Formations of Autonomous Vehicles Using Global Positioning Systems (GPS)

Kourosh Rahnamai; Kevin Gorman; Andrew Gray; Payman Arabshahi
Jet Propulsion Laboratory
California Institute of Technology
Pasadena, CA 91109
{rahnamai,payman,gray}@jpl.nasa.gov
{krahnama@wnec.edu}kgorman@wnec.edu

Abstract—In this work we present the design, development, and testing of a hardware testbed for formation movement, using a fleet of autonomous R/C (Remote Control) vehicles equipped with Global Positioning Systems (GPS). The work was formulated and executed as a student project with the intent of being both a vehicle for education and investigation into low-cost formation control. The low cost testbed is to be used for validation of communication protocols and control algorithms that with further development may find application in distributed sensing applications such as formation flying helicopters, blimps, or other aerobots. Critical to many applications of such distributed sensing platforms is navigation and collaborative navigation. This effort is part of many ongoing research efforts at NASA's Jet Propulsion Laboratory targeting low-cost distributed science gathering capability on earth and in space.

TABLE OF CONTENTS

1. INTRODUCTION

At the heart of the testbed is an autonomous system that facilitates communication between a Personal Computer (PC), a Remote Control, and a Remote-Controlled Vehicle as shown in Figure 1. The system contains a continuous control loop between the R/C vehicle, a centralized computer, and the remote controller. The feedback control system will include only GPS receivers, which will attempt to accurately guide the R/C vehicles. The GPS receivers used will allow for an accuracy range of 3-5 meters. Using Differential GPS (DGPS) will in fact allow for an accuracy of fraction of a meter.

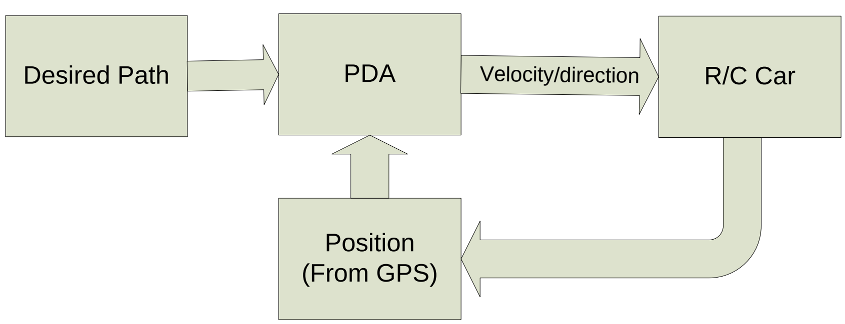

Figure 1. Basic block diagram of system

[1] 0-7803-8870-4/05/$20.00© 2005 IEEE
[2] IEEEAC paper #1411, Version 3, Updated November 22, 2004

The GPS used limits the control loop to update rate of one sample per second. GPS data is obtained from the GPS receiver through the serial port of a Personal Digital Assistant (Sharp Zaurus) which is then sent to the PC via a wireless network card. This data is then interpreted and used by the PC to send a signal to the Remote Control via a Digital to Analog Converter (DAC) in order to navigate the R/C vehicle from a certain initial point to a destination point, executing a specified formation. This study creates a platform for controlling many agents in collaborative manner.

2. PROBLEM STATEMENT

To improve reliability and science return value, and to reduce cost, many ground or atmospheric experiment will deploy two or more cooperative agents (vehicles, helicopters, blimps, or other flying devices) to execute maneuver while maintaining a tight formation. As an initial experiment to design such a testbed and investigate the required hardware and software, formation movement of RC vehicles were selected for this project. This selection reduces the scope of the project such that it is manageable as a two semester senior project performed by a team of three students and also reduces the problem from a 6-dimensional (for flying devices) to a 3-dimensional control system.

