International Journal of Robotics and Automation (IJRA)
Vol. 6, No. 2, June 2017, pp. 80~98
ISSN: 2089-4856, DOI: 10.11591/ijra.v6i2.pp80-98

Networked Service Robots Control and Synchronization with Surveillance System Assistance

Doug Kim
Faculty of Electrical and Computer Engineering Technology, Farmingdale State College, USA

Article Info

Article history:
Received Jan 20, 2017
Revised Feb 16, 2017
Accepted Mar 2, 2017

Keyword:
Adaptive control
Synchronization
Mobile robots
Navigation control
Surveillance

ABSTRACT

This paper proposes an efficient navigation control and synchronization mechanism of multiple networked robots for operation in large confined areas. An adaptable grid based navigation and control strategy is adopted to eliminate potential colli-sions among robots. Unexpected obstacles are handled and the speed of individual robot is maintained using the node-ordering technique. The proposed navigation control and synchronization mechanism is scalable and can be easily extended for multi-cell large environment. The obstacles information is gathered through local information by the robots for better planning of the navigation. The system collaborates with the existing surveillance systems in case of additional visual information is necessary. The interaction with the surveillance system is mini- mized to reduce potential overhead. The proposed methodology is evaluated for a large-scale simulation with multiple robots.

Corresponding Author:
Doug Kim,
Electrical and Computer Engineering Technology,
Farmingdale State College,
2350 Broadhollow Rd. Farmingdale, NY, USA.
Email: kimdc@farmingdale.edu

1. INTRODUCTION

There have been numerous studies on the multi-robot navigation and control systems for decades for various health service applications as well as for the warehouse applications [1][2][3][4]. Navigation and control of multiple robots in a large confined area is particularly interesting since the robots can be used to support various human activities. In these applications, the robots navigate within the area for delivering the objects to the destinations. Within the area, these robots use multiple sensors to navigate through the complex environment [5][6][7].

One of the key objectives for the navigation system is to control the multiple robots simultaneously without interfering with the human traffics [8]. Moreover, the robots must be able to handle static obstacles as well as unexpected dynamic obstacles [9]. As the number of robots in operation increases, the navigation system must be designed to minimize performance degradation, thus requiring the system to be scalable. When navigation and control of multiple robots are considered, one of two typical control strategies can be used, namely distributed or centralized control schemes. An advantage of distributed navigation control is that the computational load of the centralized server is minimized by letting the robot making autonomous navigational decisions [10][11]. However, in the large confined area, the distributed control strategy suffers from reduced robot utilization when unexpected obstacles appear in the navigation path. On the other hand, the centralized control strategy may not be effective in a small area where unexpected obstacles are rare. The centralized control strategy is more effective in a large area with obstacles since the centralized control mechanism can make proper adjustments to all robots affected by the changes in the navigation environment [12].

Moreover, many of indoor infrastructures have the visual surveillance networks that can assist the navigation control system by providing the visual information [13]. The visual surveillance network can provide information about the navigational environment to the robot through the centralized mechanism in case of navigation uncertainty (i.e., failed localization, etc) [14]. The navigation system must be able to interact with existing surveillance systems with minimum impact on normal surveillance operation. Utilizing the existing surveillance system becomes critical in terms of reducing the level of uncertainty in navigation environment for the robots.

In this paper, an efficient scalable navigation control and synchronization mechanism of multiple networked robots for operation in large confined areas is proposed. An adaptable grid based navigation and control strategy is adopted to eliminate potential collisions among robots. The node-ordering technique is proposed and incorporated into the overall mechanism where the speed of individual robot is adjusted to avoid the navigation deadlocks as well as handling of the unexpected obstacles in the navigation paths. During the navigation, the information about obstacles in the navigation paths is gathered locally by the robots for improving the overall path planning and navigation. The proposed system collaborates with the existing surveillance systems in case additional visual information is necessary. The interaction with the surveillance system is minimized so that the overhead caused by collaborating with the navigation system would not hinder the normal surveillance operation.

The remainder of this paper has 5 sections. In Section 2, we present the overview of the application model and the proposed approach including the problem descriptions. Section 3 presents the basic navigation principles including centralized synchronization and control methodology for the large-scale environment. Section 4. presents the extension of the proposed method which incorporates static and dynamic obstacle handling in the navigation environment. In Section 5, we evaluate the proposed path planning methodology with a large-scale scenario. Finally, our contribution is summarized in Section 6. along with future work.

2. MOTIVATION AND APPROACH

2.1. Application Model and Motivation

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.

The navigation system must meet the power and timing requirements. This application model can be used in different types of service environment such as providing wheelchair services and delivering tools within the manufacturing facilities. A larger environment can be covered by introducing multiple wireless network clusters across the entire service area for improved communication reliability.

2.2. Problem Description and Issues

One of the key objectives for the navigation system is to control the multiple robots simultaneously without interfering with human traffics. Moreover, the robots must be able to handle static obstacles as well as unexpected dynamic obstacles. As the number of operating robots increases, the navigation system must be able to handle congestion caused by robots themselves. The navigation system must be able to interact with existing surveillance systems with minimal impact on the normal surveillance operation. The visual surveillance system can assist the navigation system in making more accurate navigation decisions. Overview of the interaction between the navigation and surveillance system is illustrated in Figure 2. In order to handle the dynamic changes in the navigation environment, the navigation system closely interacts with the path planning system where the navigation paths for the robots are created. While the path planning system is an integral part of the overall system, we limit our scope to the navigation control system and interfacing with the path planning system.

2.3. Approach and Assumption

In this paper, we propose a grid-based robot navigation. The grid tt is defined as a rectangular region. The entire environment is divided into grids as shown in Figure 3. The server maintains the location

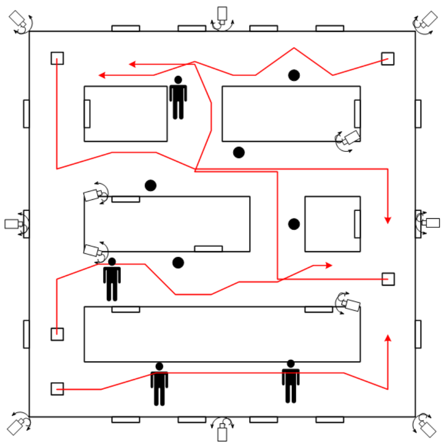


Figure 1. Illustration of multiple robots navigating in the presence of thuman traffics. Visual assistance to the navigation system is supported by the existing surveillance system

Information of all grids with unique indices. The segment information from the path planning for each robot includes robot index, sequence of grid indices and navigation deadline. The segment usually indicates confined spaces (e.g. corridor) in the environment. The number of grids in the segment of each robot may differ. Given grids for each robot to navigate through, the system directs each robot to the grids assigned to them while exchanging necessary information at each grid. The communications overhead is minimized while maximizing the navigation flexibility, the proposed navigation system incorporates adaptive grid resolution mechanism. The adaptive grid resolution provides a capability of fine navigation when the robots need to navigate through obstacles rather than taking different paths or wait until the obstacles are cleared.

