Proceedings of the 17th World Congress
The International Federation of Automatic Control
Seoul, Korea, July 6-11, 2008

Distributed Control Design for Underwater Vehicles

J. Balderud*, L. Giovanini* and R. Katebi*

*Industrial Control Centre, University of Strathclyde, Glasgow, UK

Email: r.katebi@eee.strath.ac.uk

Abstract: The vast majority of control applications are based on non-interacting decentralized control designs. Because of their single-loop structure, these controllers cannot suppress interactions of the system. It would be useful to tackle the undesirable effects of the interactions at the design stage. A novel model predictive control scheme based on Nash optimality is presented to achieve this goal. In this algorithm, the control problem is decomposed into that of several small-coupled mixed integer optimisation problems. The relevant computational convergence, closed-loop performance and the effect of communication failures on the closed-loop behaviour are analysed. Simulation results are presented to illustrate the effectiveness and practicality of the proposed control algorithm.

1. INTRODUCTION

Control of multiple-input multiple-output (MIMO) systems can be accomplished either by a centralized multivariable controller or by a set of independent single-input single-output (SISO) controllers. Despite the superior closed-loop performance of centralized multivariable controllers, there are many reasons for which the decentralized control is the dominating and preferred method used in practice. In fact, the latter approach exhibits a list of advantages over the former one: flexibility in operation, failure tolerance, simplified design and tuning, etc. (Campo and Morari, 1994). Also, communication between controllers, start up and shut-down schemes and identification experiments are more complicated for the centralized control. The multivariable controllers such as model predictive controllers (MPC) are usually used in a supervisory mode with decentralized PID controllers stabilising the system at the regulation control level.

In spite of their practical benefits, single loop controllers cannot often suppress the effects of interactions of MIMO process (Johansson and Rantzer, 1999). When the process interactions are significant, it is of key importance to choose the right control structure, i.e. right pairing of the inputs and outputs (Conley and Salgado, 2000), (Lee and Edgar, 2002). However, appropriate control structure selection and controllers tuning are not sufficient to eliminate the input–output coupling.

Previous work on distributed MPC is reported in (Oschs *et al.*, 1998). The proposed algorithms use a wide variety of approaches, including multi-loop ideas (Oschs *et al.*, 1998), decentralized computation using standard coordination techniques (Giovanini *et al.*, 2007), robustness to the actions of others (Camponogara *et al.*, 2002), (Jia and Krogh, 2001), (Jia and Krogh, 2002), penalty functions and partial grouping of computations (Keviczky *et al.*, 2004). The key point is that, when decisions are made in a decentralized fashion, the actions of each subsystem must be consistent with those of the other subsystems, so that decisions taken independently do not lead to a violation of the coupling constraints. The decentralization of the control is further complicated when disturbances act on the subsystems making the prediction of future behaviour uncertain.

In this paper, we consider situations where the distributed controllers can exchange information several times every sample. The objective is to achieve some degree of coordination among agents that are solving MPC problems with locally relevant variables, costs, and constraints, but without solving a centralized MPC problem. Such coordination schemes are useful when the local optimization problems are much smaller than a centralized problem. Here we assume that the connectivity of the communication network is sufficient for the subsystems to obtain information regarding all the variables that appear in their local domain. In this case, we are interested in identifying conditions under which the agents can perform multiple iterations to find solutions to their local optimization problems that are consistent in the sense that all shared variables converge to the same values for all the agents. We also show that when convergence is achieved using this type of coordination, the solutions to the local problems collectively solve an equivalent, global, multi-objective optimization problem. In other words, the coordinated distributed computations solve an equivalent centralized MPC problem. This means that properties that can be proved for the equivalent centralized MPC problem (e.g., stability) are valid for the solution obtained using the coordinated distributed MPC implementation.

The paper is organized as follows. In Section 2, a distributed MPC algorithm based on Nash optimality is proposed. In

Section 3, the convergence condition of the distributed predictive control algorithm for linear models is analysed. The nominal stability and the performance deviation under communication failure are analysed in Sections 4. A simulation example is provided to demonstrate the efficiency of the distributed MPC algorithm in Section 5. Conclusions are given in Section 6.

