Set-Consensus using Set-Valued Observers

Daniel Silvestre, Paulo Rosa, João P. Hespanha, Carlos Silvestre

Abstract—This paper addresses the problem of reaching consensus when the position measurements are corrupted by noise and taken at different time instants. Two different scenarios are considered, namely: when the sensor function is performed on a group of nodes and sent to those nodes at the same time; and a second case where the transmission is unicast but neighboring nodes receive the message due to the shared nature of the wireless medium. A solution involving Set-Valued Observers (SVOs) is proposed to maintain and update the set-valued estimates for the positions and how to drive the agents. The algorithm is proved to enforce convergence of the nodes to clusters up to a factor dependent on the uncertainty. The performance of the proposed algorithm is assessed through simulations, illustrating, in particular how the choice for selecting how to divide the area can reduce or increase the number of clusters.

I. INTRODUCTION

The consensus problem is a distributed control task in which a set of agents have the objective of agreeing on a function of their initial state values by exchanging messages dictated by a given communication topology. Several multi-disciplinary applications of consensus algorithms have been reported in the literature. These include distributed optimization [1], [2]; motion coordination tasks, such as flocking, leader following [3], and rendezvous problems [4]; and resource allocation in computer networks [5].

Many contributions are available in the literature to solve consensus problems using linear distributed algorithms, in which each agent computes a weighted average between its state value and the state values of the agents to which it can communicate (see, e.g. [6], [7]). Many variations of this problem have been addressed in the literature considering, e.g., stochastic packet drops and link failures [8], [9], quantized data transmissions [10], and time-varying communication connectivity [6], event-triggered communications [11], and self-triggered communications [12].

Even though the problem of consensus is largely studied, most of the contributions assume the states of nodes to be exact values. In this article, we address the problem when two main issues are present: the measurements are performed with errors, which motivates the calculation of a set of possible “true state” realizations; and the measurements are performed at different time instants, which caters for the need to propagate the state using the dynamics of the system. In [13], the authors address the former issue, but specify a particular shape for the sets and assume all-to-all communication. By resorting to a different technique, we aim to relax these assumptions at the expenses of a higher computational cost.

We use the Set-Valued Observers (SVOs) framework, whose concept was introduced in [14] and [15] (further information can be found in [16] and [17] and the references therein) as a way to represent and propagate the set-valued state estimates. The approach allows us to easily consider any kind of linear dynamics for the agents, and also to incorporate disturbances and model uncertainties. Furthermore, it will allow us to use the predicted actions of each agent to better choose the control law to apply at each node based only on local information.

The main contributions of this paper are as follows:

- the use of SVOs to update the set representing the uncertainty about the position of the nodes;
- it is proposed that each node can estimate the future position of their neighbors from the information received in the shared medium;
- the positions of the nodes are shown to converge to the vicinity of the remaining nodes, with this vicinity being dependent on a measure of the uncertainty.

The remainder of this paper is organized as follows. In Section II, we describe the problem of set-consensus and its motivation, as well as the main issues involved in its formulation. The first scenario where the antenna transmits the whole information regarding a partition to all the nodes is presented in Section III. A different case is shown in Section IV, where the antenna transmits to only one node and the remaining receive it due to the nature of the wireless medium, and how the control input of each neighbors can be seen as a disturbance. The main properties are given in Section V and illustrated in simulations through Section VI. Concluding remarks and directions of future work are provided in Section VII.

Notation : The transpose of a matrix $A$ is denoted by $A^\top$. For vectors $a_i$, $(a_1, \ldots, a_n) := [a_1^\top \ldots a_n^\top]^\top$. We let $1_n := [1 \ldots 1]^\top$ and $0_n := [0 \ldots 0]^\top$ denote $n$-dimensional vector of ones and zeros, respectively, and $I_n$ denotes the identity matrix of dimension $n$. Dimensions are omitted when clear from context. The vector $e_i$ denotes the canonical vector whose components are equal to zero, except for the $i$th component. The symbol $\otimes$ denotes the kronecker product.

