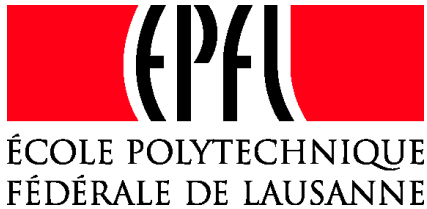
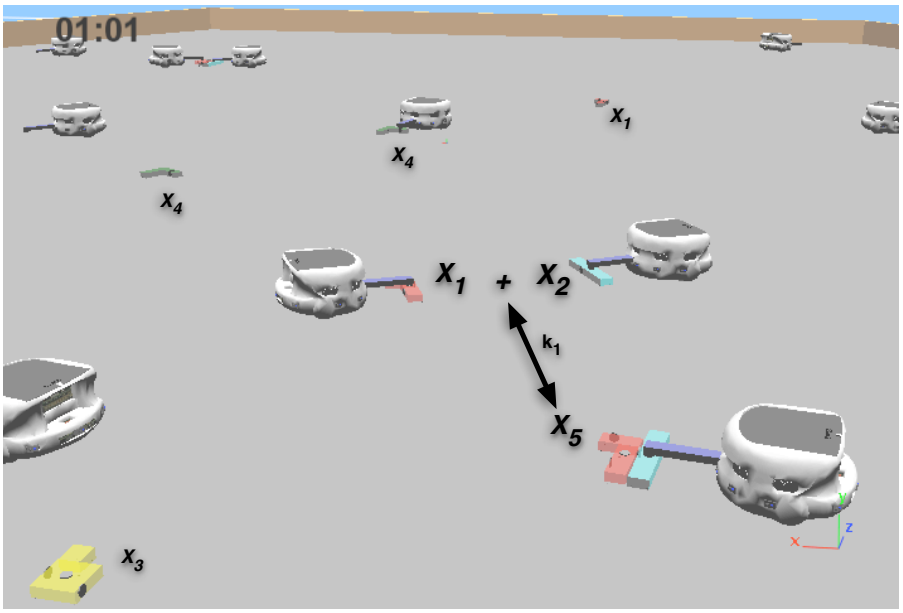


SS 2008, IC-SIN, No. KUMAR-SWIS-MP12

Start: 19.02.2008
Finish: 15.08.2008

Hybrid Reactions Modeling for Top-down Design Framework

Loïc Matthey



Professor:	Alcherio Martinoli, Vijay Kumar
Assistant:	Gregory Mermoud, Spring Berman

This page intentionally left blank.

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

SWIS

SS 2007-2008, No. KUMAR-SWIS-MP12	Start:	19.02.2008
	Finish:	25.08.2008

Title of the project

Hybrid Reaction Modeling of the Extended Self-Assembly Problem

Author **Loïc Matthey**

Project description

We propose a new modeling framework inspired by chemical reaction processes. Our approach consists in defining the processes and the interactions within the system in term of reactions. Such a definition can be applied to many systems, ranging from biochemical systems to swarm robotics. In particular, we aim at exploiting the toolbox developed in the context of hybrid system modeling and simulation.

The concept of extended self-assembly is the following: given a set of passive building blocks A, B, C, and D, how to obtain, with a maximal yield, the products X, Y, and Z using a set of N active transporters? What is the smallest set of reactions leading to these products? More importantly, how shall we design the building blocks and their transporters in order to fit this set of reactions? The reaction set may also involve intermediate products and be influenced by external factors. We draw inspiration from the DNA translation, and more specifically the tRNA transport molecules, which bring protein building blocks to the ribosomes. Our research may have impact on the understanding of such biological processes occuring at the nanoscale.

We envision a modeling framework that is sufficiently general to accommodate with the extended self-assembly problem, as well as the classical self-assembly problem. The validation of our models will be achieved using realistic simulation (Webots), or numerical simulations (Matlab) of various robotics systems. First, we will implement a simple example of assembly using robots as active transporters.

Tasks

- Do a literature review of actual reaction rate modeling and hybrid system simulations.
- Formalize the augmented self-assembly problem.
- Propose a modeling framework for the augmented self-assembly problem.
- Solve the smallest reaction set problem and the transporter behavior problem.
- Choose a test case study and model it using the framework.
- Simulate the test case using Matlab and/or Webots.
- Compare the results of the model and the simulation.
- Depending on the results, modify and propose improvements to better fit the data. Iterate on the last steps.

Oral presentations	The date and time of the intermediate and final presentations will be specified mutually by the student, supervisor, and professor. Presentations and accompanying slides must be in English and in MS Powerpoint format. Rehearsal presentations with the project supervisor before the official talk are strongly encouraged.
Final report	All of the student's work shall be submitted on a CD (report, presentation, source code, documentation, media, etc.) as well as a hard copy of the report (English, double-sided, unbound). One copy will be required for the SWIS library (delivered to Corinne Farquharson, administrative assistant of SWIS, BC232), and additional copies may be requested by the project supervisor for his/her records. The final report must use the standard cover page and include a copy of this extended proposal just after the cover page. The report must be submitted in PDF format, and the source files (Latex recommended, MS Word accepted) should be contained in the CD-ROM, as well as the final presentation. A complete draft of the report must be submitted to the responsible supervisor at least *two weeks* before the final deadline. Revisions and comments will be returned in a timely fashion and will need to be incorporated into the final version of the report.
Web visibility	The student's name will be listed on the SWIS web site (under people/undergraduate) and hyperlinked to an appropriate home page if possible (either personal, or a brief page on people.epfl.ch). At the end of the project, with the help of the supervisor, a 1-page summary, a definitive project abstract, and one picture of the project will be posted (see http://swis.epfl.ch/teaching/student_projects/ for examples). It should be finalized and approved by the supervisor. Additional movies and pictures can be posted according to SWIS guidelines and after supervisor approval. This page will remain on the SWIS site under the section "past student projects" and the student's name will be moved to the "alumni" section. The project will not be considered concluded until the dedicated web page is available on-line.
Supervision	Spring Berman (UPenn), Vijay Kumar (UPenn), Grégory Mermoud
Place of work	University of Pennsylvania, USA
Recommended literature	DT Gillespie. Stochastic simulation of chemical kinetics. Annual Review of Physical Chemistry, 58:35–55, 2007 Eric Klavins. Programmable self-assembly. IEEE Control Systems Mag., 27:43–56, 2007 GM Whitesides and B Grzybowski. Self-assembly at all scales. Science, 295(5564):2418–2421, 2002
Signature of the Professor	
	Prof. Alcherio Martinoli, Swarm-Intelligent Systems Group

Abstract

This report presents the work accomplished by Loïc Matthey during his Master project from Ecole Polytechnique Federale de Lausanne (EPFL). This project was a joint work between the Distributed Intelligent System and Algorithms (DISAL) Laboratory at EPFL, Switzerland and the General Robotics, Automation, Sensing and Perception (GRASP) Laboratory at University of Pennsylvania, United States of America. It took place in the Master spring-summer semester 2008.

We present a theoretical framework to design Top-down control scheme for arbitrary systems. Being able to control a complex system using high-level instructions only is a promising and attractive paradigm. Our approach is based on the use of a Chemical Reaction Network model, used as a proxy to derive the control schemes. To test the application of our method, we consider the Top-down control of a realistic multi-robots assembly platform, simulated using a 3D physics simulator, Webots.

First we present the modeling of the robotic platform using a Chemical Reaction Network. The free parameters are precisely fitted. We simulate the system using an ODE approximation and an exact stochastic simulation. We find that the model can be made to fit quantitatively to the experimental data, especially when using a stochastic simulation approach.

Second we define an optimization and control scheme for a class of Chemical Reaction Networks. We prove convergence results and write the optimization problem as a linear program of the time of convergence of the system under constraints on the equilibrium value. It allows us to design sets of reaction rates producing a specified converged behavior, in polynomial time. This optimization provides precise controls of the system using only high-level goals.

Finally, we map the optimized model down onto the realistic physical assembly platform. We find that the system can be controlled using the optimized parameters of the model level, but that small discrepancies can have disruptive effects.

This page intentionally left blank.

List of Figures

List of Tables

Contents

Chapter 1 Introduction

1.1 Problem overview

Self-assembly is everywhere.

At every scale, systems interact, collaborate and combine to create new bigger scale systems. Crystals are formed by nanoscale assembly of carbon atoms, cell membranes by the arrangement of fatty acids into a lipid bilayer and human beings by the organization and cooperation of their trillions of living cells.

Yet this process, being maybe so general and vast, is still tremendously unknown.

The study of self-organizing systems gives insight into the organization patterns of their parts, and could help understanding and then modifying them.

The recent field of Swarm intelligence applies the self-organizing principle to many systems and applications, ranging from algorithmic procedures (routing of packets, meta heuristics) to team of multiple robots. This approach makes sense when the number of robots increases to the point where a centralized or classical control methodology is not tractable anymore. Interestingly, a similar problem occurs when the scale of robots and components starts to shrink down dramatically. If the environment is intrinsically random and unknown, the robustness factor promoted by self-organizing systems becomes a key factor.

Our interest goes towards that direction. We want to study systems whose dimension is shrinking to the level where classical approaches are not applicable anymore. Furthermore, we want to model those systems, and create a framework providing a complete control flow to modify the behavior of those systems.

This might seem fairly trivial, but when the system under consideration is hard to study by definition and not well-known, even the simplest control over them or insight in their behavior becomes an appreciable achievement.

Our approach is the following:

- We propose an abstract way of describing the problem under study and the actions needed to achieve its control. Our main claim is that it is possible to divide the problem into two parts: an intrinsic system, on which we have no control, and an augmented system, which encompass our additions and modifications made to modify the behavior.

- We propose to use a Chemical Reaction Network mathematical framework through all this process to model the system under study. This framework will proves itself useful for its flexibility and expressive power at the scale we are studying.
- We present a way to control the system via a Top-down design approach, first working on the model and then mapping it back onto the studied system. Top-down design peaks the interest nowadays, as being able to control a complex system using high-level instructions only is a promising characteristic.
- Everything is presented and verified by referring to a specific system that we create and study: a robotic platform performing a self-assembly of products.

We call it the **Hybrid Reactions Modeling for Top-down Design Framework** (HyRToD). "Hybrid" because we will use both ordinary differential equations approximations and stochastic simulations to simulate the model, depending on the context.

The robotic platform is actually simulated on a computer, by using a realistic 3D physics simulator named Webots [1]. Webots is based on ODE, an open source physics engine for simulating 3D rigid body dynamics [2]. Such a simulator allows us to performs systematic experiments faster than real-time and with null fabrication costs.

This might seems strange to apply a framework we claim to be thought for micrometer scale dynamics onto a high level robotic platform. We actually design the robotic platform to give it characteristics usually shown at a smaller scale, and therefore only take advantage of the robotic platform as a model system easy to measure and modify. This work focuses on this robotic platform as a first test for our framework. Further works will consider smaller scale applications to assess our initial assumption on our framework.

1.1.1 Relations to biological processes

Even though we apply our method to a robotic implementation, a fairly high scale system by all means, we claim that this method is applicable to many different systems, especially the ones governed by random dynamics.

We chose to create a robotic platform performing a self-assembly task on purpose. Having robots carrying the building blocks and assembling them can be thought of as an idealization of the self-assembly process taking place into the cell, for example the protein synthesis. If we allow the building blocks to move around and assemble on their own, the added robots will behave like enzymes, promoting some reactions.

Moreover, our method, using a Chemical Reaction Network model, is very easy to apply on biological processes. This model has been extensively used in the study of biological systems, and is very well understood by the community working in this field. This is an added factor to the development of further interdisciplinary cooperations between engineering and life sciences.

1.2 Outline

This report is organized as follows: Chapter 2 defines precisely our goals and the abstract problem definition and control flow we aim to study. Chapter 3 goes over the theoretical

notions used in our work, and gives pointers to the available literature on the subject. Chapter 4 presents extensively the specific system we are studying, namely the physical robotic simulation of an assembly task. Chapter 5 introduces the representation of our specific system into a Chemical Reaction Network notation, presents how we fitted the free parameters and compare the simulated results with the physical measurements. Chapter 6 is dedicated to the optimization step applied on our mathematical model in order to control its behavior. Chapter 7 presents the Top-down mapping of the modified model towards the physical system. Chapter 8 concludes the work and assess its validity and shortcomings.

This page intentionally left blank.

Chapter 2 Project description

2.1 Problem definition

We phrase the problem to solve as follows:

> Consider an intrinsic complex system with observable dynamics and a measurable performance metric. Let this intrinsic system attains a performance metric value X. Introduce agents into the system with designed specific behaviors, getting an augmented system. Can we design such behaviors so that the performance metric of the augmented system attains an new value Y, corresponding to a better or specific behavior?

We will refer at that question as the **Intrinsic System Augmentation Problem** (ISAP). We believe that this formulation accurately describe an engineering methodology for different applications. Moreover, we argue that it is easy to represent different problems with that framework.

The problem is decomposed in its most abstract formulation in Section 2.2. But for this project and thus the rest of this report, we only look at a specific instance of it. Referring to the vocabulary and decomposition of Figure 2.1, we have:

1. An intrinsic complex system representing an assembly task of a puzzle (Section 4.1). This assembly task is either a self-assembly process or an assembly process depending on the components we put in and how we look at it. This intrinsic complex system is created on a simulated robotic platform (Section 4.3).

2. A mathematical model based on a Chemical Reaction Networks formulation (Chapter 5). We chose this formulation for its versatility and power, and because it is a well studied model with efficient simulations and theoretical insights.

3. An control/optimization of the model using a Convergence time optimization scheme, following the work done on Rapid Mixing Markov Chains for redistribution of a swarm of robots on multiple sites [3]. This performs a continuous optimization of our Chemical Reaction Network, namely its reaction rates. We do not address directly in this work the discrete optimal design of this Chemical Reaction Network. See Chapter 6.

4. An augmented system presented in two different ways: either modifying the behavior of the available agents or introducing new agents with designed behaviors. These are

two aspects of the same augmentation process, which have different applicability fields and intrinsic difficulties. See Chapter 7.

2.2 Decomposition

Starting from the definition of the ISAP, we derive a very abstract decomposition into smaller scale components. See Figure 2.1 for the general decomposition of the problem. See Section 2.2.1 for a precise definition of each elements.

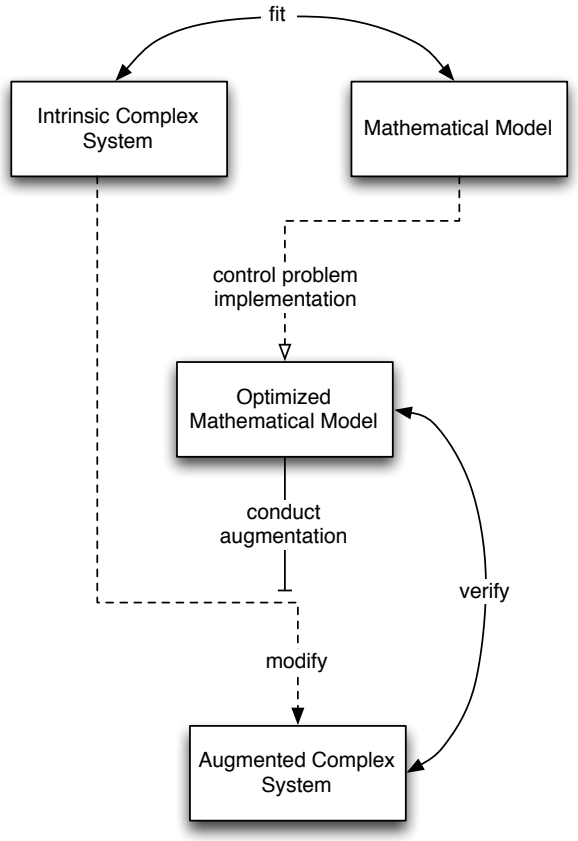


Figure 2.1: Intrinsic System Augmentation Problem decomposition, top-level components only.

Here is the rationale behind this decomposition:

1. We have an Intrinsic Complex System that we are able to measure in some way. We will actually present a different formulation of this Intrinsic Complex System, for cases where the real system is not easily measurable, the Compliant Platform. We need to have some insight on this Intrinsic Complex System, because we want to model it mathematically.

2. We construct and fit a Mathematical Model of this Intrinsic Complex System. We can use this Mathematical Model to predict the Intrinsic Complex System, and will do several iterations to get the best model possible. Different modeling approaches can be taken, as well as simulations strategies for each of them.

3. We take this Mathematical Model and express it as a control problem to be solved. The goal can be to optimize the model for a given metric, or to change its behavior towards a specific one. This can take a lot of forms, depending on the modeling framework used and the level of plasticity available in the model and initial system. This new model can also be simulated, to verify its behavior.

4. This Augmented Mathematical Model is used to direct the augmentation of the Intrinsic Complex System into an Augmented Complex System. By "augmented", we mean modifying the system global behavior using one of some of the following ideas: adding new components, modifying behaviors, modifying components. This is a Top-down approach to complex system control. Once we know how to augment the intrinsic system, we have to verify that it indeed behaves like the optimized model. Hence we perform several iterations of the augmentation, so that the optimized mathematical model actually captures the new Augmented Complex System.

5. We can then study the Augmented Complex System, to see what was changed for it to behave accordingly to our goals. This could give insights for processes that are hard to study, especially when taking the Compliant Platform approach.

2.2.1 Project components

Intrinsic Complex System

See Figure 2.2 for the diagram. This component represent the actual system we want to study and modify.

There are two possibilities for this component:

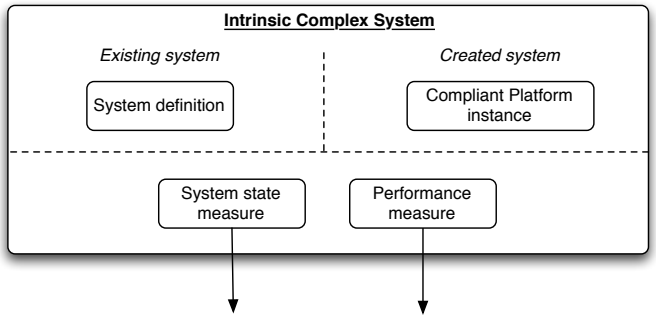


Figure 2.2: Intrinsic Complex System component. Black arrows show inter-component communication, with other top-level components. Compliant Platform Instance in bold is defined in another diagram.

Existing system: In that case, we have a complex system already existing. Such applications could be existing platforms for self-assembly, or an existing natural process. We need to be able to measure the state of this system in some way, as well as assessing its performance according to a desired metric. These informations are then used by the Mathematical Model component or to assess the performance of the system.

Created system: If we do not have a complex system to observe, or if the actual complex system is not measurable, we can bypass that by **creating** an intrinsic complex system. For that we introduce a **Compliant Platform instance**. A Compliant Platform is a real or simulated platform allowing a big variety of problems reproduction. The aim is to propose a set of agents that can reproduce any given problem compatible with their hardware capabilities. Moreover, it is possible to ensure certain properties, for example a well-mixed property.

In this work, we use a Created system approach to reproduce a self-assembly task using a macro-scale robotic platform. The system is created using a physical simulator, Webots. More on that is presented in Chapter 4.

Mathematical model

The Mathematical model component aims at reproducing as well as possible the Intrinsic Complex System, while being quicker to simulate. See Figure 2.3 for the diagram of this component.

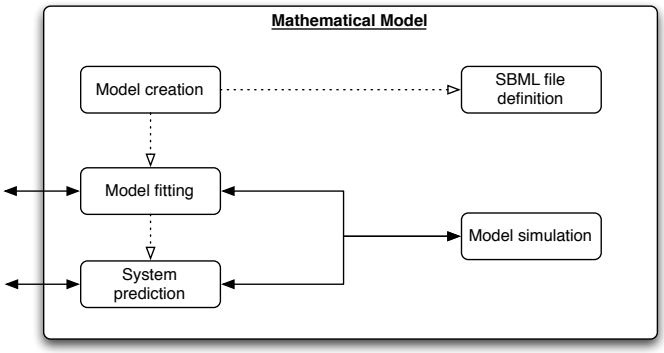


Figure 2.3: Mathematical Model component. Black arrows show inter-component communication. Dotted arrows show termination dependencies between components.

- The first step is to create the model. This creation consists on the choice of a modeling notation and depends on the knowledge we have about the system. As precised earlier, we are working with Chemical Reaction Networks, so our models will be done in this framework. A convenient and standardized format for such networks exists: System Biology Markup Language (**SBML**) [4]. This is a XML-based file format designed to store systems of chemical reactions. This is the closest to an accepted standard we found to write our mathematical models.

- The model has to be fitted in some way to the Intrinsic Complex System. If we are using an existing complex system, then this is a quite complex problem, especially if we do not have precise insight in the behavior of the system. We can use methods like Bayesian Inference or MCMC (Markov Chain Monte Carlo) to fit the model on the experimental data. If we are using the Compliant Platform, than we assume that we can measure much more precisely the processes taken place, and this model fitting is more straightforward.
- We also need a simulation framework for the mathematical model. For Chemical Reaction Networks, a lot of literature is available on that. As we will present in Section 3.2.2, we use either a direct Stochastic simulation or a simple ordinary differential equation solver.

Optimized Mathematical model

We use the mathematical model of our system as a thinking abstraction and an optimization medium. The model is easier to manipulate and adapted to common optimization and design techniques.

Moreover, we will use optimizations scheme that work "blindly", that is which have no insight into the system concepts. We think it makes the optimization more fair. Humans tend to make bad assumptions or look only for particular patterns when trying to optimize a system, we think enforcing the "blindness" in the algorithm could prevent that.

Our models are multi-affine systems of equations, which are not trivial systems to analyze and optimize. We will address this issue and how we tackled it in Chapter 6.

See Figure 2.4 for this component's diagram.

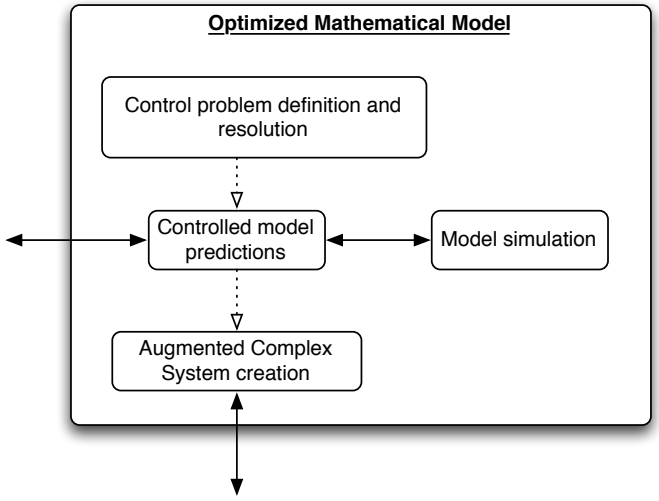


Figure 2.4: Augmented Mathematical model component. Black arrows show inter-component communication. Dotted arrows show termination dependencies between components.

Augmented Complex System

We then need to map the new mathematical model onto the complex system, a step we call system augmentation. This is a Top-down optimization approach, which we think is more appropriate for the kind of complex systems we are handling. This makes even more sense if we do not know a-priori how to change the behavior of the intrinsic complex system, because of its complexity. Working on the model gives another level of abstraction that helps to understand the acting processes of the complex system.

Identifying what has to be changed to produce the behavior of the modified model is a great challenge. As we will see in Section 6.3, our test case is actually very easy to modify.

See Figure 2.5 for the component's diagram.

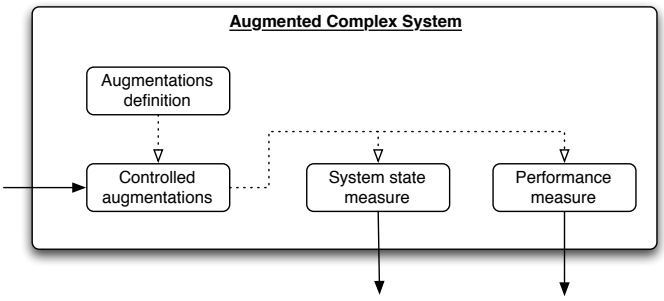


Figure 2.5: Augmented Complex System model component. Black arrows show inter-component communication. Dotted arrows show termination dependencies between components.

2.3 Examples

We quickly present some examples of several systems into our Intrinsic System Augmentation Problem framework.

2.3.1 Nanoscale self-assembly

The intrinsic system consists of the possible interactions and bonds. The augmented system can be abstracted as any modification applied to the system, that modify the intrinsic behavior. For example, changing the pH of the solution so as to activate different sticking surfaces is an action of the augmented system.

2.3.2 LEURRE project

LEURRE is a project on building and controlling mixed societies composed of animals and artificial agents [5]. A small robot capable of infiltrating a cockroach group was developed. The cockroach group is put in a arena with several shelters of specific luminosity. Cockroaches decide on a shelter according to the luminosity and the number of cockroaches under it. This

is a self-organized decision process. The robot were able to infiltrate this group and to direct the global decision of the group. The infiltrated robots made the cockroaches go under a light shelter, a configuration which was never attained with the cockroaches group only.

Intrinsic system: the cockroach group. The metric is the probability of the different shelters as final decision.

Augmented system: the cockroaches and the robots. The robots choose a different shelter, this action in turn modify the final probability of the shelters.

2.3.3 Enzymes

In living cells, chemical reactions take place whenever compounds need to be transformed or created. For chemical processes needing energy to occur, one usually see an *energy barrier* mechanism, see Figure 2.6. Before a reaction can occur, activation energy has to be provided, to pass the barrier. When adding enzymes to the system, they catalyze the reaction and produce a virtual decrease of the needed activation energy. Enzyme can act by improving fitting of compounds, stabilizing transitions or modifying orientations.

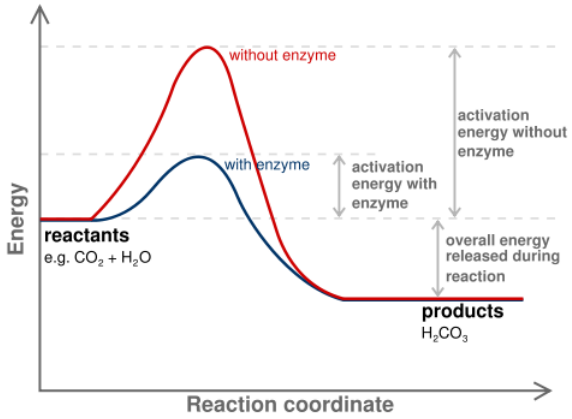


Figure 2.6: Energy barrier and catalyst effect of enzymes.

Intrinsic system: The original chemical reaction, with specific activation energy and rates.

Augmented system: The catalyzed chemical reaction with the introduction of the enzyme. The enzyme performs an action (binding with change of conformation) that reduces the activation energy and increase the rate of the chemical reaction.

2.3.4 RNA translation into proteins

In living cells, DNA contains the blueprints for every functional proteins. In the cell nucleus, it is first transcripted into RNA, which is translocated to the cytoplasm to be translated

into proteins. The RNA strands contains the building plan, and special proteins, called ribosomes, attach on it in order to "read" it and assemble amino acids (the basic building blocks of proteins) according to this plan. These amino acids are created as a linear chain (*first* structure), that fold onto itself according to low-energy bounds, acquiring a tridimensional structure called a *conformation* (*second* and *tertiary* structure). A protein is a sequence of amino acids in a specific conformation that allows its functional activity.

Intrinsic system: Ribosomes assemble amino acids according to the RNA code. The obtained first structure protein then folds itself into a specific conformation.

Augmented system: Chaperone protein helps the folding of the protein, possibly modifying the obtained conformation or allowing the initial one under different environment conditions (heat-shock response).

Chapter 3 Field overview

3.1 Self-assembly engineering

This work has been triggered by an interest in the simulation and modeling of self-assembling processes. Such process can take many forms, from nano-scale assembly [6, 7, 8] to control of biomolecules [9, 10, 11, 12] up to modular robotics [13, 14]. This field is gaining more and more attention nowadays [15].

3.1.1 Microscale assembly

Of all these applications, microscale assembly is the one which gathered the most interest in the last few years and which promises the most interesting future applications [15, 16].

While pursuing the race towards even more miniaturization, we are facing new problems that current technologies and methodologies have trouble solving. The lithography process, used to create all the microchip used now, is getting to its limit [15]. New approaches become necessary.

The current technology for microscale assembly is still in its infancy [15]. The current state of research aims at attaching pieces together at specific positions. This either creates bigger scale components, or combines functional devices created via traditional methods. Several methods are currently under study [17, 18, 19, 6], ranging from attaching mechanisms to prototyping methods.

However, such mechanisms are still far from the kind of control we have on the higher scale assembly, and all those processes have a very low production yield. But microscale assembly opens the door to a whole new world of possibilities for integration, system repairs and even active drugs.

An interesting distinction for self-assembly, made by Whitesides [15], is the difference between *static* and *dynamic* self-assembly. In static self-assembly, the components once formed stay stable and stop dissipating energy. In dynamic self-assembly, energy is dissipated and should be produced or given in some way. A living cell is a typical example of dynamic self-assembly.

Our works aims at studying such dynamical self-assembly, yet at a scale closer to biology (millimeter scale) than microscale scale.

3.1.2 Modular robotics

As we aim at using robots as a platform, our work is similar to studies done in modular robotics.

Modular robotics encompass any robotic system that can deliberately change its own shape, in order to adapt to new circumstances, perform new tasks or recover from damage [13][20].

A work close to our approach is the one done by E. Klavins on programable self-assembly [14, 21, 22, 23]. It revolves around the assembly of triangular robots, moving around randomly on an air table and capable of assembling themselves according to a given plan.

The plan itself is constructed with a grammar approach, working with *graph grammars*. A graph grammar is a set of rules transforming a graph when applied on it. The assembly is represented as a sequence of application of rules, transforming the initial set of products into a final graph representing the final assembly. Klavins showed methods to construct graph grammars automatically for a given final assembly [23].

These grammars are then used with the robots to converge to a final shape constructed only by self-assembly.

In the first versions of this approach, the particularities of the assembling process, such as geometric difficulties and disassemblies due to shocks, were not taken into account. Klavins accounted for them by measuring the kinetic rate constants of assemblies, and then trying to modify the plan accordingly [14].

Our approach on the other hand, directly takes into account those reaction rates, making them central and essential to our approach. We think that finding an "optimal" theoretical plan is useless when this plan could become "sub-optimal" under the constraints of the reactions rates. These rates directly show the physical characteristics of the system to assemble, they are not easily modified.

This is also why we will use an approach using Chemical Reaction Networks for our plans and models: they are build to take into account the intrinsic reaction rates of the systems.

Furthermore, we study a system of heterogenous parts, adding a specificity and complexity requiring different analysis and techniques.

3.2 Chemical reaction networks

3.2.1 Theory

Through this project, we use a Chemical Reaction Networks notation and framework as mathematical model. This has been introduced in the context of chemical processes in 1979 [24] and has been very researched since then.

This level of representation is at the same time very general, offering representation of very different processes, and also quite precise and detailed, allowing to construct full dynamic simulations of the system behavior on a computer. This introduction to chemical reactions is adapted from the textbook of J.Wilkinson [25].

A general chemical reaction takes the form:

$$m_1 R_1 + m_2 R_2 + \ldots + m_r R_r \rightarrow n_1 P_1 + n_2 P_2 + \ldots + n_p P_p \tag{3.1}$$

Where r is the number of reactants and p the number of products. R_i is the ith reactant molecule and P_j the jth product molecule. m_i is the number of molecules of R_i consumed in a single reaction step, and n_j the number of molecules of P_j produced. The coefficients m_i and n_j are known as *stoichiometries*.

A chemical reaction networks consists of several of these reactions, possibly sharing reactants and/or products. If a reaction can occur in both directions, meaning that the products in the right part can be transformed in the reactants of the left part with the same stoichiometries, we call this reaction *reversible*. A reversible reaction is written as follows (for a simple dimerisation example), see Eq. (3.2).

$$2P \rightleftharpoons P_2 \tag{3.2}$$

Such networks represent the possible *actions* of the systems and the relations between the elements. But it does not represent the dynamics directly. To add this information, we have to make an assumption on the type of dynamics governing the system.

In chemical system, a classical governing dynamic is a mass-action stochastic kinetics [26]. In this formulation, we associate to each reaction R_i a *stochastic rate constant*, c_i, and an associated *rate law* (or *propensity function*) $h_i(x, c_i)$, where $x = (x_1, x_2, \ldots, x_u)$ is the current state of the system. The form of $h_i(x, c_i)$ (and the interpretation of the rate constant c_i), is determined by the order of reaction R_i. In every cases, the propensity function has the same interpretation: conditional on the state being x at time t, we then have that the probability that an R_i reaction will occur in the time interval $(t, t + dt]$ is given by $h_i(x, c_i)dt$ [25].

The classical orders of reactions and their propensity functions are as follows:

Zeroth-order $R_i : \emptyset \xrightarrow{c_i} X$

This represents a constant rate of production of a chemical specie.
$h_i(x, c_i) = c_i$.

First-order $R_i : X_j \xrightarrow{c_i} ?$

This is the spontaneous transformation of a reactant into new products.
$h_i(x, c_i) = c_i x_j$, as there are x_j molecules of X_j.

Second-order $R_i : X_j + X_k \xrightarrow{c_i} ?$

This represents a reaction between a pair of reactants.
$h_i(x, c_i) = c_i x_j x_k$, for all combined pairs of molecules X_j, X_k. Special case if $X_j = X_k$: $h_i(x, c_i) = c_i \frac{x_j(x_j-1)}{2}$.

Higher orders Those can be transformed back into second-order reactions, as we make the assumption that a third-order reaction is impossible and actually corresponds to the combined effect of two second-order reactions.

This allows then to simulate exactly the modeled process assuming we know all the rate constants and rate laws.

Nowadays, simulation of multiscale systems have become the new interest. A multiscale system is characterized either on the timescale aspect or the copy number of reactants [27].

1. For the timescale aspect, the different scales arise when some reactions are much faster than others. They then quickly reach a stable state, and the global dynamics of the system is driven by the slow reactions.

2. For the copy number of reactants, the difference comes from the relative size of the populations. Species with a small population are best viewed as discrete stochastic processes, while the large populations should follow a deterministic model.

Such systems, called *stiff* systems, present new problem to commonly used simulations algorithm. They also are of increasing interest in system biology, as a lot of real biological process operates on multiple scales.

3.2.2 Simulation algorithms

Several ways of simulating chemical reaction networks are available.

Ordinary differential equation

The simplest one, and the most used by chemists because of thermodynamical limits and number of molecules involved, is to use the associated ordinary differential equation (ODE). One can represent the populations (or concentration, given a finite volume V) of all products, and treat the reactions as outflow and inflow acting on those populations. If we take the simple dimerisation system (3.2), assuming a forward rate k^+ and a backward rate k^-, we obtain:

$$\dot{P} = -k^+ \frac{P(P-1)}{2} + k^- P_2 \tag{3.3}$$

$$\dot{P_2} = k^+ \frac{P(P-1)}{2} - k^- P_2 \tag{3.4}$$

Such a transformation is automatic for any chemical reaction network with reactions up to second-order. We can then simulate it using classical numerical integration methods. Note that ODE use continuous number for the populations. Therefore, this approximation can be wrong when the copy number of elements (the number of elements) is small. In classical chemical contexts, the copy numbers are very high (near Avogadro's number), so this is not an issue.

Gillespie Stochastic simulation algorithm

It has been shown by Gillespie [28, 29, 30, 31, 32], that it is possible to perform an exact simulation of a chemical reaction networks. The algorithm is referred to as the Direct Method or Gillespie Stochastic Simulation Algorithm (SSA). It takes advantage of the fact that the time-evolution of a reaction network can be regarded as a stochastic process, and, because the propensity functions depends only on the current state, the system is a continuous time Markov process with a discrete state space. The time to the next reaction follows a exponential distribution $Exp(h_0(x,c))$, with $h_0(x,c) = \sum_{i=1}^{v} h_i(x,c_i)$ and v reactions. The type of

reaction firing is independent of that time, and is given by the probability $h_i(x, c_i)/h_0(x, c)$. We can then simulate the system for each reaction events, up to a desired finishing time.

This algorithm, however, is highly inefficient when the number of products and reactants increases. Several optimization have thus been proposed to cope for that limitation.

Tau-leaping

Gillespie first proposed optimization, the tau-leaping optimization [33], aims at make the system evolve for a time τ where a certain amount of reactions fire instead of simulating every reaction. It is based on the assumption that the propensity functions $a_j(x)$, governing the rates of firing of the reactions, stays nearly constant for a certain time τ. It is then possible to approximate the number of reaction firings during that time τ by a Poisson process of rate $a_j(x)\tau$.

Automatic ways of finding τ also have been proposed [34], as well as different variations of the tau-leaping: Implicit tau-leaping (performs better for stiff systems) [35], Trapezoidal tau-leaping (more efficient than explicit tau-leaping), and the latest explicit-implicit tau-leaping (combination of the two regimes) [36].

The principle of simulating several reactions events at the same time is also used in another very known algorithm, called the Gibson & Bruck Next Reaction algorithm [37].

Multiscale systems

To simulate multiscale systems with different timescales, Gillespie proposed the Slow Scale Stochastic Simulation Algorithm (ssSSA) [38, 36, 39, 40]. This algorithm uses a quasy steady-state approximation for the fast reactions of the system. The algorithm explicitly simulates only the slow reactions, the fast ones take values governed by steady-states assumptions of convergence. Gillespie defines for that virtual fast processes, that are not touched by the slow reactions. These virtual fast processes can then gives the new populations for the fast species, without simulating them explicitly.

Other ways of simulating multiscale systems have been proposed [41]. One of them suggests to simulate the fast reactions using a deterministic approximation [42, 43, 44]. The goal is to replace the stochastic processes of the fast reactions with big population by an ODE. In this manner, fast simulation of the fast reactions can be attained, while keeping stochastic simulations for the slow reactions. This Stochastic-deterministic approximation may still pose some convergence problems, as no real proofs of convergence toward the stochastic averages of the initial fully stochastic system have been provided.

To complete this overview, there are also algorithm simulating the reactions in a spatially-dependent context, by using diffusion methods [45, 46, 47, 48].

Several toolkits implementing those simulations algorithm are available [49, 50, 51].

3.3 Considerations on the assembly plan

Continuing on the discussion with Klavins' approach to self-assembly, we discuss the problem of the assembly plan and its relation to the reaction rates.

The complete problem of constructing a final assembly from initial parts can be divided into two parts: a discrete and a continuous one.

1. The discrete part consists of the assembly plan itself. It represents a finite and discrete set of rules to construct the final target.

2. The continuous part is the rate of evolution of the assembly, driven by the assembly plan but subject to continuous reaction rates. Those rates can take continuous values which will affect the final outcome of the assembly.

We argue that, taken to the limit, the problem is actually completely continuous. The reaction rates, when pushed toward 0, will deactivate a part of the assembly plan automatically.

We wish then to study the optimization of these continuous reaction rates, as we think they might give more insight on the relations between parts of the plan and as they encompass the same power as the discrete part.

In order to go in that direction completely, one would need to consider the "full assembly plan". Such a plan would consists of every possible assembly steps towards the creation of a final assembly. Indeed, it would become quite big quickly, but pruning is possible, mainly because we assume that we have heterogenous pieces that have highly specific assembling sites. Such a plan is easily obtained using any enumeration method, for example Pólya enumeration [52].

But in this work, we only consider a subset of this "full assembly plan". We assume that we are given a part of this plan, which already creates the final assembly. We then study only the effect of the reactions rates on this plan, and see what parts of it an optimization technique will push forward or cut down. This is an assumed simplification for the current work.

Chapter 4 Puzzle test-case implementation

We apply our framework and methodology on a specific problem: A puzzle assembly task. The goal is to assemble several heterogenous pieces together to create a specific final shape. This is done using a robotic platform, simulated using a realistic physics simulator: Webots [1].

This allows us to get a measurable system which could be transformed into a real platform pretty easily.

4.1 Definition of the puzzle test-case

We define the puzzle test-case as follows:

- Let a puzzle of square shape, with area 25, be constructed out of 5 pieces of area 5 each with different given shapes.
- Let the final assembly shapes S_k of this puzzle be know.
- Let the set of assembly plans P_k leading to the final shapes S_k be known.
- Let the puzzle pieces assemble by bi-directional connections. One connection is enough for two pieces to be attached. These connection and their positions on the different pieces are known.
- Pieces can be assembled and disassembled.
- Piece can lie around or move randomly. We study those two possibilities, but we concentrate on the first one.
- Consider an arena of sufficiently large size so that small scale interactions dynamics can be ignored.
- Fill this arena with n_i initial pieces of each shape i. Let these n_i number be the exact numbers needed to construct N final assemblies.
- Consider M robots, able to pick up pieces and to make them assemble and disassemble.
- Allow a recognition by the robots and by the pieces of the shapes and connection points when an encounter occur.
- **Then:**
  How can you manipulate those initial pieces so that after a time T_f, the number of

assembled puzzles X_{S_k} corresponds to desired values?

We introduced here the goal of this whole test case: to control the output of the system in term of assembled puzzles.

This can also be applied on the fly, to control what the system should produce. We want here to take advantage of the modularity of the platform, as our robots can produce any desired assembly.

An application of that flexibility can be what we call a *green manufacturing* process. We mean by that the automatic recycling of finished puzzles S_1 to create new puzzles S_2, only by telling the robots what we want as final assembly. This will be studied in Chapter 6.

4.2 Scale and complexity considerations

We have a lot of different possibilities for the robot behaviors. We chose to consider different directions depending on the available information and capabilities of the robots. If we want to produce something really scalable, then using robots as simple as possible is interesting. But on the other hand, this would most likely affect the performances. So we will try to measure this with respect to several considerations.

	Assembly plan known	**Local plans only**
Local information	*Current study*	*Future work*
Global information	Market-based, Assembly line	Market-based

Table 4.1: Robot behavior depending on available informations.

The first distinctions we make are shown in Table 4.1. The most important criteria is the availability of information about the robots and pieces positions and states. If we have a Global information state, then the problem reduces to a classical assembly at the macro-level. With multiple robots, this could be solved using Market-based strategies, which do not interest us here. So we only consider having Local information about the pieces and robots positions.

The next distinction is the availability of the full assembly plan. Knowing the full assembly allows to optimize a-priori a plan and to stick to it when building the puzzle. But this needs some computing capabilities and communications between pieces and robots. A more crude possibility is forbidding this full knowledge, and having to recreate the global plan only from local connections possibilities.

We are currently studying the *Local info / Assembly plan* case. The *Local info / Local plan* case is very interesting but will be done in further works.

Furthermore, we have the following choice to make: *should the pieces be disassembled or not*? As we will develop during the project, this depends on the possibility of bad assemblies and on the flexibility of creation needed. Indeed, if we want to apply our system to the *green assembly* process, we have to be able to destroy final products into simpler ones.

This is in accordance with biology, which tends to reuse products and compounds for different purposes. This allows a flexibility and adaptation necessary when we do not know

the goal a priori.

A quick precision should be done on the moving capabilities of pieces. We concentrate here on the task of assembling immobile pieces. In this case, the robots behaves like transporters. But, seen more abstractly, this is the same as having moving pieces on their own. We also study the case of pieces moving around randomly. This simulates more closely a self-assembly task, driven only by geometric constraints for the assembly. An interesting scenario is to add robots to the system with moving pieces. In this setup, the added robots behaves like enzymes: they modify the system by acting on it. This creates three different scenario: the *robot-transporter*, the *self-assembling pieces* and the *mixed assembly*.

In all this section, we only consider forward assemblies of pieces, that is, we never disassemble things. This will be explored further on, in the Augmentation step, Chapter 7.

4.3 Webots implementation

We chose to develop our puzzle test case using the realistic physic simulator Webots [1]. This allows us to simulate robots and assembly process, while still being affected by noise and geometric properties. We could have developed a simpler simulator, for example a point-based simulator for an assembly process, but we think that the added physical reality of Webots makes it easier to understand how real-world problems could behave in our framework.

Webots offers directly a capability to assemble our puzzle pieces: connectors. These connectors behaves like active electromagnets, that can be turned on and off. The goal is to mimic the assembly process of molecular compounds, tied by low-energy bounds.

The first implementation of the controllers for the robot transporters scenario on Webots has been created by Spring Berman for her project in the course MEAM620 by V. Kumar at the University of Pennsylvania. Loïc Matthey created the Webots worlds and subsequently modified the controllers code to improve scalability, add the support for arbitrary assembly plans, change the movement patterns and create the self-assembly and mixed-assembly scenarios.

4.3.1 Pieces

A piece consists of a solid body, several small feet and several connectors. There is only one top connector for the robots to carry the piece around. There are several side connectors, to connect to other pieces, their number depend on the piece type. See Figure 4.1 for an example of such a piece.

We created a set of four different pieces, each with different shapes and different connecting capabilities. See Figure 4.2 for the different pieces.

These pieces are endowed with several other capabilities:

1. They have a radio emitter/receiver to communicate with robots or other pieces. The communication range is set to 40cm for the pieces.

2. They can activate or deactivate their connectors at will.

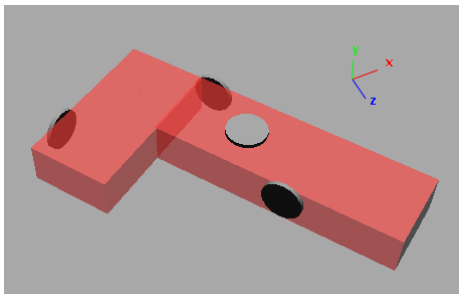


Figure 4.1: Piece overview. Top connector is for the robots, side connectors are for the other pieces.

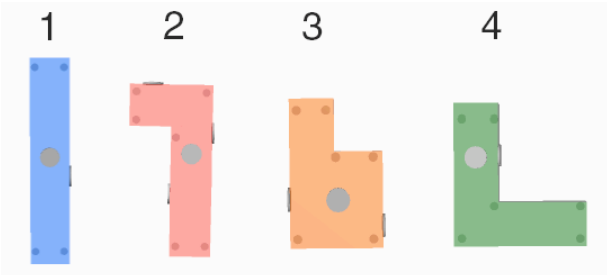


Figure 4.2: Four different pieces created, with their different connecting capabilities.

3. They know what type they are and where are their connectors.

4. They know the assembly plans to create the different final puzzles. They also know how they should be oriented for optimal assembly with a give other piece. We will see later that this can be relaxed.

5. They are fairly intelligent, meaning that they have computational capabilities. The pieces can communicate with other robots, maintain a internal state of their situation.

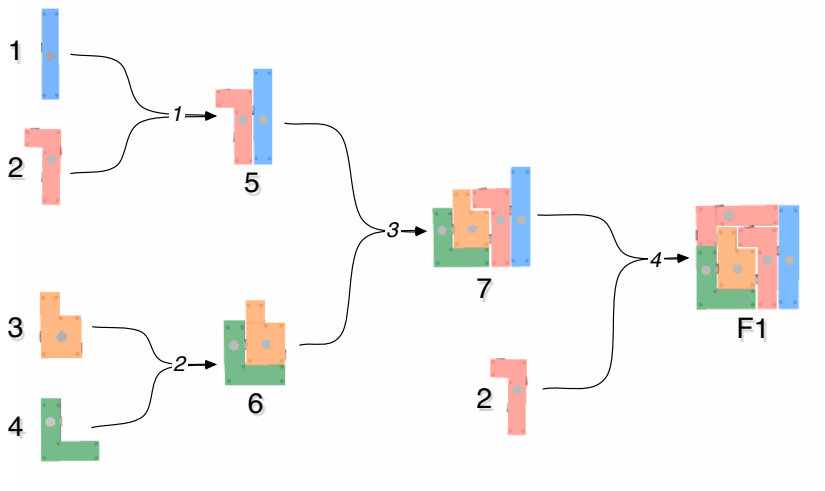
Assembly plans

We consider two final puzzles in our project. There are several way of assembling them. We first study two specific plans, but will generalize that when trying to control the chemical reactions network. The two plans and the different mid-assemblies resulting are presented in Figure 4.3.

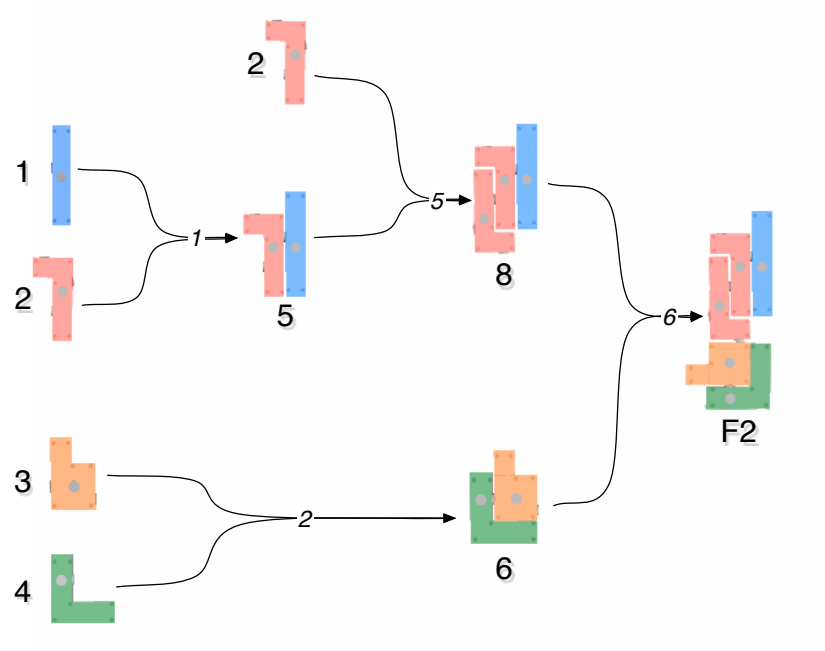
4.3.2 Robots

For the robots, we used the KheperaIII model available in Webots. It offered a small scale yet not too crude mobile robot for our first implementation.

In order to manipulate the pieces, we equipped the robots with a protruding carrying arm (see Figure 4.4). This arm consists of a simple bar with a mobile Connector at its end. The Connector is allowed to turn around 360° using a rotational Servo, in order to orient



(a) First final puzzle plan

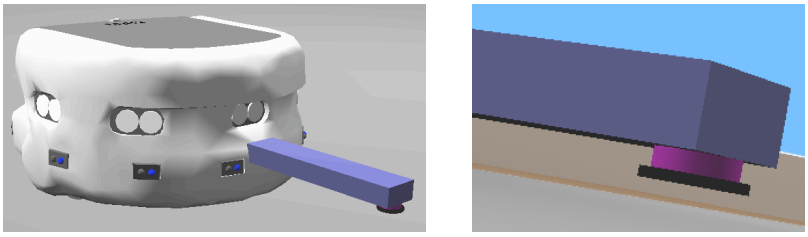


(b) Second final puzzle plan

Figure 4.3: Assembly plans for the two final puzzles considered. All groups of connected pieces (mid-assemblies or products) are given an unique name in a form of a number. Arrows show the assembly steps, with their name as number.

the carried piece in any possible direction. The length of the arm is sufficient to rotate any mid-assembly without hitting the robot's body.

When being carried, the piece does not touch the ground, as they are very light-weight.



(a) KheperaIII robot.

(b) Protruding arm, with rotating connector.

Figure 4.4: KheperaIII robot model in Webots, with protruding arm.

The robots have similar components to the pieces:

- They have a radio emitter/receiver, to communicate with pieces and other robots. The communication range is set to 60cm for the robots. This local radio is also used as a bearing detection mechanism, giving the relative angle between two emitter/receivers. This is used when a robot needs to grab a piece, or when the piece has to be rotated by the rotating arm of a given angle.
- They can control the rotation of the servo at the tip of their protruding arm.
- They communicate with their carried piece to know what type it is and what is its relative angle.
- They know the assembly plans to create the final puzzles.
- They move around randomly in the arena, while avoiding other robots and walls using their infra-red distance sensors.

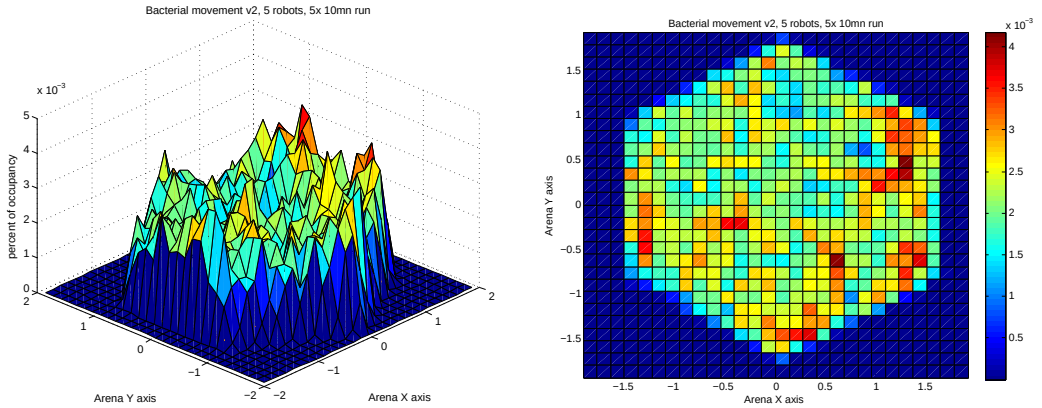
Movement pattern

We want our robots to be evenly distributed around the arena in average. This property, the *well-mixed* property, allows us to use non-spatial mathematical models.

In order to satisfy this property, the robots have to move around in a specific manner. We chose to make them move in a bacterial-like movement. This movement, "chemotaxis", allows bacteria to move around, search for nutriments and avoid dangers. It is based on a forward movement, and random "tumbling". A "tumble" is a random turn. The bacteria sample the concentration of nutriments or dangerous chemicals, and performs a temporal integration on them while moving. An increase in a nutriments concentration tends to reduce the number of tumbling, promoting movement towards the spacial gradient. When the gradient is constant, the bacteria performs tumbling at a constant rate [53, 54, 55, 56].

In our case, we do not follow any gradient. We only make the robots move forward for a random distance, and then turn randomly around, before moving forward again. This creates a random movement that is supposed to cover more uniformly the space.

We verified this assumption using Webots and our robots. See Figure 4.5 for the average space covered by five robots over 5 runs of 10 minutes each. We see that the space is nearly evenly covered, which shows that our property is ensured.



(a) Covered space, 3D

(b) Covered space, 2D

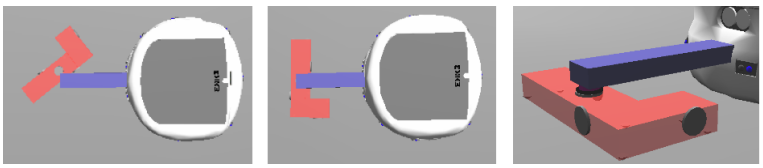
Figure 4.5: Average coverage of the arena by 5 robots moving in a brownian-like motion, over 5 runs of 10 minutes.

Behavior

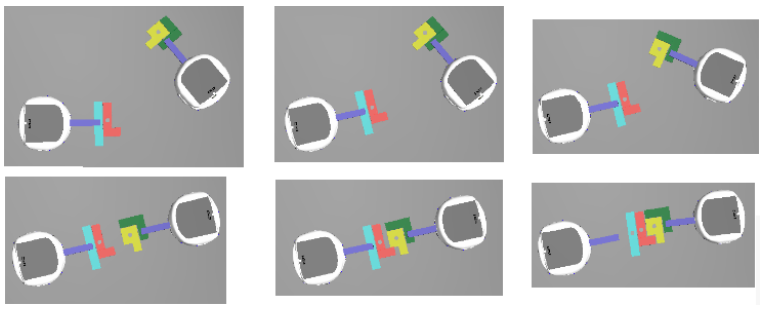
The robots and the pieces are placed randomly in an hexagonal arena of fixed size. They can only communicate in a local range: 40cm for robot to piece, 60cm for robot to robot. The behavior is then as follows:

- Robots move around, searching for lying pieces. They avoid the walls and other robots using a Braitenberg vehicle controller. The move around randomly following the bacterial-like movement pattern presented before.
- Robots and pieces broadcast messages locally, telling their current configuration and state. A configuration is a unique name for a set of assembled pieces, for all possible assemblies present in the plans we are using to build the final puzzles.
- When a robot receives a message from a free piece (i.e. they are in a small communication range), it aligns with it, go towards it and carries it. This alignment uses relative range and bearing offered by the emitter/receiver nodes of Webots. See Figure 4.6(a).
- While carrying the piece, the robot start moving around again, searching for another robot with a compatible piece. Robots communicate with small range messages broadcasted at all time.
- When two robots carrying pieces come into communication ranges, they exchange message and look into the assembly plan. If their pieces correspond to no stored assembly step, they moves away from each other.
- If their pieces can be assembled, the robot start an assembly procedure. According to the piece type and the assembly plan, the robot first orient their pieces so as to show the good connector in front. Again we use the relative range and bearing of emitter/receiver nodes to perform that alignment. This step will be relaxed in a experimental scenario to account for a random orientation of pieces for the assembly. See Figure 4.6(b).
- Then the robot align each other. The robot starts to approach, allowing the pieces to

(a) Encountering between a robot and a piece. The robot aligns itself with the piece.



(b) Alignment of piece by the rotating connector



(c) Approach between two robots and assembling of pieces.

Figure 4.6: Assembly behavior of robots and pieces.

lock to each other. When the two pieces are locked, one of the robot leaves, letting the other one with the assembled pieces. This robot resume searching for a new piece to assemble with, while the newly freed robot starts looking for a lying piece. See Figure 4.6(c)

4.3.3 Experiment platform

Our goal is to reproduce experiments extensively and study the data in Matlab. We thus need a pretty robust system, as well as a centralized way to prepare and store these experiments.

The robustness is ensured by adding several checks and reset capabilities in the behaviors of the robots and pieces. There are still problems that could arise, for example due to physical simulation problems, or a discrepancy between the actual state of the simulation and the way the robots see it. We can only measure what the robots know, so this can create some problems.

As a centralized medium for the experiments, we use a supervisor node in Webots. This supervisor takes care of the experiments and writes the results to different files. It resets the experiments after a maximum elapsed time and takes care of the initial random positions of all pieces and robots. When an assembly step occurs, robots send specific messages to that supervisor, which will save them accordingly.

4.3.4 Python world generator

Webots does not provide an easy way of varying the components of a given world. However, as we want to control easily the number of robots, pieces, the size of the arena and several other parameters. Therefore, we created our own Webots world generator, in Python.

This world generator is available online on the mailing group of Webots, as it was build to be easily extended. It takes the following inputs:

- A set of template files. They store parts of a classical Webots VRML world file, but with added free parameters in them.
- A input XML file describing the world to create. This file defines which templates to use and how many instances of them to create and finally assigns values to the free parameters.

It is easy to add new templates and extend this to different applications.

This generator allows us to generate experimental worlds containing different numbers of pieces and robots easily. We study for now a world with 5 pieces and 4 or 5 robots, and a world with 15 pieces and 15 robots.

4.4 The robot transporters scenario

Characteristics: Lying pieces, robots carry and assemble them.

This system represents either a self-assembly task if we abstract the robots, or a transporting and assembly task. The pieces can not move and rely on the robots to create a puzzle.

4.4.1 Simulation results

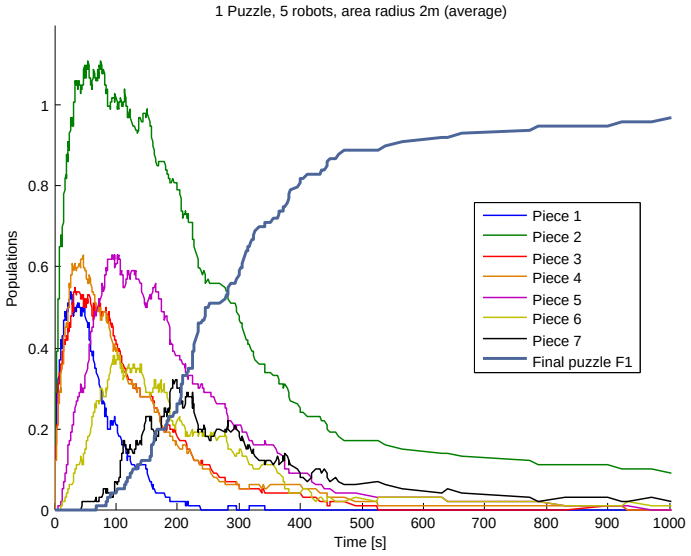
Experiment 1: 5 pieces and 4 robots, final puzzle F1 only

Setup:

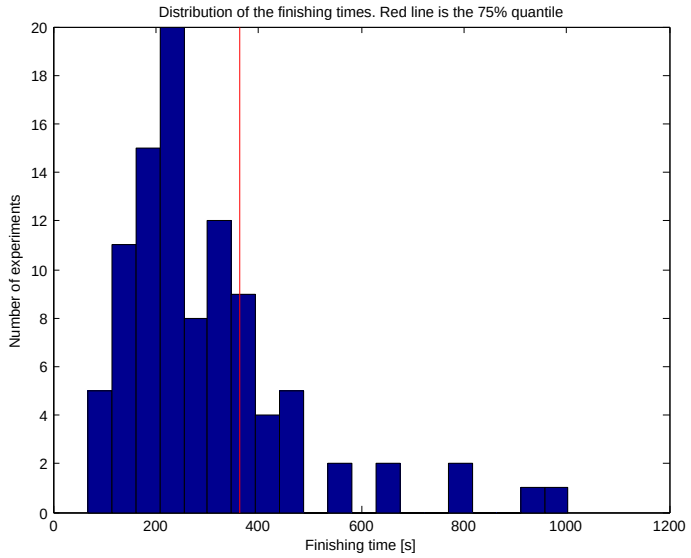
- Hexagonal arena, radius 2m.
- 100 experiments.
- 20 minutes maximum per experiment.
- Pieces and robots initialized at random positions.
- Pieces are aligned by the robots before an assembly.
- Robots follow the plan to create the final puzzle F1 only, given in Figure 4.3(a).

We see on Figure 4.7(a) that the final puzzle F1 follows an exponential saturating curve, tending toward 1. This is what we expected, as only one puzzle can be created. The curve does not attain exactly 1, meaning that some assemblies are not successful. Indeed, we have a success rate of assembly after 20 minutes of 96%.

But looking at Figure 4.7(b), which shows the histogram of the times of creation of the final puzzles over all experiments, we see that 75% of the puzzles are actually completed after



(a) Averaged populations of products over time.



(b) Histogram of the finishing times of the final puzzles F1. Red line at 400 seconds shows the 75% quantile.

Figure 4.7: Physical simulation results for the robot transporter scenario, Experiment 1: 5 pieces, 4 robots and final puzzle F1 only.

6 minutes and 40 seconds on average. This is a good results, as it means that most of the experiments were completed quickly.

Experiment 2: 15 pieces and 15 robots, final puzzle F1 only

Setup:

- Hexagonal arena, radius 3m.
- 100 experiments.
- 20 minutes maximum per experiment.
- Pieces and robots initialized at random positions.
- Pieces are aligned by the robots before an assembly.
- Robots follow the plan to create the final puzzle F1 only, given in Figure 4.3(a).

Figure 4.8(a) shows a similar behavior than before. The curve is smoother, due to the bigger amount of pieces and possible final puzzles. The curve again tends exponentially towards the maximal puzzle number, 3. But it converges to an even smaller number, as more assemblies goes wrong. After 20 minutes, we have a success rate of assembly of 3 puzzles of 80% only. This shows that some things can still go wrong in our physical simulations, which affects the final assembly yield.

Looking at Figure 4.8(b), we see that the 75% quantile for the successfully assembled 3 final puzzles is at 11 minutes. This is still a pretty good result, which shows that our approach is scalable to a higher number of pieces and robots, assuming that the space available for the movements does not become too small.

Experiment 3: 5 pieces and 5 robots, final puzzles F1 and F2

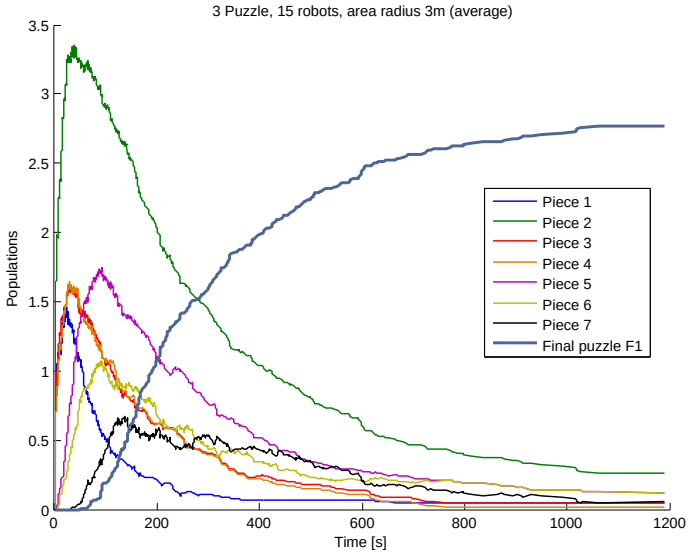
Setup:

- Hexagonal arena, radius 2m.
- 100 experiments.
- 20 minutes maximum per experiment.
- Pieces and robots initialized at random positions.
- Pieces are aligned by the robots before an assembly.
- Robots follow the plans to create the final puzzles F1 and F2. See Figure 4.3.

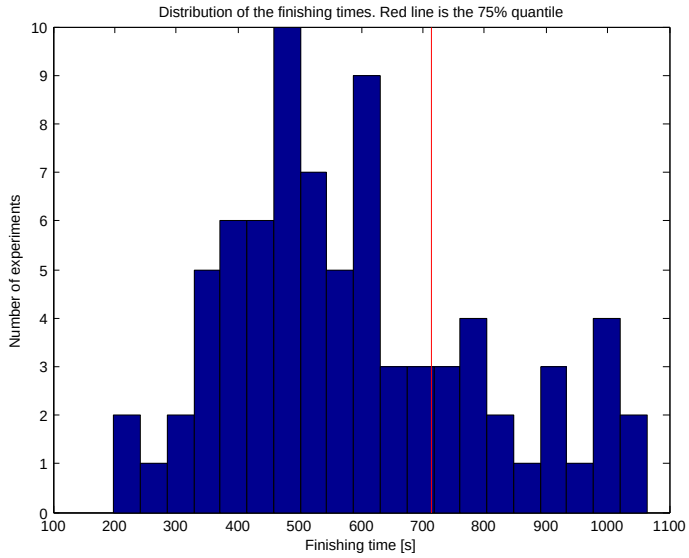
Figure 4.9(a) shows an interesting result. We see that the two final puzzle converge to a value that sum (more or less) to 1. But the distribution between the two assemblies is not even, we have 60% of final puzzle $F2$ and 40% of final puzzle $F1$. By looking at the assembly plans and the available initial pieces, we think it is due to reaction 5. This reaction uses piece 5, which is created early, and uses another piece 2, which is easily available (purple curve and green curve). Compared to reaction 3, which uses a piece 5 and a piece 6, which takes more time to be produced, there is less time dependence on the path to F2. This tends to promote it.

This discrepancy triggered the idea of being able to control the ratio between the two final puzzles, by modifying the system. This will be the subject of our Augmentation step and optimization of the model, Chapter 6 and 7.

From Figure 4.9(b), we see that the 75% quantile for the successfully assembled final puzzle is at 4 minutes 30 seconds, with a success ratio of 97%. This is again very good, few assemblies go bad, even with the two possible final puzzles.



(a) Averaged populations of products over time.



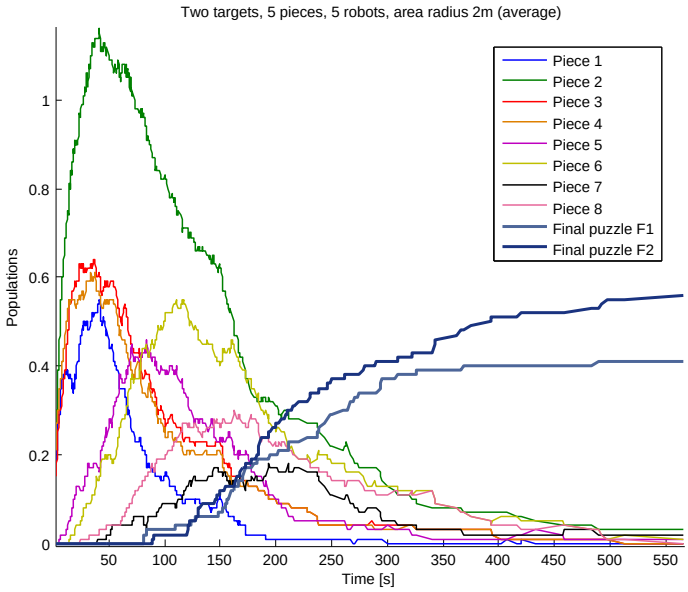
(b) Histogram of the finishing times of the final puzzles F1. Red line at 400 seconds shows the 75% quantile.

Figure 4.8: Physical simulation results for the robot transporter scenario, Experiment 2: 15 pieces, 15 robots and final puzzle F1 only.

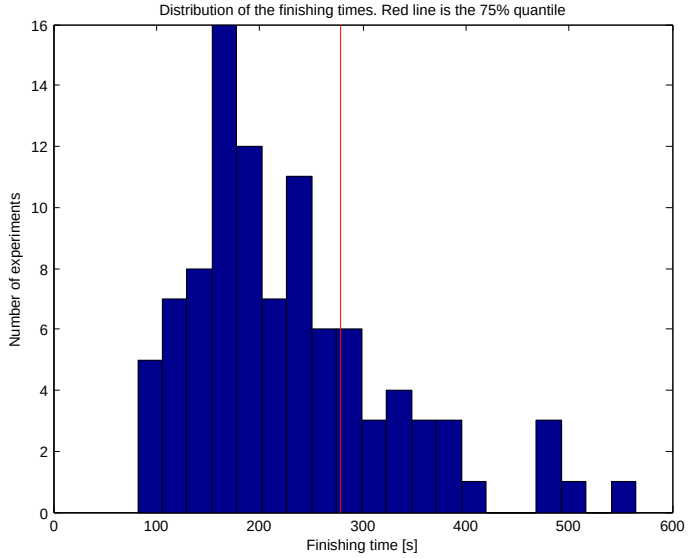
4.5 The self-assembling pieces scenario

When the pieces can move around and assemble on their own, robots are not necessary anymore. This scenario is closer to a real nano self-assembly task, but at a macro-scale level.

We developed such a scenario in Webots, using the same pieces with several modifications:



(a) Averaged populations of products over time.



(b) Histogram of the finishing times of either final puzzle F1 or F2. Red line at 400 seconds shows the 75% quantile.

Figure 4.9: Physical simulation results for the robot transporter scenario, Experiment 3: 5 pieces, 5 robots and final puzzles F1 and F2.

- The pieces are pushed by individual forces, of randomly chosen direction. The pieces have a low friction coefficient with the floor to allow easy movement.
- There are forces applied onto the pieces when they approach the walls too closely. This introduce a wall-avoidance in a smooth fashion. The repulsive force F_r applied is

directed toward the center of the arena and an amplitude inversely proportional to the current distance to the walls:

$$F_r \sim \frac{1}{\left(\sqrt{x^2+y^2} - R_a sin(\frac{\pi}{3})\right)^2} \cdot \begin{pmatrix} -x \\ -y \end{pmatrix}$$

$R_a sin(\frac{\pi}{3})$ is the radius of the incircle to the hexagonal arena of radius R_a.

- The pieces attracts each others in a small radius. This was introduced to improve the encountering rate, which was not comparable to the one we had before. Indeed, a collision of two pieces is less likely than the encountering of two communication circles as we had before. This force attracts the pieces for some time, and then repulse then, to mimic a missed assembly.

All these modifications create a scenario where the assembly rates are much smaller than before, but which can still create some final puzzles. Unfortunately, due to robustness issues, we did not manage to get systematic experiments in time for that scenario. Its study will be done in further works.

4.6 The mixed assembly scenario

We can combine the two last scenarios into this fully complex one. The pieces can move around and assemble on their own, but can also be carried around and assembled by robots.

In order to make the carrying possible, the pieces stop moving when a robot is trying to grab them. Moreover, a free piece cannot interact with a carried piece. The robot has to drop it first.

This scenario closely resembles a biological process with enzymatic components. The pieces assemble following their own dynamics, which are improved by the robots via their specific actions and orientation capabilities.

Again we did not manage to completely study this scenario. We leave it as further work, not without regrets.

Chapter 5 Mathematical model of the puzzle test-case

5.1 Model definition

We introduce now the mathematical model used to represent our puzzle test-case system. As told earlier, we use a chemical reaction network framework (see Section 3.2 for background on this subject).

We only consider the robot transporters scenario, the other scenario can be modeled in the same fashion.

We assign a reaction to each assembly step in the creation of the final puzzles. Looking back at Figure 4.3, each numbered assembly step corresponds to a reaction in our chemical reaction network. Furthermore, we add 4 new reactions, representing the grabbing of lying pieces by the robots. The products and reactants are the different mid-assemblies, plus the 3 lying free pieces and the robots. All reactions are second-order reactions, as they depend on the encountering of two different reactants upon reaction.

We obtain the following chemical reaction network (Equation (5.1)):

$$\begin{aligned}
X_R + X_1^f &\xrightarrow{e_1} X_1 & \qquad X_R + X_3^f &\xrightarrow{e_3} X_3 \\
X_R + X_2^f &\xrightarrow{e_2} X_2 & \qquad X_R + X_4^f &\xrightarrow{e_4} X_4 \\
X_1 + X_2 &\xrightarrow{k_1} X_5 + X_R & \qquad X_2 + X_7 &\xrightarrow{k_4} X_{F1} + X_R \\
X_3 + X_4 &\xrightarrow{k_2} X_6 + X_R & \qquad X_2 + X_5 &\xrightarrow{k_5} X_8 + X_R \\
X_5 + X_6 &\xrightarrow{k_3} X_7 + X_R & \qquad X_6 + X_8 &\xrightarrow{k_6} X_{F2} + X_R
\end{aligned} \tag{5.1}$$

X_R is a robot, X_i^f are the free lying pieces, X_k are the carried pieces and X_{Fj} the final puzzles.

The variables can be defined as the number of each piece type, where the number is a continuous function of time [29]. This network can then be transformed in the following associated ODE system (Equation (5.2)):

$$
\begin{cases}
\dot{x_R} &= -\sum_{l=1}^{4} e_l x_R x_l^f + k_1 x_1 x_2 + k_2 x_3 x_4 + \\
&\qquad k_3 x_5 x_6 + k_4 x_2 x_7 + k_5 x_2 x_5 + k_6 x_6 x_8 \\
\dot{x_1^f} &= -e_1 x_R x_1^f \\
\dot{x_2^f} &= -e_2 x_R x_2^f \\
\dot{x_3^f} &= -e_3 x_R x_3^f \\
\dot{x_4^f} &= -e_4 x_R x_4^f \\
\dot{x_1} &= e_1 x_R x_1^f - k_1 x_1 x_2 \\
\dot{x_2} &= e_2 x_R x_2^f - k_1 x_1 x_2 - k_4 x_2 x_7 - k_5 x_2 x_5 \\
\dot{x_3} &= e_3 x_R x_3^f - k_2 x_3 x_4 \\
\dot{x_4} &= e_4 x_R x_4^f - k_2 x_3 x_4 \\
\dot{x_5} &= k_1 x_1 x_2 - k_3 x_5 x_6 - k_5 x_2 x_5 \\
\dot{x_6} &= k_2 x_3 x_4 - k_3 x_5 x_6 - k_6 x_6 x_8 \\
\dot{x_7} &= k_3 x_5 x_6 - k_4 x_2 x_7 \\
\dot{x_8} &= k_5 x_2 x_5 - k_6 x_6 x_8 \\
\dot{x}_{F1} &= k_4 x_2 x_7 \\
\dot{x}_{F2} &= k_6 x_6 x_8
\end{cases}
\tag{5.2}
$$

Obviously, this notation is less compact, yet has the same meaning.
We can also represent the network in matrix form:

$$\dot{\mathbf{x}} = \mathbf{SKy}$$

S is the stoichiometric matrix, containing the stoichiometric coefficients m_r and n_p as defined in Equation (3.1) in Section 3.2.1. **K** is the matrix of stochastic constant rates. y is a vector of compounds, in our case the set of all bilinear terms in Equation (5.2) for example.

We can also relax the x_R and x_i^f terms, if we decide to look only at the real assembly process therein. Doing so is consistent if we assume that we have a big number of robots to carry the pieces around, and that they grab the pieces very quickly compared to the actual assembly process. This approach is similar as doing a quasi-steady state simplification, for a multiscale system where the robots are acting quicker than the rest of the system. Such a relaxation simplifies the whole system and its analysis, we will use it in the next chapter.

5.1.1 Simulation of the model

We simulate our models using two different approaches:

1. ODE solving. We use Matlab to solve numerically the system (5.2), using a classical ode45. We use libSBML [57] for Matlab to write and read from SBML files onto ODE files.

2. Stochastic Simulation Algorithm. We use the StochKit toolbox [50], developed by Petzold et al. StochKit is a simulation framework for stochastic simulations written in C++. It allows a very fast exact simulations of chemical reaction networks.

5.2 Parameter fitting

5.2.1 Theoretical value of reaction rates

The chemical reaction network is easy constructed from the assembly plans considered, but we still need to find values for the stochastic constant rates k_i and e_l.

We decompose the stochastic constant rates as follows:

$$k_i = p_i^e \cdot p_i^a \tag{5.3}$$

where p_i^e is the probability of an encounter between two elements and p_i^a the probability of successful assembly.

p_i^a can be easily measured, or assumed to be 1 if the robots manage to align the pieces correctly before each assembly step.

If we assume that our model is non-spatial, i.e. that the probability that two product are at a given position is independent of the time and uniformly distributed over the available arena space, we can make an initial informed guess on the encountering probability. We take an approach used in [58, 59, 60], giving the following relation for the encountering probability:

$$p_i^e \sim \frac{1}{A_{total}} v_r T w_d^i \tag{5.4}$$

Where A_{total} is the arena size, v_r is the average speed of an element, T is the timestep (fixed to 1 in our case) and w_d^i the width of detection of an element. This expression can actually be linked back to the literature on chemical process simulations [30, 41, 48]. For chemical process, the probability of collision depends on the volume swept by one molecule, which gives the probability that another molecule will collide it in the next dt. Equation (5.4) is exactly the same, as shown on Figure 5.1. In our case, as we work in a 2D plan, we only have surface swept, which is exactly given by the right part of (5.4). Dividing by the total arena size and assuming the elements are distributed uniformly in the arena, this gives directly the probability of encountering between two elements.

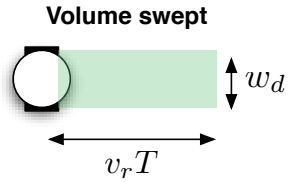


Figure 5.1: Graphical interpretation of the encountering probability and link to the volume swept used in chemical simulations.

In our test case, we measured v_r over 50 simulation runs using the robot movement pattern described earlier, the average speed was $0.128 m/s$. A_{total} is also easily computed ($= 6(R^2 \frac{\sqrt{3}}{4})$, i.e. the sum of the six equilateral triangles of radius R). w_d^i becomes the double of the communication radius between robots or pieces, as it defines the range for the start of an assembly.

Rates verification

We checked this initial guess by measuring the encountering rates in Webots. We create a world with one lying piece and one searching robot, we measure the time to encounter, over 200 experiments. As discussed in the theory of chemical reaction networks, these times are distributed following an exponential of the encountering probability. We load the times in Matlab, and fit an exponential distribution to get this probability (Figure 5.2).

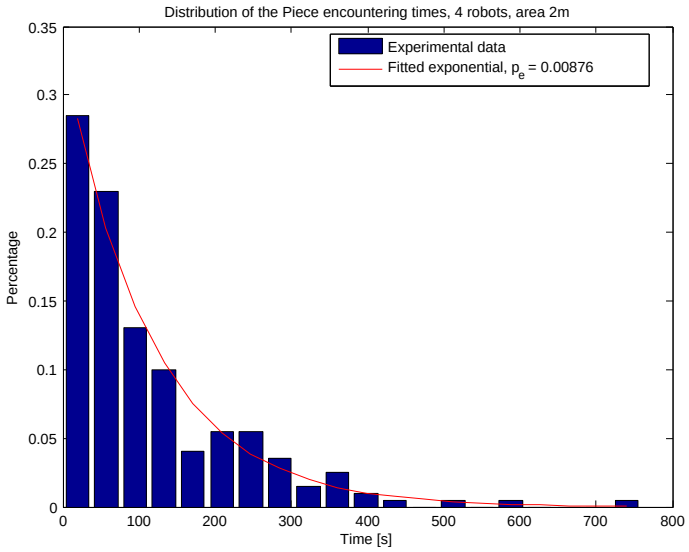


Figure 5.2: Encountering times for the Webots experiment, with the fitted Matlab exponential.

We also add "dummy" robots in the arena, that only avoid each other without looking for the piece. It measures the effect of overcrowding on the well-mixed property we're trying to ensure within our Webots simulation.

The results of the comparison between the theoretical and measured rates of encountering are shown in Figure 5.3. We see that the theoretical guess is pretty accurate, even tough it is overestimated when more robots are present. The added robots seems to disturb the capacity of a robot to encounter a piece. This can be due to the fact that the robots perturbs the trajectories while avoiding each others, overcrowding some areas and forbidding the access to others.

The same measurement was performed for robot to robot encountering and gave similar results. We do not show it here.

5.3 Comparison with physical simulation

We simulate the chemical reaction network (5.1) using the two approaches presented in Section 5.1.1. Depending on the initial conditions for the number of robots and pieces, we have different experiments, following Section 4.4.

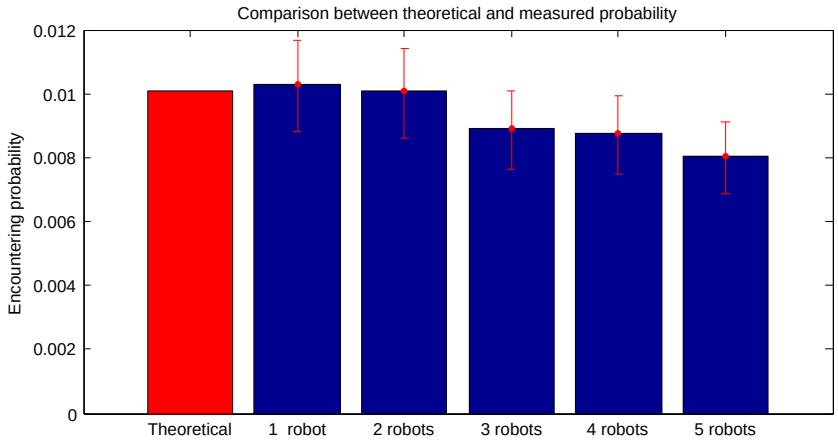


Figure 5.3: Comparison between the theoretical guess for the encountering probability p_e^i and the measured encountering probability in Webots. The red intervals represents the confidence intervals for the fitted encountering probability.

Stochastic constant rates values

Through all these experiments, the stochastic constant rates e_l and k_i take similar values, conditioned by specific parameters presented in each experiment setup.

Encountering rates $e_l = \frac{1}{A_{total}} \cdot v_r \cdot w_d^p \quad \forall l \in \{1..4\}$.

Reaction rates $k_i = p_i^a \cdot \frac{1}{A_{total}} \cdot v_r \cdot w_d^r \quad \forall i$.

Arena size $A_{total} = 3 \cdot R_a^2 \cdot \frac{\sqrt{3}}{2}$.

Average speed $v_r = 0.128$ measured in Webots.

Piece communication width $w_d^p = 2 \cdot 0.4m$, pieces are set to communicate in a $40cm$ radius.

Robot communication width $w_d^r = 2 \cdot 0.6m$, robots are set to communicate in a $60cm$ radius.

Probability of successful assembly p_i^a depends on the experiment being studied, more precisely on the assembly plan used. We measured it in Webots, over 100 runs, for the assembly plan creating only the final puzzle F1 and for the assembly plan creating the final puzzles F1 and F2. Results will be specified in the experiments' setup.

5.3.1 Experiment 1: 5 pieces and 5 robots, final puzzle F1 only

Setup:

Initial conditions: $X_R = 5, X_1^f = 1, X_2^f = 2, X_3^f = 1, X_4^f = 1, X_i = 0, X_{Fk} = 0$.

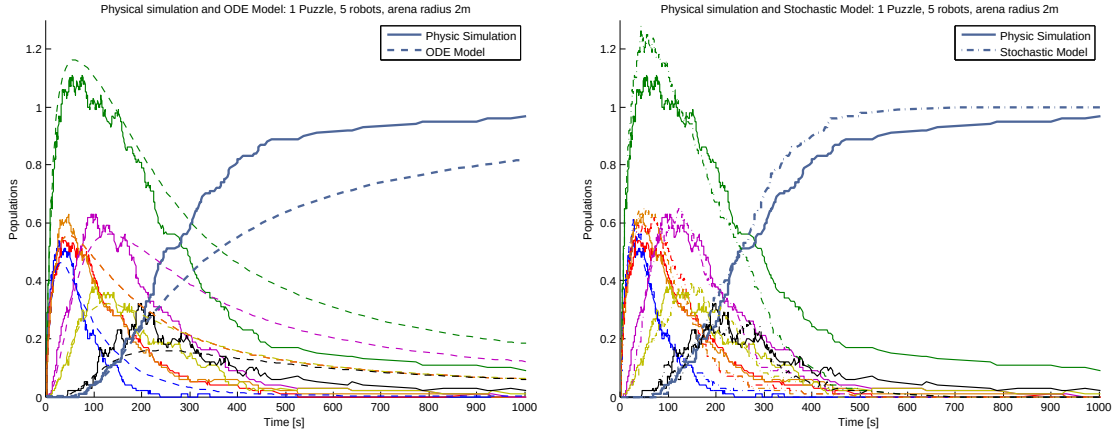
Experiment duration: 20 minutes.

Number of experiments (Stochastic simulation only): 100.

Arena size: $R_a = 2m$.

Probability of successful assembly: all p_i^a where more or less equal to 0.98. Thus we set $p_i^a = 0.98$.

See Figure 5.4 for the comparison between the simulated chemical reaction networks and the Webots physical simulation. The results from the Webots simulation are taken from Section 4.4.1. We show the averaged populations over 20 minutes.



(a) ODE simulation vs physical Webots simulation

(b) Stochastic simulation vs physical Webots simulation

Figure 5.4: Comparison between the models simulations and the physical Webots simulation for the puzzle test-case, scenario 1, experiment 1.

We see that they fit closely to the physical simulation. Two differences arise:

- The ODE simulation population values are too small, because of the slow copy number of components (Figure 5.4(a)). When doing an ODE approximation, one assumes that the number of components is big enough so that using continuous numbers does not have much effects. This is not the case here, where we work with numbers smaller than 10. On the other hand, the Stochastic simulation accurately captures this property, as it works with discrete numbers.
- The stochastic simulation attains 1, whereas the physical simulation stays below (Figure 5.4(b)). These results show that, while there are problems in the Webots simulation (pieces stuck together, blocked against the walls), this is not taken into account in the model. We could modify the model by introducing ways of deactivating some pieces to try to fit this difference. But we do not take much interest in that for this current work.

5.3.2 Experiment 2: 15 pieces and 15 robots, final puzzle F1 only

Setup:

Initial conditions: $X_R = 15, X_1^f = 3, X_2^f = 6, X_3^f = 3, X_4^f = 3, X_i = 0, X_{Fk} = 0$.

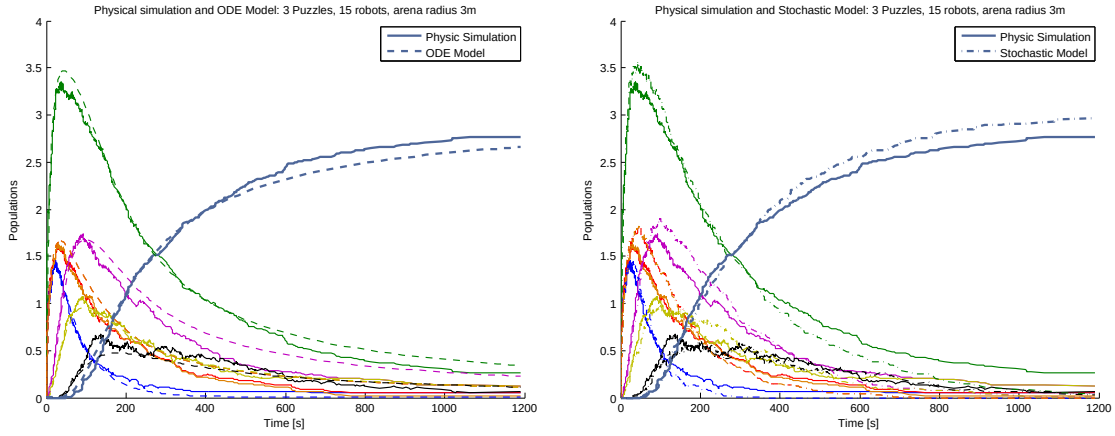
Experiment duration: 20 minutes.

Number of experiments (Stochastic simulation only): 100.

Arena size: $R_a = 3m$.

Probability of successful assembly: $p_i^a = 0.98$.

Under the same assumptions than for experiment 1, see Figure 5.5 for the comparison in this scenario.



(a) ODE simulation vs physical Webots simulation (b) Stochastic simulation vs physical Webots simulation

Figure 5.5: Comparison between the models simulations and the physical Webots simulation for the puzzle test-case, scenario 1, experiment 2.

The results are similar to experiment 1. With a higher number of elements, the ODE simulation is closer to the correct physical simulation. The stochastic simulation is still very accurate but attains the maximal number of puzzles whereas the physical simulation converge to a lower value.

5.3.3 Experiment 3: 5 pieces and 5 robots, final puzzles F1 and F2

Setup:

Initial conditions: $X_R = 5, X_1^f = 1, X_2^f = 2, X_3^f = 1, X_4^f = 1, X_i = 0, X_{Fk} = 0$.

Experiment duration: 20 minutes.

Number of experiments (Stochastic simulation only): 100.

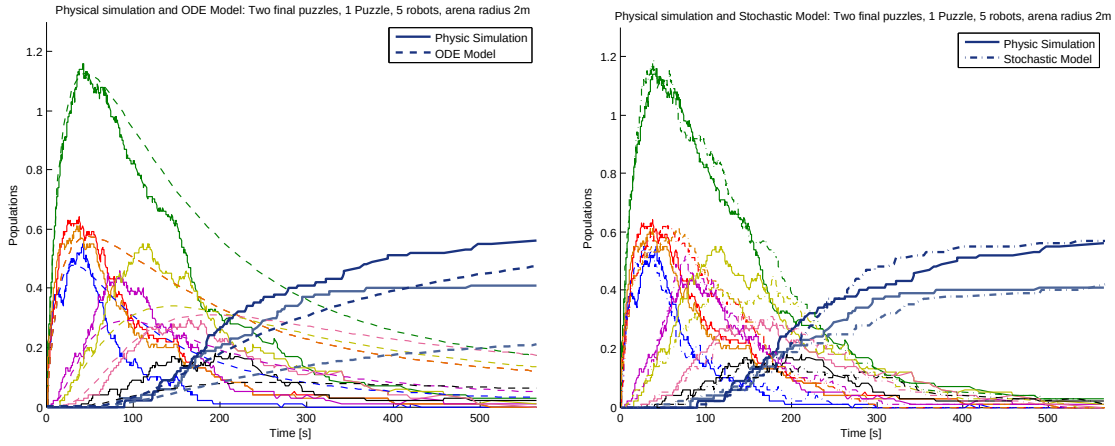
Reaction i	1	2	3	4	5	6
$\mathbf{p_i^a}$	0.9777	0.9074	0.9636	0.9737	0.833	1.0

Table 5.1: Probability of successful assembly for experiment 3. Measured over 100 experiments.

Arena size: $R_a = 2m$.

Probability of successful assembly: Results of the measures are shown in Table 5.1. We see that now, the probabilities are not equal and depend on the assembly step. Especially, the success of reaction 5, to create piece 8, is significantly smaller. p_i^a is set accordingly for each reaction i.

See Figure 5.6 for the results.



(a) ODE simulation vs physical Webots simulation

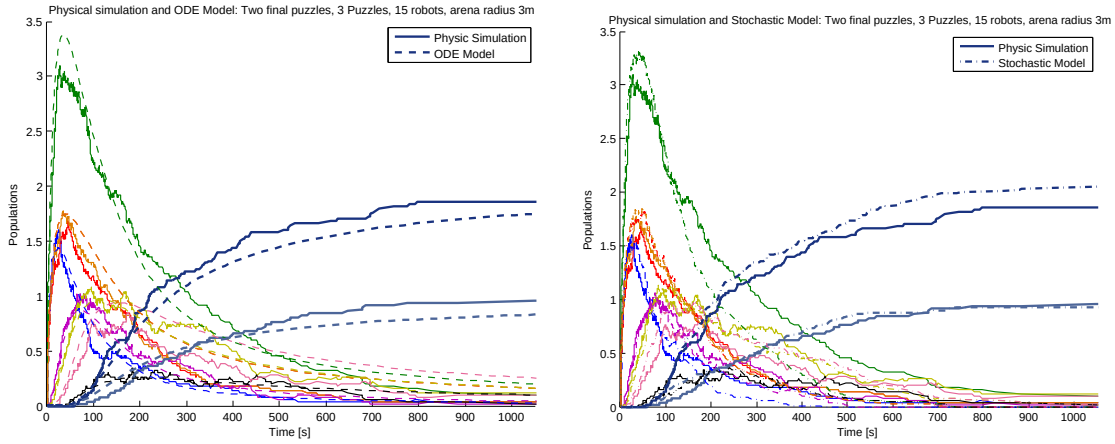
(b) Stochastic simulation vs physical Webots simulation

Figure 5.6: Comparison between the models simulations and the physical Webots simulation for the puzzle test-case, scenario 1, experiment 3.

We see that the ODE model fails to capture the time to convergence, and underestimates the speed of variations. It also seems that they do not converge to the same values, which could be a problem. However, using the Stochastic simulation with the exact same rates, we see that the fitting is much better. The low copy numbers again has a big impact on the capacity to use the ODE model as an approximation.

But the rates we use for the ODE model produce a good fit when using the Stochastic simulation, which shows that our model accurately describes the physical system's dynamics. We thus think that, if the copy number is big enough, the analysis with the ODE model should be sufficient.

We confirm this hypothesis by performing the same two final puzzles experiment, with 15 pieces and 15 robots and a bigger arena (like Experiment 2). The parameters are adapted accordingly. We obtain the results shown in Figure 5.7.



(a) ODE simulation vs physical Webots simulation

(b) Stochastic simulation vs physical Webots simulation

Figure 5.7: Modified experiment 3 with 15 pieces and 15 robots, to show the effect of a larger copy number.

This confirms our assumption that the only problem with the ODE model discrepancy of convergence is the low copy number. In this new scenario, the fitting of the ODE simulation is good and converge to the physical values (Figure 5.7(a)). This tells us that we can indeed use the ODE as a model, but only for a copy number large enough. We will take advantage of that fact in the next chapter.

5.4 Final considerations

The comparison between the models and the Webots physical simulations shows a close fit of our model to the experimental data. It proves that this model is accurate for our test-case and considered problem. The stochastic simulation captures the correct behavior when the number of elements is small, which is in accordance with the assumptions on algorithms differences.

Therefore, we will use directly the model into an optimization process, before mapping back the new model onto the physical simulation. This approach makes sense, as thanks to the accuracy of our model, we can then hope that the physical simulation will behave in the same way. The only constraint is that we need to work with big copy numbers with the ODE, to avoid convergence problems.

Some problems are still present. For example, the initial guess for the encountering probability becomes wrong when the number of robots and pieces grows. The guess relies on a strong non-spatiality constraint, which our physical simulation can not ensure in several cases. Our models do not capture the failures of the physical simulations, which hinder the overall performance and reliability of the assembly process.

This page intentionally left blank.

Chapter 6 Chemical reaction networks control and design

Having successfully modeled our puzzle test-case using a Chemical Reaction Network, we turn to the next step: modifying this model so that it behaves in a desired way.

This modification could be making it more efficient (i.e. optimizing it for a given metric) or just attaining a specific goal (i.e. create a specific final distribution). We define the control problem as follows:

> Let a system of chemical reactions R, defined by a matrix of rate constants K, a stoichiometry matrix S and a vector of products and reactants y. Let the initial condition x_0 be known also. What should be the matrix K_c so that the system converges as close as possible to a desired equilibrium $\hat{X}$. Moreover, can we find the matrix K_f that converge as close as possible to $\hat{X}$ at the fastest rate?

This is a design problem as well as an optimization problem. In the second proposition, we are optimizing for fast convergence subject to a constraint on the equilibrium.

This part has been done in close collaboration with Spring Berman.

6.1 Overview of possibilities

Our first goal was to use a previous optimization scheme used by S. