JUGGERNAUT

Final Report
IMDL Spring 1998
Edmund Loftus
Scott Shamblin

ABSTRACT

This report documents the design and implementation of a prototype autonomous agent. The purpose of the agent is to assist in the rescue of combat pilots forced down in enemy territory. This requires that the agent have rough-terrain capability and inverted running capability. Accordingly, the prototype was built as battery-powered treaded vehicle with a mid-mounted body allowing ground clearance while righted or inverted. The sensors on the prototype which would allow it to fulfill this role are compass, infrared sensors and emitters, bump sensor, traction monitor, and inversion sensor. The infrared sensors are used for obstacle avoidance and edge detection behaviors. The traction monitor detects immobility, and the inversion sensor detects the attitude of the agent. The compass and bump sensor were not implemented in this project. The robot used a Talrik II processor and sensor board with a Motorola 68HC11 processor. The program was written in IC. Indoors, the robot displayed satisfactory obstacle avoidance, edge detection, traction monitoring, rough-terrain capability, and inverted running. Performance outdoors was unsatisfactory due to tread design.

EXECUTIVE SUMMARY

This prototype autonomous agent, called Juggernaut, was constructed to test the feasibility of using a robot to assist in the rescue of military aviators shot down while in enemy-held territory. This would be accomplished by using the robot to send coded radio messages containing the whereabouts of the pilot to rescue forces while simultaneously driving away from the pilot. In this way, the pilot could contact the rescue forces without revealing his position to enemy forces. In order to navigate outdoors, the agent needs to be robust and be able to negotiate rough terrain. Furthermore, the robot must be able to avoid permanent immobility due to unfortunate interaction with obstacles. To satisfy these requirements the robot was constructed of an aluminum box to house the motors, processor, and sensors. The aluminum construction offers superior resistance to environmental contamination and can withstand the rigors of outdoor operation. To facilitate navigation of rough terrain, Juggernaut rides on twin, steel-belted, polyurethane treads which mesh with toothed wheels adapted from an automobile application. Locomotion is provided by twin heavy-duty, six-volt motors turning the treads through a custom-built, two-stage, worm-to-spur gear reduction. To help insure that the robot can navigate through obstacles unrestrained, sensor features are included which alert the robot to impending obstacle collision, loss of mobility, presence of edges, and inversion due to rolling. Obstacle avoidance and edge detection are accomplished using specially-mounted infrared detectors and emitters. There is a set to monitor the left and right sides of the robot near the top, and another set near the bottom. The top set is used for obstacle avoidance and the bottom set is used for edge detection. They are arranged symmetrically about the robot's center axis so that they switch functions when the robot is inverted. Located in the front of the robot, these sensors are angled upward so that small, surmountable objects are undetected and can subsequently be driven over, but large obstacles are avoided. Traction monitoring is accomplished using an encoder wheel towed behind the robot. Each revolution of the wheel sends a signal to the computer indicating that it is moving. In the event that the robot becomes immobile and the encoder stops turning, the robot will exhibit an escape behavior consisting of reversing and turning. Inversion is monitored by a sensor consisting of a steel ball confined to a plastic housing. When the robot is righted, no contact is made. When the robot becomes inverted, a connection is made which the processor recognizes as the inversion signal. The robot then performs all functions in the inverted mode. The robot performed well in the indoor tests, with all functions operating as intended. The Juggernaut exhibits good all-terrain capability, as it negotiates many obstacles with ease. Those obstacles which prove too difficult simply activate the traction monitor, causing Juggernaut to retreat and find a new path. Outdoor testing uncovered a serious flaw in the tread design. The open tread design is prone to becoming clogged with organic material, causing the treads to tighten and the chassis to deform. This problem resulted in chronic tread derailment due to wheel misalignment.

Introduction

During a conflict, the loss of aircraft to enemy fire is a real possibility. Pilots of these aircraft are then left stranded behind enemy lines. The first and foremost concern of the downed pilot is to secure his position and evade capture. Pilots may be unable to contact rescue forces since any radio transmission could reveal their position to the enemy. The objective of this project is to construct a robot that could allow a downed pilot to communicate more effectively with rescue forces without the risk of disclosing his location. The prototype robot, called Juggernaut, demonstrates some of the concepts needed to accomplish the objective stated above. The Juggernaut has all-terrain capability, it can operate inverted, and it can detect if it is immobile. These functions augment the standard IR obstacle avoidance sensors.

