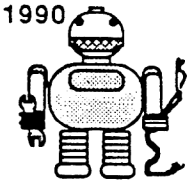
September 1990

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:

September 4:	RSSC September Meeting, MTI College: Topic - Robot Programming
September 8:	RSSC Robot Project Workshop, *The Robot Company*
September 30:	Computer Swap Meet at Advanced Computer Products
October 2:	RSSC October Meeting, MTI College: Topic - Motors
October 6:	RSSC Robot Project Workshop, *The Robot Company*
November 6:	RSSC August Meeting, MTI College: Topic - Forth Language
November 10:	RSSC Robot Project Workshop, *The Robot Company*
November 25:	Computer Swap Meet at Advanced Computer Products

---

AUGUST 7th RSSC MEETING

We had another good turnout at our August 7th meeting with over 20 people in attendance. This month we had no featured speaker for the evening. Instead, the meeting was led by Tom Carroll, and we discussed a variety of old RSSC business activities along with some proposed new activities.

Jerry Burton discussed the RSSC participation at the Orange County Fair and ACP swap meet. In addition, Pete Movesian described some activities that occurred at the OC Fair with his Max robot.

Mark Frank brought the RSSC Robot to the meeting and discussed the current and future work done. Mark has done an absolutely excellent job, mostly by himself, in constructing the robot.

Scott MacGillivray brought up the prospect of organizing a robot contest possibly for next March. The contest would help focus the members' activities on preparing a robot. The contest could be made up of several types of events; for example, a maze running event and a predetermined path to control the robot through by voice in the shortest amount of time. If anyone has suggestions, bring them up at the next meeting. It is hoped that the contest could be held during the ACP swap meet.

Jerry Burton led a discussion on future goals of the RSSC. He distributed a questionnaire soliciting comments on organizational improvements and RSSC activities. We hope to have the results of the questionnaire to discuss at the next meeting, and they will summarized in the next newsletter.

AUGUST 11th RSSC ROBOT PROJECT WORKSHOP

On August 11th, 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. He had added a remote control switch box to drive it around instead of having to carry it. The robot is getting very heavy with the batteries and motors installed.

Most of the discussions at the workshop were on mounting the various components that the RSSC has acquired. Hopefully, we will have photographs of the robot for next month's newsletter. The robot is quickly getting to a stage where we will be able to start creating new sensors and software to experiment with.

Mark Frank deserves a lot of thanks for the extra effort he has contributed to putting the RSSC robot together. I am sure I'm not alone in thanking Mark for the tremendous job he has done!

Our next workshop will be held on September 8th at 10:00 a.m., again at *The Robot Company*, 881 West 18th Street, Costa Mesa.

THE ROBOTEER

(Editor's note: Jerry Burton has again contributed material with another fine article...hopefully this will become a regular column.)

Last month I briefly covered the overall motion control problem, commonly referred to as 'Navigation'. This month I'll go into a little more detailed discussion of what the Navigation modules have to do and how to do it (refer to last months diagram Overall Architecture).

Planning and navigation, at the execution level, can most easily be described as a collection of desired behaviors in carrying out a specific goal. For example, "Don't run into things!", "Go to the end of the hall and turn right!", "Stay to the right of the hallway except when passing!", "Watch for the doorway just ahead to your right and enter that room!", "Follow that person!", "Go into mapping mode and build an updated environment map of this room", "Get me a Beer", etc.

Control structures that rely entirely on a previously generated environment map (E-map) do not provide the essential adaptability necessary for coping with unexpected events, e.g. someone or something 'appearing' in the robot's path when according to the previously generated map the path should have been clear. Although the E-map is used as the primary means to determine feasible paths through the environment it should be augmented by a NAVIGATION module that can adapt to unforseen events.

These events might include unanticipated obstacles (a chair has been moved), moving objects (people or pets moving through the robot's path), or the recognition of a landmark in a seemingly inappropriate location (a beacon or landmark is not where the robot thinks it should be). These unexpected events should influence, in an appropriate manner, the course that the robot takes to achieve the desired goal. The NAVIGATION module could use a localized avoidance strategy to modify the commands from the PLANNING module to move around or avoid obstacles, while still striving to achieve the ultimate end point.

The PLANNING module needs some idea of the structure of the static environment and where there is "free" space through which it can travel. A MAPPING module for generating and updating a path map (P-map) should be executed prior to executing the PLANNING module. If the map already exists it is used as an input to the PLANNING module. If the map does not exist services of the MAPPING module must be run to generate it.

There has been a lot of work done on modeling of the environment as a collection of 3-dimensional geometric objects. Many of them so complex that it takes a dedicated VAX computer just to do the calculations. Since, we must rely on a 286 running at 12 mhz, we must keep the process simple. The easiest mapping method I have found so far is to merely project everything in the environment onto a 2-dimension grid, like tiles on the floor. If each tile in the internal model represents a 6x6 inch tile in the outside world, and we merely use 1 bit to represent whether the corresponding tile is full (1) or empty (0), then we have a very compact way of storing information about the external environment in which the robot is to operate.

