A KINEMATIC VIRTUAL POTENTIALS TRAJECTORY PLANNER FOR AUV-S

Matko Barišić, Zoran Vukić, Nikola Mišković

University of Zagreb,
Faculty of Electrical Engineering and Computing,
Laboratory for Underwater Systems and Technologies

Abstract: This paper deals with trajectory planning for an autonomous, non-communicating submerged vehicle (AUV). Most maneuvering, esp. that related to obstacle avoidance, in a typical mission scenario for an AUV consists of motion at single submerged depth. Therefore a trajectory planning scheme operating in 2D has sufficient merit and applicability. A scheme for trajectory planning with cross-layer features, such as implicit inclusion of obstacle-avoidance and forming up with other moving agents, is developed in a simulated environment, at a kinematic level. The trajectory planner is based on virtual potentials, an approach that guarantees good extendibility, scalability and performance in a hard-real-time hardware-in-the-loop system.

Keywords: Autonomous mobile robots, Trajectory planning, Gradients, Robot kinematics, Robot navigation

1. INTRODUCTION

Trajectory planning for autonomous underwater vehicles (AUV) is a daunting challenge. Difficulties and constraints are posed by both the engineering typical of AUVs, and by the features of the environment. This problem is magnified when, rather than a single AUV, the control problem is extended to a group of AUVs. A key feature of a submerged theater of operations of such groups of AUVs is impossibility of reliable, high-bandwidth communication. This precludes any shortcuts, simplifications or any method relying on at least partial communication either between AUVs, or between an AUV and some form of a supervisory command, control and communication center. The autonomy of a trajectory planning method must be complete, strict and unequivocal.

Also, measurement for purposes of trajectory planning in AUVs relies on high-processor-commitment operations: nonlinear filtering, applying transforms, 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 and high-processor-commitment feature extraction.

Section 2 explains how the stated conditions, constraints, problems and features have influenced our choice of the trajectory planning method and elaborates on the virtual potential approach. Section 3 presents the simulation results for a virtual potential method trajectory planner with problem space constrained to 2D, wherein stability problems are resolved and local minimum avoidance, obstacle- and collision-avoidance, and a limited amount of formation behavior are achieved in terms of the certain settings and modifications of the virtual potential method described in Section 2. The constraint of the problem space to 2D doesn't theoretically flaw the arguments made, and is in turn consistent with predominant modes of usage and mission profiles of actual AUVs, where the craft are given navigation tasks at either a constant depth, or with depth controlled by a separate control loop altogether. Section 4 gives closing comments, plans for further research and surmises our findings.

2. THE VIRTUAL POTENTIALS METHOD

The design constraints for a trajectory planner implemented on AUVs are as follows:

1. The trajectory planner must be stable,
2. The trajectory planner must be optimal with respect to energy spent on propulsion,
3. The trajectory planner must be completely autonomous and rely exclusively or predominantly on measurement and sensing, rather than on communication,
4. The trajectory planner needs to be implemented in such a way as for the synchronization of measurement and feature extraction off of the primary environment sensor and the control signal calculation algorithm to be explicitly addressed, avoided or resolved.

Out of the large number of approaches to this problem, currently pursued venues of research can be loosely grouped (with significant amounts of overlap and ambiguity) into the following methodologies:

1. Graph-theoretical approaches, see (Fax and Murray, 2003; Meshabi, 2004; Olfati-Saber and Murray, 2002, Sepulchre et al., 2005),
2. Virtual potential method, see (Fiorelli et al. 2004; Mureau et al., 2003; Örgen et al., 2003; Barišić et al., 2006),
3. Iterative methods, based on receding horizon MPC, mixed integer programming, dynamic programming, or simulation of state machines – most notably in the field of coordinated control of a formation of unmanned aerial vehicles, see (Beard and McLain, 2003; Bellingham et al., 2002; Cassandras and Li, 2002; Earl and D'Andrea, 2002; Stanković, Stanojević and Šiljak, 2005).

