Distributed consensus for metamorphic systems using a gossip algorithm for $CAT(0)$ metric spaces

Anass Bellachehab and Jérémie Jakubowicz

Telecom SudParis, Institut Mines-Télécom, UMR CNRS 5157 SAMOVAR
9 Rue Charles Fourier, 91000 Évry
France

Abstract. We present an application of distributed consensus algorithms to metamorphic systems. A metamorphic system is a set of identical units that can self-assemble to form a rigid structure. For instance, one can think of a robotic arm composed of multiple links connected by joints. The system can change its shape in order to adapt to different environments via reconfiguration of its constituting units. We assume in this work that several metamorphic systems form a network: two systems are connected whenever they are able to communicate with each other. The aim of this paper is to propose a distributed algorithm that synchronizes all the systems in the network. Synchronizing means that all the systems should end up having the same configuration. This aim is achieved in two steps: (i) we cast the problem as a consensus problem on a metric space and (ii) we use a recent distributed consensus algorithm that only make use of metrical notions.

Keywords: gossip protocol, cubical complex, metamorphic systems

PACS: 89.70

INTRODUCTION

Recently, there has been considerable progress in the design and building of reconﬁgurable or metamorphic robots [1, 2, 3]. These robotic systems are constituted of multiple identical modules that can connect/disconnect from each other and form a rigid structure, referred to as a conﬁguration or a state. Each single metamorphic system can dynamically reconfigure, i.e change its state by rearranging its modules according to a set of predefined rules. We assume that the multiple metamorphic systems are able to communicate with each other and share information about their respective states, thus forming a communication network. We are interested in the problem of distributed consensus among metamorphic systems: each system is in its own initial state, and we would like all of them to end up in a common conﬁguration.

Distributed consensus is a fundamental and thoroughly studied problem. There are several well known algorithms solving it [4, 5]. The majority of these algorithms either deal with vector data (e.g. [6]) or ordered data (e.g. [7]). In [6] for instance, a distributed consensus algorithm is detailed that uses pairwise arithmetic averages of
the data. But, without specific assumptions, configurations cannot be averaged. A distributed consensus algorithm is provided in [7] that relies on pairwise maximum computations. Unfortunately, in the setting of metamorphic robots, configurations cannot be easily ordered. Hence, the need to find an appropriate framework to cast the problem.

In [8, 9], an embedding of the state space – the space of all possible states of a given metamorphic system – into a continuous space is proposed. It turns out, that this framework is well suited to a recent consensus algorithm [10] that relies on pairwise midpoint computations. More precisely, following [11], the state space embedding yields a $\text{CAT}(0)$ metric space. The main contribution of this paper is to propose a distributed consensus algorithm that provably converges while being well adapted to the state space of metamorphic systems.

The paper is organized as follows. The first section details the mathematical background underpinning the metamorphic systems state space embedding, as well as give a mathematical description of the consensus problem. The second section exposes the random pairwise midpoint algorithm, and the role of the $\text{CAT}(0)$ metric condition to ensure convergence.

**FRAMEWORK**

**Metamorphic systems: Definition and examples**

A definition of metamorphic systems is given by E.H. Ostergaard in [1]. A metamorphic system (or reconfigurable system) is a system:

1. That consists of several identical and physically independent unit modules;
2. Whose unit modules can be connected to each other in many possible ways in order to form rigid structures;
3. Whose unit modules can disconnect and reconnect while the system is active;
4. That can by itself change the way the modules are connected, i.e. it is fully automatic.

A popular type of metamorphic systems is the lattice-based system, where the unit modules are located on a lattice like the atoms of a crystal. Each two adjacent modules are supposed to be connected to each other; a module can disconnect from a neighbor and move to another cell of the lattice following a set of prescribed rules. While many modules may move simultaneously during a reconfiguration, their movement can always be decomposed into elementary movements which are based on a single module performing a sequence of disconnect, move, and connect actions. Such an elementary action is called a generator [8]. The nature of the lattice depends on the geometry of the modules, it can be hexagonal, squared, dodecahedral, etc....
An example of a metamorphic system is the robotic arm which consists of \( n \) attached links, inside a \( n \times n \) grid with one of its extremities attached to the basepoint \((0,0)\) of the grid (see Figure 1).

![Figure 1. Example of a lattice based reconfigurable system: the robotic arm. The edges in red indicate the presence of a unit module, a black edge indicates its absence. The arm is attached at its basepoint \((0,0)\). Here, an elementary movement has been performed by the module at end of the arm, which changes the overall form of the system.](image1)

We assume in this work that the metamorphic systems form a network; two systems are connected whenever they are able to communicate. The aim is to synchronize all the systems that compose the network, \( i.e. \) all the systems should have the same configuration, as shown in Figure 2.