2. DISTRIBUTED MODEL PREDICTIVE CONTROL

MPC is formulated here as resolving an on-line open-loop optimal control problem in a moving horizon style. At decision instant *k*, the controller samples the state of the system $x(k)$ and then solves an optimization problem of the following form to find the control action:

$$\min_{U(k)} \frac{1}{2}(X(k)_{ref} - X(k))^T Q(X(k)_{ref} - X(k)) + \frac{1}{2}U(k)^T RU(k)$$

$$\text{s.t. } X(k) = Gx(k) + HU(k) \text{ and } TU(k) \le b \qquad 1$$

where the state, input and output predicted trajectories are given by

$$X(k) = \{x(k,k)\cdots x(k+n_p,k)\},$$
$$X_{ref}(k) = \{x_{ref}(k,k)\cdots x_{ref}(k+n_p,k)\}$$
$$U(k) = \{u(k,k)\cdots u(k+n_c,k)\}$$

G and H are the observability and Hankel matrices of the system and $n_c \le n_p$. The variables $x(k+i,k)$, $x_{ref}(k+i,k)$ and $u(k+i,k)$ are, respectively, the predicted state, the reference state, and the predicted control at time $k+i$ based on the information available at time *k* and the global system model:

$$x(k+1) = Ax(k) + Bu(k),$$
$$y(k) = Cx(k), \qquad 2$$

where $u \in R^{n_u}$, $x \in R^{n_x}$ and $y \in R^{n_y}$.

The centralised MPC control problem, described by (1) can be converted into a decentralised MPC control problem (Giovanini *et al.*, 2007) by decomposing the system into a number of smaller-scale MPC control problems that are subsequently solved in an iterative manner. Necessary for this approach is the existence of a communication network over which the decentralised MPC controller can communicate and coordinate their actions. The following assumptions are also necessary:

1. The local states of each subsystem $x_l(k)$ are accessible,
2. The controller agents communication is synchronous,
3. Controllers can communicate several times within a sampling time interval.

This set of assumptions is not restrictive. In fact, if the local states are not accessible they can be estimated from local outputs $y_l(k)$ and the control inputs using a Kalman filter. The assumptions 2 and 3 are also valid since in most control problems, the sampling time interval is longer than the computational and the communication times.

The first step in the development of a decentralised MPC control is to approximate the cost function as,

$$J(X(k),U(k)) \approx \sum_{l=1}^{n_a} \beta_l J_l\left(X(k),U_{i\in\mathcal{L}_l}(k),U_{i\in\mathcal{I}-\mathcal{L}_l}(k)\right) \qquad 3$$

where $\beta_1 \cdots \beta_{n_a}$ is a set of weights that satisfy $\beta_l \ge 0$. The notation $U_i(k)$ denotes $U_i(k) = [u_i(k,k)\cdots u_i(k+n_c,k)]^T$, where the notation $u_i(k)$ refers to the i[th] system input. The set $\mathcal{I}$ is assumed to be such that $U(k) = U_{i\in\mathcal{I}}(k)$ whilst the intersections of the sub-sets $\mathcal{L}_1 \cdots \mathcal{L}_{n_a} \in \mathcal{I}$ are assumed to be empty sets. Note that (3) correspond to a multi-objective description of the original problem (Chankong and Haimes, 1983), where the weights β_l are employed to define the influence of each local performance index J_l on the optimization problem. Using the approximations from above n_a coupled optimisation problems are formed,

$$\min_{U_{i\in\mathcal{L}_l}(k)} J_l\left(X(k),U_{i\in\mathcal{L}_l}(k),U_{i\in\mathcal{I}-\mathcal{L}_l}(k)\right) \quad l = 1,\ldots,n_a$$

$$\text{s..t. } X(k) = Gx(k) + HU(k) \text{ and } T_l U_{i\in\mathcal{L}_l}(k) \le b_l \qquad 4$$

in which $U_{i\in\mathcal{I}-\mathcal{L}_l}(k)$ are assumed known. The optimisation problem has now been transformed into a dynamic game of n_a agents, where each agent searches for optimal decisions in response to the decision of others. Therefore, the behaviour of the global system will emerge from the iterative solution of the n_a coupled optimization problems. The goals of the decomposition are to reduce the complexity of the optimization problem by ensuring that each sub-problem in (4) is smaller than the overall problem. The price paid to simplify the optimisation problem is the requirement to calculate the interconnection between the sub-problems, which can be solved by means of *Nash optimality* concept (Nash, 1951).

