Kinematic Simulative Analysis of Virtual Potential Field Method for AUV Trajectory Planning

M Barisic[*], Z. Vukic[*] and N. Miskovic[*]

matko.barisic@fer.hr, zoran.vukic@fer.hr, nikola.miskovic@fer.hr

[*] University of Zagreb, Faculty of Electrical Engineering and Computing, Department of Computer and Control Engineering, Laboratory for Underwater Systems and Technologies, Zagreb, Croatia

Abstract—**This paper deals with an analysis of applicability, capabilities, benefits and pitfalls of using a virtual potential field approach to autonomously planning trajectories in non-communicating autonomous underwater vehicles (AUV-s). Virtual potentials represent an approach to this problem with cross-layer design features. Examples of different layers of control that can be achieved with the same fundamental approach are: obstacle-avoidance, energy-optimal trajectories, forming up with other moving agents, controlled formation fragmentation into well-posed sub-formation etc. This paper shows, on the basis of extensive simulated experiments, that such a trajectory planner based on virtual potentials, guarantees good extendibility, scalability and performance in a hard-real-time hardware-in-the-loop system.**

I. Introduction

Trajectory planning for autonomous underwater vehicles (AUV) is a prolific field of research. Challenges arising in this field arise both from the typical engineering and the constraints it poses on the capabilities of individual AUVs, and by the features of the environment the AUV executes its missions in. Typical trajectory-planning problems are further escalated when, rather than a single AUV, a need arises to compute in an online fashion trajectories of multiple AUVs cooperating in the same theater of operations. A key feature of such a submerged theater of operations is the lack of reliable, high-bandwidth communication. This significantly limits the methodologies to be used. Also, it imposes severe practical, implementation-wise constraints on the algorithms implemented inside a single AUV that statistically arrive at some estimates of the state of other AUVs from sensory observation of their action in the theater. Since no communication can take place between any single AUV and a supervisory command, control and communication center that would centralize the total information necessary for optimal trajectory planning, the autonomy of a trajectory planning method implemented algorithmically aboard every single AUV must be complete.

As stated, apart from self-observation and measurement of internal AUV's states, in such a scenario there is need for measurement and observation from which the internal states of cooperating AUVs can be deduced. In addition to that, the process of deduction itself usually consists of high-processor-commitment operations: nonlinear filtering of, applying transforms to, and performing regression and classification on signals arriving from slow-refresh-rate sensors. Therefore the trajectory planner algorithm must take into account and manage (i.e. by multithreading and multitasking) the synchronicity between low-processor-commitment trajectory calculations, high-processor-commitment feature extraction and no-processor-commitment but large sample time sensing.

Section 2 explains how the virtual potential method was implemented in order to deal, in what the authors think is, an optimal approach, with the stated conditions, constraints, problems and features. Section 3 presents the results of computer-run simulations of this setup. For purposes of proving the concept, and without influencing the implementation in mind for this method in its final embedded form, the problem space is constrained to 2D. Through inspection of simulation and follow-up redesign of the method, stability problems are resolved and local minimum avoidance, obstacle- and collision-avoidance, and a limited amount of formation behavior are achieved. The constraint of the problem space to 2D doesn't impact the applicability of the algorithms arrived at in this section when the predominant modes of usage and mission profiles of actual AUVs are taken into account. These consist of craft being given navigation tasks at either a constant depth, or with depth controlled by a separate decoupled control loop altogether. Section 4 gives closing comments, plans for further research and surmises our findings.

II. The Virtual Potentials Method

The design of a successful trajectory planner implemented on non-communicating AUVs that might at some point in time be applied to coordinated multi-vehicular missions must obey the following requirements:

1. stability (finiteness of the planned trajectory in the time and space domains),

2. propulsion energy-optimality or parsimony,
3. autonomity, without presupposing communication of internal states between coordinated AUVs, but rather reliance on outside observation of their actions in the theater of operations,
4. explicit addressing, avoidance or resolution of synchronization issues between measurement and feature extraction off of the dominant sensor, and the control algorithm in the code implementation of the algorithm

Currently pursued research efforts of the global control engineering community fall predominantly (with significant amounts of overlap and ambiguity) into the following methodologies:

1. Approaches exploiting graph theory, focusing on issues of formational stability, addressed in work such as [1], [2], [3], [4].
2. Virtual potential method approaches, pursued in [5], [6], [7], [8].
3. Iterative methods in the broadest and most general sense, based on a plethora of techniques (receding horizon MPC, mixed integer programming, dynamic programming, or simulation of state machines, to name but the few), covered extensively in [9], [10], [11], [12], [13].

