UNDERGRADUATE REPORT

GPS Based Location Determination: GPS Feedback Control of Robots

by Adrian Cottin
Advisor: P.S. Krishnaprasad

UG 2003-1

Technical Research Report

<u>GPS Based Location Determination:</u>
GPS Feedback Control of Robots

By Adrian Cottin
Fall Semester 2002

Advisors: P.S. Krishnaprasad, S. Andersson, and F. Zhang

Intelligent Servosystems Laboratory

The Institute for Systems Research

OVERVIEW

The purpose of my research was to integrate the Global Positioning System into motion control of nonholonomic robots using the Motion Description Language (MDLe) developed in the Intelligent Servosystems Laboratory. The robot used in my research is an all terrain robot. An instruction plan was used to move the robot to a goal position using GPS coordinates in feedback control, while at the same time avoiding obstacles on its path. Use of GPS for feedback control is an excellent method for outdoor navigation since it allows the robot to correct its course taking into consideration irregularities in the terrain.

GLOBAL POSITIONING SYSTEM (GPS)

GPS is an all-weather, worldwide, continuous coverage, satellite based radio navigation system. It is composed of 24 satellites placed in six, nearly circular, 12-hour orbits (see figure 1) [1]. GPS was developed by the Department of Defense in the early 1970s. The mission of this system is to allow military ships, aircrafts and ground vehicles to determine their location anywhere in the world. It was originally meant for classified operations, but civilians were granted use of the system. However, the accuracy for civilians was reduced [2].

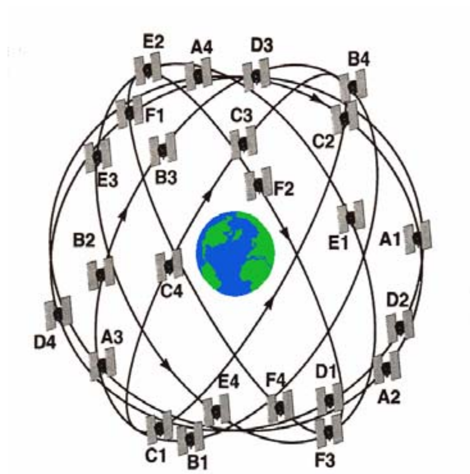


Figure 1: GPS Satellites Orbits

The system is based on time of flight ranging. The GPS receiver contains an internal clock that is used to note the time of arrival of the satellite ranging code. From this information, the GPS receiver determines the time required for the signal to propagate from the satellite to the receiver. Since the signal travels at the speed of light, we can determine the distance by multiplying the time by the speed of light. This distance is called pseudorange because it is biased by the receiver and the satellite clock errors [1]. At any given time, the GPS receivers need to have at least 4 satellites in range in order to calculate an accurate position. This is because the system has to calculate 4 different unknowns: latitude, longitude, altitude and the clock error. The clock error has to be calculated in order to obtain the actual range. In order to obtain a more accurate position (close to 2cm), we used Differential GPS (DGPS). DGPS involves two receivers. One is the base and the second one, in this case the robot, is the rover or mobile station. The base will estimate the error and then broadcast correction data to the rover station allowing it to correct its position (see Figure 2).

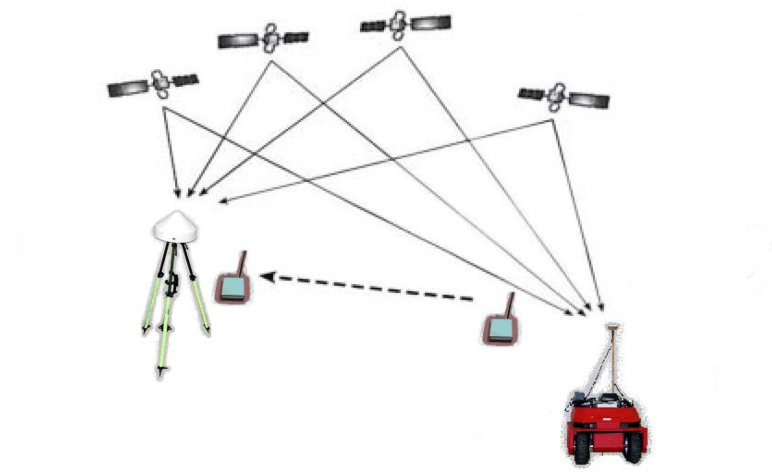


Figure 2: Differential GPS Setup

EXTENDED MOTION DESCRIPTION LANGUAGE (MDLe)

