Multi System Mission Control for Teams of Unmanned Marine Vehicles – Software Structure for Online Replanning of Mission Plans

Thomas Glotzbach, Technische Universität Ilmenau; Fraunhofer Center for Applied Systems Technology, Ilmenau/Germany, thomas.glotzbach@tu-ilmenau.de
Matthias Schneider, Technische Universität Ilmenau/Germany, schneider.matthias@tu-ilmenau.de
Peter Otto, Technische Universität Ilmenau/Germany, peter.otto@tu-ilmenau.de

Abstract

The realization of cooperative Multiple Unmanned Marine Vehicles (MUMVs) is a real challenge, due to the conditions of the marine environment. In this paper we show the necessary steps of developments to enable already existing, single autonomous vehicles to participate in a team mission. We describe the design of the control software and put an emphasis on the task of Online Replanning of the vehicles’ mission plans. This Online Replanning is one of the critical issues in the realization of vehicle teams, as existing single autonomous vehicles usually do not use this functionality.

1. Introduction

In this paper we give an account of the proceedings in the European Research Project GREX, which aims to realize cooperating teams of single autonomous marine vehicles. Starting from the basics we discussed in Glotzbach et al. (2007a), where we gave a general overview of different concepts of autonomy and the realization of a simulator for Multiple Unmanned Marine Vehicles, we will now turn to the GREX dedicated hard- and software technologies that need to be integrated into the already existing vehicles to enable cooperative behavior.

The GREX project is not for the development of new marine vehicles for cooperative behavior, but uses already existing vehicles of different supplier (‘heterogeneous vehicles’). This results in the requirement to consider the strict limitations of the vehicle providers concerning the implementation of hard- and software of their vehicles. We will summarize these demands and describe how they influenced our decision making for the final control strategy. In general it can be stated that the usual proceedings for single vehicle missions should also be applied to team missions: The mission need to be described using mission plans that will later be executed by the single vehicles. The mission plans should be adapted to each other, and there is need for adaptation of the plans during mission execution (what we refer to as Online Replanning).

In this framework we will show the significance of an adequate Team-Oriented Mission Planning (TOMP) for both considering the requirements of the providers and dealing with the typical constraints in marine and underwater scenarios. We will show that a coordinated team behavior for heterogeneous marine vehicles requires a good adapted combination of offline mission planning and online replanning during execution. Especially, the communicational constraints of underwater environments demand a high level of autonomy for the vehicles. Thus, an optimal team mission plan need to be designed to minimize the needs of online replanning. Nevertheless, there will never be a realization of a coordinated vehicle team, where every vehicle only follows its own mission plan, even if these plans were designed with great care. So the online mission replanning has the same

1 GREX is the Latin word for herd, team, or swarm. The research project GREX is funded by the Sixth Framework Programme of the European Community (FP6-IST-2006-035223). Participants of the GREX-project are the following companies and institutions: ATLAS ELEKTRONIK GmbH (Germany), Centre of IMAR at Department of Oceanography and Fisheries at the University of the Azores (Portugal), Ifremer (France), Innova S.p.A. (Italy), Instituto Superior Tecnico IST - Lab: Institute for Systems and Robotics ISR (Portugal), MC Marketing Consulting (Germany), ORANGE ENERGY Consulting (Portugal), SeeByte Ltd. (United Kingdom), Institute for Automation and Systems Engineering of the Technische Universität Ilmenau (Germany). See GREX (2008) for further information.
importance as the pre-mission planning. While we demonstrated our proposal for a concept of Team-Oriented Mission Planning in Glotzbach et al. (2007b), we will concentrate on online mission replanning in this paper.

The employed marine vehicles need to be provided with new hard- and software technologies to cover the needs of cooperative behavior. For example, the vehicles need to be equipped with communication technology, especially for inter-vehicle-communication. As most vehicles involved in the GREX project are underwater vehicles, the ability for both aerial and underwater communication is required. Specifically the last one is usually not part of the typical abilities of single autonomous underwater vehicles. A specific software module must be employed to organize the communication with other vehicles and a static/moving base station. There is also the need for a new navigation approach. As the avoidance of collision is one of the most important tasks, a certain software module is responsible to run time-continuous models of the position of all other vehicles, relative to the own position. These position estimations are based upon the messages sent between the vehicles and the calculated distances and directions. Another software module has the task to realize the interfaces between the existing hard- / software of the vehicles and the new, GREX specific ones. Last but not least, there will be a software module called Team Handler which has to monitor the mission progress under the aspects of the overall team mission and need to interact in any cases of discrepancies. This module will be responsible for the execution of the Multi System Mission Control, consisting of Mission Monitoring and Online Mission Replanning. We will describe the software structure of this module.

