CALCUTRON
# Table of Contents

- Introduction ................................................................. 3
- Hybrid Based Control ..................................................... 4
- Hybrid Architecture ....................................................... 5
- Object Avoidance System ............................................... 7
- Find and Capture Egg .................................................... 10
- Deposit Egg ................................................................. 12
- Sensor System ............................................................. 13
  - Sonar Sensor .............................................................. <page>
  - Touch Bumpers .......................................................... <page>
  - IR Distance Sensor .................................................... <page>
  - IR Egg Identification Sensor ....................................... <page>
  - Photo Resistive Light Sensor ....................................... <page>
  - System Interrupts ...................................................... <page>
  - The Gate ................................................................. <page>
  - Goal Detection Array ................................................ <page>
  - Egg Color Switch ..................................................... <page>
  - Unused Systems ....................................................... <page>
    - Claw and Lift Mechanism ......................................... <page>
    - Side Mount Distance Sensors ................................. <page>
  - Contest Performance ............................................... <page>
- Appendix: Labview Generated C Code .............................. <page>
Introduction

The context of the Robot Egg Hunt contest is the educational theory of constructionism by Seymour Papert. His writings suggest that students learn effectively by building objects based on what they are learning. The robots are the objects of our learning. Through the robots we experienced the common challenges of an engineering environment: teamwork, time constraints, and material limitations. The robot contest is also a collaboration with the graphic arts students much like what we would experience in small company. We worked together with them to develop a theme and marketing solution for our robot.

The contest consisted of an enclosed rectangular arena. There were several black and yellow eggs within the arena. Two robots compete at each round. At the beginning of each round the robots are designated the color of eggs to capture. The robots have two minutes to bring as many of its own eggs to the goal where a bright light is shining.

Hybrid Based Control

All of the robots developed during the contest used control systems that were discussed in class. One of the challenges in controlling robot behavior is to develop a control system that allows the robot to adapt in dynamic environments while maintaining a high level of performance. Various robot architectures have arisen to solve these problems. Among the architectures are deliberate control, reactive control, hybrid control, and behavior based control. These control systems fall within the control system spectrum where at one extreme you have complete sequential control at the other extreme you have reactive control. In sequential control the robot goes through a lockstep method of deciding behavior. In reactive control each module has its own stimulus and response mechanism that acts autonomously from the other modules. Calcutron falls somewhere in the middle where it is uses a hybrid architecture.

![Figure 1: Layout of hybrid architecture](image)

Figure 1: Layout of hybrid architecture
The hybrid system that was created is a three layer system with an arbiter. At the top layer is the sensing and awareness. In this layer Calcutron acquires information about its environment. The robot is in a continuous loop where the information is sent down from the Sensing layer and is processed by the Arbiter. The Arbiter decides whether to perform one of four tasks: Deposit Egg, Capture Egg, Find egg or Avoid object. Each of these responses is a finite state machine that is triggered by a stimulus. The four responses are prioritized. The highest priority is given to the object avoidance module. If there are no objects blocking Calcutron then the next highest priority is the Deposit Egg. Then it is the Wander and Find Egg module. Lastly is the Capture Egg module.

**Hybrid Architecture**

The hybrid architecture is composed of a top-level main function that calls the other functions. The main function is in the outermost while loop. This main function calls the sensing and arbitration modules. The arbitration is expressed with a case structure within which various behaviors are called. The left most loop is an infinite while loop that exits only when the start button is pressed. Therefore, after the code is loaded onto Calcutron, he remains idle until the start button is pressed.

Figure 2. Main routine
The decision loop is the method of arbitrating the various input signals. When the while loop has not been interrupted by the time interrupt, the arbiter decides among several sensory inputs such as right ET sensor, front ET sensor, left sonar, front bumper sensors etc. Each sensor is has a priority. When any sensor is activate it places its priority value into an array. A decision is made based on the priority within the array.

