Real-Time Automated Modeling and Control of Self-Assembling Systems

Grégory Mermoud, Massimo Mastrangeli, Utkarsh Upadhyay and Alcherio Martinoli

***Abstract*— We present the M^3 framework, a formal and generic computational framework for modeling and controlling stochastic distributed systems of purely reactive robots in an automated and real-time fashion. Based on the trajectories of the robots, the framework builds up an internal microscopic representation of the system, which then serves as a blueprint of models at higher abstraction levels. These models are then calibrated using a Maximum Likelihood Estimation (MLE) algorithm. We illustrate the structure and performance of the framework by performing the online optimization of a bang-bang controller for the stochastic self-assembly of water-floating, magnetically latching, passive modules. The experimental results demonstrate that the generated models can successfully optimize the assembly of desired structures.**

I. Introduction

The controlled formation of structures and patterns is a fundamental task of distributed robotic systems. In particular, self-assembling systems come in several varieties, which can be classified based on, for instance, the role of energy and information, the control modality (e.g., centralized versus distributed), the size of the components and their type—active (e.g., autonomous robots) or passive (e.g., MEMS, macromolecules). Overall, the ongoing convergence between robotic minimalism and the increasing sophistication of micro/nanosystems allows one to envision a unifying perspective for self-assembling systems across scales [1]. While soundly grounded in current technological trends and supported by a few remarkable theoretical frameworks, such appealing perspective still lacks substantial and consolidated modeling methodologies, as compared to the wide-ranging efforts in components development [2], [3].

Among the approaches to model robotic systems, hybrid automata stand out for capturing both continuous and discrete state variables [4]. When dealing with distributed systems, one often uses a combination of multiple abstraction levels, which often includes probabilistic and graphical models [5]. However, most modeling methodologies are not sufficiently systematic to be carried out in an automatic fashion [5], [4], [6]. Alternative methods for automated model construction adopt completely different strategies based on evolutionary computation [7]. In spite of their attractive flexibility and versatility, these methods are computationally expensive, and they yield gray-box models whose structure and parameters are difficult to anchor back to the original system. In particular, they rely on a single level of abstraction, even in the context of collective systems (e.g., biological and chemical reaction networks [8], [9]), thereby precluding any mapping between microscopic and macroscopic states.

The control of self-assembling system has been tackled by several previous works, which generally rely on control of local interactions among building blocks [10], [11], [12]. Napp et al. [13] control the formation of heterodimers by adjusting only global parameters of the system (i.e., light intensity); however, the underlying model is designed and calibrated manually, and they do not investigate the formation of more complex structures.

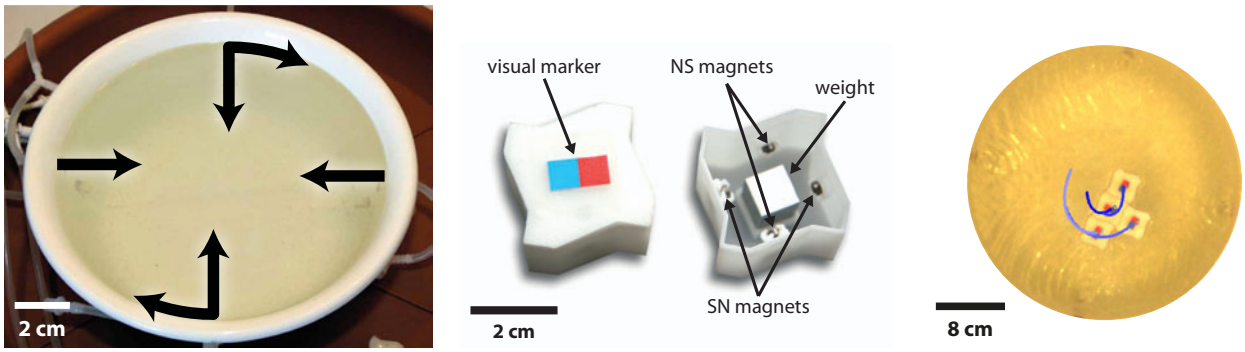
In this work, we present the M^3 framework, a formal and generic computational framework that allows for the automatic construction and calibration of models of distributed stochastic systems of purely reactive robots. Based on a generic microscopic representation of a system, the framework generates an associated Chemical Reaction Network (CRN) model in real-time, and allows for an optimal, global control of the system as well. We hereby experimentally demonstrate the performance of the M^3 framework by modeling and optimizing, in real time and without human supervision, a stochastic self-assembling system of water-floating passive modules.

II. Experimental Setup

We study the stochastic self-assembly of target structures of 3-cm-sized water-floating devices, denoted as *blocks* hereafter, within a circular water-filled tank. The experimental setup consists of the tank, with six inlets and four outlets connected to four diaphragm pumps (see Fig. 1(a)), four blocks, an overhead camera and a workstation. The cuboidal, centrosymmetric blocks are passive and endowed with four SmCo permanent magnets (one on each side's center) for mutual aggregation, as well as with a visual marker for tracking purposes (see Fig. 1(b)). The weight of each block (about 17.3 g, compared to a buoyancy limit of 21.9 g) was trimmed for reliable floatation. The blocks are not self-locomoted; instead, they are stirred by the fluid flow produced within the tank by the peripheral pumps. As a result, the blocks describe trajectories with well-defined geometric features, yet with a strong stochastic component [14].

The tank, of approximately 30 cm in diameter, has four inlets perpendicular to the wall and other two almost tangential, allowing to create flows with both radial and circular components. Additionally, the four outlets are placed at the bottom of the tank so as not to interfere with the surface flow.

The authors are with the Distributed Intelligent Systems and Algorithms Laboratory (DISAL), School of Architecture, Civil and Environmental Engineering, École Polytechnique Fédérale de Lausanne. Grégory Mermoud and Massimo Mastrangeli are sponsored by the Nano-Tera.ch research initiative in the framework of the SelfSys project. gregory.mermoud@epfl.ch



(a) Water-filled tank (b) Self-assembling block (c) Visual tracking

Fig. 1. The experimental setup: (a) Water-filled tank with 6 inlets (4 orthogonal and 2 tangential to the wall); (b) Internal details of a water-floating block, including the latching mechanism composed of four permanent magnets with different pole orientation—north-south (NS) and south-north (SN), respectively; (c) Real-time visual tracking of four blocks during an experiment (the blue lines show a short history of the trajectory of each block).

