Adaptive Control for Autonomous Underwater Vehicles

Conor McGann, Frederic Py, Kanna Rajan, John Ryan, Richard Henthorn
Monterey Bay Aquarium Research Institute, Moss Landing, California
{cmcgann, fpy, kanna.rajan, ryjo, henthorn}@mbari.org

Abstract

We describe a novel integration of Planning with Probabilistic State Estimation and Execution. The resulting system is a unified representational and computational framework based on declarative models and constraint-based temporal plans. The work is motivated by the need to explore the oceans more cost-effectively through the use of Autonomous Underwater Vehicles (AUV), requiring them to be goal-directed, perceptive, adaptive and robust in the context of dynamic and uncertain conditions. The novelty of our approach is in integrating deliberation and reaction over different temporal and functional scopes within a single model, and in breaking new ground in oceanography by allowing for precise sampling within a feature of interest using an autonomous robot. The system is general-purpose and adaptable to other ocean going and terrestrial platforms.

Introduction

Autonomous Underwater Vehicles (AUVs) (Yuh 2000) are untethered mobile robotic platforms used by the oceanographic community. AUVs carry sophisticated science payloads for measuring important water properties (Ryan 2005), as well as instruments for recording the morphology of the benthic environment with advanced sonar equipment (Thomas 2006). The extensive payload capacity and operational versatility of these vehicles offer a cost-effective alternative to traditional ship-based oceanographic measurements.

Existing mission control practices rely on manually scripted plans generated a priori that are tedious to construct and, more importantly, inflexible to changes during mission execution. This prevents in-situ adaptation of mission structure essential to improving operation in an environment as dynamic and uncertain as the ocean (Rudnick and Perry 2003). Ocean exploration also requires adaptation to pursue unanticipated science opportunities. Safe and effective adaptation requires a balanced consideration of mission objectives, environmental conditions and available resources. Moreover, as mission durations increase, sustained presence in the ocean requires the ability to deal with off-nominal conditions.

We have developed and deployed an onboard Adaptive Control System that integrates *Planning* and *Probabilistic State Estimation* in a hybrid Executive. Estimation, Planning, and Execution must be integrated in order to inform planning systems with predictions of the most likely evolution of the environment. Probabilistic State Estimation integrates a number of science observations to produce a likelihood that the vehicle sensors perceive a feature of interest. Onboard planning and execution enables adaptation of navigation and instrument control based on the probability of having detected such a phenomenon. It further enables goal-directed commanding within the context of projected mission state and allows for replanning for off-nominal situations and opportunistic science events.

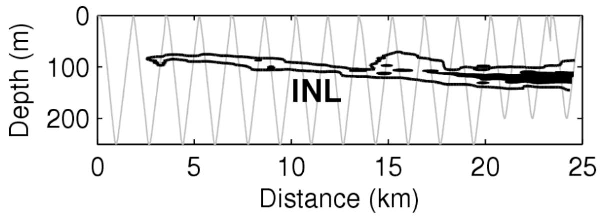

Fig. 1: INL mapped by AUV in Monterey Bay, California. The AUV survey track is shown in gray.

We have tested our system in the coastal ocean where complex interactions between physical, chemical, geological, and biological processes often occur. Our studies have targeted two highly unpredictable phenomena frequently encountered in the coastal waters of central California. (1) Intermediate Nepheloid Layers (INL), (Fig 1), are fluid sheets of suspended particulate matter that originate from the seafloor (McPhee-Shaw 2004, Ryan 2005). (2) Estuarine plumes are outflows from land-influenced coastal waterways into coastal waters. In these studies our aim is to dynamically adapt the control of an AUV after detecting a feature of interest, to take water samples within the feature and to modify the survey spatial resolution while projecting the impact to the mission plan.

The novelty of this work is twofold. We integrate deliberation and reaction over different temporal and functional scopes within a single agent and a single model that covers the needs of high-level mission management, low-level navigation, instrument control, and detection of unstructured and poorly understood phenomena. Secondly, we break new ground in oceanography by allowing scientists to obtain samples precisely within a scientific feature of interest using an autonomous robot.