D. Silvestre, C. Silvestre are with the Dep. of Electrical and Computer Engineering, Instituto Superior Técnico, ISR, 1046-001 Lisboa, Portugal. {dsilvestre,cjs}@isr.ist.utl.pt. This work was supported by the FCT project PEst-OE/EEI/LA0009/2011. The work from Daniel Silvestre is partially funded with grant SFRH/BD/71206/2010, from Fundação para a Ciência e a Tecnologia.

P. Rosa is with Deimos Engenharia, Lisbon, Portugal. paulo.rosa@deimos.com.pt.

João P. Hespanha is with the Dept. of Electrical and Computer Eng., University of California, Santa Barbara, CA 93106-9560, USA. J. Hespanha was supported by the U.S. Army Research Laboratory and the U.S. Army Research Office under grants No. W911NF-09-1-0553 and W911NF-09-D-0001. hespanha@ece.ucsb.edu
The notation \( ||.|| \) refers to \( ||v|| := \sup_i |v_i| \) for a vector, and \( ||A|| := \sigma_{\text{max}}(A) \). The \( i \)th coordinate of a vector \( v \) is denoted by \( v_i \).

II. PROBLEM STATEMENT

We look at the problem of having a group of \( n \) robots or agents that are trying to reach consensus over their positions. Due to desired costs savings or environment constraints, the robots are only equipped with receivers and have no sensing and self-localization capabilities. A tower takes measurements of the position and velocity of each robot by using, for example, a vision-based system, and uses directional antennae to forward that information to the nodes. To avoid the computational cost of maintaining sets for all the positions and velocities of the nodes in the network and have a solution suitable for large scale networks, we look to solve the problem in a distributed fashion. An illustration of the problem is depicted in Fig. 1. It is remarked that, although we will analyze this particular setup throughout the paper, the problem formulation can be extended to a myriad of other examples.

In the aforementioned setup, each node \( i \) will receive a set \( X_j \) for each of their neighbors \( j \) corresponding to the position and velocity with the corresponding measurement errors and possible disturbances. In our context, neighbors refer to nodes that are sufficiently close so as to belong to the same strip of field to which the central tower communicates. The tower defines \( m \) partitions of the terrain in such a way to cover the whole space where nodes can be. However, sensing and communication is not performed all at the same time and can be described by the set of points, \( x \), satisfying

\[
\begin{bmatrix}
M(k)A_{\Delta}^{-1} - M(k)A_{\Delta}^{-1}E(k) \\
C(k + 1) & 0 \\
0 & 0 \\
0 & I \\
0 & -I
\end{bmatrix}
\begin{bmatrix}
x \\
d
\end{bmatrix}
\leq
\begin{bmatrix}
m(k) + \hat{u}(k) \\
y(k + 1) \\
y(k + 1)
\end{bmatrix}
\]

where \( \hat{u}(k) := M(k)A_{\Delta}^{-1}B(k)u(k) \) and \( M(k) \) and \( m(k) \) in our problem defined the received measurement set and

\[
A_{\Delta} = A_{0} + \sum_{\ell=1}^{n_{\Delta}} \Delta_{\ell} A_{\ell}
\]

and \( \Delta_{\ell} \) is the realization of each uncertainty in the vector \( \Delta_{\ell} \).

This procedure assumes an invertible transmission matrix.
When this is not the case, we can adopt the strategy in [20] and solve the inequality

\[
\begin{bmatrix}
1 & -\Delta_x & -E(k) \\
-1 & \Delta_x & E(k) \\
0 & 0 & 1 \\
0 & 0 & -1 \\
C(k+1) & 0 & 0 \\
-C(k+1) & 0 & 0 \\
0 & M(k) & 0
\end{bmatrix}
\begin{bmatrix}
x^-
\end{bmatrix}
\leq
\begin{bmatrix}
0 \\
y(k+1) \\
-y(k+1) \\
m(k)
\end{bmatrix}
\]  

(3)

by applying the Fourier-Motzkin elimination method [21] to remove the dependence on \(x^-\) and obtain the set described by

\[
M(k+1)x \leq m(k+1)
\]