Definition 1: A group of control decisions $U(k) = \{U_{i\in\mathcal{L}_1}(k)\cdots U_{i\in\mathcal{L}_{n_a}}(k)\}$ is said to be Nash optimal if

$$J_l\left(X(k),U^q_{i\in\mathcal{L}_l}(k),U^q_{i\in\mathcal{I}-\mathcal{L}_l}(k)\right) \le J_l\left(X(k),U^{q-1}_{i\in\mathcal{L}_l}(k),U^{q-1}_{i\in\mathcal{I}-\mathcal{L}_l}(k)\right) \quad l = 1,\ldots,n_a \qquad 5$$

If the Nash optimal solution is achieved, each sub problem does not change its control decision $U_{i\in\mathcal{L}_l}(k)$ because it has achieved the locally optimal objective under the above conditions; otherwise, the local performance index will degrade. Each subsystem optimizes its objective function using its own control decision assuming that other subsystems' solutions are known and optimal. So, if condition is satisfied, the whole system has arrived at an equilibrium point (attractor) in the coupling decision process.

Since the mutual communication and the information exchange are adequately taken into account, each subsystem solves its local optimization problem if the other subsystems' optimal solutions are known. Then, each agent compares the newly optimal solution with that obtained in the previous iteration and checks if the terminal condition is satisfied. If the algorithm is convergent, all the terminal conditions of the n_a agents will be satisfied, and the whole system will arrive at Nash equilibrium at this time. The sub-problems (4) can be solved using an iterative algorithm. As the controllers do the best for themselves by solving their problems in response to the decisions of the others, their iterative search for solutions give rise to a dynamic game. Therefore, in this scenario two fundamental questions naturally arise: the behaviour of each agent's iteration process during the negotiation process and the location and number of attractors of the decentralized problem.

3. CONVERGENCE ANALYSIS

To study the convergence properties of the proposed method, the centralised MPC problem is written in a form where the equality constraints have been eliminated, i.e

$$\min_{U(k)} \frac{1}{2} U(k)^T M U(k) + f^T U(k)$$

$$\text{s.t. } TU(k) \le b \qquad 6$$

wehre $M = \left(H^T Q H + R\right)$ and $f^T = -(X(k)_{ref} - Gx(k))^T QG$.

Now, in order to ensure that the proposed decentralised algorithm converges to the solution of the centralised problem, defined by (13), each of the sub-problems should inherit the following structure,

$$\begin{aligned}&\min_{U_{i\in\mathcal{L}_l}(k)} \frac{1}{2} U_{i\in\mathcal{L}_l}{}^T M_{\{i\in\mathcal{L}_l, j\in\mathcal{L}_l\}} U_{i\in\mathcal{L}_l} + \alpha f^T_{\{i\in\mathcal{L}_l\}} U_{i\in\mathcal{L}_l} + \alpha U_{i\in\mathcal{I}-\mathcal{L}_l}{}^T M_{\{i\in\mathcal{I}-\mathcal{L}_l, j\in\mathcal{L}_l\}} U_{i\in\mathcal{L}_l}\\ &\text{s.t. } T_{\{i\in\mathcal{K}_l, j\in\mathcal{L}_l\}} U_{i\in\mathcal{L}_l}(k) \le b_{\{i\in\mathcal{K}_l\}} \text{ for } l = 1,\ldots,n_a\end{aligned} \qquad 7$$

where the time index k is dropped for clarity of presentation and the matrix-vector notation $A_{\{i\in\mathcal{M}, j\in\mathcal{N}\}}$ denotes a matrix formed from the $\mathcal{M}$ rows and $\mathcal{N}$ columns of the general matrix A and where the parameter α is a positive scalar. Note that the constrained optimal solution to the sub-problems satisfies the following first order optimality conditions,

$$\begin{aligned}U_{i\in\mathcal{L}_l}(k) &= -\left(M_{\{i\in\mathcal{L}_l, j\in\mathcal{L}_l\}}\right)^{-1}\left(\alpha f_{\{i\in\mathcal{L}_l\}} + \alpha M^T_{\{i\in\mathcal{I}-\mathcal{L}_l, j\in\mathcal{L}_l\}} U_{i\in\mathcal{I}-\mathcal{L}_l}(k) + \alpha T^T_{\{i\in\mathcal{K}_l, j\in\mathcal{L}_l\}} \lambda_l\right),\\ 0 &= \Lambda_l\left(T_{\{i\in\mathcal{K}_l, j\in\mathcal{L}_l\}} U_{i\in\mathcal{L}_l}(k) - b_{\{i\in\mathcal{K}_l\}}\right) \text{ for } l = 1,\ldots,n_a\end{aligned} \qquad 8$$

