AUTONOMOUS AND DECENTRALIZED MISSION PLANNING FOR CLUSTERS OF UUV

L. Giovanini, J. Balderud and R. Katebi

Industrial Control Centre, University of Strathclyde, Glasgow,
Graham Hill Building, 50 George St., Glasgow G1 1QE, UK

Abstract: This paper considers a method for improved operational robustness in applications where clusters of Unhabitated Underwater Vehicles (UUVs) are deployed for the purpose of achieving a shared common mission. These applications are typically characterized by their strict requirement for coordination between the individual UUVs. The proposed method is based on the assumption that the shared mission objectives and the associate constraints can be stated in terms of a mixed-integer quadratic optimization problem and known by all UUVs. Thus on the outset, the method considered in this paper is similar to a centralized approach but solved in a decentralized manner: the problem is decomposed into smaller sub-problems and solved in a parallel using a distributed Nash-based game approach. Sufficient conditions for numerical convergence of the method together with performance results are also presented. Finally, simulation results are presented to demonstrate the main features of the proposed method.

Keywords: game theory, mission planning, MPC, multiobjective optimization.

1. INTRODUCTION

The problem of allocating resources and assigning tasks in multi-team system is an extremely important step in insuring that maximum overall performance of the system is achieved. A mechanism that allows for reallocation of resources and reassignment of tasks is important in the control of complex dynamic systems especially when the initial deployment of resources and assignment of tasks appear to be ineffective in yielding satisfactory results. Examples are dynamic systems that are controlled by a group of agents where each group is divided into several teams and each team is allocated a specific task. As the operation of the overall system progresses, the group may reassess his initial task assignment among the teams and may decide that a different assignment could yield better overall performance of the system. This case, a reassignment of tasks and a redeployment of resources will have to be performed. In a similar manner, when a specific team completes its initial assignment, the group may consider two options: it may decide to terminate this team's activity (i.e., retire the team) or reassign the team to another ongoing task. In the former case, the control of the system will continue but with fewer teams and in the latter the team may be merged with one of the remaining teams to help improve its ability to complete its task. These are all important, but very complex, issues that need to be considered in any control architecture that involves a multitude of teams and tasks.

While cooperation between teams is desirable, it can be very complicated to implement. To perform these missions, acceptable algorithms must be solved in real time, taking into account the need for task precedence and coordination, timing constraints and feasible trajectories. One of the main difficult features of a cooperative control problem is complexity. The size of the problem (e.g. number of vehicles, number of tasks, resources and performance constraints, disturbances, etc.) is one form of complexity, however coupling between tasks interaction dominates the complexity problem (Chandler et al., 2002).

Emerging cooperative decision and control algorithms of different classes have been proposed for solving such problems. These algorithms are based on customized combinatorial optimization methods including: mixed integer linear programming (*MILP*) (Richards et al., 2002; Schumacher et al., 2004) capacitated transhipment problem (Schumacher et al., 2002), and iterative capacitated transhipment problem (Chandler et al., 2002). Due to the special characteristics of the problem and the requirement for a tractable solution, all of the proposed algorithms are suboptimal in some sense.

Many different candidate cooperative control algorithms have been developed, implemented, and simulated (Alighanbari et al. 2003, Chandler et al., 2002, Guo and Nygard, 2002; Murphy, 1999; Nygard et al., 2001; Schumacher, 2001), but, due to the complexity of this problem, all of these algorithms have been heuristic in nature. Many of these algorithms also do not meet all of the requirements of the assignment problem, i.e. assignment coordination, task precedence, and feasible trajectories.

This paper considers a method for improved operational robustness in applications where UUVs are deployed for the purpose of achieving a shared mission objective. The method under consideration assumes that the shared mission objectives and the associate constraints can be stated in terms of a constrained quadratic optimization problem. The method also assumes that the shared mission objectives and the structure of the quadratic optimization problem is known by all UUVs. Thus on the outset, the method considered in this paper is similar to a centralized approach, where the optimization problem would have been solved by a supervisory. However, instead of solving the optimization problem in a centralized manner, the problem is decomposed into smaller sub-problems and solved in a parallel and decentralized manner using a distributed Nash-based game approach (Waslander et al., 2003)]. The paper will show that the proposed method allows for a higher degree of autonomy in each UUV, and thus better overall operational robustness.

