Adaptive Cruise Control

TI-Nspire™   Investigation   Student   50 min

Introduction

Basic cruise control is where the car's computer automatically adjusts the throttle so that the car maintains a constant speed. This is a great feature when the car is travelling on a straight open highway with no other vehicles nearby. Problems occur when the car negotiates a corner, travels down a steep hill or encounters another vehicle travelling in the same direction but somewhat slower. Adaptive cruise control accounts for all of these alternatives. In this activity you will learn how to control the speed of a vehicle (TI-Innovator Rover) based on data being collected from its sensor(s).
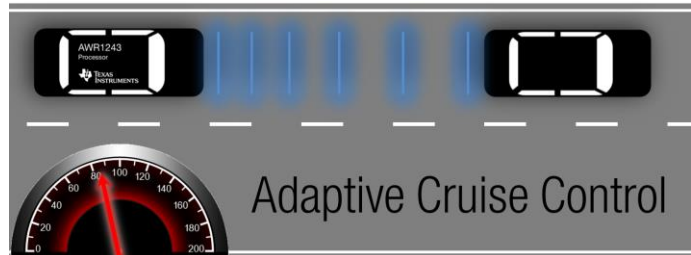
In this activity you will use the sensor on the front of Rover to make adjustments to Rover's speed.

To see some of the features of Adaptive Cruise Control, check out the video: http://bit.ly/AdaptiveCruise

Reading Rover's Sensor

Some Adaptive Cruise Control (ACC) systems will bring the vehicle to a stop if the vehicle in front has already stopped and then resume moving as the vehicle in front moves off. This is the aim of the first part of this investigation. To achieve this result you will need to know some basic commands:

- Send "RV FORWARD"
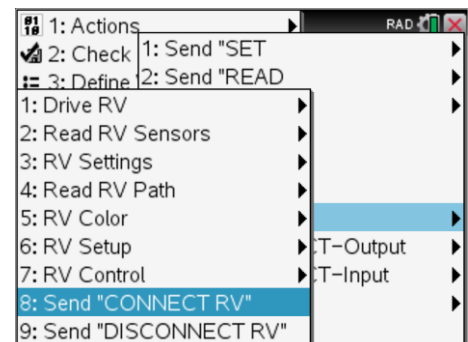- Send "READ RV.RANGER"
- Send "RV STOP"
- If ... Then

Create a new TI-Nspire document and insert a program titled:

ACC

The first line of any Rover program starts with:

Send "RV CONNECT"

Author: P. Fox

The next step is to read the Ranger on the front of Rover.

> Hub > Rover (RV) > Read RV Sensors >

> Send "READ RV.RANGER"

Reading "Ranger" is not enough; the result must also be returned to the calculator and stored.

> Get d

Note that 'd' is simply a variable where the result will be stored. For simplicity it has been called d to represent the distance.

Tip

> "Display" commands are useful tools to make sure your program is working as expected. To see what your program is doing it is a good idea to place the following command after the 'get' statement:
>
> > DispAt 1,"Distance: ",d

If Rover is too close to the vehicle in front it should remain stationary. If on the other hand the vehicle is more than 20cm away it may be appropriate for Rover to move.

This can be coded as:

> If d>0.2 Then
> > Send "RV Forward 1.5"
> > Endif

The Drive command is available from:

> Hub > Rover (RV) > Drive RV > FORWARD

Place Rover on a smooth flat surface, preferably the ground (so it will not fall off the table). Place a box in front of Rover, approximately 15cm away. Press Ctrl + R to save and run the program.

Question: 1

Does Rover do anything?

Question: 2

Move the box so that it is approximately 40cm away. Press [Enter] to run your program again. What happens this time?

Question: 3

Change the: Send "RV FORWARD" command to read: Send "RV FORWARD eval(10d-0.2)" Press Ctrl + R to run the program again. Try several different locations for the box and explain what is happening.

Author: P. Fox

TEXAS INSTRUMENTS

So far the program only takes one look at the object in front. What if the object moves? Rover must repeatedly look at the object in front and adjust its movements accordingly.

Two adjustments need to be made to the program.

In the IF statement, an 'ELSE' option is included:

> Else
> Send "RV STOP"

Now if Rover is travelling towards the box and the box gets closer than 0.2m away, Rover will stop immediately.

The second change is to make the program run repeatedly. This will be done with a "WHILE" loop.