Each pump's flow rate can be controlled individually up to a maximal value of 600 ml/min. This flexible configuration allows us to investigate a variety of different flow patterns and associated block trajectories. Indeed, perpendicular inlets generate irregular trajectories, and induce block collisions in the middle of the tank, but they exhibit dead spots near the walls. Conversely, tangential inlets generate circular flows that prevent dead spots, but lead to regular, closed trajectories which do not favor collisions.

The combined effects of mutual magnetic forces and block shape geometry lead to the precise pair-wise self-alignment of blocks upon close proximity (about 0.5 cm), when not hampered by fluidic drag forces. In fact, the interblock bonds were designed to be reversible, as depending on the interplay between the magnitude of the magnetic (about 16 mN per bond according to FEM simulations) and the local hydrodynamic forces acting on the blocks, the latter being modulated by the fluid flow regimes. As a result, the stability of all the assembled block structures corresponding to local system energy minima could be controlled by the modes of fluidic stirring in the tank, whereas the 2-by-2 square structure—labelled D in Fig. 2 and corresponding to the global system energy minimum—was irreversible and acted as absorbing state in the system dynamics.

To monitor the evolution of the system in real time, we use an overhead camera to track a two-color passive marker located at the top of each block. SwisTrack [15], an open-source software package developed in our laboratory, allows us to track the pose of the blocks. Both their position (x,y) and orientation (θ) are logged in real time at a rate of approximately 30 Hz[1]. These data are then transmitted to the modules responsible for the construction of the model and the optimization, described in sections IV and V.

III. Problem Statement

All feasible structures formed by the assembly of four of our blocks are reachable and can be enumerated (see Fig. 2). In a well-mixed system, each structure has an intrinsic probability of being formed, which depends not only on its own geometry, but also on the parameters of the system. For instance, the assembly A is unlikely to be observed in a small tank because of the high probability of collision between the blocks.

[1]The sampling period (about 33 ms) is much smaller than the average inter-collision time, which is around 1.15 s.

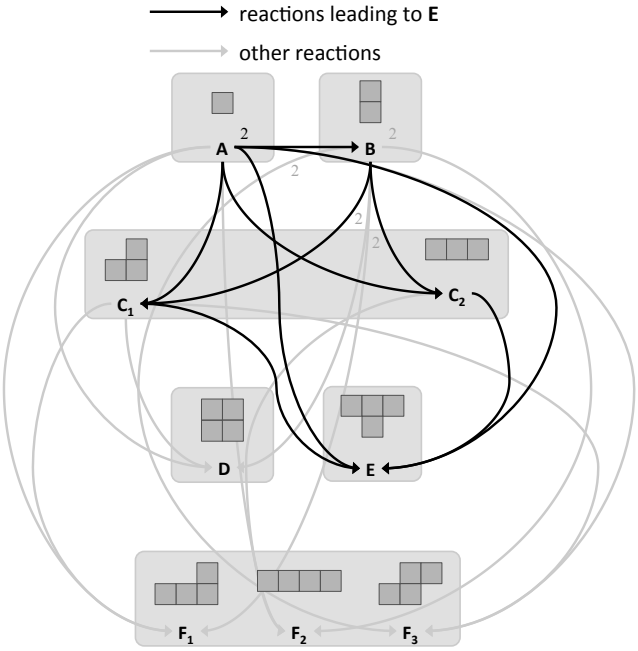


Fig. 2. Graphical representation of all assemblies that can be formed out of four blocks and the forward reactions that lead to them. Chiral copies of assemblies F_1 and F_3 are not included. The shaded rectangles indicate assemblies with the same connection topology (using a 4-neighbors topology). Black arrows denote the reactions that lead to the target structure E, whereas gray arrows other forward reactions in the system. For readability's sake, reverse reactions and stoichiometric factors equal to 1 are omitted.

In this work, we consider a non-well-mixed system (see [16] for more details about this type of systems), which allows us to further tune the probabilities of structure formation by optimizing dynamically the control parameters of the system (e.g., those governing the agitation of the system). More formally, the research question that we address in this

paper is the following: Given a stochastic multi-unit system with a finite set of agitation modes $\mathcal{M} = \{m_0, \ldots, m_n\}$, what is the mode m_i to be selected at time t that minimizes the time to form a given target structure T?

In the present case, we consider only two modes of agitation (as in a bang-bang controller) corresponding to two different pump configurations that lead to different agitation schemes. In mode m_0, the fluid flow induces smooth and regular block trajectories and only marginal differences in their relative velocities, thereby allowing for a high stability of the formed aggregates but relatively few interactions. In mode m_1, the blocks exhibit a much more erratic movement, dominated by the stochastic perturbations of the water surface (Faraday waves) caused by pumps-induced tank vibrations. The consequently higher kinetic energy of the blocks increases the collision rate, but also the instability of the aggregates.

IV. Modeling Framework

In this work, our methodology makes the fundamental assumption that the robots are strictly *reactive*, that is, *all behavioral changes can be interpreted as the result of interactions with other robots or with the environment*. A corollary of this assumption is that one can associate each behavior of the robot to some condition on its interaction configuration. Indeed, when designing a robot's controller, one naturally arranges the different interaction configurations that the robot can be in into classes indexed by a set of behaviors. For instance, the designer can group all situations in which the robot is close to another robot, and associate the resulting class to a *obstacle avoidance* behavior. Following this methodology, the controller of each robot naturally reflects the most important states of the robot. As a result, one can use the robot's controller as a blueprint to construct a meaningful partition of the continuous phase space, and thereby deriving models at higher abstraction levels (see for instance [5]). This approach opens up new opportunities for automating model construction and calibration.

We hereby rely on a computational framework, called the M^3 framework, which allows for the automatic construction of models of a multi-robot system. The global structure of this framework is depicted in Fig. 3. The general idea of our approach is that we observe (or simulate) an existing system, and the model is built based on the observations (i.e., trajectories) collected during these experiments (or simulations). Internally, the framework builds up a microscopic representation of the system based on these observations as well as on a list of interactions of interest specified by the user. This representation, called the Canonical Microscopic Model (CMM), is a *formal* and *generic* description of a reactive multi-unit system, and it serves as a blueprint for the construction of a macroscopic model, specified using the Chemical Reaction Network (CRN) formalism. Finally, at each time step, the optimal mode of agitation is determined using the optimization scheme described in Section V, and transmitted to the (real or simulated) actuators.

