

- 1 -

A SYSTEM AND METHOD FOR MANAGING A PLURALITY OF VEHICLES

FIELD OF INVENTION

- 5 The present invention relates to a system and method for managing a plurality of vehicles, particularly for transportation of objects in a manufacturing and/or logistics facility.

BACKGROUND

10

Autonomous mobile vehicles are increasingly being put into service in the next phase of the industrial evolution, whereby the vehicles are configured to navigate autonomously to their intended destinations to carry out desired tasks. This increased adoption of autonomous mobile vehicles is driven by explosive growth in
15 the global supply chain, and a drive to increase yield/productivity using automation.

When using a fleet of autonomous mobile vehicles, it is typical for the vehicles to be configured in a manner which prevents collision with each other. One option is
20 to rely on mutual avoidance methodologies between the vehicles. However, such methodologies are prone to congestion when high numbers of vehicles are traversing within limited floor space/paths, consequently leading to failure of the desired tasks being carried out.

25 Another option would be to generate collision free paths for each vehicle prior to each vehicle's journey. However, whenever there are high numbers of vehicles traversing within limited floor space/paths, congestion typically results which consequently also leads to failure of the desired tasks being carried out.

30 The failure of the aforementioned options is typically due to issues such as, for example:

- 2 -

- a dynamic nature of movement for the plurality of vehicles during the course of operations;
 - imperfect communications network conditions for the plurality of vehicles;
 - 5 - space constrained facilities with narrow passage ways and/or low ceiling heights for traversal of the plurality of vehicles;
 - human intervention or interaction lacking time-based precision with the plurality of vehicles leading to delay/damage to the plurality of vehicles;
 - static obstacles at the facilities (like lifts and doors) and dynamic obstacles (like
 - 10 humans and trolleys) which hamper movement of the plurality of vehicles;
 - multi-level facilities which require multiple maps and access routes;
 - time pressures in the facilities which can cause the plurality of vehicles to function in an undesired manner.
- 15 It is evident that many challenges need to be addressed with regard to managing a plurality of vehicles.

SUMMARY

- 20 In a first aspect, there is provided a system for managing a plurality of vehicles, the system including at least one data processor configured to:
- transmit, from a plurality of sensors integral with the plurality of vehicles, real-time data of an ambient environment and the plurality of vehicles;
 - process, at a multi-agent system, the real-time data;
 - 25 carry out, at the multi-agent system, stateful initialisation of virtual planning agents;
 - generate, at the multi-agent system, actions for the virtual planning agents at a pre-determined juncture;
 - run, at the multi-agent system, a simulation for the generated actions;
 - 30 determine, at the multi-agent system, if all virtual planning agents have non-conflicting paths in space-time; and

- 3 -

receive, at the plurality of vehicles, planned routes for the respective vehicles,

wherein the non-conflicting paths are determined using a trajectory of each virtual planning agent, the trajectory including a parameter of a first volume of
5 each virtual planning agent representing a second volume of each corresponding vehicle.

In a second aspect, there is provided a data processor implemented method for managing a plurality of vehicles, the method comprising:

10 transmitting, from a plurality of sensors integral with the plurality of vehicles, real-time data of an ambient environment and the plurality of vehicles;

processing, at a multi-agent system, the real-time data;

carrying out, at the multi-agent system, stateful initialisation of virtual planning agents;

15 generating, at the multi-agent system, actions for the virtual planning agents at a pre-determined juncture;

running, at the multi-agent system, a simulation for the generated actions;

determining, at the multi-agent system, if all virtual planning agents have non-conflicting paths in space-time; and

20 receiving, at the plurality of vehicles, planned routes for the respective vehicles,

wherein the non-conflicting paths are determined using a trajectory of each virtual planning agent, the trajectory including a parameter of a first volume of each virtual planning agent representing a second volume of each corresponding
25 vehicle.

In a third aspect, there is provided a server for managing a plurality of vehicles, the server including at least one data processor being configured to:

receive, from a plurality of sensors integral with the plurality of vehicles,
30 real-time data of an ambient environment and the plurality of vehicles;

process, at a multi-agent system, the real-time data;

- 4 -

carry out, at the multi-agent system, stateful initialisation of virtual planning agents;

generate, at the multi-agent system, actions for the virtual planning agents at a pre-determined juncture;

5 run, at the multi-agent system, a simulation for the generated actions;

determine, at the multi-agent system, if all virtual planning agents have non-conflicting paths in space-time; and

transmit, to the plurality of vehicles, planned routes for the respective vehicles,

10 wherein the non-conflicting paths are determined using a trajectory of each virtual planning agent, the trajectory including a parameter of a first volume of each virtual planning agent representing a second volume of each corresponding vehicle.