The structure of the paper is as follows. We first motivate our approach by introducing key concepts of our design. We then describe the synthesis of planning and probabilistic state estimation within a hybrid executive based on these ideas. Results from sea trials in Monterey Bay, California are presented next. We conclude with a review of related efforts and a discussion of future work.

Key Concepts

The Sense-Plan-Act (SPA) paradigm for robot control embeds planning at the core of a control loop. Planning is typically the dominant cost and can limit the reactivity of an agent. Furthermore, the world can (and often does) change at a faster rate than the planner can plan. In this situation, the agent may thrash if the internal state of the plan gets out of synch with the actual state of the world. So, while SPA offers a general representational and computational framework for control, it is problematic for application in systems that require extensive deliberation and fast reaction.

For example, in our domain, an AUV may be tasked with a set of objectives to be accomplished over the course of a mission. Correctly selecting a feasible subset of these objectives, and deciding the order in which they should be accomplished has a big impact on the utility of a mission. To do this effectively requires deliberation over the full temporal extent of the mission. In contrast, instrument control decisions require faster reaction times (e.g., 1 second) but can be taken with a more myopic view of implications to the plan without loss in utility. This suggests that the control responsibilities of the executive can be factored according to how far to look ahead (temporal scope), which state variables to consider (functional scope) and required reaction time (latency). To exploit this we allow the executive to be *partitioned* into a coordinated collection of control loops, each with *its own internal SPA cycle*, distinguished explicitly by these parameters above.

Although partitioning can be a powerful tool to reduce planning complexity it is unreasonable to require all planning to complete at the rate at which the environment is changing. Rather, planning within each control loop must be completed at the rate at which it must react to provide effective control. In order to allow a uniform quantization of time throughout the model, yet permit different rates of reaction for different control loops, it becomes necessary to allow sensing and planning to be interleaved. This allows multiple state updates to occur during deliberation, keeping the plan in synch with an evolving world state. To support this we *decouple synchronization of the plan from deliberation* over goals and allow them to be interleaved.

Our emphasis on *partitioning* and *synchronization* to flexibly and efficiently integrate deliberation and reaction offers a novel variation of the Sense-Plan-Act paradigm. We now describe the application of these ideas to the problem of adaptive mission control.

The Teleo-Reactive Executive

The structure of the integrated control system is shown in Fig 2 implemented as a *Teleo-Reactive Executive (T-REX)* where each component is an instance of a *Teleo-Reactor (*McGann 2008*)*. Three reactors are organized hierarchically: a *Mission Manager*, a *Navigator*, and a *Vehicle Control Subsystem (VCS)*. For purposes of discussion the VCS is conceptualized as a functional layer. Each reactor receives goals to drive its behavior, sends goals to task a lower level reactor, and receives observations on system state variables of interest. For example, the *Mission Manager* may receive a goal to conduct a volume survey in a designated region, with constraints on finish time, depth bounds, or spatial resolution ranges. It may generate a plan at a high level that includes steps to navigate to a target location, turn to a heading etc. This high-level plan considers the full temporal scope of the mission (e.g., 10 hours). Near-term navigation steps of the plan are dispatched as goals to the *Navigator*. The *Navigator* will plan over a shorter time horizon (e.g., 60 seconds). Ultimately, mission commands (e.g., *ascend*, *get a GPS fix*, *descend*) are generated and dispatched to the VCS. Feedback from the VCS takes the form of vehicle and command state updates. Fig 2 shows a common framework for integrating deliberation and reaction for the *Mission Manager* and *Navigator* that we call a *Deliberative Reactor* based on the EUROPA$_2$ (Frank 2003) constraint-based temporal planning library.

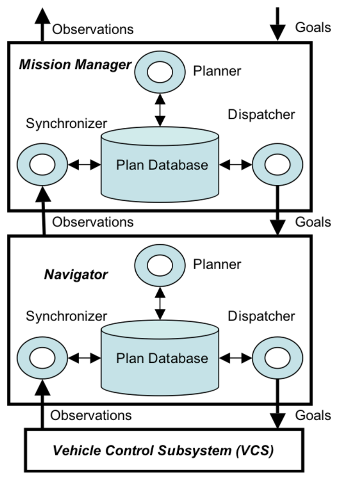

Fig. 2: Integrated System Design

The Plan Database

A constraint-based temporal plan is a form of constraint network (Mackworth 1992). Algorithms designed to operate on the plan during planning are directly exploited in execution to propagate state updates, detect conflicts, and reason about temporal precedence. A single model is used to capture the domain details at all levels of abstraction. The model is applied using a propositional inference engine triggered by changes in the plan. The dependency structure of the plan is utilized in execution to aid recovery from plan failure. The plan, model and plan manipulation algorithms are contained in the Plan Database.

