Proceedings of the 41st IEEE
Conference on Decision and Control
Las Vegas, Nevada USA, December 2002

Specification and design of coordinated motions for autonomous vehicles[1]

João Borges de Sousa[2] Fernando Lobo Pereira[3]

Abstract

New concepts for operational missions to be performed, conceivably in a near future, by networked vehicles and systems are presented. It is shown that, at the heart of those concepts are simple coordination and control patterns that can be expressed in terms of sets and of the relations among the elements of those sets. A specification and design framework for the networked operation of multiple vehicles and systems is proposed, and related to recent developments in control, computation, and networking.

Keywords: Optimization, Control design, Coordination, Networked vehicles, Specification.

1 Introduction

In this paper we propose a specification and design framework for the networked operation of multiple vehicles and systems. This framework was motivated by concepts for operational missions to be performed in a near future – probably during this decade – by networked vehicles and systems. These concepts pose new challenges to control engineering, networking, and computation. First, we require vehicles, devices, and controllers with evolving access capabilities to dynamically interact with each other, by exchanging messages and data, in order to execute tasks. Second, we need different notions of what should be observed under different circumstances, giving rise to questions of when two different systems exhibit the same behavior. Third, we need motion coordination models that are highly application dependent, and that may change with time for the same application. Forth, we need to incorporate in the analysis and control synthesis the aspects of communication and concurrency that are at the heart of these systems. Fifth, we need to consider data provisioning systems to manage and maintain representations of data exchanged with network-centric multi-vehicle systems. This challenge requires a convergence of methods and techniques from control engineering, networking and computer science[1].

We use sets and the relations among the elements of these sets to specify the behavior of networked vehicles and systems. These simple concepts are enough capture the structure of interacting systems, information structures, and team composition and tasking: each vehicle/system provides atomic services; groups of vehicles provide complex services that cannot be delivered by one single vehicle; complex services are defined by predicates on the positions and atomic services of groups of vehicles and systems; services are available in a neighborhood of the service providers; a team is a set of vehicles capable of delivering complex services and of controlling the way these services are delivered; teams interact to use services from other teams; teams rendezvous, assemble or disassemble at designated areas and times.

We organize design in this framework in terms of the activities of each team. The activities concerning the maintenance of the integrity of the team take precedence over the others. This requires a layered structure of controls. The first layer ensures the integrity of the team, the second layer controls service delivery and coordination with other teams, and the third layer concerns team composition tasking, assembling and disassembling teams.

The paper is organized as follows. In section 2, we discuss a number of selected case studies that have served as laboratories for experimenting ideas, and from which scientific concepts can now start to be distilled, and we contrast those developments with new concepts for the operation of networked multi-vehicle systems, where we have identified new patterns of coordination and control. In section 3 we briefly describe the specification framework. In section 4, we discuss coordination and control problems in this framework, and relate them to recent developments in the control of distributed systems, communication, and computation. In section 5, we draw some conclusions.

[1]Research supported by Fundação da Ciência e Tecnologia project Cordyal and by Fundação das Universidades Portuguesas project COOP.

[2]Departamento de Engenharia Electrotécnica e de Computadores, Faculdade de Engenharia da Universidade do Porto, Rua Dr. Roberto Frias, 4200-465, Porto, Portugal, E-mail: jtasso@fe.up.pt

[3]DEEC-FEUP, E-mail: flp@fe.up.pt.

[1]See [17] for a more thorough review of the research concerning the description, analysis, controller design, simulation, and implementation of distributed systems.

2 Systems and concepts

2.1 Introduction

Networked vehicles and systems are receiving much attention for the potential that they represent to perform new military and civil tasks.

Today, and partly due to our involvement in the design and implementation of networked multi-vehicle systems, such as the ones described below, we have a better understanding of the underlying coordination and control issues [8].

Automated Highway System. The Automated Highway System (AHS) concept from the Partners for Advanced Transit and Highways (PATH) proposes a fully automated Intelligent Vehicle/Highway Systems policy [16]. This AHS is a distributed and hierarchical control system that aims to increase highway capacity, safety and efficiency without building more roads. In the AHS traffic is organized into groups of tightly spaced vehicles called platoons. Platoons are required to coordinate their motions, and in each platoon vehicles are required to maintain tight spacings between them.