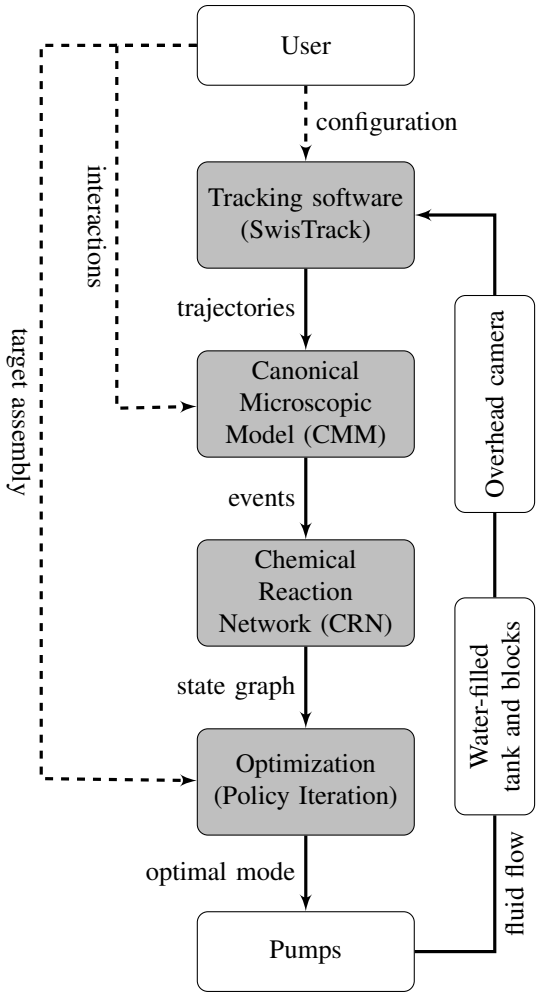


Fig. 3. Overview of the M^3 framework as deployed in this study, and the different types of information flowing between its constitutive modules. Gray-shaded nodes are computational entities whereas other nodes are physical entities. Dashed arrows denote flows that are not automated, but need to be performed only once prior to the experiment. Note that the closed-loop control is completely automated.

Hereafter, we provide a brief summary of the theoretical foundations of this framework. A more detailed description will be published elsewhere.

A. Canonical Microscopic Model (CMM)

We define our system as a set of coupled hybrid automata [17], called *particles* that interact through a set of interactions $\mathcal{I} = \{I_1, \ldots, I_t\}$ (see Definition 1). Each particle in the set $\mathcal{P} = \{P_1, \ldots, P_m\}$ represents a robot in the target system. The set $\mathcal{P}$ may be partitioned into an arbitrary set of *classes* of particles $\mathcal{C}_i$ (such that $\bigcup_i \mathcal{C}_i = \mathcal{P}$), which denote particles that have similar control graphs (see below). The state of each robot has two distinct components: (i) an n-dimensional *continuous* component $\vec{x} = [x_1, \ldots, x_n]^\mathsf{T} \in \mathbb{X}$ that typically denotes the physical state of the particle (e.g., its position, orientation and velocity in physical space, its temperature, its battery level, etc.)[2], and (ii) a *discrete*

[2]We write $\mathcal{X}_i = \{x_{1,i}, \ldots, x_{n,i}\}$ the *set* of state variables of particle P_i.

component $\xi \in \mathcal{V}$ that denotes the logical state of the particle, called *control mode*, that is, a vertex of the finite directed multigraph $G = (\mathcal{V}, \mathcal{E})$, called the *control graph* of P. The edges in $\mathcal{E}$ are called *control switches*. Most importantly, each control mode ξ_j is labeled with a unique function $\phi_j : \mathcal{I} \to \mathbb{N}_0$, called an *interaction configuration*. The function $\phi_j(I)$ denotes the number of interactions of type I that are active in mode ξ_j. In other words, each control mode is associated to a unique interaction configuration, and vice versa, such that there exists a one-to-one map $\Phi : \mathcal{V} \to \{\phi_1, \ldots, \phi_k\}$.

Definition 1 [Interaction] An interaction I is defined as a triplet $(\mathcal{C}_i, \mathcal{C}_j, cond)$ where $\mathcal{C}_i$ and $\mathcal{C}_j$ are two classes of particles that may interact through I; the predicate $cond$ describes the conditions in which I is active, and whose free variables are in $\mathcal{X}_{i'} \cup \mathcal{X}_{j'}$, with i', j' the indices of the interacting particles.

The CMM exhibits a few key properties: (i) it describes a given distributed system as a set of coupled hybrid automata, thereby allowing for a natural coupling between the continuous and discrete components of the state space; (ii) the underlying assumptions of the CMM allow the algorithmic construction of the control space $\mathcal{V}$ of its constitutive particles solely based on their trajectories in the *continuous state space* $\mathbb{X}$; (iii) because the control modes in $\mathcal{V}$ are mapped to a unique interaction configuration, they form a partition of the *continuous phase space* $\mathbb{S}$, that is, the continuous space of the entire system; (iv) ultimately, by a proper aggregation of those control modes, one can obtain a more tractable and meaningful set of *meta*states, which we denote $q_1, \ldots, q_r$. Importantly, this process of aggregation is precisely the mental process carried out by the designer of a robotic system. The latter *meta*states are the basis for an algorithmic conversion of any CMM into an equivalent macroscopic representation based on the CRN formalism.

As mentioned earlier, the control space is built iteratively as observations of the system are collected (see Algorithm 1): starting from an initial control space $\mathcal{V} = \{0\}$, which contains only a non-interacting mode, each newly observed control mode $\xi > 0$ is appended to $\mathcal{V}$. As depicted by Fig. 3, the construction of the CMM is based on trajectories collected either in simulation or in real experiments. From the point of view of the framework, the nature of the trajectories has no importance.