where Λ_l is a diagonal representation of the vector λ_l, and where λ_l is a set of Lagrange multipliers. Using (8), the proposed iterative algorithm can be expressed in terms of a centralised constrained difference equation,

$$\begin{aligned}U^q(k) &= -\tilde{M}^{-1}\left(\alpha f + \left(\alpha M - \tilde{M}\right) U^{q-1}(k) + \alpha T^T \lambda^q\right),\\ 0 &= \Lambda^q\left(TU^q(k) - b\right)\end{aligned} \qquad 9$$

where matrix $\tilde{M}$ is a block-diagonal matrix of matrices. The sub-problems are unconstrained, i.e. if the matrix T and the vector b are empty, it is trivial to show that the decentralised algorithm, expressed by (16), converges to the solution of the centralised MPC problem. This follows by rewriting as:

$$U^q(k) = -\tilde{M}^{-1}\left(\alpha f + \left(\alpha M - \tilde{M}\right) U^{q-1}(k)\right) = (I - \alpha\tilde{M}^{-1}M) U^{q-1}(k) - \alpha\tilde{M}^{-1} f \qquad 10$$

Thus by making the scalar parameter α sufficiently small the system converges to a steady state condition, given by,

$$U^{ss}(k) = -M^{-1} f \qquad 11$$

which corresponds to the unconstrained solution of (1).

In the presence of constraints, it is slightly more complicated to derive the conditions under which the decentralised algorithm converges to the solution of the centralised MPC problem. In order to show that the algorithm indeed converge it will be shown that each iteration decreases the value of the global cost function, i.e. that the iteration satisfies the following condition,

$$\left(\frac{1}{2} U^{qT} M U^q + f^T U^q\right) - \left(\frac{1}{2} U^{q-1T} M U^{q-1} + f^T U^{q-1}\right) \le 0 \qquad 12$$

By substituting $U^q(k)$ in (9), the following expression can be obtained:

$$\begin{aligned}&\left(\frac{1}{2} U^{qT} M U^q + f^T U^q\right) - \left(\frac{1}{2} U^{q-1T} M U^{q-1} + f^T U^{q-1}\right) =\\ &-\alpha\frac{1}{2}\left(MU^{q-1} + f + T^T\lambda^q\right)^T M_1\left(MU^{q-1} + f + T^T\lambda^q\right) - \alpha\frac{1}{2}\left(MU^{q-1} + f\right)^T \tilde{M}^{-1}\left(MU^{q-1} + f\right)\\ &+\alpha\frac{1}{2}\left(T^T\lambda^q\right)^T \tilde{M}^{-1}\left(T^T\lambda^q\right)\end{aligned} \qquad 13$$

where

$$M_1 = \tilde{M}^{-1} - \alpha\tilde{M}^{-1} M \tilde{M}^{-1} = \tilde{M}^{-1/2}\left(I - \alpha\tilde{M}^{-1/2} M \tilde{M}^{-1/2}\right)\tilde{M}^{-1/2}.$$

Consequently, if the parameter α is made sufficiently small such that the matrix $M_1 \succ 0$ and if the term $\alpha 1/2\left(T^T\lambda^q\right)^T \tilde{M}^{-1}\left(T^T\lambda^q\right)$ is smaller than the other right hand-side terms the whole of the right-hand side of (13) evaluates to a negative value, which means that the global cost-function decreases at each iteration. To prove that the term $\alpha 1/2\left(T^T\lambda^q\right)^T \tilde{M}^{-1}\left(T^T\lambda^q\right)$ is smaller than the other right-hand side terms the n_a decentralised problems are expressed as a centralised problem,

$$\min_{U^q(k)} \frac{1}{2} U^q(k)^T \tilde{M} U^q(k) + \alpha f^T U^q(k) + \alpha U^{q-1}(k)^T (\alpha M - \tilde{M}) U^q(k)$$

$$\text{s.t. } TU^q(k) \le b \qquad 14$$

The solution to the problem described by (14) is given by (7), which when inserted into the cost-function gives the following optimal cost value,

$$\begin{aligned}&\min_{U^q(k)} \frac{1}{2} U^{qT} \tilde{M} U^q(k) + \alpha f^T U^q + \alpha U^{q-1T}(\alpha M - \tilde{M}) U^q =\\ &-\frac{1}{2}\left((\alpha M - \tilde{M}) U^q + \alpha f\right)^T \tilde{M}^{-1}\left((\alpha M - \tilde{M}) U^q + \alpha f\right) + \frac{1}{2}\alpha^2\left(T^T\lambda^q\right)^T \tilde{M}^{-1}\left(T^T\lambda^q\right)\end{aligned} \qquad 15$$