Multi-cell relay structure is used in the proposed navigation system to be scalable to cover the large areas. While there are distributed and centralized control mechanisms, we use the centralized communication model where the system controls individual robots for navigation. The centralized mechanism has a lot of advantages over the distributed mechanism especially in confined areas for overseeing the entire navigation environment which allows more effective resource utilization. To handle a large number of concurrent navigation of robots without potential collisions and deadlocks, the speed control management mechanism using the node-ordering is proposed where the node-ordering guarantees the effective flow of robot navigation.

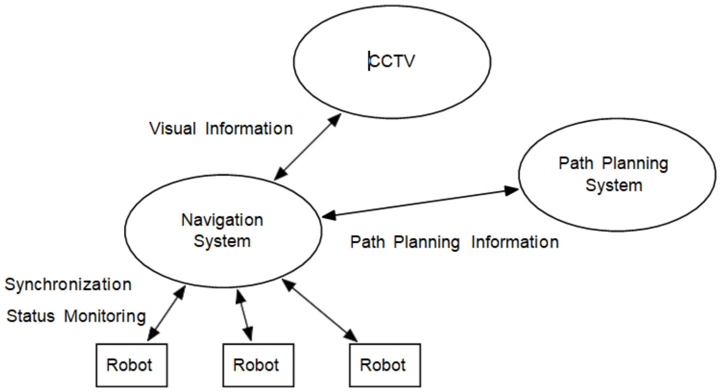


Figure 2. Illustration of functional interactions. The navigation system communicates with multiple robots and existing surveillance system for visual assistance. The path generation of the robots are supported by the path planning system

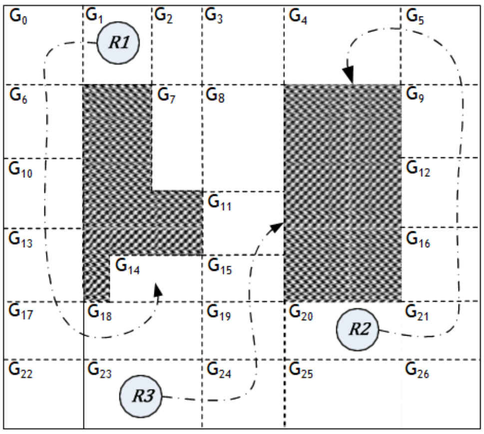


Figure 3. Illustration of grid-based map representation and navigation. A path of the robot is represented by a sequence of grids

To acquire more accurate visual information for the navigation system, the navigation system collab- orates with the existing surveillance system. While minimizing the interaction with the surveillance system, the navigation system fully utilizes the sensor information from the robots for constructing the obstacles map dynamically which is used for controlling the flow of the robots. With the minimum amount of interaction, the proposed system is designed to monitor the location information of obstacles in addition to the localized information gathered by the robots in the area. This periodic monitoring enables the navigation system to obtain the changes in navigation capacity of the corridors. The combined information is used for the adaptive navigation by providing the information to the path planning for rerouting of the robots.

3. RESULTS NAVIGATION AND SYNCHRONIZATION STRATEGY

3.1. Grid Based Concurrent Navigation

Figure 4 illustrates the overview of the multiple robot concurrent navigation platform. The navigation server controls the robots through a communication link. The navigation server also communicates with the existing surveillance system for visual assistance. The navigation server, the robots, and the surveillance system share the common map for the synchronized coordination. The map is represented with grids as illustrated in Figure 5. The size of the grid is adjustable where the coarse grid is typically used for the normal navigation and the fine grid is used for navigating in finer resolution. Each robot may be assigned with different grid size depending on the navigation environment of the area.

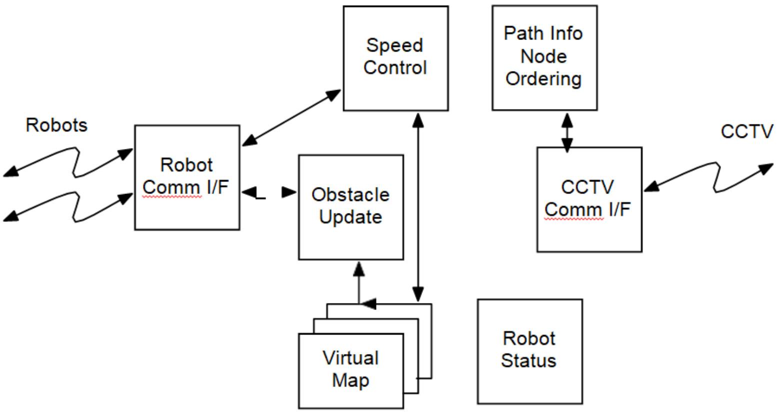


Figure 4. Overall operational Workflow. The server communicates with the multiple robots simultaneously. The server also communicates with the existing visual sensor network for obtaining visual data. The server navigates and synchronizes the robot using the common map and obstacle information

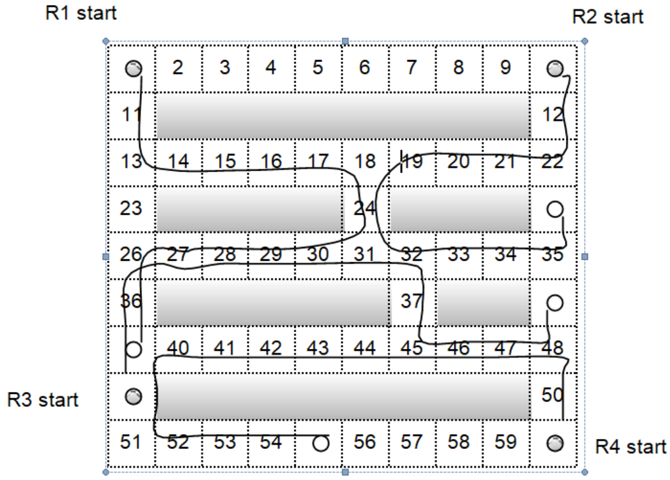


Figure 5. Illustration of the grid based map and its graphical representation. Each corridor is represented with an edge while each intersection is represented with a node. Multiple grid resolutions are possible. Each segment has its own capacity

The path of the robot (i.e., source to destination) is represented as a sequence of grids where the center positions 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 server maintains the path information of all robots in the internal data structure as illustrated in Table 1. Each robot may have different grid types. Additionally the list of segments for robots to navigate through is also maintained within the server.

Table 1. The Path Information Data Structure Maintained by the Server. The Path Information of the Robots are Generated by the Path Planning System

Robot Index	Grid Type	Grid List	Segment List	Node List
...	...	...	...	
...	...	...	...	
...	...	...	...	

The robots are capable of localizing itself within the confined area and detect any unexpected obstacles with their own proximity sensors. In cases when the robot fails to localize itself, the robot requests visual localization assistance from the surveillance systems through the server.

3.2. Synchronization and Communication Protocol

The robots navigate by following the grid sequence provided by the server. Figure 6 illustrates basic functional mechanism of the robots communicating with server for synchronization and control. Each timethe robot arrives at a grid, the robot sends the arrival notification to the server, and continues to the next grid in the sequence. Upon receiving the arrival notification, the server responds with acknowledgement to the robot. 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. This interaction is illustrated in Figure 7. As long as $d/v_{max} > T\ c$, the continuous navigation is possible where d represents the distance to the next grid, v_{max} is the maximum speed of the robot, and T_c represents the maximum server response time.