MDLe is a language for hybrid motion control which allows one to compose complex, interrupt driven control laws from a set of simple rules [3]. MDLe is used to control kinetic state machines (e.g. robots). MDLe is used by implementing a *plan*. A *plan* is composed of a string of *atoms* together with an interrupt. Each *atom* is a combination of an *interrupt quark* and an *action quark.* In our research, MDLe was used to control the robot, and several new *quarks* were introduced into MDLe to meet our objectives.

ROBOT CONTROL SYSTEM

The robot lives in what is called the Special Euclidean Group SE(2), i.e., the robot's position is a point (x, y, Θ); where x and y are its position in the plane and Θ is its heading angle. In our research we used an iRobot Corporation ATRVMini (see figure 3). The equations of motion that describe the motion of the robot are:

$$\dot{x} = U_f \cos(\Theta)$$
$$\dot{y} = U_f \sin(\Theta)$$
$$\dot{\Theta} = U_\Theta$$

where, U_f is the forward translational velocity control variable and U_Θ is the rotational velocity control variable of the robot. An action quark is either an open loop or feedback control law. In our project we used feedback or closed loop controls. As shown in figure 3, the robot uses the data from the GPS receiver and compass to constantly update its position and correct itself in case it gets off course. The robot uses feedback from the sonars to detect objects on its path and eventually avoid them. After it has cleared the object, it will go back to the GPS and compass data to calculate the best path to the goal position.

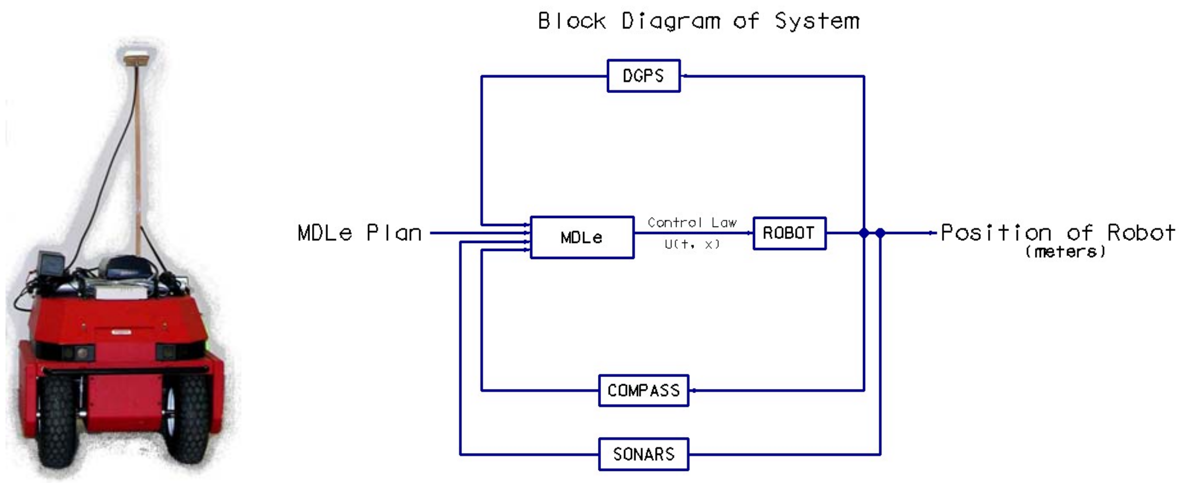


Figure 3: ATRVMini and System Block Diagram

One of the advantages that GPS has over odometry is that the accuracy on the position will not decay over time. The accuracy of odometry decays because the tires slide and slip. This will make the use of odometry on grass or other types of terrain unreliable. However, GPS cannot be used indoors. Carrier-sensed DGPS will give an accurate position within 2 cm as long as it keeps the differential fix. In the event it loses the differential fix, it will correct its position automatically when the differential fix is obtained again unlike the odometry that will have to be reset after the robot has been moving for some time.

