1. DR. RAMESH YERRABALLI: So we've looked at a transducer which
2. was a potentiometer that gave us a voltage value
3. from a corresponding variable resistance.
4. Jon, can we look at a different device that
5. has some significance that we can then use to control something?
6. DR. JONATHAN VALVANO: Yeah, let's look at the sharp sensor.
7. The sharp sensor is an infrared distance sensor.
8. That means it takes a distance to an object
9. and it will convert it to a voltage.
10. This means if we connect it up to PE2 in our microcontroller
11. then our software has access.
12. Because it's an A to D, we can get a number
13. which is between 0 and 4,095 which represents the distance.
14. In order to understand this, we need to look
15. at the voltage created as a function of distance of this sensor.
16. And this is a very unusual and nonlinear sensor that has this sort of behavior.
17. It has two phases here.
18. This first phase where it's really close,
19. and that's less than 100 millimeters, it behaves in this weird way.
20. But after 100 millimeters it behaves in a very simple way
21. such that the distance is approximately equal to some constant
22. over the voltage.
23. And so I can use this relationship to write software such that my distance
24. variable-- this is now in software, in millimeters--
25. is equal to the sum calibration constant.
26. I did calibrate it.
27. And so the result I got was 241,814 divided by this ADC value--
28. this number that comes from ADC.
29. So what we have is a software conversion here
30. which converts the number 0 to 4,095 which is measured by the a
31. to d into a distance in millimeters which works from about 100
32. millimeters out to 700 millimeters.
33. DR. RAMESH YERRABALLI: So just to recall,
34. the voltage to distance graph here is non-linear.
35. Whereas for our potentiometer the voltage
36. to distance-- if you think of the potential meter
37. as measuring distance-- then that would be a linear graph.
38. DR. JONATHAN VALVANO: That's correct.
39. And the software doesn't mind dealing with non-linearity
40. because we can just compensate for it in this equation.
41. But what do we use it for?
42. DR. RAMESH YERRABALLI: Well, we've been promising
43. that we will build a robot car that can avoid colliding with walls by sensing.
44. So let's put this to work.
45. DR. JONATHAN VALVANO: All right.
46. So the next step in our control system is to look at the blocks.
47. The parameter we want to control is called a state variable.
48. In our case it will be where we are in the road.
49. We're then going to use transducers to estimate what we know
50. and learn where we are in the road.
51. And then we're going to use a control equation, which in our case
52. will be software.
53. And then use these outputs to drive actuators that affect the real world.
54. So this is our world, which is our robot on the road.
55. And we're going to estimate where we are on the road--
56. DR. RAMESH YERRABALLI: By using a transducer that is our infrared sensor.
57. DR. JONATHAN VALVANO: And the A to D converter.
58. DR. RAMESH YERRABALLI: And we perform some logic
59. which is the control equation.
60. This is the intelligence part of our code.
61. DR. JONATHAN VALVANO: And we're actually going
62. to drive 2 pulse with modulated outputs for the left and right motors.
63. And you notice that a control system has a loop, a feedback loop.
64. And again, the desire is to drive the state variable to a certain set point.
65. All right, let's put the pieces together.
66. Let's look at more detail of the state variable.
67. Here's our road.
68. And we're going to have a wall and another wall.
69. And we want to avoid the wall.
70. And we're going to have our robot car here, which
71. is moving in this direction.
72. And we're going to have 2 sensors on the front of the robot.
73. And we're going to measure the distance between the robot and left wall
74. and the robot and the right wall.
75. So we'll call this Dleft, that's the distance to the left wall,
76. because we're going this way.
77. And this is the distance to the right wall.
78. And how do you know if we're in the middle of the road?
79. DR. RAMESH YERRABALLI: I guess Dleft should be equal to Dright.
80. DR. JONATHAN VALVANO: Yeah, so what we're going to see
81. is, we're going to say we are going to look at our error.