Figure 3: Decision loop

Once a decision has been made by the arbiter, sequence of steps are performed within the selected behavior. After the sequence of steps are performed, Calcutron then returns to sensing to decide on the next sequence of steps.
The Avoidance Objects module responds to the inputs from three different sensors. This module can be seen within the main function (see figure 1). It is not a separate subvi. It is the cluster of sensors that feed into the case structure. There is a front ET sensor that detects objects directly in the front. There is also a ET sensor that detects objects to the right of the robot. And then there is a sonar sensor that detects objects to the left of the robot. The object avoidance module is constantly acquiring data from the environment. The input data is compared to a constant distance threshold. An object that is too close will generate a reading greater than the threshold and the object avoidance module will subsume control of the motors. The direction of turning is determined by the case structure seen in the figure below. If the object is directly in the front, it calls the back away subvi otherwise it calls the change direction subvi.

There are three possible scenarios for object avoidance. The object can be in the front, to the left, or to the right. Calcutron is capable of detecting object on the left when an interrupt is generated from the left touch sensor. In this case it would call the turn right module. Above shows the call to the turn right module. In this module Calcutron would back away and then move its right side and left side wheels in opposite directions to turn away.

If the right touch sensor is activated however the response would be to call the turn left module as seen in the above figure. Similarly, if the touch bumpers were activated in the front it would call the reverse module which would call Calcutron perform a more extensive back and turn away.
Figure 5: Turn left module has been invoked.

Figure: Reverse module has been invoked.
Calcutron goes into a Wander and Find Egg state when it is not interrupted by either the Deposit Egg module or the Object Avoidance module. In this state it goes forward but veers slightly towards the left until it hits the wall and then it turns towards the right by 45 degrees then it goes straight again with a tendency towards the left again until it hits the wall and then it repeats this process. The idea is that it encounter some of the eggs in the middle of the court additionally other eggs at towards the wall would also be captured as well. The understanding is that many of the eggs will end up towards the edge of the court.

Calcutron has front mounted sonar sensor that points towards the ground to detect the presence of an egg in front of its gates. When the sonar reading falls below 20 this indicates that there is an egg in front of the gate. At which time it enters the Capture Egg state. In this state it stops wandering. It invokes the Egg ID module to identify the egg. Below shows the Egg ID module.

**Figure:** Egg ID module is called.

The Egg ID module checks the egg type. There is a toggles switch that has been built. The egg type is set by the toggle switch at the beginning of the contest. It is connected to the digital inputs on the handy board and corresponds to either 1 or 0 for a yellow egg or black egg. The figure below shows the egg identification sequence.
The Sense module is called. There are several possible outputs from the sense module. It is the all encompassing sensory input module. Among the values it grabs from the sensors is the IR Sensor for egg detection. When the sensor is below 500 the egg is a yellow egg. If it is above 500 it is a black egg. The exclusive NOR logic is used to connect the switch with IR sensory input. This is because the egg identification is the same logic as the egg type switch.
The Deposit Egg module is activated when the Arbiter realizes that a certain amount of time has passed. This value is hard coded into the program as 30 seconds. The deposit egg module calls two other modules. First it calls the light-seeking module. This module causes Calcutron to move the motor proportionally towards the location of the greatest amount of light which is the goal. It continues to go towards the area of greatest amount of light. Once a threshold amount of light has been detected and a high mounted IR distance sensor detects the crossbar over the goal Calcutron then calls the dump egg module. In this module, Calcutron opens the gate, backs up, closes the gate then it pushes the egg into the goal.

Below is the deposit egg module. The gate is connected to a servo motor. When the servo motor is set to 825 the gate is opened. Calcutron then backs away for 3 seconds. Then it closes its gates by setting the servo motor to position 150. Then it goes forward and pushes the eggs into the goal.

At the end of the sequence it backs away and turns left. Then it exits the behavior module and continues wandering and looking for more eggs.
A robot’s sensors are essential for it to react to its environment. All sensors connected to Calcutron are either digital or analog. The handy board contains 12 analog input and 8 digital inputs. Electrically, both analog and digital input connections look similar. All if the connections contain a three ports: sensor, +5V supply, and ground. The sensor port and the ground port is required by all sensors. The sensor port is a 5 volt connection with a 47K resistor, this acts as a voltage divider if a sensor is connect. The ground port completes the sensor circuit for the voltage divider. A 5 volt supply port is available, however not all sensors need it.

The sensor values are called from the module called All Sense. This module is located within the top level while loop.