The MAPPING module should provide a local wander mode to generate 2 maps. The E-map defines the constraints on the robot, i.e. where it can and can't go. The E-map forms an enclosed space. Free space is contained within the boundaries of the E-map. A specific sub-set of the 'free' space is a P-map, containing all the feasible paths. Once these maps are generated they can be used by the PLANNING & NAVIGATION modules. Both the E-map & P-map should be able to be modified by the

user, using the ENVIRONMENT_DEFINITION module.

Planning is the task of determining a given path from the robot's current position 'A' to some arbitrary final destination point 'B'. The path so chosen should provide a set of subgoals, such that the robot can move from its initial position to the desired end position, using nothing more than the path map (P-map) provided by the MAPPING module. The PLANNING module provides the NAVIGATION module with a series of X,Y positions, representing a movement of the robot.

Navigation, on the other hand, is the task of generating a sequence of motor control commands that carry out the desired path, while avoiding ALL obstacles in the desired path, both known (via the E-map) and unanticipated.

When an unanticipated event occurs the NAVIGATION module must immediately stop the robot and attempt to determine whether the event is static or dynamic. A dynamic event means the obstacle is moving with respect to the robot and the conflict may resolve itself if the robot merely stalls temporarily. If the object is static then the NAVIGATION module must determine what local moves are necessary to get 'around' the obstacle so that it can resume the course set by the PLANNING module.

The traditional way of looking at an architecture diagram is to view it as a hierarchy, with 'Planning' at a higher level than 'Navigation', implying that there are decisions and control structures in Planning that affect the way Navigation is to respond. However, in this design, it is possible that when Navigation needs to do local obstacle avoidance that it would CALL upon the Planning module to determine an appropriate path and re-execute Navigation to carry out this sub-goal. This implies each object must be re-enterant.

A better way to 'view' the diagram is to imagine you were looking down on the diagram from above, and that all modules were at the SAME level, i.e. no one object (module) has any higher control than any other. You could think of a hierarchy of modules depending upon the level of CALL (the number of times the module has been called).

One of the major investigations and trade-offs that needs to be examined is whether NAVIGATION should use a self-contained sub-path planning strategy versus using the more global PLANNING modules strategy.

If all movement is planned by the PLANNING module then when an obstacle is encountered the first thing that has to be done is to use the services of the MAPPING module to incorporate this 'new' information into the E/P maps and then use the services of the PLANNING module to find a path around the obstacle. The advantage of this approach is that the E/P maps are always current and you wouldn't have to go into a global re-mapping cycle as often. The disadvantage is that it requires more time than a local avoidance strategy would.

If however, the NAVIGATION module has a simplified method of getting around the obstacle the other two modules would not be called. The advantage is that the robot would seem more goal oriented and react faster and the local avoidance strategy would be totally independent of the global strategy. The disadvantage is that since 'new' obstacles are not put into the E/P maps when they are encountered, the next time the robot has to traverse the same area, it will encounter the same 'new' obstacle, again and again, until the MAPPING module is executed in global mode to incorporate the 'new' obstacle into the maps.

As I mentioned last month the design being presented here is object oriented and event driven. By Object oriented, I mean there are self contained modules (objects) that provide specific services for other objects. By event driven, I mean it is impossible to determine what order the objects will be called or even how many times, the control sequences are entirely determined by what events the robot encounters while trying to achieve an overall goal.

Next month I'll go into more detail on the methods of building E/P maps as a precursor to the Planning problem.

Jerry Burton

MISCELLANEOUS BUSINESS

Steve Hodges, president of SynPet, still has six or seven complete Newton robots for sale. He can be contacted at *PETLAND* pet store, (208) 375-0400.

Jerry Burton has for sale an HPC board from the Newton robot for $200 and motor controller board for $100. In addition, he has acquired an extra *HEROID* robot in mint condition, which is available for the best cash offer.

Tom Carroll has a complete robot that he built several years ago that he is now interested in selling. The robot has a complete drive motor and wheel set and is housed in a body. It is on display at *The Robot Company*, the asking price is $200.

UPCOMING RSSC EVENTS

For our September 4th meeting, Jerry Burton will present information on robot programming and building a knowledge base for intelligent robotic platforms. The presentation will be directly applicable to the RSSC robot and to the members that are building Newton-class machines.

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

In November, Don Golding will lead a presentation on the Forth Computer Language. This language has many features that lend itself to use in robots, and part of the RSSC robot programming will utilize Forth.

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

September 4th MEETING AGENDA

1) Business Agenda
   - Robot Contest Ideas
   - Questionnaire Results Summary
2) Presentation of RSSC robot design and construction status
3) Presentation - Jerry Burton will provide a presentation on programming robots
4) RAM (Random Access Meeting) - bring something of interest to share with the membership!

I hope to see the entire membership there, along with any and all interested individuals!

Scott MacGillivray, Editor