The authors have also considered the actual experimental setup based on the relatively low-cost VideoRay Pro II ROV submersible converted into an AUV setup. Taking into account all of the above, a method, designed from ground up, from the class of virtual potential-based approaches was chosen. In addition to fulfilling all the stated design requirements, virtual potentials-based methods, such as the one described in this paper, also have the following advantages:

1. The method is intuitive and easily understood even by undergraduate control engineering students,
2. The method is at the worst border-line stable (i.e. proof of BIBO stability is trivial), since it is based on physical processes, thus obeying the 2nd Law of Thermodynamics,
3. The method is formally propulsion energy-optimal (in the worst case parsimonious rather than optimal due to implementation-driven trade-offs and caveats), due to the same reason stated in 2,
4. The method possesses a large number of method-specific independently settable parameters; These parameters' values do not arise from physical constraints on the trajectory planning problem or the nature and engineering of the AUV used; Methods that have this characteristic usually are a good foundation for this type of research, exploring optimality,
5. The method has cross-layer-design features; This allows a large set of behaviors to be implicitly programmed in (even at a later point, by manipulating parameters and data, rather than the algorithm) without large coding overhead or programmatic hybridization of the implementation code,
6. The method lends itself naturally to object-oriented programming implementation; The method is encapsulated into human-readable, transparent and therefore easily extendible code,
7. The method scales well and its complexity behaves orderly. The complexity increases linearly with the addition of additional agents or obstacles, and geometrically with the addition of a dimension of the problem space, both in terms of the number of sub-function calls and processing time.

The method itself is based on determining the gradient of a virtual potential field. The potential field subjected to the numerical calculation of the gradient in the vicinity of the AUV is given by adding influences of all perceived features of the theater of operations. The comprising potential levels are in turn given or dictated as evaluations of *potential distribution functions* (PDFs) attributed to every type of feature – different classes of obstacles, other agents, goal- or way-point. The merits and construction of this method were first proposed in [8]. The definitions of PDFs for every class of obstacle arising from classification of the observed theater of operations are implemented locally as a library of functions aboard an AUV. The method itself is described by (1 – 7).

$$E\left(\bar{p},k\right)=\sum_{i} f_{obj(i)}\left(\bar{p}\left(k\right)\right) \qquad (1)$$

Where:
- E is the virtual potential,
- $f_{obj(i)}$ are the PDFs, functions defining potentials over the (x, y) vector-space, arising due to the existence of all collision-critical, motion-relevant objects in the sensed portion of the theater of operations,
- $\bar{p}\left(k\right)=\left(x\left(k\right),y\left(k\right)\right)$ is the position vector at which the potential E is evaluated at time index k. Subsequently subscript indices to $\bar{p}$ will designate more specific points in the (x, y) vector-space.

$$\vec{F}\left(k\right)=\left\lceil \max_{i}\left[E\left(\bar{p}_{AUV}\left(k\right)\right)-E\left(\bar{p}_{\varepsilon i}\left(k\right)\right)\right]\right\rceil^{F_{max}} \angle \left(\arg\max_{i}\left[E\left(\bar{p}_{AUV}\left(k\right)\right)-E\left(\bar{p}_{\varepsilon i}\left(k\right)\right)\right]\cdot\gamma\right) \qquad (2)$$

Where:
- $\vec{F}\left(k\right)$ is the directional controlling force reproducing the trajectory at time index k,
- $\bar{p}_{AUV}\left(k\right)$ is the position vector of the AUV at time index k,
- $\bar{p}_{\varepsilon i}\left(k\right)$ is the position vector of the i-th out of the total of n_{γ} equally radially distributed sample points in the ε-vicinity of the AUV, defined in (3) and (4).