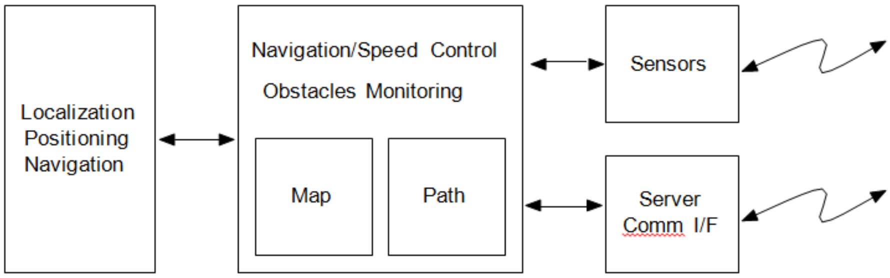


Figure 6. Functional diagram of the robot operation. The operation consists position computation, naviga-tion and speed control, and interface with the server

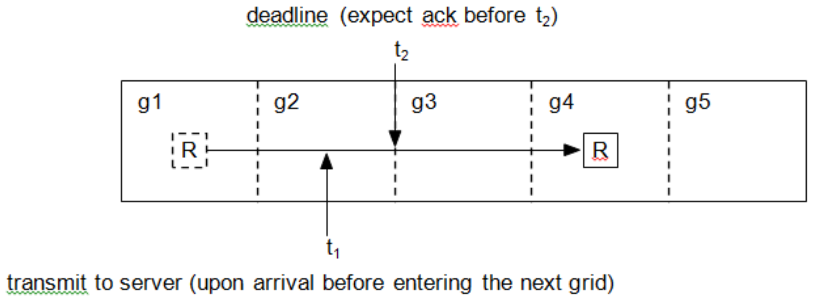


Figure 7. Illustration of communications between the robot and the server. The robots sends arrival message to the server at t_1 and expects the server acknowledgement by t_2

Hence, the robots make communication with the server at every grid. As shown in Figure 8, the frequency of the robot communication depends on the size of grids. If the grid size is small, the communication traffic increases. Moreover, the communication traffic increases as the number of the robots increases. During the normal operation, the system prefers to use as large grid size as possible. Because the server receives all the status information from the robots, the server can always maintain the locations of all robots. The server ensures that no two or more robots navigate to the same grid at the same time to avoid any potential collision by controlling the speeds of the robots as we will discuss in Section 4.

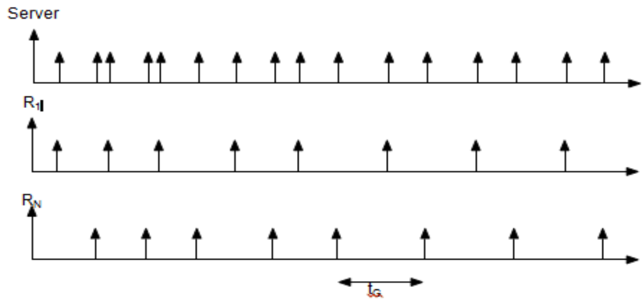


Figure 8. The server and robots communication frequencies. The frequency of the communication depends on the size of the grids and the number of robots.

The server maintains the robot status information as in Table 2. 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.

Algorithm 1 illustrates the robot navigation algorithm. In addition to the process of the basic grid based navigation, the process of handling the obstacles as well as the interaction with the server are described. These processes are discussed in the next section where we consider dynamic unexpected obstacles as well as interferences from the other robots.

3.3. Extension to Multi-Cell Environments

In order to support the large-scale environment, multiple communication relays need to be deployed so that the server and the robots can communicate reliably throughout the large area that may not be covered by a single radio link. Each relay link is represented by a cell, which is a typical way of dividing a large coverage area in a wireless communication network.

Algorithm 1 Robot Navigation Algorithm

```
Robot is initially in idle;
Robot start to navigate when the server initiate the process;
Robot Navigation Process;
for (each time interval) do
        determine the current position;
        if (unknown position) then
            stop;
            notify the server by calling notify position();
        end if
        if (right before the next grid) then
            if (previous grid status not received) then
                stop;
            else
                notify the server by calling notify grid();
                continue;
            end if
        end if
end for
Obstacle Event Process;
wait for sensor event;
if (obstacle detected) then
        stop;
        notify the server by calling notify obstacle();
end if
Server Event Process;
wait for event from server;
if (robot initialization) then
        initialize the robot (path, settings) and stop;
end if
if (modify setting) then
        modify the setting (speed, sensor, path) and continue;
end if
if (resume) then
        resume with the given path;
end if
if (stop) then
        stop and wait for next command;
end if
Notification To Server Process;
notify grid();
notify obstacle();
notify position();
```

Table 2. Illustration of the Data Structure Maintaining the Status of All Robots in the System. This Data Structure is Maintained and Updated by the Server.

Robot Index	Speed	Navigation Status	Current Segment	Grid Type	Current Grid	Next Node	Sensor Range	Server Status
	0	STOP	2	C	23	2	FAR	Stop Due to Position Error
	0	STOP	3	C	45	3	NEAR	Stop Due to Obstacles
	0	STOP	2	C	35	6	FAR	Stop at the Grid Due to No Ack
	0	IDLE	1	C	12	12	FAR	OK
	25	MOVING	4	F	56	8	NEAR	OK
...	...	...	...	...	...	...	...	

We assume that multiple cells in the environment have distinctive communication channels with robots. Each robot is assumed to have the capability to communicate through all channels. Since the communication coverage is affected by the environment factors, it is not certain to identify the exact locations where the handoff from a cell to another cell should occur before losing the communication link. Therefore, the system requires an effective handoff protocol, which can switch from a cell to another under the coverage uncertainty.

During the operation, each relay maintains the list of connected robots as illustrated in Figure 9. This list of robots is continuously updated by the relays and sent to the server. This enables the server to always select the appropriate relay to communicate with the robots. If multiple simultaneous communication channels are available by the robots such that the robots are visible to the multiple relays, the server sends the message to all of the relays that can reach the robot. If the number of robots is large and/or the grid size is small, the communication traffic may be too heavy to be handled by a relay. Thus, the load balancing among the relays should also be possible with the configuration.

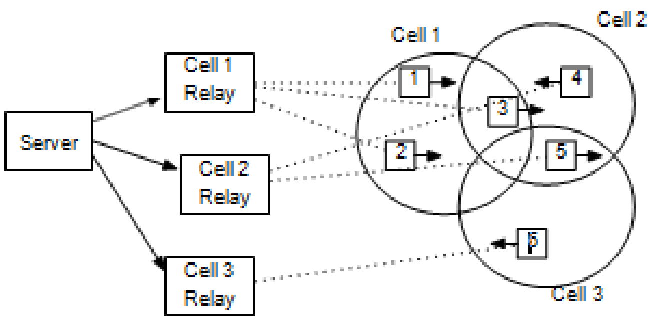


Figure 9. Illustration of transmitting synchronization message to a robot through multiple relays

3.4. Role of Surveillance Network

Typical robot navigation systems do not have the capability of acquiring visual information. The proposed navigation system interfaces with the existing surveillance network for visual assistance. The proposed system is designed to periodically monitor the location information of obstacles in addition to the information gathered by the robots in the area. This periodic monitoring assists the navigation system to identify any unexpected change in the capacity of the corridors.

