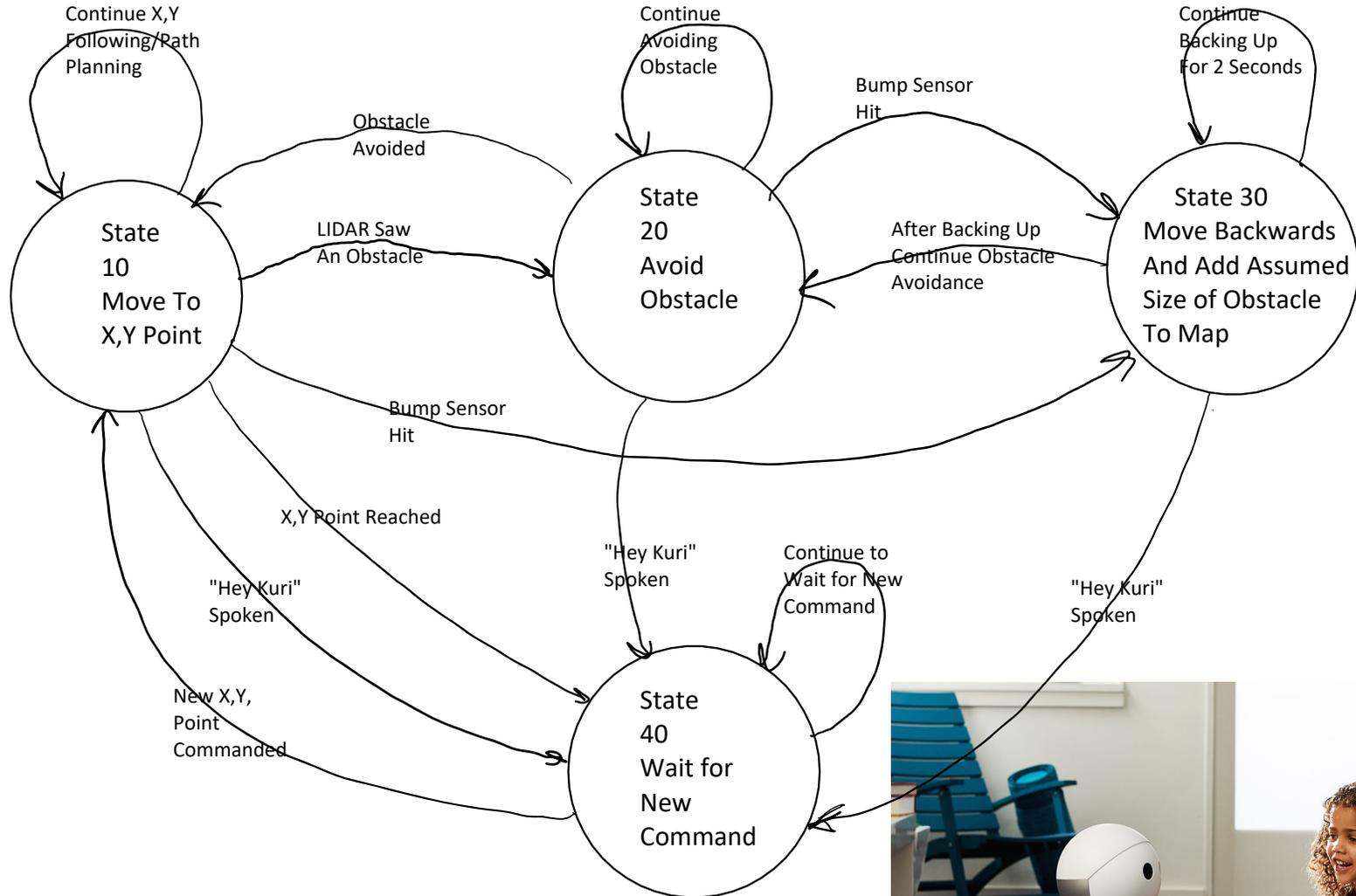


Kuri StateMachine

Sunday, February 10, 2019 2:23 PM

Kuri Robot Told to Go to the Kitchen

Sensors:	LIDAR for distances in one plane
	Bump sensor for low obstacles
	Microphone Listening for New Commands. "Hey Kuri" "Go to Kitchen"