Algorithm 1 Iterative construction of the control space $\mathcal{V}$

Require: $\mathcal{V}_i = \{0\}, \forall i \in \{1, \ldots, m\}$
  while $t < t_{end}$ **do**
    for each $P_i \in \mathcal{P}$ **do**
      Update $\vec{x}_i(t + \Delta t)$ according to observations
      if an interaction has occurred/ended **then**
        Compute new interaction configuration ϕ'
        if $\neg\exists \xi' \in \mathcal{V}_i$ s.t. $\Phi(\xi') = \phi'$ **then**
          Create ξ' and updates Φ s.t. $\Phi(\xi') = \phi'$
          Append ξ' to $\mathcal{V}_i$ and $e = (\xi_i, \xi')$ to $\mathcal{E}_i$
        end if
        $\xi_i(t + \Delta t) \leftarrow \xi'$
      end if
    end for
    $t \leftarrow t + \Delta t$
  end while

B. From the CMM to Macroscopic Models

The formalism of hybrid automata is interesting from a theoretical point of view, yet not very practical numerically. To enhance its applicability, we hereafter introduce the notion of Chemical Reaction Network (CRN), and we show how one can automatically convert the CMM into a CRN.

The general idea of our approach is to use the control graph G of the particles of the system as a blueprint of the model structure. It is often necessary to refine the resulting models *a posteriori*, in order to account for hidden control modes, which are relevant to either the performance metric or the accuracy of the model. For instance, in aggregation scenarios [18], [19], pieces of information that are critical to the evaluation of the performance metric, such as the size or the shape of the aggregates, are not available to the robots (and therefore not reflected in their controller). To keep track of these global arrangements of interaction, we introduce the notion of *interaction graph* and *aggregate*.

Definition 2 [Interaction Graph] The interaction graph $G^{\text{int}} = (\mathcal{P}, \mathcal{E}^{\text{int}})$ is a graph whose edge set $\mathcal{E}^{\text{int}}$ is given by $(P_i, P_j) \in \mathcal{E}^{\text{int}}$ iff $\exists I \in \mathcal{I}$ such that P_i and P_j interact through I.

Definition 3 [Aggregate] We note $\mathcal{A} = \{A_1, \ldots, A_q\}$ the set of all feasible *aggregates*, that is, connected subgraphs of the interaction graph G^{int}.

Definition 4 [Chemical Reaction Network] Similarly to previous works [20], [21], we define a CRN $\mathfrak{N} = (\mathcal{R}, \mathcal{S})$ as a set of N reactions $\mathcal{R} = \{R_1, \ldots, R_N\}$ acting on a finite number M of species $\mathcal{S} = \{S_1, \ldots, S_M\}$. Each reaction R is defined as two vectors of nonnegative integers specifying the stoichiometry of the reactants, $\vec{r}_R = [r_{R,1}, \ldots, r_{R,M}]$, and the products, $\vec{p}_R = [p_{R,1}, \ldots, p_{R,M}]$, respectively. The stoichiometry denotes how many copies of a given reactant or product is required or produced, respectively, when a reaction takes place. For example, assume a CRN with $\mathcal{S} = \{A, B, C\}$, the reaction $A + 3\,B \rightharpoonup A + 2\,C$ is represented by the following vectors:

$$\vec{r} = [1\ 3\ 0]$$
$$\vec{p} = [1\ 0\ 2]$$

The CRN being a *population* model, it keeps track of *how many* individuals of each species are present in the system at a given time. The state of the CRN is therefore given by the vector $\vec{X} \in \mathbb{N}^M_{\geq 0}$, whose elements specify the number of individuals of each species. A reaction R may occur iff the number of reactants is sufficient, that is, $\vec{X} \geq \vec{r}_R$ element-wise. When reaction R occurs, the new state $\vec{X}'$ is simply given by:

$$\vec{X}' = \vec{X} - \vec{r}_R + \vec{p}_R = \vec{X} + \vec{\nu}_R \tag{1}$$

with $\vec{\nu}_R$ the population change caused by reaction R.

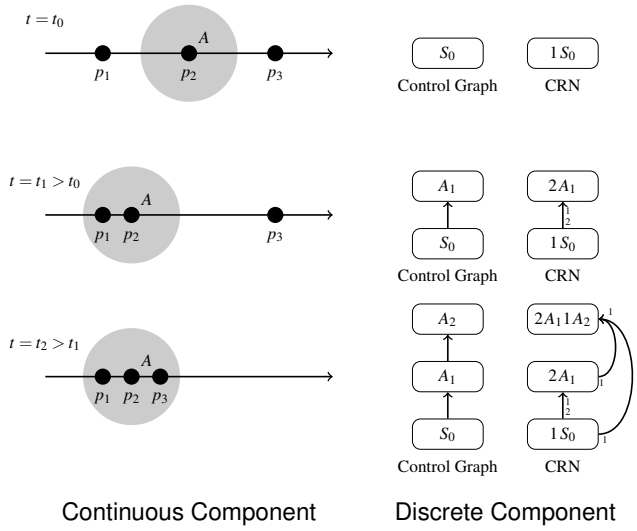


Fig. 4. Illustration of the model construction for a 1-dimensional system of 3 particles $\{P_1, P_2, P_3\}$ of the same class, and one interaction A, which is active whenever two particles are closer than a given distance, illustrated by the gray-shaded circle. At $t = t_0$, no interaction has occurred yet, and both the control graph of the particles contains only one state S_0, which denotes the absence of interaction. Incidentally, the CRN has only one species, and no reaction. Upon the encountering of P_1 and P_2 at $t = t_1$, a new control mode A_1 is appended to the control graph, and the reaction $2 \cdot 1\, S_0 \rightharpoonup 2\, A_1$ is added to the CRN. Finally, at $t = t_2$, P_3 joins the aggregate formed by P_1 and P_2, thereby making P_2 switch to a new control mode A_2, and creating a new reaction $1\, S_0 + 2\, A_1 \rightharpoonup 2\, A_1 1\, A_2$. As a reference, the assemblies C_1 and C_2 in Figure 2 would be written $2\, A_1 1\, A_2$ using this notation.