LANDMARK BASED NAVIGATION

The idea of robot motion control using landmarks has been used extensively for localization on a global map [4], [5]. These approaches typically require that all navigable areas be accurately mapped on a "global" coordinate system so that a robot can get from one landmark to another. For this project, I used the motion description language MDLe to control the robot to move between landmarks. The landmarks used are the buildings and other important spots of the University of Maryland Mall. The use of GPS will allow the creation of a global map for robot navigation.

OBJECTIVES

These were the objectives of the project for the Fall Semester 2002:

1. Debug the MDLe Quark CirclePointgps() which controls the robot to navigate a circle around a desired point with a desired radius while avoiding obstacles along its path using GPS data and the compass angle.
2. Divide the AvoidGo() Quark into two quarks: One avoidance quark and a Go quark that will control the robot to a goal position.
3. Create a GPS coordinate map of the University of Maryland Mall area that includes all buildings and navigation paths to be used for landscape based navigation.
4. Create several MDLe plans that will cover the different areas of the mall

EXPERIMENTAL PROCEDURES

Over the summer, Timothy Merkin and I incorporated the GPS module into the CirclePoint() quark and created CirclePointgps() quark. This quark did not work properly and we were unable to fix it due to time constraints. The problem with this quark is that the robot tends to spin on the same point very fast instead of circling the desired point.

I tried different approaches in order to solve the problem. First, I set boundaries for the maximum angle the robot could turn. This approach did not work because the robot turned at variable angles, but it did not change after it reached the boundary.

During the test performed, I found that inside the lab while using the compass module, there were inaccuracies in the angle reading. There are magnetic fields inside the lab that produce a heading error higher than 50%. Then, I tried to reduce or almost eliminate the use of the compass on the robot. With the help of S. Andersson and P. Sodre, I wrote a function to approximate the current heading angle of the robot given the initial heading angle by the compass. After several tests, I concluded this method did not solve the problem because the margin of error between the actual and approximated heading angle increased very fast. I plan to continue with new trials when the weather permits outside tests.

The next objective was to divide the AvoidGo() quark into two different quarks. The first one was an action quark named Avoid. This quark will control the robot to avoid obstacles on its path. It will function until the path is completely clear. To work along with this quark, I wrote two interrupt quarks: ToAvoid() and ExitAvoid(). The underlying algorithm for these quarks is based on the avoidance algorithm developed by F. Zhang from the Intelligent Servosystems Lab. These interrupt quarks are used in MDLe plans to switch back and forth between the desired action such as Go(), MoveToPoint(), CirclePoint(), etc, and the Avoid() quark. ToAvoid() will exit the running action quark if the sonars perceive an object on the way. Then the Avoid() quark will control the robot until the path is clear. Then the ExitAvoid() quark will exit the Avoid() quark, allowing the robot to return to the desired action.

