Safety and Security in Networked Robotic Systems Via Logical Consensus

Simone Martini[1], Adriano Fagiolini[1], Antonio Bicchi[1] and Gianluca Dini[2]

[1] Interdepartmental Research Center "E. Piaggio", Faculty of Engineering
University of Pisa, Italy

[2] Dipartimento dell'Informazione, Faculty of Engineering, University of Pisa, Italy

Abstract. The problem of allowing a set of autonomous mobile robots to plan their motion by reaching consensus on logical observations of the environment is studied in this paper. The particularity of the work is that the information on which the consensus is sought is not represented by real numbers, but rather by logical values such as the presence of an obstacle, of another robot, or of a person. Previous work by the authors considered the problem of allowing a set of agents to consent on the value of a logical vector function by communicating over a network. In this paper, we present application of this result to the motion planning decision problem and show its effectiveness through simulation.

1 Introduction

In the last decades, robotics has undergone a gradual yet constant migration of research interests from monolithic systems with a unique robot to distributed multi–agents composed of several semi–autonomous robots. Various motivations give reason for this trend, among which is the possibility to achieve this desirable properties, such as scalability, reconfigurability, robustness, etc. Recent years have indeed witnessed important developments in the definition of decentralized and cooperative control strategies for applications, such as intelligent transportation, surveillance, flocking, formation control, sensor coverage, patrolling, etc., all involving teams of robotic agents (see e.g. [1, 2]). Most of these solutions require that agents consent on the value of a common quantity of interest. This is achieved by means of consensus algorithms that are dynamical systems, where every agent has a state that is updated through local measurement and data received from its neighbors in a communication network [3–5]. A typical form of consensus is described by the continuous–time linear system

$$\dot{x}(t) = A\,x(t) + B\,u(t)\,, \tag{1}$$

where $A \in \mathbb{R}^{n\times n}$ is a strongly connected doubly–stochastic matrix, $B \in \mathbb{R}^{n\times m}$ is the input matrix, and $u \in \mathbb{R}^m$ is a control law. The flourishing literature on this topic have studied continuous– and discrete–time, synchronous and asynchronous, and quantized versions of such systems and has provided useful results on properties such as characterization of equilibria, and convergence rate [6, 7].

Proceedings of ICINCO 2009
6th International Conference on Informatics in Control, Automation and Robotics

Furthermore, the actual achievement of the system goal is theoretically guaranteed only under the hypothesis that all agents harmoniously act and cooperate, whereas if some of them do not follow specification the whole system is at risk [8]. This motivates the emerging interest toward techniques that make more robust existing multi–agent systems by detecting the presence of intruders in various different settings (see [9]). The EU, in the current Seventh Framework Programme, has suggested Security and Safety in Automation and Robotics as one of the main aspects over which research communities should place their efforts. Research on this field focused on detection of faults and anomalies in networked control systems, but the main theory and tools developed in the project are strongly based on the existence of one or more centralized supervisors [10]. The challenge in these systems is to find strategies to detect and isolate possible intruders, without the use of any form of centralization. This requires understanding what level of intelligence must be embedded in the automation component to provide satisfactory guarantees of performance, while remaining economically viable.

Another important fact that must be taken under consideration is that control and automation systems are implemented on embedded devices, having resource and real–time constraints that are much more severe than customary desktop and enterprise computing [11]. Therefore, these constraints must be taken into account when designing and building security solutions for any of such networked embedded system. In the Sixth Framework Programme of the EU, the project RUNES pioneered some solutions guaranteeing security for resource constrained networked embedded systems [12, 13].

In this context, we focus on the problem of coordinated motion for a set of mobile semi–autonomous robots. The problem is studied also within the current EU project CHAT [14] and the Network of Excellence CONET [15]. We propose a solution that requires limited communication and computation complexities. The solution is based on so–called *logical consensus* systems, that are algorithms allowing a set of agents to consent on a number of decisions depending on logical inputs of the environment. The proposed solution allows the agents to plan their motion by reaching consensus on logical values based on local observation of the environment. This endows every agent with the capability to react to unexpected changes in the environment, such as the presence of an obstacle or of an intruder. Indeed, during the execution of their plans, features of the environment that were unknown at planning time, or that unexpectedly change, can trigger changes in what the agents should do. Under suitable joint conditions on the visibility of agents and their communication capability, we provide an algorithm generating logical linear consensus systems that are globally stable that allow each agent to update its path according to the actual configuration of the environment.