Finally, another important characterizing quantity for R is its propensity function a_R, which is defined such that $a_R(\vec{x}, \cdot)\, dt$ is the probability that one reaction R will occur in the next time interval $[t, t+dt)$ and $dt \to 0$, given that the current state of the system is $\vec{X}(t) = \vec{x}$. When a_R depends only on the current state of the system (Markov property), waiting times are exponentially distributed, and the CRN is exactly as defined in [21].

Now one can write the CMM as a CRN $\mathfrak{N}$, whose species are the aggregates in $\mathcal{A}$ and the reactions are represented by the interactions in $\mathcal{I}$. Therefore, the reactions

$$A_i + A_j \xrightarrow{I^+} A_k \tag{2}$$

$$A_k \xrightarrow{I^-} A_i + A_j \tag{3}$$

exist iff $A_i \|^I A_j = A_k$.

The only remaining components of $\mathfrak{N}$ to be defined are the propensity functions, whose identification is described in the next section.

1) Rate Identification: In our framework, the analysis of the process dynamics provides a precise estimate of the reaction rates, and, to some extent, a measure of the validity of this estimate. The time t until the next firing of reaction R is an exponential random variable with mean $1/a_R(\vec{x})$, that is, its probability density is given by

$$f(t) = a_R(\vec{x}) \cdot e^{-a_R(\vec{x}) \cdot t} \tag{4}$$

where $\vec{x}$ is the state of the CRN (i.e., a population vector), and $a_R(\cdot)$ is the propensity function of the reaction R. Importantly, the form of $a_R(\cdot)$ depends on the type of the reaction R: assuming that the system is in dynamic equilibrium, one can use the law of mass-action as propensity function (see [21] for more details). For the sake of simplicity, we shall summarize all forms of the propensity function using the following notation:

$$a_R(\vec{x}) = k_R \cdot \widetilde{a}_R(\vec{x}) \tag{5}$$

where k_R is the *rate* of reaction R and $\widetilde{a}_R(\vec{x})$ has the appropriate form according to the stoichiometry of R, and does not depend on k_R.

Therefore, the problem we intend to solve hereafter is the following: Given a sequence of events $(e_1, \ldots, e_n)$, with $e_i = (R_i, t_i, \vec{x}_i)$, what is the most likely rate vector $\hat{\vec{k}} = \left[\hat{k}_1, \ldots, \hat{k}_N\right]$ of the underlying CRN? More formally, we want to solve the following problem:

$$\hat{\vec{k}} = \operatorname*{argmax}_{\vec{k}}\ \mathcal{L}(\vec{k} | e_1, \ldots, e_n) \tag{6}$$

$$= \operatorname*{argmax}_{\vec{k}}\, f(e_1, \ldots, e_n | \vec{k}) \tag{7}$$

where $\mathcal{L}(\vec{k} | e_1, \ldots, e_n)$ is the likelihood of the rate vector $\vec{k}$ given the sequence of events $(e_1, \ldots, e_n)$.

We can write the probability $f(e_i | \vec{k})$ of a single event e_i as follows:

$$f(e_i | \vec{k}) = a_{R_i}(\vec{x}_i) \cdot e^{-a_0(\vec{x}_i) \cdot t_i} \tag{8}$$

where

$$a_0(\vec{x}) \triangleq \sum_{j=1}^{M} a_j(\vec{x}) \tag{9}$$

Since we assume independence of events, we can write:

$$\mathcal{L}(\vec{k} | e_1, \ldots, e_n) = \prod_{i=1}^{n} a_{R_i}(\vec{x}_i) \cdot e^{-a_0(\vec{x}_i) \cdot t_i}. \tag{10}$$

For the sake of simplicity, we will omit the arguments of $\mathcal{L}$ in the sequel.

Now, we can try to solve the optimization problem formulated by Equation 7. To make our problem simpler (both from an analytical and a numerical standpoint), we work with the natural logarithm of the likelihood function:

$$\ln \mathcal{L} = \ln \left(\prod_{i=1}^{n} a_{R_i}(\vec{x}_i) \cdot e^{-a_0(\vec{x}_i) \cdot t_i} \right) \tag{11}$$

$$= \sum_{i=1}^{n} \left(\ln a_{R_i}(\vec{x}_i) - a_0(\vec{x}_i) \cdot t_i \right) \tag{12}$$

First, we need to compute the gradient of the log-likelihood function $\ln \mathcal{L}$:

$$\nabla \ln \mathcal{L} = \left(\frac{\partial \ln \mathcal{L}}{\partial k_{R_1}}, \ldots, \frac{\partial \ln \mathcal{L}}{\partial k_{R_N}} \right) \tag{13}$$

with

$$\frac{\partial \ln \mathcal{L}}{\partial k_{R_j}} = \sum_{i=1}^{n} \left(\frac{1}{a_{R_i}(\vec{x}_i)} \frac{\partial a_{R_i}(\vec{x}_i)}{\partial k_{R_j}} - \frac{\partial a_0(\vec{x}_i)}{\partial k_{R_j}} \cdot t_i \right) \tag{14}$$

where

$$\frac{\partial a_{R_i}(\vec{x}_i)}{\partial k_{R_j}} = \frac{\partial k_{R_i} \cdot \widetilde{a}_{R_i}(\vec{x}_i)}{\partial k_{R_j}} = \begin{cases} \widetilde{a}_{R_i}(\vec{x}_i) & \text{if } R_j = R_i \\ 0 & \text{otherwise} \end{cases} \tag{15}$$

and

$$\frac{\partial a_0(\vec{x}_i)}{\partial k_{R_j}} \cdot t_i = \frac{\partial a_{R_j}(\vec{x}_i)}{\partial k_{R_j}} \cdot t_i = t_i \cdot \widetilde{a}_{R_j}(\vec{x}_i) \tag{16}$$

Replacing these terms into Equation 14, we obtain

$$\frac{\partial \ln \mathcal{L}}{\partial k_{R_j}} = \sum_{i=1}^{n} \left(\frac{\mathbf{1}_{R_i=R_j}}{k_{R_i}} - t_i \cdot \widetilde{a}_{R_j}(\vec{x}_i) \right). \tag{17}$$

where $\mathbf{1}_{R_i=R_j}$ is the indicator function. A local extremum of the function $\ln \mathcal{L}$ corresponds to a zero of the gradient

$$\nabla \ln \mathcal{L} = \big(0, \ldots, 0\big) \tag{18}$$

which is equivalent to writing

$$k_j = \frac{\sum_{i=1}^{n} \mathbf{1}_{\{R_i=R_j\}}}{\sum_{i=1}^{n} \big(t_i \cdot \widetilde{a}_{R_j}(\vec{x}_i)\big)} \tag{19}$$

for $j = 1, \ldots, N$. Importantly, the rate of reaction $R = R_j$ also depends on events that do not involve R. For this point to be a maximum of $\ln \mathcal{L}$, we need the Hessian matrix $H(\ln \mathcal{L})$ to be negative-definite, which can be easily demonstrated.