The paper is organized as follows. In Section 2, a distributed *MPC* algorithm based on Nash optimality is proposed. In Section 3, the convergent condition of the distributed predictive control algorithm for linear models is analyzed. The nominal stability and the performance deviation under communication failure are analyzed, respectively, in Sections 4 and 5. A simulation example is provided to demonstrate the efficiency of the distributed *MPC* algorithm in Section 6. Conclusions are given in Section 7

2. OPTIMAL CONTROL PROBLEM

In this paper, we present a reconfigurable navigation algorithm for autonomous navigation in a highly complex environment, the controller is desired to avoid obstacles, compensate the effects of the sea currents or other vehicle constraints and must heed vehicle state and control constraints. We will assume that an off-line path planner has provided a series of waypoints to the goal; however, because the controller will be robust to uncertainties in the a priori information from which path plans were generated, the way points do not have to define feasible straight line paths in the environments. We also assume that a scene analysis software exists that localizes active threat obstacles as a function of the vehicle's position.

The control problem is formulated as a discrete finite-time quadratic tracker with penalties on control effort and on tracking error relative to a reference trajectory. The problem is to determine the sequence of actuator commands $u(k)$ $k\in[l,\dots,l+N-1]$ that minimizes a cost function, which will be a function of the trajectory and control action. Besides, the geometrical features of the environment and obstacles can be translated into inequalities constraints of output variables to pose a constrained optimization problem, such that the vehicle navigates safely and adaptively.

To pose the current problem as a static multi-assignment is problematic, since the cost of performing a set of tasks by any vehicle is a function of the order in which the tasks are performed. This is not the case in a static multi--assignment.. Suppose that $S_i=\{\Gamma_{i1}, \dots, \Gamma_{in}\}$ is an ordered subset of tasks, then we refer to S_i in the context of autonomous vehicle V_j performing the tasks in S_i in the order in which they appear in S_i. The first and last tasks performed by V_j are Γ_{i1} and Γ_{i1} respectively. One can associate a cost with V_j performing tasks in S_i. A feasible partition of tasks is an allocation of disjoint subset of tasks for each of the vehicles to perform; so that every task is performed by some vehicle and all timing (coordination) constraints on the tasks are met. The problem of resource allocation may be posed as finding the minimum cost partition of the set of tasks, where $\mathcal{P}$ is any feasible partition of tasks.

Suppose there are n tasks and m vehicles, the task assignment problem can be solved through the following mixed-integer optimization problem

$$
\begin{array}{ll}
(1.a) & \min\limits_{I_l(k),\, U_l(k)} \sum_{l=1}^{m} J_l\left(I_l(k), X_l(k), U_l(k)\right) \\
 & st. \\
(1.b) & x_l(k+i+1,k) = A_l x_l(k+i,k) + B_l u_l(k+i,k) \quad x_l(k,k) = x_l(k) \\
(1.c) & \sum_{l=1}^{m}\sum_{j=1}^{n} I_{lj}(k) = n \\
(1.d) & \left|C_l x_l(k+i,k) - C_j x_j(k+i,k)\right| > \delta \qquad \forall k,\ \delta>0,\ l,\ \ j=1,\dots,\ m \\
(1.e) & G\left(X_l(k),\ U_l(k)\right) \le 0
\end{array}
$$

where $[A_l\ B_l]$ $l=1,\dots,m$ is the dynamical model of the l th vehicle, $I_{lj}(k)$ is an integer variable than can only assume the values 0 or 1, and $J_l(\cdot)$ is the cost function for the lth vehicle $l=1,\dots,m$ is given by

$$J_l = \sum_{j=1}^{n} I_{lj}(k) T_{lj}\left(X_l(k), U_l(k)\right) \qquad (2)$$

where the state and control trajectories are given by $X_l(k)=[x_l(k|k)\dots x_l(k+N|k)]$ and $U_l(k)=[u_l(k|k)\dots u_l(k+N-1|k)]$ respectively.

The variables $x(k+i|k)$ and $u(k+i|k)$ are, respectively, the predicted state and the predicted control at time $k+i$ based on the information at time k and system model. The constraints $G(X_l(k), U_l(k))$ represent physical limits in the system and can also be the geometrical characteristics of the operating area.