15 In a final aspect, there is provided a non-transitory computer readable storage medium embodying thereon a program of computer readable instructions which, when executed by one or more processors of a server for managing a plurality of vehicles, in communication with the plurality of vehicles, cause the server to carry out a method for managing a plurality of vehicles, the method embodying the steps
20 of:

receiving, from a plurality of sensors integral with the plurality of vehicles, real-time data of an ambient environment and the plurality of vehicles;

processing, at a multi-agent system, the real-time data;

25 carrying out, at the multi-agent system, stateful initialisation of virtual planning agents;

generating, at the multi-agent system, actions for the virtual planning agents at a pre-determined juncture;

running, at the multi-agent system, a simulation for the generated actions;

30 determining, at the multi-agent system, if all virtual planning agents have non-conflicting paths in space-time; and

transmitting, to the plurality of vehicles, planned routes for the respective

- 5 -

vehicles,

wherein the non-conflicting paths are determined using a trajectory of each virtual planning agent, the trajectory including a parameter of a first volume of each virtual planning agent representing a second volume of each corresponding
5 vehicle.

It will be appreciated that the broad forms of the invention and their respective features can be used in conjunction, interchangeably and/or independently, and reference to separate broad forms is not intended to be limiting.

10

DESCRIPTION OF FIGURES

A non-limiting example of the present invention will now be described with reference to the accompanying drawings, in which:

15

FIG 1 is a first schematic view of a first system for managing a plurality of vehicles;

FIG 2 is a schematic view of an example server shown in FIG 1;

FIG 3 is a schematic view of an example vehicle shown in FIG 1;

FIG 4 is a schematic view of a second system for managing a plurality of vehicles;

20 and

FIGs 5A to 5O are flow charts of an example for a method for generating navigation routes for a plurality of vehicles.

DETAILED DESCRIPTION

25

Embodiments of the present invention provide users with a system and method for generating navigation routes for a plurality of vehicles. When generating the navigation routes, the system and method is configured to carry out at least the following aspects for each vehicle, for example:

30

- optimise payloads;

- 6 -

- minimise distance travelled;
 - minimise idle durations for the vehicles;
 - prioritise tasks;
 - minimise delays at remote areas;
- 5 - optimise costs for each route;
- manage parking durations/locations for the vehicles;
 - manage power levels for the vehicles; and
 - ensure workability of the vehicles.
- 10 It should be noted that in the following description, the term "vehicle" is interchangeable with the term "robot", whereby a vehicle is an object that can be controllable and/or programmable to carry out desired tasks. In addition, the term "agent" is defined to mean a virtual entity representing a vehicle.
- 15 An example of a first system 100 for generating navigation routes for a plurality of vehicles will now be described with reference to FIG 1. In this example, the system 100 includes a plurality of vehicles 120, a communications network 150, and a central server 170. While the central server 170 is shown to be a single processing device, it should be appreciated that the central server 170 can be a plurality of
- 20 processing devices. The plurality of vehicles 120 can be configured to receive data from the central server 170, whereby the data is processed at the plurality of vehicles 120 in order to enable movement of the respective vehicles 120.
- The communications network 150 can be of any appropriate form, such as the
- 25 Internet and/or a number of local area networks (LANs). It will be appreciated that the configuration shown in FIG 1 is for the purpose of example only, and in practice the vehicles 120, and the central server 170 can communicate via any appropriate mechanism, such as via wired or wireless connections, including, but not limited to mobile networks, private networks, such as an 802.11 network, the
- 30 Internet, LANs, WANs, or the like, as well as via direct or point-to-point connections, such as Bluetooth, or the like.

- 7 -

Further details of the respective components of the system 100 will be provided in the following paragraphs.

5 ***Central Server 170***

The central server 170 of any of the examples herein may be formed of any suitable processing device, and one such suitable device is shown in FIG 2. The central server 170 is typically administered by, for example, a vehicle fleet
10 management entity or a facility management entity.

The central server 170 is able to communicate with the plurality of vehicles 120, and/or other processing devices, as required, over the communications network 150 using standard communication protocols.