Mobile Offshore Base. The Mobile Offshore Base (MOB) system illustrates a problem of coordination of surface vessels that has received significant attention due to its applicability to ship automation: docking, cargo transfer between moving vessels, platooning in restricted waterways, etc..[6]. The Mobile Offshore Base is a very large floating ocean structure meant to provide the same capabilities as an on-land army base. The modules forming the MOB must be able to perform long-term station keeping at sea, in the presence of waves, winds and currents. The modules must have precise relative position control with respect to each other. The string, however, is allowed to drift.

PISCIS. The PISCIS project addresses the specification and the implementation of a data collection system for real-time oceanography based on the operation of multiple Autonomous Underwater Vehicles (AUV) [7]. One of the thrusts of this project is the problem of adaptive sampling of oceanographic phenomena with heterogeneous AUVs with limited capabilities. The fundamental idea underlying adaptive sampling is to increase the survey efficiency by concentrating measurements in regions of interest. Thus, to map an ocean front, one might first run a very coarse survey to localize the front, and then concentrate operations in the front vicinity. Vehicles have limited communication capabilities, and coordination is restricted to the exchange of data and commands at pre-determined waypoints.

In the AHS and MOB systems, information and commands are exchanged among multiple vehicles, and the roles, relative positions, and dependencies of those vehicles change during operations. In the PISCIS system, and due to limited communication capabilities, coordination arises mainly from the a-priori functional coordination and resource allocation of heterogeneous vehicles with complementary capabilities. But networked systems have the potential for more complex patterns of coordination and control. In order to encompass all of these issues we need to consider potential applications from diverse fields, each presenting its own unique challenges to our efforts to generalize and formalize.

2.2 New operational concepts

Let us start with a preview of an oceanographic mission conceivable in a near future.

Imagine two teams of Autonomous Imagine two teams of Autonomous Underwater Vehicles (AUVs) that must coordinate their motions to find the local minimum of some oceanographic phenomenon. The first team, denoted as LPS, provides a Local Positioning System (LPS) service to other team. This system can be viewed as an underwater version of the Global Positioning Service (GPS). The second team, denoted as S, uses the LPS service for localization and provides a search service.

The Local Positioning System works as follows. Each LPS vehicle has a GPS receiver and an acoustic transponder – the vehicle is required to operate at the surface to receive the GPS signal. The transponder emits regularly, with a known frequency, an acoustic ping encoding the time t when it was emitted, and the name and location of the emitter at time t. The time t and the position of the emitter are given by the GPS receiver.

The AUVs from the S team are equipped with an acoustic system. This system detects the acoustic pings, and decodes them to extract the position and name of the emitter, and the time when the acoustic ping was emitted. This information together with the time of arrival of the ping is used to calculate the distance between the AUV and the emitter. To calculate its absolute position the AUV needs to receive pings from at least three sources – vehicles from LPS. In order to minimize the calculation error we require these three sources to be the vertices of a triangle. Due to attenuation, the LPS service is only available within a neighborhood P(t) of the LPS team.

The search service works as follows. The S team implements a modified version of the simplex optimization algorithm to find the local minimum of the phenomenon. Each vehicle has a suite of oceanographic sensors – to sample the phenomenon – and a low-bandwidth underwater acoustic communication system – to implement the search strategy in coordination with

the rest of the team and with the LPS team.

In terms of motion coordination the S team assumes the role of the leader. The LPS team controls the motions of its vehicles in order to keep all the elements of the S team inside P(t). It does this based on the information exchanged with the S vehicles.

Hereafter we designate this mission preview as the oceanographic mission, or simply the mission.

This mission exhibits `patterns of coordination and control` that are quite general: 1) vehicles provide atomic services; 2) teams provide complex services by controlling the positions and atomic services of its constituent vehicles; 3) services become available in a neighborhood of the service providers; 4) vehicles in a team implement motion coordination algorithms to distribute the execution of tasks among them; 5) teams interact to provide services to other teams, and to utilize the services provided by other teams (they have to stay within the corresponding neighborhoods of the service providers); 6) teams interact at temporal and spatial rendezvous to coordinate their future activities; 7) some teams exist to provide services to other teams, while others may also use services from other teams to implement motion coordination algorithms that pertain only to the members of the team; 8) teams are assembled and disassembled at spatial-temporal rendezvous; 9) the composition of teams can be done hierarchically.