- $\lceil \cdot \rceil^{F_{max}}$ is the upper limit of the $\vec{F}(k)$'s modulus to F_{max}, an implementation-specific parameter (depending on the AUV). To facilitate clarity, heretofore the operation of applying the upper limit on the modulus of a vector $\vec{v}$ will be written as $bound(\vec{v}(k), v_{max})$

$$\vec{p}_{\varepsilon i}(k) = \vec{p}_{AUV}(k) + \varepsilon_{AUV} \cdot \begin{bmatrix} \cos\gamma \\ \sin\gamma \end{bmatrix} \tag{3}$$

$$\gamma = \frac{2\pi}{n_\gamma} \tag{4}$$

Where:
- ε_{AUV} is the *spatial resolution*, also called *granularity* of the numerical sampling of the gradient of $E(x,y)$, and is a method-specific independent parameter,
- n_γ, the *angular resolution*, is a method-specific independent parameter.

The $\vec{v}(k)$, trajectory set-velocity for the time instance k, is arrived at by using the bilinear numerical integration of (2):

$$\vec{v}(k) = bound\left(\frac{T}{2}\left(\vec{F}(k) + \vec{F}(k-1)\right) + \vec{v}(k-1), v_{max}\right) \tag{5}$$

Where:
- v_{max} is an implementation-specific (AUV dependent) parameter

The control inputs of "set forward speed" v_{set} and "set course" ψ_{set} are thus easily as the argument and modulus of (5):

$$\psi_{set}(k) = \arg(\vec{v}(k)) \tag{6}$$

$$v_{set}(k) = |\vec{v}(k)| \tag{7}$$

This, tied into a closed time-loop form (a programmatic loop) completes the kinematic level of the algorithm. In addition to the latter, the final control law or algorithm must also include feedbacks or updates to the dataset representing the trajectory-planning relevant model of the theater of operations, and a schema of addressing the dynamic modes and behaviors of the craft itself. The former must in the case of virtual potential method occur through some type of nonlinear or stochastic filtering or other feature extraction techniques performed on sensed signals. Such an algorithm ensures dynamic, reactive trajectory planning for an AUV in a nondeterministic, unstructured, dynamically changing environment.

However, significant exploration of performance of the kinematic level of the algorithm must be performed to ensure the optimization of the values of the many introduced method-specific independently settable programmatic parameters (the $A^{\pm}$, n_γ, ε_{AUV} etc). Only when optimality and meeting of all formal requirements is assured at the kinematic level, further modifications, caveats and trade-offs needed to compensate or subsume non-ideal or non-linear craft dynamics can be implemented with any hopes of maintaining satisfactory performance of the algorithm.

For simulation purposes, the position of the AUV in the following sample time is evaluated by numeric bilinear integration of the set velocity vector, in equation 8.

$$\vec{p}_{AUV}(k+1) = \frac{T}{2}\left(\vec{v}(k) + \vec{v}(k-1)\right) + \vec{p}_{AUV}(k-1) \tag{8}$$

The theater of operations is represented as a set of a varied number of objects of one of three classes, each represented by a distinctive PDF:

1. a rectangular obstacle PDF, f_{orth}

$$f_{orth}(\vec{p}) = e^{\frac{A^+}{r(\vec{p})^2}} - 1 \tag{9}$$

$$r(\vec{p}) = \sqrt{disc(\vec{p})^T \left(\left| \cdot \mathbf{R}(-\phi)\,\vec{p} - \vec{p}_{cen} \cdot \right| - \begin{bmatrix} a \\ b \end{bmatrix} \right)^{\cdot 2}} \tag{10}$$

$$disc(\vec{p}) = \left| \cdot \mathbf{R}(-\phi)\,\vec{p} - \vec{p}_{cen} \cdot \right| \cdot\!> \begin{bmatrix} a \\ b \end{bmatrix} \tag{11}$$