Representing the various states of a metamorphic system by their lattice configuration will prove to be insufficient for finding a simple consensus protocol. Following [8] we represent a system configuration as a point in a cubical complex [12, p.97]. The 0-dimensional skeleton of this complex is the set of states, and two vertices are linked by an edge if their corresponding states differ by a single action of a generator. A \( k \)-cube of the complex represents \( k \) commutative movements – \( i.e. \) movements that are non-overlapping whatever their order (see [8] for a rigorous definition).

![Figure 2. In this example we are given five robotic arms. In the leftmost figure – describing the initial state – each arm has its own configuration. The rightmost figure represents a consensus state, in which all the arms share a common configuration.](image2)
Having described the model chosen for metamorphic systems, we next review the mathematical framework of the consensus problem.

**Framework of the consensus problem**

**Network**

Following the approach of [6], we model the network of metamorphic systems by a connected graph $G = (V, E)$ with vertices $V$ represent the metamorphic systems – we will call them agents for short – and whose edges $E$ represent the communication links between these agents. We assume the graph to be undirected, which means that if an agent can communicate with another agent then the converse is also true; this hypothesis is not too unrealistic if we suppose that all the agents are identical and that the movement speed of the agents is very small compared to communication speed. When a communication link exists between two agents we say that the two agents are neighbors. We denote by $\mathcal{N}(v)$ the set of all neighbors of the agent $v \in V$.

As in [6], we assume that the time model is asynchronous, i.e. that each agent has its own Poisson clock that ticks with a common intensity $\lambda$ (the clocks are identically made), and moreover, each clock is independent from the other clocks. When an agent clock ticks, the agent is able to perform some computations and wake up some neighboring agents. This time model has the same probability distribution than a global single clock ticking with intensity $N\lambda$ and selecting uniformly randomly a single agent at each tick. This equivalence is described, e.g. in [6]. Notice also that link $e = \{v, w\}$ is not necessarily used by agents $v$ and $w$ at a given time.

**Data space**

Each node $v \in V$ can store a value $x_v \in \mathcal{S}$, where we recall $\mathcal{S}$ is the state complex mentioned above. The goal is to drive the agents towards a consensus state i.e a state where all the agents agree on a common value $x \in \mathcal{S}$. We assume that the cubical complex is equipped with the metric induced from the Euclidean $L^2$ metric on each cube. The 1-skeleton of the complex is a metric graph called the transition graph.

Initially each node $v$ has an initial value $x_v(0)$ and $X_0 = (x_1(0), \ldots, x_N(0))$ is the tuple of initial values in the network. Obviously, a consensus state has the form $X_\infty = (x_\infty, \ldots, x_\infty)$ with: $x_\infty \in \mathcal{S}$. We denote by $x_v(k)$ the value stored by the agent $v \in V$ at the $k$-th iteration of the algorithm, and $X_k = (x_1(k), \ldots, x_N(k))$ the global state of the system at instant $k$. 
An additional requirement for the state complex $\mathcal{S}$ is to be a $\text{CAT}(0)$ metric space, meaning a space in which the $\text{CAT}(0)$ inequality is verified:

**Definition 1 (CAT(0) inequality).** Assume $(X,d)$ is a metric space and $\Delta = (c_0, c_1, c_2)$ is a geodesic triangle with vertices $p = c_0(0)$, $q = c_1(0)$ and $r = c_2(0)$. Let $\bar{\Delta} = (\bar{p}, \bar{q}, \bar{r})$ denote a comparison triangle (a triangle with same edge lengths as $\Delta$) in $\mathbb{R}^2$. $\Delta$ is said to satisfy the $\text{CAT}(0)$ inequality if for any $x = c_0(t)$ and $y = c_2(t')$, one has:

$$d(x,y) \leq \bar{d}(\bar{x}, \bar{y})$$

where $\bar{x}$ is the unique point of $[\bar{p}, \bar{q}]$ such that $d(p,x) = \bar{d}(\bar{p}, \bar{x})$ and $\bar{y}$ on $[\bar{p}, \bar{r}]$ such that $d(p,y) = \bar{d}(\bar{p}, \bar{y})$.

A metric space is said to be locally $\text{CAT}(0)$, if any sufficiently small geodesic triangle verifies the $\text{CAT}(0)$ inequality. It is said to be globally $\text{CAT}(0)$ if any geodesic triangle verifies the $\text{CAT}(0)$ inequality.

Any state complex can be shown to be a locally $\text{CAT}(0)$ space [11]. The global $\text{CAT}(0)$ propriety requires an additional constraint on the state complex. In [11], a combinatorial criterion based on the notion of posets with inconsistent pairs is provided to verify whether a state complex is globally $\text{CAT}(0)$. In this paper we shall suppose that the state complex of the metamorphic systems involved is globally $\text{CAT}(0)$.