15

The components of the central server 170 can be configured in a variety of ways. The components can be implemented entirely by software to be executed on standard computer server hardware, which may comprise one hardware unit or different computer hardware units distributed over various locations, some of
20 which may require the communications network 150 for communication. A number of the components or parts thereof may also be implemented by application specific integrated circuits (ASICs) or field programmable gate arrays.

In the example shown in FIG 2, the central server 170 is a commercially available
25 server computer system based on a 32 bit or a 64 bit Intel architecture, and the processes and/or methods executed or performed by the central server 170 are implemented in the form of programming instructions of one or more software components or modules 202 stored on non-volatile (e.g., hard disk) computer-readable storage 203 associated with the server 170. At least parts of the software
30 modules 202 could alternatively be implemented as one or more dedicated hardware components, such as application-specific integrated circuits (ASICs)

- 8 -

and/or field programmable gate arrays (FPGAs).

The server 170 includes at least one or more of the following standard, commercially available, computer components, all interconnected by a bus 205:

5

1. random access memory (RAM) 206;
2. at least one computer processor 207, and
3. external computer interfaces 208:
 - a. universal serial bus (USB) interfaces 208.1 (at least one of which is
 - 10 connected to one or more user-interface devices, such as a keyboard, a pointing device (*e.g.*, a mouse 209 or touchpad),
 - b. a network interface connector (NIC) 208.2 which connects the server 170 to a data communications network 150; and
 - c. a display adapter 208.3, which is connected to a display device 210 such as
 - 15 a liquid-crystal display (LCD) panel device.

The server 170 includes a plurality of standard software modules, including:

1. an operating system (OS) 211 (*e.g.*, Linux or Microsoft Windows);
- 20 2. web server software 212 (*e.g.*, Apache, available at <http://www.apache.org>);
3. scripting language modules 213 (*e.g.*, personal home page or PHP, available at <http://www.php.net>, or Microsoft ASP); and
4. structured query language (SQL) modules 214 (*e.g.*, MySQL, available from <http://www.mysql.com>), which allow data to be stored in and
- 25 retrieved/accessed from an SQL database.

Together, the web server 212, scripting language 213, and SQL modules 214 provide the server 170 with the general ability to allow devices connected to the network 150 to access the server 170 and in particular to provide data to and

30 receive data from the database 201.

- 9 -

The boundaries between the modules and components in the software modules 202 are exemplary, and alternative embodiments may merge modules or impose an alternative decomposition of functionality of modules. For example, the modules discussed herein may be decomposed into submodules to be executed
5 as multiple computer processes, and, optionally, on multiple computers. Moreover, alternative embodiments may combine multiple instances of a particular module or submodule. Furthermore, the operations may be combined or the functionality of the operations may be distributed in additional operations in accordance with the invention. Alternatively, such actions may be embodied in the structure of circuitry
10 that implements such functionality, such as the micro-code of a complex instruction set computer (CISC), firmware programmed into programmable or erasable/programmable devices, the configuration of a field- programmable gate array (FPGA), the design of a gate array or full-custom application-specific integrated circuit (ASIC), or the like.

15

Each of the steps of the processes performed by the server 170 may be executed by a module (of software modules 202) or a portion of a module. The processes may be embodied in a non-transient machine-readable and/or computer-readable medium for configuring a computer system to execute the processes. The
20 software modules may be stored within and/or transmitted to a computer system memory to configure the computer system to perform the functions of the module.

The server 170 normally processes information according to a program (a list of internally stored instructions such as a particular application program and/or an
25 operating system) and produces resultant output information via input/output (I/O) devices 208. A computer process typically includes an executing (running) program or portion of a program, current program values and state information, and the resources used by the operating system to manage the execution of the process. A parent process may spawn other, child processes to help perform the
30 overall functionality of the parent process. Because the parent process specifically spawns the child processes to perform a portion of the overall functionality of the

- 10 -

parent process, the functions performed by child processes (and grandchild processes, etc.) may sometimes be described as being performed by the parent process.

5 **Vehicle 120**

An exemplary embodiment of the vehicle 120 is shown in FIG 3. As shown, the vehicle 120 includes the following components in electronic communication via a bus 306:

10

- a location sensor 310;
- RAM 304;
- non-volatile memory 303;
- an image capture component 308;
- 15 - vehicle controller 307;
- a display 302;
- a transceiver component 305 that includes N transceivers; and
- N processing components 301.

