Agents Coordination in Flat Hierarchical Society-oriented Systems

Mihaela R. Cistelecan

Technical University of Cluj-Napoca, str. Daicoviciu 15, RO-3400, Cluj-Napoca, Romania

Abstract: The paper proposes a framework for engineering a coherent society-oriented system using the concept of sliding-mode control from systems theory. The agents are coordinated through local control laws using a switching logic related to the sliding manifolds. The sliding manifolds stand for sub-plans and therefore they are updated every time a change in planning is required. The agents have the ability of self-adaptation by adjusting the structure of the sliding manifolds. The design is performed using tools from the algebraic framework. We also address the problem of deriving the sliding manifolds from the given specifications.

1 INTRODUCTION

Nowadays, a topic of great interest for many different research areas is the study of the distributed systems. The outstanding feature of these systems is that their architecture looks like a network consisting of many different interacting entities. The control systems and computer science communities pursue different ways in their study of the distributed systems. The approaches from the perspective of the computer science emphasize the agent behaviour and give an analysis of the outcome of the system. The (decentralized) decision making system and interaction protocols are often given in a very complex form and the outcome is a descriptor of the "adequate guess" of the decision. These approaches are important because they always deal with complex agents - rational agents - that make decisions based on logical assertions or plans. Mathematically, the behaviour of the rational agents, may be described by either discrete event systems (DES) or hybrid systems, (Ortiz et al., 2001), (Antsaklis and Passino, 1993). Since the interaction among entities is based on social rules these systems are most often called either social systems or society-based systems (SoS). Aside from the fact that the interaction is based on social rules, often the agent is not the only type of entity that is encountered in SoSs. In SoSs the interaction may be represented through different kinds of action, such as cooperation, collaboration, competition, negotiation,

communication. Sometimes, the interaction histories are also important, (Sen and Dutta, 2002). While the analysis problem for SoS is quite well developed, the synthesis problem still poses many challenges. On the other hand, the approaches from the perspective of the control systems community deal with not so sophisticated agents, no planning is implemented by agents, but the synthesis problem is successfully solved using tools from control theory. The systems that these approaches deal with are generally called multi-agent systems. The features revealed by the two kinds of approaches may also be noticed by comparing (Fisher and Wooldridge, 2003) and (Yan et al., 2003).While the technological systems that obey social laws are dynamical systems and therefore they should deal with traditional concepts like equilibrium and system trajectory still, embedding the planning action into their dynamics is not a trivial matter.

In this paper, we do not dwell on the philosophical aspects of the social systems, but stick to the idea of engineering a system that eventually accomplishes the given task. Therefore, we are interested only in the synthesis problem. Our research tries to develop a design procedure for SoSs by using concepts from the perspective of dynamical systems theory. The basic idea of the proposed framework, essentially coming from (Cistelecan, 2004a), is that the coordination in a SoS should be made dependent on the dynamics of the interaction among entities. Therefore, we are in favor of a systematic off-line development

of a coordination mechanism implying the following: 1. a decentralized control law that take into account the interaction among entities 2. formalization of the coherence problem as a stability problem; 3. a distributed planning induced by the sliding manifolds. In our research we transform the hybrid systems that describe the agents behavior into dynamic polynomial systems. The purpose of this paper is to extend the analysis of the framework already introduced in (Cistelecan, 2004b) by investigating thoroughly the problem of deriving the sliding manifolds from the problem specifications.

This paper is organized as follows. In Section 2 we investigate the issues that are important for the SoSs design. In Section 3 we show the mathematical model of the agent. In Section 4 we introduce the idea of a nominal control system for the proposed framework. In Section 5 we introduce the control law as a sliding mode control law. In the next two sections we explain the basic ideas by using an example. In Section 8 we give some preliminaries for the problem of self-adaptation, leaving for a next paper a more detailed mathematical formalization. We conclude the paper by Section 9.

2 COHERENCE AND GOAL CODING

The flat hierarchical (one-level) SoSs do not benefit from an external supervisor, instead, some agents, the "leaders" or "managers", have the authority to control indirectly the whole dynamics by directly controlling the entities they interact with. A "leader" may influence "ordinary" agents but remains insensitive to them. A leader agent has its own dynamics that can be prescribed or modified only by an external source. For example, for a task of collecting objects, if the working area is to be changed a leader should decide when and where to move next the whole group of agents. Obviously, by concatenating many flat hierarchies a multi-level hierarchy could be obtained. If the "leaders" embedded on a level of the hierarchy are given references from a higher level, then the decision from the higher level is refined at the lower level.

When designing SoSs the following four issues are important: 1. *How to represent the agent behaviour by a comprehensive model;* 2. *How to model the goal to be accomplished by the SoS;* 3. *How to model the interactions;* 4. *How to infer the appropriate decision so that eventually the SoS achieves the goal.*

Two observations are in place here. First, inside a SoS the agents functionality is often different. An example taken from the RoboCup game shows that a goalkeeper and a player obey different assignments of the defend-attack protocol. Secondly, a SoS is composed of many different entities, some of them - the active entities or agents - are able to take decisions, some of them - the passive entities - just influence the decisions taken by the active entities. Examples of passive entities are: landmarks, obstacles, queues, (non-intelligent) targets on the military field, ball, gate (RoboCup), interrogated global data-bases.

Despite the difference in their protocol and the type of passive entities the SoS consists of, the agents have to implement a (decentralized) decision making process that guarantees the SoS coherence. The (coordination) coherence of a SoS is related to its ability to reach the goal state or to accomplish the given task. Generally, the goal state is given as a set of predicates on some of the SoS entities - the tactical entities. For example, if a SoS contains agents, obstacles and landmarks the specifications related to the target formation are given only with reference to agents and landmarks but not to obstacles. It is intuitive that in a general sense the formation can be stated as a collection of relations, $\mathcal{R}_{ij}$, among the tactical entities. Thus, the ultimate goal the SoS should achieve could be coded into the SoS structure through a target formation, that imposes the desired relations, $\mathcal{R}_{ij}^d$, among the tactical entities.

