ESE 501 Digital System Specification and Modeling
Department of Electrical and Computer Engineering
Stony Brook University

Robot Navigation in Confined Areas

Figure 1 illustrates a large and confined environment where multiple robots navigate in the presence of human traffic. Both static and dynamic obstacles may be placed. The robots are controlled by the navigation system which handles all services concurrently. Typically, the environment has the cameras for surveillance operations. The navigation system may interact with the surveillance system for more accurate decision making for the robot navigation. (In the project, we will not consider the surveillance system)

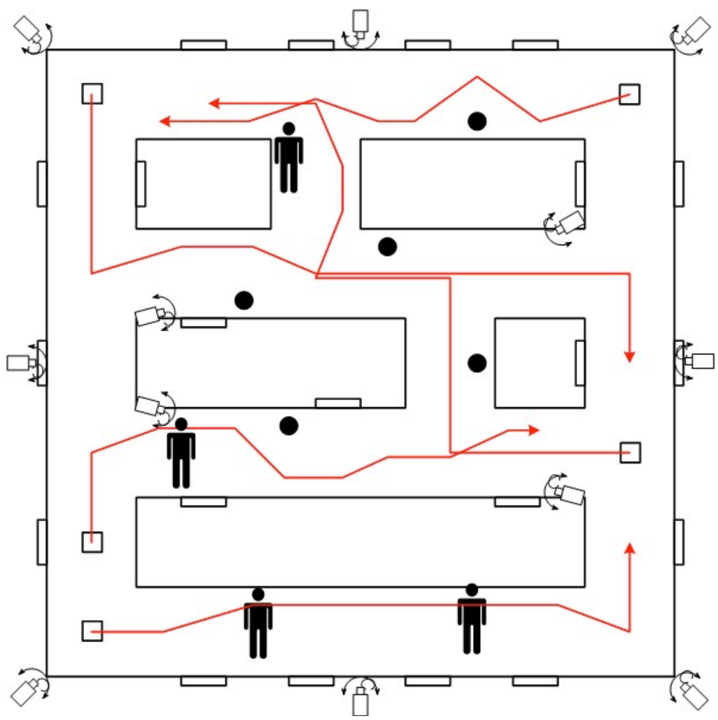

Figure 1. Application Model

This application model can be used in different types of service environment such as providing wheelchair services and delivering tools within manufacturing facilities.

The map is represented with grids as illustrated in Figure 2. The path of the robot (i.e., source to destination) is represented as a sequence of grids where the center positions (as well as the dimension) of the grids are annotated in the map. The navigating server can determine the location of the robot by its current grid index. The navigation server ensures that two robots never stay in the same grid at any given time. The robots are capable of localizing itself within the confined area and detect any unexpected obstacles with their own proximity sensors.

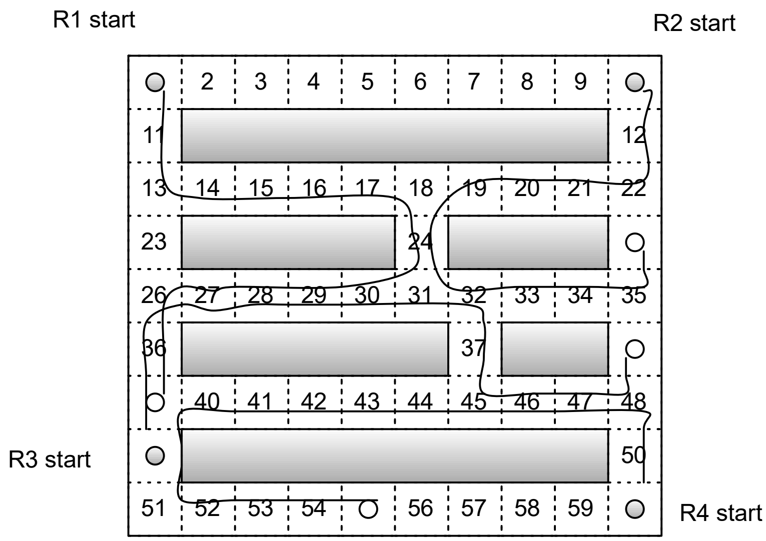

Figure 2. Grid Based Navigation

The server maintains the path information of all robots in the internal data structure as illustrated in Figure 3. Each robot may have different grid types. Additionally, the list of segments for robots to navigate through is also maintained within the server.

Robot Index	Current Grid	Next Grid	Grid List	Segment List	Node List

Figure 3. Server Data Structure

The robots navigate by following the grid sequence provided by the server. Figure 4 illustrates basic functional mechanism of the robots communicating with server for synchronization and control. Each time the robot is about to leave the current grid, the robot sends the crossing notification to the server, and continues. Upon receiving the notification from the robot, the server responds with acknowledgement to the robot (STOP or CONTINUE). If the server's acknowledgement is not received before leaving the current grid, the robot stops. The reason for the stop is to avoid potential collision by violating multiple robots in the same grid condition.