82. In other words, we are going to be unhappy
83. if the distance to the left wall is different--
84. so we'll calculate the difference to the right wall.
85. And so the goal state is to drive the error to 0.
86. Because if this state variable error is 0 we're in the middle of the road.
87. So you remember our robot had 2 wheels.
88. And so what we have is PWM outputs for both wheels.
89. And so for each PWM outputs-- for instance
90. we have the right wheel high over the right wheel high plus right wheel low.
91. And this controls the power to the right motor.
92. In this example we're going to fix this to 40%.
93. So the right wheel is going to spin away at 40%.
94. The interesting part will be on the left wheel.
95. On the left wheel we're going to take the left high and let it change.
96. And again, the pulse with modulated signal for the left wheel
97. is the left high-- the time in which the left output is high
98. plus the time in which the left is low.
99. This pulse width modulation here, this duty cycle, is going to vary.
100. We're going to let it vary somewhere between 30% and 50%.
101. So as this varies between 30% or 50% it's
102. going to be going straight-- if its 40, 40.
103. 40, 30 will turn one way.
104. 40, 50 will turn the other way.
105. So this parameter here is the one I'm going to adjust in my control loop.
106. So let's look at the software flow chart.
107. This is a real time problem so we're going
108. to use SysTick interrupts to run our control loop.
109. So every periodic time, at about 100 times per second,
110. we're going to run this control loop.
111. We're going to measure the distance to the right and the distance to the left.
112. We're going to calculate the error, which
113. we saw last time, which was the difference
114. between the left and the right.
115. And now comes the fun part.
116. What do we do with it?
117. What I'm going to do with it is I'm going to actually look at the left high
118. and I'm going to adjust it.
119. Either spin it faster or slower, depending upon the error.
120. And I'm going to do this in a typical control way by using a number.
121. It's called the gain of this control system.
122. And I'm going to take the error and multiply it times 200
123. and subtract it off.
124. So if I'm too far to left, it'll turn right.
125. If I'm too far to the right, I'll turn left.
126. Notice what happens here if the error is equal to 0.
127. What happens if the error-- if I'm in the middle of the road, what
128. does my control system do?
129. DR. RAMESH YERRABALLI: So if the error is equal to 0, the left high stays put.
130. It doesn't change.
131. DR. JONATHAN VALVANO: So if we're happy, we'll stay happy.
132. DR. RAMESH YERRABALLI: And remember that we've
133. kept the right high at a constant.
134. So that doesn't change either.
135. So now we don't change either of them, we stay the course.
136. DR. JONATHAN VALVANO: So if we're in the middle,
137. we'll just keep being in the middle.
138. There's one more couple of problems we have
139. to deal with is that we have to make sure that it doesn't go below 30%.
140. And so if the duty cycle goes below 30%, we will set the duty cycle at 30%
141. so it doesn't completely stop or try to spin backwards.
142. If it's OK then we will continue.
143. So if we look at the duty cycle here of the left wheel, if it is too slow
144. we'll fix it at too slow.
145. Similarly, if it's too fast, we'll look at the duty cycle.
146. If it is greater than 50% then we will set the duty cycle at 50%.
147. And these two comparisons will force the range here to between 30% and 50%.
148. We do have to calculate the other side.
149. The left low has to be set so that this number here is a constant.
150. And that constant is 80,000 minus the left high.
151. And then we return.
152. So these are the steps of my control loop.
153. We sense our inputs.
154. We decide if we're were happy.
155. We adjust the actual actuator output so that we become more happy.
156. In other words, if we're too far to the left, we go right.
157. If we're too far to the right, we go left.
158. And this is a gain.
159. In other words, how fast do we turn?
160. In a control system we want to make sure we don't go crazy.
161. So we'll check and make sure we have a minimum and a maximum value that
162. makes sense for our actuator.
163. And then we perform the output.
164. And zoom, our robot is spinning.