If the waiting times of a reaction are not exponentially distributed, it means either that the underlying reaction is not memoryless, or that some relevant features have been neglected (in particular, spatiality). In such cases, one would need to either (i) use an appropriate simulation scheme, in order to capture the characteristic distribution of the waiting times, or (ii) modify the structure of the model to mitigate the impact of this reaction (e.g., by augmenting the state space of the model with states that account for previous reaction partners, as proposed by Napp *et al.* [16]). However, such research avenues are beyond the scope of this paper, and we do not perform any further refinement of the resulting models.

V. Optimization

As stated in Section III, we aim at favoring the formation of a predefined target structure T. We show how this problem is equivalent to another well-known problem, that is, the solving of Markov Decision Processes (MDPs). Indeed, forming the structure T is equivalent to attaining a target population $\vec{x}_t = (x_{t,1}, \ldots, x_{t,M})$ such that

$$x_{t,i} = \begin{cases} 1 & \text{if } S_i = T, \\ 0 & \text{otherwise.} \end{cases} \tag{20}$$

Therefore, our problem consists in determining the mode $m_s \in \mathcal{M}$ to be selected given an initial population $\vec{x}_s$ such that the expected time to reach $\vec{x}_t$ is minimized.

For each mode $m \in \mathcal{M}$, we have an estimate of the propensity function $a_R^{(m)}(\vec{x})$ for each reaction $R \in \mathcal{R}$. Denote $k_{ij}^{(m)} = a_R^{(m)}(\vec{x}_i)$ the rate of the reaction R whose associated population change is $\vec{\nu}_R = \vec{x}_j - \vec{x}_i$ if mode m is selected. Furthermore, we define the following quantities:

$$\lambda_i^{(m)} = \sum_j k_{ij}^{(m)}, \qquad p_{ij}^{(m)} = \frac{k_{ij}^{(m)}}{\lambda_i^{(m)}}. \tag{21}$$

Note that each state has only one optimal choice for the mode which minimizes the expected time it takes to reach the target population $\vec{x}_t$ and this choice is independent of the past states or how much time has been spent in the present state.

Denote by T_{ij} the expected time it takes the system to attain the population $\vec{x}_j$ for the first time if it starts with population $\vec{x}_i$ and makes optimal choice for the mode at each subsequent state. Hence, T_{ij} is the optimal *first-passage time* from population $\vec{x}_i$ to $\vec{x}_j$. We consider the target population $\vec{x}_t$ to be an absorbing state, which is reasonable if the experiment halts as soon as the desired state is attained.

Now for T_{ij} to be optimal, it is easy to show that they must satisfy:

$$T_{it} = \min_{m \in \mathcal{M}} \left\{ \sum_{j \neq i,t} p_{ij}^{(m)} \cdot T_{jt} + \frac{1}{\lambda_i^{(m)}} \right\} \tag{22}$$

This equation reiterates the Markov property of the system, that is, the expected time to reach population $\vec{x}_t$ is the sum of expected time to reach the state via any of its neighbor (except $\vec{x}_t$ itself) and the expected time to exit the present state. $N-1$ such equations can be written for different $i \neq t$ for T_{it}.

This equation is a Bellman equation corresponding to our Markov Decision Process [22], and can be solved to obtain the expected times and the optimal modes for each population state. We used the Policy Iteration method to solve the equations in our case. The optimization is performed upon each aggregation or disaggregation event observed in the system, and every 10 seconds otherwise. Previous solutions are kept in memory, and used for initializing the subsequent iterations to speed up the optimization process.

VI. Results and Discussion

To demonstrate the effectiveness of our automatic model building framework and the relevance of our optimization algorithm, we performed four distinct experiments using the assembly E depicted in Fig. 2 as target structure and with different control algorithms: (I) mode m_0 only, (II) mode m_1 only, (III) randomized control, where the two modes alternate randomly with an average switching period of 15 s, and (IV) optimized control, in which the optimizer selects the most appropriate mode of agitation as a function of the current state of the system and of the current state of the model. The performance of the system is given by the time of the first occurrence of the assembly E, denoted as first-passage time, and bounded by the maximal duration of the run.

Each experiment consists of a series of 40 runs of 30 minutes each. Each run starts with all blocks being isolated and at random locations. In experiment IV, the optimization relies on an initial model based on the observations made

during two series (one per mode) of 10 runs of 5 minutes each. However, as explained earlier, the model is constantly enhanced, both qualitatively (e.g., if a new type of aggregate is discovered) and quantitatively (i.e., the reaction rates are adjusted) as the experiment progresses.

The underlying models are constructed based on a single interaction that is active between two blocks when they are both close to each other and appropriately aligned. As a result, several assemblies that are actually distinct from each other cannot be discriminated by the model, as seen by Fig. 2.

The choice of E as target structures was made because it can be univocally mapped to a unique species of the CRN, and it can be formed out of both C_1 and C_2. Indeed, the assembly D can also be univocally mapped to a unique species of the CRN, but cannot be formed out of C_2. As a result, the optimizer cannot effectively decide which mode of agitation should be applied when a trimer (i.e., C_1 or C_2) is present since these two assemblies are topologically undistinguishable. Note however that this is by no means an intrinsic limitation of our methodology, but rather a consequence of the simplicity of the underlying model.