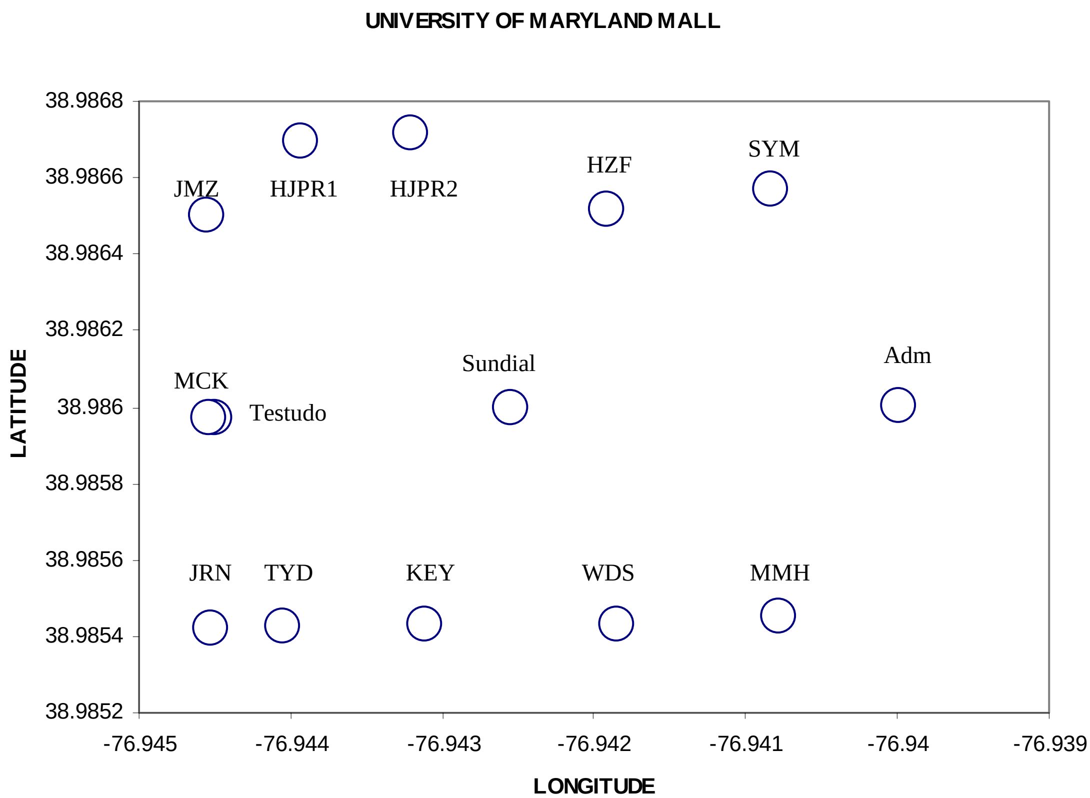
The next step was to create a GPS coordinates map of the University of Maryland Mall area that includes all buildings and navigation paths to be used for landmark based navigation. This procedure took several weeks and was delayed several times due to the weather conditions. Nevertheless, I was able to obtain an accurate GPS coordinate position for all the buildings in the mall area as well as the sundial, Testudo and the fountain. Also, most the entrances to the center area paths of the mall area are mapped which will be used in the MDLe plans to create a demo project of the area.

For this part of the project, I had a problem with the coverage range of the radio modems used to communicate between the base and mobile stations. In order to accomplish all measurements I had to try different positioning of the base station in order to assure a link between both stations to cover the entire mall area. The best location was the area at the sundial which provided a big enough radius to transmit the signal between the base station and the mobile station on the ATRVmini without losing the differential fix on the DGPS systems.

Finally, I am currently working on creating plans to run a demo on the mall to control the robot around all the buildings. This will be tested during the Spring Semester 2003. Please refer to the experimental results for a detailed map of the area, and a list of the different Atoms for possible use on a demo plan.

EXPERIMENTAL RESULTS

This is the map of the entire mall area along with the GPS coordinates obtained:



Building	**Abbreviation**	Latitude	**Longitude**
Administration	Adm	38.98600427	-76.93999465
Marie Mount Hall	MMH	38.9854522	-76.9407811
Woods Hall	WDS	38.98542914	-76.94184801
Francis Scott Key Hall	KEY	38.98543036	-76.94311733
Tydings Hall	TYD	38.98542571	-76.94404439
Journalism	JRN	38.98542173	-76.94453022
McKeldin Library	MCK	38.98597213	-76.94450441
Jimenez Building	JMZ	38.98649885	-76.94455577
H.J. Patterson Bldg Entrance 1	HJPR1	38.98669529	-76.94392781
H.J. Patterson Bldg Entrance 2	HJPR2	38.98671368	-76.94320836
Holzapfel	HZF	38.98651514	-76.94191272
Symons Hall	SYM	38.98656978	-76.9408288
Sundial		38.98597057	-76.94453436
Testudo		38.98599765	-76.94254119

Finally, this plan will control the robot to cover the entire mall area starting at the administration building and will end at Symons Hall.

Plan = Start at the Administration building

TO MMH
(Atom (gpsStop -76.9407811 38.9854522)(gogps -76.9407811 38.9854522))

FROM MMH TO WDS
(Atom (gpsStop -76.94184801 38.98542914)(gogps -76.94184801 38.98542914))

FROM WDS TO KEY
(Atom (gpsStop -76.94311733 38.98543036)(gogps -76.94311733 38.98543036))

FROM KEY TO JRN
(Atom (gpsStop -76.94453022 38.98542173)(gogps -76.94453022 38.98542173))

FROM JRN TO MCK
(Atom (gpsStop -76.9444985 38.9854607)(gogps -76.9444985 38.9854607)))
(Atom (gpsStop -76.94449171 38.98559919)(gogps -76.94449171 38.98559919))
(Atom (gpsStop -76.94450441 38.98597213)(gogps -76.94450441 38.98597213))