By assuming that $U^{q-1}(k)$ is feasible with respect to the constraints (which it is, provided that $U^0(k)$ is feasible) the optimal cost-function value at iteration q must be less than or equal to the value of the cost function when $U^q(k)$ is

sinc(*i*) Laboratory for Signals and Computational Intelligence (http://fich.unl.edu.ar/sinc)
L. Giovanini, J Balderud & R Katebi; "Distributed Control of Underwater Vehicles"
Proc. of the I MechE, Part M: J. of Engineering for the Maritime Enviroment, Special Issue on Marine Systems. Vol. 222, No. 2, pp. 95--107, 2008.

replaced by $U^{q-1}(k)$. This subsequently leads to the following inequality condition,

$$\begin{aligned}&\alpha^2\left(T^T\lambda^q\right)^T\tilde{M}^{-1}\left(T^T\lambda^q\right)\le\frac{1}{2}\left((\alpha M-\tilde{M})U^q+\alpha f\right)^T\tilde{M}^{-1}\left((\alpha M-\tilde{M})U^q+\alpha f\right)\\&+U^{q-1T}\tilde{M}U^{q-1}+U^{q-1T}\left((\alpha M-\tilde{M})U^{q-1}+\alpha f\right)\le\\&\left(\tilde{M}^{-1}(\alpha M-\tilde{M})U^q+\alpha\tilde{M}^{-1}f+U^q\right)^T\tilde{M}\left(\tilde{M}^{-1}(\alpha M-\tilde{M})U^q+\alpha\tilde{M}^{-1}f+U^q\right)\le\\&\alpha^2\left(MU^q+f\right)^T\tilde{M}^{-1}\left(MU^q+f\right)\end{aligned} \quad 16$$

Since, according to the above, the term $\left(T^T\lambda^q\right)^T\tilde{M}^{-1}\left(T^T\lambda^q\right)$ is always less than the term $\left(MU^{q-1}(k)+f\right)^T\tilde{M}^{-1}\left(MU^{q-1}(k)+f\right)$ it follows that the right hand side of (16) is negative definite. This, in turn, means that if the scalar parameter α is small enough the decentralised algorithm converges to the solution of the centralised MPC control problem.

4. COMMUNICATION FAILURES

The proposed controllers use the communication network to coordinate each other to accomplish the control objectives. To study this problem, the failures in the communication system are modelled introducing two matrices:

The *connection matrix* E, defined by

$$E=\left[e_{ij}\right],\quad i,j=1,\ldots,n_a \quad 17$$

which characterises the communication structure of the system. An element $e_{ij}=1$ indicates connection between *i* and *j* subsystems, while $e_{ij}=0$ shows no connection between these agents.

The *communication failure matrix* T, defined by

$$T=\left[t_{jj}\right],\quad j=1,\ldots,n_a \quad 18$$

models the communication failures. An element $t_{ij}=1$ corresponds to a perfect communication between agents *i* and *j*, while $t_{ij}=0$ corresponds to a communication failure between these agents. A failure between agents *i* and *j* at a given sample is represented with the transition from 1 →0 of the corresponding element t_{ij} of T.

With these preliminaries, the prediction model under the communication failure at time instant *k* can be rewritten as follows

$$X(k)=GETx(k)+HETU(k). \quad 19$$

Then, the control action of the entire system is given by

$$U^q(k)=K_0ETU^{q-1}(k)+K_1ETx(k)\quad q>1. \quad 20$$

At each sample, the system states $x(k)$ are known in advance and remain constant during the iterations, thus any change of the communication structure during the iterations does not modify this term. Then, the convergence of decentralised control scheme under the communication failure is determined by

$$\left\|\rho\left(K_0ET\right)\right\|_1\le 1.) \quad 21$$

Under communication failure, each agent cannot exchange information properly. In the extreme case ET = 0, the convergence condition (21) is always satisfied, and this situation corresponds to the fully decentralized control architecture where the convergence and stability depends on the structure selection.