20 Although the components depicted in FIG 3 represent physical components, FIG 3 is not intended to be a hardware diagram; thus many of the components depicted in FIG 3 may be realized by common constructs or distributed among additional physical components. Moreover, it is certainly contemplated that other existing and yet-to-be developed physical components and architectures may be utilized to
25 implement the functional components described with reference to FIG 3. The vehicle 120 is preferably encased in a robust chassis, and the respective components of the vehicle 120 also are preferably robust so as to be able to withstand regular impacts without being damaged. In some embodiments, the vehicle 120 is configured to carry loads, and can also be configured to dispense
30 the loads carried by the vehicle 120.

- 11 -

The location sensor 310 can be configured to receive signals from communication nodes to generate location coordinates using, for example, triangulation processes. Alternatively, in outdoor applications, the location sensor 310 can be configured to receive signals from satellites of a Global Positioning System (GPS) and/or Global Navigation Satellite System (GLONASS) network. This allows the vehicle 120 to provide its location in a form of location coordinates whenever necessary.

In general, the non-volatile memory 303 functions to store (e.g. persistently store) data (including images captured by the image capture component 308) and executable code received by and/or required by the vehicle 120 to carry out desired tasks. In some embodiments, for example, the non-volatile memory 303 includes bootloader code, modem software, operating system code, file system code, and code to facilitate the implementation of one or more portions of the method as well as other components well known to those of ordinary skill in the art that are not depicted for simplicity.

In many implementations, the RAM 304 is realized by flash memory (e.g., NAND or NOR memory), but it is certainly contemplated that other memory types may be utilized as well. Executable code in the non-volatile memory 303 is typically executed by one or more of the N processing components 301 to effectuate the functional components.

The image capture component 308 generally operates like a camera to capture images, such as, for example, of objects located around the vehicle 120, whereby the images can be used to determine/confirm a position of the vehicle 120.

In addition, the vehicle controller 307 provides instructions to the vehicle 120 to traverse to a desired location(s) and carry out desired tasks, for example, moving, stopping parking, loading, unloading and so forth. It should be appreciated that the instructions provided by the vehicle controller 307 can be due either to signals

- 12 -

received from the central server 170. The vehicle controller 307 can also include a set of physical controllers on the vehicle 120.

5 The display 302 can be, for example, a LCD or LED panel, which can be used to display a graphical user interface for inputting instructions to the vehicle controller 307, or to display information pertaining to the vehicle 120, for example, faulty codes and status indicators.

10 The transceiver component 305 includes N transceiver chains, which may be used for communicating with external devices via wireless networks. Each of the N transceiver chains may represent a transceiver associated with a particular communication scheme. For example, each transceiver may correspond to protocols that are specific to local area networks, cellular networks (e.g., a CDMA network, a GPRS network, a UMTS networks), and other types of communication
15 networks.

Referring to FIG 4, there is shown a second system 400 for generating navigation routes for a plurality of vehicles. In this example, the system 400 includes a vehicular component 402 continually interacting with an environment 416. The
20 vehicular component 402 comprises both a virtual form and a physical form of vehicles or robots. By interacting with the environment 416, the vehicular component 402 is able to make appropriate decisions for each individual vehicle or robot.

25 The vehicular component 402 includes a multi-agent system 404, and a robotic fleet 410. In some instances, individual vehicles of the robotic fleet 410 are each represented virtually in the multi-agent system 404. The multi-agent system 404 is typically able to simulate movements of the robotic fleet 410 to ensure that desired tasks of the robotic fleet 410 can be carried out.

30

The environment 416 typically encompasses aspects such as, for example, layout

- 13 -

of a facility, configuration of production line, human movements at the facility, machinery placements at the facility, computing resource placements at the facility, sensor placements at the facility, and so forth.

5 The environment 416 typically provides inputs/percepts 412 (obtained using sensors) to the multi-agent system 404, whereby the inputs/percepts 412 includes, for example, pending tasks, path obstructions, route modifications and the like. In addition, the robotic fleet 410 also provides inputs/percepts 406 (obtained using sensors) to the multi-agent system 404, whereby the inputs/percepts 406 includes,
10 for example, respective positions, respective states, respective actions, and so forth.

After processing the inputs/percepts 406, 412, the multi-agent system 404 transmits decisions 408 to the robotic fleet 410, the robotic fleet 410 being
15 instructed to act on the decisions 408, which includes navigation routes for the respective fleet units.

Consequently, when the robotic fleet carries out the decisions 408, the robotic fleet 410 then provides outputs/actions 414 to the environment 416, the outputs/actions
20 414 including information on respective payloads, resolutions of path conflicts, obstacle avoidance routes and so forth. Subsequently, both the environment 416 and the robotic fleet 410 continually feed back the inputs/percepts 412, 406 to the multi-agent system 404, enabling appropriate decisions to be made by the vehicular component 402 in real-time.