2 Problem Statement

We consider application scenarios requiring computation of a set of p *decisions*, $y_1, \dots, y_p$, that depend on m logical *events*, $u_1, \dots, u_m$. Such events may represent e.g. the presence of an intruder or of a fire within an indoor environment. More precisely, for any given combination of input events, we consider a *decision task* that requires computation of the following system of logical functions:

$$\begin{cases} y_1 = f_1(u_1, \ldots, u_m)\,, \\ \cdots \\ y_p = f_p(u_1, \ldots, u_m)\,, \end{cases} \tag{2}$$

where each $f_i : \mathbb{B}^m \to \mathbb{B}$ consists of a logical condition on the inputs. Let us denote with $u = (u_1, \ldots, u_m)^T \in \mathbb{B}^m$ the input event vector, and with $y = (y_1, \ldots, y_p)^T \in \mathbb{B}^p$ the output decision vector. Then, we will write $y = f(u)$ as a compact form of Eq. 2, where $f = (f_1, \ldots, f_p)^T$, with $f : \mathbb{B}^m \to \mathbb{B}^p$, is a logical vector function. It is worth noting that computation of f is *centralized* in the sense that it may require knowledge of the entire input vector u to determine the output vector y.

Our approach to solve the decision task consists of employing a collection of n agents, $\mathcal{A}_1, \ldots, \mathcal{A}_n$, that are supposed to cooperate and possibly exchange locally available information. We assume that each agent is described by a triple $\mathcal{A}_i = (\mathcal{S}_i, \mathcal{P}_i, \mathcal{C}_i)$, where $\mathcal{S}_i$ is a collection of sensors, $\mathcal{P}_i$ is a processor that is able to perform elementary logical operations such as $\{\texttt{and}, \texttt{or}, \texttt{not}\}$, and $\mathcal{C}_i$ is a collection of communication devices allowing transmission of only sequences of binary digits, 0 and 1, namely strings of bits. Although we assume that every agent has the same processing capability, i.e. $\mathcal{P}_i = \mathcal{P}$ for all i, we consider situations where agents may be *heterogeneous* in terms of sensors and communication devices. Due to this diversity as well as the fact that agents are placed at different locations, a generic agent i may or may not be able to measure a given input event u_j, for $j \in 1, \ldots, m$. Therefore, we can conveniently introduce a *visibility matrix* $V \in \mathbb{B}^{n \times m}$ such that we have $V(i, j) = 1$ if, and only if, agent $\mathcal{A}_i$ is able to measure input event u_j, or, in other words, if the i–th agent is directly *reachable* from the j–th input. Moreover, for similar reasons of diversity and for reducing battery consumption, each agent is able to communicate only with a subset of other agents. This fact is captured by introducing a *communication matrix* $C \in \mathbb{B}^{n \times n}$, where $C(i, k) = 1$ if, and only if, agent $\mathcal{A}_i$ is able to receive a data from agent $\mathcal{A}_k$. Hence, agents specified by row $C(i, :)$ will be referred to as C–neighbors of the i–th agent. The introduction of visibility relations between inputs and agents immediately implies that, at any instant t, only a subset of agents is able to measure the state of each input u_j, for all j. Therefore, to effectively accomplish the given decision task, we need that such an information *flows* from one agent to another, consistently with available communication paths. We require all agents reach an agreement on the centralized decision $y = f(u)$, so that any agent can be *polled* and provide consistent complete information. In this perspective, we pose the problem of reaching a *consensus on logical values*.

In this view, we can imagine that each agent $\mathcal{A}_i$ has a local *state* vector, $X_i = (X_{i,1}, \ldots, X_{i,q}) \in \mathbb{B}^q$, that is a *string of bits*.
Then, let us denote with $X(t) = (X_1^T(t), \ldots, X_n^T(t))^T \in \mathbb{B}^{n \times q}$ a matrix representing the network state at a discrete time t. Hence, we assume that each agent $\mathcal{A}_i$ is a *dynamic node* that updates its local state X_i through a *distributed* logical update function F that depends on its state, on the state of its C–neighbors, and on the reachable inputs, i.e. $X_i(t+1) = F_i(X(t), u(t))$. Moreover, we assume that each agent $\mathcal{A}_i$ is able to produce a logical output decision vector $Y_i = (y_{i,1}, \ldots, y_{i,p}) \in \mathbb{B}^p$ through a suitable distributed logical output function G depending on the local state X_i and on the reachable inputs u, i.e. $Y_i(t) = G_i(X_i(t), u(t))$. Let us denote with $Y(t) = (Y_1^T(t), \ldots, Y_p^T(t))^T \in \mathbb{B}^{p \times q}$

a matrix representing the network output at a discrete time t. Therefore, the dynamic evolution of the network can be modeled by the following *distributed finite–state iterative system*:

$$\begin{cases} X(t+1) = F(X(t), u(t))\,, \\ \quad\;\; Y(t) = G(X(t), u(t))\,, \end{cases} \tag{3}$$

where we have $F = (F_1^T, \ldots, F_n^T)^T$, with $F_i : \mathbb{B}^q \times \mathbb{B}^m \to \mathbb{B}^q$, and $G = (G_1^T, \ldots, G_n^T)^T$, with $G_i : \mathbb{B}^q \times \mathbb{B}^m \to \mathbb{B}^p$.

In this perspective, we are interested in solving the following design problem:

Problem 1 (Globally Stable Synthesis). Given a decision system of the form of Eq. 2, a visibility matrix V, and a communication matrix C, design a logical consensus system of the form of Eq. 3, that is compliant with C and V, and such that, for all initial network state $X(0)$, and all inputs u, there exists a finite time $\bar{N}$ such that the system reaches a consensus on the centralized decision $y^* = f(u)$, i.e. $Y(t) = \mathbf{1}_n\,(y^*)^T$, for all $t \geq \bar{N}$.

3 Distributed Map Synthesis for Logical Consensus

In this section a solution for Problem 1 is presented consisting of an algorithm that generates an optimal distributed logical linear consensus system. More precisely, the algorithm produces a (C, V)–compliant linear iteration map F minimizing the number of messages to be exchanged, and the time needed to reach a consensus (a.k.a. *rounds*).

To achieve this we first need to understand how the agent network can reach a consensus on the value of the j–th subterm l_j in the decision system of Eq. 2. Without loss of generality, let us pose $l_j = u_j$ and consider the j–th column V_j of the visibility matrix V that also describes the visibility of l_j. Then, we need a procedure for finding to which agents the value of input u_j can be propagated. First note that vector V_j contains 1 in all entries corresponding to agents that are able to “see” u_j, or, in other words, it specifies which agents are directly *reachable* from u_j. Then, it is useful to consider vectors $C^k V_j$, for $k = 0, 1, \ldots$, each containing 1 in all entries corresponding to agents that are reachable from input u_j after *exactly* k steps. The i–th element of $C^k V_j$ is 1 if, and only if, there exists a *path* of length k from any agent directly reached by u_j to agent $\mathcal{A}_i$. Recall that, by definition of graph diameter, all agents that are reachable from an initial set of agents are indeed reached in at most $\mathtt{diam}(G)$ steps, with $\mathtt{diam}(G) \leq n-1$. Let us denote with κ the *visibility diameter* of the pair (C, V_j) being the number of steps after which the sequence $\{C^k V_j\}$ does not reach new agents. Thus, given a pair (C, V_j), we can conveniently introduce the following *reachability matrix* R_j, assigned with input u_j:

$$R_j = \begin{pmatrix} V_j & CV_j & C^2 V_j & \cdots & C^{n-1} V_j \end{pmatrix}\,, \tag{4}$$

whose columns *span* a subgraph $G_{\mathcal{R}}(N_{\mathcal{R}}, E_{\mathcal{R}})$ of G, where $N_{\mathcal{R}}$ is a node set of all agents that are *eventually* reachable from input u_j, and $E_{\mathcal{R}}$ is an unspecified edge set, that will be considered during the design phase. Computing the span of R_j is very simple and efficient, and indeed all reachable agents, that are nodes of $N_{\mathcal{R}}$, are specified by non–null elements of the Boolean vector $I_j = \sum_{k=0}^{n-1} C^k V_j = \sum_{k=0}^{n-1} R_j(:, i)$, that is the logical sum of all columns in R_j and that contains 1 for all agents for which there exists at least one path originating from an agent that is able to measure u_j. Then, we can partition the agent network into $N_{\mathcal{R}} = \{i \mid I_j(i) = 1\}$, and $N_{\bar{\mathcal{R}}} = N \setminus N_{\mathcal{R}}$, where $N = \{1, \ldots, n\}$. In this perspective we can give the following:

Definition 1. *A pair (C, V_j) is* (completely) reachable *if, and only if, the corresponding reachability matrix $R_j(C, V_j)$ spans the entire graph, i.e. $N_{\mathcal{R}} = N$.*

The design phase can obviously concern only the reachable subgraph $G_{\mathcal{R}}(N_{\mathcal{R}}, E_{\mathcal{R}})$ of G, and in particular will determine the edge set $E_{\mathcal{R}}$. Moreover, observe that a non–empty unreachable subgraph $G_{\bar{\mathcal{R}}}$ in a consensus context is a symptom of the fact that the design problem is not well–posed, and it would require changing sensors' visibility and locations in order to have a reachable (C, V_j) pair.

Let us suppose that only agent $\mathcal{A}_1$ is able to measure u_j. Then, a straightforward and yet optimal *strategy to allow the information on u_j flowing* through the network is obtained if agent $\mathcal{A}_1$ communicates its measurement to all its C–neighbors, which in turn will communicate it to all their C–neighbors without overlapping, and so on. In this way, we have that every agent $\mathcal{A}_i$ receives u_j from exactly one minimum–length path originating from agent $\mathcal{A}_1$. The vector sequence $\{C^k V_j\}$ can be exploited to this aim. Indeed, it trivially holds that $C^k V_j = C(C^{k-1} V_j)$, meaning that agents reached after k steps have received the input value from agents that were reached after exactly $k-1$ steps. Then, any consecutive sequence of agents that is extracted from non–null elements of vectors in $\{C^k V_j\}$ are (C, V_j)–compliant by construction. A consensus strategy would minimize the number of rounds if, and only if, at the k–th step, all agents specified by non–null elements of vector $C^k V_j$ receives the value of u_j from the agents specified by non–null elements of vector $C^{k-1} V_j$. Nevertheless, to minimize also the number of messages, only agents specified by non–null elements of vector $C^k V_j$ and that have not been reached yet must receive u_j. If vector $I_j = \sum_{i=0}^{i=k} C^i V_j$ is iteratively updated during the design phase, then the set of all agents that must receive a message on u_j are specified by non–null elements of vector $C^k V_j \wedge \neg I_j$. By doing this, an optimal pair (C^*, V_j^*) allowing a consensus to be established over the reachable subgraph is obtained.

Observe that is $C^* = S\, C \leq C$, where S is a suitable selection matrix.

This procedure actually gives us only a suggestion on how to construct consensus system that solves Problem 1. Indeed, we can prove in following Theorem 1 that a simple logical linear consensus algorithm of the form

$$x(t+1) = F_j\, x(t) + B_j\, u_j(t)\,, \tag{5}$$

where $F_j = C^*$, $B_j = V_j^*$, and $x \in \mathbb{B}^n$, allows a consensus to be reached through the entire reachable subgraph. The state x must be interpreted as the network *distributed estimation* of the value of the subterm l_j or u_j. It is indeed a vector and not a matrix, since we are concerned here only with the j–th input.