Once the convergence of iterates can be guaranteed, the next issue to be addressed is the effect of the communication failures on the closed-loop stability. In order to establish the effect of communication failures on the closed-loop system, the control action is given by

$$U(k)=\left(I-K_0ET\right)^{-1}K_1ETx(k)=K_fx(k) \quad 22$$

leading to the closed-loop system

$$x(k+1)=\left(A-BIK_f\right)x(k) \quad 23$$

The stability of the closed-loop system under communication failures is determined by

$$\left|\mathrm{eig}\left(A-BI(I-K_0ET)^{-1}K_1ET\right)\right|<1. \quad 24$$

Under the communication failure, each agent cannot exchange information properly therefore, the stability of the closed-loop system will depend on the dynamic characteristics of the interactions between subsystems. In the extreme case ET = 0, the stability condition is always satisfied corresponding to the full decentralized architecture.

Finally, the effect of communication failures on the closed-loop performance is analysed. Under the communication failure, each controller cannot exchange information properly, leading to a deterioration of the closed-loop performance. The magnitude of the degradation depends on the effect of the failure on the system, which is given by

$$\tilde{J}(k)\le\left(1+\frac{\|W(k)\|}{\mathrm{eig}_m(F)}\right)J^*(k) \quad 25$$

where $J^*(k)$ is the optimal performance without failures, $\mathrm{eig}_m(F)$ denotes the minimal eigenvalue of

$$F=\left(K_1^{-1}\left(I-K_0\right)-H\right)^TQ\left(K_1^{-1}\left(I-K_0\right)-H\right)+R, \quad 26$$

and

$$\begin{aligned}&W(k)=S^T(k)\left(H^TQH+R\right)S(k),\\&S(k)=2I-\left[I+\left(I-K_0\right)^{-1}\left(I+K_0-2K_0ET\right)\right]^{-1}.\end{aligned} \quad 27$$

Proof. See (Giovanini and Balderud, 2006).

The magnitude of the closed-loop performance degradation is quantified by W(k) while an estimation of fault-free performance is provided by $\mathrm{eig}_m(F)$. In the extreme case

sinc(i) Laboratory for Signals and Computational Intelligence (http://fich.unl.edu.ar/sinc)
L. Giovanini, J Balderud & R Katebi; "Distributed Control of Underwater Vehicles"
Proc. of the I MechE, Part M: J. of Engineering for the Maritime Enviroment, Special Issue on Marine Systems. Vol. 222, No. 2, pp. 95--107, 2008.

where ET = 0, the resulting closed loop corresponds to the full decentralized control architecture, given by

$$J_{Dec}(k) = \left(1 + \frac{\| W_{\max} \|}{\mathrm{eig}_m(F)}\right) J^*(k) \qquad 28$$

where

$$W_{\max} = S_{\max}^T \left(H^T Q H + R\right) S_{\max}^T,$$
$$S_{\max} = 2I - \left[I + \left(I - K_0\right)^{-1}\left(I + K_0\right)\right]^{-1}. \qquad 29$$

5. SIMULATION AND RESULTS

An underwater towing application, which involves four UUVs that tows a non-actuated towing object along a predefined path, is considered in this simulation study. The control problem is to ensure that the actions of the UUVs are coordinated such that the towing object remains on the path.

By assuming that the towing forces generated by each UUV is linearly related to the towing energy and by denoting the towing forces generated by the i:th UUV, $(F_{i_x}(k), F_{i_y}(k), F_{i_z}(k))$, and the position of the towing object $(x(k), y(k), z(k))$, the above control problem can be stated in terms of a centralised optimal control problem,

$$\min_{U(k)} \ (S(k) - W(k))^T Q (S(k) - W(k)) + \Delta U(k)^T R \Delta U(k)$$
$$s.t. \ \ W(k) = Gw(k) + HU(k), TU(k) \le b \qquad 30$$

The matrices Q and R in (30) are symmetric and positive definite. The vector S(k) represents the reference path whilst the vector W(k) represents the predicted positions of the towing object. The vector w(k) denotes the dynamic states of the towing object at the discrete time-instant *k*. The equality constraints accounts for the dynamics of the towing object. The dynamic capabilities of the UUVs are accounted for by the second term of the value function and the inequality constraints. The dynamics of the UUVs are thus modelled as a set of value-function penalties and inequality constraints rather than as a set of equality constraints, which eliminates the need for a detailed dynamic model of each UUV. Note that since these constraints and penalties represents conservative estimates of the dynamic capabilities of the UUVs they will not be utilized to their full potential (for that, a nonlinear model is needed).