Figure 3 illustrates a plan fragment 1432 seconds into mission execution. It contains two state variables: the

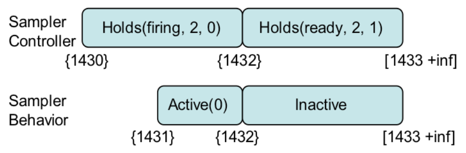

Fig 3: The plan database at *t*=1432

SamplerController and the *SamplerBehavior*. The former is an *internal* state variable of the *Navigator*, the latter is *external*, reflecting the status of a behavior for sampler activation resident in the VCS. Each reactor is the owner of its internal state variables, and publishes their values during synchronization. Each reactor subscribes to observations on its external state variables and must reconcile its internal state with these external values. The evolution of each state variable is recorded (when in the past) and projected (when in the future) in a *timeline*. The *SamplerController* oscillates between *firing* and *ready*. Each state includes a bound on the number of samples that can be used (e.g., 2), and a record of the number of samples taken thus far. The *start* and *end* times of each state are shown. The figure shows that the *SamplerBehavior* was activated at t=1431 and remained active for 1 second. The transition to the *Inactive* state is concurrent with the transition of the *SamplerController* to a *ready* state. Notice that this parameter has incremented after the controller has triggered. Finally, observe that the end time of the *ready* state is unbound, since the *SamplerController* may remain in a ready state indefinitely.
To illustrate these ideas further, consider the model rule below applied to all values of the *SamplerController* state variable of the predicate *Holds*.

$$\forall a \in \text{Holds} \Rightarrow \exists\, b \in \text{Holds}$$
$$a.\text{start} < a.\text{end} \;\wedge\; a.\text{end} = b.\text{start}$$
$$a.\text{max} = b.\text{max} \;\wedge\; b.\text{count} = a.\text{count} + 1$$

an example constraint

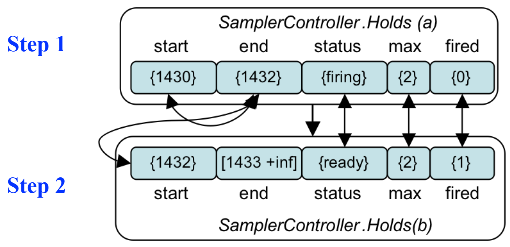

Fig 4: A constraint network fragment

A fragment of the partial plan where this rule applies is shown in Fig. 4. There are two values as before. However, each is now depicted as a *token* that acts to constrain the set of values a state variable may hold over some temporal extent. All tokens have *start* and *end* variables. Different predicates may contain different parameter variables. Bi-directional arrows indicate constraints between variables in accordance with the model rule above. The implication from the model is reflected by the directed arrow from token *a* to *b*. Only the *end* variable of *b* is unbound in this example; all other variables were bound to singletons as the plan is executed. In general, the ability to leave parts of the plan flexible supports a *least-commitment* approach to planning where only necessary restrictions are imposed allowing final details to be refined at or close to the point of execution.

Embedding the Plan Database in Execution

The Plan Database is *embedded* in the control loop by *synchronizing* observations with the plan and *dispatching* the plan as goal-requests to the responsible reactor (e.g., the VCS). Dispatching the plan is detailed in (McGann 2008). The goal of synchronization is to produce a consistent and complete view of the plan at the *execution frontier*. Observations are received for external state variables. For example, say that *SamplerBehavior* is an external state variable whose value can be either *Active* or *Inactive*. A message from the VCS indicating that the behavior has completed becomes a *SamplerBehavior.Inactive* token in the plan database. Fig 5 illustrates the situation before and after this observation is synchronized.