Beside the ultimate goal of a problem it might be necessary to impose partial goals or strategies for each agent. From a mathematical point of view these are restrictions imposed to the agents behaviour. Partial goals are often dealing with some order relations and generally they require sub-plans in order to be accomplished. For example, the ultimate goal for the RoboCup game may be coded as a zero distance between the ball and the adversary's gate. The partial goals are related to the game strategy. Note, for example, that if the ball is surrounded by many players from both teams, a strategy should be defined so that the ball is made attractive only for two adverse players and repulsive for all the others.

The interactions among entities may be described mathematically by relations among them. If the ultimate goal is coded into the SoS structure as a target formation the decision making system should always minimize the misfit measure $(e_{ij}, q_{e_{ij}}) = \mathcal{R}_{ij}^d \asymp \mathcal{R}_{ij}$ between the desired, $\mathcal{R}_{ij}^d$, and actual, $\mathcal{R}_{ij}$ relations; $\asymp$ stands for a comparison operator. Stated in this manner, the SoS problem of goal achieving resembles the well-known regulator (tracking) problem from the control systems theory if $\mathcal{R}_{ij}^d$ are time invariant (time-varying). Moreover, the coherence concept defined for SoSs becomes equivalent to the stability concept from control systems theory. Thus, if the dynamical system of the misfit measure is stable and converges towards the origin, the SoS behaves coherently and consequently, the goal is achieved. When the communication among agents is only partial, the "partial information control" can be stated as given by the ro-

bust control framework that is formalized by the control theory.

3 THE MODEL OF THE AGENT

From a philosophical point of view, in our approach the agent consists of three functional components: Eye, Mind and Body.

The Eye has the functionality of some generalized sensors, that is the unification of physical and logical sensors. Therefore, the Eye stands for the ability to detect, perceive or "see" the neighboring entities. Like in (Cistelecan, 2004a) the agents are allowed to "see" each other when some conditions are met and the pooling of the required conditions is the task of the generalized sensors. The conditions are given in relation to the misfit measure $(e_{kj}, q_{e_{kj}})$ between the entities k and j:

$$
\begin{aligned}
(e_{kj}, q_{e_{kj}}) = (g_1(x_k, q_k, x_j, q_j, w_{kj}, q_{d_{kj}}), \\
g_2(x_k, q_k, x_j, q_j, w_{kj}, q_{d_{kj}}))
\end{aligned} \quad (1)
$$

where g_1 and g_2 are polynomials and the target formation is described through $R^d_{kj} = (w_{kj}, q_{d_{kj}})$.

We can notice that the scheduling system - dynamical and distributed - is intrinsic to our control framework: as soon as a generalized sensor fires, the synchronization (communication) with the perceived tactical entity (agent, respectively), may be initiated.

The Ag_k's Eye may be as follows:

$$
z_k^+ = Z_k(z_k, e, q_e, \delta) \quad p_k(t) = P_k(e, q_e, z_k, \delta) \quad (2)
$$

where $(e \in R^n, q_e \in (Z/_{pZ})^l)$ is the misfit measure gathered from all SoS entities and q_e is a vector of variables in the Galois field with p elements $\{0, ..., p-1\}$, with the usual multiplication and addition modulo p, where p is prime, (Gunnarsson and Plantin, 1998).

It is possible to introduce temporal relations through "clock" variables. Two temporal relations often used in distributed systems are "agent k communicates with agent j within δ_t time units and "agent k does some activity after agent j completes some activity". For the first example, if the communication is controlled through the "clock" variable q_{clk}, the logical error variable in the form $q_e = \delta_t - q_{clk}$ or $q_e = min\{\delta_t, q_{clk}\}$ should influence any other variable involved in the communication process. For the second example, a possibility is through the following variables:

$$
q_{e_k} = o_j; \quad z_k^+ = Z_k(z_k, q_{e_k}); \quad p_k = z_k
$$

where o_j is a binary signal that informs Ag_k about the state of activity performed by Ag_j, and Z_k should be interpolated from the table that describes the possible histories of the variables o_j and z_k.

The Mind implements the interaction among entities and also the planning activity. This component is mathematically modeled as a sliding manifold together with a method for adjusting the structure of the sliding manifold. The planning capability is due to embedding of the logic components into the sliding manifold. The partial goals, strategies and sub-plans are coded into the sliding manifold.

As defined in the control systems theory the sliding mode control is implemented for a continuous dynamical system $\dot{x} = f(x, u)$, $x \in \mathcal{R}^n$, $u \in \mathcal{R}^m$ as the following discontinuous control law: $u_i = u_i^+(x)$ if $s_i(x) > 0$ and $u_i = u_i^-(x)$ if $s_i(x) < 0$, where $s_i(x) = 0$ is the sliding manifold for the control law u_i and $i = 1, .., m$. The stability of the dynamical system is guaranteed by either the reaching condition $s_i \dot{s}_i < 0$, $i = 1, .., m$ or the Lyapunov type condition $\sum_i d_i s_i \dot{s}_i < 0$, $d_i > 0$. The sliding mode regime is described by $s_i(x) = 0$.

In the proposed framework the decision is taken, as is customary in the conventional sliding mode control, so that the system state approaches the sliding manifold and through it the goal state. The Mind adapts the structure of the sliding manifold and the control law according to the environment, as will be detailed in Section 8. The Ag_k's Mind is modeled as a sliding manifold that aggregates the two vectors that consist of e_{kj} and $q_{e_{kj}}$. We assume the sliding manifold on the following form:

$$
\begin{aligned}
s_k(t) = \sum_{j=1}^{N_k} c_{kj} \mathcal{S}_{kj}(e_{kj}, q_{e_{kj}}) = \\
= \sum_{j=1}^{N_k} c_{kj} h_{kj}(q_{e_k}) e_{kj} = 0
\end{aligned} \quad (3)
$$

where $q_{e_k} \in Z_p[q_{e_{kj}}...q_{e_{ki}}...]$ and N_k is the number of entities perceived by Ag_k. The coefficients c_{kj} are estimated in the off-line design stage so that the whole dynamics is stable. The mappings $h_{kj} : (Z/_{pZ})^{l_k} \rightarrow Z$ are also estimated off-line so that the sliding manifolds implement the required strategies or sub-plans. The mappings h_{kj} should make the entities j attractive/ repulsive to some degree for Ag_k.