In all cases where a unique generic agent $\mathcal{A}_i$ is directly reachable from input u_j, an optimal communication matrix C^* for a linear consensus of the form of Eq. 5 can be iteratively found as the incidence matrix of a *input–propagating spanning tree* having $\mathcal{A}_i$ as the root. Then, an optimal pair (C^*, V_j^*) can be written as $C^* = P^T\,(S\, C)\, P$, and $V_j^* = P^T\, V_j$, where S is a selection matrix, and P is a permutation matrix. Furthermore, C^* has the following lower–block triangular form:

$$C^* = \left(\begin{array}{cccc|c} 0 & 0 & \cdots & 0 & 0 \\ \tilde{C}_{i,1} & 0 & \ldots & 0 & 0 \\ \vdots & & & \vdots & \vdots \\ 0 & \cdots & \tilde{C}_{i,\kappa_i} & 0 & 0 \\ \hline 0 & \cdots & 0 & 0 & 0 \end{array} \right) , \tag{6}$$

and $V_j^* = P^T V_j = (1, 0, \ldots, 0)^T$. It is worth noting that the optimal pair (C^*, V_j^*) preserves the reachability property of the original pair (C, V_j). This can be shown by direct computation of the reachability matrix R_j^*, but it is omitted for the sake of space.

We are now ready to consider the more general case with ν, $1 \leq \nu \leq n$ agents that are reachable from input u_j, and let us denote with $A = \{i_1, \ldots, i_\nu\}$ the index set of such agents. Then, the optimal strategy for propagating input u_j consists of having each of the other agents receive the input measurement through a path originating from the nearest reachable agent in A. This naturally induces a network partition into ν disjoint subgraphs or spanning trees, each directly reached by the input through a different agent. Let us extract ν independent vectors $V_j(i_1), \ldots, V_j(i_\nu)$ from vector V_j having a 1 in position i_h. Then, the sequences $\{C^k V_j(i_h)\}$ are to be considered to compute the optimal partition. Let us denote with κ_i, for all $i \in A$ the number k of steps for the sequence $\{C^k V_j(i)\}$ to become stationary. Therefore, we have that the visibility diameter of the pair (C, V_j) is $\texttt{vis-diam}(C, V_j) = \max_i\{\kappa_i\}$. Without loss of generality, we can image that $\kappa_1 \geq \kappa_2 \geq \cdots \geq \kappa_\nu$. Therefore, for the generic case, there exist a permutation matrix P and a selection matrix S such that an optimal pair (C^*, V_j^*) can be obtained as $C^* = P^T (S\, C)\, P$, $V_j^* = P^T V_j$, where

$$C^* = \texttt{diag}(C_1, \ldots, C_\nu)\,, V_j^* = (V_{j,1}^T, \ldots, V_{j,\nu}^T)^T\,, \tag{7}$$

and where each C_i and $V_{j,i}$ have the form of the Eq. 6. Finally, the actual optimal linear consensus algorithm is obtained choosing $F_j = P\, C^*$, and $B_j = P\, V_j^*$.

Algorithm 1 allows computation of an optimal pair (C^*, V_j^*) as in Eq. 7. Its asymptotic *computational complexity* is in the very worst case $O(n^2)$, where n is the number of agents, and its *space complexity* in terms of memory required for its execution is $\Omega(n)$. However, its implementation can be very efficient since it is based on Boolean operations on bit strings. Finally, *communication complexity* of a run of the consensus protocol in terms of the number of rounds is $\Theta(\texttt{vis-diam}(C, V_j))$.

To conclude, we need to prove that a so–built logical consensus system does indeed solve Problem 1. Hence, for the general case with $\nu \geq 1$ agents that are reachable from input u_j, we can the state the following result (the proof is omitted for space limitation):

Theorem 1 (Global Stability of Linear Consensus). *A logical linear consensus system of the form $x(t+1) = C^*\, x(t) + V_j^*\, u_j(t)$, where C^* and V_j^* are obtained as in Eq. 7 from a reachable pair (C, V_j), converges to a unique network agreement given by $\mathbf{1}_n\, u_j$ in at most $\texttt{vis-diam}(C, V_j)$ rounds.*

Algorithm 1 Optimal Linear Synthesis by Input–Propagation.

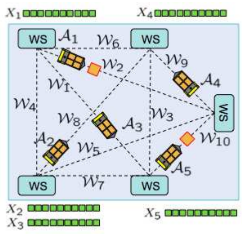
Inputs: C, V_j
Outputs: Minimal pair (C^*, V_j^*), permutation P.