The cost function $J_l(k)$ quantify the cost of allocating a group of task in a given sequence to vehicle l. It includes the transition cost between two successive tasks, $T_{lj}(x_l(k), u_l(k))$, and the integer variable $I_{lj}(k)$ $j=1,\ldots, n$ that indicate that indicates which tasks are executed by vehicle l. This cost function introduces a coupling effect between the vehicles through the task-vehicle pairing. This coupling effect has a positive effect on the system behaviour because it introduces a cooperative effect between the vehicles. The transition cost $T_{lj}(x_l(k), u_l(k))$ includes the distance to be navigated and the amount of energy required to move from one location to another. This term represents some of the objectives of the mission. In general, this term is a nonlinear function of the vehicle's states (position and speed) and the control inputs, however for the case of minimum distance navigation $T_{lj}(x_l(k), u_l(k))$ can be written as a quadratic function

$$T_{lj}\left(x_l(k),\ u_l(k)\right)=\sum_{i=1}^{N}\left\|x_T-C_l x_l(k+i)\right\|_{Q_l}^2 + \left\|C_l\left(x_l(k+i)-x_l(k+i-1)\right)\right\|_{Q_l}^2+\left\|u_l(k+i)\right\|_{R_l}^2 \qquad (3)$$

where x_T is the position where the task j should be performed and N is the time horizon.

The constraint (1.c) is included to ensure that all tasks will be performed and if a task is not performed by two vehicles simultaneously, unless the task explicitly required two or more vehicles to be performed. In this case, a similar constraint to (1.c), but only applied to a specific task j should be formulated

$$\sum_{l=1}^{m} I_{lj}(k)=m_j \quad j\in[1,\ldots,n], \qquad (4)$$

where m_j is the number of vehicles required to perform the task. The constraint (1.d) is employed to avoid the collision between vehicles by ensuring that distance between the vehicles is bigger than a minimum distance δ.

The optimization problem (1) distribute the task such that all the tasks are assigned to a vehicle in a way that the overall cost function and minimize and the constraints are satisfied. In this way, the resulting task assignment is optimal for the information available at time k. The optimization produces an open-loop optimal control sequence in which the first value of the decision variables is applied to the system $u_l(k)=u_l(k|k)$. Then, the controller waits until the next control instant and repeats this process to find the next control action. If any new information is acquired later, the optimization problem can modify the task distribution in order to improve the efficiency or to satisfy the new constraints.

There are two interesting situation: a) one situation is when $n>m$, in this case there are more vehicles than tasks therefore $n-m$ vehicles can be assigned to tasks resulting in a faster fulfilment of the tasks, and b) an alternative situation happens when $n<m$, in this case there are more tasks than vehicles. These two situations lead to multiple assignment problems.

3. DISTRIBUTED MODEL PREDICTIVE CONTROL

Model predictive control (*MPC*) is formulated as resolving an on-line open loop optimal control problem in moving horizon style. Using the current state, an input sequence is calculated to minimize a performance index while satisfying some specified constraints. Only the first element of the sequence is taken as controller output. At the next sampling time, the optimization is resolved with new measurements from the plant. Thus both the control horizon and the prediction horizon move or recede ahead by one step at next sampling time. The purpose of taking new measurements at each sampling time is to compensate for unmeasured disturbances and model inaccuracy, both of which cause the system output to be different from its prediction.

For large-scale systems, because of the effect of horizon N, the number of optimized control variables $U(k)$ at each sampling time are highly dimensional, the computation is intensive which requires high performance computers or advanced algorithms. To avoid the prohibitively high on-line computational demand, this work proposes a distributed scheme under network environment. These autonomous subsystems are connected via network with dynamic input coupling among them, share the common resources, communicate and co-ordinate each other in order to accomplish the whole objective. Given that the behaviour of the system is described by m agents, the cost function of the system J can be rewritten as follows

$$J=\sum_{l=1}^{m} J_l\left(I_l(k),X(k),U_l(k),U_{l\neq p}(k)\right) \qquad (6)$$

for $p=1,\ldots,m$ where the original cost function has been decomposed into m performance indexes related with the local decision variable $U_l(k)=[u_l(k|k)\cdots u_l(k+N|k)]$ $l=1, \ldots, m$. However, the outputs each subsystem are still related to the remaining decision variables $U_{l\neq n}(k)$ $n=1, \ldots, m$. In this way, the optimization (1) can be decomposed into m coupled optimization problems