Use the SHIFT key to highlight all the programming code from:

> Send "READ RV.RANGER"
> ...
> EndIf

From the menu select:

> Control > While ... EndWhile

Use the condition shown opposite: "d>0.1"

Warning

The loop will only be active 'while' the value of d is greater than 0.1. An additional statement can be added to the program before the start of the While loop: d:=1.

Alternatively the value of d can be set from the Calculator application prior to running the program.

Question: 4
Run the new version of the program and explore what happens whenever the box is moved away or toward Rover.

Extension – Adjusting the Throttle
The program so far has Rover travelling at full speed when the road ahead is clear and stopping when an obstacle is detected. Adaptive cruise control fills provides more flexibility by gradually slowing as an object appears closer. It is possible to adjust the speed of Rover's motors gradually using slightly different commands.

> "SET RV MOTORS LEFT # RIGHT # "

The # symbol used here represents a numerical value between -255 and +255 where the negative indicates counter-clockwise (CCW) rotation of the motor and positive is clockwise (CW). It must be noted however that 'small' values will fail to provide sufficient power to rotate the motors. A series of "IF" statements could be used to vary the power to each motor, however in this activity a "Piece-wise function" will be used.

    Author: P. Fox

TEXAS INSTRUMENTS

Navigate to a Calculator application. A rule or mathematically speaking, a 'piecewise-function' will be defined that determine the amount of power sent to each motor.

Actions > Define

Type in the function name: $s(x) =$

From the templates menu select the piecewise template.

In the piecewise template select "4" pieces.

The image opposite shows that the definition for the first two pieces. When x (distance) is larger than 1m the function will return the value 255, this means the motor will receive full power and rotate CW.

If the distance is between 0.5m and 1m then the motor will receive 200 units of power, almost 80%.

Complete the definition so that rover's motors will receive 150 units of power when the distance is between 0.2m and 0.5m; and no power if the distance is less than 0.2m.

Return to the ACC program. Remove the entire IF statement and replace it with the SET RV.MOTORS command.

Hub > Rover (RV) > RV Control > SET RV.MOTORS

The LEFT and RIGHT commands can be obtained from the Rover menu – RV SETTINGS. The 'eval' command is an abbreviation of the word 'evaluate' and can be obtained from the HUB menu.

Send "SET RV.MOTORS LEFT eval(-1·s(d)) RIGHT eval(s(d))"

Press Ctrl + R to save and run the program.
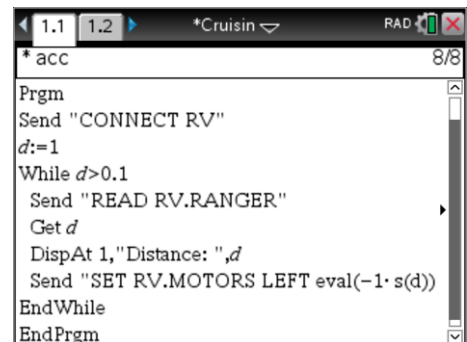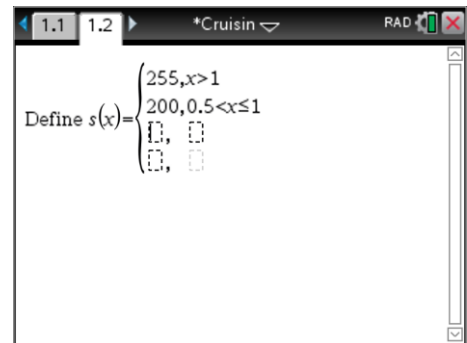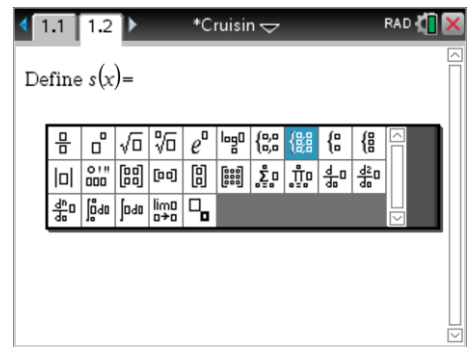
Question: 5

How does Rover respond when an object is placed or moved along in front as it is driving?

Question: 6

The evaluation contained in the LEFT argument of the RV.MOTORS command multiplies the result by negative 1. What does this do and why is it necessary?

Question: 7

The magnitude of the power sent to via the LEFT and RIGHT arguments is the same. How would Rover move if these values were different? How could this be used for obstacle avoidance?

Author: P. Fox