Having in mind these options, the class of virtual potential-based methods, in addition to fulfilling all the stated prerequisites for a trajectory planner in a coordinated AUV, also has the following distinguishing characteristics:

1. The method is intuitive and easily understood by students and experts alike,
2. The method is inherently at least border-line stable (i.e. proof of BIBO stability is trivial), by being based on physical processes that obey the law of conservation of energy,
3. The method is optimal or in the worst case closely suboptimal in the sense of energy spent on maneuvering and guidance, due to the same reason stated in 2,
4. The method is very malleable and tunable through a large number of numerical parameters; These parameters' values do not arise from physical constraints on the trajectory planning problem; A method with a large number of independently and freely tunable method-specific parameters is a good choice for an experimental setup and exploration of optimality,
5. The method has cross-layer-design features, allowing a plethora of behaviors to be implicitly programmed in without large coding overhead or programmatic hybridization of the implementation code; Obstacle counter-navigation, collision avoidance, forming up and formation maintenance, controlled formation breakup into sub-formations etc. can all be achieved without significant additions or editions of the implementation code, but rather through the manipulation of method parameters or nigh-trivial redefinitions of class methods,
6. The method is well suited to object-oriented programming implementation and when considered with due care, can be encapsulated into easily readable, transparent and therefore easily extendible code,
7. The method is well scalable and behaves predictably and linearly in computational complexity and processing time.

The method itself is based on the addition of virtual potentials. These arise from evaluated *potential distribution functions* (PDFs) attributed to every environmental feature – obstacle, deemed influential in the trajectory planning sense. The merits and construction of this method were first proposed in (Barišić et al. 2006). Since the potentials are virtual their definitions (actually definitions of PDFs for a given class of obstacle) are implemented locally aboard an AUV. The method itself is described in equations 1 – 7.

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

Where:
- *E* is the virtual potential,
- $f_{obj(i)}$ are the PDFs, functions over the (*x*, *y*) vector-space that describe the contributions of all collision-critical, motion-relevant objects in the staging area to the potential field regulating the AUV's motion,
- $\vec{p}\left(k\right)=\left(x\left(k\right),y\left(k\right)\right)$ is the point at which the potential *E* is evaluated at time index *k*.

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

Where:
- $\vec{F}\left(k\right)$ is the directional controlling force reproducing the trajectory at time index *k*,
- $\vec{p}_{AUV}\left(k\right)=\left(x_{AUV},y_{AUV}\right)$ is the position vector of the AUV at time index *k*,
- $\vec{p}_{\varepsilon i}\left(k\right)$ are positions of the n_{γ} points in the ε-vicinity of the AUV, defined in eq. 3 and 4.
- $\lceil\cdot\rceil^{F_{\max}}$ represent limiting on the upper side the size of the vector $\vec{F}\left(k\right)$ to F_{max}, an implementation-

specific parameter (depending on the AUV). To facilitate clarity, heretofore the operation of limiting on the upper side the size (modulus) of a vector will be written as $bound\left(\vec{v}\left(k\right), v_{max}\right)$

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

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

Where:
- ε_{AUV} is the radius of some vicinity of *AUV* dictating the *spatial resolution* of numerical sampling of *E*, and is a method-specific independent parameter,
- n_γ, the *angular resolution*, is a method-specific independent parameter.

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

Where:
- $\vec{v}\left(k\right)$ is the calculated required velocity of the AUV at time index *k*
- v_{max} is an implementation-specific (AUV dependent) parameter

From eq. 5 the control inputs of "set forward speed" v_{set} and "set course" ψ_{set} are easily obtained in eq.-s 6 and 7.

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

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

Although the foreseen usage of this trajectory planner is in a real AUV, significant work in kinematic simulation the results of this trajectory planning scheme is necessary. The requirement for simulation-based analysis arises since there are a large number of freely tunable, purely method-specific numerical parameters. These need to be tuned and set up before this method can be coded and committed to controlling a real AUV. In the simulation scheme, dynamics of a real AUV are disregarded. The issue of work presented herein is to detect, resolve and tune the behavior of the method itself. This is instrumental in assuring that the method itself is optimally set up before further modifications, arising from the need to compensate for non-ideal or non-linear craft dynamics, are implemented. Therefore, the position of the AUV in the following sample time is evaluated by numeric integration of the set velocity vector, in equation 8.

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

As for the PDFs characterizing detected obstacles, these were defined as three distinct classes:

1. a rectangular obstacle PDF – f_{orth}

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

Where:
- A^+ is the repulsive ponder of the obstacle (positive potentials are attributed to repulsive action), a method-specific independent parameter,
- $r\left(\vec{p}\right)$, *the characteristic radius*, is defined by 10 and 11.

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

Where:
- **R(·)** is the matrix executing the rotation of the vector anteceding it (multiplying it from the right),
- ϕ is the angle of rotation of the orthogonal obstacle measured from the positive *x*-semiaxis of the global coordinate system of the simulation,
- $\vec{p}_{cen}$ is the vector of coordinates of the obstacle's center,
- the $\left|\cdot\ldots\cdot\right|$ operator is element-wise absolute value,
- the $^{\cdot n}$ is the element-wise *n*-th power.