The Body stands for the action the agent has to take. Actions may be motion related like in robotics but they may also be in the form of message communications or data-bases interrogation, e.t.c. The Ag_k's Body aggregates a continuous part and a logical part, as follows:

$$
\dot{x}_k = F_{1_k}(x_k, q_k, u_k, v_k) \quad q_k^+ = F_{2_k}(x_k, q_k, u_k, v_k) \quad (4)
$$

where F_{1_k} and F_{2_k} are vectors of polynomials, u_k and v_k are the control variables for the continuous and discrete dynamics, respectively, $x_k \in R^{n_k}$, and $q_k \in (Z/_{pZ})^{l_k}$ is a vector of variables in the Galois field. The passive entities dynamics is as given in (4), with $u_k(t) = 0$ and $v_k(t) = 0$.

4 NEIGHBORHOODS, ALLIANCES AND COALITIONS

Since in the technological world "intelligence" is a synonym for "complexity solver", when referring to SoSs the equivalence between intelligent behaviors and DES / hybrid systems is not exactly true. The so called "understanding of a new situation" or "refining the knowledge" are not implicitly modeled into the DES / hybrid system framework unless the agents are enriched with some further abilities. These abilities should imply robustness and consequently they should require that the SoS behave appropriately when faced with tasks and situations that were not explicitly considered during the off-line design. Therefore, the agents have to "adjust" their behaviours according to the real, actual situation. For example, when the task of some agents is to collect objects / consume resources the number of objects / resources is not exactly known, in advance, during the off-line design stage. Siltt, it is obvious that the appearance / disappearance of entities (objects, agents, processes, tasks) is strongly related to coherence. For these cases, a wise SoS should behave coherently while adapting on-line some of its parameters. We refer to this problem in Section 8.

We introduce some concepts that help us explaining the manner the agents self-adjust their behaviour as a function of the environment. Using the Eye component the δ-neighborhood, of an agent Ag_k, $\mathcal{N}_\delta\{Ag_k\}$, is defined as the set of the entities that are perceived by the agent Ag_k at the time t. The vector δ contains the firing thresholds of the generalized sensors. Since both the number and type of entities that an agent perceives are non-deterministic, it is hard to imagine at the design time all the possible configurations of a neighborhood. On the other hand, there are some special situations, related to the required strategies or special orders of actions that should be taken into account at the off-line design stage. This special situations are described through special neighborhoods that we call either alliances or coalitions depending on the team concept. Inside a team the agents collaborate for solving a problem. Therefore, for single-team-like SoSs, such as robots collecting objects, resource allocation problems, distributed sensor networks, fault-tolerant software systems, our framework works with alliances. Given for a minimal or strategic neighborhood, the δ-alliance of an agent Ag_k, $\mathcal{A}_\delta\{Ag_k\}$ is defined as the set of the tactical entities that are perceived by the agent Ag_k. For multi-adverse-team-like SoSs such as the RoboCup or CaptureTheFlag games, multi-team military operations, immune systems, beside alliances our framework works with coalitions. A coalition is defined as the set of tactical entities coming from all teams. The concepts of alliance and coalition are intended to be used in the off-line design stage in order to get a nominal control system from the problem specifications. The nominal control system should be seen here as a "minimal" (or abstracted) control system derived off-line for alliances and coalitions, that may be extended on-line in order to deal with neighborhoods. The Mind component is responsible for expanding the control given for an alliance / coalition to the control required by a neighborhood.

Similar to (Cistelecan, 2004a), in the proposed control framework, the agents are coordinated through sliding manifolds. Since the agents functionality is limited to a limited horizon by the Eye component, the SoS dynamics emerges as an autonomous switching system. The switching depends on the environment and for this reason no switching ordering should be imposed by the off-line design. Note that the sub-plans interleaving is equivalent to switching through neighborhoods, changing alliances or traditionally said, mode switching. Moreover, the sliding manifolds are tools for the "negotiations" accomplishment. Since each agent is allowed to interact only with its neighbors, the negotiation of each agent is based on its local perspective like in AI-based approaches. The importance of a temporary alliance and the difficulty to regulating threads of machine intermingled agents as they form and disolve is observed also by (Klavins and Koditschek, 2000). Concepts like "overlapping coalitions of agents" and "each agent negotiates based on its local perspective" are also defined in (Sims et al., 2003).

5 THE CONTROL LAW DESIGN

In the proposed SoS, the decision an agent should take is given by a sliding mode control law. The control system is transformed into a dynamical polynomial system. Using tools from differential algebra the specifications given, both for the continuous dynamics and logical components, are aggregated. Sliding mode for hybrid systems was already formalized in (Dogruel et al., 1996). Polynomial dynamical systems have already been used to model and / or control hybrid systems, (Forsman, 1994), (Marchand and LeBorgne, 1998), (Gunnarsson and Plantin, 1998). Sliding mode control design using tools from differential algebra is given in (Fortell, 1995).

The decision making system uses a sliding mode control law, as follows:

$$u_k = K_{1_k}(e_k, q_{e_k})tanh(s_k)$$
$$v_k = K_{2_k}(e_k, q_{e_k})tanh(s_k) \quad (5)$$

The dynamical system of the misfit measure for the SoS is as follows:

$$\dot{e} = F(e, q_e, u, v) \quad q_e^+ = \mu(e, q_e, u, v) \quad (6)$$

where $u = [u_k]_{k=1..N}$, $v = [v_k]_{k=1..N}$ and N is the number of agents. The system (6) was obtained by appending the dynamics of the misfit measure $[\begin{array}{cc} \dot{e}_k & q_{e_k}^+ \end{array}]$ for all N agents and also the algebraic constraint given by the overlapping variables.

