

Surprise!

We've just been given a robotic ant that has an FSM for its brain.

The inputs to the FSM come from the ant's two antennae, labeled L and R.

An antenna input is 1 if the antenna is touching something, otherwise its 0.

The outputs of the FSM control the ant's motion.

We can make it step forward by setting the F output to 1, and turn left or right by asserting the TL or TR outputs respectively.

If the ant tries to both turn and step forward, the turn happens first.

Note that the ant can turn when its antenna are touching something, but it can't move forward.

We've been challenged to design an ant brain that will let it find its way out of a simple maze like the one shown here.

We remember reading that if the maze doesn't have any unconnected walls (i.e., no islands), we can escape using the "right hand rule" where we put our right hand on the wall and walk so that our hand stays on the wall.

Let's try to implement this strategy.

We'll assume that initially our ant is lost in space.

The only sensible strategy to walk forward until we find a maze wall.

So our initial state, labeled LOST, asserts the F output, causing the ant to move forward until at least one of the antennae touches something, i.e., at least one of the L or R inputs is a 1.

So now the ant finds itself in one of these three situations.

To implement the "right hand rule", the ant should turn left (counterclockwise) until its antennae have just cleared the wall.

To do this, we'll add a rotate-counterclockwise state, which asserts the turn-left output until both L and R are 0.

Now the ant is standing with a wall to its right and we can start the process of following the wall with its right antenna.

So we have the ant step forward and right, assuming that it will immediately touch the wall again.

The WALL1 state asserts both the turn-right and forward outputs, then checks the right antenna to see what to do next.

If the right antenna does touch, as expected, the ant turns left to free the antenna and then steps forward.

The WALL2 state asserts both the turn-left and forward outputs, then checks the antennae.

If the right antenna is still touching, it needs to continue turning.

If the left antenna touches, it's run into a corner and needs to reorient itself so the new wall is on its right, the situation we dealt with the rotate-counterclockwise state.

Finally, if both antennae are free, the ant should be in the state of the previous slide: standing parallel to the wall, so we return the WALL1 state.

Our expectation is that the FSM will alternate between the WALL1 and WALL2 states as the ant moves along the wall.

If it reaches an inside corner, it rotates to put the new wall on its right and keeps going.

What happens when it reaches an outside corner?

When the ant is in the WALL1 state, it moves forward and turns right, then checks its right antenna, expecting the find the wall its traveling along.

But if its an outside corner, there's no wall to touch!

The correct strategy in this case is to keep turning right and stepping forward until the right antenna touches the wall that's around the corner.

The CORNER state implements this strategy, transitioning to the WALL2 state when the ant reaches the wall again.

Hey, this might even work!

Let's try it out... Meet the Roboant simulator.

On the left we see a text representation of the transition table for the FSM brain.

Each action line specifies an input pattern, which, if it matches, will set the next state and output signals as specified.

This particular version of Roboant allows the ant to drop or pickup breadcrumbs, and to sense breadcrumbs it comes across - these inputs and outputs aren't needed for this demo.

The input pattern specifies a value for the current state and antenna inputs.

The simulator highlights the row in the table that matches the current inputs.

As you can see, initially the ant is the LOST state with neither antennae touching.

On the right is a map showing our green ant standing in a maze with blue walls.

We can select several different mazes to try.

To see the ant in action, let's click the STEP button several times.

After a few steps, the ant hits the wall, then rotates counterclockwise to free its antenna.

Now it starts following the wall until it reaches a corner, at which point it keeps turning right and stepping until it's once again in contact with the wall.

Now we'll let it run and watch as the FSM patiently pursues the programmed strategy, responding to inputs and generating the appropriate output responses.

With more sensors and actuators, you can see that fairly sophisticated behaviors and responses would be possible.

In essence this is exactly what modern robots do - they too have FSM brains full of pre-programmed behaviors that let them perform their assigned tasks.

Neat!