Mission Monitoring has the task to monitor the progress of the mission and to generate replanning orders, whenever necessary. There will be several instances that are responsible for different tasks and have different possibilities for the creation of replanning orders. We will explain these instances and classify them in the framework of the Rational Behavior Model (RBM). Online Replanning is responsible for the execution of the replanning orders. This includes sorting, validation, communication between vehicles (if necessary) and finally the concrete changes of the mission plan. In this framework we will provide the current perceptions to the question whether hierarchical or peripheral realizations of the control software are more desirable. We raised this question in Glotzbach et al. (2007a).

The online change of mission plans is always high problematic and not really desired by vehicle providers. It must be kept in mind that current autonomous marine vehicles are designed for single autonomous behavior. The implementation of new mission plan elements or the deleting of existing ones should be avoided. Within the GREX project as one of the first real application with Multiple Unmanned Marine Vehicles, the possibility for replanning will be limited to fit the absolutely necessary needs. For future applications, an advanced mission replanning is in the focus of interest. We will conclude our paper with an example mission, displayed in a Virtual Reality especially designed for Marine Vehicles.

2. Basic Conditions within the GREX- research project

In this chapter we outline the general conditions for the realization of cooperating marine vehicles in general as well as the specific conditions within the GREX- project.

2.1. Consequences of the challenges in the marine scenarios

Several technologies which are applied for the realization of cooperating unmanned land/ aerial vehicles cannot be used in the marine environment. Especially two topics shall be discussed here: Self localization and Communication.

Self localization can generally be performed by absolute measurement systems (GPS, DGPS or other navigation systems) or by relative measurement of the vehicle’s movement (odometry, inertial navigation). Both methods have advantages and disadvantages. Absolute, satellite based measurement
systems depend on the reachability of the navigation signals, which requires usually sight contact to a certain numbers of satellites and maybe to a reference radio station. The resulting navigation values are faulty, but the error does not grow. Relative measurement systems deliver very exact values, but the errors grow with time.

Land and air bound systems may use a coupled navigation system to combine the advantages of both methods. For ground vehicle, the reachability of the satellite (and/or radio) signal may be difficult, especially in scenarios in cities, valleys or deep forests.

Autonomous underwater vehicles cannot use any absolute navigation while they are submerged. They can only rely on their internal navigation systems, which have to fulfill high requirements in reliability; otherwise the vehicles have to surface a lot for GPS- updates. In the last years, high-precision inertial navigation systems have been developed, but still they are very expensive; see Höbing-Hölscher and Larsen (2006), for instance. In exchange, marine surface vehicles may rely on satellite-based absolute navigation, as they usually have a very good reception of satellite and radio signals due to their position on the plane sea surface.

In the vehicle teams, which are focus of the GREX- project, this aspect produces completely new possibilities for the navigation of the underwater vehicles. They may use their internal strategies and receive position updates from the surface vehicle(s), which, on their part, use an absolute navigation strategy. Of course this procedure requires good communication abilities between the vehicles, which is the other big problem in underwater scenarios.

Although all GREX vehicles will be equipped with radio modems for surface communication, the submerged vehicles are only able to use acoustic communication to talk to each other or to surface vehicles. The first acoustic underwater communication modules are available on the market. The GREX vehicle will use modems similar to the ones described in Bannasch et al. (2006). It should be mentioned that the current abilities of acoustic underwater communications are still very bad. The communication suffers from a minor reliability and a low bandwidth. Broadcast communication was barley tested so far. These facts need to be kept in mind when developing control strategies for multiple unmanned marine vehicles. We will come back to this in chapter 3.