Set $A \leftarrow \{i \,|V_j(i) = 1\}$ ◁ nodes directly reachable from u_j
Set $I(i) \leftarrow 1$ for all $i \in A$ ◁ nodes reached from $i \in A$
Set $N \leftarrow \{1, \ldots, n\} \setminus I$ ◁ nodes not yet reached
repeat
  for all nodes $i \in A$ **do**
    Set $Adj(i) \leftarrow C^k V_j(i) \,\wedge\, \neg I(i) \,\wedge N$ ◁ new nodes
    Set $I(i) \leftarrow I(i) \vee Adj(i)$
    Set $N \leftarrow N \wedge \neg Adj(i)$
    Compute $\mathcal{I} \leftarrow \{h : Adj(i)(h) = 1\}$ ◁ index list
    for all new nodes $h \in \mathcal{I}$ **do**
      Set $\tilde{C}(h, :) \leftarrow C(h, :) \wedge Adj(i)^T$ ◁ every new node must communicate with one reach at $k - 1$
    end for
  end for
until $\exists\, i \in A \,|\, Adj(i) \neq 0$
Compute $\kappa_i \leftarrow \text{card}(I(i))$ for all $i \in A$
Find $P \mid C^* \leftarrow P^T\, \tilde{C}\, P$ has $\kappa_1 \geq \cdots \geq \kappa_\nu$ ◁ re-order
Set $V_j^* \leftarrow P^T\, V_j$

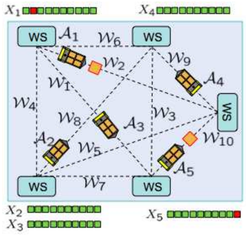
4 Application to Intrusion Detection

Consider an indoor environment with n agents $\mathcal{A}_1, \cdots, \mathcal{A}_n$ whose task is to move packages between workspaces (*WS*). Suppose that agents have the capability to compute the path associated with a task and to plan the sequence of tasks by finding an agreement with other agents in order to avoid collisions and to avoid the use of the same segment ($\mathcal{W}$) in the same moment. We assume that each agent have also the capability to detect and locate possible intruders or obstacles, such as lost packages or failed agents, in $\mathcal{W}$. The presence or the absence of an intruder in segment $\mathcal{W}_j$ can be seen as an input u_j to the system of $p = m$ logical decision $y_i(t) = u_i(t)\,,\ i = 1, \ldots, m$, that each agent is required to estimate. However, agents are able to detect the presence of intruders only within their visibility areas, which is described by a visibility matrix $V \in \mathbb{B}^{n\times m}$, with $V_{i,j} = 1$ if, and only if, an intruder in region $\mathcal{W}_j$ can be seen by agent $\mathcal{A}_i$. Moreover, let $X \in \mathbb{B}^{n\times m}$ denote the alarm state of the system: $X_{i,j} = 1$ if agent $\mathcal{A}_i$ reports an alarm about the presence of an intruder in segment $\mathcal{W}_j$. The alarm can be set because an intruder is actually detected by the agent itself, or because of communications with neighboring observers. Indeed, agents have communication devices that allows them to share alarm states with all other agents that are nearby. In this context, we aim at designing a distributed update rule of the form $X(t+1) = F(X(t), u(t))$, such that agents can achieve the same state value ($X_{i,j} = X_{k,j}\ \forall i, k$ and $\forall j$). In other terms, *at consensus*, each column of X should have either all zeros or all ones, depending on the corresponding column of $\mathbf{1}_n f(u) = \mathbf{1}_n u$.

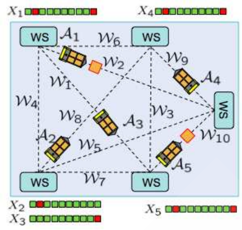
Consider first applying Algorithm 1 that produces a linear logical consensus of the form $X(t+1) = F\,X(t) + B\,u(t)$, where each row basically expresses the rule that



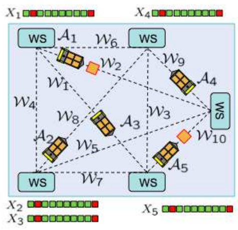
(a) $t = 0$



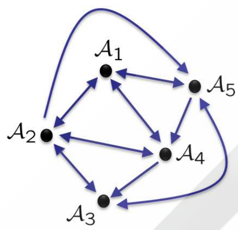
(b) $t = 1$



(c) $t = 2$



(d) $t = 3$



(e)

