RSS-I Spring 2013 Challenge Design Outline - Team 8

Overview

- Motivation
  - Robots are becoming increasingly ubiquitous in the modern world, and will need to be capable of increasingly more sophisticated autonomous behaviour
  - There are some applications, such as a mission to mars, where direct control of a robot's minute actions is infeasible. These situations require autonomous operation on the part of the robot in order to ensure mission success.
- Scenario
  - Our challenge is motivated by the scenario of a robot landing on mars
  - Upon landing, the robot must be able to determine where it is and what materials it has available, even if its prior information about its surroundings or supplies is incomplete
  - The robot must be able to collect supplies that landed with or before it, some of which may have landed off-target or been destroyed completely
  - The robot must be able to construct a shelter to protect future visitors from the elements out of the supplies that survived

Problem Statement

- Construct and program a robot that, given an imperfect map of an area, can collect a number of colored blocks of varying size at imperfectly known locations and use them to construct a primitive shelter.

Assumptions

- The Environment
  - The environment that the robot will need to maneuver through will be a maze made up of walls that are taller than the robot.
  - The ground will be level and made of a single material.
  - The walls of the maze will be a uniform neutral color, except for the location markers on them.
  - The illumination will be roughly constant
  - There will be sufficient illumination such that the blocks and their colors will be easily visible
  - There will be no inclement weather - no high-wind or sandstorm simulations are expected
  - The terrain will be smooth, though with some obstacles
- The Blocks
  - Each block will be one of three colors: red, green, or blue.
  - Each block will be cube or rectangle shaped.
  - There will be two types of blocks: one cube shaped and another made from two cube shaped blocks put together.

– The blocks will be found only on the ground of the environment.
– There will be at least four blocks in the environment.
– The blocks will be stationary (unless manipulated by the robot) after their initial placement

- The Map
  – The map will contain two main components: walls to indicate the edges of the environment and markers to indicate where the blocks should be.
  – The map will be almost completely correct when defining the walls of the maze.
  – The map will be inaccurate, but not completely wrong, when showing the locations of the blocks.

Diagrams

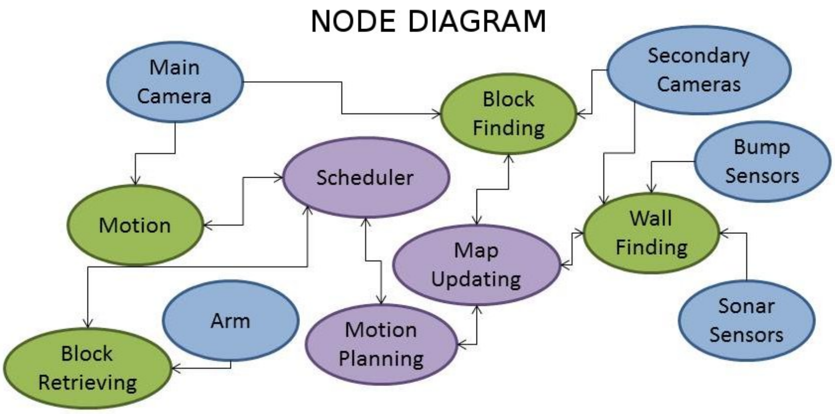

Figure 1: This picture represents the different nodes that our robot will have. The blue nodes correspond to hardware components. The green nodes represent the portions that receive the hardware readings and information. The purple nodes represent the purely internal nodes that are responsible for making decisions about what the robot should do.