On the other hand, several acoustic modems like the ones that will be used in GREX offer the possibility to determine range and bearing between transmitter and receiver. This leads to new ways for the navigation of the team vehicles. It is planed to perform team navigation, based on the already existing navigation procedures and the range and bearing information. The concept was demonstrated in Engel and Kalwa (2007). As a result, the vehicles can use a ‘relative’ team navigation to determine where they are in relation to their teammates. The advantage of this strategy is that it helps to prevent inter vehicle collision which is one of the biggest dangers during early GREX missions (these test missions will take place in secluded areas). If the strategy for inter vehicle collision avoidance is based on the navigation procedures of the single vehicles, the different individual error values may add up and lead to a dangerous situation where the vehicles are very close to each other while the navigation systems still assume them far away.

2.2. Consequences of the usage of heterogeneous marine vehicles

It is one of the principles of the GREX project not to develop own unmanned marine vehicles for the cooperating team, but to use existing ones. The idea is to create a conceptual framework and middleware systems which can easily be installed on most existing unmanned marine vehicles. The tests and demonstrations will be performed by several heterogeneous vehicles which are the property of single project participants. Fig.1 provides an overview of the different vehicles.

The decision not to develop an own vehicle was made in consideration of two important aspects. Firstly, as the price for a typical autonomous underwater vehicle today is a six-figure sum, the costs of a research project with the aim of both developing a new vehicle and the middleware system for
cooperation would exceed a reasonable limit. Secondly, it must be kept in mind that it is the intention of the GREX-project to come up with a market-ready product at the end. It can be stated that the currently existing unmanned marine vehicles are very different, and they are very adapted to each individual customer and his business field. It may be nearly impossible to convince possible customers of a GREX-product to not only buy the ‘GREX-cooperation-technologies’, but also new vehicles, which then again needed to be adapted to his specific needs. But if it is possible to enable a large number of already existing vehicles to participate in GREX-missions with only little changes in the hardware, a possible customer may use vehicles he already owns. So from an economic point of view, it was reasonable not to develop a new vehicle, but to use existing ones.

Of course, a consequence of this is that the requirements and restrictions of the vehicle manufacturers and providers should be kept in mind. As three of such institutions participate in the GREX-project, it was a good base to collect the conditions and come up with a list of consequences. One of the first important decisions was the question about the interfaces between the already existing vehicles and the GREX dedicated hard- and software. At the current state, Autonomous Marine Vehicles usually can perform a couple of basis maneuvers, like ‘Go to a certain point’ or ‘follow a certain path’. The operator has to build a sequential mission plan consisting of these elements. This plan will then be executed step by step. These basis maneuvers which we refer as ‘primitives’, or ‘single vehicle primitives’ (SVP) in this case, are perfect suited as interfaces. They are very similar for every vehicle, usually well tested and equipped with security mechanisms, like time and spatial traps. So no further information about the vehicles and their technical realizations are needed. This eases the implementation of the vehicles into the team and may also ensure confidentiality about the functionality of the vehicles. So the proceeding for the team missions will be similar to the single missions: a mission plan need to be developed for every vehicle and will be executed sequentially afterwards.

One important problem arises by using the primitives as basis for the realization of cooperating vehicle teams. To ensure close formation and make the vehicles work together, like following a defined object whose movement cannot be predicted before the mission, there is the need to change
the vehicle mission plans online. The so called ‘Online Replanning’ was one of research focuses of
the research project DeepC and is described in Pfützenreuter (2003). While in DeepC a new vehicle
was developed, the challenge now is to enable Online Replanning on the existing vehicles. This is a
very big problem, as it can be stated that in general vehicle providers and manufactures do not want
their vehicles to perform Online Replanning. At current state, single autonomous vehicles do not
show this behaviour; they simply execute the given mission plan. This is important for the operator, as
especially for submerged underwater vehicles there is usually no communication during mission. The
operator just has to wait at the planned area and the planned time for the surfacing of the vehicle. If
there was any Online Replanning and the vehicle decided to surface at another space and time, the
operator would not know that; he simply had to wait without any information whether the vehicle may
have got lost. That is why there is no Online Replanning for single autonomous vehicles.

In the intended GREX- missions, there is one important difference: Usually surface vehicles will
participate that are able to communicate both with the operator via radio and with submerged
underwater vehicles via acoustic modem. So the vehicle providers agreed to allow Online Replanning,
as long as no single primitives are deleted or new ones are added (this is impossible for most existing
vehicles). This limits replanning activities to the change of parameters of existing primitives and to
jumps within the plan. Nevertheless, it was agreed that these replannings must be limited to the
absolute needs.