To solve the above optimal control problem a model that relates the towing forces, U(k), to the predicted positions of the towing object, W(k), is needed. It is assumed that the amplitude of the drag forces generated when the towing object ploughs through the water depends linearly on the velocity, $(\dot{x}(k), \dot{y}(k), \dot{z}(k))$. The direction of the drag forces is strictly opposite to velocity direction a model describing the motion dynamics of the towing object can be obtained by applying Newton's laws of acceleration and motion.

$$w(k+1) = \underbrace{(I + t_s \tilde{A})}_{A} w(k) + \underbrace{t_s \tilde{B}}_{B} u(k) + \underbrace{t_s \tilde{V}}_{V} v(k), \qquad 31$$

where t_s is the sample time and *v(k)* are a set of stochastic disturbances.

Once a discrete-time model representation of the system has been obtained, the model can be used to compute the constraint matrices G and H, then by replacing $W(k)$ in the value function with the terms on the right hand side of the equality constraints the following centralised optimal control problem is obtained,

$$\min_U \ (S - Gw - HU)^T Q (S - Gw - HU) + \Delta U^T R \Delta U$$
$$\text{s.t} \quad TU \le b \qquad 32$$

In this instance it is desired to solve the above optimal control problem in a decentralized manner such that the problem can be distributed between the UUVs. The centralized problem is therefore partitioned into four sub-problems, where the *j*:th sub-problem have the following structure,

$$\min_{U_j(k)} \ (S - Gw - \sum_{i=1}^{4} H_i U_i)^T Q (S - Gw - \sum_{i=1}^{4} H_i U_i) - \Delta U_j^T \tilde{R}_{jj} \Delta U_j +$$
$$+ 2\sum_{i=1}^{4} \Delta U_j^T \tilde{R}_{ji} \Delta U_i \qquad \text{s.t} \ \ T_j U \le b_j \qquad 33$$

where,

$$U_j(k) = \left[u_j(k,k) \cdots u_j(k + n_c, k)\right]^T,$$
$$\Delta U_j(k) = \left[u_j(k,k) - u_j(k-1,k) \cdots u_j(k+n_c,k) - u_j(k+n_c-1,k)\right]^T,$$
$$u_j(k,k) = \left[F_{j_x}(k,k), F_{j_y}(k,k), F_{j_z}(k,k)\right]^T$$

The notation, B_j, denote the columns of the matrix B that described the influence from $u_j(k)$ and similarly where the matrices $\tilde{R}_{ij}$ has been appropriately derived from the symmetric positive definite matrix R.

A series of simulations have been carried out using a normalised model parameters as follows,

$$Q = I, R = 100I, m = 15, d = 2, t_s = 1, n_c = 20, n_p = 60,$$
$$v(k) = \left[0.1, 0.2, -0.3\right]^T + \psi(k), -0.05 \le \Delta U(k) \le 0.05$$

Fig. 1 compares the trajectories computed by the proposed algorithm and a centralised algorithm. Two slightly different versions of the proposed algorithm have been employed. In one of the versions the algorithm is allowed to iterate indefinitely until the convergence criteria has been met, whilst in the other version iterations are disallowed.

The reference path starts at a shallow depth and then spirals down to a deeper depth were some manoeuvres are performed. The performance of the centralised and the decentralised algorithms are similar. All three solutions track the reference trajectory and compensate for the disturbances well. It is also worth noticing that when limiting the number of iterations in the decentralised algorithm the performance of the control solution is only marginally affected. Fig. 2 shows snapshots (every 20th sample) of the direction and amplitude of the towing forces computed by the decentralised algorithm. Fig. 3 shows the number of iterations required by the decentralised algorithm to converge to the centralised solution. The number of iterations remains reasonably steady at approximately 20 iterations per sample. This number depends weakly on the length of the prediction and control horizon and strongly on the precision to which the control actions are determined. By accepting lower precision, the number of iterations required to reach a steady state solution can be drastically decreased.