25

In this regard, the system 400 is able to provide a robotic fleet which is able to allow respective fleet units to traverse, load payloads, wait and unload payloads in a desired manner within a facility without undesired incidents.

30 A first example of a method 500 for managing a plurality of vehicles will now be described with reference to FIG 5. The method 500 generally relies on, for

- 14 -

example, the central server 170 for carrying out an assessment of the plurality of vehicles before transmitting appropriate outcomes to the plurality of vehicles. While the method 500 is described with respect to the system 400, it should be noted that the method 500 need not be carried out in an identical system 400. For
5 example, the method 500 can be carried out in a decentralised manner, whereby multiple servers and even the processor 301 on each respective vehicle 120 are used to enable the carrying out of the method 500.

Referring to FIG 5A, at step 510, the inputs/percepts 406, 412 are processed at
10 the multi-agent system 404 to acquire real-time data from the environment 416 and the robotic fleet 410. At step 520, a stateful initialisation of virtual planning agents is performed at the multi-agent system 404. Subsequently, actions to be taken by the virtual planning agents in a current iteration are generated at step 530. Consequently, the multi-agent system 404 runs a multi-agent simulation at
15 step 540.

It is determined at step 550 if all agents' path are non-conflicting in relation to both space and time parameters. If true, planned routes are transmitted to the individual vehicles of the robotic fleet 410 at step 570. This is indicated as the decisions 408.
20 If false, dynamic course correction is performed for agents with conflicting paths at step 560, and subsequently, step 550 is carried out again.

FIG 5A shows a generalisation of many of the steps of the method 500. Subsequent FIGs 5B to 5O will depict more detailed break-downs of respective
25 steps of the method 500.

Referring to FIG 5B, there is shown a break-down of the step 520. At step 518, the multi-agent system 404 regenerates past system states. As the system 404 is configured to run in real-time and to react in a dynamic environment, the system
30 404 typically runs continuously. Thus, the decisions (408) made in the past affects decisions made in the present. As such, the regeneration of past system states

- 15 -

involves interpreting what the system 404 has done in the past and combining it with current information about the current environment. For example if an agent has already found a path to its goal a minute ago and is already heading towards its destination, it should continue on its path unless there are obstructions along
5 the agent's path. Thus the system 404 collects information from the environment and combines it together with the current agent's path to determine if rerouting is necessary (for example, if there are any obstructions along the current path).

Subsequently, the multi-agent system 404 carries out a prediction of potential
10 outcomes based on prior system states at step 522, and then proceeds to step 530. Referring to step 522, this step is related to step 518. After the system 404 builds an up-to-date understanding of the environment around it, it determines possible situations that the fleet of agents may encounter and decides what the fleet of agents should be doing consequently - whether each agent of the fleet
15 should continue on their existing path, adopt a new path or plan a new path. The prediction outcomes can include, for example, re-routing agents in order to avoid collisions, continuing normal operation as earlier planned if no new situations have been determined and so forth.

20 Referring to FIG 5C, there is shown a break-down of the step 530. At step 528, the multi-agent system 404 enumerates potential decision policies based on predicted outcomes and previous system states, and subsequently, at step 532, the planning agents are populated with system states, potential outcomes and decision policies. Step 540 then follows.

25

Referring to FIG 5D, there is shown a break-down of the steps 540 and 550. At step 538, it is determined at the multi-agent system 404 if all agents have completed the planning process. If true, all decision outcomes are consolidated at step 544. Step 570 subsequently follows.

30

If false, the multi-agent system 404 then selects an agent with a highest priority at

- 16 -

step 542, and subsequently attempts to find a feasible path to a goal at step 546. At step 548, it is determined at the multi-agent system 404 if a feasible path can be found. If false, step 560 follows, and the multi-agent system 404 is updated with the decision outcomes at step 552. If true, step 552 also follows. Subsequently, at
5 step 554, the decision outcome is communicated to other planning agents, and step 538 follows again.

Referring to FIG 5E, there is shown a break-down of the step 560. At step 562, it is determined at the multi-agent system 404 if there are any alternative destinations
10 en route to the goal. If false, the planning agent remains at the current location at step 564, and proceeds to step 550.

If true, the multi-agent system 404 attempts to locate a feasible path to the new goal at step 566, and at step 568, it is determined if the feasible path can be
15 found. If true, step 550 follows. If false, step 562 follows.