3. Realisation of Cooperating Marine Vehicles – The Main Ideas

It is clear what it will be necessary to install new software on the vehicle to enable the cooperative
behavior. As every vehicle may have different hardware specifications, it was decided that the
software will run on a specific GREX- hardware that must be implemented into the vehicles and must
be connected to the already existing hardware (Vehicle Internal Control). Fig.2 provides an overview.

There will be four software instances that run on the new hardware. They realize all technologies that
are necessary for the cooperative behavior and expand the abilities of the single autonomous vehicle.
The Communication Module is responsible for all communication, both between different vehicles
and different modules within one vehicle. For inter vehicle communication, all available possibilities
(radio and/or acoustic modem) will be used. The Team Navigation Module will realize a relative
position estimation of all vehicles, according to the description given in chapter 2.1. The Team
Handler Module is responsible for the cooperative behavior. This means, the Multi System Mission
Control is performed within this module. The Mission Control will monitor the progress of the
mission, based on position and payload data, and will decide, whether or not the execution of the current mission plans will still lead to a successful mission. If not, it will perform the Online Replanning. This will be described in chapter 4. Last but not least, there is the GREX Interface Module (GIM) which is the interface between GREX- and vehicle hardware. In this chapter we discuss the question, whether the team control software of the Team Handler should be organized in a hierarchical or peripheral way, introduce the term ‘Multi Vehicle Primitives’ and summarize the meaning of an adequate mission planning.

3.1. Hierarchical vs. Peripheral realization

In Glotzbach et al. (2007a), we discussed two different control structures to enable cooperative behavior of single autonomous vehicles, as shown in Fig.3. In the hierarchical approach, there was always one vehicle dedicated as leader. This vehicle runs the special team software and was responsible for coordinating all team participants. The approach leads to clear control structures, but creates a lot of communication needs, as the leader must be involved in every decision. Alternative, we suggested a peripheral approach, where every vehicle runs team dedicated software. As long as two vehicles are in communication range, they will talk to each other and coordinate their plans. This approach seems more feasible for underwater scenarios, but does not have clear control structures. As it can be seen in Fig.3, both the (already existing) control software for the vehicles and the team dedicated software are organized according to the Rational Behavior Model (RBM). The Rational Behavior Model, a typical control structure in robotic research, suggested in Kwak et al. (1992), is a hierarchical architecture with three levels named Strategical, Tactical and Executive Level. If this model is used for single autonomous robots, the Strategical Level is usually assigned with the decisions concerning the whole mission. The Tactical Layer is responsible for the execution of the current maneuver. The Executive Level has the task to generate the concrete steering commands for the actuators of the robot. As stated, the control software for single vehicles is usually designed in this way. We will also design the Team Level Software in that way, as good experiences were made in the DeepC- project.

Fig.3 a & b: Hierarchical and Peripheral Control Structure for the Team Behaviour, as suggested in Glotzbach et al. (2007a)

As Online Replanning needs to be minimized, there is the need for an adequate offline mission planning that need to address the special needs of cooperative vehicles. Therefore, it was decided to develop special primitives for the planning of the Team Mission Plan. This plan describes the activities that shall be performed by the whole team. The primitive for cooperative behavior were called ‘Multi Vehicle Primitives’ (MVP). Fig.4 shows the MVP ‘Coordinated Path Following’ as an example. Here, the vehicles are commanded to move on a specified path in a given formation. The other two parameters are the desired velocity and the acceptable error $\varepsilon$. This MVP is translated for every vehicle to a Path Following- SVP, while the velocities need to be adapted online. The block
Keep Formation as Multiple Base Primitive (MB) is responsible for this algorithm. We will show this in the example in chapter 4.

![Diagram](image)

**Fig.4:** Example for the design of a Multi Vehicle Primitive (MVP) and the translation into Single Vehicle Primitives (SVP)

The important issue of the introduction of MVPs is that the question whether the realization should be hierarchical or peripheral is not that fundamental anymore. Instead, it is possible to realize every MVP in another way. So the optimal way can be found for each situation. By now, a couple of MVPs were designed to fulfill all requirements of the missions that shall be realized at the end of the project. The implementation of the first MVPs will be ready this summer.