However, the surveillance system usually has its own operation and may not have computational resources to support the navigation system. In order to minimize the overhead to the surveillance systems, two service parameters are defined. Tservice is the minimum time between the services and Nservice is the number of requests that the surveillance systems may accommodate for the server. Each service is segment based. For example, if two robots request the localization from the same corridor, the server makes a single request to the surveillance system. All information about the robot locations and obstacles are provided per service basis. These two parameters are determined by the navigation system based upon the number of robots and traffic load or dynamically by the surveillance system based upon the current operational load.

Figure 10 illustrates the request patterns by three robots. R1 and R3 are in the segment 1 and R2 is in segment 2. The server receives all the requests but service by the surveillance system depends on the parameters. First example shows that not much of service delay since $N_{service}$ and $T_{service}$ are relatively large. But in the second example show that significant service delays incurred due to low service parameters.

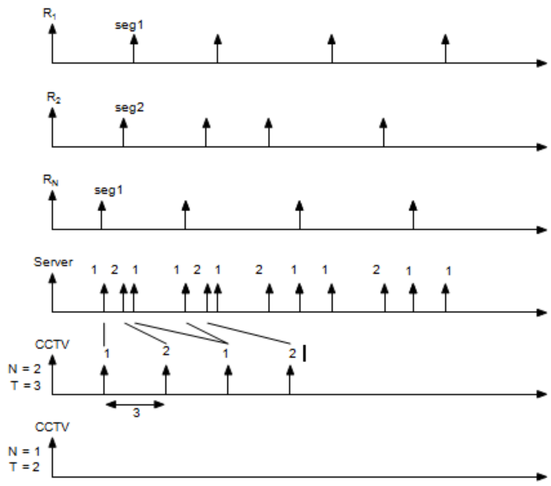


Figure 10. Illustration of service timing by the surveillance system

4. ENVIRONMENT ADAPTATION

4.1. Dynamic Obstacle Management

From the navigating robot's perspective, there are three different types of obstacles. The first type is other robots that are navigating in the same corridor. This type of obstacles is not critical to the system operation since the navigation server has the information about the navigation directions and the positions of all robots. Hence, when there are multiple robots navigating within the same corridor, the sensor ranges are adjusted accordingly to eliminate unnecessary stoppage by the robots. The second type is when the obstacles the moving objects such as humans. When the robot encounters the moving obstacles, prior to sending the message to the server, the robot monitors the changes in distance from itself to the obstacle. If the distance is changing, the robot considers it as a moving obstacle and waits until the obstacle is cleared from the sensor range. Once it is cleared, the robot resumes the navigation. The third type is the static obstacle that is semi-permanently placed within the corridor. When the robot is faced with the static obstacles, the robot needs additional path information to reroute and bypass the obstacles. Illustration of the data structure with obstacle information as shown in Table 3. The obstacle information is main-tained for each grid. The grid flag is set whenever the robot is stopped due to an obstacle(s). When the next robot goes through this grid without stopping, the flag is cleared.

Table 3. Illustration of the data structure with obstacle information

Grid Index	Grid Type	Obstacle Status	Obstacle Time	Segment Index
1	C	NO	---	1
2	C	YES	2:12	2
3	F	No	---	3
...	...	...	...	...

The static obstacles can disturb navigation of other robots in later times. In order to minimize unnecessary stopping time, the obstacle map is generated. When the robot is stopped due to an obstacle, the server annotates the position of the obstacle in the grid. The obstacle map data structure is shown in the Table 3. The navigation server uses the obstacle map information to determine the grid type and the sensor range of the robots entering the corridor with the obstacle to avoid stoppage. Moreover, the path planning system uses the obstacle map information for more efficient initial path design. When the obstacles are cleared as the next robot does not detect the obstacle, the server removes the obstacle from the obstacle map. Thus, the navigating robots provide the necessary information to constantly keep the obstacle map up to date. When the navigating server receives the visual assistance from the surveillance system, the obstacle map can also be generated by detecting obstacles visually.

When the static obstacles are detected after the initial path planning, the capacity of the segments becomes less than predicted. This may potentially generate the navigation deadlock causing the path planning system to replan the affected paths. Hence, the navigation system constantly updates the path planning system with obstacle information. Since the focus of this paper is on the navigation, the path planning system is not discussed. We simply assume that the path planning system is capable of providing the navigation system with new paths when necessary such as in this situation.

4.2. Virtual Grid Adaptation for Fine Grain Navigation

Grid adaptation during the navigation is very critical for maximizing the robot utilization. The grid changes to the finer grid whenever the obstacles are detected or there are other robots navigating in the same corridor. In the normal navigation, the largest possible grid size is used since the usage of smaller grid sizes increases the communication overhead. When obstacles are detected, the size of grid must be changed to provide fine navigation around the obstacles if possible rather than requesting a new path. The grid adaptation is illustrated in Figure 12. The grid size can be changed either by the request of the robots or by the server after learning through other robots reporting the obstacles. The illustration of robot has a front distance sensor to detect any obstacle as shown in Figure 11.

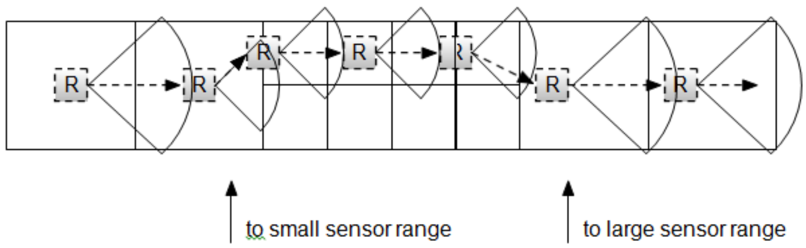


Figure 11. The robot has a front distance sensor to detect any obstacle.

The range of the sensor depends on the navigational condition. In normal coarse grid based navigation, the range is set to a large value. When fine navigation is necessary (i.e., avoiding obstacles), the range is reduced for a quicker reaction. The navigation server sends the range setting to the robots

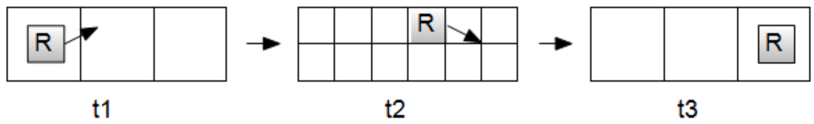


Figure 12. Multiple grid configurations are utilized in the system. A finer grid type is used when fine navigation is necessary to navigate through obstacles.

A realistic situation is shown in Figure 13 where the finer grids are used at the intersection of the map. Depending on the presence of obstacles, the size of grid changes for the finer navigation. Three robots are used for the illustration where each robot has a specific path from the initial location to the destination. The timing relationship is illustrated in Figure 14. The sensor is used to detect the obstacles by the robot. The range of the sensor is set by the navigation server based on the size of grid being employed. This is illustrated in Figure 11. When the obstacles are detected, the robot stops and sends message to the server and waits for next command. If the obstacles are cleared before the response from the navigation server arrives the robot sends message to the navigation server indicating the cleared path. If the obstacles are not cleared for certain amount of time, the navigation server may regenerate the path with the finer grid sizes. Figure 15 illustrates navigation paths of three robots without any obstacles. In Figure 16, an obstacle caused some delays for the robots performing service 1 and service 2.