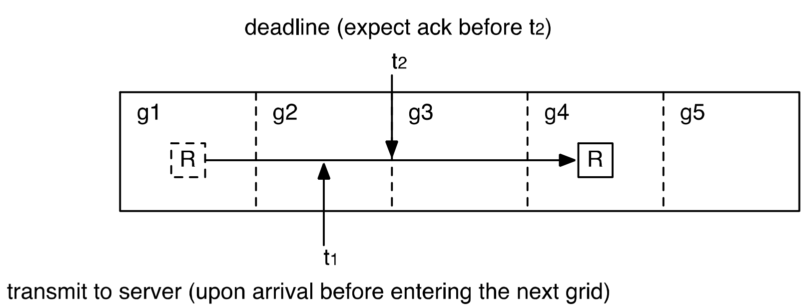

Figure 4. Communication Synchronization

The server maintains the robot status information as in Figure 5. The status for each robot includes the current speed, the navigation status whether the robot is currently in idle, stop, or moving. The current location of the robot is represented by the segment and grid indices. The range of the proximity sensor and the server status are also maintained. The server status indicates the outstanding responses that the server must perform. The speed, sensor range, grid type, and initial location are initialized when the path is assigned. However, these parameters may change by the server depending on the navigation conditions.

Robot Index	Navigation Status	Current Grid	Next Grid	Next Node	Speed	Sensor Range	Grid Type	Server Status
0	STOP	23	2	1	0	NEAR	C	Stop due to obstacle
1	IDLE	45	N/A	2	0	NEAR	C	No Service
2	MOVING	4	2	3	1	FAR	F	Stop at the Boundary

Figure 5. Robot Status Data Structure

When the paths are generated for the robots, the node-ordering data structure is also updated. The node-ordering data structure indicates the incoming ordering of all robots for each node as illustrated in Figure 6. Outgoing robots are not considered. The data structure for the node-ordering is illustrated in Figure 7. The entries for each node indicate the robot indices, the earliest arrival time, the latest arrival time, and the expected arrival time. The earliest arrival time and the latest arrival time are determined during the path generation suggesting that the robot should arrive at the node after the earliest arrival time but before the latest arrival time. The expected arrival time is estimated by the server and should be between the two timing parameters.

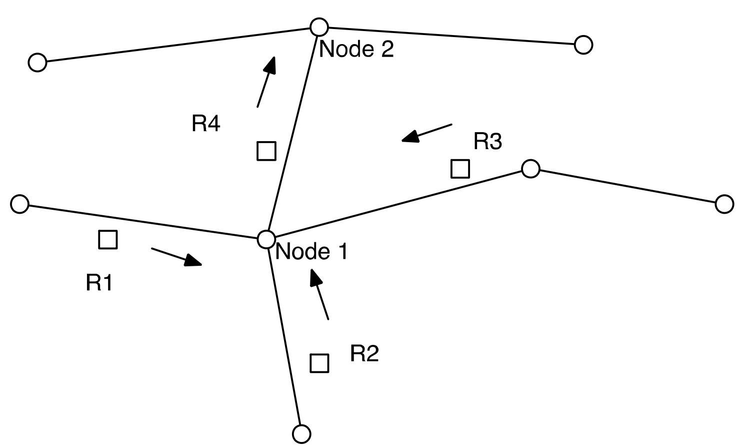

Figure 6. Node Merging

Node 1	Node 2	...	Node N
R_1:(T_e, T_l, T_a)	R_6:(T_e, T_l, T_a)	...	R_7:(T_e, T_l, T_a)
R_2:(T_e, T_l, T_a)		...	R_9:(T_e, T_l, T_a)
R_4:(T_e, T_l, T_a)		...	
...	...	...	...

T_e: Earliest Arrival Time

T_l: Latest Arrival Time

T_a: Expected Arrival Time

Figure 7. Node Ordering Data Structure

Based on the timing parameters specified in the node-ordering data structure, the speeds of the robots are computed according possible speed control scenarios are shown in Figure 8.

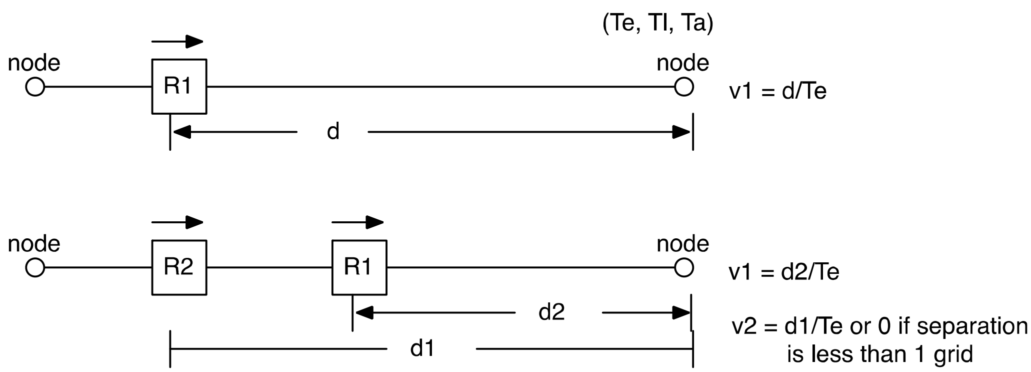

Figure 8. Speed Control Scenarios