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

Where:
- $\cdot>$ is the element-wise logical operation >, where numerical values are assumed according to the rule $\{\bot \equiv 0, \top \equiv 1\}$
- *a*, *b* are half-length and half-width, respectively, of the orthogonal obstacle (length being the girth along the nominal *x*-axis of the obstacle-centric coordinate system and width being the girth along the nominal *y*-axis)

2. a circular obstacle PDF, f_{circ}

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

Where:
- r_0 is the radius of the obstacle.

For purposes of having meaningful and goal-driven navigation, a fourth PDF class was introduced – that attributed to the desired goal-point of the AUV. The goal-point PDF, f_{GP}, is given in 13.

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

Where:
- $\vec{p}_{GP}$ are the coordinates of the goal-point
- σ is the reach of the goal-point, determining how far the attractive influence of the goal-point extends in

the theater of operations. This is a method-specific independent parameter.
- A^{-} is the attractive ponder of the goal-point, a method-specific independent parameter.

A detailed view of an example obstacle with its repulsive influence on potential levels surrounding it is given in fig. 1, featuring a rectangular obstacle.

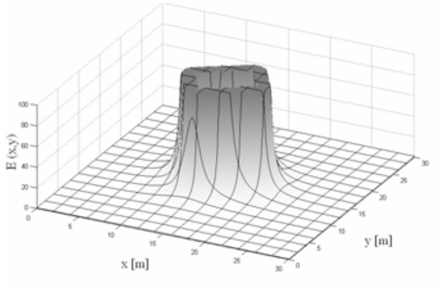

Figure 1: Example of a rectangular obstacle

3. QUALITATIVE ASSESSMENT OF THE TRAJECTORY PLANNER AND ADDED FUNCTIONALITIES

The basic stability test has been set up in the simulation environment, and produced results displayed in figure 2.

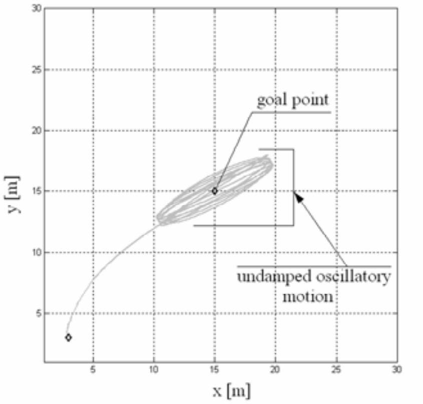

Figure 2: Basic stability test setup with 1 attractive goal-point positioned centrally

A critical lack of stability can be clearly seen. This is due to the fact that while the method is in principle energetically conservative there are no siphons for the overall kinetic energy in the system. In order to assure asymptotic static stability such an influence needs to be explicitly added to the algorithm. This is realized by the simulated action of viscose friction given by 14, influencing the controlling force, given in original, unmodified form in eq. 2, to produce eq. 15.

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

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

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

For μ = 0.4, the stability test has been recreated in figure 3.

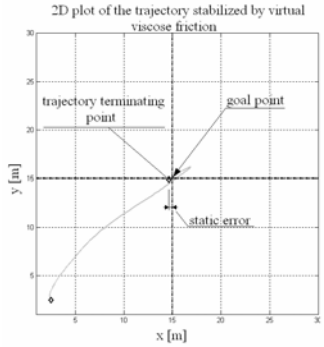

Figure 3: Recreated basic stability test

The introduction of the virtual viscose friction asymptotically stabilises the trajectory. However, a slight static error of positioning is introduced. However, due to real-world holonomic and arising under-actuation constraints of most AUVs, this is not a critical practical consideration. Due to these factors, an AUV always needs to have a percise, „target specific", „small scale" control law implemented for position-keeping at the set point.

Such a „small scale" scheme that can function with this „large scale" trajectory planner is being developed by Mišković et al. (2006). Having this in mind, the appearance of this static error in the trajectory planner is not a critical issue.

A different experiment, with a moderately cluttered theater of operations, is now performed for the thus modified trajectory planner. The simulation results are presented in fig. 4.

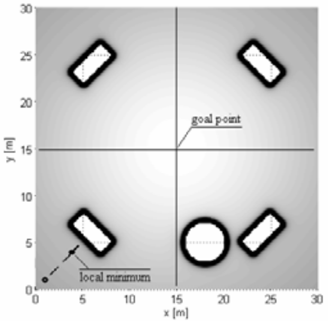

Figure 4: Results of trajectory planning in a relatively cluttered theater of operations

Figure 4 demonstrates a flaw of virtual potential methods in general. The trajectory unpredictably terminates in a local minimum. Analytical treatment of this problem is, depending on the PDFs considered, either intractable, or when tractable, often an NP-hard computational problem. Therefore, a scheme is proposed where in order to avoid local minima, a short-time „perturbation“ is injected into the theater of operations.