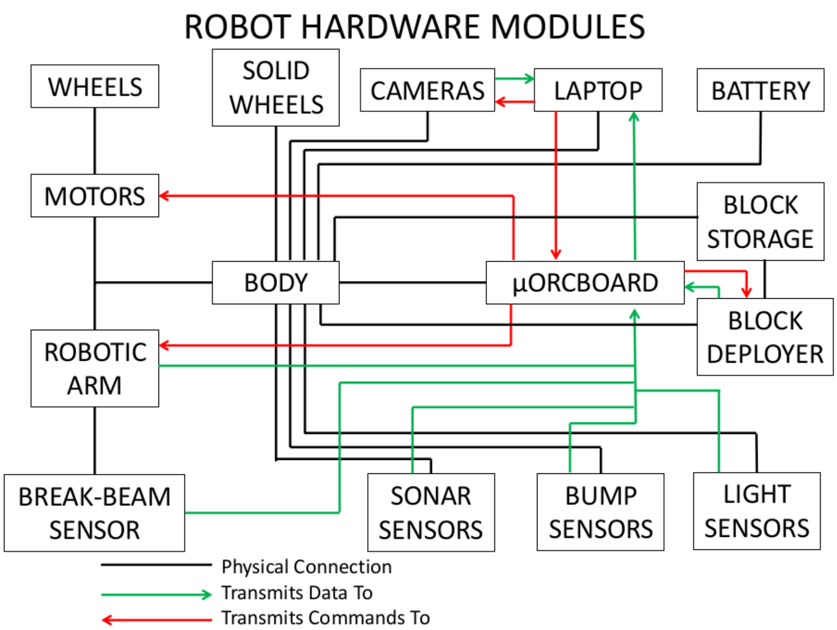

Figure 2: This diagram shows all of the physical components of the robot, together with the physical connections and data transmission routes between them.

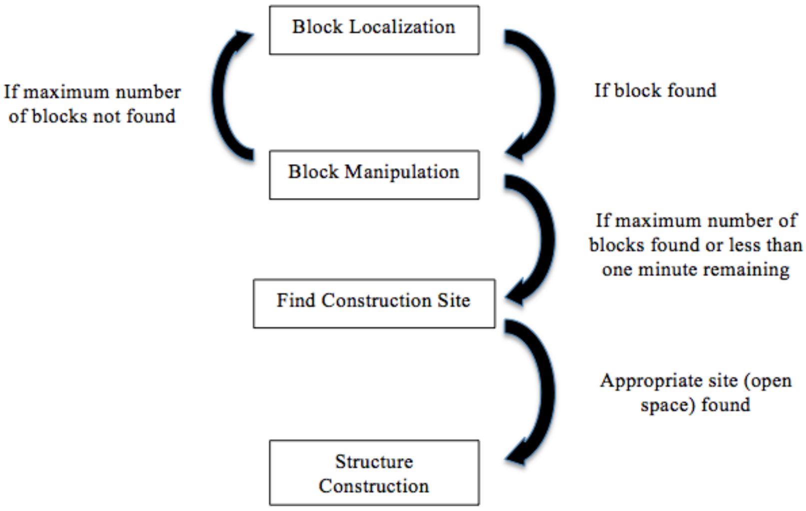

Figure 3: This diagram shows the high-level states and the state transition conditions.

Technical Approach

Note: the ID Construction Site module is not included, as we intend to carry all of the collected blocks with the robot, and construct a structure in any nearby feasible location upon having collected a sufficient number of blocks.

1. Planning/Navigation (written by)

   - Mapping
     - The robot will maintain an internal map including goal points (blocks), the robot's current position, obstacles, and map boundaries. Entries in this map will have a certainty rating based on how well they correspond with recent data taken of that entry.
     - The map will further be used in conjunction with odometry to determine the robot's pose. Known fiducials and objects/obstacles will be correlated with observed phenomena and previous pose estimations in order to produce estimates of the robot's current pose.
   - Motion Planning
     - The robot's internal map will be transformed into a Configuration-space map, which uses methods employed in lab 6 to correctly re-interpret the map in terms of extended obstacles and a point-like robot.
     - A sampling-based algorithm (likely an RRT) will be run in order to construct a roadmap of free-space nodes connected through free paths. A planner will be called to construct a path through currently available free nodes to the nodes nearest to the desired free point(s).
     - The roadmap-construction algorithm will run continuously (whenever the processor can be spared) in order to constantly improve the robot's mapping capabilities.
     - The roadmap-construction algorithm will be equipped with methods for adding, removing, and moving obstacles so that optimal planning may be maintained even as the robot gains better estimates of the layout of its surroundings.