We can find the same patterns of interactions in other problem domains, for example in applications involving Unmanned Air Vehicles (UAV)s (see [5] for an extensive survey).

UAVs are in high demand for military, scientific and civilian applications. Military operations present the most challenging scenarios. For military operations UAVs are tasked to "dirty","dull", and "dangerous" operations. "Dirty" refers to reconnoitering areas that may be contaminated, "dull" applies to surveillance or sentry duty, while "dangerous" is related to obvious threats, such as those posed by the suppression of enemy air defenses (SEAD). In SEAD missions we have several phases of team composition and tasking: 1) spatial and temporal rendezvous of teams of UAVs to minimize exposure to detection, or to assemble an appropriate number of assets at a given location in preparation for the next phase of the mission; 2) zone/target based coordination of teams; 3) implementation of mission-dependent tactics. In zone interdiction missions, we have teams of UAVs that coordinate their motions to maximize the size of the area where the interdiction service is provided. In reconnaissance missions, we have teams implementing search-based algorithms, with the integration of data from different sensors mounted on different vehicles. In attack missions, we have teams of vehicles implementing attack tactics in conjunction with other vehicles that provide targeting information and battle damage assessment.

3 Specification

We strive to identify an abstract and compact representation of the underlying coordination and control patterns: a vocabulary of relevant concepts and rules that determine how they can be used, and from which we can assert which properties will hold in the implementation, and which properties will not be present in the implementation. In summary, we want to reason about the specification, to find a design that refines it, and prove facts about the design and the specification.

In what follows we consider a set V of vehicles $1, \ldots, N$. The equations of motion for the ith vehicle are:

$$\dot{x}_i(t) = f_i(t, x_i(t), u_i(t)) \quad u_i(t) \in U_i$$

Consider the jth partition of V. At time t, $x_{i,j}$ denotes the state of all vehicles from this partition.

We specify team composition and tasking in terms of sets and relations among the elements of those sets. Sets of interest are: `Vehicles, AtomicServices, ComplexServices, SpatialTemporalPlaces, SpatialPlaces` (the names are self-explanatory). Next, we briefly introduce the vocabulary of relevant concepts and rules that determine how they can be used, and how we can reason about team composition and tasking. We present the general rules, and illustrate the encoding of a specific instance with reference to the oceanographic mission.

SpatialTemporalPlaces, SpatialPlaces. These are sets defined in the 3D space where all of the vehicles evolve, or defined in the product space of the 3D space and time. We use these sets to specify rendezvous, composition, termination, assembling or disassembling activities.

Vehicles and atomic services. There are several types of vehicles. Each vehicle has an unique name and provides atomic services. The function `Type` returns the type of each vehicle. The predicate `ProvideAtomicService` represents the atomic services provided by each vehicle. Atomic services have attributes, such as range, represented by the predicate `RangeAtomicService`.

`Vehicles`=$\{L_1, L_2, L_3, S_1, S_2, S_3\}$, `Type`($L_1$)=LPS,
`ProvideAtomicService`($L_1, \{GPS, Radio, Acom\}$),
`ComplexServices`={Position,Sensor,Comnetwork}

Complex services. We combine the atomic services from different vehicles to deliver a complex service at

an element of SpatialPlaces. The combination is established with links represented by the predicate Link. The predicate RequiredVehicles represents the vehicles required to deliver the service. The predicate RequiredConfiguration represents the structure of the links – also termed configuration – to be established by those vehicles to deliver the service. This predicate implicitly defines the information structures required to implement the service. The set-valued function ServiceArea returns the set (from SpatialPlaces) where the complex service is available.