Where:
- A^+ is the *repulsiveness* of the obstacle (positive potentials are attributed to repulsive action), a method-specific independent parameter,
- **R(·)** is the matrix executing the rotation, in two dimensions, about the origin of the coordinatae system, of the vector to the right of it,
- ϕ is the (approximate) actual sensed angle of rotation of the detected orthogonal obstacle measured from the positive x-semiaxis of the global coordinate system of the simulation,
- $\vec{p}_{cen}$ is the (approximate) actual sensed vector of coordinates of the obstacle's center,
- the $|\cdot \ldots \cdot|$ signifies element-wise absolute value,
- the $^{\cdot n}$ is the element-wise n-th power.
- $\cdot>$ is the element-wise pseudo-logical operation >, mapping to the $\mathbf{N_0}$ subset {0, 1} by assuming {$\bot \equiv 0$, $\top \equiv 1$}
- a, b are (approximate) actual sensed semi-dimensions along the x- and y-axes of the obstacle-centric coordinate system.

2. a circular obstacle PDF, f_{circ}

$$f_{circ}(\vec{p}) = e^{\frac{A^+}{\left(\|\vec{p}-\vec{p}_{cen}\| - r_0\right)^2}} - 1 \tag{12}$$

Where:
- r_0 is the radius of the obstacle.

3. The goal-point PDF, f_{GP}

$$f_{GP}\left(\vec{p}\right) = -A^{-} \cdot \mathrm{e}^{-\frac{\left|\vec{p}-\vec{p}_{GP}\right|^{2}}{2\sigma^{2}}} \qquad (13)$$

Where:

- $\vec{p}_{GP}$ are the coordinates of the goal-point
- σ is the *attractor reach* of the goal-point, a method-specific independent parameter.
- A^{-} is the *attractiveness* of the goal-point, a method-specific independent parameter.

An example of the repulsive potential present in the theater of operations on account of an obstacle is given in figure 1, featuring a rectangular obstacle.

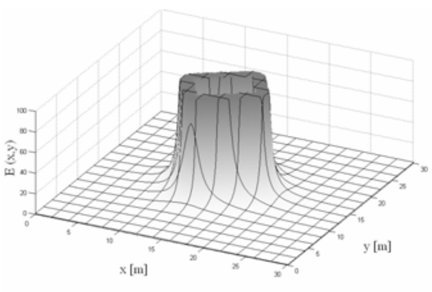

Figure 1: Example of a rectangular obstacle

III. Qualitative Assessment of the Trajectory Planner and Necessary Modifications

Figure 2 gives insight into the limited (oscillatory) BIBO stability of the algorithm developed in (1 – 7) in section 2.

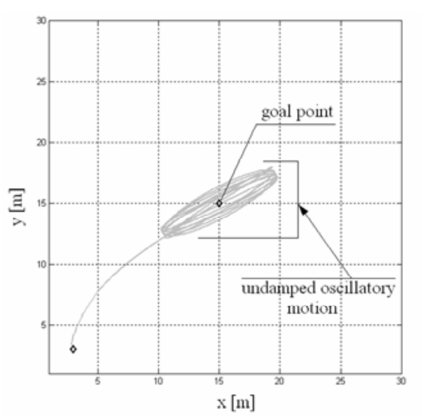

Figure 2: Oscillatory BIBO stability of the algorithm tested with 1 centrally positioned goal-point.

The precessive oscillation is present due to the lack of siphons for the overall kinetic energy in the system. In order to assure asymptotic static stability a revision including some mode of siphoning or degrading kynetic energy needs to be explicitly added to the algorithm. This is realized by introducing *virtual viscose friction* according to (14), thus modifying (2) to (15).

$$\vec{F}_{fric}\left(k\right) = \xi \cdot \left|\vec{v}\left(k-1\right)\right| \angle \; \pi + \arg\left(\vec{v}\left(k-1\right)\right) \qquad (14)$$

$$f_{GP}\left(\vec{p}\right) = -A^{-} \cdot \mathrm{e}^{-\frac{\left|\vec{p}-\vec{p}_{GP}\right|^{2}}{2\sigma^{2}}} \qquad (15)$$

Where:

- μ is the virtual viscose friction coefficient, a method-specific independent parameter.

A repeated setup of the stability test with μ = 0.4, produces asymptotically stable and finite trajectory planned according to figure 3.