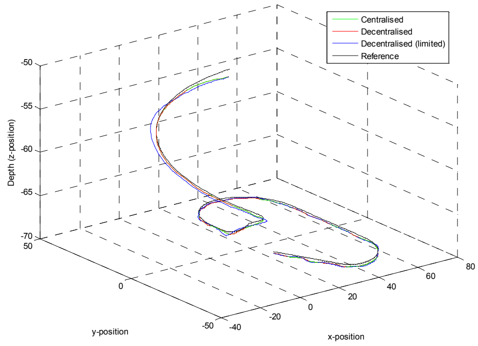

Fig 1: A 3-D view of a performance comparison between the centralised and decentralised MPC

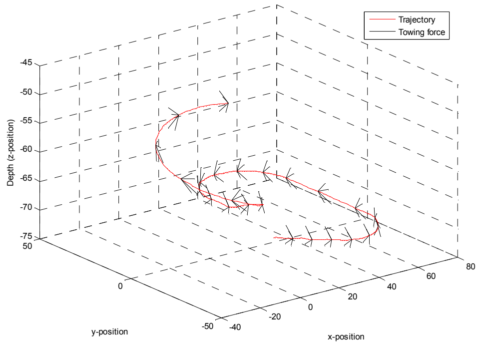

Fig 2: A 3-D view of the towing forces computed by the decentralised algorithm.

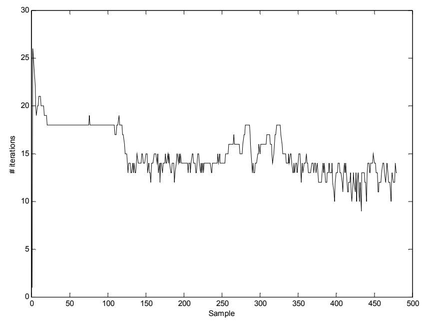

Fig 3: The number of iterations at each sample to converge to centralised MPC solution.

6. CONCLUSIONS

The main advantage of the proposed scheme is that the on-line optimization of a large-scale multi-agent system can be converted to that of several small-scale systems, thus can significantly reduce the computational complexity while keeping satisfactory performance. The method is also capable of handling communication delays and failure. The design parameters for each agent can tuned separately, which provides more flexibility for the analysis and applications. The convergence, stability and performance of the distributed control scheme provide a better understanding of the algorithm.

Acknowledgements

The authors are grateful for the financial support for this work provided by the Engineering and Physical Science Research Council (EPSRC) Platform Grant Industrial Non-linear Control and Applications GR/R04683/01.

REFERENCES

Campo P. and M. Morari, “Achievable closed-loop properties of systems under decentralized control”, *IEEE* Trans. *on Automatic Control*, vol. 39(5), 932–943, 1994.

Camponogara E., E. Jia, B. Krogh and S. Talukdar, “Distributed model predictive control”, IEEE Control System Magazine, pp. 44-52, 2002.

Chankong V. and Y. Haimes, *Multiobjective Decision Making: Theory and Methodology*, Elsevier Science Publishing, 1983.

Conley A. and M. Salgado, “Gramian based interaction measure”, in *Proc. of Conference on Decision and Control*, Australia, pp. 5020–5022, 2000.

Dubey P. and J. Rogawski, “Inefficiency of smooth market mechanisms”, *J. Math. Econ.*, vol. 19, 285–304, 1990.

Johansson K. and A. Rantzer, “Decentralized control of sequentially minimum phase systems”, *IEEE Tran.s on AC*, vol. 44(10), 1909–1913, 1999.

Jia D. and B. Krogh, “Distributed model predictive control,” in *Proc. ACC*, Arlington, pp. 2767-2772, 2001.

Jia D. and B. Krogh, “Min-max feedback model predictive control for distributed control with communications”, in *Proc. ACC*, Anchorage, pp. 4507-4512, 2002.

Keviczky T., F. Borrell and G. Balas, “Model Predictive Control for Decoupled Systems: A Study on Decentralized Schemes,” in *Proc. ACC conf.*, 2004.

Lee J. and T. Edgar, Interaction measure for decentralized control of multivariable processes, in *Proc. of ACCl Confe*, Anchorage, pp. 454–458, 2002.

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

Oschs S., S. Engell and A. Draeger, “Decentralized vs. model predictive control of an industrial glass tube manufacturing process,” in *Proc. 1998 IEEE Int. Conf. Control Applications*, Italy, pp. 16-20, 1998.

Giovanini L. and J. Balderud, "Decentralized control via Nash bargaining", International Control Conference, Glasgow, 2006.

Giovanini L., J. Balderud and R. Katebi, “Autonomous and Decentralized Mission Planning for Clusters of UUVs”, International Journal of Control, in press, 2007.