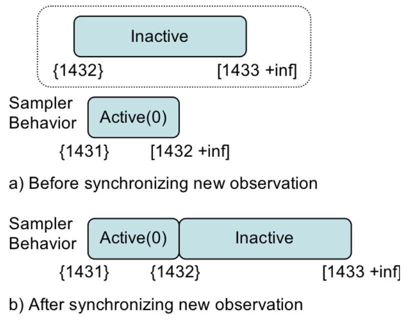

Fig 5: Synchronizing an observation with the plan

The observation is received at t=1432. The last observed value for the *SamplerBehavior* was *active*. This new observation is resolved by insertion into the plan, resulting in a restriction of the end time of the *active* state. The connections in the underlying constraint network propagate the implications of this restriction throughout the partial plan. This process is repeated for all observations and for all implications that impact the execution frontier. In the event that the plan is not consistent with the observations, the *plan is discarded, and the relaxed execution frontier is synchronized with observations*.

The Planner

Automated planning refines an incomplete plan into one with sufficient detail to support execution. Planning and execution are interleaved seamlessly while operating on a shared database. Planning is broken down into atomic steps, where each step leaves the database in a consistent state. The planner is invoked one step at a time, in order to permit synchronization to take place. Updates applied during synchronization are visible in the next step of the planner so further refinement of the plan will account for the changing world. The planner is invoked whenever flaws occur in the plan. Flaws stem from either externally

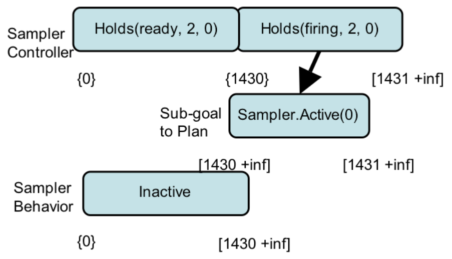

Fig 6: Sub-goal implied by the model to be planned

requested goals (e.g., from another reactor, or a user-driven input) or implied by the model and feedback from the environment.

For example, consider the partial plan depicted in Fig 6. The *Sampler Controller* timeline is transitioned to trigger the sampler, based on observations made from the environment. In order to actually trigger the instrument, the model requires the *SamplerBehavior* to be *Active*. Consequently, the model will introduce a sub-goal for the *SamplerBehavior* for the planner. Thus, while there is a clear separation between synchronization and deliberation, the flow of information through the shared partial plan structure provides a continuous integration of state as they are interleaved.

State Estimation

To enable the AUV to adapt its mission based on feature detection, we employ a Hidden Markov Model (HMM) (Rabiner 1986). To execute this HMM, we encode it directly *within* the unified representational and computational framework of a Deliberative Reactor allowing for a seamless integration of state estimation through synchronization and planning. The sensor data we use depends on the feature of interest; for instance we use two optical sensor readings for INL detection. HMMs are useful since the stochastic nature of these models can correlate the type of features we want to detect with the sensor observations. And such models can also be learned using past mission data (Fox 2006).

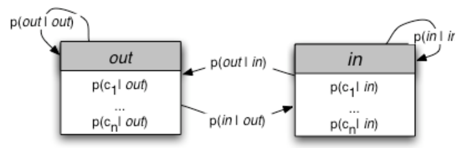

Fig 7: The HMM used by the state estimator.

Fig 7 shows two hidden states: *in* - expressing that the AUV is *in* the feature of interest – and its alternate *out*. The model specifies the set of probabilities of observation c_i given state s_i, $p(c_i \mid s_i)$, and the probabilities of transition from s' to s, $p(s|s')$. In our implementation the probabilities of transition are fixed empirically ((Fox 2006) shows how can they be learned). This model is used to compute the new probabilities $\{p(in)_t, p(out)_t\}$ using (1):

$$p(s_t) = a_t * p(c_t|s_t) * \Sigma_{s' \in \{in,\, out\}}\, p(s_t|s'_{t-1}) * p(s'_{t-1}) \qquad (1)$$

with a_t as a normalization factor such that :

$$\Sigma_{s \in \{in,\, out\}}\, p(s_t) = 1 \qquad (2)$$

Equations (1) and (2) encapsulate HMM based belief. In so doing, they express a constraint between current estimated state, the last observation and the previous estimated state. Fig 8a shows an HMM execution trace representation where arrows represent the dependence between states and observations. Fig 8b shows the timelines managing our state estimation where arrows stand for constraint propagation between tokens, indicating a strong similarity between the two.