$$\begin{aligned}
&\texttt{RequiredVehicles}(\{L_1, L_2, L_3\}, Position) \\
&\texttt{ServiceArea}(L_1, L_2, L_3, t, Position) = P(t), \\
&\texttt{ServiceArea}(L_1, L_2, L_3, t, Comnetwork) = C_L(t), \\
&\texttt{ServiceArea}(S_1, S_2, S_3, t, Comnetwork) = C_S(t) \\
&\texttt{Link}(L_1, L_2, Radio)
\end{aligned}$$

Team. A team is a set of vehicles – e.g. the i-th element from the j-th partition of the set V – capable of delivering complex services, and of controlling the way these services are delivered (by adjusting the set given by ServiceArea). Each complex service can be delivered in several alternative configurations. Each configuration is given by a state-constraint: the conjunction of K atomic state-constraints $\phi^k_{i,j}(x_{i,j}(t)) \leq 0$, where the $k-th$ constraint represents the $k-th$ link of the configuration. The service is delivered when at least one of those configurations holds (disjunction of state-constraints). Two or more services are delivered simultaneously when several configurations hold at the same time (conjunction of state-constraints). Internally, the team is characterized by the links and relations established among its components. The predicate ReachTemporalSpatialPlaces represents reachability of an element from TemporalSpatialPlaces when the initial position of a vehicle, or of a team, belongs to another element of TemporalSpatialPlaces. E.g., the initial position ($time \times space$) of L_1 is located within the set X_0, and we want to check if L_1 reaches X_f.

$$\texttt{ReachTemporalSpatialPlaces}(L_1, X_0, X_f)$$

Interactions. The team is the basic entity in all interactions. Externally a team is defined by its services, and by dynamic and state constraints describing the way those services may evolve in the 3-dimensional space. There are several modes of interaction among teams: teams rendezvous at elements of SpatialTemporalPlaces; teams make services available to other teams in the corresponding ServiceArea, and utilize services from other teams when located within the corresponding ServiceArea. Additionally, teams may also negotiate the calculation of spatial-temporal rendezvous on the fly. Teams may be required to coordinate their motions and services in the way the elements of a single team do it. This means that we may have a team of teams with different external and internal behaviors. Each team may be able to participate simultaneously in several interactions with other groups. For example, the interactions between the LPS and S teams are specified as follows:

$$\forall x \in LPS, Pos(x) \in P(t) \wedge (C_L(t) \cap C_S(t)) \neq \emptyset$$

Actions. The actions available to each team concern: establishing interactions with other teams; modifying the internal structure or implementing an internal coordination algorithm; adding or subtracting a vehicle; or rendezvous with other vehicles or teams, to name just a few. These actions are triggered either by the occurrence of external (uncontrollable events), or when predicates on positions of teams and vehicles and on sets from SpatialTemporalPlaces become true. Some actions take time. For the same team, parallel and sequential execution of actions is allowed.

Plans. A plan specification consists of a description of: 1) initial teams; 2) internal behaviors for each of those teams; 3) interactions among all of the teams; 4) rendezvous places where new teams can be assembled and disassembled, and where the internal behaviors and the interactions among teams can be modified.

4 Control framework

4.1 Introduction

Here, we are not concerned with planning. We assume that a plan is given. We leave this task to the mission specialist: an oceanographer for oceanographic mission; a military officer for the UCAVs. At this point the questions of interest are: 1) is the plan implementable? If so, what is the control framework, and how to design the controllers, and the control structure?

4.2 Models

In the previous two sections, we enumerated the patterns of coordination and control that we consider fundamental to represent team composition and tasking, and expressed them in terms of sets and relations among the elements of those sets. The sets of interest include sets containing entities that evolve in the world (e.g. Vehicles), and sets representing the dynamic behavior of vehicles, or of teams. For example, the set required to evaluate the predicate $\texttt{ReachTemporalSpatialPlaces}(L_1, X_0, X_f)$ is the reach set of the vehicle L_1, when the initial position belongs to the set X_0. This is the key idea behind this specification and control framework:

To express the mission plan in terms of sets describing the evolution and constraints on the evolution of all teams, and of all vehicles within a team, and in terms of the actions taking place at spatial/temporal rendezvous.