In order to provide a solution for the set-valued state estimate for all the possible realizations of the parameter, \(\Delta\), we review the material described in the sequel. Let the coordinates of each vertex of the hypercube \(H := \{\delta \in \mathbb{R}^{n\Delta} : |\delta| \leq 1\}\) be denoted by \(\theta_i, i = 1, \ldots, 2^{n\Delta}\). Using (3), we compute \(X_{\theta_i}(k)\) with \(\Delta^* = \theta_i\). Thus, the set of all possible states at time \(k+1\) can be obtained by

\[
X(k+1) = \bigcup_{\theta_i \in H} \text{Set}(M_{\theta_i}(k+1), m_{\theta_i}(k+1))
\]

where we make the union for all the vertices \(\theta_i\) and where \(M_{\theta_i}\) and \(m_{\theta_i}\) are obtained using (3). The convex hull, \(\hat{X}(k+1)\), of set \(X(k+1)\) is then obtained by using the methods described in [22], since, in general, the set \(X(k+1)\) is non-convex even if \(X(k)\) is convex. For additional properties of the set \(X(k)\), the interested reader is referred to [23] and references therein.

The main goal of this paper is to take the sets constructed in this manner and make the agents converge to a consensus both in position and velocity.

### III. Broadcast Solution Using Position

We start by looking at the problem of when the measuring sensors track each of the agents in the desired angle of communication and then forward a message containing this information. In this case, each node in that strip will receive, at the same time instant, the position and velocity measurements for all its neighbors, which, however, may have been taken at different time instants. In such a setup, all the neighboring nodes have the same information and will perform the same tasks.

We propose a solution where each node takes the Minkowski sum (i.e., \(X + Y\) denotes the set of vectors \(z \in \mathbb{R}^n\) such that \(z = x + y\) for \(x \in X\) and \(y \in Y\)) of the received position sets and calculates which actuation it should use to drive itself to that position.

The new position set is given by

\[
X_i(k+1) = \alpha X_i(k) + (1 - \alpha) \frac{1}{|N_i|} \sum_{j \in N_i} X_j(k)
\]

where the parameter \(\alpha\) is used to model a possible drawback from having node \(i\) changing too much its position. Each node can have different values for \(\alpha\) to reflect their diversity. We stress again that the sets \(X_j(k)\) are built using the SVO update scheme from the sets \(X_j(k - k_j)\) that were received and which correspond to the position and velocity estimates of the agents in the vicinity.

The actual control law can be found by computing the translation that better changes \(X_i(k)\) to fit \(X_i(k+1)\). This resorts to solving an optimization problem to find such control input, i.e.,

\[
v = \arg \min_{x,y} \max_j \left(\|(v + x) - y_j\|\right)
\]

subject to

\[
x \in X_i(k), \quad y_j \in X_j(k)
\]

where \(X_i(k+1)\) is defined as in (3).

Our optimization variable is \(v\), the translation vector, which is equivalent to the velocity vector that is needed to drive the system from where it is at the present time instant, to the weighted average of the set-valued positions of the remaining nodes in the vicinity of the node.

Alternatively, one could also solve the problem in a slightly different setting by focusing on reducing the distance of the node position to that of its neighbors. The problem would be rewritten as

\[
v = \arg \min_{x,y} \max_j \left(\|(v + x) - y_j\|\right)
\]

subject to

\[
x \in X_i(k), \quad y_j \in X_j(k)
\]

(5)

The focus of this work is not on addressing this issue, but it is stressed that one possible approach to this problem is to find the circumference that best fits each of the polygon in two dimensions, and then use their centers to compute the translation vector. Depending on the agent dynamics, the control input will be different but, essentially, aims at driving the position according to the vector \(v\). The whole algorithm can be summarized in Algorithm 1.

#### Algorithm 1 Set-consensus without position estimation

**Require:** Sets \(X_j(k - k_j)\).

**Ensure:** Computation of the velocity to be applied.

1. **for each** \(j\) **do**
2. \(\text{/* Update sets } X_j(k - k_j) \text{ to get } X_j(k) */\)
3. \(X_j(k) = \text{updateSVO}(X_j(k - k_j))\)
4. **end for**
5. \(\text{/* Find translation vector } v */\)
6. \(v = v^* \text{ where } v^* \text{ is found using (4) or (5)}\)
7. \(\text{/* Compute } u */\)
8. \(u(k) = v\)

### IV. Unicast Solution Using Estimation

In the previous section, we discussed how the nodes can determine their velocities based on the information received regarding their neighbors. In such a scenario, the nodes did not need to take into account the variability of the actions of their neighbors, as they all would receive the
same information. In this section, we consider a different assumption that the tower will send unicast messages with the information of a single agent. However, due to the shared medium, their neighbors will also receive that information. In such a setup, the destination of the message is unaware of their neighbors, but their neighbors discover their presence since they also receive the message.

The proposed solution is to augment the state of the SVOs with the states of the neighbors. The set $X_i(k)$ will be constructed using the tools described in Section II, but considering the information received as observation of the whole system with states as the concatenation of positions $x_i$ and $x_j$, $j \in N_i$, i.e., all the neighbors of node $i$.

In order to take into account the possible actions that neighboring nodes take as the result of receiving information for their own neighbors, we use a disturbance term as in (1). The new definition for the set $X_i$ means that, before calculating the control input, we need to project $X_i$ on the $i$th coordinate to obtain the set-valued estimate for node $i$ position and discard the positions of the neighboring nodes.

The new algorithm is as described in Algorithm 2.

**Algorithm 2** Set-consensus with position estimation

**Require:** Sets $X_j(k-k_j)$.
**Ensure:** Computation of the velocity to be applied.

1. for each $j$ do
   2. /* Construct set $X_i(k)$ */
   3. Add an observation to (3) corresponding to $X_j(k-k_j)$
4. end for
5. $X_i(k) = \text{updateSVO}(k)$ using (3)
6. $X_i(k) = \text{projects}(X_i(k), i)$
7. /* Find translation vector $v$ */
8. $v = v^*$ where $v^*$ is found using (4) or (5)
9. /* Compute $u$ */
10. $u(k) = f(v)$

### V. MAIN PROPERTIES

In this section, we present a convergence result for the proposed algorithm which ensures that all the nodes converge to a cluster, where the distance among themselves depends on the size (or uncertainty) of the sets, as described in the sequel. We define an overbounding ball of radius $\epsilon$ and center $c$ to be denoted as $B_{\epsilon}(c)$.

**Theorem 1:** Take $n$ nodes running Algorithm 2 and define $\epsilon_{\text{max}}$ such that $\exists \epsilon_i, i \leq n : \forall k, X_i(k) \subset B_{\epsilon_{\text{max}}}(c_i)$. Then, all of the nodes converge to at most $m$ clusters, where each of these clusters is defined as a neighborhood $2\epsilon_{\text{max}}$, i.e., for a given center $c_g$, $g \leq m$ we have that $\exists g$:

$$\forall i : \lim_{k \to \infty} x_i(k) \in B_{\epsilon_{\text{max}}}(c_g)$$

**Proof:** Let us start by using the assumption that there is an overbounding ball at all times for the sets $X_i(k)$, which means that we can study the convergence of $B_{\epsilon_{\text{max}}}(c_i(k))$ instead of the sets $X_i(k)$, where we made explicit that the center varies with time.

Notice that the control input $u(k)$ is only going to shift $c_i(k)$, which means that we can focus on determining if the centers of the bounding balls are converging.

Let us define a Lyapunov function for the evolution of the centers of the overbounding balls

$$V(k) = \max_{i,j} ||c_i(k) - c_j(k)||$$

which obviously is bounded below since the distance cannot be negative. Take node $i$ to be the one with the largest $x$ coordinate and $j$ to be the one with the smallest (the same reasoning applies to the $y$ coordinate). From solving the optimization problem (5), $[c_i(k+1)]_x \leq [c_i(k)]_x$ and $[c_j(k+1)]_x \geq [c_j(k)]_x$ since both nodes minimize their distance to the remaining nodes. Thus,

$$V(k+1) \leq V(k),$$

and the inequality is only strict if the nodes $i$ belong to the same neighborhood. Therefore, if we divide the analysis for each of the strips of terrain covered by the antennae, we get the strict inequality, meaning that $V(k)$ is monotonically decreasing. In addition with the fact that $V(k) > 0$ except when all the centers are equal and, in which case, $V(k+1) = V(k)$. By the discrete-time version of the La Salle Invariance Principle, the centers are all converging to a common value as $k \to \infty$, at which point, $\max_{i,j} ||x_i(k) - x_j(k)|| = 2\epsilon_{\text{max}}$, thus concluding the proof.