Fig. 1. (a)–(d) Run of the linear consensus system with 2 intruders (brown squares) in segment $\mathcal{W}_2$ and $\mathcal{W}_{10}$, respectively. The figure sequence shows that a correct agreement is reached (components of the state X_i of every agents are `green` or 0, when no intruder is detected in the corresponding segment, `red` or 1 otherwise). (e) Considered communication graph C.

an observer alarm is set at time $t + 1$ if it sees an intruder (through u), or if one of its C–neighbors was set at time t. The visibility diameter of this pair (C, V) is 3, which will correspond to the maximum number of steps before consensus is reached. Fig. 1 shows snapshots from a typical run of this linear consensus algorithm where every agents converge to consensus after 3 steps. It is clear that using this method it is not necessary that the system stops in the case that an intruder is detected in the area. By sharing local information with other agents, each agent is able to execute its task by excluding unavailable segments and by finding alternative paths to reach the goal.

5 Conclusions

In this work we considered the problem of the safety and security in the coordinated motion of mobile robotics systems. The problem is studied through a novel consensus mechanism where agents of a network are able to share logical values. We propose an algorithm producing optimal logical consensus systems. By reaching consensus on logical values based on local observation of the environment agents are able to update its path according to the actual configuration of the environment and to solve the motion planning decision problem.

References

1. Olfati-Saber, R.: Flocking for multi-agent dynamic systems: algorithms and theory. IEEE Trans. on Automatic Control 51 (2006) 401–420

2. Figueiredo, L., Jesus, I., Machado, J., Ferreira, J., Martins de Carvalho, J.: Towards the development of intelligent transportation systems. Proc. IEEE Intelligent Transportation Systems (2001) 1206–1211
3. Olfati-Saber, R., Fax, J.A., Murray, R.N.: Consensus and Cooperation in Networked Multi–Agent Systems. Proc. of the IEEE (2007)
4. Ren, W., Beard, R., Atkins, E.: Information consensus in multivehicle cooperative control. IEEE Cont. Syst. Mag. 27 (2007) 71–82
5. Blondel, V., Hendrickx, J., Olshevsky, A., Tsitsiklis, J.: Convergence in Multiagent Coordination, Consensus, and Flocking. Proc. IEEE International Conference on Decision and Control (2005) 2996–3000
6. Fang, L., Antsaklis, P., Tzimas, A.: Asynchronous Consensus Protocols: Preliminary Results, Simulations and Open Questions. IEEE Int. Conf. on Decision and Control and Eur. Control Conference (2005) 2194–2199
7. Bertsekas, D., Tsitsiklis, J.: Parallel and Distributed Computation: Numerical Methods. (2003)
8. Baras, J.: Security and trust for wireless autonomic networks: Systems and control methods. European Journal of Control 13 (2007) 105–133
9. Franceschelli, M., Egerstedt, M., Giua, A.: Motion Probes for Fault Detection and Recovery in Networked Control Systems. American Control Conference, Seattle, WA, June (2008)
10. Necst - networked control systems tolerant to faults. (Sixth Framework Programme, Priority 2, Information Society Technologies, STREP, IST-004303)
11. Dzung, D., Naedele, M., von Hoff, T., Crevatin, M.: Security for industrial communication systems. Proceedings of the IEEE 93 (2005) 1152–1177
12. Bicchi, A., Danesi, A., Dini, G., La Porta, S., Pallottino, L., Savino, I.M., Schiavi, R.: Heterogeneous wireless multirobot system. Robotics and Automation Magazine, IEEE 15 (2008) 62–70
13. Dini, G., Savino, I.: An efficient key revocation protocol for wireless sensor networks. In: Proc. IEEE International Symposium on World of Wireless, Mobile and Multimedia Networks, IEEE Computer Society Washington, DC, U.S.A (2006) 450–452
14. Chat - control of heterogeneous automation systems: Technologies for scalability, reconfigurability and security. (Seventh Framework Programme, IST-2008-224428)
15. Conet - the cooperating objects network of excellence. (Seventh Frame Programme, 2007-2-224053)