Referring to FIG 5F, there is shown a break-down of the steps 546 and 566. At step 910, a route tracker, tracker and clock of the multi-agent system 404 is initialised. In addition, unexplored regions are also mapped. At step 915, it is
20 determined if initial checks of the agents pass. If false, the clock is reset and the algorithm is terminated at step 920. Step 548 or 568 then follows.

If true, a next point in unexplored regions is selected at step 925. If unexplored regions are available, step 930 follows. If there are no unexplored regions, step
25 920 follows. Subsequently, it is determined at step 930 if the current point of the agent is at the goal for the agent. If true, it is determined at step 940 if the agent is able to remain at the goal until $t_{\max}$, $t_{\max}$ being a maximum simulation time that the agent should be able to remain at the goal before the simulation terminates. If true, a trajectory for the agent is plotted at step 945, before proceeding to step 920. If
30 false, step 920 follows. It should be noted that trajectory refers to a path in space-time.

- 17 -

If false for step 930, the multi-agent system 404 gets all adjacent points to the current point at step 935, and proceeds to step 950 to get subsequent adjacent points to the current point.

5

Subsequently, timings for the respective agents are generated at step 955, and it is determined at step 960 if a route to an adjacent point is available in space-time. If true, the point is added to the route tracker at step 965, and step 950 follows.

- 10 If false for step 960, it is determined at step 970 if exploratory time window φ has been exceeded. If true, step 950 follows. If false, it is determined at step 975 if current point is available after waiting for Γ , a pre-determined waiting time. If true for step 975, step 960 follows. If false for step 975, step 950 follows. The exploratory time window φ is a maximum duration that the agent can remain
15 stationary at its current location while determining subsequent movement for the agent.

- Referring to FIG 5G, there is shown a break-down of step 915. From step 910, it is determined at step 1010 if the agent is at the goal. If true, step 920 follows. If false,
20 it is determined at step 1015 if the agent's goal is occupied. If true, step 920 follows. If false for step 1015, it is determined at step 1020 if there is a path from agent's start point to the goal. If false for step 1020, step 920 follows. If true for step 1020, step 925 follows.

- 25 Referring to FIG 5H, there is shown a break-down of step 925. From step 915, step 1025 appends a start point of an agent to unexplored regions. Subsequently, it is determined at step 1030 if there are unexplored regions. Step 950 can also lead to step 1030.

- 30 If false for step 1030, step 920 follows. If true for step 1030, step 1035 follows by removal of point with highest priority from unexplored regions, and step 1040

- 18 -

follows with the set point being defined as the current point. Step 930 then follows.

FIG 5I follows with a break-down of step 935. Step 1045 follows step 930. At step 1045, the current time is updated to a time when the agent arrives at the current point. Subsequently, at step 1050, all adjacent points are gotten for the current point. Step 950 then follows.

Referring to FIG 5J, there is shown a break-down of step 950. From step 935, step 1055 determines if there are more adjacent points. If false, step 925 follows. If true, step 1060 computes a cumulative cost for the agent to reach respective adjacent points. At step 1065, the adjacent point with lowest cost is selected. At step 1070, it is determined if traversal to the adjacent point creates a circular loop. If false, step 955 follows. If true, step 1075 follows. Steps 965, 970 and 975 can also lead to step 1075. At step 1075, it is determined if there are more adjacent points. If false for step 1075, step 925 follows. If true for step 1075, an adjacent point is removed from consideration at step 1080, and step 1065 follows.

Referring to FIG 5K, a break-down of step 955 is shown. From step 950, step 1085 generates timings such as, for example, arrival times, travel times, stationary times, and so forth. Subsequently, step 1090 follows, which adds routes from a current point to an adjacent point for the route tracker. Step 960 then follows.

Referring to FIG 5L, a break-down of step 960 is shown. From either step 955 or 975, step 1095 determines if a route to the adjacent point is available. If false, step 970 follows. If true, step 1100 determines if the agent can remain for Θ (a pre-defined time) at an adjacent point after arrival at the adjacent point. If false for step 1100, step 970 follows. If true for step 1100, step 965 follows.

Referring to FIG 5M, a break-down of step 965 is shown. Following from step 960, at step 1105, it is determined if the adjacent point is in the route tracker. If false, step 950 follows. If true, at step 1110, it is determined if the cost to the adjacent

- 19 -

point is less than the cost to the current point. If false for step 1110, step 950 follows. If true for step 1110, step 1115 follows, and an adjacent point is added to the route tracker. Subsequently, priority is generated for the adjacent point at step 1120, and step 1125 follows, which adds priority and an adjacent point to
5 unexplored regions. Step 950 then follows.