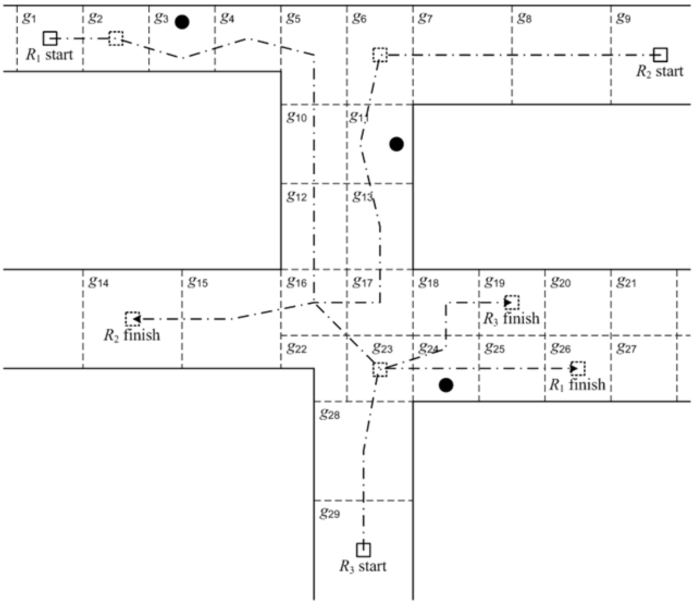


Figure 13. Illustration of the virtual grid adaptation for the finer navigation to avoid the obstacles.

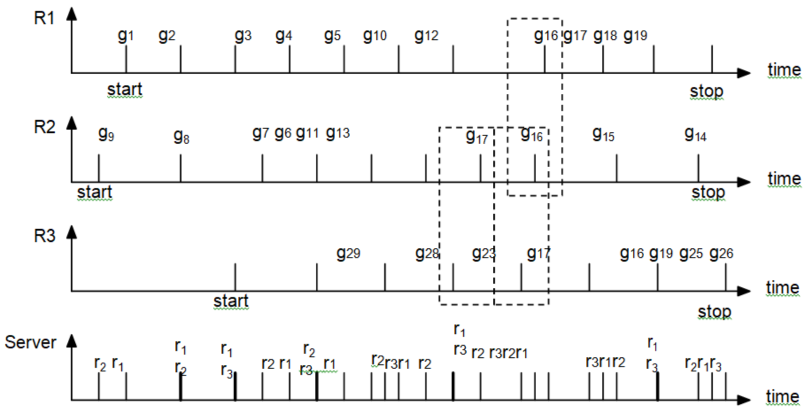


Figure 14. Illustration of the timing relationships for message exchanges between the server and the robots.

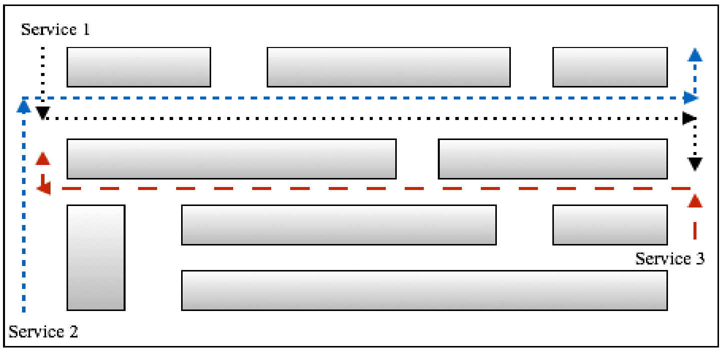


Figure 15. Illustration of initial path planning for 3 service requests

Algorithm 2 Navigation Server Algorithm

```
Wait for (robot events) or (Periodic Timer);

Robot Event Process;
Execute for all events in the queue;
for (each robot event in the queue) do
    if (grid passing notification) then
        check for possible collision
        if clear then call grid ack robot();
        call obstacle map() to clear the obstacle if set;
        otherwise call stop robot();
        update robot status table;
    end if
    if (stop due to obstacle) then
        check for possible collision
        if clear then call grid ack robot();
        obstacle map() to set the obstacle on the map;
        call otherwise call stop robot();
        update robot status table;
    end if
end for

Periodical Robot Speed Adjustment;
for (each time period) do
call speed_control();
end for

Periodical Check Stopped Robot;
for (each time period) do
    Check the status of all stopped robot;
    Notify the robot with appropriate command;
end for

Server Update Functions
update status();
update resource();
update node ordering();
update obstacle map();
speed control();

Robot Configuration Functions
initialize robot();
revise robot();
resume robot();
stop robot();
grid ack robot();
```

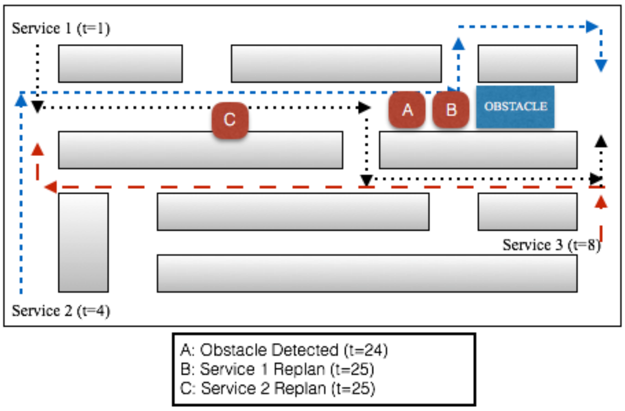


Figure 16. Illustration of path replanning due to an obstacle

If there was not a mechanism with which navigation server requests replanning of the navigation paths, the system would not have been able to complete the services in time as waiting for the obstacle to disappear would withstand uncertain amount of delay. Figure 17 illustrates the effects of replanning requests and response from the path planning with robots' navigation speeds.

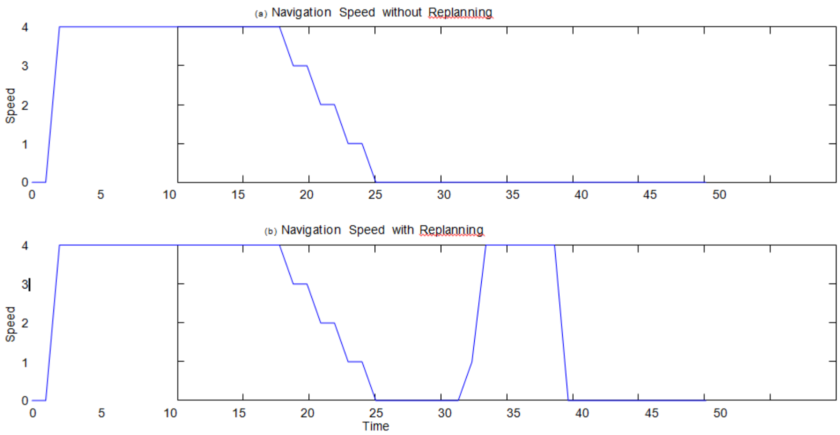


Figure 17. Illustration of effects of path replanning due to an obstacle

4.3. Speed Control with Node-Ordering

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 18. Outgoing robots are not considered. The data structure for the node-ordering is illustrated in Table 4. 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.