### 3.2. Team-Oriented Mission Planning (TOMP) as precondition

![Diagram](image)

**Fig.5:** Concept of Team-Oriented Mission Planning (TOMP)

Summarizing the explanations up to this point, it can be stated that for the realization of cooperative behavior for unmanned marine vehicles, it is important to find a good, well-balanced proportion between an Offline Mission Planning and an Online Mission Replanning. We provided an overview on our suggestion for Team-Oriented Mission Planning (TOMP) in Glotzbach *et al.* (2007b) and will only give a short overview here. The main idea is shown in Fig.5. From a general MVP-Pool, the operator builds the Team Mission Plan (TMP) in a GREX Meta-language at Team Level. This plan will be translated by the planning software into a couple of Single Vehicle Mission Plans (SVMP), one for each vehicle. These plans are formulated in GREX Meta-language at Vehicle Level which is based on the planning language of the Mission Planning Software SeeTrack® by the project partner SeeByte. This software is already available for single autonomous vehicles and will be upgraded for vehicle teams during the project. More details can be found in Cormack (2006) and Arredondo and Cormack (2007).

The Single Vehicle Mission Plan is translated into the Real Vehicle Languages by the GREX Interface Module (GIM). It will be the task of each vehicle provider or manufacturer to develop the
part of the GIM which is responsible for the translation, starting from a completely documented Meta-language. This proceeding is already in use for single autonomous vehicles and has proven successful, as SeaTrack® is a market-ready product and currently in use by several end users.

The proposed concepts also enable an easy implementation of new MVPs, if new mission scenarios claim for that. The MVP- Pool can be expanded without the need of any other changes. Also, new single vehicles can be introduced into the ‘GREX- world’ by simply developing a new GIM for them.

4. Multi System Mission Control

The task of Multi System Mission Control which is accomplished in the Team Handler Module can be split into Mission Monitoring and Mission Replanning. The main task of the Mission Monitoring is the recording of deviations between the claimed and the real state of a particular vehicle. Monitoring is also responsible to suggest replanning activities to overcome the detected deviation or problem. Mission Replanning is responsible to change the Mission Plan according to the specifications made by Mission Monitoring. Changes at the mission plan may result from coordination, maneuver management, exception treatment or optimization intents. The Technische Universität Ilmenau gained first experiences in the concept and realization of Mission Monitoring and Mission Replanning for one single AUV in the research project DeepC, as described in Pfützenreuter (2003). In the DeepC-project, the aim was the development of a 2.4 tons single autonomous underwater vehicle with a diving depth of 4000 meters and a mission time up to 60 hours. For this purpose, it was essential to develop a monitoring and replanning system that was able to deal with all possible exceptional circumstances and to optimize a mission plan to reach the best possible mission result (in terms of benefit for the user) in the current situation. The Mission Monitoring uses a Knowledge Base to create the Replanning Requests that are executed by the Replanning Module. Afterwards, the replanning is checked with the usage of a digital sea chart to detect impossible routes. The principle is shown in Fig.6.

As said before, the vehicles employed within the GREX- project do not execute any mission replanning on their own, and the replanning activities ordered by the GREX- dedicated software should be limited to absolute necessary changes, which excludes complex replanning or optimization. Nevertheless, this is an important topic for future research activities, so we developed a structure for Multi System Mission Control that can fulfill all the requirements of the real missions in the test phase, but can also be used for extended research activities in the area of mission replanning, that will only be executed in the simulator at the moment.

Fig.6: Structure of the Mission Control within the DeepC- project, see Pfützenreuter (2003)
4.1. Mission Monitoring

Mission Monitoring is responsible for the observation of all current states and for the creation of replanning commands in case of deviations. The prior execution of a Team-Oriented Mission Planning, as described above, is an important precondition to minimize the work of Mission Monitoring. Starting from this, the work of Mission Monitoring is separated into several instances that are executed simultaneously. Several monitoring tasks run at the same time and use different algorithms to create replanning commands. The concrete algorithms for each instance will usually depend on the current primitive of the mission plan.