2. Finding Objects (Kirsten Olson)

   - Wall Finding
     - Bump Sensors: Bump sensors will be used initially to find a wall. The internal map will be modified to correspond with what the sensors say.
     - Sonar Sensors: After the initial finding of the wall, sonar sensors will be used to navigate using distance from a known wall to find location.

    - Primary Camera: The main, forward facing camera will be used to recognize the fiducials on the walls. This information will be used to update current position.

- Block Finding
    - Primary Camera: The primary camera will be used to locate blocks and update the robot's knowledge of where the block is as the robot moves closer to the block. This camera will be able to tell where a block is in relation to the robot very accurately.
    - Secondary Cameras: Cameras mounted to the sides and back of the robot provide a 360 degree view at all times. These cameras' only goal is to locate blocks and add markers to the map corresponding to the places that blocks were found.

3. Gather and Store Blocks (outline written by , final section will be written by)

    - Gathering Blocks
        - Upon reaching a block, the robot will extend its arm and attempt to pick up the block
        - The robot may use camera-based methods to correct its arm movement/location online as it tries to pick up the block (our estimation of the feasibility of this, as well as many of the details about picking up blocks, will be greatly refined after lab 7 has been started)
    - Storing Blocks
        - The robot will store all of the blocks in shaped container mounted on the robot.
        - The blocks will be inserted into a funnel, which will ensure that the blocks have the correct pose in the container
        - The container will be shaped such that the blocks are aligned in a configuration ideal for the construction of the final structure

4. Transportation/Construction ( )

    - Block Transportation
        - The arm will deposit blocks into a funnel that guides the blocks into a storage container.
        - The container can fit five single blocks in a vertical one-dimensional tower.
        - The container will have an open base that rests on the floor; the blocks inside will contact the ground.
        - The robot will look for blocks until it has found and obtained five single blocks.
    - Structure Construction
        - The back panel of the storage container will be a thin aluminum sheet actuated by a servo such that it can swing out on a hinge.
        - When the robot has reached a proper construction site (generally the current location of the robot unless it happens to be a non-open space), the back panel will swing open. The robot will drive forward, leaving a vertical tower of five blocks behind.

5. Concentration - 4 Cameras (outline written by , final section will be written by)

    - A camera will be mounted in the center of each side of the robot's square chassis, elevated slightly above the rest of the robot's structure (specific configuration depends on not interfering with the location of the robot arm from lab 7)
    - These cameras will feed video into the robot's logic at a relatively slow framerate, in order to minimize the additional CPU burden
    - These cameras will provide almost 360 degree vision (actual amount is dependent on the specific cameras' FOV)
        - This eliminates the need for the robot to spin in a circle in order to map its environment, which saves time and eliminates possible odometry errors introduced by frequent spinning
        - More importantly, this allows the robot to be updating the objects on its map in all directions as it is completing other tasks. This will result in a much higher amount of viewing time per object to be identified (increasing our ability to analyze the object precisely) and will decrease the possibility of missing objects during motion.

Milestones

Note: Implementation of any item listed here includes construction/coding, testing, debugging, etc.

- First Week
  - Complete all mechanical structures and attach them to the robot (transport and construct module, 4 camera setup)
  - Implement the Find Objects/Vision modules in code
  - By the end of this week, our robot should be able to mechanically store and place blocks, as well as identify walls and objects with its cameras.
- Second Week
  - Implement the Block Gathering module in code
  - Implement the Construct Structure module in code
  - By the end of this week, our robot should be able to gather blocks and construct a structure given a set of blocks in storage, along with being able to identify blocks and walls.
- Third Week
  - Implement Planning and Navigation in Code
  - By the end of this week, our robot should be able to identify blocks and walls, navigate to the blocks (avoiding the walls), gather and store them, and then construct a structure with them. This covers the entire challenge.
- Fourth Week
  - Further testing/debugging
  - Optimizations and improvements
  - This week is left for additional testing and debugging if required, as well as for optimizing and improving already-implemented modules in the robot

Decision-Making Process

- Begin by listing the Pros and Cons of each side of a decision
- Debate the problem over these Pros and Cons, attempting to reach agreement
- If agreement cannot be reached in an appropriate amount of time, a vote will be taken and consensus will be reached