The robot consists of a treaded, aluminum platform which is protected from the environment. The design of the platform is such that the robot can operate effectively while overturned. Each tread is driven by a dedicated motor and gear assembly. Two 7.2V rechargeable Makita batteries power the motors. The robot's behavior is controlled by a Talrik II board receiving information from the following sensors:

1. IR
2. Traction monitor
3. Inversion detector

Some of these sensors serve dual purposes. For example, the IR devices perform not only the obstacle avoidance function, but also edge detection. The IR devices are also situated such that they can determine if an obstacle is too high to drive over. The traction monitor detects if the robot is immobile, thus precluding the need for a bump sensor. The inversion detector is a ball-bearing which makes contact with a switch when inverted.

This report covers the development and results of the Juggernaut project conducted during the Spring 1998 semester. The report is divided into seven sections:

1. Integrated System- Description of operating system.
2. Mobile Platform- Development of platform for stated objective.
3. Actuation- Explanation of drivetrain operation and control.
4. Sensors- Scope, theory and objectives of sensors.
5. Behaviors- Development of behaviors, coding intricacies.
6. Experimental Results- Juggernaut operation.
7. Conclusions- Summary of work accomplished.

Integrated System

The body of the robot consists of an aluminum box with removable top, bottom, front and rear access panels. Mounted on each side are two cogged wheels which interface with the drive treads. The motor batteries are also mounted on the sides of the robot. The front of

the robot houses the infrared devices and inversion sensor, the middle section is devoted to the microprocessor and batteries, and the rear houses the drivetrain and motor controls. The traction monitor is mounted externally on the rear access panel.

The robot operates with a hierarchical control algorithm, meaning that different situations have different levels of response priority based on their relative effect on the robot's operation. The following list shows the hierarchy of control actions for the robot:

1. Inversion
2. Edge detection
3. Traction
4. Obstacle avoidance

The arbitrator receives instructions from the above sensor subroutines. The control actions above are listed in order of descending priority. For example, the robot must know if it is inverted before it can know to perform any other function properly. If the robot did not know that it was inverted, the motors would run backward and the robot would operate erroneously. Once inversion is established, the next most important function is edge detection, because driving over a ledge of sufficient height could cause the robot permanent damage. The third item in the hierarchy is traction monitoring because the robot is ineffective if it is immobile and could become trapped in an obstacle avoidance maneuver, much like the early TJ robots. The final level of priority is obstacle avoidance. This is the fundamental function of any autonomous agent, because it allows the robot to navigate around objects in the environment.

This structure meets the objective of providing a prototype capable of demonstrating some of the features needed to assist pilots in the field. The robot has a rugged platform which allows for outdoor operation, sensors which help insure the robot will continue to operate despite hindrances imposed by the environment, and a control algorithm which prioritizes the information received from the sensors, thus insuring that more serious threats to the robot's operation are handled first.

Mobile Platform/Actuation

The aluminum box which comprises the body of the robot is a heavily reinforced structure designed to withstand tension from the treads. An anti-torque arm in the front and stout two-point axle support in the rear help prevent the chassis from deflecting excessively due to the belt tension. All rolling surfaces are supported by ball-bearings. These help to insure smooth operation of the drivetrain. The drive consists of a two-stage worm-to-spur reduction for each wheel. The reduction is 145:1, yet fits in a compact area in the rear of the robot. The motors and worm gear reduction are an integral unit which can be moved forward and aft, allowing for the possibility of changing the final spur gear drive ratio at a later time. The worms and worm-gears were donated by Berg. The motors are six-volt units probably designed for use in small electric vehicles for children. The wheels are cam drives from automobiles. They were selected because they were inexpensive and readily

available. The treads are of steel-belted polyurethane construction and were custom built and donated by Mektrol. The drive batteries are mounted on the sides of the robot, between the treads. This is a convenient location for accessibility and does not use up valuable interior space.

Sensors

Three types of sensors were used to allow the robot to negotiate rough terrain. An inversion sensor detects if the robot is inverted while a traction sensor detects movement with respect to the ground. Infrared sensors and emitters are used to detect both high edges and insurmountable objects .

The inversion sensor consists of a metal ball bearing contained in a plastic chamber with a pair of contacts at one end. The chamber is situated such that the ball touches the contacts when the robot is inverted but not when it is righted. The contacts are connected between the +5 volt pin and the signal pin of analog port 7. Thus, when the program reads a value of 255 from port 7 (indicating inversion), it reassigns which analog port is associated with which variable (left_eye, right_edge, etc.). It also renames motor0 and motor1 and reverses their directions so that turns will be executed in the proper direction. One problem encountered with the sensor configuration is that when inverted, vibrations induced by the robots motion caused the ball to make intermittent contact. The result was that the robot repeatedly reversed direction and generally became ineffectual. This was rectified by enabling a counter variable which recorded the number of times the port gave the same reading. The program can only change the sensor and motor assignments if the counter is high enough, indicating a series of accurate readings.