First, as shown in Fig. 5, our results support the intuitive argument that self-assembly, as any self-organized process, requires a subtle interplay between "exploitation" and "exploration"—as expressed by the low-agitation m_0 and the high-agitation m_1, respectively. Indeed, both experiments I and II exhibit poor performance even as compared to the naive strategy that alternates between the two modes of agitation randomly. More importantly, our results show that one can drastically improve the performance of the system by optimizing the mode of agitation as a function of the system's state. Indeed, we observe a 40% and 66% decrease of the average and median first-passage time, respectively, under optimized control. When observing the system (Fig. 6), it looks like the strategy adopted by the optimizer is intuitive: the mode m_0 (low agitation) is active as long as assemblies that may lead to E (i.e., assemblies A, B, C_1, and C_2) are present, and switches to the mode m_1 (strong agitation) as soon as some incorrect tetramer is formed. However, the optimization also exhibits some interesting and less obvious behaviors. First, when only single blocks are present in the system, it sets the mode m_1 so as to favor mutual collisions. Upon the formation of a dimer B, the system may switch to mode m_0 in order to preserve it; however, while most reactions have clearly different rates for m_0 and m_1 (typically, one order of magnitude or more), the reaction $\mathsf{A}+\mathsf{B} \rightarrow \mathsf{C}_x$ exhibits relatively similar rates in either mode, thereby allowing for a dynamic switching between two behaviors, as a function of the time spent in each. For instance, the optimizer may select mode m_0 in order to conserve the formed dimer, but as the experiment progresses, the reaction rate of trimer creation in mode m_0 decreases, until it becomes smaller than the rate associated to mode m_1, thereby leading to the selection of the latter. This type of adaptive behavior is a built-in feature of our automated modeling approach, which is usually obtained using ad-hoc learning strategies (e.g., reinforcement learning) elsewhere.

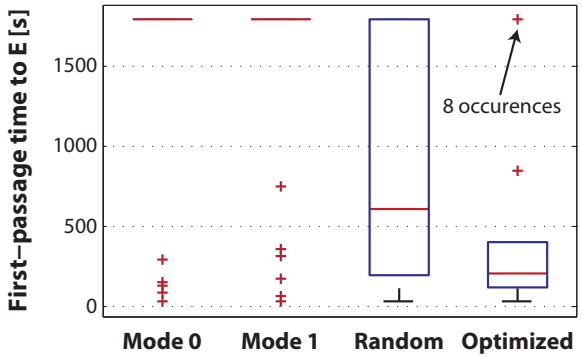


Fig. 5. Box plot of the first-passage time to the target structure E obtained over 40 runs of 30 minutes each for experiments I to IV. On each box, the central mark is the median, the edges of the box are the 25th and 75th percentiles, the whiskers extend to the most extreme data points not considered outliers, and outliers are plotted individually. Both experiments I (mode 0 only) and II (mode 1 only) exhibit a poor performance due to the unfavorable exploration *vs* exploitation balance when using a unique mode of agitation. The mean/median first-passage time of the optimized experiment (IV) is 524/205 seconds versus 930/612 seconds for the randomized experiment (III). A Mann-Whitney test rejects the null hypothesis that these two distributions of first-passage times are from the same distribution with equal medians with a p-value of $5.8 \cdot 10^{-3}$.

VII. Conclusion and Future Work

In this paper, we introduced the M^3 framework, a generic computational framework for the automatic, real-time modeling and control of stochastic and reactive multi-robot systems. We briefly summarized the theoretical foundations of the framework, and we demonstrated its relevance by deploying it for modeling and controlling the stochastic self-assembly of 3-cm-sized passive water-floating blocks. We described how the resulting models can be used to optimize a bang-bang controller, and our results show a significant improvement of the performance of the system with respect to strategies based on single modes of agitation or a random switching between the modes.

In future, we plan to investigate the use of more complex models (by adopting a 8-neighbors topology, for instance), and how they may enhance the overall performance of the system. Also, we aim at demonstrating the generality and the scalability of our approach by applying it to larger ensembles and other platforms. The strict requirement of perfect observability is currently an obstacle to the applicability of the M^3 framework in some circumstances (e.g., microscale self-assembly), but future theoretical developments and the use of more advanced machine learning methods shall allow for relaxing this requirement [9].

VIII. Acknowledgements

The authors would like to acknowledge Emmanuel Droz and Maria Boberg for technical support as well as Sven Gowal and José Nuno Pereira for useful discussion. Furthermore, we would like to acknowledge the Nano-Tera.ch research initiative, which partly sponsored this research in the context of the SelfSys project.