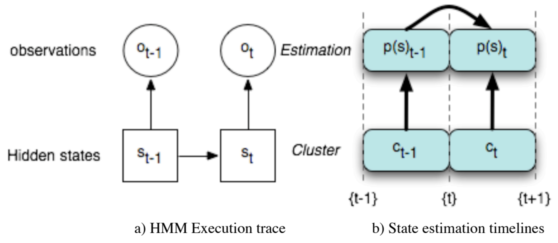

Fig 8: Similarity between HMM and Timeline Representations

The model is applied in execution as follows:

1. Sensor data is classified by the VCS so that it computes the corresponding cluster as in (Fox 2007). This cluster value is provided as an observation on the *Cluster* timeline to the Navigator.
2. The observation is inserted in the plan during synchronization.
3. The implications of this observation on the *Estimation* timeline, expressed in the model, are applied and resolved during synchronization. Constraints then propagate the probability distribution according to (1).

At this stage we have the estimation deduced from the HMM. The transition from the estimation to decision making can then be done directly. For instance to decide when to take a water sample the steps are:

4. If the probability of being within a feature of interest (such as an INL) dominates the distribution and other sampling conditions are satisfied (e.g spatial constraints dispersing sampling locations), then a sample should be taken. This rule is evaluated during synchronization as the constraint network is propagated.
5. Should this rule fire, a new token will be generated to transition the *SamplerController* into a firing state. This triggers a new deliberation phase to resolve the implications of this transition (i.e., select a sampling canister, and generate an action to fire it).

We use a similar process for altering the spatial resolution of the survey using the HMM to compute the probability of having seen the feature during the last vehicle transect. This in turn is used to determine the spatial separation between the previous and next transect.

Results

The evaluation of our system focusses on 2 key issues: the efficacy of the adaptive behaviour for science and the efficiency of the integration. The first requires us to test that the system is able to predict the occurence (or absence) of the pheonomena and adjust instrument and navigation control accordingly. With the second, we are concerned with reactivity in a real-world environment.

Our results are based on extensive simulations and multiple sea trials over a period of a year, culminating in science driven surveys. The surveys applied the integrated system onboard a *Dorado* AUV, operating at a speed of 1.5 m/s, to

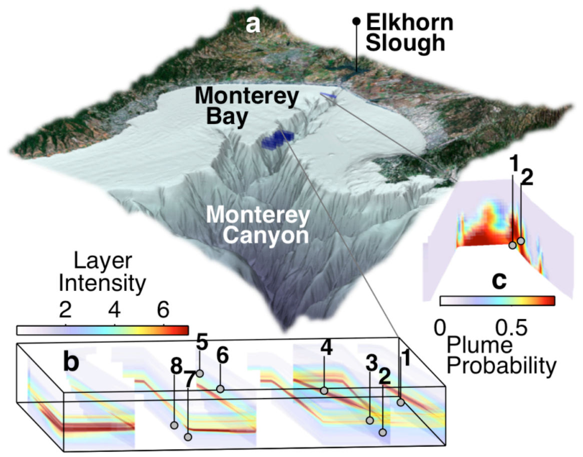

Fig 9: a) AUV survey volumes (blue) within the context of regional seafloor and land topography. b) INL optical backscattering intensity (x10^{-3} m^{-1} at 676 nm). c) Plume detection probability (relative). Water sample locations are indicated in b) and c).

detect and sample an INL in Monterey Canyon and a coastal plume caused by outflow from the Elkhorn Slough (Fig 9a). Water sampling employed a system developed for AUV applications (Bird 2007). INL intensity, indicated by optical data, is shown in Fig 9b along with locations of water samples. Plume probability computed by the HMM is shown in Fig 9c along with the water sample locations. In both missions, our system was able to detect the feature of interest, adapt the resolution of the survey consistent with the finding (or lack thereof), and sample the feature. In both missions the state estimator correctly excluded all measurements that did not correspond to the feature of interest, with no associated triggering of the water-sampler. For the INL survey, the system generated and dispatched a total of 188 commands over a period of 4 hours over the canyon axis. Using conventional approaches, the operator would have needed to specify all commands sent to the VCS a priori without a guarantee of having sampled with the precision expected for science.