Based on the timing parameters specified in the node-ordering data structure, the speed of the robots are computed according to Algorithm 3. If any of the timing is violated, paths must be rescheduled. Possible speed control scenarios are shown in Figure 19. For the single robot case, the speed is computed withthe Te and distance d. In the second case, the speed of the leading robot is computed as the single robotcase. The following robot has two possibilities. If the distance from the following robot is less than one grid distance, the speed is set to zero. Otherwise, the same method for computing the speed is applied. Hence, the speed of the robots in case there are multiple robots, the speed for the earliest arrival robot is computed first. To maintain 1 grid distance, the speed control must be processed often.

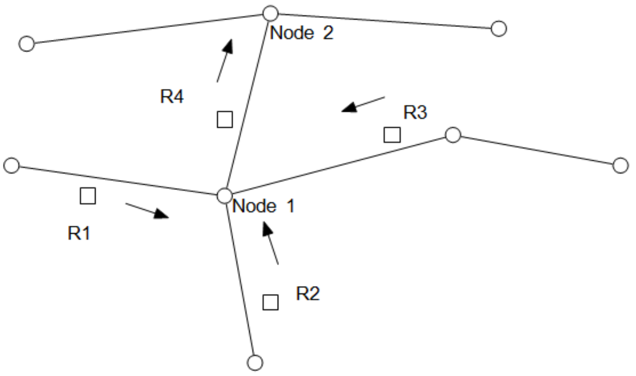


Figure 18. Illustration of situations when multiple robot goes through the same node.

The ordering is maintained by the node-ordering data structure. The robot goes through the node as specified in the node- ordering data structure. In order to maintain the ordering, speed control of individual robot is necessary.

Table 4. Illustration of the data structure maintaining the arrival ordering of the robots for each node in the graph representing the map.

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)		…	
…	…	…	…

Te: Earliest Arrival Time
TI: Latest Arrival Time
Ta: Expected Arrival Time

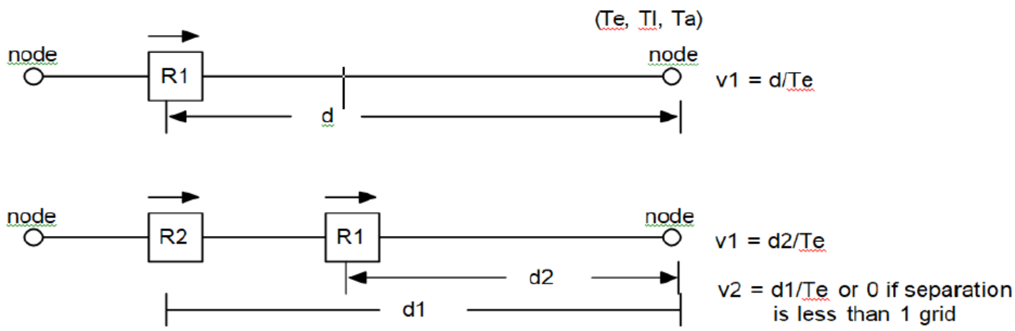


Figure 19. Illustration of the scenarios for speed control computation

Algorithm 3 Speed Control Algorithm

This algorithm is periodically called by the Navigation Server;
Node Table is used. Used Parameters: T_e, T_l, T_a
T_e: Should arrive after this time, T_l: Should arrive before this time; T_a: Estimated arrival time

for (each node in the Node Table) **do**
    for (each robot in the node in order) **do**
        Estimate arrival time T_{temp} of this robot;
        if ($T_{temp} < T_e$) **then**
            decrease the speed so that this robot can arrive at T_e;
            Set $Ta = Ttemp$;
        else if ($T_{temp} > T_l$) **then**

```
            increase to maximum possible speed so that the robot can arrive between Te and Tl;
            If condition cannot be satisfied with the maximum speed, path planning is necessary;
             Set Ta = Ttemp;
        end if
        Make sure that arrival of this robot is later than the robot in front;
        If violates the condition, adjust the speed;
        Update the speed in the robot status table and notify the robot;
    end for
end for
```

4.4. Overall Interaction and Message Formats

In summary, the robots send messages to the server containing the flag information. First flag indicates that the robot is crossing the next grid. The second flag indicates that the position is undefined and request for assistance. With this flag, the server will try to provide the accurate location of the robot. Third flag indicates that the robot is stopped due to the obstacles detected by the sensor. The robot will expect the new set of path to navigate away from these obstacles. The last flag indicates that the robot stopped because the response from the server upon the first flag is not received. This will avoid any possible collision among the robots and guarantees that no robot will occupy in the same grid.

Figure 20 summarizes the message format that the server sends to the robots. There are two types of message. First one transmits with simple data message. There are 6 possible messages. First one is to respond to the robot grid communication message. The second will make to robot to move if the robot was previously stopped. Previously grids are used in this case. Third indicates that the new path is generated. This message will be sent with the new set of grids. Fourth simply changes the speed of the robots. This message may be the result of the speed control. Fifth message toggles the robot navigation from auto to manual. When in manual, the grid-by-grid navigation is performed instead of a sequence of grid. Data message format is also illustrated in the figure.

Path Info Message

Grid Type	Grid List	New/Revised	Sensor Range	Speed Setting

Navigation Status Message

Message Type	Data (Position, Speed, Sensor Range, Speed setting)

Message Type:
0 (OK), 1 (GO), 2 (STOP), 3 (New Position), 4 (New Speed), 5 (New Sensor Range)

Figure 20. The message formats for path information transmitted by the server to the robots

5. EVALUATION

5.1. Simulation Setup

A large-scale map with complex confined corridors and the dimension as illustrated in Figure 21 is considered in the simulation. To cover the entire map, 4 cells are assumed as shown in the figure. The visual sensor coverage sectors are divided into the corridors. The dimension of the fine grid is 2m, and the multiple of the fine grid dimension represents the capacity (i.e., if the corridor width is 6m, the capacity is 3). The dimension of the coarse grid is the maximum width of the corridor. In the simulation, the network delays between the robots, cells, and the server is 200ms. The maximum speed of the robot is 4m/s and average object speed is 2m/s. The illustration of a mobile robot (Dr Robot: X80) equipped with a laser range finder (Hokuyo: UTM-30LX) as shown in Figure 22.