**Figure:** All Sense module being called within main().
The All Sense module contains two digital readings, two sonar, and three analog readings. The two digital readings are from the touch bumpers. There are two sonar readings, one that sits above the gate for detecting the presence of an egg. The other sonar reading is for left-sided object avoidance. The three IR sensors are for IR distance sensors, IR egg detection, and IR right sided object avoidance. All of the sensory values are acquired when Sense All is called and the values for each reading are thresholded to its respective value. For example, the right IR object avoidance has threshold of 300. Any value above 300 would cause the IR sensor to place a priority value into a priority array. The array generated would look something like the array below.

<table>
<thead>
<tr>
<th>Sensor</th>
<th>Priority Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Right IR sensor</td>
<td>300</td>
</tr>
<tr>
<td>Left Sonar Sensor</td>
<td>200</td>
</tr>
<tr>
<td>Front IR Sensor</td>
<td>100</td>
</tr>
<tr>
<td>Right Touch Sensor</td>
<td>1000</td>
</tr>
<tr>
<td>Left Touch Sensor</td>
<td>2000</td>
</tr>
</tbody>
</table>

The algorithm then searches the array to find the maximum priority value. The array index for this maximum is placed into the case structure and the behavior is chosen. In this case the maximum array priority is 2000 therefore the array index of 4 would be generated and sent to the case structure decision.
The sonar sensors are similar to the way bats sense. It sends out a high frequency signal and it waits for a response. The speed of the transmitted sound is one foot per millisecond. It generates a distance reading based on the time elapsed between sending and receiving the signal. The readings encountered by Calcutron were between 0 and 600. The errors associated with sonar sensors may come from reflection off of non flat surfaces.

Figure: Left Side Mounted Sonar Sensor
It may also generate a false reading when an opponent sends out a signal as well. Other sources of error may be nearby surfaces. A flat surface near the sonar sensor could easily cause false readings. It is therefore essential that there is a clear path for the sonar sensor to send out signals. The sonar sensors were mounted at two locations. One was mounted at the left side of the vehicle as a left-sided object avoidance sensor. This sensor had a threshold of 20. Any readings below 20 meant that there was an object to avoid that was to the left of the vehicle. The other sonar sensor was above the gate. This one was for egg detection. It had a threshold of 9. Any readings below 9 means that there is an egg near the gate. The sonar sensor for egg detection was more error prone because it had objects nearby and the threshold value was so low.

**Touch Bumpers**

The touch sensors uses a digital input. It therefore can generate either true or false. There are two types of microcircuit for touch sensor: normal open and normal closed. A normal open means that it is open circuited until it is touched, therefore the circuit experiences is high impedance until it is touched. The V sense at that point would be close to 5 volts. When it is touched it is closed circuited. The V sense at that moment would be close to zero volts. A normal closed circuit means that it is closed circuited when there is no one touching the sensor. When the sensor is touched then it is open circuited. The type of touch bumper sensor used by Calcutron is a normal open. The logic in the software reflects this characteristic. The software generates false when the bumper is untouched and it generates true when it is touched. In an untouched state V sense has a voltage of 5 volts which generates false at the software level When it is closed circuited V sense has a voltage of 0 volts which generates true at the software level. There is a spring mechanism that pushes the touch sensor out in its rest state. There are two touch sensors mounted at the front of Calcutron. One is mounted to the left and the other is mounted to the right.

**Figure** Full Bumper (Front View)
There is also an arm lever that protrudes out from the touch sensors such that any slight touch would generate a true signal at one of the touch sensors to the Handy Board. Calcutron can differentiate between a right touch and a left touch. Frontal touch is also detectable. The crossbar connects to the arm levers of the right and left touch sensors. This arm lever allows both sensors to be activated simultaneously. The touch sensors were integrated into the object avoidance behavior. A left touch sensor activation would cause Calcutron to turn to the right. A right touch sensor activation would cause Calcutron to turn to the left. A frontal activation would cause Calcutron to back away significantly and then turn away.