$$\min_{U_l(k)} J_l\left(X(k),U_l(k),U_{l\neq p}(k)\right) \qquad (5.a)$$
$$st.$$
$$x(k+i+1,k)=Ax(k+i,k)+B_l u_l(k+i,k)+\sum_{\substack{p=1\\p\neq l}}^{m} B_p u_p(k+i,k) \quad x(k,k)=x(k) \qquad (6.b)$$
$$\sum_{l=1}^{m}\sum_{j=1}^{n} I_{lj}(k)=n \qquad (6.c)$$
$$J_l\left(X(k),U_l^q(k),U_{p\neq l}^q(k)\right)\le J_l\left(X(k),U_l^{q-1}(k),U_{p\neq l}^{q-1}(k)\right) \quad l,p=1,\ldots,\ m \qquad (6.d)$$
$$G\left(X(k),U_l(k),U_{p\neq l}(k)\right)\le 0 \qquad (6.e)$$

where $U_{l\neq n}(k)$ is assumed given. The constraint (6.d) has been added to ensure the convergence of the

sinc(*i*) Laboratory for Signals and Computational Intelligence (http://fich.unl.edu.ar/sinc)
L. Giovanini, J Balderud & R Katebi; "Autonomous and Decentralised Mission Planning for Clusters of UUVs"
Mini Symposia in Marine Systems, International Control Conference 2006. sep, 2006.

coupled optimization problems and the stability of the closed-loop system. This constraint requires the cost function values to remain constant or to decrease at each time step, relative to the cost value computed using the current measurement and the restriction of the input. Because the cost function is defined on an infinite horizon, and includes contributions from both the state and the input, this can be thought of as a generalized state contraction constraint. However, because the state and input are required to contract on an infinite horizon, this constraint is less conservative than a finite horizon contraction constraint.

At each sample time k, the right hand side of (6.d) is constant and the left hand side is a strictly convex function of $U^q(k)$ $q\geq 0$. If the objective function $J_l(X(k),U_l(k),U_{l\neq p}(k))$ $l=1, \ldots, m$ is also strictly convex, which means that the optimization problem (Dpc) is convex. This fact implies that any local minimum is also the global minimum. The fact that the objective function is strictly convex also implies that the solution, if it exists, is unique. The existence of a feasible solution $U(k)$ implies the existence of a unique optimal solution $U^*(k)$.

Given that the communication network is reliable and with capacity that allows the subproblem to exchange information while they solve their local optimization problem, then such distributed problem can be solved by means of the Nash optimality concept (Nash, 1951).

Definition: *A group of control decisions* $U(k)=[U_1(k)\ \cdots\ U_m(k)]$ *is said to be Nash optimal if*

$$J_l\left(X(k),U_l^q(k),U_{p\neq l}^q(k)\right)\leq J_l\left(X(k),U_l(k),U_{p\neq l}^q(k)\right) \quad (7)$$

If the Nash optimal solution is achieved, each subproblem does not change its control decision because it has achieved the locally optimal objective under the above conditions; otherwise the local performance index $J_l(X(k),U(k))$ will degrade. Each $U_l(k)$ subsystem optimizes its objective function using its own control decision assuming that other subsystems' solutions are known and optimal. So, if condition (7) is satisfied, the whole system has arrived to an equilibrium point (attractor) in the coupling decision process (see Figure 1). Since the mutual communication and the information exchange are adequately taken into account, each subsystem subsystems solves its local optimization problem provided that the other optimal solutions are known. Then, each agent compares the newly optimal solution with that obtained in the previous iteration

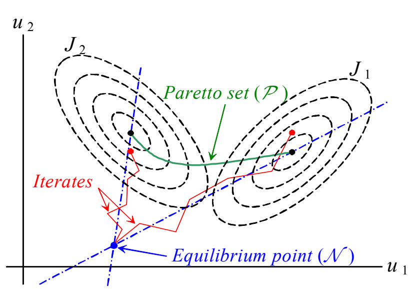

Fig 1. Trajectories in decision space traced by two subsystems

and checks if the terminal condition is satisfied. If the algorithm is convergent, all the terminal conditions of the m agents will be satisfied, and the whole system will arrive at equilibrium at this time.