In the research resulting in this paper, „perturbation“ is implemented in the form of a temporary „ghost“ goal-point. This goal-point is set up according to figure 5, and eq.-s 16 and 17.

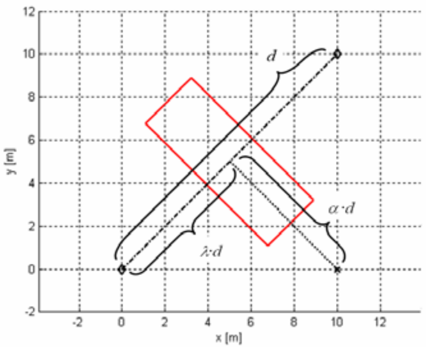

Figure 5: Schematic explanation of the „ghost goal-point“ method of perturbation from the local minimum

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

$$\vec{v} = \begin{bmatrix} y_{GP} - y \\ x_{GP} - x \end{bmatrix};\ d = \left\| \vec{v} \right\|;\ \tilde{\vec{v}} = \frac{\vec{v}}{d} \qquad (16)$$

$$\vec{r}_{GGP} = \begin{bmatrix} x_{GGP} \\ y_{GGP} \end{bmatrix} =$$

$$= \lambda \cdot d \cdot \tilde{\vec{v}} + \alpha \cdot d \cdot \mathbf{R}\left(\frac{\pi \cdot (1 + 2\rho)}{2} \right) \cdot \tilde{\vec{v}} \qquad (17)$$

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 length of the line connecting (x, y) and (x_{GP}, y_{GP})
- λ is the tunable fraction (method-specific parameter) of the length d dictating the position of the normal
- α is the tunable multiple (method-specific parameter) of the length d dictating the position of the „ghost“ goal-point along the normal
- ρ is the random binary value deciding the side – left or right, off to which the „ghost“ goal-point is offset from the line connecting (x, y) and (x_{GP}, y_{GP})
- $\mathbf{R}(\phi)$ is the matrix executing the rotation of the vector anteceding it (multiplying it from the right)
- $\tilde{\vec{v}}$ is the unit-vector encoding the direction of the line connecting (x, y) and (x_{GP}, y_{GP})

The „ghost“ goal-point is deleted out of the percieved collection of PDFs once the AUV is within its ε_{ghost}-vicinity. In a variety of realistic, relatively uncluttered environments, it is to be expected that along the new relative bearing of the goal-point, the AUV will not encounter local minima.

The simulated experiment displayed in figure 6 was rerun, with the inclusion of the local-minima-avoiding scheme. The results are displayed in figure 6.

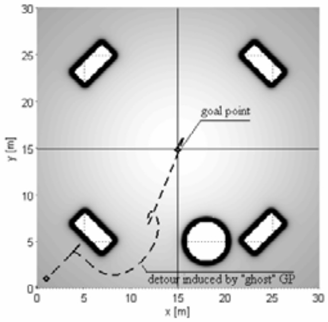

Figure 6: Recreated trajectory planning simulation in a relatively cluttered theater of operation with the local minima avoidance scheme implemented

The extension of this method to multiple AUVs yields, for a realistic theater of operations (as presented in the figure), simulation results presented in figure 7. In this experiment, for each AUV planning its trajectory, other AUVs are classified as circular obstacles with $\{r_0 = 0.5, A^+ = 2.5\}$.

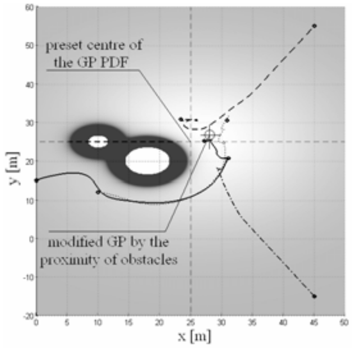

Figure 7: Basic grouping / clustering / formation simulation results (various dashed styles of trajectories used to represent trajectories of different agents; Start and end points marked)

It can be seen from this simulation result that a certain level of formation-like behavior is achieved. It is worth noting that this is the case although no further modifications were applied to the algorithm. Neither was the library of PDFs expanded with additional or more complex mathematical forms.