The following proposition links the existence and uniqueness of geodesics and midpoints with the $\text{CAT}(0)$ propriety:

**Proposition 1.** If $x$ and $y$ are two points in a globally $\text{CAT}(0)$ space; there is a unique geodesic $[x,y]$ and the midpoint denoted by: $\langle x + y \rangle$ is always well defined and unique.

In the next section we expose a distributed algorithm that relies on distributed midpoint computation in $\mathcal{S}$ to drive a system of identical metamorphic systems into a consensus configuration.

**ALGORITHM**

**Description**

Assuming the following hypotheses:

1. The graph is undirected and connected
2. The clocks of the agents are independent, identical Poisson clocks
3. The data space $\mathcal{S}$ is a complete, globally $\text{CAT}(0)$ metric space

We propose the following algorithm [10]:

...
Algorithm Random Pairwise Midpoint

**Input:** a graph $G = (V, E)$ and the initial nodes configuration $X_v(0), v \in V$

for all $k > 0$ do

At instant $k$, uniformly randomly choose a node $V_k$ from $V$ and a node $W_k$ uniformly randomly from $N(V_k)$.

Update:

$X_{V_k}(k) = \frac{X_{V_k}(k-1) + X_{W_k}(k-1)}{2}$

$X_{W_k}(k) = \frac{X_{V_k}(k-1) + X_{W_k}(k-1)}{2}$

$X_v(k) = X_v(k-1)$ for $v \not\in \{V_k, W_k\}$

end for

---

Calculating the geodesic between two points and their midpoint

In order to implement the Random Pairwise Midpoint algorithm we need to be able to compute the midpoint of any two points $x, y \in \mathcal{S}$. In order to calculate their midpoint, which we will denote by $\langle \frac{x+y}{2} \rangle$, we first need to find the geodesic $[x, y]$ between them. For that we have an algorithm by [11] that takes as its input the cubical complex, the two points $x, y$ and returns a cube sequence $\{C_1 \ldots C_n\}$ (not necessarily of the same dimension) and a series of points $\{p_1, \ldots, p_{n-1}\}$ such that $p_i \in C_i \cap C_{i+1}$ and from which the geodesic will pass.

![Diagram](image)

**FIGURE 3.** Example of the sequence of points $\{p_1, p_2, p_3, p_4, p_5\}$ that the geodesic between $x$ and $y$, drawn in red passes through, the cube sequence $\{C_1 \ldots C_6\}$ contains this geodesic.

After we are given the sequence of points $\{p_1 \ldots p_{n-1}\}$ and the cube sequence we have that: $d(x, y) = d_E(x, p_1) + \sum_{i=1}^{n-2} d_E(x_i, x_{i+1}) + d_E(p_{n-1}, y)$ where $d_E$ is the Euclidean distance induced on each cube.

In order to find $\langle \frac{x+y}{2} \rangle$ the midpoint of $x$ and $y$, first we need to find the cube $C_{i_0}$ containing it. For that let $i_0 = \min \left\{ 1 \leq i \leq n-1 \mid d(x, p_i) \geq \frac{d(x, y)}{2} \right\}$, then we have:
\( \langle \frac{x+y}{2} \rangle \in C_{i_0} \) and \( \langle \frac{x+y}{2} \rangle \in [p_{i_0}, p_{i_0+1}] \) with

\[
d\left( p_{i_0}, \left\langle \frac{x+y}{2} \right\rangle \right) = \frac{d_E(p_{i_0}, p_{i_0+1})}{2} + \frac{1}{2} \sum_{i=i_0+1}^{n} d_E(p_i, p_{i+1}) - \frac{1}{2} \sum_{i=1}^{i_0-1} d_E(p_i, p_{i+1}).
\]

Analysis of the algorithm

It is known from [10] that under the above mentioned hypotheses, the Pairwise Midpoint Algorithm converges asymptotically towards a consensus state, at a linear rate. This analysis treats midpoint computation as a black box single operation and does not address the cost of performing the midpoint computation in terms of the size of the state complex. A thorough analysis of this question can be found in [11], where the authors demonstrate that complexity of computing a geodesic between two points in a \( \text{CAT}(0) \) cubical complex is polynomial with respect to the cardinal of its associated poset with inconsistent pairs.

CONCLUSION

In this paper we have presented a distributed method based on the \( \text{CAT}(0) \) gossip algorithm in order to achieve consensus among several identical metamorphic robots. This approach presents the advantage of having exponential convergence, but is limited by the complexity of constructing the state complex associated to a given metamorphic system, which can sometimes be prohibitive. In the case where one can construct such a state complex, it would interesting to investigate the possibility of implementing a distributed optimization scheme using a distributed version of the proximal point algorithm in \( \text{CAT}(0) \) spaces [13].

ACKNOWLEDGMENTS

This research has been funded by the “Futur et Ruptures” PhD program from Institut Mines-Telecom.

REFERENCES