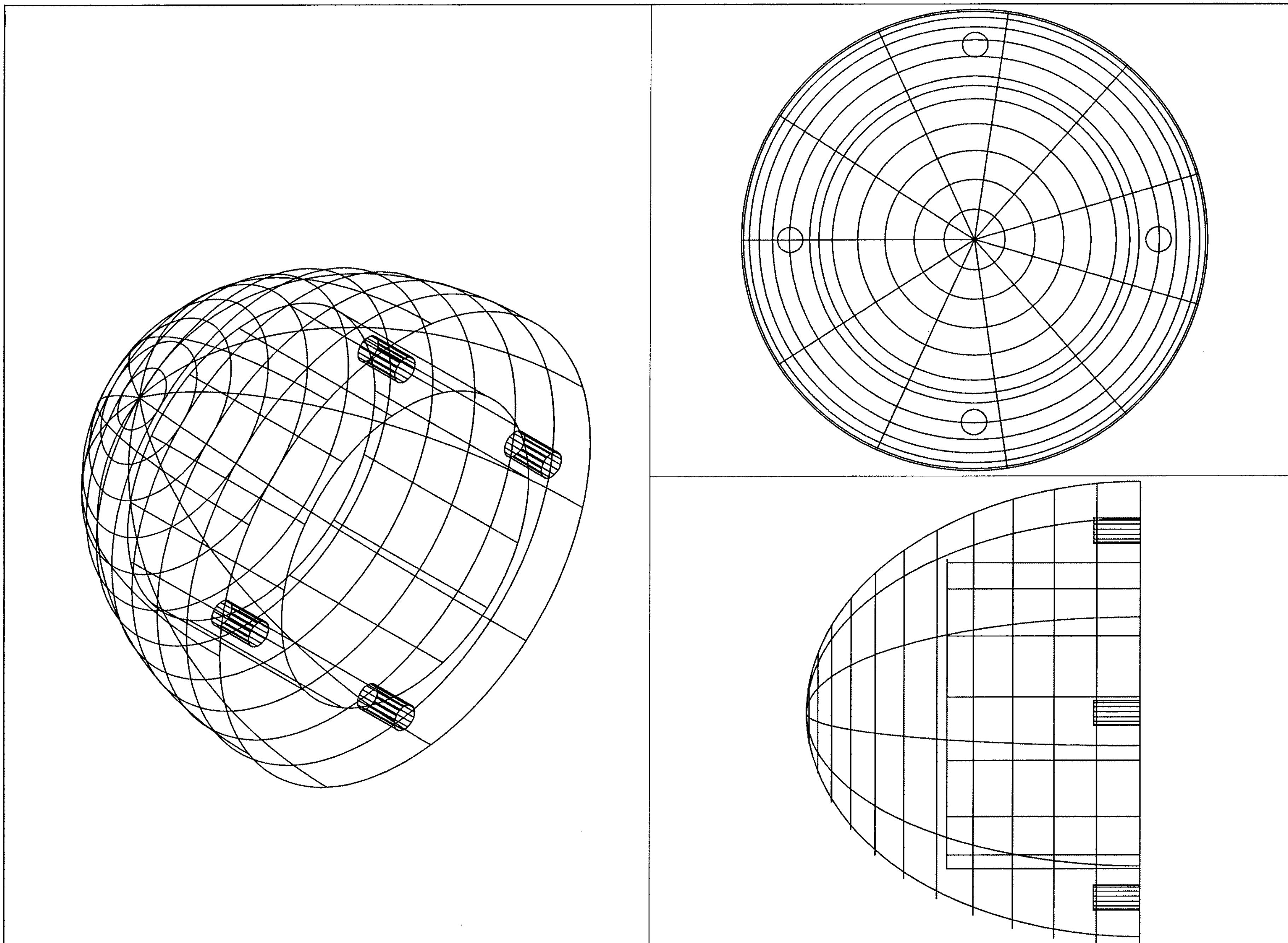
In the future it may be wise to consider the following:

- The electronics section could be extended in order to increase space for the non-moving batteries. This would also permit a larger number of batteries to be added without affecting neutral buoyancy.
- The housing should be made out of a stronger material such as aluminum. The ABS pipe is beginning to show some wear as a result of the rough modified hose clamps holding it together. A metal housing would also help prevent excess heat build-up in the engine by conducting heat to the surrounding water.
- The actuator seems to be loose at the base. It may need to be taken apart and tightened.
- As stated in section 5, a current probe is needed to determine the exact current drawn by the motor as a function of depth.
- Testing in the Tow Tank is desirable to see how the controller performs at increased depth since the Trim Tank is limited to a depth of one meter.

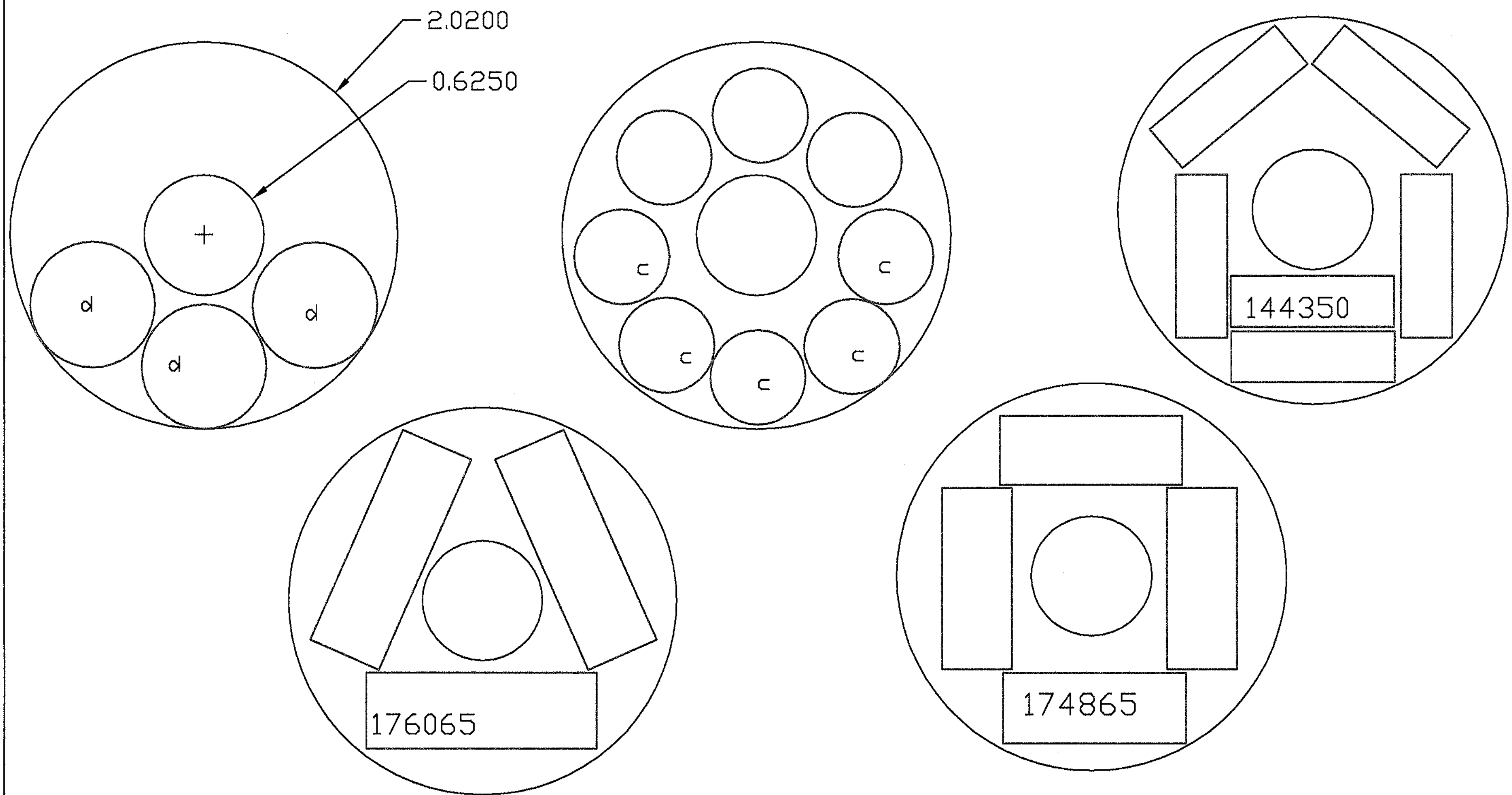
The next phase of the project will consist of purchasing the batteries and fabricating a mounting mechanism for the electronics section, as well as a moveable tray to move the battery weight back and forth along the actuator. The weight of the mounting system will have to be as light as possible in order to accommodate enough batteries. The COM of the system will need to be calculated and a pitch controller designed. As well, it may be desirable to change the coding on the Rabbit™ controller so that the controllers are on board instead of controlled through MATLAB. This will improve the response time of the engine. The details of the electronic hardware can be found in [5]. The engine should run on a timed cycle so that the delays between the electronics can be included in the design of the controller.