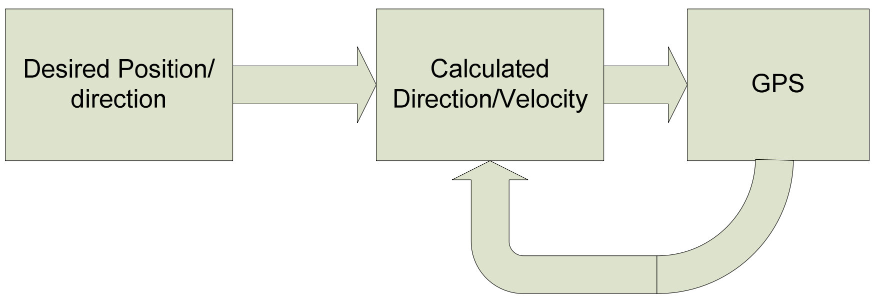

Figure 2. Software block diagram for software system

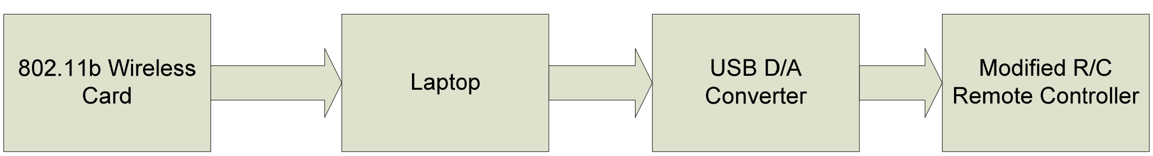

Figure 3. Block diagram for controlling the R/C Car

This project required implementation of a wireless network that would transfer GPS data from the PDA to the PC. This data was then used to send command signal through the remote control to the R/C car to create formations of one or more R/C cars. These cars would travel in their respective formations based upon the GPS data the PC had received. The idea for this project was a joint decision between Western New England College and the Jet Propulsion Laboratory (JPL). A team of three students, Brain Hall, Shawn Welch, and Andrew Willouer implemented this design as their senior project.

In order for this system to be autonomous, it must sustain a continuous control loop between the R/C vehicle, a centralized computer, and the remote controller. The control loop uses only GPS receivers which accurately guides the RC vehicles. The configuration allows for an accuracy range of 3-5 meters.

3. OBJECTIVES

This project had five objectives to accomplish.

1) Interface the remote of the R/C car to the centralized computer through the use of a DAC.

2) Use a PDA (Sharp Zaurus SL-5600) to act as the control unit of the R/C vehicles, and to gather the raw GPS data from the GPS receivers.

3) Augment a GPS receivers to each PDA controlled R/C vehicle. This was done using the I/O port and a Serial I/O cable on the PDA.

4) Develop a wireless network communication between RC vehicles and the PC. One PC is used as the central command and control unit for the system.

5) Demonstrate and test run of the autonomous vehicles completing a formation movement anywhere from a simple circular path to a figure 8.

4. SYSTEM DESIGN

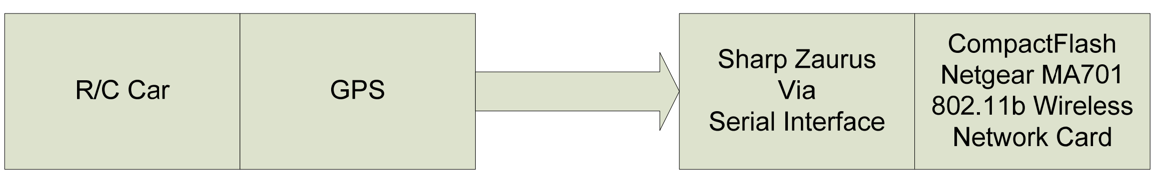

Figure 4. Block diagram for the Remote Control Car

In this design, the PC will control the path of the remote control car using the GPS system. The path of car is predetermined. The basic block diagram for the testbed is as show in Figure1.

5. DETAILED SYSTEM DESIGN

Our system design contained three major pieces: the software, the hardware, and the integration of the software and the hardware into a single system. In the next sections, we will discuss the details of each one of these blocks.

6. DETAILED SYSTEM DESIGN (SOFTWARE)

The software design of our testbed utilizes UDP (User Datagram Protocol). This is a connection-less unreliable protocol. This protocol was selected for these reasons. If data is lost during transmission, the data needs to be interpolated. If this isn't done, we may be receiving previous points that are no longer valid. The software was designed to allow for multiple vehicles to be controlled at the same time.

When the GPS sends the new NMEA point to the Zaurus via the serial port, the Zaurus will convert this data into latitude and longitude. This data is then sent back to the laptop via a data packet. In this design, each remote vehicle will have an identifier associated with it such that the laptop, "Core Server", can control multiple vehicles collectively.

The data from the GPS is sent to the Zaurus once a second. This will limit the performance of our system. The speed of the update also affected the velocity of the vehicle. If the velocity for the vehicle was too fast, data points from the GPS wouldn't be gather quick enough and we will have poor performance. Unfortunately, this parameter of our system can't be adjusted since this is a standard GPS.