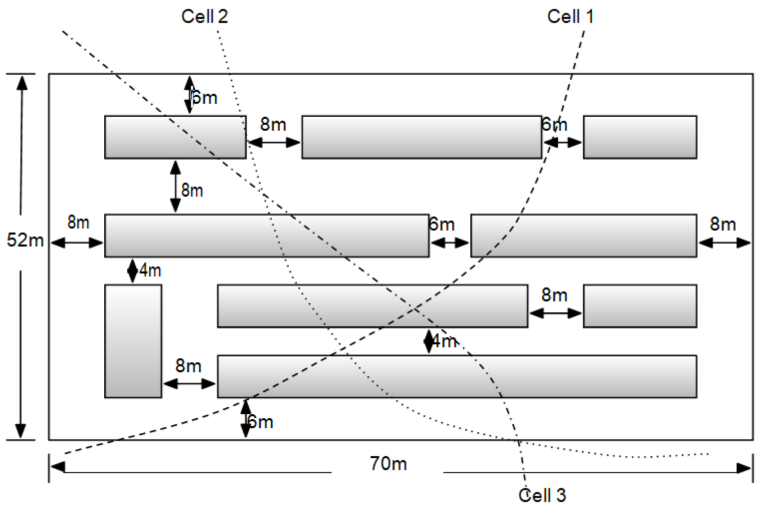


Figure 21. Illustration of simulation input for navigation of multiple robots

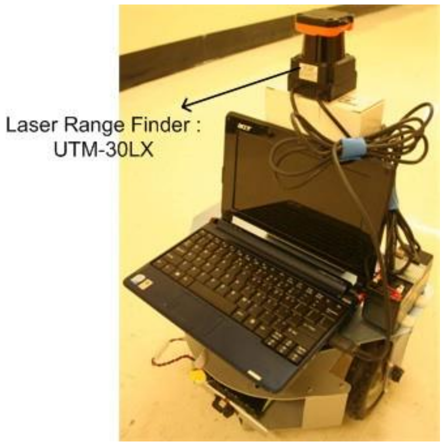


Figure 22. The illustration of a mobile robot (Dr Robot : X80) equipped with a laser range finder (Hokuyo: UTM-30LX).

5.2. Multiple Robot Synchronization with Speed Control

Figure 23 illustrates the speed variation of the robots when all the robots are navigating at the same time. Solid lines indicate the speed and the navigation duration of the robots when they are individually operational. Due to the node ordering, the speed variations and the duration changes are visible when the robots are simultaneously operational. Moreover, due to the lower speed of the robots, the finish times are also extended.

Interaction between the server and the surveillance network is also evaluated. In the simulation, the robots randomly generate the localization errors and request the server for the assistance. The interaction is evaluated with different values of Nreq and Tservice. Note that the corridor index in terms of segment number is illustrated with the requests. If the requests are from the same corridor, the server does not send additional request to the surveillance network since the surveillance network provide information on the corridor. In the first case, the values for Nreq=2 and Tservice=10sec are chosen and the second case, the values for Nreq=4 and Tservice=2sec are chosen. In both cases, the robots generate random localization failures on average every 2sec. The comparison of the speeds are shown in Figure 24.

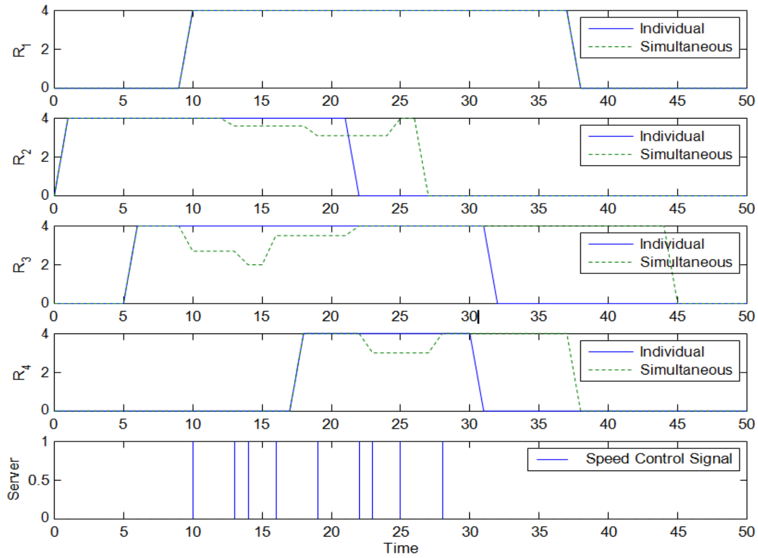


Figure 23. Speed variations as a function of the time when all robots are simultaneously navigating. The speed reduction is due to the node ordering properties.

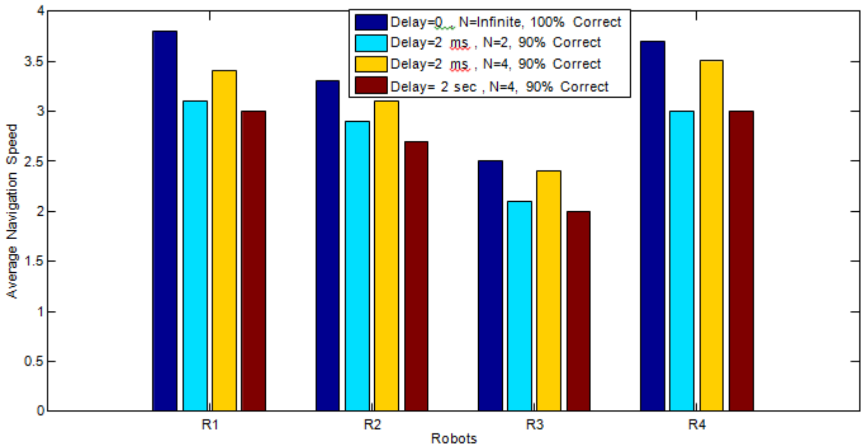


Figure 24. The comparison of the speed for different overhead factors for the surveillance network

5.3. Navigation Effects on Communication Traffics

For each simulation performed in the previous section, the communication traffics are evaluated. The communication messages include normal grid arrival status messages, stopping due to obstacles, etc. Figure 25 illustrates the communication traffic pattern between the robots and the server when the robots are navigating simultaneously. The communication traffic patterns are compared when the speed and the grid size ratio is varied as illustrated in Figure 26. As shown in the figures, the number of communication messages is increased as the speed increases or the grid size decreases. If the network delays are presented, the number of traffic increases as there are additional message exchanges between the robots and the server to resolve the stopped situations.

Figure 27. illustrates the traffic patterns between the robots, the server, and the CCTV system. In this evaluation, random position requests are generated which requires the assistance from the visual CCTV network. In the plot, in addition to the response message initiated by the robots, the change of grid message, change of path message, and change of sensor message are also illustrated. It is clearly shown that the as the number of obstacles and robots increases, the higher communication traffics by the server and the robots.