The infrared sensors are Sharp model GP1U58. Each sensor and emitter pair is mounted on a common bracket so that the sensors and emitter point in the same direction. All four sensors are mounted in the front behind the polycarbonate panel. Two of the sensor-emitter pairs are mounted towards the top pointing up while the other two are mounted towards the bottom, pointing down. In this orientation, the upper sensors can be used for selective obstacle avoidance while the lower sensors are used for edge detection, as shown in Figure 1. When the robot is inverted, the upper and lower sensor pairs reverse roles, allowing the edge detection and obstacle avoidance functions to operate normally. The sensors are connected to analog ports 3 through 6. As stated in the previous paragraph, the ports to be read for obstacle avoidance(left_eye and right_eye) and those read for edge detection(left_edge and right_edge) are determined by the inversion detection subroutine.

Traction monitoring is accomplished by using a reed switch and magnet trigger. The traction monitor consists of a pivoting arm with a small wheel on the end, as shown in Figure 2. A magnet is embedded in the wheel and the reed switch is positioned such that it is activated by the magnet for approximately 30 degrees of wheel rotation. As long as the wheel rotates, a pulse is sent to the analog port. If the robot stops, the wheel on the traction monitor stops, and the robot knows that it is immobile. The reed switch is connected between the +5 volt pin and the signal pin on analog port 2. When contact is

made, the analog port records a value of 255. When no contact is made, the value is considerably lower. Through experimentation, it was determined that when the wheel was stationary, the sensor returned values of either 255 or less than 100. As in the inversion detection code, a counter was created to record the number of times that the current value was equal to the old value, or when the value was less than 100 The pivoting arm allows the wheel on the traction monitor to roll on the ground in both the righted and inverted configuration.

Construction of the compass course monitor was begun and code was written to support its inclusion in the robot. A compass was purchased and modified to fit in the robot body, materials were located to shield the compass from magnetic interference, and tiny IR emitter/detector units were purchased to provide directional information. Figure 3 shows the experimental configuration of the compass system. However, it became apparent that the configuration devised for the IR sensors was not compatible with the compass since they themselves had sufficient magnetic properties to cause large vertical deflection of the needle. This deflection held the needle firmly against the compass interior, rendering it incapable of rotation. This occurred late in the project, and its implementation was canceled in favor of more pressing requirements.

Behaviors

The robot exhibits most of the behaviors needed to meet the objectives. The edge avoidance behavior causes the robot to back up and turn right any time the lower pair of sensors detect low levels of infrared light. In obstacle avoidance mode, the robot turns left or right depending on which sensor detects the greatest intensity of infrared light. If the readings are similar, the robot backs up and then turns right. The traction monitor activates any time the robot is immobile for too long (~1.5 seconds). This causes the robot to back up and turn right. In addition the robot can perform all of these behaviors equally well while inverted. This is possible because the inversion sensor detects the robots attitude and modifies the sensor variable assignments and the motor directions and variable assignments. Changing the direction and assignment of each motor is necessary to allow the robot to perform maneuvers properly. For instance, the commands which cause the robot to back up and turn left while righted would cause it to drive forward and turn left while inverted. Finally, when the previous behaviors are inactive, the robot drives straight and swiftly forward, charging over insignificant obstacles.

Results

Qualitative data on the Juggernaut would occur in the form of graphs of the IR detector output as a function of obstacle distance from the robot. However, it was found that the output readings of the IR detectors were highly variable. Threshold values for the obstacle avoidance and edge detection subroutines had to be constantly modified to account for these variations. For testing the robot, the threshold values were typically set so that the robot could approach a reflective obstacle to within 1.5 feet.

The all-terrain capability was tested first on cluttered workbenches and piles of debris. Initial results were promising, the robot negotiated these obstacles quite easily. The robot functioned well in the outdoor environment, until a crucial shortcoming was discovered: outdoor debris would become tangled in the treads, resulting in excessive tread tension. Even the braced wheel mounts could not withstand this force, and the end result was tread derailment. This occurred immediately before demo day, and the problem could not be corrected in time to provide an outdoor demo. The robot functioned well on terrain that was free of excessive clutter; the failure occurred while testing the robot in an area of dense shrubbery with much leaf litter.

