System Overview

Distributed processing using 4 BASIC Stamp microcontrollers

Communications Board
- Receives GPS position data from navigation controller
- Receives temperature data from temperature sensor
- Transmits telemetry to ground station
- Stop command from ground station sets pin 7 high

Navigation Board
- Receives position data from GPS receiver
- Monitors output from rangefinder controller
- Operates vehicle remote control (steering, forward/backward)

Rangefinder Board
- Receives distance data from rangefinder
- If obstacle is detected at certain range, sends stop command to navigation controller

Ground Station Board
- Receives telemetry from Comm. Board
- Allows human operator to send stop command to comm.
Control Board
EEPROM Memory map

Comm.

Usage  | Program   | Address
---    |-----------|--------
90%    | Remote CODE | 0x07FF

Control

Usage  | Program                | Address
---    |------------------------|--------
16%    | Centering program      | 0x0    
59%    | Main loop              | 2 KB   
8%     | Obstacle avoidance     |        
2%     | Halt program           |        

Rangefinder

Usage  | Program               | Address
---    |-----------------------|--------
16%    | SRF04range1           | 0x07FF 

Ground Station

Usage  | Program   | Address
---    |-----------|--------
65%    | GS CODE   | 0x07FF
Ground Station relays command packets including 1-byte command
Remote/Car relays telemetry packet upon receipt of command
  • Includes temperature and x and y estimator positions of car
Command Packets: (6 Bytes Total)

Sample: 253007002001000254

- Beginning of Packet
- Can number
- Ground Station Number
- Command Size (#of Bytes)
- Command:
  - 0=Transmit Telemetry
  - 1=Stop Car
  - 2=Restart Car if Stopped
- End of Packet

Command Packet:

- No input means car telemeters data and continues normal operation
- User input of 1 issues stop car command and continues telemetering
- User input of 2 issues restart car command and continues telemetering

06-12-03
Remote Packets:

- Issued upon receipt of ground station packets.
- Includes 5 Bytes:
  - Car Temp,
  - Low/High Byte of x-postion
  - Low/High Byte of y-postion
Comm. System Overview

- Remote (car) board
  - Control system estimator (x and y-postion) values relayed via serial connection between control and remote comm. Board.
  - Flow control for accurate serial communications
  - Thermistor circuit for measuring temperature at car.
  - Radio Shack Radio for RxTx

- Ground Station Board
  - Midland Radio for RxTx
  - Serial cable connection to basic stamp program on laptop
  - User can input stop command by pressing 1 key repeatedly
  - User can issue restart command by pressing 2 key repeatedly
Comm. Results

- Car guided along L-shaped path
- Relay of x & y positions over comm. Link
- After run, data dump of control board EEPROM (including x & y Estimator positions) compared to Telemetered values
EEProm vs. Telemetered Data

- 8/18 x-y positions successfully received
- “L-shaped” path of car well characterized by telemetered data
The vehicle carries all of these components with it. There is no human operator and no way to command the vehicle except through a “stop” command from the ground station.
Using GPS, the vehicle measures the angular distance between its velocity vector and the vector to the target. The control system attempts to drive this angle to zero.
Main:
pause(LOOP_DELAY)
GOSUB GetGPSEstimate
GOSUB SaveTelemetry
GOSUB SendTelemetry
TURN = 0
IF (abs(phi-127) <= PHI_MIN) THEN TurnEquals0
TURN = 1
TurnEquals0:
GOSUB ServicePosition
GOSUB ServiceStopCommand
IF (TURN = 0) THEN NO_SERVICE_ANGLE
GOSUB ServiceAngle
NO_SERVICE_ANGLE:
GOTO Main

EVENTS AND SERVICES FRAMEWORK FOR NAVIGATION PROGRAM

Gets the data from the GPS and scales and translates it to coordinates within the playing field.

The vehicle only turns if it is significantly off-course.

Sends the vehicle’s coordinates to the comm. Controller.

Calculates the turn time, sets the turn timer, and sends the turn command to the remote control.
Event: Vehicle is off-course  
Service: Turn Wheels, Set Turn Timer

Event: Turn Timer Expires  
Service: Straighten Wheels

Event: Obstacle is detected  
Service: Reverse, turn wheels

Event: Obstacle gone  
Service: Drive Forward

START

Driving Forward, Transmitting Telemetry

Turning

Event: Vehicle Arrives at Target  
Service: Stop

END

Backing up and turning
Results

→ Target area so small that vehicle could cross it in less than one program cycle

→ Vehicle eventually stopped in target area after several minutes of searching

→ Vehicle collided several times with obstacles, because the program cycle was too long to permit quick response
Estimation tasks and interface

- Estimate initial position (Centering)
- Estimate position
- Estimate velocity
- Calculate angle to target

Estimator
States:
Pos, vel, angle

GPS
TURN?
FORWARD?