The following four instances were defined and will be realized:

- **Coordination of Mission Plans to guarantee the formation/inter vehicle collision avoidance**
  This instance will mainly adapt the velocities of the vehicles. This can be necessary whenever the preservation of a close formation is desired. As described before, the way to reach a close formation is to plan adequate paths for all vehicles in TOMP. Then the formation stays theoretically intact, as long as all vehicles use exactly the correct velocity. As this can never be achieved in reality, there is the need for velocity adaptation. A concrete example for this is shown in chapter 4.3. Other primitives may require similar algorithms. In every case this primitive has to guarantee for the prevention of inter collision avoidance. If there are intersections in the planned paths of the vehicles (at the present moment it is still not decided whether this should be prohibited), this instance has to adapt the velocities to prevent collisions.

- **Modification of Mission Plans to realize obstacle avoidance**
  This instance will be responsible for a Team-Oriented Obstacle Avoidance. This does not include inter vehicle collision avoidance, because this task is executed by Team-Oriented Mission Planning and the Adjustment-Instance by creating paths with preferable few intersections and by the Coordination-instance by adjusting the velocities. Obstacle avoidance corresponds to the recognition and evading of all external objects which are not part of the team. Single vehicles may employ sonar systems and automatically avoid single obstacles. By the use of long-range sonar it is possible to plan and realize an optimal path through an area with many obstacles. For single autonomous vehicles, this was shown in Eichhorn (2004). The same principle can also be employed for vehicle teams, where the preservation of a formation or only slight formation changes is demanded. As this requires very specific vehicle equipment and the areas of the first demonstrations of the GREX scenarios are assumed to be held in secluded areas, this instance will only be realized in theoretical research.

- **Adjustment of Mission Plans to realize high-level coordination within a mission plan element**
  This instance is responsible for executing the current primitive in the correct way. This includes the monitoring of primitive time and area (if the primitive contains adequate time and spatial traps) as well as the replanning of vehicle paths to reach the goal of the primitive. If, for example, the vehicles have to follow a methane plume to find a hydrothermal vent, this instance has to replan the paths according to the values of the methane sensors of the vehicles. In this case, the team will have a hierarchical realization as one vehicle takes over the leadership, collects the values of all vehicles, calculates new track data and spreads them among the others.

- **Reorganization of Mission Plans in cases of special situations/emergencies**
  This instance can perform significant changes, like cancel the mission, reorganize the mission (if a single vehicle has to leave the team due to an error), optimize the plan (only for theoretical research)
Looking at the four instances in the sequence in which they are described here shows that the mightiness of the instances grows from the top, while the frequency of their activation sinks. This proceeding corresponds to the suggestion of the Rational Behavior Model, as stated before. We use this architecture for the realization of Multiple Unmanned Marine Vehicles, where the intent of the three levels stays exactly the same as in the use for single autonomous robots (see above). It is possible to assign the four described instances of the Mission Monitoring to the three levels of the Rational Behavior Model, as it is shown in Fig. 7.
Fig. 8 shows the complete structure of the Team Handler with the Mission Monitoring on the left side. The interfaces of the Team Handler to the other software modules are the Input and Output Processors for receiving and transmitting of data telegrams like commands, measurement values or reports. The main routine of the Team Handler decides which of the monitoring instances need to be activated according to the type of incoming data or to a special event like a timer. The corresponding instance gets the data from the main routine and may produce replanning commands. An example for this will be shown in chapter 4.3.

Each Instance may send requests to the Data Management and receive information, like mission plan elements or vehicle parameters. If for example a new turn needs to be planned, the corresponding Instance needs to know the minimum turning radius of all involved vehicles. The instance ‘Adjustment’, which is responsible for the monitoring of the current primitive as a whole, has to detect when a primitive is terminated. A corresponding message will then be sent to the Data Management. Each instance can produce Replanning Commands which are handed over to the Mission Replanning Module. Also, instances may produce information or commands for other software modules which are sent to the corresponding addressee via the output processor.

4.2. Mission Replanning

The Mission Replanning is depicted in the right part of Fig. 8. It receives the replanning commands from the Monitoring. These commands are stored in a deque. A deque is a data storage (deque = Double Queue) which is designed like a stack, but can be accessed on both sides. It is the task of a Sorter to put the commands into the deque, while a Selector will chose the sequence in which the commands shall be executed. Therefore the commands will have a priority tag which is set by the monitoring instances. Both Sorter and Selector will have the task to decide which replanning command is most important and need to be executed prior to other ones. Also, both have the ability to delete single commands, for example if a replanning command for a certain parameter is still not executed, and already a new one is handed over. Also, a replanning of a velocity in order to keep a close formation will simply be ignored in the presence of a replanning command which cancels the whole mission. At the current state, the exact design of Sorter and Selector has not been made and will depend on further research activities.