During initial testing of the robot, it quickly became apparent that the TIPs were pulling too much current. They became extremely hot and the motors would actually not function when they were in the circuit. To remedy this situation, the TIPs were removed and the robot was run on full power all of the time. This arrangement proved satisfactory when used with appropriately placed "wait" statements. For example. Including a 1 second delay between forward and reverse allowed the motors to lose the inertia of rotation in the first direction before switching to the second.

Conclusions

The following list details the work accomplished during the course of this project.

Platform Accomplishments

1. Completely enclosed aluminum body.
2. Custom-built drivetrain with adjustable final-drive ratio.
3. Ball-bearing rolling surfaces.
4. Front anti-torque wheel mounts.
5. Clear front access port facilitates IR experimentation.
6. Cushioned CPU and motor driver.
7. Toothed wheels and treads.
8. Custom-built battery holders.

Sensor Accomplishments

1. Adjustable infrared obstacle avoidance.
2. Adjustable infrared edge detection.
3. Inversion sensing.
4. Traction monitoring.

Behavior Accomplishments

1. Obstacle avoidance.
2. Edge detection.
3. Immobility detection.

4. Inversion sensing.
5. Inverted operation.
6. Hierarchical sensor arbitration.
7. Effective straight-line operation.

There main limitations of the project were tread compatibility with outdoor debris, excessive time spent on drive construction, excessive time spent on body modification, and inability to construct compass navigation device.

Recurring problems with maintaining tread alignment, coupled with catastrophic outdoor test results, lead to the inevitable conclusion that the tread configuration needs redesigning to allow true outdoor operational versatility. Improvements in the tread configuration would include smaller drive wheels to help prevent intrusion of objects into the treads, guards to preclude intrusion of objects into the treads and wider treads for more traction. Some thought has also been given to a bifurcated uni-tread, which would consist of two belts wide enough to enclose the entire platform. Such a configuration would permit superb all-terrain capability, but would be limited by sensor access to the environment.

Replacing the custom drives with commercial units would save time on construction and allow interchanging with different robots. The current drive arrangement is essentially limited to the Juggernaut platform, since much of the drive is integral with the chassis. Stock drive components would decrease the complexity and enhance the flexibility of a future Juggernaut.

Modifying the aluminum box to accommodate the high-tension belts was time-consuming. Constructing a box from scratch would probably have resulted in a stronger chassis, since the main spars would have been made larger, and rigidity designed into the necessary areas instead of retrofitted.

The compass course monitoring system was to be one of the prime developments on this project. Having an innate sense of straight-line traveling, combined with the ability to maintain a specified heading regardless of obstacles, was deemed a worthwhile endeavor to pursue. Since so much time was required for the platform and drive development, little time remained to develop this desired feature. In addition to the lack of time, problems which arose during initial construction of the compass proved too difficult to solve rapidly. The compass system is one which definitely merits further effort.

Code implementation was one of the few procedures in this project that went better than expected. Almost all of the code was written before any testing of the individual subroutines was completed. Thus, many errors were expected upon running the program for the first time. While there were quite a few errors, IC indicated the location and type of each error, making de-bugging almost trivial. Consequently, in less than an hour the program was running on the CPU and the robot was behaving as expected. Only minor modifications were needed to perfect the traction monitoring and inversion sensing.

Writing the code for reversing the motors was more complicated, but it still went smoothly with few errors. Much credit is owed to Drew Bagnell for explaining the need and procedure for manual motor reversing in IC. Overall, writing and de-bugging the program was one of the easiest and least time-consuming tasks in the whole project.