Position

Angle to target
GPS Data

- Data resolution is 3 meters
- Bad measurements occur (± 3 meters typical)
- Binary resolution x10 (0.3 meters) in SW
Centering

- Take mean of 10 GPS measurements
- Define as center of grid:
  \[ x = 32767, \]
  \[ y = 32767 \]
- All GPS measurements are shifted relative to center

Y longitude

65535 (16 bit)

32767

Initial position

X latitude

0 32767 65535 (16 bit)
Estimation

If NOT FORWARD no operation
else
If No turn:

**Propagator:**
\[ \text{pos}_{k-} = \text{pos}_{k-1+} + \text{vel}_{k-1+} \]
\[ \text{vel}_{k-} = \text{vel}_{k-1+} \]

**Corrector:**
\[ \text{pos}_{k+} = \text{pos}_{k-} + (\text{GPS}_k - \text{pos}_{k-}) / \text{GAIN} \]
\[ \text{vel}_{k+} = \text{pos}_{k+} - \text{pos}_{k-1+} \]

If Turn:

**Corrector:**
\[ \text{vel}_k = \text{SPEED} \times (\text{pos} - \text{target}) / ||\text{pos} - \text{target}|| \]
(Assume on target, and normalize to speed)

Indices:
- \( k \): time
- \( k- \): estimate given \( k-1 \) measurements
- \( k+ \): estimate given \( k \) measurements

Time delay

pos, vel

GPS

pos, vel
Estimation

- Calculate phi
- Return phi and pos

\[ \begin{align*}
\text{X latitude} & \leq 65535 \\
\text{Y longitude} & \end{align*} \]
Hardware Restrictions

- No continuous control in the RC-car’s turning rate. Modes available are:
  - Left turn
  - Straight
  - Right turn
- No continuous speed control. Modes:
  - Forwards
  - Backwards
  - Stop
• So as to provide continuous control, we tried to pulse the wheels and drive at higher frequencies (10 – 20Hz.)

• This technique proved non applicable, because
  – Delay time from board-command to actual turning execution is approx. 120 ms
  – Wheels do not instantly turn back to inertial position after turn execution

• Accordingly, pulsing leads to unpredictable and chaotic behavior of the RC-car
As pulsing the drive proved impossible, we decided to
  – Operate the car at an invariable speed
  – Operate the car at an invariable turning radius
  – Obtain arbitrary paths by piecing together straight trajectory segments and bent segments respectively
Turning Angles

- In a series of experiments, we determined the ratio of pulse duration vs. turning angle.
- Taking into account the limited computational ability of a Basic Stamp board, a linear approximation relation is used to implement the turning rate.
Experiment Data

Pulse Duration for execution of left turn

Pulse Duration [ms] vs. Turn Magnitude [degrees]
Pulse Duration for execution of left turn

\[ y = 11.131x - 124.03 \]

\[ R^2 = 0.9825 \]
Ultrasonic Range Finder Overview

- Devantech SRF04 ultrasonic range finder used to measure distance from obstacle to car.
- SRF04 works by transmitting an ultrasonic pulse and measuring the time it takes to receive the pulse echo.
- Output from the SRF04 is in the form of a variable-width pulse that corresponds to the distance to the target.
- Car would avoid obstacles based on range finder data.
Ultrasonic Range Finder Features

• Voltage - 5v
• Current - 30mA Typ. 50mA Max.
• Frequency - 40KHz
• Max Range - 3 m
• Min Range - 3 cm
• Sensitivity - Detect 3cm diameter broom handle at > 2m
• Input Trigger - 10uS Min. TTL level pulse
• Echo Pulse - Positive TTL level signal, width proportional to range.
• Small Size – (1.7 in x .8 in x .7 in height) 43mm x 20mm x 17mm height
Ultrasonic Range Finder Setup

- Echo pulse output connected to pin 7 on Rangefinder board
- Trigger pulse input connected to pin 6 on Rangefinder board
Performance of Sensors

• Temperature sensor
  - Temperature sensor worked well during testing but only transmitted data intermittently during the test
  - Probable cause due to a loose connection to a pin on the Comm. Board

• Range finder
  - Range finder performed as expected and was able to detect obstacles within a two meter distance
  - Data was processed on the Rangefinder Board and a signal was sent to the obstacle avoidance program on the control board upon detection of an obstacle
Conclusions

• Things Learned:
  – Basic Electronics
  – Serial Communication
  – Hardware/Software Integration
  – Attitude Estimation and Control of an R/C Car
  – Basic GPS Navigation