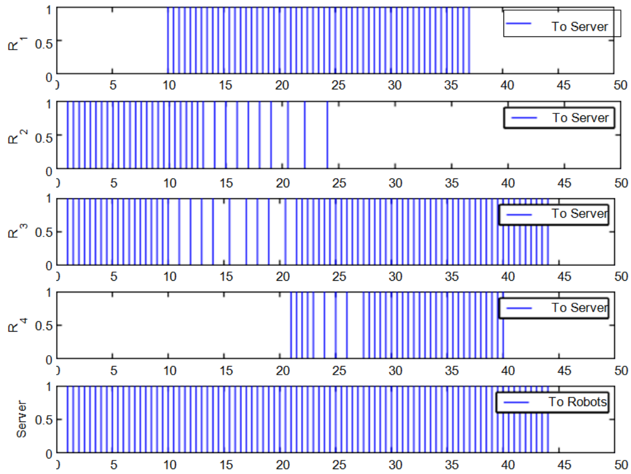


Figure 25. Communication traffics between the robots and the server in the ideal situations

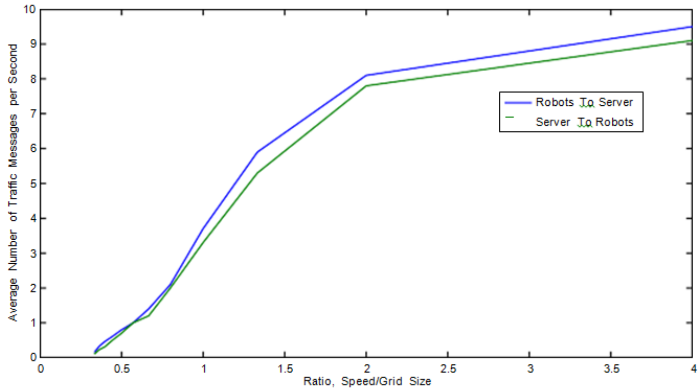
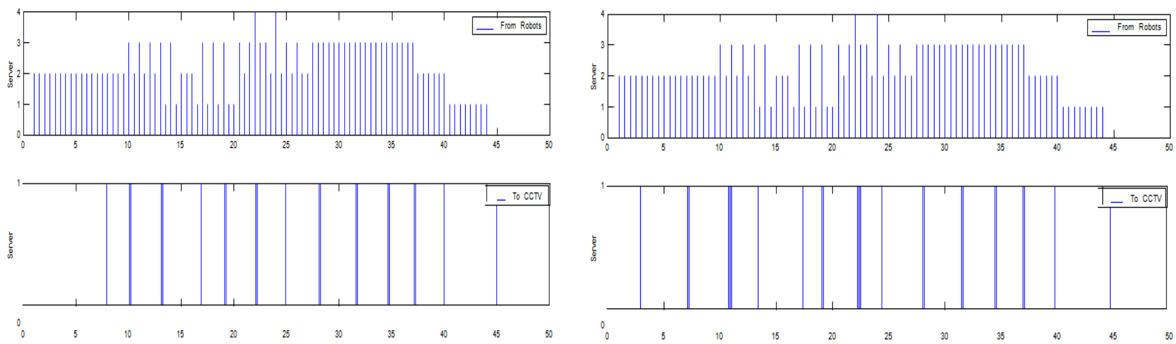


Figure 26. Communication traffics between the robots and the server as a function of the speed and grid size ratio



(a)CCTV Traffic with $N_{service} = 2$ (b) CCTV Traffic with $N_{service} = 4$

Figure 27. Communication traffics between the robots and the server and the CCTV when the visual assistance is required.(a) Nservice = 2 (b) Nservice = 4. In both case, Tservice = 5sec.

6. CONCLUSION

This paper proposes an efficient navigation control and synchronization mechanism of multiple net-worked robots for operation in large confined areas. Adaptable grid based navigation and control strategy is adopted to eliminate potential collisions among robots. Unexpected obstacles are handled and the speed of individual robot is maintained using the node-ordering technique. The proposed navigation control and synchronization mechanism is scalable and can be easily extended for multi-cell large environment. The obstacles information is gathered through local information by the robots for better planning of the navigation. The system collaborates with the existing surveillance systems in case any additional visual information is necessary. The interaction with the surveillance system is minimized to reduce potential overhead. The proposed methodology is evaluated for a large-scale simulation with multiple robots.

REFERENCES

[1] Cludio Rodrigo Torres, Jair Minoro Abe, Germano Lambert-Torres and Joo Incio da Silva Filho (2011). Autonomous Mobile Robot Emmy III, Mobile Robots - Current Trends, Dr. Zoran Gacovski (Ed.), ISBN: 978-953-307-716-1, InTech, Available from: http://www.intechopen.com/books/mobile-robots-current-trends/autonomous-mobile-robot-emmy-iii
[2] Annica Kristoffersson, Silvia Coradeschi, and Amy Loutfi, "A Review of Mobile Robotic Telepres-ence," Advances in Human-Computer Interaction, vol. 2013, Article ID 902316, 17 pages, 2013. doi:10.1155/2013/902316
[3] Stephen McKeague; Jindong Liu; Guang-Zhong Yang, "Hand and body association in crowded environ- ments for human-robot interaction," Proceedings of the IEEE International Conference on Robotics & Automation, pp. 2061-2188, 2013
[4] S. J. You; S. H. Ji," Design of a multi-robot bin packing system in an automatic warehouse," Proceedings of International Conference on Informatics in Control, Automation, & Robotics, pp. 533-538, 2014
[5] Ling Mao, Jiapin Chen, Zhenbo Li and Dawei Zhang. Relative Localization Method of Multiple Micro Robots Based on Simple Sensors. Int J Adv Robot Syst, 2013, 10:128. doi: 10.5772/55587
[6] Jungyoul Lim; Seok Ju Lee; Girma Tewolde; Joonsoo Kwon, "Ultrasonic-sensor deployment strategies and use of smartphone sensors for mobile robot navigation in indoor environment," Proceedings of the IEEE International Conference on Electro/information Technology, pp. 593-598, 2014.
[7] Shuai Zhang; Huawei Liang; Hui Zhu; Daobin Wang; Biao Yu, "A camera-based real-time polarization sensor and its application to mobile robot navigation," Proceedings of the IEEE International Conference on Robotics & Biomimetics, pp. 271-276, 2014.
[8] Hiroshi Noguchi; Takaki Yamada; Taketoshi Mori; Tomomasa Sato, "Mobile robot path planning using human prediction model based on massive trajectories," Proceedings of the Symposium of Image, Signal Processing & Artificial Vision, pp. 1-5, 2013.
[9] Fabio Espinosa; M. Robinson Jimnez; Luis Ruiz Crdenas; Juan Camilo Aponte, "Dynamic obstacle avoidance of a mobile robot through the use of machine vision algorithms ," Proceedings of the IEEE/RSJ International Conference on Intelligent Robots & Systems, pp. 1-7, 2012.
[10] Xie Lihua, "Distributed cooperative control and optimization," Proceedings of the International Con- ference on Control Automation Robotics & Vision, pp. 1-1, 2014.
[11] Adam Wiktor; Dexter Scobee; Sean Messenger; Christopher Clark,"Decentralized and complete multi- robot motion planning in confined spaces," Proceedings of the IEEE/RSJ International Conference on Intelligent Robots & Systems, pp. 1168-1175, 2014.
[12] C. Serkan Karagz; H. I; Bozma; Daniel E. Koditschek, "Coordinated Navigation of Multiple Independent Disk-Shaped Robots," IEEE Transaction on Robotics, pp. 1289-1304, Vol 30, Issue 6, 2014.
[13] Wen-Chung Chang; Chia-Hung Wu; Wen-Ting Luo; Huan-Chen Ling, "Mobile robot navigation and control with monocular surveillance cameras," Proceedings of the International Automatic Control Con- ference, pp. 192-197, 2013.
[14] Tae-Seok Lee; Jae-Guk Lee; Beom-Hee Lee, "A visual landmark design method for high speed multiple robots," Proceedings of the IEEE/SICE International Symposium on System Integration, pp. 573-578, 2011.