The question "is the plan implementable" can be answered when we evaluate all the predicates in the plan. To do this, we need to calculate reach sets (describing the evolution of a dynamic system), invariant sets (describing the locations where the permanence within a certain set is ensured), and solvability sets (describing the locations from which a system can evolve to reach a given set). This way we are able to formally relate design and specification. The question is how to do this? We use duality between functions and sets (see [9]), and the fact that the reachability, solvability, and invariant sets of dynamical systems are the level sets of appropriate value functions (see [12]) to formulate these problems in the framework of dynamic optimization. We do not claim that these problems can be easily solved. In fact, that is not usually the case. However, we believe that this framework provides a valuable understanding of the underlying problems.

4.3 Layered controls

From the control design perspective, and for each team, we interpret the intra-team patterns of coordination and control as problems of control under switching constraints, defined on the product space of the state-space from the constituent vehicles, and target sets, defined on the projection of that space onto the 3D space. We interpret the inter-team patterns of coordination and control as problems with set-valued unilateral or bilateral constraints, and target sets defined on the 3D space. Another aspect of coordination is the negotiation, on-the-fly of temporal and spatial rendezvous. Hence, we must allow for control synthesis on the fly.

We organize design in this framework in terms of the activities of each team: 1) to maintain the integrity of the team and to deliver complex services; 2) to adjust the way services are delivered; 3) to implement motion coordination algorithms (either by coordinating the relative motions of its constituent vehicles, or by attaining some area). The activities concerning the maintenance of the integrity of the team take precedence over the others. This requires a layered structure of controls. The first layer ensures the integrity of the team, the second layer controls service delivery and coordination with other teams, and the third layer concerns team composition tasking, assembling and disassembling teams.

The first layer of control concerns the integrity of the team. This means the satisfaction of the combination rules. Since a set of combination rules indexes multiple configurations, we need to synthesize switching rules among those configurations. The fact that a group can switch between configurations while providing the same service means added flexibility. In terms of switching rules, we are investigating the extension of the work on optimal switching among value functions to this context [2, 1, 14]. We need means of characterizing invariant sets and the corresponding constraints they impose on control, and also means of controlling/changing links in order to establish switch configurations.

Now we are able to turn our attention to the more familiar control problems that have to be solved in this framework. The advantages of defining a framework for the coordinated motion control involving relative and absolute, also designated by collective, dynamics are now clear. For each one of these dynamics, several types of control problems can be considered. As typical examples, we may consider: invariance, reachability, monotonicity w.r.t. a scalar field (see [9, 3, 4, 15]). This way, motion coordination and control patterns can be expressed in terms of simple concepts by combining the different types of relative and collective dynamic control problems.

Coordinated invariance. As we can infer from the above technological and service specifications are mapped onto set-based joint state constraints arising in spaces of relative distances and/or velocities. These constraints are typically described by ellipsoids [10, 11, 12] and/or polyhedra. Formally, we want to characterize the sets of states and controls that maintain the overall system invariant w.r.t. to these constraints. This characterization depends on the information structures, or, to be more precise, on who knows what, and when.

Collective invariance. Motion specification for the group is another example of coordinated invariance, in this case with respect to a set that moves in space. For example, we may want the ensemble of vehicles to stay inside this set – in our scenario the S group is required to stay inside the set $P(t)$. This is another layer of invariance on the top of the previous one.

Coordinated reachability with joint invariance. A set of vehicles may switch among different set-based constraints, information strategies, and coordinated invariance strategies in order to achieve some goal. This goal may be one of reachability in the sense that one, some, or all vehicles are required to reach a certain set. The possibility for reconfiguration makes the system more flexible and hence more adaptable to changes required to attain the target set.

Collective monotonicity w.r.t. a scalar field with joint invariance. A different paradigm of coordination involves using a set of vehicles to descend some scalar field while maintaining some joint invariance. One possible interpretation is that of stabilizing the system that emerges from the composition of several vehicles.