**Figure** Touch Sensor Detection.

---

**IR Distance Sensor**

The “ET sensor” is the distance sensor that uses infrared light to measure the distance. The actual part used was the Sharp GP2D12 which can measure from 10 cm to 80 cm. Since the minimum was at 10 cm, it had to be displaced slightly within the robot so that it does not pickup any false readings. As
was discovered any distances less than 10 cm would generate readings indicating that the object is farther away. The sensors is a reflective optosensor which is considered an active sensor since it generates its own light source. The ET sensor has two elements, an emitter and a receiver. Infrared light is sent out from the emitter and a phototransistor is the receiver. The ET sensor is connected to an analog port and it generates a continuous reading. As an object gets closer the sensor, the reading from the sensor would get higher. There are three wires connecting the sensor to the Handy board. There is the ground, power, and the Vsense. ET sensors were originally attached to front and the right side of Calcutron. This would have allowed Calcutron to detect objects directly in the front and to the right. The sensors were mounted high enough such that they were above the level of the egg. Therefore they would serve only to avoid obstacles. The sensors readings ranged from 50 to 900. They were threshold at a reading of 300. This maintained 2.5 inches between Calcutron and any object. At 100% motor speed, it allowed just enough time for Calcutron to respond to obstacles in its path.

IR Egg Identification Sensor

The infrared identification sensor is a reflectance sensor that generates readings between 100 and 900. An emitter sends out a light and the detector receives the light. Different colored surfaces acquire light at different readings. Yellow eggs generate lower readings than black eggs. The problem with the IR sensor is that it has to be held close to the egg to accurately detect the egg. It also had a tendency to
generate false readings. When Calcutron encountered a yellow egg that was close to the IR sensor, it would generate a reading that was less than 500. Blacks eggs easily generated readings above 500. The ambient reading was above 500. False readings occur when the egg is not close enough to the sensor and it generates the ambient room reading.

The inclination therefore is to towards either extremes. Calcutron either capture both yellow and black eggs. The other extreme is that it would not capture any eggs. The strategy was to allow to capture a lot of eggs and hopefully the slight differentiation that it does do would be enough to offset the number of opponent eggs.

System Interrupts

For Calcutron's overall system we used the start and stop buttons located on the handyboard controller. Once the program is loaded onto the handyboard it continuously runs so in order to not have the robot running around except during the rounds we included a digital signal capture in the sense module that read when the stop button was pressed, the arbitrator sent control into a mode that consisted of a stop state which stopped the motors from running. It then sat in a tight loop waiting for the start button to be pressed. Once this occurred, the final state in this mode was a start module, which started the motors at full power going forward. Control then passed back to the sense module/ arbitrator loop. Shown in the picture below is the tight loop waiting for the press of the start button.
The Gate

The primary function of the gate was to entrap eggs under the body of Calcutron. It was constructed of a basic frame with a pivot shaft running across the top of the gate. It also had two levers at a ninety degree angle to the gate pointed forward. On Calcutron's right side was the lever to close the gate which had an elastic string tied to it to pull the gate to the closed position. On the left was the opening lever which was connected by several strands of thread to a servor which controlled the gate position. As mentioned earlier the gate was opened to allow eggs to be "eaten" by Calcutron and then closed to trap them under the body. This was done by sending a signal to the servor to move to position 825 to open the gate and then set to position 150 to remove the tension from the servor which allowed the elastic string to pull the gate closed.

Picture: Gate opening lever (at right)
Gate is the part with the red wires
The goal detection array consisted of three photo-resistive light sensors encased in tubular shielding of about an inch and one half in length. The array was lifted on a mast to the height of the light which identified the goal. This was done to try to eliminate as much ambient light as possible by only allowing light in the plane of the goal light to enter directly. The three tubes all pointed toward the front of Calcutron. The center tube was aligned directly forward and the tubes on either side were rotated approximately forty-five degrees from the center tube. The raw readings were measured and then we essentially determined the fraction of darkness that
Figure: Top view of the light sensor.

