October 1990

ROBOT BUILDER

The official publication of the Robotics Society of Southern California
P.O. Box 3227, Seal Beach CA 90740, Meetings the 1st Tuesday @ 7:00 PM at MTI College

UPCOMING EVENT CALENDAR:

October 2:	RSSC October Meeting, MTI College: Topic - Forth Language
October 6:	RSSC Robot Project Workshop, ***The Robot Company***
November 6:	RSSC August Meeting, MTI College: Topic - Motors
November 10:	RSSC Robot Project Workshop, ***The Robot Company***
November 25:	Computer Swap Meet at Advanced Computer Products
December 4:	RSSC September Meeting, MTI College: Topic - TBD
December 8:	RSSC Robot Project Workshop, ***The Robot Company***

SEPTEMBER 4th RSSC MEETING

We had another good turnout at our September 4th meeting with over 20 people in attendance.

We discussed several old business issues, such as the Society's logo contest (that never gets anywhere), ideas on how the Society can use the donated multimeter, and the ongoing saga of our organizational by-laws.

New business started with a discussion on the proposed robot contest. Don Golding volunteered to be contest coordinator and help organize the Society's efforts. Jerry Burton briefly reviewed the membership survey results (see article). Joe McCord suggested organizing a Christmas party in December and there are several members that showed similar interest. Joe was going to look into a location and date.

The featured speaker for the evening was Jerry Burton who discussed robot programming. He started with the different ways to program, such as using assembler, a high level language, or custom language and their relative pros and cons. In addition, he evaluated the method used in the SynPet Newton robot knowledge base, which is similar to a C like interpreted file.

SEPTEMBER 8th RSSC ROBOT PROJECT WORKSHOP

On September 8th, the RSSC had another robot project workshop at *The Robot Company* shop in Costa Mesa with over 10 members present. Mark Frank brought the RSSC robot and deserves a lot of thanks for the extra efforts he has contributed to putting the RSSC robot together.

Much of the morning's conversions were on future plans and needs for the robot. The main tasks left to do are the installation of the sonar sensors and the head and motor/encoder assembly.

Carol Petts presented a list of some her thoughts on how the Society should go about organizing and planing robot contest. See her list of suggestions in the article on the Robot Contest.

Our next workshop will be held on October 6th at 10:00 a.m., again at *The Robot Company*, 881 West 18th Street, Costa Mesa.

ROBOT CONTEST IDEAS

(Editor's Note: The following suggestions were prepared by Carol Petts.)

"1st Annual Robot Rendezvous"

First order of Business - We must decide if is for fun or profit!

<u>Events</u> (Consider doing each for both mouse-sized and Newton-class robots)

- Maze (navigation)
- Sprint (speed race)
- Soda Can race (go straight, turn right, stop, pick up can, return, and hand it to owner

- Obstacle course (varied terrain and obstacles)
- Also have above categories for voice commanded robots and telerobots
- A "Free-for-all" or "unlimited" category for people to demonstrate unique robots. Taking turns all day.

Awards

- Have lots of them! Could be money, robot toys/kits, components, books, ribbons, tickets, and etc. Possibly get some industry donations?
- Besides speed winners in each category, have awards such as Most Intelligent, Best Vision, Best Navigation, Best Voice Response, Most Friendly, Best Telerobot, Fastest, Most Accurate, Most Original, Best Looking, and a few reserved for anything special that seems to warrant an award.
- Categorize by age group (e.g., grade school, jr. high, high school, college, adult amateur, and adult professional. We should encourage kids to participate.

Publicity Avenues

IEEE, local papers, local TV/radio, local schools, hobby magazines (e.g., electronics, ham radio). Ideally, send out monthly to catch new people and give all time to prepare.

Location

Fairgrounds, park, parking lot (school or church)

Planning

- Encourage/require advance registration so we know what categories are popular. Discount decreases as contest approaches.
- Make mailing list of participants and use it send reminders and any change of plans.
- Be ready to provide specs (e.g., dimensions for maze and soda can race) well ahead of time to help aid participants in their design.
- Offer help to participants who want it.

Miscellaneous Thoughts

- Sell (cheap) tickets to people who want to watch. This year's observers may be next year's participants
- Maybe wait until day of contest to announce it as "annual"?
- Should RSSC members be allowed to compete in the events they help set up? How do we avoid "unfair advantage" charges?
- Invite guest speakers from industry, sci-fi fields to help draw interest?
- Invite (and charge!) vendors to sell and demonstrate wares
- Do it at end of month to maximize available time for people to see it in the monthly publications
- Do it on a day of robotics historical significance?

Carol Petts

MEMBERSHIP SURVEY RESULTS

(Editor's Note: The following survey results were compiled and summarized by Jerry Burton.)

The following questionnaire was handed out at the August meeting. There were 20 handed out, and 14 responses were turned in. Any of the members who were not in attendance should feel free to make a copy of the questionnaire and give it to me or any other Board member; your input is greatly appreciated. Hopefully, you will let the Board know what direction you want **YOUR** society to take and we will respond accordingly.

Questionnaire

Because this is the end of our first year of operation, the Board thought it would be a good idea to poll the membership concerning the direction the membership would like to take in the future.

In multiple-choice questions, merely circle your desired response.

1- Concerning our 7-9 p.m. monthly meetings, do you think they are:

a) Too long b) Too short
c) Just right

If you answered a) or b), how long should they be ? ___________

2- Concerning the technical content of the meetings, do you think they are:

a) Too technical b) not technical enough c) just right

3- Concerning the format of the meeting. We currently divide the meeting into three sections; General Business, Speaker/Tape Presentation, & Open I/O. Please comment on this format and make any suggestions:

____
____
____
____

4- Concerning the Saturday 10 a.m. Lab meetings. Do you normally attend them ?