The nominal SoS structure (see Section 4) is developed off-line. The decentralized control laws and the sliding manifolds are estimated off-line so that when they work together on-line the coherence of the SoS is guaranteed. The algebraic framework has been chosen in order to get a parametric control solution. The concepts of alliance and coalition are used in the off-line design stage in order to get the nominal control system from the problem specifications. The Mind component adds, on-line, extra-terms both to the sliding manifolds and control laws while coherence is preserved.

Note that the error vector in (6) contains both continuous and logical variables. Since we are interested in a possible practical implementation of the proposed control system using available computational tools, we choose to transform the non-homogeneous error vector into a homogeneous one containing only continuous time variables. This is possible by a transformation of the error vector as follows:

$$\begin{array}{c} [\begin{array}{cc} \dot{e} & q_e^+ \end{array}] \longrightarrow \dot{\omega}^T = [\begin{array}{ccc} \dot{e} & \dot{q}_e & \dot{p} \end{array}] \\ \omega^T = [\begin{array}{ccc} e & q_e & p \end{array}] = [\begin{array}{ccc} \omega^1 & \omega^2 & \omega^3 \end{array}] \\ = [\begin{array}{ccc} [\omega_k^1] & [\omega_k^2] & [\omega_k^3] \end{array}]_{k=1,..,N} \end{array} \quad (7)$$

where a logical variable z^+ given as $z^+ = Z(z,e)$ is replaced by the continuous time derivative $\dot{z} = (Z(z,e) - z)p$, $p = \sigma((Z(z,e) - z)^2)$, and $\sigma(x)$ is the sigmoid function. Thus, the off-line design works not with (6) but with the polynomial system:

$$\dot{\omega} = f(\omega, u, v) \quad (8)$$

For the nominal SoS structure the Lyapunov function is chosen in the following form:

$$V_{\mathcal{A}}(\omega, \eta) = (s^{\mathcal{A}})^T P(\eta) s^{\mathcal{A}} + K_1^T K_1 + K_2^T K_2 \quad (9)$$

where:

$$\begin{array}{c} K_i^T = [K_{i_k}(\omega_{k_i})]_{k=1,N}, \quad i = 1, 2, \\ (s^{\mathcal{A}})^T = [s_1^{\mathcal{A}}(\omega_1), ..., s_N^{\mathcal{A}}(\omega_N)] \end{array} \quad (10)$$

N is the number of agents, and $s_k^{\mathcal{A}}$, as given by (3), are constructed for alliances (and coalitions). The superscript $\mathcal{A}$ stands for alliance in a generic sense. More precisely, the nominal SoS structure should work with many different alliances: $\mathcal{A}_1, ..., \mathcal{A}_s$. Each alliance $\mathcal{A}_j$ induces a different manifold $s_k^{\mathcal{A}_j}$. We assume that for the set $\{s_k^{\mathcal{A}_1}, ..., s_k^{\mathcal{A}_s}\}$ there exists a generic $s_k^{\mathcal{A}}$ where polynomials S_{kj} are gathered from all $s_k^{\mathcal{A}_1}, ..., s_k^{\mathcal{A}_s}$. Then, for each alliance the sliding manifold is obtained by deleting some terms from the generic sliding manifold, $s_k^{\mathcal{A}}$.

6 EXAMPLE: AGENTS COLLECTING OBJECTS

In this section we show through an example that the sliding manifolds can be used as tools for accomplishing cooperation and collaboration in SoSs. We mean by collaboration an explicit exchange of information.

For a group of agents collecting objects, the ultimate goal could be coded as a zero distance between any agent and any object (no object on the field) and, in order to assure the coverage of the whole area, a given distance among agents. The ultimate goal is given through R_{kj}^d, as follows:

$$\begin{array}{c} (BT_k = F \bigwedge BT_j = F) \bigwedge \\ \bigwedge (BK_k = E \bigwedge BK_j = E) \bigwedge \\ \bigwedge (\neg CO_k \bigwedge \neg CO_j) \bigwedge (\rho_{kj}^a = \rho_{jk}^a = w_{kj}) \end{array} \quad (11)$$

where $k, j = 1, .., N$, $k \neq j$, N is the number of agents, BT, BK stand for the battery and basket, respectively, ρ_{jk}^a and w_{jk} are the actual and desired distances between agents k and j, and $\neg CO$ stands for 'no object on the field'. As an example, we assume that the battery and basket may be full (F/2), empty (E/0) or nonempty (NE/1).

Since each agent has to collect many objects, the agent should reason about the order of approaching the objects. It is intuitive that this order should be based on the distances between agents and objects, ρ_{kj}^o, but sometimes some other important considerations change this order. For example, note that the ultimate goal can be accomplished only if some passive entities like reloading-stations (RS) and storehouses (SH) are introduced into the SoS. Thus, the battery can be loaded if it is empty and the basket can be cleared when it is full. When the battery is empty or the basket is full the agent must go to RS or SH respectively, ignoring the objects that are on its way. On the other hand, a well designed SoS should allow the agent, in non-critical situations, to charge its battery or clear the basket, while collecting the objects encountered on its way. Moreover, the collaboration among agents should allow a change of the order the objects are collected.

If the actions required by agents are: collect objects (CO), keep distance from agents (KD), approach agent (AA), approach reloading-station (AR), approach storehouse (AS), look for a RS (?RS), look for a SH (?SH), some of the rules that have to be wired into the sliding manifolds are the following:

$$\begin{array}{c} (BT = F, BK = E) \longrightarrow (CO, KD) \\ (BT = NE, BK = E) \longrightarrow (CO, KD, ?RS) \\ (BT = NE, BK = NE) \longrightarrow \\ \longrightarrow (CO, KD, ?RS, ?SH) \\ (BT = E) \longrightarrow (?RS, AR, AA/AR) \\ (BK = F) \longrightarrow (?SH, AS, AA/AS) \end{array}$$

$$(12)$$

where AA/AR and AA/AS have the meaning: approach the RS or SH that is in the neighborhood of another agent. When $q_{req}^{RS/SH} = 1$ Ag_k asks the neighboring agents (if any) if there is a RS/SH in their neighborhood. If the answer is positive, $q_{ack}^{+} = 1$ and e_{ack}^{+} is given the smallest distance to an agent that has the RS/HS in its neighborhood. Otherwise $q_{ack}^{+} = 0$ and Ag_k starts to wander based on a given constant ϖ.

If the logical state of an agent implies $(BT = NE, BK = NE)$ and both RS and SH are in the $\mathcal{N}_\delta\{Ag_k\}$ some logical variables $q_{e_{kj}}^{RS/SH} = G_k(BT, BK, v_k) \in (Z/_{3Z})$, decide an order of approaching the entities. We give a simpler, but not so flexible, solution to this case in the next section.

The agents may implement a collaboration if they are willing to help each other in collecting objects. In order to implement collaboration each agent Ag_k should estimate a logical error variable $q_{e_{kj}}^{coll} = max(q_k, q_j)$, where the value of q_j should be communicated by Ag_j, as it is not visible. The variable q_k that should be appended to the Ag_k's Body:

$$\dot{x}_k = Ax_k + Bu_k, \quad A \in \mathcal{R}^{2 \times 2}, B \in \mathcal{R}^{2 \times 1} \quad (13)$$

is as follows:

$$q_k : (Z/_{3Z})^3 \longrightarrow (Z/_{3Z})$$
$$q_k^{+} = Q_k(WE, BT, BK) \quad (14)$$

where WE is related to the number of objects in $\mathcal{N}_\delta\{Ag_k\}$ and stands for the working effort (0 - no object to collect, 1 - a reasonable number of objects, 2 - many objects to collect). A possible assignment for q_k is as follows: $Q_k(0,2,0) = 0, Q_k(1,2,0) = 1, Q_k(0,2,1) = 1, Q_k(2,2,3) = 2, Q_k(2,2,0) = 2$, where

$$
\begin{array}{lll}
q_k = 0 & means & Ready\&ReadyToHelp \\
q_k = 1 & means & \begin{bmatrix} NotReady\& \\ DoNotNeedHelp\& \\ CouldHelp \end{bmatrix} \\
q_k = 2 & means & NotReady\&NeedHelp
\end{array}
$$

If the sliding variable s_k is made dependent on $q_{e_{kj}}^{coll}$ the agents may agree to collaborate.

The sliding manifold implementing these rules for the $\mathcal{N}_\delta\{Ag_k\}$ is as follows:

$$s_k(t) = \sum_j c_{kj}^o h_{kj}^o e_{kj}^o + \sum_j c_{kj}^a h_{kj}^{coll} e_{kj}^a +$$
$$+ \sum_j c_{kj}^{RS/SH} h_{kj}^{RS/SH} e_{kj}^{RS/SH} + \sum_j c_{kj}^a h_{kj}^a e_{kj}^a +$$
$$+ h_1^{ack} e_{ack}^{RS/SH} + h_2^{ack} \varpi = 0 \quad (15)$$

where: $h_{kj} : (Z/_{pZ})^{l_k} \longrightarrow Z$, e_{kj} is the error between Ag_k and the entity j that belongs to $\mathcal{N}_\delta\{Ag_k\}$, c_{kj} are the sliding mode coefficients, the superscripts "o", "RS/SH" and "a" stand for objects, RS/SH and agents, respectively. The order of approaching the

neighboring entities is controlled through the mappings h_{kj} that depend on some logical variables q_{e_k}. Moreover, if an agent is willing to give some help only after "sleeping" for δ_t minutes then a logical error variable depending on the "clock" variable should influence the mappings h_{kj}^{coll}.

7 AGENTS MAKING SUB-PLANS

The ordering relations, strategies and sub-plans are wired into the sliding manifolds during the off-line design stage. The mappings h_{kj} are estimated for alliances during the off-line stage. Let us assume that Ag_k is in a logical state characterized by $(BT = NE, BK = NE)$ and its neighborhood consists of a RS, a SH and an object: $\mathcal{A}_\delta\{Ag_k\} = \{RS, SH, o\}$. A coherent behaviour of the agent would imply a sequence of plans so that if, for example, at the beginning it is closer to RS than to SH it heads to the RS while collecting the objects it encounters on its way. In order to explain how sub-plans are constructed we assume a set of sub-plans $\{P1, P2, P3\}$ that stands for the following configurations (partitions):

$$
\begin{array}{ccccc}
\mathcal{C}_1 & |e^o| < |e^{RS}| < |e^{SH}| & & \rightarrow & P1 \\
\mathcal{C}_2 & |e^{RS}| \geq |e^o| \wedge |e^{RS}| < |e^{SH}| & & \rightarrow & P2 \\
\mathcal{C}_3 & |e^{RS}| > |e^{SH}| \wedge e^o = 0 & & \rightarrow & P3
\end{array}
$$
$$(16)$$

where e^{RS}, e^{SH}, e^o are the errors between Ag_k and RS, SH and object, respectively. In order to track the change of configuration some logical variables should be appended to the Ag_k's Body, as follows:

$$q_{k_1} \vdash |e^{RS}| \sim |e^{SH}|; \quad q_{k_2} \vdash |e^o| \sim |e^{SH}|;$$
$$q_{k_3} \vdash |e^{RS}| \sim |e^o|; \quad [BK = 2]; \quad [BT = 2]$$
$$(17)$$