The actual processing of the data is performed by the laptop. The program on the laptop decoded the NMEA data from the GPS and calculated the direction and velocity. The direction and velocity information was then sent to the remote control. A simplified block diagram is shown in Figure 2.

7. DETAILED SYSTEM DESIGN (HARDWARE)

There are five major hardware blocks in our testbed: the sharp Zaurus PDA, the PC, the GPS unit, the R/C Car and the USB data acquisition that is integrated into the R/C controller unit.

PDA—The sharp Zaurus SL-5600 has a powerful 400 MHz StrongARM processor. The SL-5600 also has expansion slots and 64 MB of internal memory. The PDA has a CompactFlash slot, which can be used for memory expansion, GPS, or network interface adapter. There is also a SansDisk slot for memory expansion. Lastly, there is a serial connection slot. The Zaurus runs under an embedded version of Linux. This allows for cross-platform development as well as a stable operating system with multi-tasking capability. Most of the development for the Zaurus was completed on a desktop, then cross-compiled and installed on the Zaurus.

The PC—The PC served two purposes in our testbed – providing the predetermined path and the calculation for the feedback controller. The laptop used is a Quantex 600 MHz Pentium III with 256 MB of RAM running Windows XP.

GPS unit—The Deluo serial GPS was chosen for its compatibility with the Sharp Zaurus (requires no drivers to be installed), being lightweight, and easy of mounting on our chosen R/C vehicle. In order to connect this GPS receiver to the PDA, we chose a Serial I/O cable, which is specifically designed for the Zaurus. This allows us to use the I/O port on the bottom of the Zaurus, and also have the serial GPS plugged into this port through the Serial I/O cable. This cable is also made so that when plugged into the PDA, it still allows the use of the keypad on the Zaurus. The GPS data is used to closed the feedback loop in our system. If no GPS signal is found, the system will continue to sit in an idle state until the signal is found. The data is gathered though the GPS is in Raw NMEA format (Figure 5), which is the National Marine Electronics Association, which is an industry standard. There are only certain parts of this data that are of importance to us. The first important data resided in the '$GPGSA,A,3,13' data segment. The 'A,3,13' part of the data indicates if the GPS receiver is gathering data or not by reading what the second character is. If it is a '1', then there is no relevant GPS data. If it is a '2' or '3', then there is 2- or 3-dimensional GPS data, respectively. The other lines of this raw data are not very important to us as they are only satellite ID lines, except for the '$GPRMC' line of data. This line contains three important pieces of data that are pertinent to our system. The number '4206.8767' represents the current latitude, and the number '07231.3201' represents the current longitude, and the number after the longitude '000.0; represents the speed in knots of the receiver. Once we were able to get this raw data from the GPS receiver, we had to transfer this data into the PDA. The GPS only updates the received data from the satellites once per second.

```
$GPGGA,202323.005,4206.8767,N,07231.3201,W,1,07,01.5,00068.7,M,-33.6,M,,*51
$GPGSA,A,3,13,30,04,,05,,24,,10,17,,,02.6,01.5,02.1*00
$GPGSV,2,1,08,13,28,062,39,30,24,295,38,04,44,072,39,07,13,145,*7A
$GPGSV,2,2,08,05,28,249,38,24,72,001,38,10,55,219,35,17,48,288,43*71
$GPRMC,202323.005,A,4206.8767,N,07231.3201,W,000.0,027.7,250204,014.8,W*68
```

Figure 5. Raw GPS data

After extracting required data from the GPS raw information this data is then transferred into the PDA via the serial port. The GPS runs on 5 volts. It was decided to power the GPS via a 9 volt battery connected to a 5 volt regulator (LM7805).

R/C vehicle—We chose an R/C vehicle manufactured by Team Associated. This particular R/C vehicle is the model RC10T3. This vehicle comes from the factory RTR (ready to run) and is basically ready to go out of the box, with included remote controller, receiver, and speed controller, all which are needed in order to operate the R/C vehicle. This R/C requires four AA batteries for the remote, and a 7.2V rechargeable Nickel-Metal Hydride (Ni-MH) battery to power the car. We chose this particular car because it has plenty of room on the chassis to accommodate several hardware items required for this project. There is plenty of extra room in order to place sensors, GPS receivers, PDA's, etc. These types of cars are also easier to gear down to have a lower top speed in order not to create excessive delay in feedback loop which would lead to instability. Also this car has open gearbox which would be ideal for mounting true car speed sensors. The front suspension on the car also does not suffer from a condition called bump steer. This is when the front wheels of the car turn in when the suspension is compressed. This is good due to the fact that if angel sensors are placed upon the front end of the car, the angle will remain constant if the suspension is under compression. These are the easiest cars to work with and modify to meet the needs of this project. Also the gearbox was changed to slow down the car and allow proper course correction using one sample per second data update rate from GPS unit.