The current replanning command to execute is handed over to the Replanner. This is the central instance of Mission Replanning. It will perform the real replanning and spread the information to every other involved module. In this framework it has to decide whether the current replanning commands affects only the own vehicle plan (local replanning) or also other vehicles (global replanning). Local replannings can concern velocity adaptation, for example, while global replannings may change tracks or jump to alternative parts of the mission plan. Of course only the current leading vehicle may produce (and execute) global replanning commands.

Finally, the changing of the plan is executed, and the new plan elements are inserted into the Data Management. At the same time, the changes are transmitted to the corresponding Grex Interface Module to also influence the real mission plan of the vehicle. If necessary, the replanning command is also sent to the Team Handler of other vehicles. After this message reaches another vehicle, the Main Loop of the corresponding Team Handler gets the message from the Input Processor and directly hands it over to the Sorter. In general, replanning commands from other vehicles will have a very high priority, as only the leading vehicle will produce global replanning commands.

Whenever a plan change is handed over to the Grex Interface Module, the command will be translated into the ‘real vehicle language’. This part of the GREL- software need to be produced by each particular vehicle provider, as only they may know the internal functionality. Of course they may also validate the claimed changes. Afterwards, they will send back status info to the Replanner, stating whether the replanning was performed successfully or the replanning has been denied. As it is shown in Fig. 8, this status info will be sent from the GIM to the Team Handler, and handed over from the Input Processor to the Main Routine and from there to the Replanner in the Mission Replanning. The
The replanning process is finally finished when a positive Status Info is received by the Replanner. If the replanning was denied by the vehicle, a special situation occurs, that is difficult to handle. If the replanning command was a global one, maybe other vehicles have already executed it. If one vehicle denies the replanning, it may be necessary to undo all other changes, which is again replanning action for the other vehicles. If then another vehicle denies this ‘undoing’, a situation occurs where different vehicles have different mission plans which may lead to a disaster. For land vehicles, it may be a possible solution that all vehicles have to test a replanning, before they really execute it. Only if all vehicles agree, the replanning will finally be done. In the marine scenarios, this proceeding does not work due to the large latency in communication. If the Team Handler of a leading vehicle commands a plan change, it may take several minutes before all vehicles have sent back the status info. To overcome this problem, it was decided in the project group that at the moment every denying of a replanning command by a vehicle results in a mission abort. Then it can be checked by the engineers, why the replanning was denied. As the instances of the Mission Monitoring have access to the parameters of all involved vehicles, such a denying of replanning should not occur too often, at least after the test phase.

4.2. Example

As an example we assume a situation where three vehicles have to perform a lawn mowing maneuver, for example to scan an area and to collect data. For this reason they have to establish and preserve a close triangle formation. The first step to solve this challenging task is an adequate Team-Oriented Mission Planning to provide all vehicles with a relevant mission plan. These plans contain paths for every vehicle; these path are designed in a way that if all vehicles will run absolutely the desired velocity, the formation will remain in the desired pattern. We showed in Glotzbach et al. (2007b) that it is possible to perform such a path planning. Of course it will never be possible that the vehicles run with absolute equal velocities, which is why the Online Replanning is necessary.

In the described situation there is the MVP ‘M_CoordinatedPathFollowing’ active which was depicted in Fig.4. In this primitive, the Mission Monitoring will mainly employ the Coordination-Instance, while the other three instances will only be used in abnormal situations. The Coordination-Instance will be employed by the Main Routine whenever new position values are received from the Team Navigation Module. These telegrams will be sent from the Team Navigation to Team Handler in regular intervals; they contain the estimated current position of all vehicles, based on navigation data of the vehicles own navigation, range and bearing measurements of the acoustic modems and position estimations by Kalman Filters. With the position of all vehicles and the knowledge of the mission plans, the Coordination-Instance will calculate the ‘Level of performance’ \( \Theta \) of the current path (straight line or arc) for all vehicles which runs from zero to one. The goal is to equalize these levels for all participating vehicles. This task can be fulfilled individually by all vehicles. If \( \Theta_m \) of vehicle \( m \) is smaller than the mean value of the team, vehicle \( m \) has to increase its velocity, and vice versa. The control algorithm used for this task is suggested from the project partner Instituto Superior Tecnico IST in Ghabcheloo et al. (2005). If there are \( p \) vehicles, vehicle \( k \) must adapt its velocity \( v_k(t) \) according to the following equation:

\[
v_k(t) = v_k'(t) + \sum_{i=1}^{p} K_i \cdot (\Theta_i(t) - \Theta_k(t))
\]