where $\sim$ stands for $\{>, <, =\}$ and $q_{k_i} \in Z/_{3Z}$. For the configuration $\mathcal{C}_1$ the sub-plan $P1$ implements an aggregated movement towards the RS and object. If this movement does not disrupt the movement closer to the RS then the object is collected. After the object was collected, if the evaluation of the field shows that the distance to RS is higher than the distance to SH then, a switch from $P1$ to $P3$ is done. On the other hand, if the movement closer to the object implies at a moment that the agent gets farther from the RS then the sub-plan $P2$ should replace $P1$. The switching among sub-plans is required when a divergent movement of the sliding variable is encountered. For example, when the agent starts to go away from the RS while approaching the object the sliding mode reaching condition $s_k \dot{s}_k < 0$ should not be fulfilled any longer. Consequently, a new sub-plan is required in order to correct the agent dynamics by implementing a repulsive field around the object. Notice that for each sub-plan the stable sliding mode dynamics

should implement a movement of the system closer to the goal. For the configurations given above the sliding manifolds should be implemented on the following form:

$$
\begin{aligned}
P1 &:: & s_k &= X_a^{RS} e^{RS} + X_a^o e^o + X_r^{SH} e^{SH} = 0 \\
P2 &:: & s_k &= X_a^{RS} e^{RS} + X_r^o e^o + X_r^{SH} e^{SH} = 0 \\
P3 &:: & s_k &= X_r^{RS} e^{RS} + X_a^{SH} e^{SH} = 0
\end{aligned}
$$
(18)

where X_a, X_r stand for the variables that induce an either attractive or repulsive movement, respectively. Note that X_a, X_r are generic coefficients that represent multiplications of the mappings h_{kj} by the sliding mode coefficients c_{kj}.

A quantified formula written in the theory of reals (Jirstrand and Glad, 1997) can describe the switching of sub-plans. For example, the transition $P1 \mapsto P2$ can be described, as follows:

$$
\begin{aligned}
\exists e_1^{RS}, e_1^{SH}, e_1^o &\in [-\delta, \delta], \\
\exists e_2^{RS}, e_2^{SH}, e_2^o \in [-\delta, \delta], \exists e_3^{RS}, e_3^{SH}, e_3^o &\in [-\delta, \delta], \\
\varphi &\models \\
((s_{k_1} > 0 \wedge \mathcal{C}_1) \Rightarrow (s_{k_2} > 0 \wedge \mathcal{C}_2 \wedge s_{k_2} > s_{k_1})) & \\
\vee((s_{k_1} < 0 \wedge \mathcal{C}_1) \Rightarrow (s_{k_2} < 0 \wedge \mathcal{C}_2 \wedge s_{k_2} < s_{k_1})) & \\
\Rightarrow (s_{k_2}^+ > 0 \wedge s_{k_3} > 0 \wedge s_{k_2}^+ > s_{k_3} \vee & \\
\vee s_{k_2}^+ < 0 \wedge s_{k_3} < 0 \wedge s_{k_2}^+ < s_{k_3} \vee s_{k_2}^+ s_{k_3} < 0) &
\end{aligned}
$$
(19)

where

$$
\begin{aligned}
s_k(t_1) = s_{k_1} &= X_a^{RS} e_1^{RS} + X_a^o e_1^o + X_r^{SH} e_1^{SH} \\
s_k(t_2) = s_{k_2} &= X_a^{RS} e_2^{RS} + X_a^o e_2^o + X_r^{SH} e_2^{SH} \\
s_k^+(t_2) = s_{k_2}^+ &= X_a^{RS} e_2^{RS} + X_r^o e_2^o + X_r^{SH} e_2^{SH} \\
s_k(t_3) = s_{k_3} &= X_a^{RS} e_3^{RS} + X_r^o e_3^o + X_r^{SH} e_3^{SH}
\end{aligned}
$$
(20)

where the variables $X_a = c_{\mathcal{A}} h_a(\mathcal{C}) = c_{\mathcal{A}} k_a \zeta_a(\mathcal{C})$ and $X_r = c_{\mathcal{A}} k_r h_r(\mathcal{C}) = c_{\mathcal{A}} k_r \zeta_r(\mathcal{C})$, $\zeta_a(\mathcal{C}) = 1$, $\zeta_r(\mathcal{C}) = -1$, $c_{\mathcal{A}}$ are the sliding mode coefficients that should be estimated in the off-line design stage so that the dynamics of the whole distributed system is stable. The variables k are estimated from the relation that results after the quantifier elimination and cylindrical algebraic decomposition act on (19).

If a neighborhood contains two objects instead of one object as was wired into the sliding manifold using the alliance given above, $\mathcal{N}_\delta\{Ag_k\} = \{RS, SH, o1, o2\}$, the variables X_a and X_r change their values only because of the new computed sliding mode coefficients, $\hat{c}$, as will be shown in Section 8. The new sliding manifold used by the agent Ag_k in order to accomplish its goal is as follows:

$$
\begin{aligned}
s_k(t) = &\hat{c}^{RS} k^{RS} \zeta^{RS}(\mathcal{C}) e^{RS} + \hat{c}^{o1} k^{o1} \zeta^{o1}(\mathcal{C}) e^{o1} + \\
&+ \hat{c}^{o2} k^{o2} \zeta^{o2}(\mathcal{C}) e^{o2} + \hat{c}^{SH} k^{SH} \zeta^{SH}(\mathcal{C}) e^{SH}
\end{aligned}
$$
(21)

When the problem specifications require that every alliance contain at least two agents, the alliance given above has to be changed so that the most stringent condition is that the two agents perceive each other.

Therefore, the sliding manifold for the agent Ag_k must contain a term referring the distance between the two agents. The peer agent may be attractive or repulsive like any other entity in the neighborhood of Ag_k. From the above given analysis it is obvious that the mappings $\zeta(\mathcal{C})$ are very important. Still, their computation is not trivial and we have to investigate this topic thoroughly. An important feature of the mappings $\zeta(\mathcal{C})$ is that they have to capture a given order of the logical variables. For example, if we study an alliance of the form $\mathcal{A}_\delta\{Ag_k\} = \{RS, SH, o1\}$ the mappings should be computed using the following ordering for the logical variables: $BT \succ BK \succ \{q_{k_1}, q_{k_2}, q_{k_3}\}$ The importance degree of $\{q_{k_1}, q_{k_2}, q_{k_3}\}$ depends on the configuration and therefore it should be properly revealed by the mappings h_{kj}.