Algorithm

Step 1: At sampling time instant k, each subsystem makes initial estimation of their decision variables and communicates it to the other agents, let the iterative index $q=0$

$$U_l^q(k)=\left[u_l^q(k)\cdots u_l^q(k+M)\right] l=1,\ldots,\ m. \quad (8)$$

Step 2: Each agent solves its optimization problem (6) simultaneously to obtain its solution

Step 3: Each agent checks if its terminal iteration condition is satisfied

$$\left\|U_l^q(k)-U_l^{q-1}(k)\right\|_\infty \leq \varepsilon_l \quad l=1,\ldots,\ m. \quad (9)$$

If all the conditions are satisfied, then end the iteration and go to **Step 4**; otherwise

$$q=q+1,\ U_l^q(k)=U_l^{q-1}(k) \quad l=1,\ldots,\ m \quad (10)$$

all agents exchange this information through communication and go to **Step 2**.

Step 4: Computes the instant control law

$$u_l(k)=\left[I\ 0\cdots 0\right]U_l^{q-1}(k)\ l=1,\ldots,\ m. \quad (11)$$

Step 5: Move horizon to the next sampling time, $k+1\rightarrow k$, and go to **Step 1**, repeat the process.

In distributed control, each subsystem can work independently to achieve its local objective, but can not accomplish the entire objective on its own. To this purpose, the subsystems communicate, coordinate and negotiate with each other, exchanging information through a communication network. The question that naturally emerges is how the decomposition of the systems and global objective affects the performance of the decentralized scheme, compared with the solution of the global cost function, and the existence of a solution of the optimization problem (6).

CONVERGENCE ANALYSIS

The convergence of the optimization problem depends on how the global problem is split and the parameters, Q_l and R_l $l=1, \ldots, m$ (Giovanini and Balderud, 2006). Therefore, it is difficult to guarantee the convergence when there is uncertainty in the system. This is the main reason to introduce the contractive constraint (6.d), which is used to guarantee the robust convergence of the optimization problem.

Theorem *When the input $U(k)$ is computed using the optimization problem* (6), *the iterative procedure converges to an equilibrium point.*

Proof. Convergence of the state and input to the origin can be established by showing that the sequence of optimal plant cost values is non-increasing.

At any iteration $q\geq 0$ of a given time step k, is known that the control law $U^q(k)$ is a feasible solution of problem (6). The plant cost results from the evaluation of the cost J_l for the feasible solution $U_l^q(k)$

$$J_l^q(k) = J_l\left(X(k), U_l^q(k), U_{p\neq l}^q(k)\right). \quad (12)$$

Once the optimal solution $U^*(k)$ has been found, the true cost of this solution can be determined by evaluating the cost function

$$J_l^*(k) = J_l\left(X(k), U_l^*(k), U_{p\neq l}^*(k)\right). \quad (13)$$

Assuming that a feasible solution have being obtained at the initial step $q=0$, given by

$$U_l^0(k) = \left[u_l^0(k), \ldots, u_l^0(k+M)\right] \quad (14)$$

the cost is given by

$$J_l^0(k) = J_l\left(X(k), U_l^0(k), U_{p\neq l}^0(k)\right). \quad (15)$$

Assuming that the first control input $u_l^0(0)$ is applied to the system, the states at the next sampling time is given

$$x(k+1) = Ax(k) + Bu^0(k). \quad (16)$$

The constraint (6.*d*) must be satisfied in the next iteration $q=1$, this means that the cost $J^1(k)$ can not exceed the cost of the previous iteration

$$J_l^1(k) = J_l\left(X(k),\ U_l^1(k), U_{p\neq l}^1(k)\right) \leq J^0(k) \quad (17)$$

Because they are computed using the true plant, the states sequence in (15) and (17) are identical for $q\geq 1$. Subtracting the cost functions at $q=0$ and $q=1$ we obtain

$$J_l^1(k) - J_l^0(k) \leq -\Delta U_l^0(k)^T \mathcal{R} \Delta U_l^0(k) \quad (18)$$

where

$$\Delta U_l^0(k) = U_l^1(k) - U_l^0(k). \quad (19)$$

The same argument can be repeated at subsequent iteration to show that

$$J_l^q(k) - J_l^{q-1}(k) \leq -\Delta U_l^{q-1}(k)^T \mathcal{R} \Delta U_l^{q-1}(k)\ \forall q. \quad (20)$$

This shows that the sequence of cost $\{J_l^q(k)\}$ is non-increasing and the cost is bounded below by zero and thus has a non--negative limit. Therefore as $q\to\infty$ the difference of cost $\Delta J^q(k)\to 0$ such that the cost converge to the optimal $J^q(k)\to J^*(k)$. Because R is positive definite, as $\Delta J^q(k)\to 0$ the updates of the inputs must converge to the origin $\Delta U^{q-1}(k)\to 0$ as $q\to\infty$, thus the distributed optimization problem converges to a solution □.