Appendix

1. Infrared Operation
2. Traction Monitor
3. Compass Device
4. IC Code

```
/* initializing globals */

int left_eye, right_eye, left_edge, right_edge, left_motor, right_motor;
float forward = 100.0;
float reverse = -100.0;
float stop = 0.0;
int delay = 100;
float ir_motor_l, ir_motor_r, edge_motor_l, edge_motor_r, bump_motor_l;
float bump_motor_r, tract_motor_l, tract_motor_r, comp_motor_l, comp_motor_r;
int traction_old, traction_new, motor_flip;
int obstacle_detected, edge_detected, bumped, no_traction, flip, offcourse;


void wait(int milli_seconds)
{ long timer_a;
  timer_a = mseconds() + (long) milli_seconds;
  while(timer_a >mseconds()) {
                              defer();
                             }
}


void sensor_read ()

poke(0xffb9,0xff);

{ while (1)
 {
  traction_old = traction_new;
  traction_new = analog(9);
  bump = analog(8);
  flip = analog(7);
  if (flip < 100) /* tells where ir sensors are looking when righted */
   {left_eye = analog(3);
    right_eye = analog(4);
    left_edge = analog(5);
    right_edge = analog(6);
    motor_flip = 1;
    left_motor = 0;
    right_motor = 1;
   }
  else
   {left_eye = analog(5); /* tells where ir is looking when upside down */
    right_eye = analog(6);
    left_edge = analog(4);
    right_edge = analog(3);
    motor_flip = -1; /* reverses motor direction when upside down */
    left_motor = 1; /* tells which motor is on the left when upside down */
    right_motor = 0;
   }
 }
}
```

```
void edge_detection ()
/* if edge is detected, back up then turn right */
{ while (1)
 {
  if left_edge <115 || right_edge <115)
   { edge_detected = 1;
     edge_motor_l = reverse;
     edge_motor_r = reverse;
     wait(500);
     edge_motor_l = stop;
     edge_motor_r = forward;
     wait(250);
   }
  else
   { edge_detected = 0;
   }
 }
}


/* traction monitor */
void traction_monitor()

{
int i = 0;
while (1)
 {
   if (traction_new == traction_old)
     { i = i + 1;
     }
   else
     { i = 0;
     }
   if (i > 2000) /* if traction is lost, back up and turn right */
     { no_traction = 1;
       tract_motor_l = reverse;
       tract_motor_r = reverse;
       wait(500);
       tract_motor_l = forward;
       tract_motor_r = stop;
       wait(250);
     }
   else
     { no_traction = 0;
     }
 }
}

/*  object avoidance  behavior */

void object_avoidance()
{ while (1)
 {
  if (right_eye >99 || left_eye >90 )
```

```
    {obstacle_detected = 1;
     if (right_eye > left_eye)
      {ir_motor_r = forward;
       ir_motor_l = stop ;
        wait(250);
      }
     else if ( left_eye > right_eye )
      {  ir_motor_r = stop ;
         ir_motor_l = forward;
         wait(250);
      }
     else
/* if object is perpendicular to path, back up then turn right */
      {  ir_motor_l = reverse;
         ir_motor_r = reverse;
         wait(250);
         ir_motor_l = forward;
         ir_motor_r = stop;
         wait(250);
      }
    }
  else
    { obstacle_detected = 0;
    }
  }
}


/* heading detection */

void compass_module ()
{


}


/*  bump  sense behavior */

void bump_behavior() /* if bumped, back up and turn right */
{ while(1)
  { if (bump < 190)
    { bumped = 1;
      bump_motor_l = reverse;
      bump_motor_r = reverse;
      wait(150);
      bump_motor_l = forward;
      bump_motor_r = stop;
      wait(150);
    }
   else
    { bumped = 0;
    }
  }
```

```
}

/*  this arbitrates between object avoidance, bump behaviors */
void behavior_arbitrate()
{
float mpowerold_l = 0.0;
float mpowerold_r = 0.0;
float mpowernew_l, mpowernew_r, mpowerx_l, mpowerx_r;

  while(1)
 {
  if (edge_detected)
     {  mpowernew_l = edge_motor_l;
        mpowernew_r = edge_motor_r;
     }
  else if (bumped)
     { mpowernew_l = bump_motor_l;
       mpowernew_r = bump_motor_r;
     }
  else if (no_traction)
     { mpowernew_l = tract_motor_l;
       mpowernew_r = tract_motor_r;
     }
  else if (obstacle_detected)
     { mpowernew_l = ir_motor_l;
       mpowernew_r = ir_motor_r;
     }
  else if (offcourse)
     { mpowernew_l = comp_motor_l;
       mpowernew_r = comp_motor_r;
     }
  else
     { mpowernew_l = forward;
       mpowernew_r = forward;
     }
  mpowerx_l = (mpowernew_l + 19.0 * mpowerold_l)/20.0;

  mpowerold_l = mpowerx_l;

  mpowerx_r = (mpowernew_r + 19.0 * mpowerold_r)/20.0;

  mpowerold_r = mpowerx_r;

  motor(left_motor, mpowerx_l * flip_motor);
  motor(right_motor, mpowerx_r * flip_motor);

 }
}

void main()
 {
  start_process(sensor_read());
  start_process(edge_avoidance());
  start_process(traction_monitor());
```

```
 start_process(obstacle_avoidance());
 start_process(compass_module());
 start_process(behavior_arbitrate());
}
```