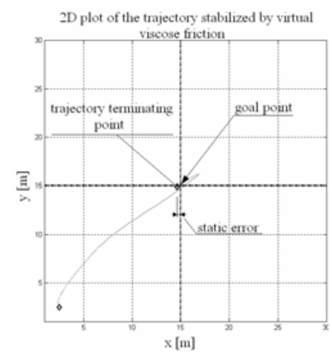

Figure 3: Asymptotic BIBO stability of the revised algorithm

A slight static error of positioning is introduced. However, since having a different control architecture for controlling most AUVs „in the small“ cannot be avoided, AUVs being nonlinear control objects, this is not a critical consideration. An example of a „small scale“ control architecture that can function with this „large scale“ trajectory planner is being developed by Mišković et al. [14].

An experiment with a moderately cluttered theater of operations was performed in order to test the algorithm for robustness in case of clutter. The simulation results are presented in figure 4.

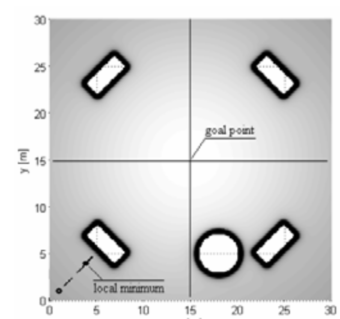

Figure 4: Testing the algorithm to cluttered operating conditions

The experiment demonstrates a flaw of virtual potential methods in general. The trajectory apriori-unpredictably terminates in a local minimum. Analytical treatment of this problem is, depending on the PDFs considered, and features of the dataset representing all potential influences, very difficult. Even numerical methods often lead to NP-hard computational problems or in the best-case scenario to algorithms that perform poorly in real-time. Therefore, the algorithm was extended with a hybridized logic scheme whereby agents are „preturbed out of" local minima.

In the algorithm finally developed the nature of the perturbation was a temporary „ghost" goal-point replacing for a given period of time the original operator-set or preprogrammed goal-point. This goal-point is reset by the AUV doing the planning of its own trajectory, much like the proverbial „carrot on a stick". The principle by which its position is injected into the AUV-local dataset describing the theater of operations follows from the position of the „masking" obstacle and the original goal-point according to figure 5. The mathematical description is given by (16) and (17).

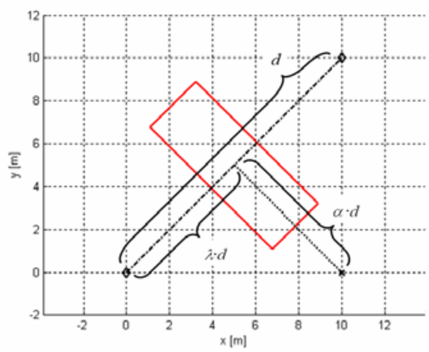

Figure 5: Schematic of the „ghost goal-point" method of circumnavigating local minima

The (x, y) coordinates of the „ghost" goal-point are calculated according to:

$$\vec{v} = \begin{bmatrix} y_{GP} - y \\ x_{GP} - x \end{bmatrix};\ d = \|\vec{v}\|;\ \tilde{\vec{v}} = \frac{\vec{v}}{d} \tag{16}$$

$$\vec{r}_{GGP} = \begin{bmatrix} x_{GGP} \\ y_{GGP} \end{bmatrix} = $$
$$= \lambda \cdot d \cdot \tilde{\vec{v}} + \alpha \cdot d \cdot \mathbf{R}\left(\frac{\pi \cdot (1+2\rho)}{2}\right) \cdot \tilde{\vec{v}} \tag{17}$$

Where:
- (x, y) is the current position of the AUV
- (x_{GP}, y_{GP}) is the global goal-point
- (x_{GGP}, y_{GGP}) is the „ghost" goal-point serving to perturb the AUV from the local minimum
- d is the distance $(x, y) - (x_{GP}, y_{GP})$
- λ is the tunable fraction (method-specific parameter) of the length d governing the position of the normal
- α is the tunable multiple (method-specific parameter) of the length d dictating the offset of the „ghost" goal-point from the connection $(x, y) - (x_{GP}, y_{GP})$
- ρ is the random binary value deciding the sidedness of the offset
- $\mathbf{R}(\phi)$ is the rotation of the position vector by obstacle pose angle ϕ
- $\tilde{\vec{v}}$ is the unit-vector in the direction from (x, y) to (x_{GP}, y_{GP})