In the particular application of mission planning, the system is constituted by a set of decoupled dynamical systems (vehicles) which are coupled through the cost functional and the constraints (1.*c*) and (1.*d*). Therefore, for this application the decentralized algorithm will converge to a solution as long as the optimal problem (1) is feasible.

STABILITY ANALYSIS

The stability of the closed-loop system, like the convergence of the optimization problem, depends on how the global problem is split and the parameters, Q_l and R_l $l=1, \ldots, m$ (Giovanini and Balderud, 2006). Therefore, it is difficult to guarantee the convergence when there is uncertainty in the system. This is the main reason to introduce the contractive constraint (6.*d*), which is used to guarantee the robust stability of the system.

Theorem *When the input is computed using the optimization problem* (6) *the origin is an asymptotically stable equilibrium point with a region of attraction consisting of all* $x_0 \in R^n$.

Proof. The closed-loop stability can be proven by shown that the input and the true plant state converge to the origin, and then it is shown that the origin is an stable equilibrium point for the closed-loop system. The combination of convergence and stability gives asymptotic stability.
Convergence of the state and input to the origin can be established by showing that the sequence of optimal plant cost values is non-increasing. This fact can be proved following a similar procedure like in Theorem 1. Then, the stability emerges from the fact that the cost function J is a Lyapunov function of the closed-loop system. □

The combination of convergence and stability implies that the origin is asymptotically stable equilibrium point of the closed-loop system.

SIMULATION AND RESULTS

Considered in this section is a cluster of UUVs deployed on a search and exploration mission, where the mission objective is to search and explore the area in the vicinity of a pre-defined search path.

During path following search and exploration missions that involves multiple search agents it is normally desired to utilize the full potential of the combined search and exploration capability of the deployed agents. During these types of missions it is normally also desired to organize the search agents such that they travel in formation along the search path. This is to ensure that the agents remain closely together which, in turn, gives the cluster a better and more structured ability to adapt appropriately to unforeseeable changes in the environment, such as changing search paths and/or the discovery of non-penetrable obstacles, and a better and more structured ability to cope with failures of individual search agents.

In order to maximize the area searched by the UUVs, whilst simultaneously keeping the UUVs reasonably close together, the UUVs should be organized into line formation that is perpendicular to the direction of the search path. It is assumed that there are N UUVs positioned along this formation line and that each UUV has a search range equal to L units of distance. The position, at time instant k, of each UUV (along the formation line) is denoted $Y(k)=[y_1(k)\ \cdots\ y_N(k)]$. The optimal position of the UUVs, that maximizes the area searched by the UUVs, can then be expressed in terms of the solution to the following constrained quadratic optimization problem,

$$\begin{aligned}&\min_{Y(k)} \sum_{i=1}^{N-1}\left(2L - y_i(k) + y_{i+1}(k)\right)^2 + \sum_{i=1}^{N} y_i^2(k) \\ &\quad st. \\ &\tilde{A}Y(k)^T \leq \tilde{b}\end{aligned} \quad (21)$$

The first term in the above cost-function ensures that the UUVs are spaced $2L$ units of distance apart

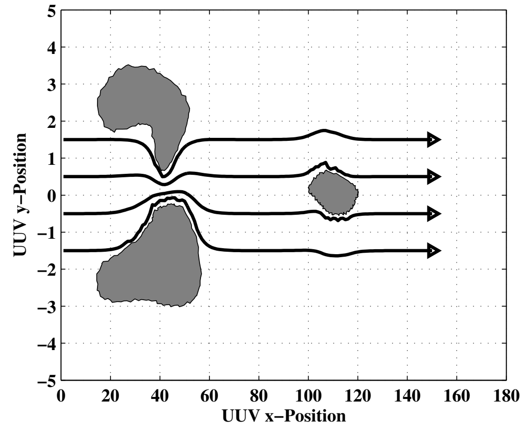

Fig 2. Optimal search trajectories for the UUVs.

during optimal conditions. The second term ensures that the formation line is centered on the search path. The set of linear inequality constraints in the above optimization problem provide a systematic way of incorporating the natural constraints created by any non-penetrable obstacles.