However, forming up of the AUVs within the vicinity of the goal-point is not robust. Rather, it is strongly influenced by the presence, pose, dimensions and repulsive influence of the nearby obstacles. This is due to the fact that the presence of nearby and/or strongly repulsive obstacles contributes to the distortion of the goal-point-local equipotential lines. As AUVs seek to uniformly distribute themselves in steady state along the equiopotentials, distorted shapes of the equipotentials result in asymmetric formations.

IV CONCLUSIONS AND FURTHER WORK

This paper gives a description and provides insights critical to the actual implementation of a virtual potential method-based trajectory planner for AUVs. The trajectory planner is an example of an approach based on simple numerical mathematical calculations. Such calculations are very efficiently implemented at assembly-language level in contemporary embedded computer processors. It is shown that this planner produces stable, energetically conservative trajectories. A mechanism for reducing the influence of local minima is proposed and tested through simulation. Also, it is shown by simulation that the algorithm already features a class of forming-up behavior when run in multiple AUVs within the same theater of operations. However, this behavior is non-robust and environmentally dependent without extending the implemented classes of PDFs. This behavior emerges without any communication of state between the algorithms run on separate platforms.

Further work on the algorithm will be concentrated on several advances. One is the software design which will endeavour to take into account the specifics of the implementation in one of the high-level languages who are yet real-time-efficient and fast at both compile-time and run-time. In that respect, hardware-in-the-loop operation will be supported, and the algorithm will be able to control a real AUV (or an ROV run by a top-side computer).

Another advance is the exploration of the extension of PDF classes. Separate classes of PDFs need to be implemented, to be used to represent cooperating AUVs. These PDFs should be radially nonmonotonic, i.e. should feature local minima at certain geometric configurations. It is supposed, as supported by equivalent problems in crystalline physics, that a scheme supporting the appearance of formations robust to other features of the environment, is likely to emerge.

REFERENCES

Barišić, M., Vukić. Z. and Mišković N. (2006), “Design Of A Coordinated Control System For Marine Vehicles”, *Proceedings of the 7th IFAC Conference on Manoeuvring and Control of Marine Craft*, Proceedings on CD

Beard, R. W. and McLain, T. W. (2003). “Multiple UAV Cooperative Search under Collision Avoidance and Limited Range Communication Constraints”, *Proceedings of the 42nd IEEE Conference on Decision and Control,* pp. 25-30

Bellingham, J. S. et al. (2002). “Cooperative Path Planning for Multiple UAVs in Dynamic and Uncertain Environments”, *Proceedings of the 41st IEEE Conference on Decision and Control,* pp. 2816-2822

Cassandras C. G. and Li, W. (2002). “A Receding Horizon Approach for Solving Some Cooperative Control Problems”, *Proceedings of the 41st IEEE Conference on Decision and Control,* pp. 3760-3765

Earl , M. G. and D’Andrea, R. (2002). “Modeling of a Multi-Agent System Using Mixed Integer Linear Programming”, *Proceedings of the 41st IEEE Conference on Decision and Control,* pp. 107-111

Fax, J. A. and Murray, R. M. (2003). “Information Flow and Cooperative Control of Vehicle Formations”, *IEEE Transactions on Automatic Control,* **vol. 9**, pp. 1465-1476

Fiorelli, E. et al. (2004). “Multi-AUV Control and Adaptive Sampling in Monterey Bay”, *Proceedings of the IEEE Autonomous Underwater Vehicles 2004: Workshop on Multiple AUV Operations,* pp. 134-147

Meshabi M. (2003). “State-Dependent Graphs”, *Proceedings of the 42nd IEEE Conference on Decision and Control*, **vol. 3,** pp. 3058-3063

Mišković, N. et al. (2006). “Autotuning Autopilots for Micro-ROVs”, *14th Mediterranean Conference on Control and Automation,* Proceedings on CD

Moreau L., Bachmeyer R. and Leonard N. E. (2003). “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*

Ögren P., Fiorelli E., Leonard N. E. (2003), “Cooperative Control of Mobile Sensor Networks: Adaptive Gradient Climbing in a Distributed Environment”, *IEEE Transactions on Automatic Control*, **vol. 9**, pp. 1292-1302

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

Sepulchre R., Paley D. and Leonard N. E. (2005). “Graph Laplacian and Lyapunov Design of Collective Planar Motions”, *Proceedings of the International Symposium on Nonlinear Theory and Its Application*

Stanković S. S., Stanojević, M. J. and Šiljak D. D. (2005),.“Stochastic Inclusion Principle Applied to Decentralized Overlapping Suboptimal LQG Control of a Platoon of Vehicles”, *Proceedings of EUROCON*