DAC unit—A DAC was need to interface the remote controller of the R/C car to the PC. PMD-1208LS Data Acquisition Board manufacture by Personal Measurement Devices was chosen for this purpose. This board is a USB based board, with direct plug and play capability to the PC. This board has two analog outputs, which are connected to the R/C remote controller unit for sending the steering and the throttle commands. The ease of use, and the convenient USB interface with the PC, and functionality of this unit were the major factors in selection of this board.

8. Control Systems Block

The input to our system (desired path) came from a Laptop PC. As shown in the figure 3, the position of the vehicle is received via a wireless 802.11b network card. This information is converted from NMEA format into latitude and longitude by the program running on the PC and is converted into the desired direction and velocity for the remote control vehicle. This information is sent to a USB D/A converter that is attached to a modified remote controller. The steering for the vehicle is attached to one of the analog outputs of the D/A converter. The second output is connected to the speed controller of the remote control.

The required processor speeds for both the laptop and the PDA were unknown at the time the project was started. It was determined that the processors on both the laptop and the PDA are more than adequate for this project.

9. Feedback Block

The position of the car is determined by a GPS receiver which is physically connected to the remote control car. The GPS is connected to a Zaurus PDA via the serial port. The Zaurus processes the NMEA information from the GPS. This information is then sent to the PC via the wireless network card (Netgear MA701) in the NMEA format (directly sent from GPS to PC) as show in Figure 4.

10. System Integration and Algorithms-

Both the hardware and software blocks from above were integrated into a single testbed system. The block diagram for the combined system is shown in Figure 6.

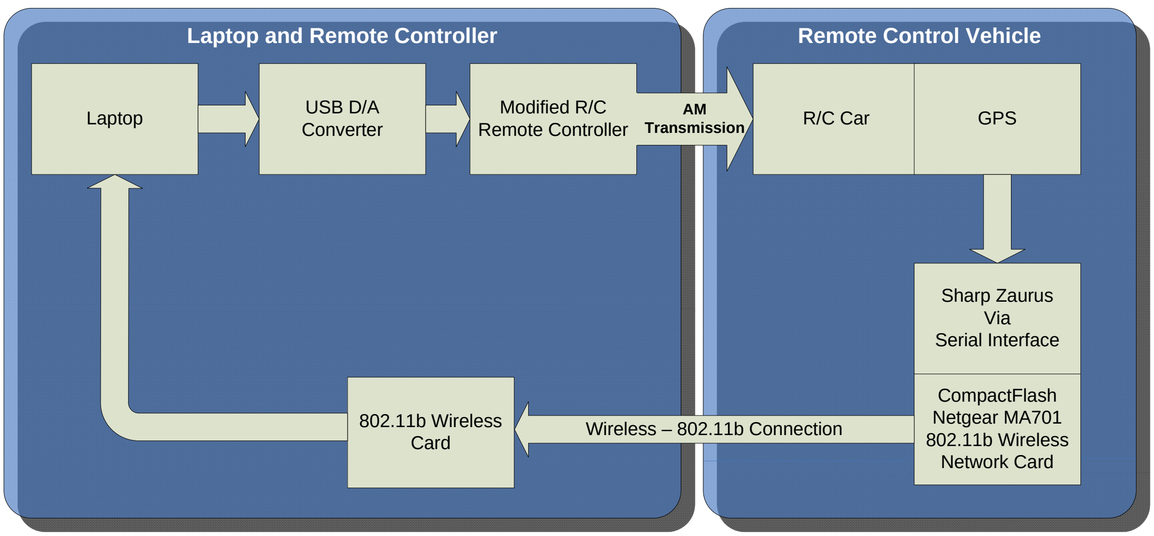

Figure 6. System Integration

11. Conclusions

Overall this project was very successful. A low budget education and research testbed has been developed that allows implementation of experiments in many different aspect and technologies that are utilized in formation and cooperative movement of many agents in 3 or dimensions. At its current state the testbed includes:

- R/C vehicles that are properly interfaced with a centralized computer through a DAC.
- PDA units that can be used as sensors hardware interface, pass through gate for collected information and sensor data or as computational resource for each agent.
- PDA units are networked with each other and the central computer using the developed networking code.