For these sea trials, our system included 28 timelines distributed across 3 reactors. The deliberative reactors ran on a 367 MHz EPX-GX500 AMD Geode stack using Red Hat Linux, with the VCS functional layer running on a separate processor. Computational analyses of both these runs indicate that approximately 20% of the time, the CPU load was at about 5%. While the agent was always synchronizing at 1Hz, and receiving observations from the VCS at that rate, there was very little change in the plan otherwise so the cost of synchronization is low with little deliberation. Approximately 40% of the time, the CPU utilization was around 10% when actively executing the HMM. Once the vehicle was within the feature, the load peaked at 25% and was dominated by the cost of synchronization to evaluate spatial constraints and sampler availability. The average CPU cost of deliberation was less than 1% and 90% of the time there was no deliberation. This low number shows that the flexible partial plan representation is incrementally refined and extended throughout the mission without any need to re-plan. Where re-planning was required, it was localized within the *Navigator* and only impacted a small time horizon validating our information model for partitioning of control.

Related Work

Most plan execution techniques are coupled with state estimation for fault detection isolation and recovery (FDIR). A classic example of such an effort was the Remote Agent Experiment (RAX) (Muscettola 1998 and Rajan 2000) where onboard FDIR was used to trigger replanning and projection of spacecraft state. RAX had three distinct components with their own models with correspondingly significant design and integration issues (Bernard 2000). Like RAX, the Autonomous Sciencecraft Experiment (Chien 2005) and other 3T architectures (Gat 1998) did not have a unified framework but relied on loosely coupled systems with distinct representations.

Other efforts have focused on deriving models of time series of state evolution of structured phenomena to build Hidden Markov models of the environment (Lenser 2005) even if the architecture does not explicitly deliberate as outlined in our work. In (Sellner 2007) plans with durative action can be repaired inline to deal with execution anomalies using probabilistic approaches. Execution in this work is however simulated within a less flexible plan representation, while dealing with kernel density estimations for duration prediction.

IDEA (Muscettola 2002, Finzi 2004) is a hybrid executive that integrates planning and execution in a single representational and computational framework with a unified declarative model. Our approach is similar in its formulation of a timeline-based representation and in its use of constrained-based temporal planning to implement a control loop. In contrast to IDEA, we separate synchronization from deliberation. We further allow synchronization to be interleaved with deliberation permitting reaction times to extend to multiple time steps if necessary. Moreover, the partitioning structure we provide automatically applies rules for synchronization and dispatch to co-ordinate among control loops and resolve conflicts. These common patterns must be encoded explicitly in an IDEA model, which we believe is unnecessary and cumbersome.

Our work is further distinguished in the underwater domain where popular techniques for AUV control have relied on reactive approaches; see (Carreras 2006) for a survey. ORCA (Turner 1991) and MCS (Palomeras 2007) while integrating deliberation onboard do not deal with state estimation or durative action representation. In the case of ORCA, the case-based planner appears to have been used in simulation only. Our work is the *first integrated Planning, Execution and Estimation framework deployed for actual oceanographic exploration*.

Conclusions and Future Work

Our field results to date are encouraging. Using a unified representational and computational framework for Estimation, Planning, and Execution has proved effective for adaptive mission control. The integration of such capabilities has resulted in precise observation and sampling of dynamic processes that had not been done to date. The scalability suggested by the CPU load data while promising, requires more rigorous empirical evaluation.

The focus of our effort going forward will be in two key areas: to investigate approaches for online learning and longer-term projection for state estimation and to investigate adaptive sampling within an information theoretic framework. We expect this will allow us to qualitatively characterize a dynamic feature in 4D (space and time) by allowing the vehicle to spatially map and sample the feature over an extended period of time. Our longer-term vision is to extend our framework to control a diverse range of mobile and immobile robotic assets that can collaboratively help quantify a range of processes in the ocean.

Acknowledgements

This research was supported by the David and Lucile Packard Foundation. We thank NASA Ames Research Center for making the EUROPA planner available, our colleagues at MBARI for their help in integration and deployment and Nicola Muscettola for detailed comments on a late draft of this paper. We also thank the anonymous reviewers with their insightful comments.