The „ghost" goal-point is removed from the dataset once the AUV is within its ε_{ghost}-vicinity. In most realistically set up environments, an AUV will not again encounter a local minima travelling from (x, y) to (x_{GGP}, y_{GGP}). However, if it does, the process of adding „intermediary ghost" goal-points can be *stacked*.

Extending the trajectory-planning algorithm to include „ghost" goal-points, and running it in the same simulational setup displayed in figure 4 results in the trajectory planned below:

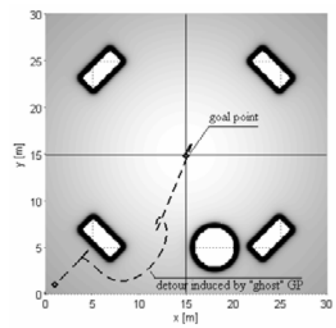

Figure 6: Recreated trajectory planning simulation in a relatively cluttered theater of operation with the local minima avoidance scheme implemented

Implementing this kinematic trajectory-planner on multiple simulated AUVs gives simulation results presented in figure 7. In this experiment, for each AUV planning its trajectory, other AUVs are classified as circular obstacles with $\{r_0 = 0.5, A^+ = 2.5\}$.

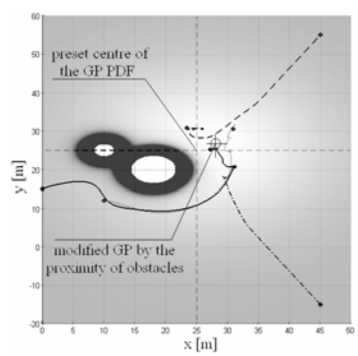

Figure 7: Simulation of clustering / forming up of multiple AUVs

It is plainly obvious that even without including explicit equations and modifications of the control law to include forming-up and clustering of the AUVs, a certain level of formation-like behavior is achieved. This is especially satisfactory when it is considered that from the point of view of code carried aboard an AUV, no further changes were necessary. Neither was the code of the algorithm changed or upgraded, nor was the library of PDFs expanded with additional or more complex functions.

However, it is also plainly obvious that the formations achieved are not robust. The regularity of the formation is not assured, but rather strongly influenced by the presence of obstacles near the common goal-point. This influences and distorts equipotentials of the f_{GP} PDF. Since algorithms aboard every AUV seek to uniformly distribute them at points equidistant to each other along a common equiopotential, distortion of equipotentials from circles results in non-uniform and/or irregular formations.

IV CONCLUSIONS AND FURTHER WORK

This paper gives a description of a kinematic algorithm for trajectory planning based on the numerical evaluation of the gradient of a virtual potential field. The paper provides insights critical to the actual implementation of a virtual potential method-based *dynamic* trajectory planner for AUVs, based on extensive simulations at the kinematic level. This planner is an example of an approach based on simple and easily encapsulated numerical mathematical calculations. The support for all mathematical calculations featured by the method is already coded into assembly-language level firmware in contemporary embedded processors. The stability and the energetic parsimony of the trajectories produced by the planner was demonstrated in simulations. A mechanism for reducing the number of cases in which possible occurring local minima might foil the trajectory planning is proposed and tested through simulation. Also, it is shown by simulation that the algorithm already features relatively non-robust clustering behavior when run in multiple AUVs within the same theater of operations. However, due to the lack of modifications to the data initializing the method the behavior is environmentally dependent and heavily relies on the structure of the theater of operations. However, it should be kept in mind that this behavior emerges without any communication of state between the algorithms run on separate platforms. Therefore, the authors believe that further exploring how the virtual potential method can be used to tack coordinated control problems holds decisive promise for the future.