APPENDIX A

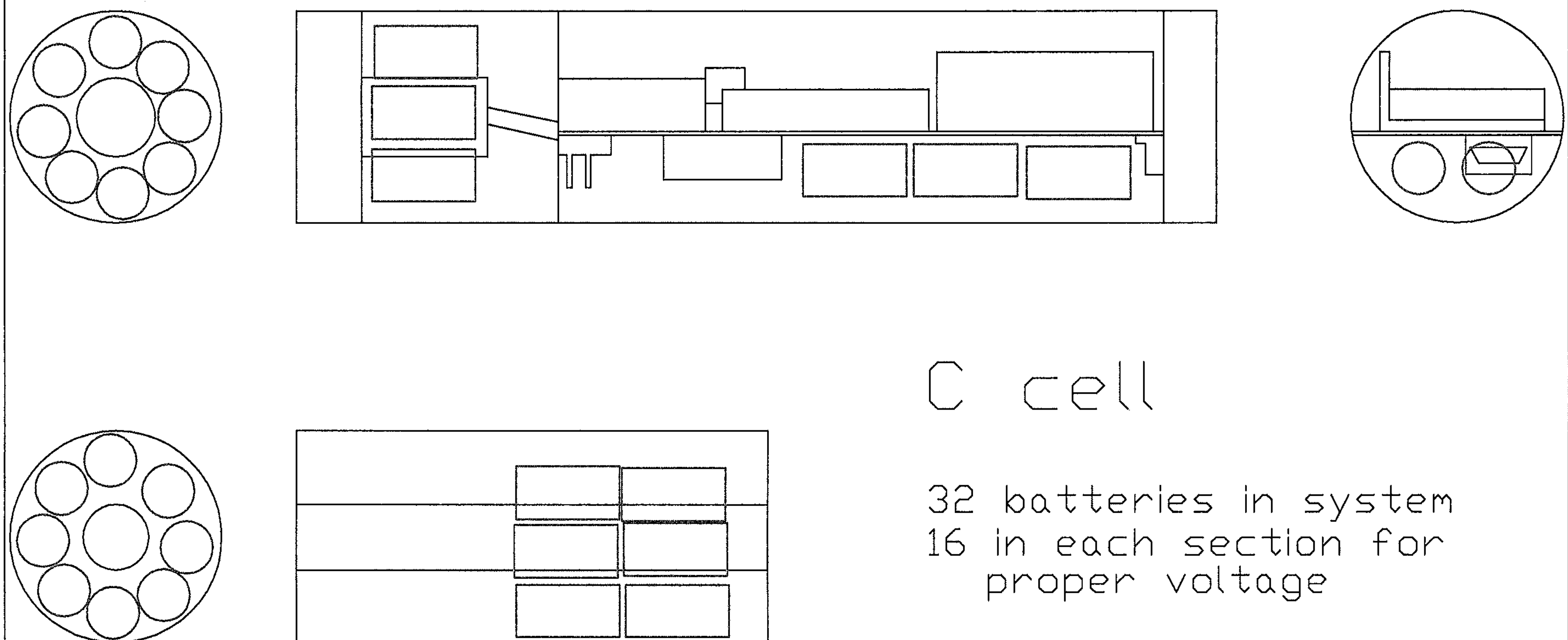
CAD DRAWINGS



NRC-IOT	May/Aug 2005	Buoyancy Engine Nosepiece
Drawn: Steven Williams	1 of 1	



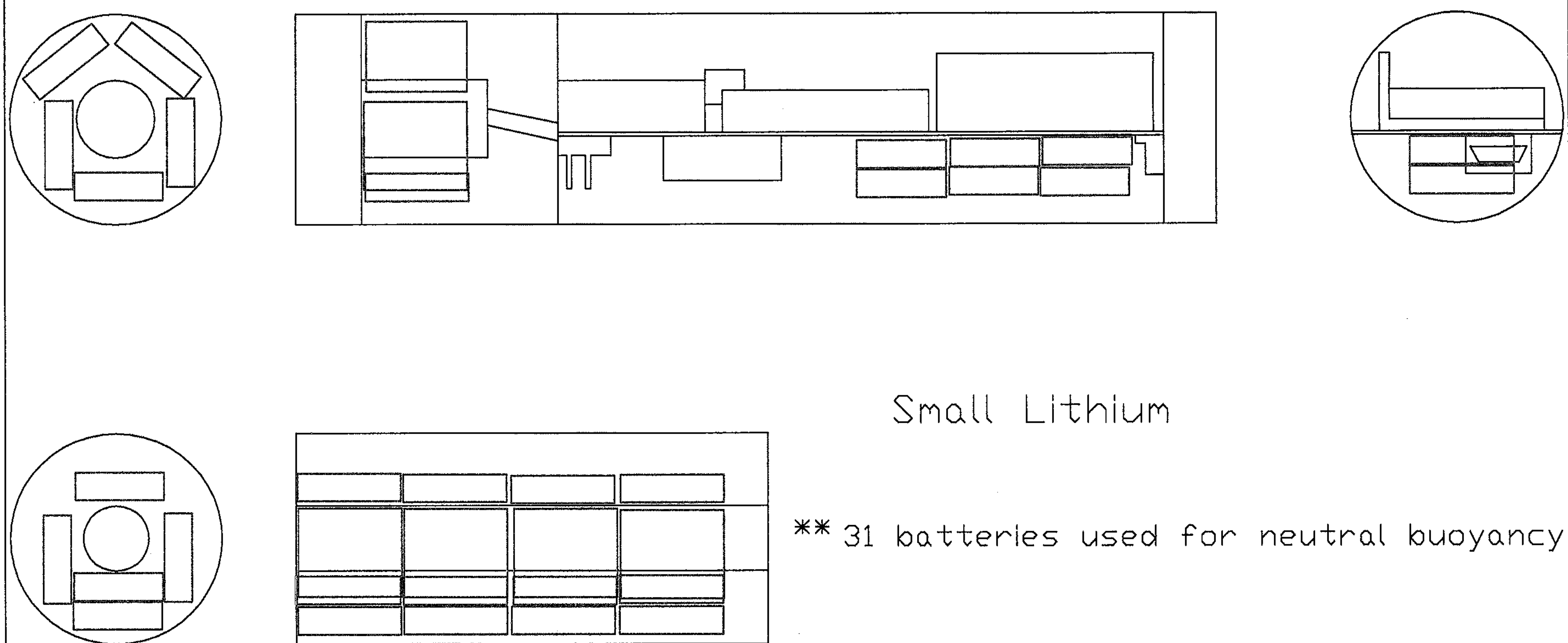
NRC-IOT	May/Aug 2005	Max Battery Configuration Around Actuator
Drawn: Steven Williams	1 of 1	



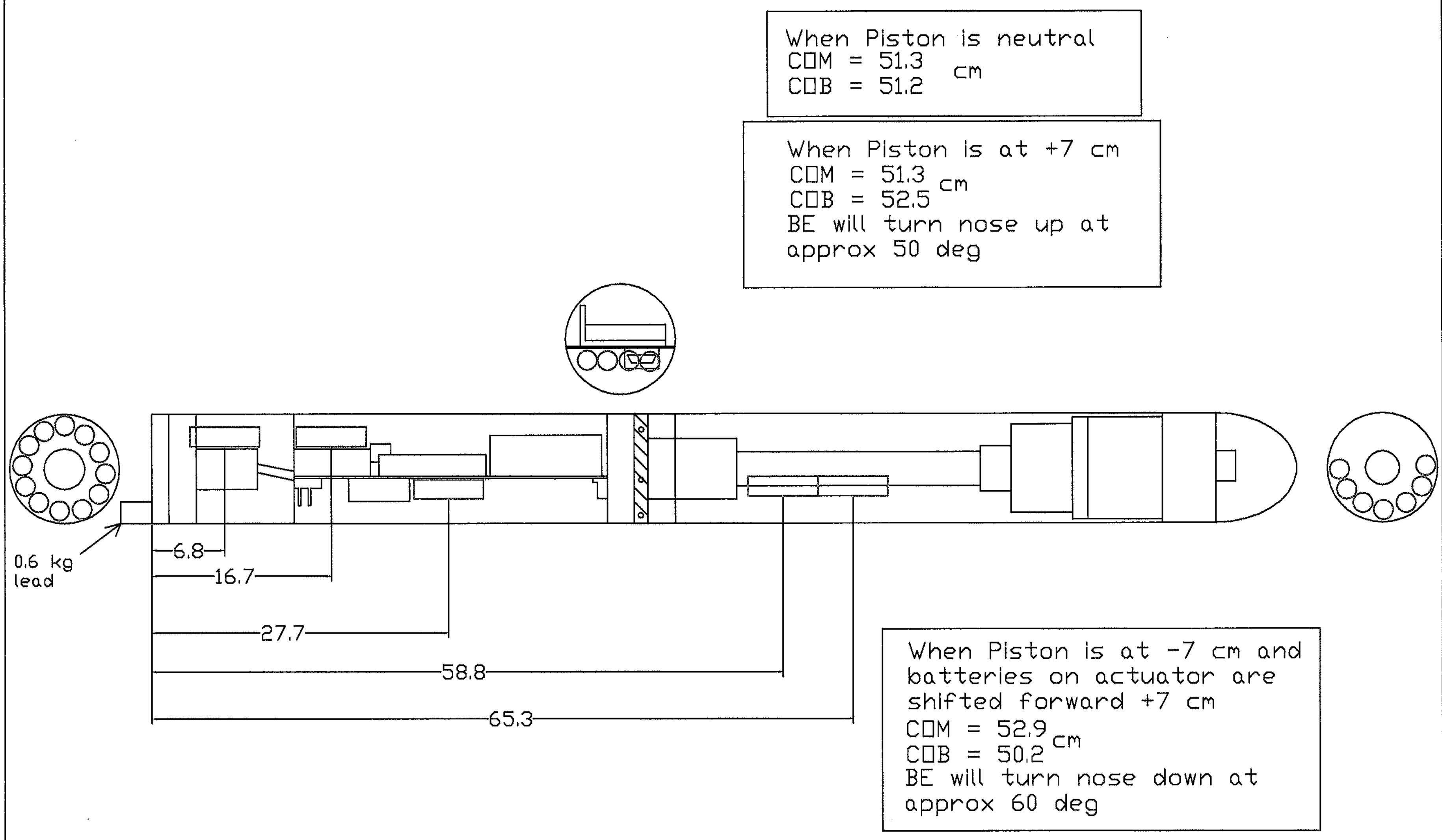
C cell

32 batteries in system
16 in each section for
proper voltage

NRC-IOT	May/Aug 2005	C Cell Batteries in Buoyancy Engine
Drawn: Steven Williams	1 of 1	



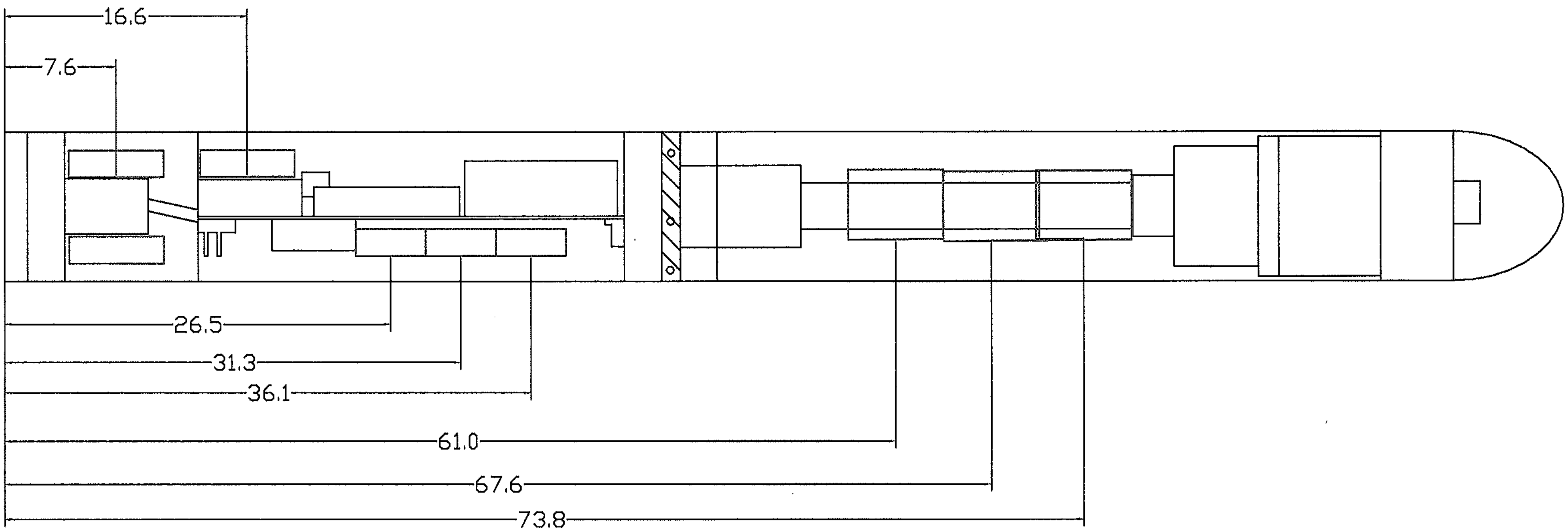
NRC-IOT	May/Aug 2005	MP 144350 Max Battery Configuration
Drawn: Steven Williams	1 of 1	



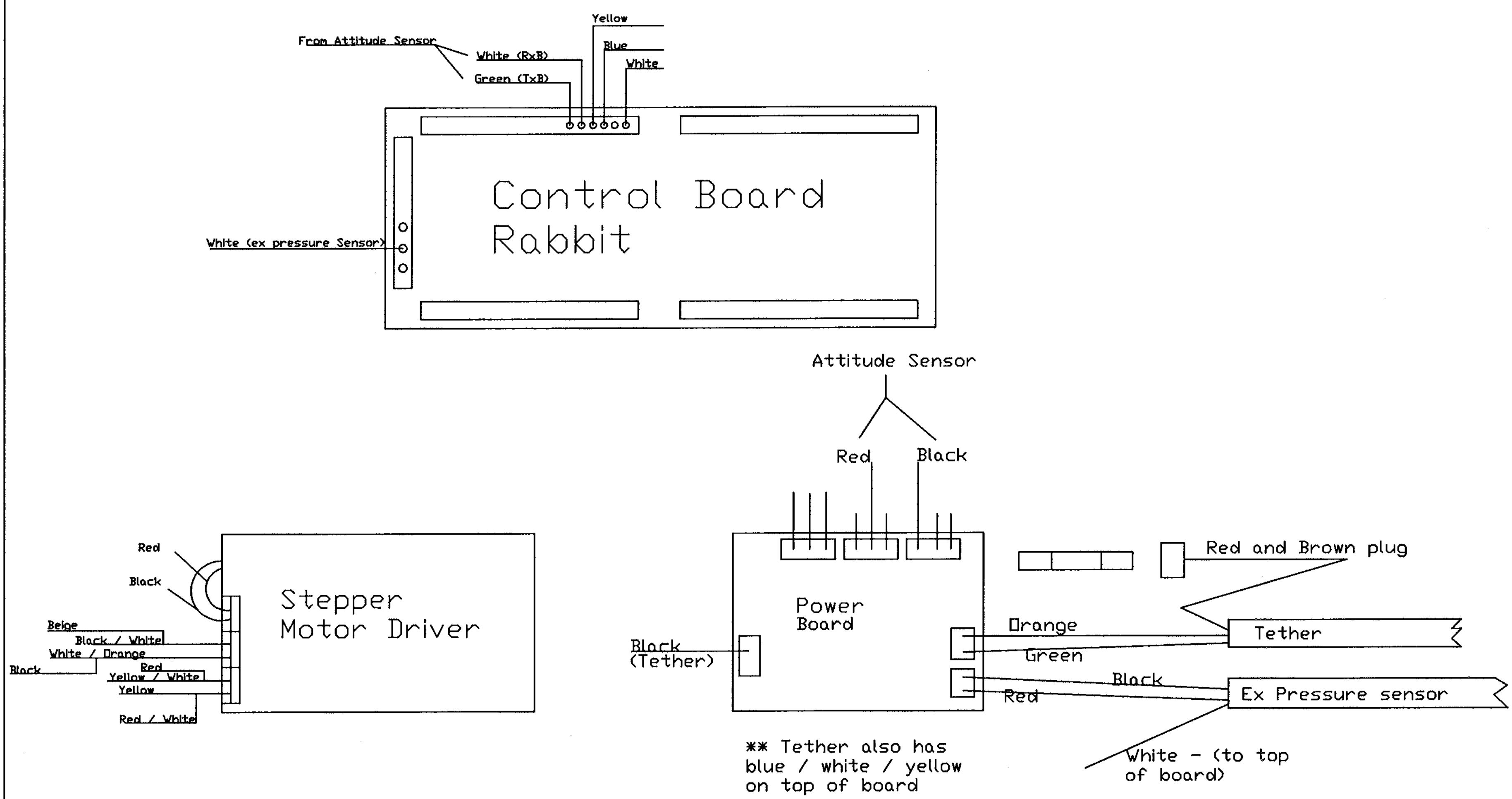
NRC-IDT	May/Aug 2005	Emoli Battery layout Providing Proper CDM and Voltage
Drawn: Steven Williams	1 of 1	

Even after moving batteries on actuator, C□M can not be located less than 53.6 cm from elec end cap

BE will not be able to turn nose up



NRC-I□T	May/Aug 2005	MP 174865 Battery Configuration in Engine
Drawn: Steven Williams	1 of 1	



NRC-IOT	May/Aug 2005	Electronics Connections Diagram
Drawn: Steven Williams	1 of 1	

APPENDIX B

MATLAB CODE

```
%---------------------------NRC-IOT----------------------------------
% Name: centerofmass.m
% Author: Steven Williams\
% Date: May 17 2005
%----------Description-----------------------------------------------
%   Calculates the center of mass of a BE hanging from 2 scales.
%   Reference point is the end cap containing pressure sensor
%
%--------------------------------------------------------------------

clear
clc
%----------input values from the linerar hanging scales----------------
F_1 = input('Please enter the mass shown on the first scale (furthest from diaphram):  ');
X_1 = input('Enter its distance from the flat end of the BE:  ');
F_2 = input('Enter the mass shown on the second scale:  ');
X_2 = input('Enter its distance from the flat end of the BE:  ');

%-------------initial values of optional weights----------------------
batt_moments_tot = 0;
batt_mass_tot = 0;
extra_moments_tot = 0;
extra_mass_tot = 0;

%===============================================
%get the number of extra weights and check input
%===============================================
amt_batt = input('Enter the total number of batteries in the system:  ' );
while amt_batt < 0
    disp('Value must be 0 or greater. Please input again:  ');
    if amt_batt < 0
        amt_batt = input('Enter the total number of batteries in the system:  ' );
    end
end

amt_extra = input ('Enter the total number of extra masses added:  ' );
while amt_extra < 0
    disp('Value must be 0 or greater. Please input again');
    if amt_extra < 0
        amt_extra = input('Enter the total number of extra masses added:  ' );
    end
end
%=================================================================
%get values for the extra weight and their distance from reference
%=================================================================

if amt_batt ~= 0
   batt_mass = input('Enter the weight of the chosen battery:  ');
   for i=1:amt_batt
        fprintf(1, 'Enter the distance to Battery %1.0f from the end cap',i);
        batt_dist(i) = input(':  ');
    end
```

```
    batt_moments = batt_mass * batt_dist;
    batt_moments_tot = sum(batt_moments);
    batt_mass_tot = batt_mass * amt_batt;
end

if amt_extra ~= 0
      for i=1:amt_extra
        fprintf(1, 'Enter the mass of extra mass %1.0f',i);
        extra_mass(i) = input(':  ');
        fprintf(1, 'Enter the distance to extra mass %1.0f from the end cap',i);
        extra_dist(i) = input(':  ');
    end
    extra_moments = extra_mass .* extra_dist;
    extra_moments_tot = sum(extra_moments);
    extra_mass_tot = sum(extra_mass);
end
%========================
%calculate center of mass
%========================
CG = ((F_1 * X_1) + (F_2 * X_2) + extra_moments_tot + batt_moments_tot) / (F_1 + F_2 +
extra_mass_tot + batt_mass_tot);

fprintf(1, 'The Center of Mass is located %3.3f cm from the end of the electronics
section',CG);
```

```
%---------------------------NRC-IOT------------------------------------
% Name: bouycenter.m
% Author: Steven Williams
% Date: June 1 2005
%----------Description-------------------------------------------------
%   Calculates the change in the center of buoyancy as the piston moves
%   its position.
%----------------------------------------------------------------------
clc
%--------------------segments of the cap-----------------------
vol_cap_1 = 57.61;
z_cap_1 = 100.89;
vol_cap_2 = 281.7;
z_cap_2 = 104.32;
%-------------------------------------------------------------

%-------------------volume of hull----------------------------
vol_hull = 10260.9;
z_hull = 50;
%------------------------------------------------------------

%  ** piston can move + or - 7 cm from its neutral position of 2.54 cm
%     inside the hull

for i=1:8
    j = -7:0;
    vol_piston(i) = 3.14159 * 3.5 * 3.5 * (j(i) - 2.54);
    z_piston (i) = 100 + ((j(i) - 2.54)/2);
end

for i=9:15
    j = 1:7;
    vol_piston(i) = 3.14159 * 3.5 * 3.5 * (j(i-8) - 2.54);
    z_piston (i) = 100 + ((j(i-8) - 2.54)/2);
end

z_v_cap1 = vol_cap_1 * z_cap_1;
z_v_cap2 = vol_cap_2 * z_cap_2;
z_v_hull = vol_hull * z_hull;

%------making an array of the COB as the piston postion changes----------
for i = 1:15
    a(i) = z_v_cap1 + z_v_cap2 + z_v_hull + (vol_piston(i) * z_piston (i));
    b(i) =  vol_cap_1 + vol_cap_2 + vol_hull + vol_piston(i);
    z(i) = ((a(i)) / (b(i))) - 51.26
end
%---------------graph of COB position----------------------------------
x = -7:7;
plot (x,z);
```

```
%---------------------------NRC-IOT------------------------------------
% Name: pointmass.m
% Author: Steven Williams\
% Date: May 17 2005
%----------Description-------------------------------------------------
%   Calculates where a point mass must be placed in oder to have center of
%   mass in the center of a Buoyancy engine
%----------------------------------------------------------------------
clear all
clc

batt_moments_tot = 0;
batt_mass_tot = 0;
extra_moments_tot = 0;
extra_mass_tot = 0;

%----------input values from the linear hanging scales----------------
F_1 = input('Please enter the mass shown on the first scale (furthest from diaphram):  ');
X_1 = input('Enter its distance from the flat end of the BE:  ');
F_2 = input('Enter the mass shown on the second scale:  ');
X_2 = input('Enter its distance from the flat end of the BE:  ');

amt_batt = input('Enter the total number of batteries in the system:  ' );
while amt_batt < 0
    disp('Value must be 0 or greater. Please input again:  ');
    if amt_batt < 0
        amt_batt = input('Enter the total number of batteries in the system:  ' );
    end
end

amt_extra = input ('Enter the number of extra masses added besides the one we are trying↙
to find:  ' );
while amt_extra < 0
    disp('Value must be 0 or greater. Please input again');
    if amt_extra < 0
        amt_extra = input('Enter the total number of extra masses added:  ' );
    end
end
%------------Input the weight and position of battery--------------------
if amt_batt ~= 0
   batt_mass = input('Enter the weight of the chosen battery:  ');
   for i=1:amt_batt
        fprintf(1, 'Enter the distance to Battery %1.0f from the end cap',i);
        batt_dist(i) = input(':  ');
    end
    batt_moments = batt_mass * batt_dist;
    batt_moments_tot = sum(batt_moments);
    batt_mass_tot = batt_mass * amt_batt;
end

%-------------Input the weight and position of extra mass----------------
if amt_extra ~= 0
```

```
      for i=1:amt_extra
        fprintf(1, 'Enter the mass of extra mass %1.0f',i);
        extra_mass(i) = input(':  ');
        fprintf(1, 'Enter the distance to extra mass %1.0f from the end cap',i);
        extra_dist(i) = input(':  ');
    end
    extra_moments = extra_mass .* extra_dist;
    extra_moments_tot = sum(extra_moments);
    extra_mass_tot = sum(extra_mass);
end
COM = 53.75;
Total_Mass = 10.15;

%=========================================================================
% calculate the point mass required position and weight required to
% put COM at 53.75 and total mass at 10.15
%=========================================================================
Point_Mass = Total_Mass - F_1 - F_2 - batt_mass_tot - extra_mass_tot;
Point_Distance = (COM * ( Total_Mass ) - (F_1 * X_1) - (F_2 * X_2) - batt_moments_tot -
extra_moments_tot) / Point_Mass;

fprintf(1, 'The mass added for neutral buoyancy must weigh %3.3f kg.',Point_Mass);
fprintf(1, 'The mass must be %3.3f cm away from the endcap.', Point_Distance);
```

```
%---------------------------NRC-IOT-------------------------------------
% Name: centerofmassDATA.m
% Author: Steven Williams
% Date: May 23 2005
%----------Description-------------------------------------------------
%   Calculates the center of mass of a BE hanging from 2 scales.
%   Reference point is the end cap containing pressure sensor
%   Receives data about batteries from a CSV file; data.txt
%   (located in same folder as the .m file)
%   Extra mases are then input by user
%-----------------------------------------------------------------------

clear
clc


a = csvread('data.txt')

%----------input values from the linear hanging scales---------------
F_1 = a(1)
X_1 = a(2)
F_2 = a(3)
X_2 = a(4)

%------------initial values of optional weights-----------------------
batt_moments_tot = 0;
batt_mass_tot = 0;
extra_moments_tot = 0;
extra_mass_tot = 0;

%===============================================
%get the number of extra weights and check input
%==============================================
amt_batt = a(5)

amt_extra = a(6)

%=================================================================
%get values for the extra weight and their distance from data.txt
%=================================================================

if amt_batt ~= 0
   batt_mass = a(7)
   for i=1:amt_batt
    batt_dist(i) = a(i+7)
    end
    batt_moments = batt_mass * batt_dist;
    batt_moments_tot = sum(batt_moments);
    batt_mass_tot = batt_mass * amt_batt;
end
%===================================================
% input the weight and position of the extra masses
```

```
%====================================================
if amt_extra ~= 0
      for i=1:amt_extra
        fprintf(1, 'Enter the mass of extra mass %1.0f',i);
        extra_mass(i) = input(':  ');
        fprintf(1, 'Enter the distance to extra mass %1.0f from the end cap',i);
        extra_dist(i) = input(':  ');
    end
    extra_moments = extra_mass .* extra_dist;
    extra_moments_tot = sum(extra_moments);
    extra_mass_tot = sum(extra_mass);
end
%=======================
%calculate center of mass
%=======================
CG = ((F_1 * X_1) + (F_2 * X_2) + extra_moments_tot + batt_moments_tot) / (F_1 + F_2 +
extra_mass_tot + batt_mass_tot);

fprintf(1, 'The Center of Mass is located %3.3f cm from the end of the electronics
section',CG);
```

```
%--------------------------NRC-IOT-----------------------------------
% Name: center_leadDATA.m
% Author: Steven Williams
% Date: June 2 2005
%----------Description-------------------------------------------------
%   Calculates the center of mass of a BE hanging from 2 scales.
%   Reference point is the end cap containing pressure sensor
%   Model using lead weights instead of batteries
%   Receives data from a CSV file: data.txt
%--------------------------------------------------------------------

clear
clc


a = csvread('data.txt')

%----------input values from the linear hanging scales---------------
F_1 = a(1)
X_1 = a(2)
F_2 = a(3)
X_2 = a(4)

%-------------initial values of optional weights-----------------------
batt_moments_tot = 0;
batt_mass_tot = 0;
extra_moments_tot = 0;
extra_mass_tot = 0;

%============================================
%get the number of extra weights and check input
%============================================
amt_batt = a(5)

amt_extra = a(6)

%================================================================
%get values for the extra weight and their distance from reference
%================================================================

if amt_extra ~= 0
      for i=1:(amt_extra)

        extra_mass(i) = a((2 * i)+5)

        extra_dist(i) = a((2 * i)+6)
      end
    extra_moments = extra_mass .* extra_dist;
    extra_moments_tot = sum(extra_moments);
    extra_mass_tot = sum(extra_mass);
end
%======================
```

```
%calculate center of mass
%========================
CG = ((F_1 * X_1) + (F_2 * X_2) + extra_moments_tot + batt_moments_tot) / (F_1 + F_2 +
extra_mass_tot + batt_mass_tot);

fprintf(1, 'The Center of Mass is located %3.3f cm from the end of the electronics
section',CG);
```

```
%---------------------------NRC-IOT----------------------------------
% Name: ser'ifelse'.m
% Author: Steven Williams
% Date: July 2 2005
%-----------------------------Description----------------------------
%  Speaks with the BE though Com 1 serial port.  Collects data from sensors
%  and stores it in the reading structure.  Ask for dive depth and then
%  proceeds to move piston until the target depth is reached.
%  Uses an "IfElse" loop to control piston
%--------------------------------------------------------------------

% Use this line before running program to create the serial port object :
%                            s = serial('COM1','baudrate',115200);


warning off all
%-retrn is used to create the effect of hitting the enter key by converting
%                               to ASCII-
retrn = [10,13];
%
%
% set header length from BE
HEADER_LENGTH = 40;

depth = input('Please enter the initial dive depth:  ');

%---Create structures with sensor categories----------------------------
start = struct('raw',[],'initial',[]);
data = struct ('raw',[],'time',[],'targetpos',[],'linearPotent',[],'PX303',[],'PX138',
[],'roll',[],'pitch',[],'yaw',[],'accelX',[],'accelY',[],'accelZ',[],'rateX',[],'rateY',
[],'rateZ',[]);
reading = struct('time',[],'Pos',[],'ExPressure',[],'depth',[],'InPressure',[],'roll',
[],'pitch',[],'yaw',[],'accelX',[],'accelY',[],'accelZ',[],'rateX',[],'rateY',[],'rateZ',
[]);
%----------------------------------------------------------------------

%----start the program on the Rabbit Controller
fopen(s)
fprintf(s, ' ')

%--------------collect and record initial constants----------------------
for j=1:HEADER_LENGTH
   start(j).raw = fscanf(s);
   %start(j).initial = start(j).raw;
end

PX303_GAIN = str2num(start(5).raw(13:23));
PX303_OFFSET = str2num(start(6).raw(15:19));
PX303_CAL_GAIN = str2num(start(24).raw(9:20));
PX303_CAL_OFFSET = str2num(start(24).raw(22:31));

PX138_GAIN = str2num(start(7).raw(14:23));
```

```
PX138_OFFSET = str2num(start(8).raw(15:20));
PX138_CAL_GAIN = str2num(start(25).raw(9:20));
PX138_CAL_OFFSET = str2num(start(25).raw(22:31));

SURF_PRESSURE = str2num(start(27).raw(20:29));

ACTUATOR_LENGTH = str2num(start(2).raw(18:23));
ZERO_POSITION = str2num(start(3).raw(16:21));
LINEAR_CAL_GAIN = str2num(start(23).raw(14:25));
LINEAR_CAL_OFFSET = str2num(start(23).raw(27:36));

MAGMETOMETER = str2num(start(11).raw(26:31));
ACCELEROMETER = str2num(start(12).raw(27:32));
GYROSCOPE = str2num(start(13).raw(23:29));
%-----------------------------------------------------------------------


count = 2; %used to move the piston every 2 cycles
retry = 1000; % used to ask if user wants a new depth after 1000 cycles

for i=1:inf
    data(i).raw = fscanf(s); % collects raw data as a string


    values = str2num(data(i).raw); % converts raw string to raw double
    %-------seperates data into structure---------------
    data(i).time = values(1);
    data(i).targetpos = values(2);
    data(i).linearPotent = values(3);
    data(i).PX303 = values(4);
    data(i).PX138 = values(5);
    data(i).roll = values(6);
    data(i).pitch = values(7);
    data(i).yaw = values(8);
    data(i).accelX = values(9);
    data(i).accelY = values(10);
    data(i).accelZ = values(11);
    data(i).rateX = values(12);
    data(i).rateY = values(13);
    data(i).rateZ = values(14);
   %----------------------------------------------------

   %----Takes raw data in structure and changes it into actual readings---
    reading(i).time = (data(i).time - data(1).time)/1000;
    reading(i).Pos = (ACTUATOR_LENGTH *(LINEAR_CAL_GAIN *(LINEAR_CAL_OFFSET-data(i).↙
linearPotent))/8.04) - ZERO_POSITION;
    reading(i).ExPressure = PX303_GAIN *((PX303_CAL_GAIN *(PX303_CAL_OFFSET - data(i).↙
PX303))- PX303_OFFSET);
    reading(i).depth = (reading(i).ExPressure - SURF_PRESSURE )/ 9.81;
    reading(i).InPressure = (PX138_GAIN * ((PX138_CAL_GAIN * (PX138_CAL_OFFSET - data(i).↙
PX138)) - PX138_OFFSET));
    reading(i).roll = data(i).roll * (360/65536);
```

```
    reading(i).pitch = data(i).pitch * (360/65536);
    reading(i).yaw = data(i).yaw * (360/65536);
    reading(i).accelX = (data(i).accelX / (32768000 / ACCELEROMETER)) * 9.81;
    reading(i).accelY = (data(i).accelY / (32768000 / ACCELEROMETER)) * 9.81;
    reading(i).accelZ = (data(i).accelZ / (32768000 / ACCELEROMETER)) * 9.81;
    reading(i).rateX = (data(i).rateX / (32768000 / GYROSCOPE)) * 9.81;
    reading(i).rateY = (data(i).rateY / (32768000 / GYROSCOPE)) * 9.81;
    reading(i).rateZ = (data(i).rateZ / (32768000 / GYROSCOPE)) * 9.81;
    %-----------------------------------------------------------------------


  if i == count % checks to see what cycle the loop is in

        if depth > reading(i).depth
            distance = reading(i).Pos - 0.02;
            fprintf(s, 'm')
            fprintf(s, distance)
            fprintf(s, sprintf('%s',retrn)) %used to print the enter command
                                            %   to the serial port
        end
        if depth < reading(i).depth
            distance = reading(i).Pos + 0.02;
            fprintf(s, 'm')
            fprintf(s, distance)
            fprintf(s,sprintf('%s',retrn))
        end

        depth_round = sprintf('%1.2f',reading(i).depth);
        depth_round = str2num(depth_round)

        if depth == depth_round
            depth = input('Please enter a new dive depth');
        end

        count = count + 2;
  end

  if i == retry
      % checks to see if user would like to change the depth of the sub
      %     after 1 minuite
        ask = input('Would you like to enter a different depth? (1 for yes, 2 for no):↙
');
        if ask == 1;
            depth = input('Please enter a new dive depth:  ');
        end
        if ask == 2
            depth = depth;
        end

        retry = retry + 1000;
  end
```

```
end
fprintf(s,'s')
fclose(s)
```

```
%---------------------------NRC-IOT------------------------------------
% Name: ser'PD'.m
% Author: Steven Williams
% Date: July 2 2005
%----------------------------Description-----------------------------
%  Speaks with the BE though Com 1 serial port.  Collects data from sensors
%  and stores it in the reading structure.  Ask for dive depth and then
%  proceeds to move piston until the target depth is reached.
%  Uses a PD control loop to control piston
%--------------------------------------------------------------------

% Use this line before running program to create the serial port object :
%                            s = serial('COM1','baudrate',115200);


warning off all
%-retrn is used to create the effect of hitting the enter key by converting
%                               to ASCII-
retrn = [10,13];
%
%
% set header length from BE
HEADER_LENGTH = 40;
%
% set Neutral Buoyancy position offset
X_off = 0;
%
% controller gain setting
k_p = 6;
k_d = 2;
%
%---Create structures with sensor categories----------------------------
start = struct('raw',[],'initial',[]);
data = struct ('raw',[],'time',[],'targetpos',[],'linearPotent',[],'PX303',[],'PX138',↙
[],'roll',[],'pitch',[],'yaw',[],'accelX',[],'accelY',[],'accelZ',[],'rateX',[],'rateY',↙
[],'rateZ',[]);
reading = struct('time',[],'Pos',[],'ExPressure',[],'depth',[],'InPressure',[],'roll',↙
[],'pitch',[],'yaw',[],'accelX',[],'accelY',[],'accelZ',[],'rateX',[],'rateY',[],'rateZ',
[]);
%-----------------------------------------------------------------------

depth = input('Please enter the initial dive depth:  ');

%----start the program on the Rabbit Controller
fopen(s)
fprintf(s, ' ')

%--------------collect and record inisital constants----------------------
for j=1:HEADER_LENGTH
   start(j).raw = fscanf(s);
   %start(j).initial = start(j).raw;
end
```

```
PX303_GAIN = str2num(start(5).raw(13:23));
PX303_OFFSET = str2num(start(6).raw(15:19));
PX303_CAL_GAIN = str2num(start(24).raw(9:20));
PX303_CAL_OFFSET = str2num(start(24).raw(22:31));

PX138_GAIN = str2num(start(7).raw(14:23));
PX138_OFFSET = str2num(start(8).raw(15:20));
PX138_CAL_GAIN = str2num(start(25).raw(9:20));
PX138_CAL_OFFSET = str2num(start(25).raw(22:31));

SURF_PRESSURE = str2num(start(27).raw(20:29));

ACTUATOR_LENGTH = str2num(start(2).raw(18:23));
ZERO_POSITION = str2num(start(3).raw(16:21));
LINEAR_CAL_GAIN = str2num(start(23).raw(14:25));
LINEAR_CAL_OFFSET = str2num(start(23).raw(27:36));

MAGMETOMETER = str2num(start(11).raw(26:31));
ACCELEROMETER = str2num(start(12).raw(27:32));
GYROSCOPE = str2num(start(13).raw(23:29));
%----------------------------------------------------------------------------

CYCLES = 1500;
count = 2; %used to move the piston every 2 cycles
retry = CYCLES; % used to ask if user wants a new depth after 1000 cycles
%
%tic; not needed now
%TIME_0 = toc;
%
for i=1:CYCLES;%inf
    i;
    data(i).raw = fscanf(s); % collects raw data as a string


    values = str2num(data(i).raw); % converts raw string to raw double
    %-------seperates data into structure---------------
    data(i).time = values(1);
    data(i).targetpos = values(2);
    data(i).linearPotent = values(3);
    data(i).PX303 = values(4);
    data(i).PX138 = values(5);
    data(i).roll = values(6);
    data(i).pitch = values(7);
    data(i).yaw = values(8);
    data(i).accelX = values(9);
    data(i).accelY = values(10);
    data(i).accelZ = values(11);
    data(i).rateX = values(12);
    data(i).rateY = values(13);
    data(i).rateZ = values(14);
   %-------------------------------------------------------
```

```
    %----Takes raw data in structure and changes it into actual readings---
     reading(i).time = (data(i).time - data(1).time)/1000;
     reading(i).Pos = (ACTUATOR_LENGTH *(LINEAR_CAL_GAIN *(LINEAR_CAL_OFFSET-data(i).↙
linearPotent))/8.04) - ZERO_POSITION;
     reading(i).ExPressure = PX303_GAIN *((PX303_CAL_GAIN *(PX303_CAL_OFFSET - data(i).↙
PX303))- PX303_OFFSET);
     reading(i).depth = (reading(i).ExPressure - SURF_PRESSURE )/ 9.81;
     reading(i).InPressure = (PX138_GAIN * ((PX138_CAL_GAIN * (PX138_CAL_OFFSET - data(i).↙
PX138)) - PX138_OFFSET));
     reading(i).roll = data(i).roll * (360/65536);
     reading(i).pitch = data(i).pitch * (360/65536);
     reading(i).yaw = data(i).yaw * (360/65536);
     reading(i).accelX = (data(i).accelX / (32768000 / ACCELEROMETER)) * 9.81;
     reading(i).accelY = (data(i).accelY / (32768000 / ACCELEROMETER)) * 9.81;
     reading(i).accelZ = (data(i).accelZ / (32768000 / ACCELEROMETER)) * 9.81;
     reading(i).rateX = (data(i).rateX / (32768000 / GYROSCOPE)) * 9.81;
     reading(i).rateY = (data(i).rateY / (32768000 / GYROSCOPE)) * 9.81;
     reading(i).rateZ = (data(i).rateZ / (32768000 / GYROSCOPE)) * 9.81;
     %-------------------------------------------------------------------
     % depth is desired depth as put by user
     % reading.depth is depth as available from BE
   depth_error(i) = depth - reading(i).depth;
   %
   if i == count % checks to see what cycle the loop is in --> entering control cycle
       % depth_error = depth - reading.depth ----> - k_p
       depth_error(i) = depth - ( reading(i).depth + 1.075*sin(reading(i).pitch*pi/180));
       if i>1
           d_depth_error(i) = (depth_error(i)-depth_error(i-1))/(reading(i).time - reading↙
(i-1).time);
       else
           d_depth_error(i) = 0;
       end
       % distance is absolute position of the piston
       distance(i) = X_off - k_p*depth_error(i) - k_d * d_depth_error(i);
       fprintf(s, 'm');
       fprintf(s, distance(i));
       fprintf(s, sprintf('%s',retrn));%used to send the enter command
                                              %   to the serial port
       depth_round = sprintf('%1.2f',reading(i).depth);
       depth_round = str2num(depth_round)
       %
       count = count + 2;
   elseif i>1
       distance(i) = distance(i-1);
   else
       distance(i) = 0;
   end

   if i == retry
       % checks to see if user would like to change the depth of the sub
       %     after 1 minuite
```

```
        ask = input('Would you like to enter a different depth? (1 for yes, 2 for no):
');
        if ask == 1;
            depth = input('Please enter a new dive depth:  ');
        end
        if ask == 2
            depth = depth;
        end

        retry = retry + 1000;
  end

end
fprintf(s,'s')
%fclose(s)
```

```
%---------------------------NRC-IOT-----------------------------------
% Name: graph.m
% Author: Steven Williams
% Date: July 30 2005
%-----------------------------Description-----------------------------
% Takes the data from serPD.m or ser'ifelse'.m and places it in doubles.
% A graph of pitch, roll, and yaw is created, as well as a graph of piston
% position and depth
%---------------------------------------------------------------------


k = input('What is the number of values in the reading struct?  ');

for i=1:k
    time(i) = reading(i).time;
    pos(i) = reading(i).Pos;
    ExPressure(i) = reading(i).ExPressure;
    depth(i) = reading(i).depth;
    InPressure(i) = reading(i).InPressure;
    roll(i) = reading(i).roll;
    pitch(i) = reading(i).pitch;
    yaw(i) = reading(i).yaw;
    accelX(i) = reading(i).accelX;
    accelY(i) = reading(i).accelY;
    accelZ(i) = reading(i).accelZ;
    rateX(i) = reading(i).rateX;
    rateY(i) = reading(i).rateY;
    rateZ(i) = reading(i).rateZ;
end

plot(time,pos,time,depth)
figure
plot(time,roll,time,pitch,time,yaw)
```

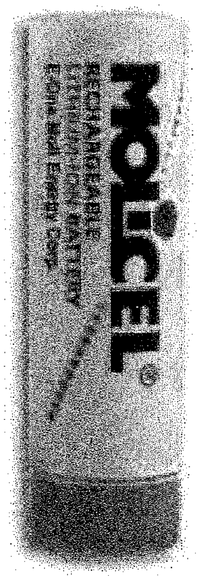
APPENDIX C

SPECIFICATION SHEETS

Product Data Sheet

MODEL: ICR-18650J pg 1/2

E-ONE MOLI ENERGY



Custom design the MOLICEL® lithium-ion rechargeable battery into your mobile device.

The ICR-18650J cylindrical cell consists of a lithium cobalt positive electrode material and a graphitic carbon negative electrode providing 3.75V and volts and 2400mAh. This cell design requires pack control circuitry.

LITHIUM-ION RECHARGABLE BATTERY

CELL SPECIFICATIONS	
Capacity (Nominal)	2400 mAh
Nominal Voltage	3.75 V
Energy	8.9 Wh
Size (mm)	18.24 x 65 mm
Weight	47 grams
Energy Density Volumetric Gravimetric	 520 Wh/l 188 Wh/kg
OPERATING SPECIFICATIONS	
Operating Voltage	4.2 V to 3.0 V
Charge Voltage	4.2 V ± 50 mV
Cutoff Voltage	2.5 V
Temperature Range Discharge Charge	 -20 °C to 60 °C 0 °C to 45 °C
Maximum Discharge	4.0 A
Maximum Charge	2.4 A
STORAGE SPECIFICATIONS	
Temperature Range Recommended < 25 °C for extended storage periods.	-20 °C to 60 °C
Recommended Voltage Range	4.2 V to 2.5 V

Please refer to Page 2 for cell discharge characteristics.

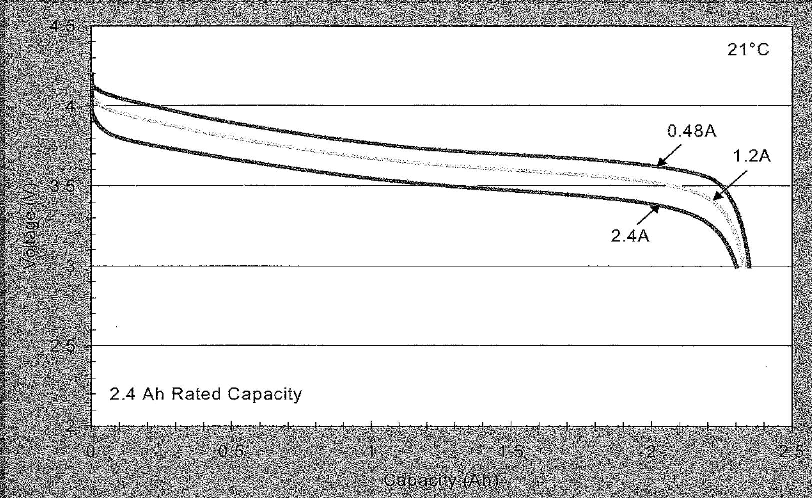
Issue Date: February 2006
Information subject to change without notice.

E-ONE MOLI ENERGY (CANADA) LTD.
20,000 Stewart Crescent
Maple Ridge, B.C. Canada V2X9E7
Tel: 604466-6654 or 800-664-MOLI
Fax: 604-466-6600
Site:www.molienergy.com

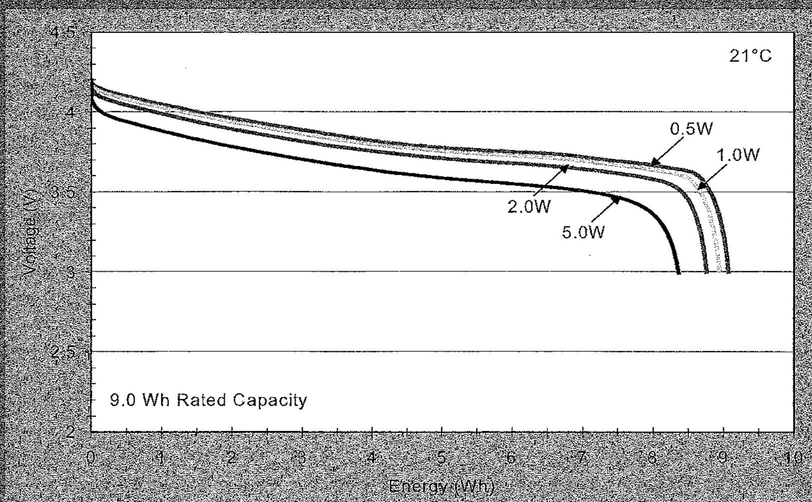
E-ONE MOLI ENERGY CORP.
7F, No.113, Sec 2, Zhongshan N. Rd.
Taipei City 104, Taiwan, R.O.C.
Tel: 886-2-25673500
Fax: 886-2-25676500
Site: www.e-one.com.tw

Product Data Sheet
MODEL: ICR-18650J pg 2/2

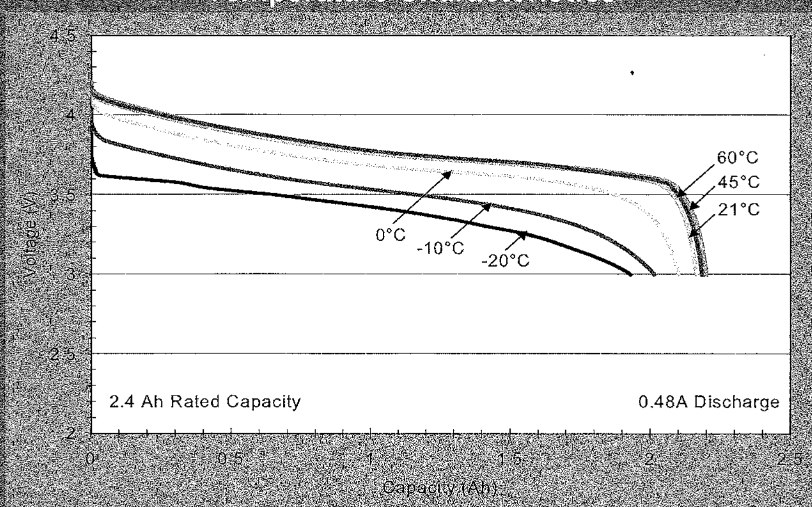
Constant Current Characteristics



Constant Power Characteristics



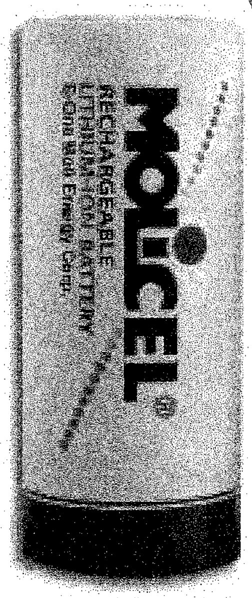
Temperature Characteristics



E-ONE MOLI ENERGY

PRODUCT DATA SHEET
MODEL : IMR-26700

MOLICEL®



Custom design the MOLICEL® lithium-ion rechargeable battery into your mobile device.

The IMR-26700 cylindrical cell consists of a manganese dioxide positive electrode material and a graphitic carbon negative electrode providing 3.8 volts and 2900 mAh.

LITHIUM-ION RECHARGEABLE BATTERY

CELL SPECIFICATIONS	
Capacity (Nominal)	2900 mAh
Nominal Voltage	3.8 V
Energy	11 Wh
Size	φ26.4 x 70 mm
Weight	100 grams
Energy Density Volumetric Gravimetric	 285 Wh/l 110 Wh/kg
OPERATING SPECIFICATIONS	
Operating Voltage	4.2 V to 3.0 V
Charge Voltage	4.2 V ± 50 mV
Cutoff Voltage	2.5 V
Temperature Range Discharge Charge	 -20°C to 60°C 0°C to 45°C
Maximum Discharge Current (Continuous)	15.0 A
Maximum Charge Current	3.0 A
STORAGE SPECIFICATIONS	
Temperature Range Recommended < 25°C for extended storage periods.	-20°C to 60°C
Recommended Voltage Range	4.2 V to 3.0 V

Information subject to change without notice.