8 ON-LINE SELF ADAPTATION

When dealing with neighborhoods the following Lyapunov function should be considered:

$$
\begin{aligned}
V_{\mathcal{N}}(\omega, \eta) = V_{\mathcal{A}} + V_r = \\
= (s^{\mathcal{N}})^T P(\eta) s^{\mathcal{N}} + (K_1^{\mathcal{N}})^T K_1^{\mathcal{N}} + (K_2^{\mathcal{N}})^T K_2^{\mathcal{N}}
\end{aligned}
$$
(22)

where: $s^{\mathcal{N}} = s^{\mathcal{A}} + s_r$ and $s_r^T = [s_{r_k}(\omega_k)]_{k=1,..,N}$ is a vector of polynomials. It is to be noticed that there exist a difference in the energy given by the Lyapunov functions in the cases of alliances and neighborhoods. This is due to the entities the neighborhoods contain but alliances do not. Therefore, the off-line design provides a control law that can be extended online so that to take into account all the entities of a neighborhood. The extension is possible because of a reserve of energy that is foreseen for a neighborhood in relation to the generic alliance. The control laws for an alliance and a neighborhood are: $u_k^{\mathcal{A}} = K_{1_k}(\omega_k) tanh(s_k^{\mathcal{A}})$, $v_k^{\mathcal{A}} = K_{2_k}(\omega_k) tanh(s_k^{\mathcal{A}})$, $u_k^{\mathcal{N}} = K_{1_k}^{\mathcal{N}}(\omega_k) tanh(s_k^{\mathcal{N}})$, $v_k^{\mathcal{N}} = K_{2_k}^{\mathcal{N}}(\omega_k) tanh(s_k^{\mathcal{N}})$, $K_{1_k}^{\mathcal{N}} = K_{1_k}(\omega_k) + k_{r_k}^u(\omega_k)$, $K_{2_k}^{\mathcal{N}} = K_{2_k}(\omega_k) + k_{r_k}^v(\omega_k)$, with $K_{i_k}(0) = 0$, i=1,2, $k_r^u(0) = 0$, $k_r^v(0) = 0$, where $k_r^u = [k_{r_k}^u]_{k=1,..,N}$, $k_r^v = [k_{r_k}^v]_{k=1,..,N}$ should be estimated.

We proceed similar to (Forsman, 1998). We refer in the following to four regions, $\Omega_{\mathcal{A}}(\eta)$, $\Omega_{\mathcal{N}}(\eta)$ and $B_{\mathcal{A}}(\eta)$, $B_{\mathcal{N}}(\eta)$, defined as follows: $\Omega_{\mathcal{A}}(\eta) = \{\omega | Q_{\mathcal{A}}(\omega, \eta) \leq 0\}$, $\Omega_{\mathcal{N}}(\eta) = \{\omega | Q_{\mathcal{N}}(\omega, \eta) \leq 0\}$, $B_{\mathcal{A}}(\eta) = \{\omega | V_{\mathcal{A}}(\omega, \eta) \leq V_{\mathcal{N}}(\omega, \eta)\}$, $B_{\mathcal{N}}(\eta) = \{\omega | V_{\mathcal{N}}(\omega, \eta) \leq d\}$

$$
\begin{aligned}
Q_{\mathcal{A}} &= \frac{\partial V_{\mathcal{A}}}{\partial s_{\mathcal{A}}} \dot{s}^{\mathcal{A}} + K_1^T \frac{\partial K_1}{\partial \omega} \dot{\omega} + K_2^T \frac{\partial K_2}{\partial \omega} \dot{\omega} \\
Q_{\mathcal{N}} &= \frac{\partial V_{\mathcal{N}}}{\partial s_{\mathcal{N}}} \dot{s}^{\mathcal{N}} + \\
&+ \left((K_1^{\mathcal{N}})^T \frac{\partial K_1^{\mathcal{N}}}{\partial \omega} + (K_2^{\mathcal{N}})^T \frac{\partial K_2^{\mathcal{N}}}{\partial \omega}\right) \dot{\omega}|_{u_k^{\mathcal{N}}, v_k^{\mathcal{N}}}
\end{aligned}
$$
(23)

Stability implies: $B_{\mathcal{A}}(\eta) \subseteq \Omega_{\mathcal{A}}(\eta)$, $B_{\mathcal{N}}(\eta) \subseteq \Omega_{\mathcal{N}}(\eta)$, $B_{\mathcal{A}}(\eta) \subseteq B_{\mathcal{N}}(\eta)$, $\Omega_{\mathcal{A}}(\eta) \subseteq \Omega_{\mathcal{N}}(\eta)$. The design has as a primary objective to find polynomials relating c_{kj}, d and η by examining the following ideals for all switching systems that are derived from alliances $\mathcal{A}_i$:

$$\begin{aligned}
\vartheta_1 &= < V_{\mathcal{A}} - V_{\mathcal{N}}, Q_{\mathcal{A}}, g_1^{\mathcal{A}}, ..., g_n^{\mathcal{A}} > \\
\vartheta_2 &= < V_{\mathcal{N}} - d, Q_{\mathcal{N}}, g_1^{\mathcal{N}}, ..., g_n^{\mathcal{N}} > \\
\vartheta_3 &= < (\omega_1^2)^q - \omega_1^2, ..., (\omega_l^2)^q - \omega_l^2 >
\end{aligned} \quad (24)$$

where $g_i^{\mathcal{A}} = \frac{\partial V_{\mathcal{A}}}{\partial \omega_i} - \lambda \frac{\partial Q_{\mathcal{A}}}{\partial \omega_i}$, $g_i^{\mathcal{N}} = \frac{\partial V_{\mathcal{N}}}{\partial \omega_i} - \lambda \frac{\partial Q_{\mathcal{N}}}{\partial \omega_i}$ and $\omega^2 = [\begin{array}{ccc} \omega_1^2 & ... & \omega_l^2 \end{array}]$ is as in (7). The Grobner basis for the ideal $\tilde{\vartheta} = \vartheta_1 + \vartheta_2 + \vartheta_3$, with the algebraic restrictions related to the overlapping variables, is computed w.r.t. the following ranking:

$$\begin{aligned}
d \prec \{\eta\} &\prec \{c_{kj}\} \prec \omega \\
&\prec \{s_r\} \prec \{K_{i_k}\} \prec \{k_r\} \prec \{\lambda\}
\end{aligned} \quad (25)$$

and the restrictions $s_{r_k} \in R[\omega_k, \eta, c_k, d]$, $K_{1_k}, K_{2_k} \in R[\omega_k, \eta, c_k, d, s_{r_k}]$, $k_{r_k}^u, k_{r_k}^v \in R[\omega_k, \eta, c_k, d, s_{r_k}]$, The self-adaptation is performed by extending alliances to neighborhoods, using the variables k_r and s_r. This extension means to implement a stable dynamics in the state space $\omega^{\mathcal{N}}$ using the stable dynamics in the state space ω, $\omega \mapsto \omega^{\mathcal{N}} = [\begin{array}{cc} \omega & \hat{\omega} \end{array}]$. The Mind component has to find the components $\hat{c}_k = \phi_1(c_k, s_{r_k})$ and the control functions $\hat{K}_{i_k}(\omega_k^{\mathcal{N}}) = \phi_{2_i}(K_{i_k}(\omega_k), k_{r_k}, \omega_k^{\mathcal{N}})$, i=1,2. The sliding manifolds $s_k^{\mathcal{A}}$, with off-line computed c_{kj} are extended on-line to the sliding manifolds $s_k^{\mathcal{N}}$ where:

$$\begin{aligned}
s_k^{\mathcal{A}} &= \sum_{j=1}^{N_k^{\mathcal{A}}} c_{kj} S_{kj}(\omega) = 0 \\
s_k^{\mathcal{N}} &= \sum_{j=1}^{N_k} \hat{c}_{kj}(c_{kj}, s_{r_k}) \hat{S}_{kj}(\omega^{\mathcal{N}}) = 0
\end{aligned} \quad (26)$$

The control gain $(K_{1_k}(\omega_k), K_{2_k}(\omega_k))$ should be extended on-line to depend on $\omega_k^{\mathcal{N}}$, as follows:

$$\begin{aligned}
(K_{1_k}(\omega_k), K_{2_k}(\omega_k)) &\mapsto \\
\mapsto (\hat{K}_{1_k}(\omega_k^{\mathcal{N}}, k_{r_k}), &\hat{K}_{2_k}(\omega_k^{\mathcal{N}}, k_{r_k}))
\end{aligned} \quad (27)$$

We will approach this extension into a future paper.

9 CONCLUSIONS

The proposed framework is intended for developing control systems for distributed systems. The control law is designed in order to implement a coherent coordination for a SoS while the dynamics is constraint by sub-plans. The agents have the ability of self-adaptation because they are able to change the sliding manifolds that are parts of the decision making process.

REFERENCES

Antsaklis, P. I. and Passino, K. M. (1993). *An Introduction To Intelligent and Autonomous Control*. Kluver Academic Publisher.

Cistelecan, R. (2004a). Dynamic networks: Decentralized control and feasibility. In *Sci. Bull. of Politehnica Univ. of Timisoara, Romania, Vol.49(63),121-127*.

Cistelecan, R. (2004b). Flat hierarchical society-oriented systems and agents coordination. In *Sci. Bull. of Politehnica Univ. of Timisoara, Romania, Vol.49(63),115-121*.

Dogruel, M., Ozguner, U., and Drakunov, S. (1996). Sliding-mode control in discrete-state and hybrid systems. In *IEEE Trans. on AC, vol. 41, no. 3 , 414-419*.

Fisher, M. and Wooldridge, M. (2003). A logical approach to simulating societies,1-14. In *G.N. Gilbert and R. Conte (eds), Artificial Societies: the computer simulation of social life, UCL Press, London*.

Forsman, K. (1994). Hybrid control systems and comprehensive grobner bases. In *Tech. Report, Automatic Control, Linkoping*.

Forsman, K. (1998). Constructive algebraic geometry in nonlinear control. In *29th IEEE CDC*.

Fortell, H. (1995). Algebraic approaches to normal forms and zero dynamics. In *Ph. D. Thesis, Linkoping Univ.*

Gunnarsson, J. and Plantin, J. (1998). Synthesis of a discrete system using algebraic methods. In *WODES96*.

Jirstrand, M. and Glad, T. (1997). Computationl questions of equilibrium calculation with application to nonlinear aircraft dynamics. In *Tech. Rep., Linkoping Univ.*

Klavins, E. and Koditschek (2000). A formalism for the composition of concurrent robot behaviors. In *Proc. of ICRA,3395-3402*.

Marchand, H. and LeBorgne, M. (1998). Partial order control of discrete event systems modeled as polinomial dynamical systems. In *IEEE Int. Conf. on Systems, Man and Cybernetics, 770-775*.

Ortiz, C., Hsu, E., and DesJardins, M. (2001). Incremental negotiation and coalition formation for resource-bounded agents. In *Proc of AAAI Fall Symposium on Negotiation Methods for Autonomous Coop. Syst.*

Sen, S. and Dutta, P. (2002). The evolution and stability of cooperative traits. In *Proc. of the First International Joint Conference on autonomous agents and Multi-agents systems, Italy,1114-1120*.

Sims, M., Goldman, C., and Lesser, V. (2003). Self-organization through bottom-up coalition formation, 867-874. In *AAMAS'03, Australia*.

Yan, P., Gao, Y., and Ozbay, H. (2003). A variable structure control approach to active queue management for tcp with ecn. In *IEEE Trans. on Control Systems Technol.*