The previous result states that the convergence for a static partition of the field is only going to yield the formation of $m$ clusters, where $m$ is exactly the number of partitions. The conclusion comes from the fact that the centers of the polytopes are all converging to a weighted average of their centers and, therefore, away from the limits of the partitions. We can see this result as the convergence of a consensus algorithm for a partitioned connectivity graph.

In the next section, we will use a simple setup to show that by varying the partitioning method to a simple round-robin partition along the two dimension of the ground yields a convergence to a single cluster, i.e., Theorem 1 is satisfied with $m = 1$.

### VI. SIMULATION RESULTS

In this section, we present some simulations results about the convergence of the true positions of the agents when only a set-valued measurement is available that is guaranteed to contain the true state. In particular, we look at a simple round-robin policy to make the nodes converge to a single cluster instead of $m$ clusters depending on the number of partitions for the ground.

The simulations considered 200 nodes randomly distributed across a 50mx50m square field with an antennae mounted on both sides. Nodes will receive the information transmitted and move according to the proposed algorithm but will only have access to the set-valued estimates of their positions, and not the true (noise-free) positions.
We consider two different scenarios: one where only one of the antennae is functioning and dividing the field into 10 partitions along the \( x \) coordinate, going in a round robin fashion over them and using an offset value to cover different ground strips at each time; and a second example, where both antennae are working in the same way but alternating every 5 time instants.

Figure 2 depicts the final distribution of the 200 nodes for the first case. In this particular run, the number of clusters is \( m = 5 \) and it is observed a common behavior where nodes align themselves with the strips of the ground. The reduction of the number of clusters from 10 to 5 is justified by the offset of the transmissions as it increases the connectivity of the network, in the sense that nodes will belong to different clusters in different time instants.

In Figure 3, it is shown the evolution of the maximum distance between two nodes in the network over time. This measure illustrates how the performance of the consensus algorithm degrades due to the poor choice of the field striping. Nevertheless, it is possible to detect when the cluster convergence happened by looking at when the maximum distance between any pair of nodes converged.

Based on these results, we introduced the second scenario where the antennae work in alternation. In this setup, an antenna along the horizontal axis and another in the vertical axis transmit in a round robin fashion between them and transmitting in a round robin between their own partitions. The idea is to increase the connectivity and to explore the fact that more nodes will belong to more than one partition over time. The study of different partitioning methods is left as a path for future research.

In Figure 4, it is shown the final distribution of the nodes after 100 time instants. We assumed a maximum radius for all measurements of 1 and the ball \( B_{\epsilon_{\text{max}}}(c) \) is shown where \( c \) was computed as the average of the centers for the overbounding balls for each measurement set. Following Theorem 1, all the nodes converged to a single cluster with radius \( \epsilon_{\text{max}} \).

In comparison with the previous scenario, we computed the maximum distance between two nodes to have a sense of the convergence rate of the algorithm. We remark that studying this convergence rate is an interesting topic even though its improvement depends on the partition schedule.

When implementing a stopping time for the nodes to declare convergence, a possibility is to consider whether the measurements that they are receiving are similar to that of the remaining neighbors, or if the current set-valued position estimate is close to the final destination of the node.

VII. CONCLUSIONS AND FUTURE WORK

This paper addressed the problem of reaching consensus based on sparse information, which is acquired by the agents...
in an asynchronous manner. Moreover, it is assumed that the measurements of the variables of interest, in particular the position and velocities of the agents, are corrupted by noise.

The solution adopted herein revolves around the concept of Set-Valued Observers, due to their ability to cope with asynchronous measurements, but also to propagate the state dynamics in the absence of sensor data.

Two scenarios are considered, namely i) one where the agents have information regarding the state and knowledge of their neighbors, and ii) another where the agents do not have such information available, and thus are uncertain regarding the future actions of their neighbors. In both cases, it is considered that the members in the vicinity of a node/agent can vary with time.

It is shown that, within the first scenario, the agents gather in clusters generated by the strips of information, while for the second setup all of the agents converge to the same region. Future research direction, therefore, may include devising an appropriate region segmentation, such that convergence is attained even for the first scenario. In addition, an adequate distribution of the information may also be exploited to increase the rate of convergence of the agents.

REFERENCES