Made Up Example of State Machine for Kuri Robot Commanded to Go to the Kitchen

```
int myStateVar = 10; // Initialize global State Variable to desired initial state
long timeint = 0;
float desiredx = Kitchen_X; //Kitchen x point trained earlier by user
float desiredy = Kitchen_Y; //Kitchen y point trained earlier by user

void myPeriodicFunction(void) { // Most of the state machines we will be writing
    // in this class will be inside a function that
    // is called at a periodic rate. Like say every 1ms.
    //This is were we do all the code that needs to happen every time in the periodic function. Many times that will be reading feedback
    //sensors and possibly calculating control laws. Sometimes control laws will be dependent on the state but still are calculated each
    //sample but not output to the system. This allows for smoother transition of control states. In addition some control laws can only be
    // calculated inside a state for example avoid and obstacle state.
    ReadFeedBackSensors(); //Gyro, Microphones for voice command, LIDAR, etc.
    CalculateControlLawtoMakeKuriGoToXYPostion(desiredx, desiredy);
    ConstantlyCheckForVoiceCommand();
    //increment a global long integer keeping track of a sample count for timing
    timeint++;
    etc.;
    switch(myStateVar) { // state machine
        case 10: // Kuri to x,y waypoints
            // Command Kuri to go to a X,Y coordinate.

            // Case 10's check to see what state to go to next periodic call
            if (LIDAR locates obstacle in path) {
                myStateVar = 20; //Avoid Obstacle
                //initialize any variables needed for state 20. Very important.
            } else if (Bump sensor pressed, Obstacle not seen by LIDAR) {
                myStateVar = 30; //Avoid Obstacle not seen by LIDAR
                //initialize any variables needed for state 30. Very important.
                BackupCount = 0;
            } else if ("Hey Kuri" Spoken to Kuri) {
                myStateVar = 40; // Wait for New Command
                //initialize any variables needed for state 40. Very important.
            } else if (X,Y position reached) {
                myStateVar = 40; // Wait for New Command
                //initialize any variables needed for state 40. Very important.
            } else {
                // continue in X,Y waypoint state, This else is not needed since
                // myStateVar is already 10 but just emphasizing the point.
                myStateVar = 10;
            }
            break;
        case 20: // avoid obstacle in the way.
            // Command Kuri to avoid obstacle with wall following or some other algorithm like the A* path planner

            // Case 20's check to see what state to go to next periodic call
            if (Obstacle Avoided and no longer in path) {
                myStateVar = 10; // go back to X,Y path following
                //initialize any variables needed for state 10. Very important.
            } else if (Bump sensor pressed, Obstacle not seen by LIDAR) {
                myStateVar = 30; //Avoid Obstacle not seen by LIDAR
                //initialize any variables needed for state 30. Very important.
                backupCount = 0;
            } else if ("Hey Kuri" Spoken to Kuri) {
                myStateVar = 40; // Wait for New Command
                //initialize any variables needed for state 40. Very important.
            } else {
                // continue avoiding obstacle, This else is not needed since
                // myStateVar is already 20 but just emphasizing the point.
                myStateVar = 20;
            }
            break;
    }
}
```

```

case 30: // Bump sensor hit so back up slowly for 2 seconds
        // Command Kuri Backup straight back
        backupCount++;

        // Case 30's check to see what state to go to next periodic call
        if (backupCount > 2000) { // Backup for 2 seconds then go back to Obstacle Avoidance

                myStateVar = 20; // Return to obstacle avoidance.
                // Assume a size for the obstacle just bumped into and add obstacle to A* Map
                //initialize any other variables needed for state 20. Very important.

        } else if ("Hey Kuri" Spoken to Kuri) {
                myStateVar = 40; // Wait for New Command
                //initialize any variables needed for state 40. Very important.
        } else {
                // continue backing up. This else is not needed since
                // myStateVar is already 30 but just emphasizing the point.
                myStateVar = 30;
        }
        break;
case 40: // Wait for new X,Y Command
        Stop Kuri Moving
        // For this simple State Machine have Kuri wait here indefinitely

        // Case 40's check to see what state to go to next periodic call
        if (New Command Given) {
                X_desired = Commanded_X;
                Y_desired = Commanded_Y;
                myStateVar = 10;
                //initialize any other variables needed for state 10. Very important.
        } else {
                // continue waiting. This else is not needed since
                // myStateVar is already 40 but just emphasizing the point.
                myStateVar = 40;
        }
        break;
}
}
}

```