FROM MCK TO JMZ
(Atom (gpsStop -76.94449771 38.98626108)(gogps -76.94449771 38.98626108))
(Atom (gpsStop -76.94451509 38.98641062)(gogps -76.94451509 38.98641062))
(Atom (gpsStop -76.94455577 38.98649885)(gogps -76.94455577 38.98649885))

FRM JMZ TO PATTERSON 1
(Atom (gpsStop -76.94369154 38.98650848)(gogps -76.94369154 38.98650848))
(Atom (gpsStop -76.94390705 38.98664112)(gogps -76.94390705 38.98664112))
(Atom (gpsStop -76.94392781 38.98669529)(gogps -76.94392781 38.98669529))

FROM PATTERSON 1 TO PATTERSON 2
(Atom (gpsStop -76.94354778 38.98669230)(gogps -76.94354778 38.98669230))
(Atom (gpsStop -76.94320836 38.98671368)(gogps -76.94320836 38.98671368))

PATTERSON 2 TO HZF
(Atom (gpsStop -76.94323933 38.98666492)(gogps -76.94323933 38.98666492))
(Atom (gpsStop -76.94338690 38.98659436)(gogps -76.94338690 38.98659436))
(Atom (gpsStop -76.94354267 38.98653065)(gogps -76.94354267 38.98653065))
(Atom (gpsStop -76.94291135 38.98653341)(gogps -76.94291135 38.98653341))
(Atom (gpsStop -76.94297009 38.9864767)(gogps -76.94297009 38.9864767))
(Atom (gpsStop -76.94284123 38.98667869)(gogps -76.94284123 38.98667869))
(Atom (gpsStop -76.9427302 38.98687838)(gogps -76.9427302 38.98687838))
(Atom (gpsStop -76.94254945 38.98707966)(gogps -76.94254945 38.98707966))

(Atom (gpsStop -76.94229125 38.98681518)(gogps -76.94229125 38.98681518))
(Atom (gpsStop -76.94230093 38.98678202)(gogps -76.94230093 38.98678202))
(Atom (gpsStop -76.94228778 38.98668384)(gogps -76.94228778 38.98668384))
(Atom (gpsStop -76.94216628 38.98660764)(gogps -76.94216628 38.98660764))
(Atom (gpsStop -76.94220215 38.98650661)(gogps -76.94220215 38.98650661))
(Atom (gpsStop -76.94191272 38.98651514)(gogps -76.94191272 38.98651514))

HZF TO SYM
(Atom (gpsStop -76.9408288 38.98656978)(gogps -76.9408288 38.98656978))

CONCLUSION

Even though the CirclePointgps() is not working properly, I was able to separate the avoidance algorithm from the AvoidGo() quark to be used for obstacle avoidance in other plans besides waypoint navigation. I was successful in generating a GPS coordinate map of the mall area which includes all buildings and some important landmarks. Also, I was successful in collecting enough points to create plans for landscape navigation around the entire mall area. Nevertheless, the accuracy of these points will have to be tested in the future.

ACKNOKLEDGMENTS

I would like to give a special thanks to Michael Goldgeier, Patrick Sodre and Ankur Mahajan for their help to my project.

WORKS CITED

[1] Jay A. Farrel and Matthew Barth. *The Global Positioning System & Inertial Navigation*, pp 142-146. McGraw-Hill, 1998.

[2] Thomas A. Herring. The Global Positioning System. *Scientific American*, pp 44-49. Feb. 1996.

[3] V. Manikonda, P. S. Krishnaprasad, and J. Hendler. Languages, behaviors, hybrid architectures and motion control. In J. Baillieul and J.C.Willems, editors, *Mathematical Control Theory*, pp. 199–226. Springer, 1999.

[4] R. Sim and G. Dudek. Mobile robot localization from learned landmarks. In *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pp. 1060–1065, 1998.

[5] A. Bandera, C. Urdiales, and F. Sandoval. Autonomous global localization using Markov chains and optimized sonar landmarks. In *Proc. IEEE/RSJ Int. Conf. on IntelligentRobots and Systems*, pp. 288–293, 2000.