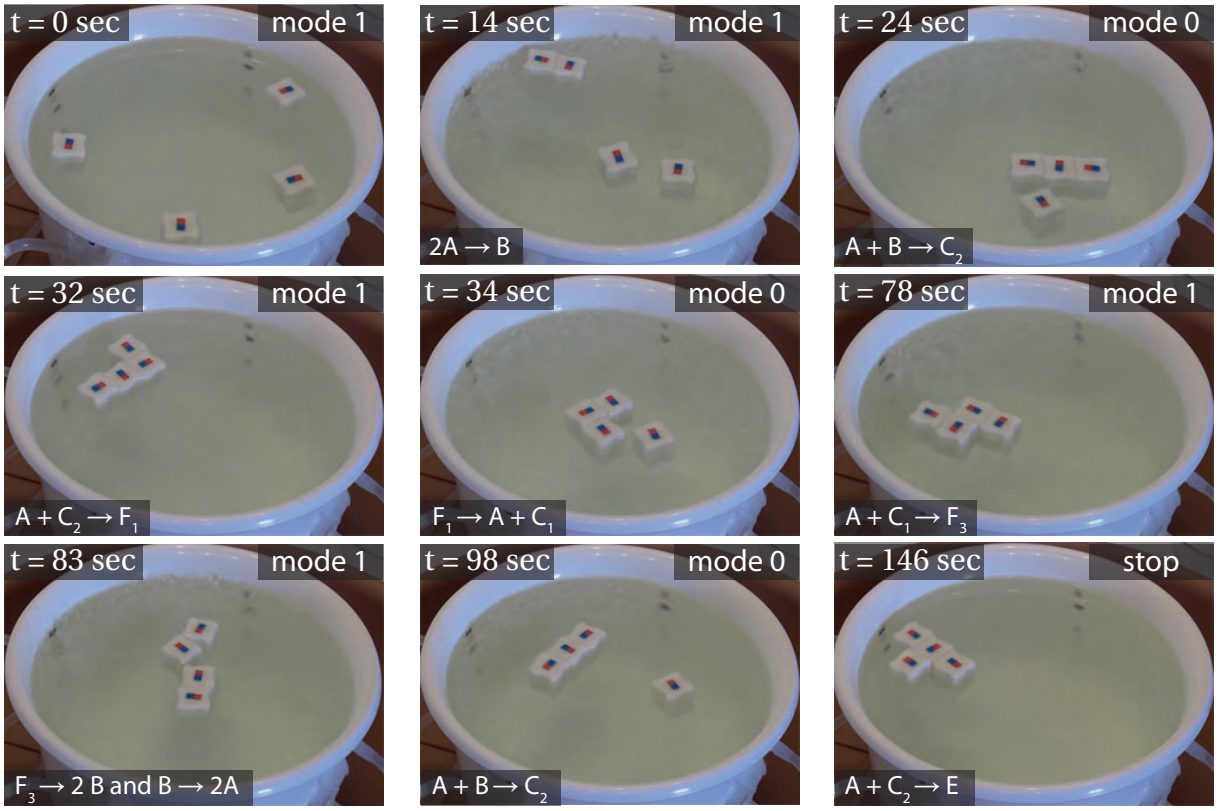


Fig. 6. Assembly sequence during a run of experiment IV (optimized control, see Section VI). The snapshots show the state of the system immediately after a reaction event. The reaction that fired is shown in the bottom left corner and the current time in the top left corner. The mode of agitation chosen by the controller is shown in the top right corner.

REFERENCES

[1] M. Mastrangeli, G. Mermoud, and A. Martinoli, "Modeling Self-Assembly Across Scales: The Unifying Perspective of Smart Minimal Particles," *Micromachines*, vol. 2, no. 2, pp. 82–115, 2011.

[2] R. Gross and M. Dorigo, "Self-assembly at the macroscopic scale," *Proc IEEE*, vol. 96, no. 9, pp. 1490–1508, 2008.

[3] T. G. Leong, A. M. Zarafshar, and D. H. Gracias, "Three-Dimensional Fabrication at Small Size Scales," *Small*, vol. 6, no. 7, pp. 792–806, 2010.

[4] D. L. Milutinovic and P. U. Lima, *Cells and robots*, ser. Modeling and control of large-size agent populations. Springer Verlag, Sep. 2007.

[5] A. Martinoli, K. Easton, and W. Agassounon, "Modeling swarm robotic systems: A case study in collaborative distributed manipulation," *Int J Robot Res*, vol. 23, no. 4-5, pp. 415–436, Jan. 2004.

[6] F. Schweitzer, *Brownian Agents and Active Particles: Collective Dynamics in the Natural and Social Sciences*, ser. Springer Series in Synergetics. Springer, Oct. 2003, vol. XVI.

[7] M. D. Schmidt and H. Lipson, "Distilling Free-Form Natural Laws from Experimental Data," *Science*, vol. 324, no. 5923, pp. 81–85, Jan. 2009.

[8] M. D. Schmidt, R. R. Vallabhajosyula, J. W. Jenkins, J. E. Hood, A. S. Soni, J. P. Wikswo, and H. Lipson, "Automated refinement and inference of analytical models for metabolic networks," in *Phys Biol*. Cornell Univ, Cornell Computat Syst Lab, Ithaca, NY 14853 USA, 2011, pp. 1–20.

[9] M. D. Schmidt and H. Lipson, "Automated modeling of stochastic reactions with large measurement time-gaps," in *GECCO '11: Proceedings of the 13th Annual Conference on Genetic and Evolutionary Computation*, Jul. 2011, pp. 307–314.

[10] M. T. Tolley and H. Lipson, "On-line assembly planning for stochastically reconfigurable systems," *Int J Robot Res*, vol. 30, no. 13, pp. 1566–1584, 2011.

[11] L. Matthey, S. Berman, and V. Kumar, "Stochastic strategies for a swarm robotic assembly system," in *2009 IEEE International Conference on Robotics and Automation (ICRA)*, 2009, pp. 1953–1958.

[12] E. Klavins, "Programmable Self-Assembly," *Control Systems Magazine, IEEE*, vol. 27, no. 4, pp. 43–56, 2007.

[13] N. Napp, S. Burden, and E. Klavins, "Setpoint regulation for stochastically interacting robots," *Auton Robot*, vol. 30, no. 1, pp. 57–71, 2011.

[14] E. Di Mario, G. Mermoud, M. Mastrangeli, and A. Martinoli, "A trajectory-based calibration method for stochastic motion models," in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2011, pp. 4341–4347.

[15] T. Lochmatter, P. Roduit, C. Cianci, N. Correll, J. Jacot, and A. Martinoli, "SwisTrack - a flexible open source tracking software for multi-agent systems," in *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2008, pp. 4004–4010.

[16] N. Napp, D. Thorsley, and E. Klavins, "Hidden Markov Models for non-well-mixed reaction networks," in *American Control Conference, 2009. ACC '09*, 2009, pp. 737–744.

[17] T. A. Henzinger, "The theory of hybrid automata," in *Proceedings of the Eleventh Annual IEEE Symposium on Logic in Computer Science (LICS'96)*, Electrical Engineering and Computer Sciences University of California at Berkeley. IEEE, 1996, pp. 278–292.

[18] N. Correll and A. Martinoli, "Modeling and designing self-organized aggregation in a swarm of miniature robots," *Int J Robot Res*, vol. 30, no. 5, pp. 615–626, 2011.

[19] G. Mermoud, J. Brugger, and A. Martinoli, "Towards multi-level modeling of self-assembling intelligent micro-systems," in *AAMAS '09: Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems*. International Foundation for Autonomous Agents and Multiagent Systems, May 2009, pp. 89–96.

[20] M. Cook, D. Soloveichik, E. Winfree, and J. Bruck, "Programmability of Chemical Reaction Networks," in *Algorithmic Bioprocesses*, A. Condon, D. Harel, J. N. Kok, A. Salomaa, and E. Winfree, Eds. Springer Berlin Heidelberg, 2009, pp. 543–584.

[21] D. T. Gillespie, "Stochastic simulation of chemical kinetics," *Annu Rev Phys Chem*, vol. 58, pp. 35–55, 2007.

[22] A. Ronald, "Dynamic programming and Markov processes," MIT Press, 1960.