References

Bird L., A. Sherman and J. Ryan. "Development of an Active, Large Volume, Discrete Seawater Sampler for Autonomous Underwater Vehicles", Proc Oceans MTS/IEEE Conf, Vancouver, Canada, 2007.

Bernard, D.E., et.al, "Remote Agent Experiment: Final Report," NASA Technical Report, Feb. 2000.

Carreras M., et.al, "Behaviour Control of UUV's" in Advances in Unmanned Marine Vehicles, by G. N. Roberts, Robert Sutton Eds, IEE Control Series, January, 2006.

Chien, S. et.al, "Using Autonomy Flight Software to Improve Science Return on Earth Observing One", Journal of Aerospace Computing, Information Communication. April 2005.

Finzi, A., F. Ingrand and N. Muscettola, "Model-based Executive Control through Reactive Planning for Autonomous Rovers", IROS 2004.

Fox M., et.al. "Robot introspection through learned hidden markov model", AIJ, 170(2):59–113, Feb 2006.

Fox M., et.al "In-situ Analysis for Intelligent Control" Proc. of IEEE/OES OCEANS Conference, Aberdeen, Scotland, UK.

Frank, J., A. Jonsson, "Constraint-based attributes and interval planning," Journal of Constraints, vol. 8, Oct. 2003.

Gat. E, "On Three-Layer Architectures" In D. Kortenkamp, R. Bonnasso, and R. Murphy, Eds, Artificial Intelligence and Mobile Robots, MIT Press, MA, 1998.

Lenser, S "On-line Robot Adaptation to Environmental Change", PhD thesis CMU-CS-05-165, August 2005.

Mackworth A.K., "The Logic of Constraint Satisfaction" AIJ 58(1-3): 3-2, 1992.

McGann C., et.al, "A Deliberative Architecture for AUV Control", To appear Intnl. Conf. on Robotics and Automation, Pasadena, Pasadena, CA, 2008.

McPhee-Shaw, E. E., et.al. "Observations of intermediate nepheloid layers on the northern California margin" Continental Shelf Research, 24, 693-720, 2004.

Muscettola N., et.al. "Remote Agent: To Boldly Go Where No AI System Has Gone Before" AIJ 103(1-2):5-48, August 1998.

Muscettola, N. et.al "IDEA: Planning at the Core of Autonomous Reactive Agents," in Proc 3rd Intnl. NASA Workshop on Planning and Scheduling for Space, Houston, TX, October 2002.

Palomeras, N., et.al "MCL: A Mission Control Language For AUV's", Control Applications in Marine Systems, Brac, Croatia, 2007.

Rabiner, L. R., "An Introduction to Hidden Markov Models", IEEE ASSP Magazine, 4–16, 1986

Rajan K., et.al, "Remote Agent: An Autonomous Control System for the New Millennium," Prestigious Applications of Intelligence Systems, 14th ECAI, Berlin, 2000.

Rudnick, D.L., and M.J. Perry. ALPS: Autonomous and Lagrangian Platforms and Sensors, Wkshp Report, 64 pp., www.geo-prose.com/ALPS, 2003.

Ryan, J.P., et.al "Physical-biological coupling in Monterey Bay, California: topographic influences on phytoplankton", Marine Ecology Progress Series, Vol 287: 28-32 2005.

Sellner B., Simmons, R., "Duration Prediction for Proactive Replanning" 3rd Intnl. Wkshp on Planning and Execution for Real-World Systems, ICAPS, 2007.

Thomas, H., et.al, "Mapping AUV Survey of Axial Seamount", Eos Trans. AGU, 87(52), Fall Meet. Suppl., Abstract V23B-0615, 2006

Turner, R. M., Stevenson, R. A. G. "ORCA: An Adaptive, Context-Sensitive Reasoner for Controlling AUVs". in Proc 7th Unmanned Untethered Submersible Technology (UUST), Durham, New Hampshire, 1991.

Yuh, J., "Design and Control of Autonomous Underwater Robots: A Survey" Autonomous Robots, 2000 8, 7–24, 2000.