4.4 Mobility of code/controllers

In what concerns controlling/changing links, we have basic entities, such as vehicles and devices, as the building blocks of the team. In our investigations we have

address this issue implicitly with the use of configurations. The configuration implicity defines the conditions under which links among the vehicles can be established, and also the way information is exchanged, i.e., the information structures. But we may want to incorporate in this framework additional services such as `ProvidesController`. This means that some entities may provide controllers to other entities, and also that we may transfer not only control authority, but also control code. We are far from understanding the implications on control theory of these operations: configurations enable structures of control, but we can change configurations to achieve other structures of control. In practice we are talking about dynamic reconfiguration and code mobility. The notion of link, not as a fixed part of the system but as a datum that we can manipulate, is essential for understanding such systems. We are following the evolution of the theories of computation from notions like value, evaluation and function to those of link, interaction and process. The Π-calculus[2], an idealized modelling programming language, is a first attempt in this direction [13]. Specifications and implementations are formalized as process expressions and verification consists in establishing a behavioral relationship between a specification and its implementation. This can be done in the underlying labelled transition system.

5 Conclusions

In this paper we have discussed the challenges to control, computation, and networking posed by new concepts for the coordination and control of networked vehicle systems, and proposed a specification and control framework for these systems. Future work concerns control design in this framework, refining the notions of behavior equivalence, and extending the specification framework to express and manipulate the realm of interactions and processes, namely in what concerns mobility of code and control.

Acknowledgments

The authors thank Professors Pravin Varaiya and Alexander Raja Senputa for stimulating discussions and valuable comments, insights and contributions.

References

[1] Martino Bardi and I. Capuzzo-Dolcetta. *Optimal control and viscosity solutions of Hamilton-Jacobi-Bellman equations*. Birkhauser, 1997.

[2] I. Capuzzo-Dolcetta and L. C. Evans. Optimal switching for ordinary differential equations. *SIAM J. Control and Opt.*, 22(1):143–161, 1984.

[3] F. H. Clarke. A proximal characterization of the reachable set. *Systems & Control Letters*, 27(3):195–7, 1996.

[4] F. H. Clarke and P. R. Wolenski. Control of systems to sets and their interiors. *Journal of Optimization Theory and Applications*, 88(1):3–23, 1996.

[5] D. Van Cleave. Trends and technologies for uninhabited autonomous vehicles. In T. Samad and G. Balas, editors, *Software-Enabled Control: Information Technology for Dynamical Systems*. IEEE Press/John Wiley and Sons, 2002.

[6] J. Borges de Sousa, A. Girard, and K. Hedrick. Real-time hybrid control of mobile offshore base scaled models. In *Proceedings of the American Control Conference*, 2000.

[7] J. Borges de Sousa and F. Lobo Pereira. A generalized vehicle-based control architecture for multiple auvs. In *Proceedings of the OCEANS '95 MTS/IEEE*, pages 1643–50. IEEE, 1995.

[8] J. Borges de Sousa and Raja Sengupta. Tutorial on autonomous and semi-autonomous networked multi-vehicle systems, decision and control conference. December 2001.

[9] F. H. Clarke et. al. *Nonsmooth Analysis and Control Theory*. Springer, 1998.

[10] A. N. Krasovskii. *Control under lack of information*. Birkhauser, 1995.

[11] A. B. Kurzhanskii. *Advances in nonlinear dynamics and control : a report from Russia*. Birkhauser, 1993.

[12] A. B. Kurzhanskii. *Ellipsoidal calculus for estimation and control*. Birkhauser, 1997.

[13] Robin Milner. *Communicating and mobile systems : the* Π*-calculus*. Cambridge University Press, 1999.

[14] F. Lobo Pereira and J. Borges de Sousa. A differential game with graph constrained dynamic switching strategies. In *Proceedings of Decision and Control Conference*. IEEE, 2001.

[15] E.D. Sontag and Y.S. Ledyaev. A lyapunov characterization of robust stabilization. *Journal of Nonlinear Analysis*, 37:813–840, 1999.

[16] P. Varaiya. Smart cars on smart roads: problems of control. *IEEE Transactions on Automatic Control*, 38(3):195–207, February 1993.

[17] P. Varaiya, T. Simsek, and J. Borges de Sousa. Communication and control in hybrid systems. In *Tutorial session for American Control Conference 2001*, 2001.

[2] The Π-calculus is a mathematical model of processes whose interactions change with time. This calculus inherits concepts and elements from a variety of process algebraic theories that have a formal algebraic basis and a behavioral semantics.