The optimization problem (21) yields the position of the UUVs at time instant *k*, and does not take into account of external influences, such as obstacles, that may affect the position of the UUVs at time step *k*+1. To better cope with external influences it is often necessary to carry out the optimization, defined by (21), over a longer time horizon. The above optimization problem can be decomposed into smaller sub-problems and solved using the iterative method previously proposed in this paper. For the decomposed problem, the optimization problem solved by each agent (UUV) is defined by,

$$\begin{aligned} \min_{y_i(k)\cdots y_i(k+H)} & \sum_{j=1}^{H}\left(2L - y_i(k+j) + y_{i+1}(k+j)\right)^2 \\ & + \left(y_i(k+j) + c_i(k+j)\right)^2 \\ & + \left(y_i(k+j) - y_{i+1}(k+j-1)\right)^2 \end{aligned} \qquad (22)$$

$$st.$$

$$\tilde{A}\left[y_i(k)\cdots y_i(k+H)\right]^T \le \tilde{b}$$

where $c_i(k+j)$ is defined as follows

$$c_i(k) = -y_i(k) + \sum_{n=1}^{N} y_n(k). \qquad (23)$$

A simulation scenario has been prepared and simulated where four UUVs are deployed on a search and exploration mission in an archipelago like environment. In this simulation scenario the formation must be continuously adapted to maintain optimal search performance and to avoid any obstacles. The simulation employs the following set of parameters, N=4, H=20, L=0.5

The results from this simulation is shown in Figure 2. It is assumed in the simulation that the position of the obstacles is known beforehand by all UUVs and that the UUVs have the capability to progress one unit of length in the x-direction at each time step. At each simulation step the agents solve (21) by interactively solving (22).

CONCLUSIONS

In this study a distributed model predictive control method based on Nash optimality is developed. The *MPC* is implemented in distributed scheme with the inexpensive agents within the network environment. These agents can co-operate and communicate each other to achieve the objective of the whole system. Coupling effects among the agents are fully taken into account in this scheme, which is superior to other traditional decentralized control methods. The main advantage of this scheme is that the on-line optimization of a large-scale system can be converted to that of several small-scale systems, thus can significantly reduce the computational complexity while keeping satisfactory performance.

Acknowledgments

The authors are grateful for the financial support for this work provided by the Engineering and Physical Science Research Council grants GR/R04683/01 and GR/S91031/01.

REFERENCES

Alighanbari M., Kuwata Y. and J. How. “Coordination and Control of Multiple UAVs with Timing Constraints and Loitering”, in Proc. of *American Control Conference* (2003).

Chandler P, Pachter M, Swaroop D, Fowler J, Howlet J, Rasmussen S, Schumacher C and K. Nygard. “Complexity in UAV Cooperative Control”, in *Proceedings of the American Control Conference*, (2002).

Giovanini L and J Balderud. “Game Approach to Distributed Model Predictive Control”, International Control Conference 2006, UK (2006).

Guo W and K Nygard. “Combinatorial trading mechanism for task allocation”, in Proc. 13th *International Conference on Computer Applications in Industry and Engineering* (2002).

Liu Y,. Simaan M and J Cruz. “An application of dynamic Nash task assignment strategies to multi-team military air operations”, *Automatica*, vol. 39(12), pp. 1469-1478 (2003).

Murphy R. “An approximate algorithm for a weapon target assignment stochastic program”, *Approximation and complexity, in Numerical Optimization: Continuous and Discrete Problems*, Kluwer Academic Publishers (1999).

Nash J. “Non cooperative games”. *Annals of Mathematics*, vol. 54(2), pp. 1951.

Richards A, Bellingham J, Tillerson M and J How. “Coordination and control of multiple UAVs”, in Proc. of *AIAA Guidance, Navigation, and Control Conference* (2002).

Schumacher C., Chandler P. and S. Rasmussen. “Task allocation for wide area search munitions via network flow optimization”, in Proc. of *AIAA Guidance, Navigation, and Control Conference* (2001).

Schumacher C., Chandler P and S Rasmussen. “Task allocation for wide area search munitions”, in Proc. of *American Control Conference* (2002).

Schumacher C, Chandler P, Pachter M and L Pachter. ``Constrained Optimization for UAV Task Assignment", in Proc. of *AIAA Guidance, Navigation, and Control Conference* (2004).

Waslander S, Inalhan G and C Tomlin. “Decentralized control via Nash bargaining”, In *Conf. on Cooperative Control and Optimization* (2003).