Referring to FIG 5N which shows a break-down of step 970, step 960 proceeds to step 1130, which advances exploratory time by Γ starting from the current time. Subsequently, step 1135 determines if the exploratory time is greater than current
10 time + φ . If false, step 975 follows. If true, step 950 follows.

With reference to FIG 5O, a break-down of step 975 is shown. Step 970 proceeds to step 1140, which updates timings, for example, arrival time, stationary time, current time, and so forth. Subsequently, step 1145 determines if an agent can
15 remain at a current point for the exploratory time. If false, step 950 follows, and if true, step 960 follows.

In steps 1095, 1100, 1145, while determining if the agent can traverse to an adjacent point, remain at the adjacent point after arriving, or remain at the current
20 point for exploratory time, a trajectory (a path in space-time) is checked to determine if there is any intersection with the trajectories of other agents. The trajectory can be perceived to be a volume that the agent occupies and how it changes over time. For example, the volume of an agent can change in both the space and time dimension as it traverses a route, it can also change in the time
25 dimension alone as the agent remains stationary, it can also scale in size or change in the space dimension alone as a forecast is made of the agent's size to prevent other agents from crossing its path as it performs activities at a desired destination such as picking objects and so forth. Such computations require a determination of the size and kinematics e.g. speed, area and volume of the actual
30 robot.

- 20 -

It should be appreciated that the system and method as described in the preceding paragraphs are configured to enable:

- a. Planning and assigning coordinated routes such that vehicles/robots share
5 the limited area allocated for vehicles/robots in space and time to efficiently manage space available. This will result in, for example, vehicles/robots moving in the same direction following each other by shared reservation rather than exclusive reservation, vehicles/robots moving in opposing directions use routes in the time allocated, such that obstruction-free bidirectional traffic is possible without
10 needing to stop/park outside the travel route in order to give way, and so forth;
- b. Resilience to poor network conditions by incorporating newly available information into the planning iteration and performing replanning to ensure all vehicles/robots reach their respective destinations on time while avoiding collisions;
- 15 c. Incorporating of a concept of a temporary parking point to generate plans so as to enable vehicles/robots to backtrack and avoid collisions in the event there is non-cooperative/non-coordinated traffic in the facility or if a vehicle/robot is broken down;
- d. Incorporating of vehicle/robot kinematics and understanding of a map of the
20 facility to ensure that appropriate vehicles/robots are safely routed through different sections of the facility;
- e. Accounting for delays caused by elements in the facility that require the vehicle/robot to interact with obstacles such as doors and lifts;
- f. Incorporating vehicle/robot speed capabilities (traversal times) such that the
25 vehicles/robots speed up or slow down to meet their target deadlines and also to avoid congestion if one of the vehicles/robots encounters delays;
- g. Incorporating prioritization into planning for each vehicle/robot to cater for varying priorities which could include situations like urgent transport run and battery replenishing run; and
- 30 h. Providing an implementation that is asynchronous and distributed in nature by maintaining all desirable properties of a centralized planner while exploiting the

- 21 -

distributed computational power across multiple vehicles/robots allowing fast generation of trajectories.

Throughout this specification and claims which follow, unless the context requires
5 otherwise, the word “comprise”, and variations such as “comprises” or
“comprising”, will be understood to imply the inclusion of a stated integer or group
of integers or steps but not the exclusion of any other integer or group of integers.

Persons skilled in the art will appreciate that numerous variations and
10 modifications will become apparent. All such variations and modifications which
become apparent to persons skilled in the art, should be considered to fall within
the spirit and scope of the invention.