E-ONE MOLI ENERGY (CANADA) LTD.
20,000 Stewart Crescent
Maple Ridge, B.C. Canada V2X 9E7
Tel: 604-466-6654 or 800-664-MOLI
Fax: 604-466-6600

Site:www.molienergy.com

E-ONE MOLI ENERGY CORP.
3F, No.147 Kang-Ning St., Hsi-Chih City
Taipei County, Taiwan, R.O.C.
Tel: 886-2-26926755
Fax: 886-2-26926757
Email:eonemoli@e-one.com.tw
Site:www.e-one.com.tw

Library Sort	**Product Specifications**	VER	II
Library Name	**Cylindrical Li-ion Rechargeable Battery**	Date	2004/4/30

Cylindrical Li-ion battery

Specification

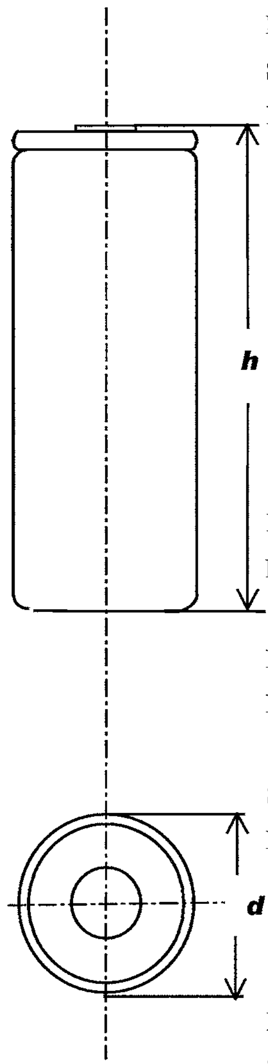
Type: ICR18650

Prepared/Date	Auditing/Date	Approved/Date
Genxiao Li	Peng kun Gao	Dragon Lv

Batteryspace.com

Library Sort	**Product Specifications**	VER	II
Library Name	**Cylindrical Li-ion Rechargeable Battery**	Date	2004/4/30

DATA SHEET



TYPE ------------------------------ CYLINDRICAL

MODEL -------------------------------- ICR18650

SPECIFICATION---------------------------- 18650

Nominal voltage --------------------------- 3.7V

Weight----------------------------------- □ 45g

C_5mAh----------------------------------- 2000mAh

Charge voltage-------------------- 4.200±0.049V

Minimum discharge end voltage------------ 2.75V

Maximum charge voltage------------------- 4.20V

Maximum continuous charge current-------- 1500mA

Maximum continuous discharge current----- 3000mA

Dimension (including shrink sleeve/label)

Diameter□d ------------------------ 18.2±0.2mm

height□h -------------------------- 64.5±0.5mm

Capacity (20°C, 0.2 C_5 to 2.75V)

Minimum capacity------------------------ 2000mAh

Internal impedance□ 20□±5□□□80mΩ

Charge conditions (20□±5□)

Standard charge--------------------- 900mA CC/CV

Fast charge------------------------ 1500mA CC/CV

Operation conditions□recommended□

Storage---temperature(15-35□)

Relative humidity(45-75%)

Pressure(86-106Kpa)

Discharge------------------------------- -20-60□

Standard charge--------------------------- 0-45□

Standard Test Conditions (Except additional quest)

Temperature---------------------------- 20□±5□

Relative humidity------------------------ 65±20%

Subject to change without prior notice

Batteryspace.com

Library Sort	**Product Specifications**	VER	II
Library Name	**Cylindrical Li-ion Rechargeable Battery**	Date	2004/4/30

1. Performance

Test item	Test conditions	Requirements
(1)Outside Appearance	Visual check	No abnormal stain, Deformation nor damage
(2) Standard test conditions	Measurements are carried out at 20 ± 5 ℃ and relative humidity of 65±20% without other specified condition. Accuracy of voltmeters and ammeters used in test is equal to or better than the grade 0.5.	
(3) Standard charge	Battery is charged continuously at the constant current of 0.5 I_t end at voltage of 4.2V, then charge at the constant voltage of 4.2V until the end current of 20mA after Pre-discharge at the constant current of 0.2 I_t until the end voltage of 2.75V/cell	
(4) Fast charge	Charge shall be conducted continuously at the constant current of 1500mA until the end voltage of 4.2V, then charge at the constant voltage of 4.2V until the end current of 20mA after Pre-discharge mentioned in Item (2).	
(5) Open-circuit voltage (OCV)		≥3.75V
(6)Rated Capacity	Discharge duration of the charged battery specified in Item (3) shall be measured at 0.2 I_t mA until the end voltage of 2.75V/cell, after rest for 0.25 hour. If the discharge duration does not reach the specified value, the test may be repeated up to three times in total.	Rated capacity: ≥100%C_5mAh
(7)Capacity high-rate discharge	Discharge duration of the charged battery specified in Item (3) shall be measured at 1500mA until the end voltage of 2.75V/cell, after rest for 0.25 hour. If the discharge duration does not reach the specified value, the test may be repeated up to three times in total.	Discharge capacity: ≥90%C_5mAh
(8) Cycle Life (20 ℃)	Carry out cycles (1500mA CC/CV(4.2V), discharge at the constant current of 1500mA after rest for 0.25 hour) at20± 2℃. The test end until the discharge capacity ≤ 60%C_5mAh	≥300 cycles
(9)Low temperature discharge	1)charge shall be conducted at Item (3)；2)The battery shall be stored under -20℃±2℃ for 16h～24h；3)Discharge shall be conducted at the constant current of 0.2I_tmA until the end voltage of 2.75V/cell。	Discharge capacity： ≥60%C_5mAh

2 Mechanical test

Test Item	Test Conditions	Requirements
(1)Vibration Test	Vibrate test sample for 90minutes each at room temperature after rated charge. Amplitude: 0.38mm(10-30Hz)，0.19mm（30-55Hz） Frequency: 10-55Hz(1oct/min)	No rupture, fire, smoke, Nor critical damage ≥90% C_5mAh

	Direction: X, Y Then measure resistance, voltage of battery and check outside appearance.	
(2) Drop Test	Drop 100% charged test sample from 1m above onto concrete board with more than 5cm thickness two times each for every direction at room temperature. Then measure rated capacity and checks outside appearance.	No rupture, fire, smoke, Nor critical damage ⌋90% C_5mAh

3 Safety evaluation

Test Item	Test Conditions	Requirements
(1) Hot Oven Test	The charged battery is to be heated in a gravity convection or circulating air oven. The temperature of the oven is to be raised at a rate of 5±2⌋ per minute. The oven is to remain for 30 minutes at 150±2⌋ before the test is discontinued.	No fire, Nor explosion
(2)Short Circuit Test	After fast charge at 20±2⌋, Connect battery terminals with electric wire (electric resistance: 50mΩ or less). And stop the test when the temperature of battery is 10⌋ lower than peak temperature.	No fire, Nor explosion
(3) Overcharge Test	After discharged at 1 I_t mA and end at 2.75V, the battery shall be charged at 3 I_t mA current with a voltage limit of 4.6V.	No fire, Nor explosion
(4)Dip test	The charged battery shall be dipped in water for 24h in an ambient temperature of 20⌋±5⌋.	No fire, Nor explosion

4 Charge State of Battery before shipment

To be determined.(Recommendation Approx. 3.75 – 3.85V 30% charge)

5 Duration of guarantee the product

We can keep on the quality in six month.

6 Handling precautions on Lithium Ion Rechargeable Battery

To assure product safety, describe the following precautions in the instruction manual of the equipment.

! Danger

- When charging the battery, use dedicated chargers and follow the specified conditions.
- Use the battery only in the specified equipment.
- Do not connect battery directly to an electric outlet or cigarette lighter charger.
- Do not heat or throw battery into a fire.
- Do not use, leave battery close to fire or inside of a car where temperature may be above 60□. Also do not charge / discharge in such conditions.
- Do not immerse, throw, and wet battery in water/ seawater.
- Do not put batteries in your pockets or a bag together with metal objects such as necklaces. Hairpins,

coins, or screws. Do not store batteries with such objects.

- Do not short circuit the (+) and (-) terminals with other metals.
- Do not place battery in a device with the (+) and (-) in the wrong way around.
- Do not pierce battery with a sharp object such as a needle.
- Do not hit with a hammer, step on or throw or drop to cause strong shock.
- Do not disassemble or modify the battery.
- Do not solder a battery directly.
- Do not use a battery with serious scar or deformation.

! Warning

- Do not put battery into a microware oven, dryer, or high-pressure container.
- Do not use battery with dry cells and other primary batteries, or batteries of a different package, type, or brand.
- Stop charging the battery if charging is not completed within the specified time.
- Stop using the battery if abnormal heat, odor, discoloration, deformation or abnormal condition is detected

During use, charge, or storage.

- Keep away from fire immediately when leakage or foul odor is detected.
- If liquid leaks onto your skin or clothes, wash well with fresh water immediately.

If liquid leaking from the battery gets into your eyes, do not rub your eyes. Wash them well with clean water and go to see a doctor immediately.

! Caution

- Store batteries out of reach of children so that they are not accidentally swallowed.
- If younger children use the battery, their guardians should explain the proper handling.
- Before using the battery, be sure to read the user's manual and cautions on handling thoroughly.
- Thoroughly read the user's manual for the charger before charging the battery.
- For information on installing and removing from equipment, thoroughly read the user's manual for the specific equipment.
- Batteries have life cycles. If the time that the battery powers equipment becomes much shorter than usual, the battery life is at an end. Replace the battery with a new same one.
- Remove a battery whose life cycle has expired from equipment immediately.
- When the battery is thrown away, be sure it is non-conducting by applying vinyl tape to the (+) and (-) terminals.
- When not using battery for an extended period, remove it from the equipment and store in a place with low humidity and low temperature.
- While the battery pack is charged, used and stored, keep it away from objects or materials with static electric charges.
- If the terminals of the battery become dirty, wipe with a dry clothe before using the battery.
- The battery can be used within the following temperature ranges. Do not exceed these ranges.
  Charge temperature range : 0□ to 45□
  Discharge temperature range : -20□ to 60□
  (When using equipment)

Rechargeable lithium-ion battery
MP 176065

High performance
Medium prismatic cell

Benefits

- Extended autonomy and life for mobile systems
- Recommended for ruggedized designs
- Easy integration into compact and light systems

Key features

- Very high energy density (350 Wh/l and 165 Wh/kg)
- Unrivalled low temperature performance
- Excellent charge recovery after long storage, even at high temperature
- Maintenance free
- Long cycle life (over 70% capacity after 500 cycles 100% DoD)

Applications

- Mobile asset tracking
- Rack-mount telecom batteries
- Small UPS
- Future soldier equipment
- Portable radios
- Portable defibrillators
- Professional portable lighting
- Electric bikes and personal mobility

Electrical characteristics	
Nominal voltage (V)	3.75
Typical capacity 20°C (Ah)	6.8 Ah @ 4.2 V *6.1 Ah @ 4.1 V*
Mechanical characteristics	
Thickness max at end of life (mm)	19.8
Width max (mm)	61
Height max (mm)	65
Typical weight (g)	155
Lithium equivalent content (g)	2.0
Volume (cm³)	73
Operating conditions	
Charging method	Constant Current/Constant Voltage (CCCV)
Charging voltage	4.20 V +/- 0.05 V per cell
Max recommended charging current* (A)	6.8 (C rate)
Charging temperature range*	– 20°C to + 60°C
Timer @ 20°C	To be set as a function of the charging current: 1C → 2 to 3 h 0.5C → 3 to 4 h 0.2C → 6 to 7 h
Max continuous discharge current (A)	13.6 (2C rate)
Pulse discharge current (A)	up to 27 (4C rate)
Discharge cut off voltage (V)	2.5
Discharge temperature range	– 50°C to + 60°C

** Consult Saft for optimized charging below 0°C*

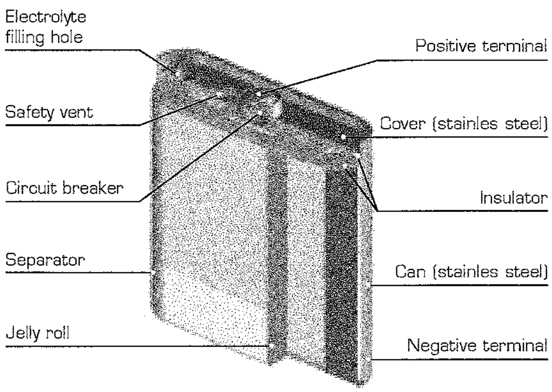
March 2005

saft

MP 176065

Technology

- Graphite-based anode
- Lithium Cobalt oxide-based cathode
- Electrolyte: organic solvents
- Built-in redundant safety protections
- Batteries assembled from MP cells feature an electronic protection circuit



Built-in protection devices ensure safety in case of:

- Exposure to heat
- Exposure to direct sunlight for extended periods of time
- Short circuit
- Overcharge
- Overdischarge

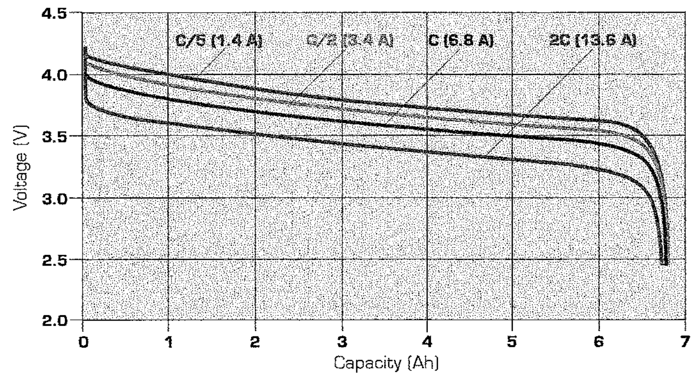
When handling Saft MP batteries:

- Do not solder directly to cell terminal
- Do not disassemble
- Do not remove the protection circuit
- Do not incinerate

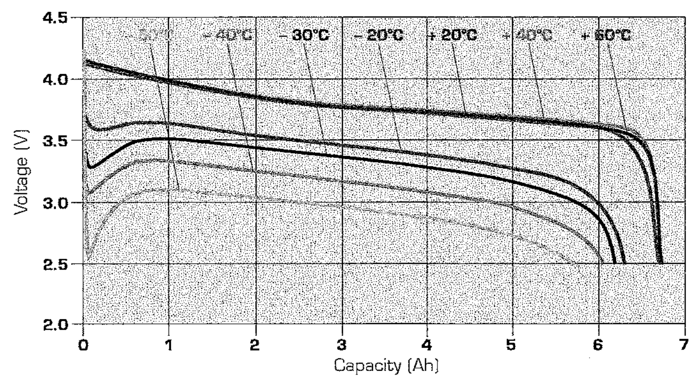
Transportation and storage:

- Store in a dry place at a temperature preferably not exceeding 30°C
- For long-term storage, keep the battery within a (30 ± 15) % state of charge

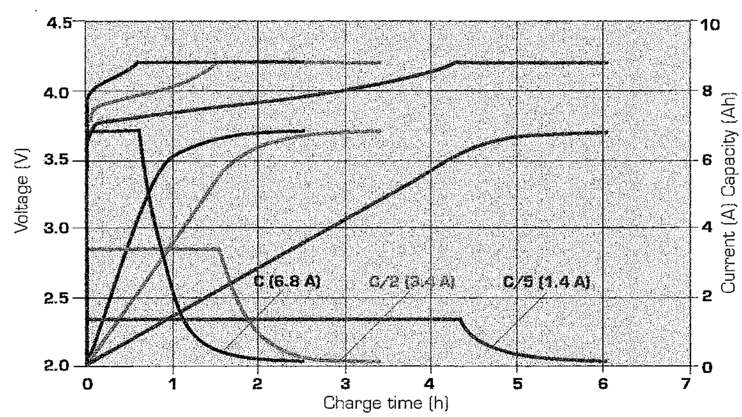
Discharge versus current at 20°C



Discharge characteristics at C/5 rate



Charge characteristics to 4.2 V at +20°C at C, C/2 and C/5 rates



Saft
12, rue Sadi Carnot
93170 Bagnolet - France
Tel +33 1 49 93.19 18
Fax +33 1 49 93 19 69

313, Crescent Street
Valdese NC 28690 - USA
Tel +1 828 874 41 11
Fax +1 828 879 39 81

www.saftbatteries.com

Doc. N° 54037-2-0305
Published by the Communications Department

Information in this document is subject to change without notice and becomes contractual only after written confirmation by Saft.

Photo credit: Saft
Produced by Arthur Associates

Société anonyme au capital de 31 944 000€
RCS Bobigny B 383 703 873

Rechargeable lithium-ion batteries
MP Series

High performance
Medium prismatic cells

Benefits

- Very high specific energy
- Maintenance free
- Long cycle life
  (> 1000 cycles with more than 80% of initial capacity remaining)

Applications

- Telecommunications
- Medical devices
- Instrumentation
- Professional video
- Ruggedized computers
- Military

etc ...

Technology

- Graphite-based anode
- Lithium Cobalt Oxide-based cathode
- Electrolyte : organic solvents
- Electronic protection circuit included

Electrical characteristics	MP 144350	MP 174865	MP 176065
Nominal voltage (V)	3.6	3.6	3.6
Typical capacity at C/2 rate @ 4.1 V, + 20°C (Ah)	2.3	4.6	5.8
Mechanical characteristics			
Thickness max (mm)	13.5	18.5	18.5
Width max (mm)	43	48	60
Height max (mm)	50	65	65
Weight max (g)	70	125	151
Volume (cm3)	25.9	47.5	62.2
Operating conditions			
Charge method	Constant Current/Constant Voltage (CCCV)		
Charging voltage	4.1 V +/- 0.04 V per cell		
Charging current	1C max., C/2 recommended		
Operating temperature			
Charge*	0°C to +50°C		
Discharge	-40°C to +60°C		
*speak to Saft for charging below 0°C			
End of charge detection			
Low current	30 mA	70 mA	100 mA
Timer	To be set as a function of the charging current: 1C → 3 to 4 h; 0.5C → 4 to 5 h; 0.2C → 7 to 8 h		

June 2004

MP Series

Precautions

For the safe performance of all Saft MP Li-ion batteries, observe the following precautions.

When handling Saft MP batteries:

- Do not expose to heat or flame.
- Do not disassemble or alter.
- Do not leave in vehicles or in direct sunlight for extended periods of time.
- Do not short circuit.
- Do not allow metal objects to come into contact with battery terminals.
- Do not immerse in water.
- Do not drop or subject batteries to shock.
- Do not solder directly to batteries.
- Do not remove the protection circuit.

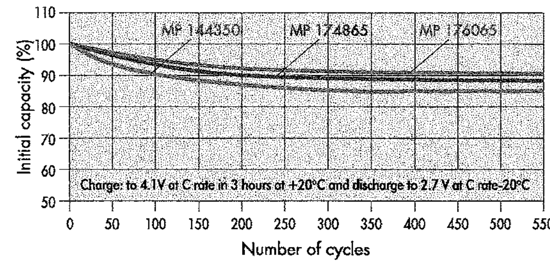
When charging and discharging Saft MP batteries:

- Use only a Saft-approved charger.
- Do not reverse battery leads when charging.
- Do not overcharge.
- Do not use in equipment not designed for their use.
- Do not overdischarge.

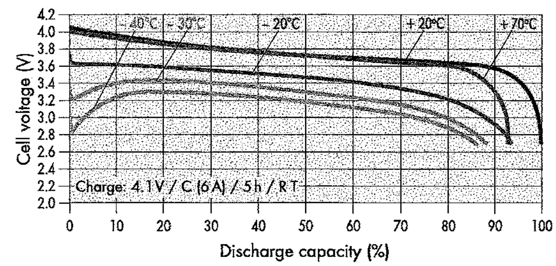
When storing Saft MP batteries:

- Store in a cool, dry place at a temperature below +25°C (+80°F).
- If stored for an extended period of time, charge batteries to between 20 % and 50 % state of charge.

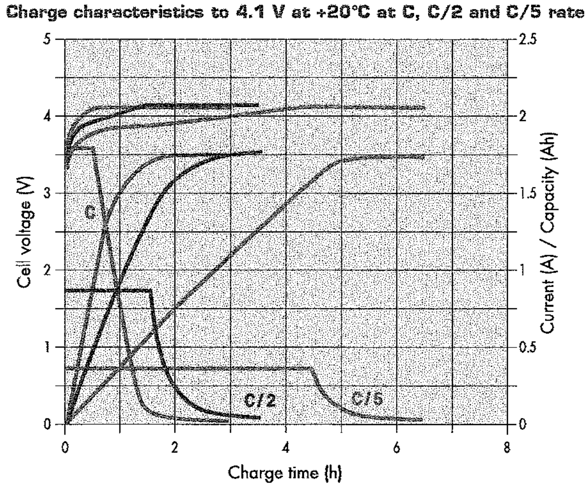
Cycling performances



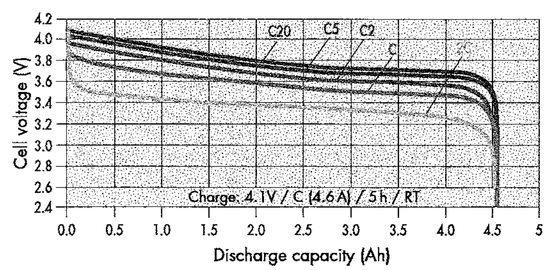
MP 176065 - Discharge characteristics at 1.2 A (C/5)



Charge characteristics to 4.1 V at +20°C at C, C/2 and C/5 rate



MP 174865 - Discharge vs current



Saft
12, rue Sadi Carnot
93170 Bagnolet - France
Tel +33 1 49 93 19 18
Fax +33 1 49 93 19 69

313, Crescent Street
Valdese NC 28690 USA
Tel +1 828 874 41 11
Fax +1 828 879 39 81

www.saftbatteries.com

Doc. N° 54035-2-0604
Published by the Communications Department

Photo credit: Saft
Produced by Arthur Associates

Société anonyme au capital de 31 944 000€
RCS Bobigny B 383 703 873

saft