- GPS units that are interfaced with the PDAs
- Only open source software is used and all the Algorithm developments are in C using only ANSI instructions

One very important aspect of this platform is the flexibility of the hardware setup. During early stages of any algorithm development and testing, the central computer can be used as the main software development and test center and PDAs can only be used as pass through gates that collect sensor data and transmit the information to the central computer for processing and command and control decisions. This platform also allows very easily transfer of the command, control and advance algorithms that are developed on the central computer to each PDA, which can create partial or complete autonomy for each agent. In this configuration the central computer is used as a pass through gate for transmitting the final command and control decisions to each agent or can be removed completely from the feedback control loop.

We have several simple modifications to this platform which will improve the performance and flexibility. The AM remote controls for R/C cars will be changed to a FM transmitter to improve fidelity and SNR of the transmitted signals. More shielding of the DAC unit will also reduce the interferences. Finally the use of differential GPS units will significantly improve the navigational accuracy. The motor control units of the R/C vehicles will also be changed to current controlled drivers which allow linear response to variant terrain contour. After these simple modifications this flexible platform can be used for many more research and senior projects in the future

REFERENCES

[1] Kourosh Rahnamai, Payman Arabshahi, and Andrew Gray, "Fuzzy Supervised Optimal Regulator for Spacecraft Formation Flying", Proceedings of the NAFIPS (the North American Fuzzy Information Processing Society), Chicago, IL, July 24-26, 2003.

[2] H. Schaub, S.R. Vadali, J.L. Junkins, and K.T. Alfried, “Spacecraft Formation Flying Control Using Mean Orbit elements”, Proc. 1999 Astrodynamics Conference, AAS 99-310.

[3] Y. Ulybyshev, “Long-Term Formation Keeping of Satellite Constellation Using Linear-Quadratic Controller”, J. Guid., Contr. & Dyn., Vol. 21, No. 1, pp. 109-115, 1998.

[4] C. Sabol, R. Burns, and C.A. McLaughlin, “Satellite Formation Flying Design and Evolution”, Proc. AAS/AIAA space Flight Mechanics Conference, Feb 1999.

[5] S.R. Starin, R.K. Yedavalli, and A.G. Sparks, “Design of a LQR controller of reduced inputs for multiple spacecraft formation flying”, Proc. American Control Conference, Arlington, VA June 25-27, 2001.

[6] Data Acquisition from Measurement Computing. 2004. Measurement Computing. 05 May 2004. http://www.measurementcomputing.com.

[7] Deluo GPS Receivers, Navigation Systems and Electronics. 2004. Deluo Electronics. 05 May 2004.

[8] DePriest, Dale. NMEA Data. 05 May 2004 http://www.gpsinformation.org/dale/nmea.htm.

[9] Making the Ad Hoc Wireless Connection. 08 April 2002. Microsoft. 05 May 2004.

[10] QNX Developer Support. 2004. QNX Software Systems. 05 May 2004.

[11] QT Reference Documentation. 2001. Trolltech. 05 May 2004. http://doc.trolltech.com/2.3/index.html.

[12] Red Hat – Linux, Embedded Linux, and Open Source Solutions. 2004. Red Hat. 05 May 2004. http://www.redhat.com/.

[13] Red Hat Linux 9. 2004. Red Hat. 05 May 2004. http://www.redhat.com/docs/manuals/linux/RHL-9-Manual/install-guide/.

[14] ROM Update. 2004. Sharp Electronics Corp. 05 May 2004. http://www.myzaurus.com/romupdate6.asp.

[15] The National Marine Electronics Association. 2003. National Marine Electronics Association. 05 May 2004. http://www.nmea.org/.

[16] Zaurus Software Index. Killefiz. 05 May 2004. http://www.killefiz.de/zaurus/showapps.php?cat=18.

BIOGRAPHY

Kourosh Rahnamai *has more than seventeen years of academic and industrial experience. His research interests are in the areas of Communications, controls, fuzzy and neural-based systems, navigation, and optical communications, spacecraft control, systems Engineering, Estimation theory, Linear and Nonlinear Kalman Filters. He has developed Laboratories for low-cost rapid prototyping and hardware in the loop test and validation of complex algorithms and has developed and analyzed mathematical models for complex multi-disciplinary systems.*