15

CLAIMS

1. A system for managing a plurality of vehicles, the system including at least one data processor configured to:
 - 5 transmit, from a plurality of sensors integral with the plurality of vehicles, real-time data of an ambient environment and the plurality of vehicles;
process, at a multi-agent system, the real-time data;
carry out, at the multi-agent system, stateful initialisation of virtual planning agents;
 - 10 generate, at the multi-agent system, actions for the virtual planning agents at a pre-determined juncture;
run, at the multi-agent system, a simulation for the generated actions;
determine, at the multi-agent system, if all virtual planning agents have non-conflicting paths in space-time; and
 - 15 receive, at the plurality of vehicles, planned routes for the respective vehicles,
wherein the non-conflicting paths are determined using a trajectory of each virtual planning agent, the trajectory including a parameter of a first volume of each virtual planning agent representing a second volume of each corresponding
20 vehicle.
2. The system of claim 1, wherein the stateful initialisation involves regeneration of past system states and subsequently determining outcomes based on the past system states.
- 25 3. The system of either claim 1 or 2, wherein the generating of actions involves determining at least one of: a priority for the plurality of vehicles, a movement cost for the plurality of vehicles, a traversal time for each of the plurality of vehicles, presence of unexplored regions, and an exploratory time for each of
30 the plurality of vehicles.

- 23 -

4. The system of any of claims 1 to 3, wherein the first volume depends on size and kinematics of the corresponding vehicle.

5. The system of any of claims 1 to 4, wherein the running of the simulation involves providing alternative destinations en route to a desired destination for each virtual planning agent.

6. The system of any of claims 1 to 5, wherein each virtual planning agent is a simulation entity.

10

7. A data processor implemented method for managing a plurality of vehicles, the method comprising:

transmitting, from a plurality of sensors integral with the plurality of vehicles, real-time data of an ambient environment and the plurality of vehicles;

15

processing, at a multi-agent system, the real-time data;

carrying out, at the multi-agent system, stateful initialisation of virtual planning agents;

generating, at the multi-agent system, actions for the virtual planning agents at a pre-determined juncture;

20

running, at the multi-agent system, a simulation for the generated actions;

determining, at the multi-agent system, if all virtual planning agents have non-conflicting paths in space-time; and

receiving, at the plurality of vehicles, planned routes for the respective vehicles,

25

wherein the non-conflicting paths are determined using a trajectory of each virtual planning agent, the trajectory including a parameter of a first volume of each virtual planning agent representing a second volume of each corresponding vehicle.

30

8. The method of claim 7, wherein the stateful initialisation involves regeneration of past system states and subsequently determining outcomes based

- 24 -

on the past system states.

9. The method of either claim 7 or 8, wherein the generating of actions involves determining at least one of: a priority for the plurality of vehicles, a
5 movement cost for the plurality of vehicles, a traversal time for each of the plurality of vehicles, presence of unexplored regions, and an exploratory time for each of the plurality of vehicles.

10. The method of any of claims 7 to 9, wherein the first volume depends on
10 size and kinematics of the corresponding vehicle.

11. The method of any of claims 7 to 10, wherein the running of the simulation involves providing alternative destinations en route to a desired destination for each virtual planning agent.

15

12. A server for managing a plurality of vehicles, the server including at least one data processor being configured to:

receive, from a plurality of sensors integral with the plurality of vehicles, real-time data of an ambient environment and the plurality of vehicles;

20 process, at a multi-agent system, the real-time data;

carry out, at the multi-agent system, stateful initialisation of virtual planning agents;

generate, at the multi-agent system, actions for the virtual planning agents at a pre-determined juncture;

25 run, at the multi-agent system, a simulation for the generated actions;

determine, at the multi-agent system, if all virtual planning agents have non-conflicting paths in space-time; and

transmit, to the plurality of vehicles, planned routes for the respective vehicles,

30 wherein the non-conflicting paths are determined using a trajectory of each virtual planning agent, the trajectory including a parameter of a first volume of

- 25 -

each virtual planning agent representing a second volume of each corresponding vehicle.

13. A non-transitory computer readable storage medium embodying thereon a
5 program of computer readable instructions which, when executed by one or more processors of a server for managing a plurality of vehicles, in communication with the plurality of vehicles, cause the server to carry out a method for managing a plurality of vehicles, the method embodying the steps of:

receiving, from a plurality of sensors integral with the plurality of vehicles,
10 real-time data of an ambient environment and the plurality of vehicles;

processing, at a multi-agent system, the real-time data;

carrying out, at the multi-agent system, stateful initialisation of virtual
planning agents;

generating, at the multi-agent system, actions for the virtual planning agents
15 at a pre-determined juncture;

running, at the multi-agent system, a simulation for the generated actions;

determining, at the multi-agent system, if all virtual planning agents have
non-conflicting paths in space-time; and

transmitting, to the plurality of vehicles, planned routes for the respective
20 vehicles,

wherein the non-conflicting paths are determined using a trajectory of each
virtual planning agent, the trajectory including a parameter of a first volume of
each virtual planning agent representing a second volume of each corresponding
vehicle.