In equation 1, \( v_k'(t) \) is the desired velocity for vehicle \( k \) according to the mission plan for the current situation. \( K_i \) is an individual proportional factor. The task of the controller is to equalize the values of \( \Theta \) for all vehicles. This controller executes the function of the Multiple Base Primitive, as shown in Fig.4.

The Coordination-Instance of the Mission Monitoring will employ this algorithm whenever new position values are available. The new velocities will be calculated not only for the own vehicle, but also for all other vehicles. The results of the calculation will be sent back to the Team Navigation.
Module. This module can even use these values to improve the position forecast until the next messages of the other vehicles arrive. This approach takes into account that the other vehicles run the same control algorithms, and they will also continuously adapt their velocities. They should calculate very similar velocity corrections, because they employ the same algorithm and they use their own Team Navigations that are supposed to deliver similar position values.

The Coordination Instance will deliver replanning commands to the Mission Replanning. These commands contain the number of the current line in the mission plan, the order to change velocity and the new value. The sorter module will receive the command and sort it into the deque. In the described scenario, it can be assumed that nearly no other replanning commands will be delivered to the Deque. There is no need for any path replanning or adaptation according to any payload data. So as long as no critical situation occurs that requires a mission abort, the only replanning activities will be velocity adaptations. The sorter and selector will also have the task to organize the replanning commands. So the frequency of velocity replannings (which will take several seconds in the vehicle hardware) can be changed to a reasonable level. The Replanner will perform the velocity changes which are all local replannings at the end.

As shown in Fig.9 and Fig.10, with the describe proceeding it is possible to establish a formation between unmanned marine vehicles while meeting all requirements and conditions of vehicle providers and marine environment. In the demonstrated mission task ‘Coordinated Path Following’, it can be stated that the control hierarchy is very peripheral oriented. The Team Handlers will initiate no extra communication to the requirements of Team Navigation. All vehicles will use fixed algorithm on the base of the position values produced by Team Navigation. The request to minimize the communication was fulfilled here.
5. Conclusion

The challenging problem to enable a team of single autonomous unmanned marine vehicles for cooperation need to be solved based on the particular task. That means, several control algorithms and proceedings need to be implemented into the main control structure which therefore need to be developed very generic. Additional, the requests of the vehicle providers and the limitations of the marine environment need to be addressed. As a basic structure, we demonstrated our concept of Multi System Mission Control and its basic concepts Mission Monitoring and Mission Replanning, with an adequate Team- Oriented Mission Planning as precondition. This meets the request to work with fixed mission plans, to minimize communication and to sustain all existing security mechanisms of the single vehicle. Also, different control strategies can be implemented into this structure. We showed the Coordinated Path Following from IST as an example. Within the GREX- project, the consortium will demonstrate the abilities of Initialization of Formation and Coordinated Target Tracking in the framework of the summer trial in July 2008 at the Azores. More practical solutions will be shown until the project ends in 2009, while several research activities will be performed theoretically using the simulator, for example mission plan optimization or dealing with the loss of a vehicle.

Acknowledgements

The authors gratefully acknowledge financial support of the European Community in the framework of the research project GREX. The project GREX (FP6-IST-2006-035223) is funded by the Sixth Framework Programme of the European Community.

References


CORMACK, A.L.R. (2006), Providing a common operator interface for port and harbour security and shallow water mine warfare, Undersea Defence Technology Europe Conf. and Exhibition (UDT Europe), Hamburg


GREX: Coordination and control of cooperating heterogeneous unmanned systems in uncertain environments, Home Page, http://www.grex-project.eu, visited on 14/03/2008

HÖBING-HÖLSCHER, U.; LARSEN, M.B. (2006), Aided inertial navigation system solutions - Application for synthetic aperture sonar, 25th Int. Conf. on Offshore Mechanics and Artic Engineering (OMAE), Hamburg