Further work on the algorithm will be concentrated on several fronts. One is the actual software design. The design will have to tackle the specifics of the implementation in one of the high-level languages which are real-time-efficient and fast at both compile-time and run-time. In that respect, the decisive criterion for the implemented design approach will be hardware-in-the-loop operation. Thus, the algorithm here presented in simulational form will be able to control a real AUV (or an ROV run by a top-side computer).

Another advance is the exploration of possible extensions of PDF classes. Separate classes of PDFs need to be implemented, to be used to represent cooperating AUVs. These PDFs should be radially non-monotonic, i.e. should feature local minima at certain geometric configurations in the foreign agent-centric coordinate system (i.e. that of the other perceived AUV, not the local one). It is supposed that, based on equivalent problems in crystalline physics, a scheme supporting the appearance of formations robust to other features of the environment, is likely to emerge.

REFERENCES

[1] J. A. Fax and R.M. Murray, "Information Flow and Cooperative Control of Vehicle Formations", IEEE Transactions on Automatic Control, vol. 9, 2003, pp. 1465-1476

[2] M. Meshabi. "State-Dependent Graphs", Proceedings of the 42nd IEEE Conference on Decision and Control, vol. 3, 2003, pp. 3058-3063

[3] R. Olfati-Saber and R.M. Murray, "Graph Rigidity and Distributed Formation Stabilization of Multi-Vehicle Systems", Proceedings of the 41st IEEE Conference on Decision and Control, vol. 3, 2002, pp. 2965-2971

[4] R. Sepulchre, D. Paley and N. E. Leonard, "Graph Laplacian and Lyapunov Design of Collective Planar Motions", Proceedings of the International Symposium on Nonlinear Theory and Its Application, 2005, Proceedings on CD

[5] E. Fiorelli, "Multi-AUV Control and Adaptive Sampling in Monterey Bay", Proceedings of the IEEE Autonomous Underwater Vehicles 2004: Workshop on Multiple AUV Operations, 2004, pp. 134-147

[6] L. Moreau, R. Bachmeyer and N. E. Leonard, "Coordinated Gradient Descent: A Case Study of Langrangian Dynamics With Projected Gradient Information", Proceedings of the 2nd IFAC Workshop on Lagrangian and Hamiltonian Methods for Nonlinear Control, 2003, Proceedings on CD

[7] P. Ögren, E. Fiorelli and N. E. Leonard, "Cooperative Control of Mobile Sensor Networks: Adaptive Gradient Climbing in a Distributed Environment", IEEE Transactions on Automatic Control, vol. 9, 2003, pp. 1292-1302

[8] M. Barisic, Z. Vukic and N. Miskovic, "Design Of A Coordinated Control System For Marine Vehicles", Proceedings of the 7th IFAC Conference on Manoeuvring and Control of Marine Craft, 2006, Proceedings on CD

[9] R.W. Beard and T. W. McLain, "Multiple UAV Cooperative Search under Collision Avoidance and Limited Range Communication Constraints", Proceedings of the 42nd IEEE Conference on Decision and Control, 2003, pp. 25-30

[10] J.S. Bellingham et al., "Cooperative Path Planning for Multiple UAVs in Dynamic and Uncertain Environments", Proceedings of the 41st IEEE Conference on Decision and Control, 2002, pp. 2816-2822

[11] C.G. Cassandras and W. Li, "A Receding Horizon Approach for Solving Some Cooperative Control Problems", Proceedings of the 41st IEEE Conference on Decision and Control, 2002, pp. 3760-3765

[12] M.G. Earl and R. D'Andrea, "Modeling of a Multi-Agent System Using Mixed Integer Linear Programming", Proceedings of the 41st IEEE Conference on Decision and Control, 2002, pp. 107-111

[13] S. S. Stankovic, M. J. Stanojevic and D. D. Siljak, "Stochastic Inclusion Principle Applied to Decentralized Overlapping Suboptimal LQG Control of a Platoon of Vehicles", Proceedings of EUROCON, 2005, Proceedings on CD

[14] N. Miskovic, Z. Vukic, M. Barisic and B. Tovornik, "Autotuning Autopilots for Micro-ROVs", 14th Mediterranean Conference on Control and Automation, 2006, Proceedings on CD