existed. It seems counter-intuitive to deal with the darkness instead of the light intensity but the low end readings from the sensor indicated the highest light intensity, while the high end readings came from the lowest light levels, so in effect the low to high values measure the amount of darkness that exists in that place. After determining this fractional portion as a floating point number and multiplying by ten to normalize to a value between 0 and 10, we separated the function of the two outer sensors and the center sensor.
The value from the outer sensors was converted to integer values and used to control whether Calcutron turned right, turned left, or continued straight. Rather than getting involved with complicated threshold values for each sensor, the conversion from floating point to integer values before comparison effectively gave us variable threshold values based on whether one of the outer sensors was reading an amount of light close ( +/- <1 after normalization between 0 and 10) or a much greater amount of light than the other sensor. The center sensor was used primarily for speed control when Calcutron was facing the goal and headed towards it. It's normalized value was compared to five and if it was greater, that value was multiplied by the normalized value (unconverted to integer) of each outer sensor and run directly to the motor as the power level on the same side as the sensor. This had the effect of slowing Calcutron as he approached the goal, and doing some fine tuning of the centering of the center sensor on the goal light. We didn't want him to go too slow when approaching the goal, so if the center sensors normalized reading was less than five, five was used as a constant instead of the normalized reading in the power level calculation. At this point we were getting close to the goal and were fairly well centered on the light so the outer sensors tended to have a normalized reading around eight which would correspond to about a forty percent power to each of the motors. This slower speed allowed a few more readings from the distance sensor so we wouldn't miss the threshold on the distance sensor. The normalized center sensor value was then compared with a hard coded threshold of one half. This would indicate to us when we were about one foot from the goal. This signal was sent to an and gate that got its other signal from a comparison of the front mount distance sensor and a threshold. The result of this and was used as the control to break out of light seeking and enter the dump eggs mode.

Egg Color Switch

In order to determine which color of eggs we were looking for we included a switch at the top of the frame of Calcutron. This was a three pole, two position switch in which we only wired ground and one pole. We then ran this to a digital input on the handyboard.

<table>
<thead>
<tr>
<th>Sensed Color</th>
<th>Desired Color</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yellow</td>
<td>T F</td>
</tr>
<tr>
<td>Black</td>
<td>F T</td>
</tr>
</tbody>
</table>

This was then sent through an xor gate with the signal from the IR reflective sensor that was determining the egg color to determine whether or not we should capture the egg. As seen by the above table, if yellow was considered true and black false, we would have a truth table for xor.
Unused Systems

Claw and Lift Mechanism

Side Mount Distance Sensors
In the first round of the contest, Calcutron faced Ms. Boltz and he seemed to be operating fine. About thirty seconds into the contest however, Ms. Boltz became disabled at the side of the goal. This created a blind canyon between Ms. Boltz, the goal wall and the side wall. Calcutron wandered into this blind canyon and was unable to escape from it. Once Calcutron switched to goal detection mode it became apparent just how stuck he was. Since Ms. Boltz was between him and the goal, all attempts to head toward the goal resulted in the front bumpers being activated and then a return to goal seeking mode. This was how the round ended. Calcutron won since in the process of becoming disabled, Ms. Boltz pushed in one of the eggs he was trying to score with.

In the second round, the opponent was Poacher. This round was another futile round as after doing some egg collecting, Calcutron attempted to go to the goal and dump them in. Poacher had become disabled in the fact that he was running in goal seeking mode only. Since Poacher was built as a faster robot and was essentially doing circles in front of the goal, every attempt by Calcutron to reach the goal was repelled when the touch bumpers were activated by contact with Poacher. Again Calcutron won by the fact that his opponent managed to bump in one of Calcutrons eggs and none of their own.

The third round was very disappointing for Calcutron. For the first time Calcutron was force to look for eggs of the opposite color and it soon became apparent that the threshold value for the identifying sensor was incorrectly programmed. Calcutron didn't collect any eggs but almost managed to push some into the goal just by plowing them in during goal detection. The opponent this round was Ex-bot which had a motor fail partway into the round. No eggs entered the goal and a dual loss was recorded.

The final round matched Calcutron up against Scrambler. Scrambler scored three eggs and Calcutron had collected at least two eggs and switched to goal seeking mode. At this point it became apparent that there was a failure with him as during goal-seeking and egg dumping modes, one of his motors was failing intermittently. Scrambler became the tournament champion and investigation of the failure pointed to the hurried reloading of code to Calcutron before the final round. Mike had allowed us to adjust the threshold on the identifying sensor but, in order to successfully load the code to Calcutron we had to reboot the computer and try to reload. After rebooting, in the haste, we started the build of the code prior to the point where Labview had loaded all the tables of connections and hurriedly hit save to get the build underway. This caused the new saved file to be built without the power settings for one of the motors when Calcutron was going forward during light-seeking and the entire dump egg mode.