Yes _____ No ______

If No, why not?

____

5- Do you find the Lab meetings

a) Very worthwhile b) OK
c) Not worthwhile

If you answered c), what would you suggest to improve them?

6- What do you expect to gain from membership in the Robotics Society?

a) Learn about robots
b) Build a robot
c) Network with others interested in robots.
d) Other, explain;

____
____

7- Are you a member of one of the three working groups (Mechanical, Electrical, System/SW).

Yes: _____ Mechanical
____Electrical _____SW

No: Why not?____________________
____

8- Any comments, cudos, or criticisms you wish to voice, do so here:

____
____

Results

1- Meeting length:
5 too short, 9 just right

Because we are limited to 7-9 p.m. at MTI and the majority thinks 2 hours is just right, we'll stay with MTI and 7-9 p.m. the 1st Tuesday of the month.

2- Technical content:
6 not technical enough, 7 just right, 1 no answer

It seems we could go into just a little more technical detail and meet more of the membership needs.

3- Meeting format:
Most people who responded said they like the format, suggested a little more open I/O in a STRUCTURED FORMAT. Also suggested that the business aspects be moved to the board meeting and only results be presented at the meeting to allow time for technical discussions and I/O.

4- Saturday lab meetings:
Most people that do not attend have other commitments.

5- Worthiness of Lab meetings:

Only six responses were given, and all just responded with OK.

I think we need to improve in this area. Perhaps when our robot is running, the Lab will prove to be of more worth.

6- Expectations:

Almost everyone answered a & c, with eight b responses.

No one provided any other expectations, so these must be the most important.

7- Working group membership:

Those who were not group members indicated they didn't feel they had enough knowledge to contribute to a special interest group.

Don't be intimidated by lack of knowledge; the real technical knowledge many of you lack will be gained by joining a group and asking questions until you are able to contribute.

8- Comments:

A couple of people gave the Board some kind words. The only critical comment was one member being critical of himself for not being more involved.

Conclusion

To summarize, the meetings could be a little more technical with more I/O and less business conducted. The Lab should be made more worthwhile. The primary goals of the Society should be to teach people about robots and provide a basis for networking. Building robots is secondary to these two primary goals.

Jerry Burton

THE ROBOTEER

(Editor's note: Jerry Burton has again contributed material with another fine article...this is becoming a regular column!)

Last month I went into more depth on the overall design, this month I'll focus on the Mapping problem and what must be achieved.

First, however, I want to clarify a point I made last month regarding using the services of other objects. In an object oriented design you have a number of services (functions) available within each object or class. In a language like C you would have library containing all the functions required and you would link the ones needed by each module. In FORTH the functions would be words that can be used throughout the program.

When using a language like C++, there is direct support for this concept. By merely INCLUDING the class definition you automatically get access to all of the services of the referenced class. For example, if the NAVIGATION module needed to check the E-map it would use the function from the MAPPING class to check whether a particular position in the E-map were full or empty, e.g. EMPTY = CHECK_E_MAP(X,Y). If the corresponding E-map tile were indeed empty it would return a value of TRUE.

If any module needed to set a particular tile it would use a function provided by the MAPPING class to set the tile corresponding to a given X,Y position to occupied or full, e.g. SET_E_MAP(X,Y,FULL).

Note that user module does not need to know anything about the structure of the E-map, it is entirely <u>hidden</u> within the implementation of SET_E_MAP. The E-map could be a list, an array of bit-vectors, a B-tree, or ??. The important point is the user (caller) can use the function without having to be concerned with HOW the function provides the requested information, it only cares that the function DOES PROVIDE the information in a specific form.

My work on the MAPPING problem has mostly been concerned with the design questions, I have not implemented any code yet (Other than the HeadScan function, since it was needed in NAVIGATION) but, it seems more logical to present a design in a kind of temporal order, i.e. First you map, then plan, and finally execute the moves to achieve a goal. The actual implementation will be done in parallel, with functions being provided by each object as needed, to get some meaningful results as quickly as possible.

First of all what services must the MAPPING class provide?. We can merely list a few and you should get an idea of what kind of things the MAPPING class must provide. A class must provide a means of creating the objects it will operate on, and provide services to other modules to operate on these objects. In the MAPPING class there are 3 objects, a Range Box, an E-map and a P-map.

1) Functions to create and maintain an E-map.

2) Functions to create and maintain a P-map.

3) Functions to do a sonar scan and maintain a local set of range boxes.

4) Functions to filter and summarize the raw sonar data.

5) Functions to utilize the NAVIGATION module to move the Robot to the next area to be mapped.

In order to think of MAPPING I pretend I am the Robot and envision how I would go about finding out what the environment is like. Remember, the purpose of MAPPING is to find out WHERE THINGS are, NOT WHAT THEY ARE. Therefore, I close my eyes and use my sense of touch to find out where things are (simulating a sonar range detector). I'm a blind person with a cane extending out in front of me and I scan the area in front of me as I move forward. When I encounter something I note its position in my internal map and continue scanning, until I have covered the entire room. Try this experiment for yourself and you will quickly find that you are not as good at remembering where things are as a Robot will be. It also gives you some appreciation of what the Robot has to work with.

The primary means I have to find out where things are is the sonar. It is not very accurate in that it has a beam width of thirty degrees and merely tells me that SOMETHING is at a certain distance WITHIN A SOLID CONE WITH A SPHERICAL BASE.

The MAPPING module will use the sonar to determine the location of obstacles to determine the boundaries of the environment in which the robot is constrained. The approach I intend to use initially is to build an environment map which breaks the environment into an arbitrary number of 'tiles' and marks each tile as either 'empty' or 'occupied'. A convenient size for a tile is a 6 inch square. Once the environment map is built the user should then display the map on a screen and use the ENVIRONMENT_DEFINITION module to assign information to selected subsets of the map for use by the PLANNING module.

The robots environment will use global coordinates where the charging station is considered to be point (0,0). This is the primary reference point for the robot and will form the center of the E-map.

When the robot is initially installed it will not have a valid E-map, so one must be generated. My initial concept of how the E-map is constructed is to use a dynamically allocated array of bit-vectors. The length of the vectors will be changed to cover the X direction, with zero being the mid-bit, and the size of the array N (number of vectors) determines the Y direction, with zero being the N/2 the entry.

There two ways this can be done, Automatic and Manual.

In the Manual mode the user could use a hand-held controller to guide the robot through the free space. In this mode the MAPPING module will make each tile it passes through a 'primary path tile' and update the P-map. It is probably not feasible to have the robot generate the E-map in this mode, as stopping for sonar scans would make the process too slow. If we had a high speed scanning laser range finder on board you could probably do the E-map as well.

However, once the user has guided the robot through the free space desired, the resulting P-map forms a basis for subsequent Automatic mapping.

In the Automatic mode, the robot should initially be in the docking station and the command MAP should be given. The robot will immediately go into Mapping Mode to determine what the environment is like. The MAPPING module should automatically generate guidance commands for input to the NAVIGATION module, based upon what it is 'seeing' with its sonar.

Initially the robot will do a HeadScan while still in its docking station, if there is at least 3 feet clear directly in front of it, it will move forward 3 feet from the docking station and enter the Scan Mode. If it can not leave its docking station due to an obstacle, it will tell the user and abort the Scan Mode.

When in Scan Mode the robot must perform a number of Scan Cycles until it has sufficient information to update the region of the E-map the robot is currently in. Each Scan Cycle consists of the following:

1- The robot will do a 360 scan with its head and update a RangeBox object, which represents a 20x20 square.

2- Turn in the direction of the maximum distance 'seen' and issue three move commands to the NAVIGATE module. Each move will be one-fourth of the maximum distance 'seen' initially.

3- After each sub-move another HeadScan will be performed. At the end of each Scan Cycle another direction will be chosen, such that the robot will go to the furthest location that it has not already been near.

In DumBot the fixed forward sonars and the rotating head sonar are used for obstacle avoidance and mapping. The head sonar is the primary means of generating the E-map and could be augmented by the fixed sonars for short ranges. Each map scan will consist of 24 sonar scans from 0 to 360 degrees in 15 degree increments (I defined a variable HeadIncmnt to control the scan increment to be used, min 2 degrees, and by experimenting with various increments I found 15 degrees to be the best).

A RangeBox object should be dynamically allocated as the robot goes through the Scan Cycle process. Each box represents a 6x6 inch square within the area being scanned. The number of boxes maintained at any one time is a function of what the robot is 'seeing', the number of boxes in both X and Y direction should be changed as required, while performing Scan Cycles.

The distance of a sonar response along a particular angle represents a 'hit' and is scored into the corresponding range box. The sonar has an antenna cone of approximately 30 degrees as shown in figure 1. The initial implementation will merely count the number of hits in the box that would be hit at the distance reported assuming that the object was 'seen' by the center of the cone (i.e. the head angle used to scan). Actually the object seen is somewhere within a 30 degree solid conical section. The larger the distance the larger the potential error.

Experiments will be performed to determine a probability as a function of range so that rather than assigning a weight of '1.0' to every hit, a fractional weight would be added as a function of range. Closer objects get higher weights (since the potential error is smaller) than objects further away.

A paper by Alberto Elfes entitled "Sonar-Based Real-World Mapping and Navigation" from the IEEE Journal of Robotics and Automation, proposes using a probability distribution for both <u>full</u> and <u>empty</u> boxes. The point he makes is that with a single 'look' you not only get a distance to the nearest object, but you also get information on what space is empty.

Figure 1 shows how the probability distributions could be approximated by

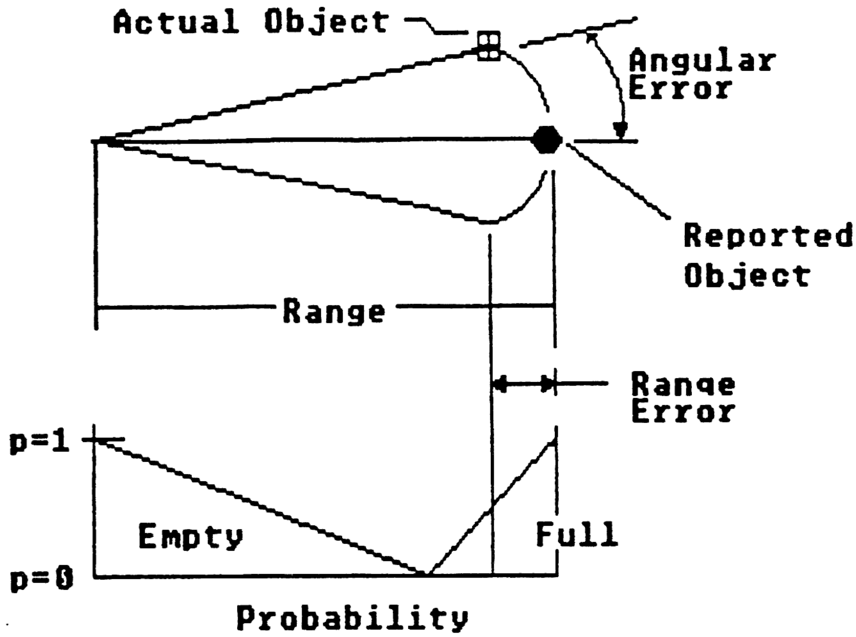


Figure 1 - Sonar Coverage and Probabilities

linear functions of range alone. Depending on the range returned the object can fall within a number of range boxes. If we assume that the object can actually be within any of the boxes that a range of that distance can hit, then we can spread the probability among the boxes. If only one box could be hit then one times the range probability mentioned earlier is added to the box. If the range could hit two boxes then each gets a value of 0.5 times the range probability, etc. He also introduces the concept of using negative weights for intervening boxes along a given angle, thereby using a single scan to indicate not only what tiles are full, but also which ones are empty. In this fashion you end up with range boxes that have the sum of ALL the scans that were taken.

After the initial modules have been done using 1-dimensional range probability distributions these multiple distributions should be investigated.

A range box represents a 6 inch square and accumulates the number of times a sonar range fell within the box, based upon the probability of the range being in a particular box. Each scan provides 24 ranges from the robot's current position. The robot then moves a fixed distance forward and takes another scan, or 24 samples. This process continues until a sufficient number of scans have been made. Experimentation with the robot will determine what number 'sufficient' is.

The resulting mosaic of range boxes contains the number of times the robot 'saw' something in a particular area and the value represents a kind of certainty measure. This number can be treated as a gray-scale code and image processing algorithms can be applied to smooth out the contours. The resulting contours will then be mapped 1-to-1 onto the tiles of the E-map tile matrix. My approach to the E-map is to define it as an array of Bit-vectors that are dynamically changed as the robot moves through the environment.

The resulting E-map only requires 1 bit per tile, therefore a 1024x1024 ft area map will require only 16K bytes, the bit-vector class provided by the Zortech C++ system provides an easily implemented method. An array of 1024 Bit-vectors each 1024 bits long is more than adequate for the average robot environment.

After each range matrix is converted a completion check should be made to determine if the exterior boundary of the E-map forms a closed polygon <u>and</u> all interior tiles have been scanned.

If closure is complete and all interior points have been covered then the robot will be directed to go to the 'home' position and mapping is complete, otherwise the robot will move to the open edge and form a new range matrix and continue the scanning process.

The MAPPING module should provide path commands to the NAVIGATION module when the robot is in mapping mode.

Once the mapping module completes its task it can generate a preliminary path map (P-map) if one doesn't already exist. The P-map is also a Bit-vector area the same size and orientation as the E-map.

There are 2 stages to creating the P-map which defines the primary path the robot can take through the E-map.

The first stage can automatically be done by the MAPPING module, the E-map's empty space will be searched and all empty tiles which are at least two tiles away from a filled tile should be marked in the P-map as a primary path tile provided the distance between the filled tiles exceeds the width of the robot by an allowable amount. Initially, the minimum corridor will be 5 tiles (1 path tile and 4 free tiles 2 on each side of the robot), for a minimum width of 30 inches. The width of the average interior door is 30 inches, so an 18 inch wide robot can fit through with 6 inch clearance on both sides.

The resultant P-map forms the <u>maximum</u> allowable primary path area, the gray area of figure 2 shows such an area, which leaves 12 inches of clearance between the P-map and the E-map when the robot is steered along any segment of path tiles an 18 inch corridor for the robot to move in is always assured.

In the second stage the user may modify the primary path generated, using the ENVIRONMENT_DEFINITION module, by designating specific tiles to be either path, occupied, or free. In this way the user can force the robot to only consider certain designated paths as well as make certain areas off limits.

Some of you may wonder why generate both an E-map and a P-map ? My reasoning is that the PLANNING module only needs to know what tiles it can use to generate guidance commands and really doesn't care about tiles that are occupied. The NAVIGATION module on the other hand doesn't care about the P-map, but may need to update the E-map if it encounters an unforseen obstacle in its path. The P-map is a sub-set of the E-map and provides a quick means of finding obstacle free paths. Another advantage is that the simple algorithm outlined above can be executed to automatically generate the P-map as an initial cut at free space.

Next month I'll cover some the requirements of the ENVIRONMENT_DEFINITION module. Next time you see me give me some feedback on how useful this series is or isn't, and perhaps some other areas of Robotics you would like to have discussed.

P.S. DumBot has a new head and the Laser Scanning project is coming along, I should have a hardware prototype by mid-October. Hopefully, I will have enough time to get into the HPC to add control for a fourth motor, so the head can have 2 degrees of freedom. We still need one of you hardware types to come up with a means of providing the amplitude of the sonar signal so we can do some experiments on how to improve the reliability and accuracy of the sonar.

Jerry Burton

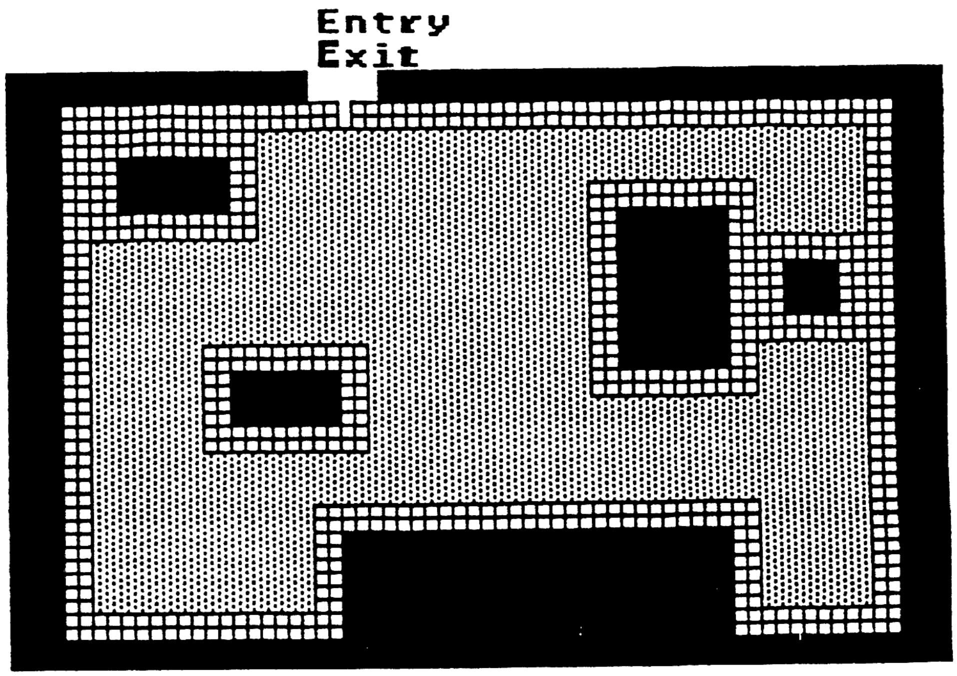


Figure 2 - E-map = Black, P-map = Gray

MOTORS FOR ROBOTS

(Editor's note: Tom Carroll has again contributed material with a general overview article on motors.)

Back in the spring of 1990, I wrote an article entitled "Robots Parts Sources all Around Us." In that article I talked about robot movement and the difficulty most people had with making easily-obtainable motors useful for their robot projects. Needless to say, much of the article concerned locating motors for robot motive ability (wheels), arms, head functions and the like. Many people have since expressed a desire to know how to select just the right motor for a particular robot task or appendage, how to understand the motors specifications (if given in such catalogs such as the one from C&H Sales) and determine if it is suitable for a certain application and how to mechanically and electrically interface these motors to the robot's particular function on one end and the controller system on the other end. I'll leave motor driver and controller circuitry to another article. I would like to continue with the questions; "What can I do with this motor I was given?" or "What does this motor spec sheet mean?"

There are many inexpensive motors that can be purchased for less than a dollar but almost all of these are high-speed ac or dc motors without an attached gear train. Most of these are too high in speed to be of direct use on a robot; they must be slowed down and the torque increased by either a belt, chain, or gear system to be useful. Unless you are a good machinist and can fabricate a gear train or can attach the motor to a leadscrew or gear train that you may already have, I would suggest that you concentrate on using GEARMOTORS (motors with attached gear trains). In addition to the more expensive surplus gearmotors available from C&H Sales and other places, this category also includes drills and portable tools, electric screwdrivers and wrenches, and toys.

Let's look at motor specs and how they relate to robot design. We will concentrate on dc applications. Motor voltage will probably we your #1 concern in the selection process. You will see many voltages listed (from 1.5 to 90 Vdc or more), but I highly suggest you use 6- or 12-volt motors, mainly because that is a typical battery supply voltage. There are a lot of nice 24- and 27-volt gearmotors in catalogs, but unless you have two 12-volt batteries, don't use them.

Another important factor to consider is the speed of the motor, both loaded and unloaded. If you are selecting motors for the wheels, you may want to refer to Table 1, which gives wheel speeds necessary to travel at 4 miles per hour (a nice, fast walking speed) for different wheel diameters. These speeds are calculated as follows: 4 mph = 21,120 ft per hour = 352 ft per minute = 4,224 inches per minute, divided by (pi) x D (wheel diameter in inches) = rpm (4,224/(pi) x D = rpm) or (1344.54 / wheel diameter = rpm). Example: an 8-inch wheel must revolve at 168.1 rpm for the robot to travel at 4 mph. For other diameters or speeds, vary the equation accordingly.

Table 1. **Wheel Speeds for 4-mph Robot**

Diameter (inches)	Speed (rpm)	Diameter (inches)	Speed (rpm)
4	336.1	8	168.1
5	268.9	9	149.4
6	224.1	10	134.5
7	192.1	12	112.0

The above wheel speeds will be the same as the gearmotor speed if you use a direct drive and are convenient as a top speed for a typical home robot, but you must also consider that this is the loaded speed (i.e., when the robot is going uphill, through grass, or thick carpeting).

So far, we have only considered the motive motors (wheels) for a robot. How about an arm motor? How do we select one of those? While not thinking about arm loading right now, and therefore torque, let's consider speed. A good way to determine needed speed is to think of the arm rotation in a full circle. Typically, an arm motion only uses about 90 degrees of rotation (1/4 of a full circle). So, let's consider a direct-drive gearmotor-arm connection for a 5-mph motor. This motor will have a complete rotation of its shaft in 12 seconds, or 1/4 of a rotation in 3 seconds. This means your robot's arm will move from straight down to facing straight out in the time it takes you to say "a thousand and one, a thousand and two, and thousand and three," probably just about right for most robots. "Wow!" you say. "All I have to do is find a 5-rpm gearmotor that runs on 12 Vdc for my robot's arms and a 135-rpm gearmotor for the 10-inch wheels." It would be nice if it was that easy. First, we must look at some other factors such as torque, efficiency, mounting, overall quality of the motor-gear system, and interfacing with the shaft.

When trying to get down to the fine details of motor systems, you need detailed specs from catalogs. Though you may pay more, your very best source for gearmotors is a surplus outfit, such as C&H Sales in Pasadena or H&R Corporation in Philadelphia (see addresses at end of article). The reasoning behind this is the motors usually have been checked out (remember, these are surplus) and the voltages and other specs are given. This is not always known about some surplus motors you may come across with strange military part numbers on the case.

Let's now look at torque, which translates loosely to "how much can my robot lift?" or "how much can it carry?" or "how thick of a carpet can it roll through?" Torque is the rotational force that allows the motor to do work and is measured by setups similar to the "prony brake" from your old physics textbooks. Complex motor-testing systems are not necessary unless you are going into full-scale production. A simple VOM or digital multimeter in series with the motor to be tested and the power supply, and a simple "fish scale" force measurement device connected to the motor through a reel of known radius can accurately measure torque (see Figure 1). Similarly, the same spring scale can measure the force necessary to pull the robot over uneven, uphill, and carpeted surfaces by connecting the scale to the base and pulling it along (see Figure 2). Take this measured force in ounces and multiply by the wheel radius in inches and you have the required torque in inch-ounces.

We've looked at the measurement of gearmotor torque and force needed to pull the robot along. Let's consider the torque required to lift a robot's arm. Generally, the greatest force against the arm is when it is lifting itself or a load that is 90 degrees straight out in front of the robot. Basically, the torque required is the weight of the arm (or arm and load) in ounces measured at the end, multiplied by the radius (arm's length). Thus, a 2-pound arm assembly with a weight measured at the end of 1 pound (16 ounces), plea a 1-pound load at the end with a radius of 15 inches would require a 480-inch-ounce torque motor (16 + 16 (32) x 15 = 480).

"Now that's quite a powerful motor just to lift 1 pound," you say. Well, you're right. That's why most experimental and small robots can only handle less than 1 pound. You can do one of several things. You can gear the motor down or use a 3:1 chain and sprocket arrangement ("3" at the shoulder and "1" on the gearmotor) to run the arm slower and thus gain a mechanical advantage that way. Or, you can shorten the arm's length (radius) or use a mechanism that shortens it when the load is out in front, much the same as a weight lifter does with the barbell close to the plane of his body. Or, you can use an internal spring and level to compensate for at least the arm's weight, or even more if you will always have a payload attached (see Figure 3).

Hopefully, you feel confident enough to start looking through the various catalogs we've talked about. Following this article are examples from the *Fall 1990 C&H Sales Catalog.* Don't be discouraged if, on your first look-through, you do not find a motor with your exact requirements. Tight robot designs may require a higher voltage battery system if you have available, for example, some very nice 24-Vdc motors that meet all your speed and torque requirements, but not voltage. Also, for you members here in southern California, you can visit the C&H Sales store where many unique motors are available that are not listed in the catalog (see attached figure of various motors from C&H sales catalog).

My rule of thumb for looking for motors is: First, look at the speed. If it's for the wheels or arms, I use the previously discussed criteria to see if it will fit my design. Then I look at the voltage: 6- to 12-Vdc are best, as I said earlier, but "great" motors of 24- to 28-Vdc can be used with two 12-volt batteries. You may notice that the catalogs say, "the motor works fine at a lower voltage <u>with reduced speed and torque</u>." That's more than true; most motors are very inefficient at much lower than specified voltages: <u>don't do it</u>. You may also notice motors of 90 Vdc - yuck! You will also notice motors that seem to have the same speed and torque as another but draw 6 to 10 amps. Why use them when a similar 0.2-amp motor that draws 30 to 50 times less current can virtually do the same job? Remember, "amps" is the amount of "food" your robot is eating at any given moment. The more amps the motors draw, the faster it is eating its food supply (battery).

After looking at voltage, look at torque. Remember, noncounterbalanced arms need lots of torque. Big robots that travel uphill or in grass or carpet need lots of torque for the wheels. Now look at basic motor construction. Plastic gears such as the 1.5-volt barbecue spit motor or the dual 6-volt motors that I specified for my "under $50 robot" for the Boy Scout article, won't last long. You get what you pay for. Look at the mounting: a four-hold flange is best. Look at the shaft - is it long enough and large enough in

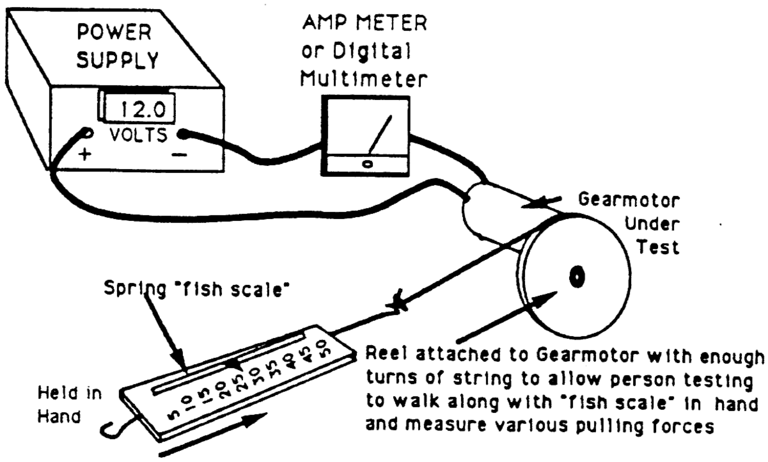


Figure 1.

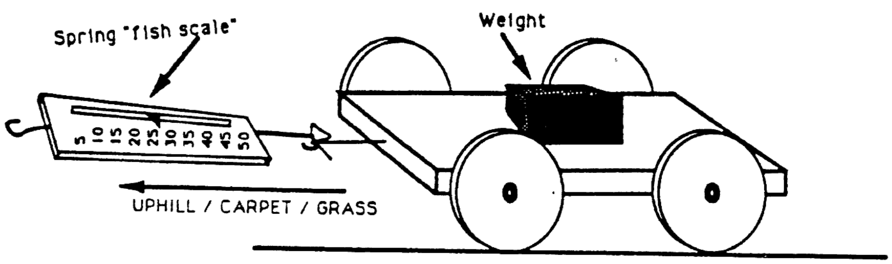


Figure 2.

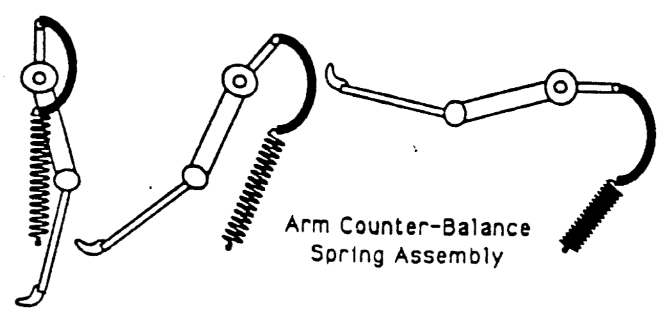


Figure 3.

diameter to mount anything to? Is the shaft connected to a gear that you have to saw off and does it have a flat side to secure a set screw? Is the thing too big to use or have a 90-degree output shaft? You may also want to consider the planetary gear (center shaft "Globe" type) motor as they seem to have better gear trains.

You're now a motor expert and can select motors for any type of robot project. You are also now finding that locating the best motor for the project is not as easy as it sounded 10 minutes ago. Send off for the following catalogs. Have patience. Visit the store in person. Good luck !

Tom Carroll

C&H Sales Company
2176 E. Colorado Blvd.
Pasadena, CA 91117-9988
(213) 681-4925 or (818) 796-4875

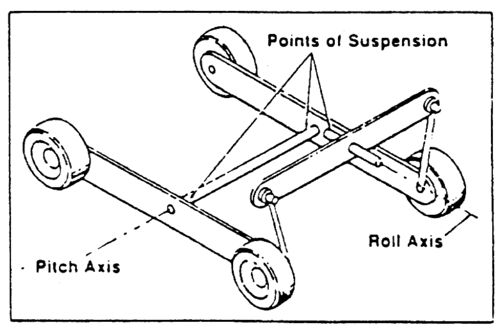
H&R Corporation
401 E. Erie Ave.
Philadelphia, PA 19134-1187
(800) 848-8001 (orders)
(215) 426-1700 (catalog)

FOUR WHEEL VEHICLE SUSPENSION SYSTEM

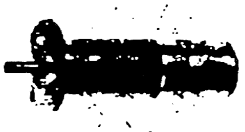
A four wheel suspension system (see figure) uses a simple system of levers with no compliant components to provide three-point contact with uneven terrain. The system provides the stability against tipping of a four-point rectangular base, without the rocking contact to which a rigid four-wheel frame is susceptible.

The elevation of each of the three body points from which the chassis is suspended is determined by the average of the elevations of two of the wheels. Thus, the suspension averages over the roughness of the terrain to some degree, and provides a smoother ride than would be obtained if the body were directly supported on three wheels located at the suspension points.

The four-wheel suspension provides a greater moment arm to resist overturning than does a three-wheel suspension of the same overall length and width. If a wheel sinks into a low spot in the terrain, the corresponding corner of the chassis sinks by a lesser amount. Thus, the tilt angle (and the attendant shift of weight that aggravates the situation) is less than occurs with a rigid four-wheeled frame that rocks at the same low spot.



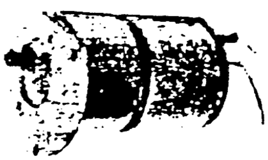
DC GEARHEAD MOTORS



SPECIAL PURCHASE

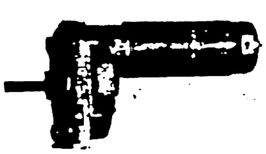
30 RPM GLOBE MOTOR 43A159. 27.5 VDC. Permanent magnet. Reversible. Continuous duty. Ball bearing. 36 RPM @ .119 amp. no load speed. 30 RPM @ 75 oz/in. torque, .228 amp. Dimensions: 7/8" dia. x 2 7/8" long. Shaft: .187" x .475" long. 4-hole flange mount 1 1/8" x 1 3/8".

Stock #DCGM8602 .....*RFE—$24.95



30 RPM BARBER COLMAN, #CYQC 43204-46. 24 volts DC. Permanent magnet. Reversible. Sleeve bearing. 34 RPM no load speed. .070 amp. 30 RPM at 100 oz/in. torque @ .250 amp. Dimensions: 2" dia. x 3 3/8" long. Shaft: 1/8" dia. x 7/16" long. Front face mount. Two tapped holes. Mounting flange is included, but is easily removed.

Stock #DCGM8859 ....$12.50



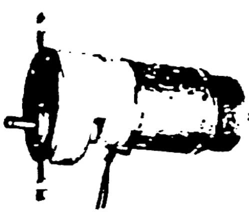
65 RPM BREVEL, #780-953075. 36 VDC. Permanent magnet. Reversible. 95 RPM no load speed. No load current .5 amp DC. 65 RPM @ 12 in/lb. load @ 1.2 amps DC. 12 VDC, 33 RPM no load. Sleeve bearing. Dimensions: (Motor) 1 1/2" dia. x 3 1/2" long. (Gearbox) 3 1/4" sq. x 2" thick. Shaft: 5/16" dia. x 1" long, with double flats. Good!

Stock #DCGM9006 ....$27.50

Good

30 RPM GLOBE MOTOR 407A350. 12 VDC. Permanent magnet, reversible. Continuous duty. Sleeve bearings. 37 RPM @ .144 amp. no load. 30 RPM @ 80 oz/in. torque. .450 amp. Dimensions: 1.5" dia. x 3.650" long. Shaft: Stepped—3/16" dia. x 1/4" long and 5/16" dia. by .250" long. Total overall length of shaft is 1/2". Two mounting ears with mounting holes provided on front of motor. ? torque

Stock #DCGM8851 ....$22.50



30 RPM SWF MOTOR. Model 402.952. 24 volts. Permanent magnet. Reversible. Continuous duty. Sleeve bearing. 35 RPM no load speed. .2 amp no load current 30 RPM @ 80 oz/in. torque, 24 VDC, .3 amp. Dimensions: 3.2" wide across flat mounting tabs x 4.1" long. Output shaft: .236" dia. x .475" long. Rear motor shaft: .191" dia. x .250" long.

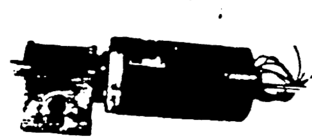
Stock #DCGM8755 ....$9.95

33 RPM PITTMAN, #GM8712-41. 19.1 VDC. Permanent magnet. Reversible. 40 RPM no load speed. No load current .2 amp DC. 33 RPM @ 80 oz/in. .40 amps DC. Sleeve bearing. Dimensions: 1 3/8" dia. max. x 3 1/8" long. Shaft: 3/16" dia. x 5/8" long. Output shaft is offset. Three #4 X 40 tapped holes on front for mounting.

Stock #DCGM9001 ....$17.50

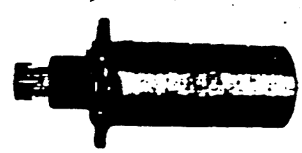
40 RPM PITTMAN. Model GM8312-CG93. 12 volts DC. Permanent magnet. Reversible. Continuous duty. Sleeve bearing. At 12 volts the no load speed is 80 RPM. No load current is .156 amp @ 60 oz/in. torque. 12 volts VDC. 40 RPM, .605 amp. This motor can also be operated at 19.1 volts input. At 19.1 volts the no load speed is 130 RPM. No load current .175 amp @ 80 oz/in. load, 19.1 VDC input the speed is 31 RPM, .770 amp. Dimensions: 1 3/8" dia. x 3 1/8" long. Shaft: .187" dia. x 1.7" long.

Stock #DCGM8752 ....$17.50



48 RPM, 1/4 H.P., 95 VDC GEAR HEAD MOTOR. Compound wound, 95 VDC, 5.1 amp, torque 210 in/lb., temperature rise 55 °C. Intermittent duty. Can be run continuously by reducing voltage on field with slight reduction in power. G.E. Motor, Model 5B-C12RA57A. Gear Model 7GS103JR9VA10, ratio 37, ball bearing. Dimensions: Overall length 13", diameter 4". Right angle gear box. Dimensions: 3" x 4 3/16", shaft 5/8" diameter x 1 7/8" long.

Stock #DCGM7502 ....$60.00



50 RPM GLOBE, #5A2005. 28 VDC. Permanent magnet. 55 RPM no load speed. No load current .180 amp DC. 50 RPM @ 80 oz/in. .35 amps DC. Ball bearing. Dimensions: 1 1/4" dia. max. x 3 3/8" long. Shaft 3/16" dia. x 1/2" long. Four mounting ears on front.

Stock #DCGM9051 ......*RFE-$39.95

MISCELLANEOUS BUSINESS

Roger Alberg has for sale a TI speech synthesis/recognition board he bought from SynPet and finds he does not need it. The board retails for approximately $1000 and we picked them up from SynPet for $400, but Roger will accept any reasonable offer. He can be contacted at (714) 634-1671.

SOCIETY MEMBERSHIP RENEWAL

Yes, it's that time of the year again! Most of the membership joined the RSSC about one year ago, so your annual membership fee has expired. Up to now we have sending out newsletters anyhow, but starting next month, if you are not paid up, you will no longer receive a newsletter. If you have any questions on the status of membership, contact our Society secretary, Jerry Burton at (714) 535-8161.

UPCOMING RSSC EVENTS

For November, Tom Carroll will give a presentation on motors and related topics. Tom will cover a lot of material directly applicable to building robots.

A little further down the line in December, Tom Carroll will provide a presentation on batteries.

NEWSLETTER CHANGES

My apologies go out to everyone for missing the deadline of sending the newsletter out before the beginning of the dated month! I have been very busy as a result of a recent change in my responsibilities at work, and my ongoing battle for making time for my graduate studies.

As a result, I am forced to pass on my assignment as newsletter editor to another member so this month will be my last newsletter. Best of luck to Roland and the future of the Society!!

Scott MacGillivray, Editor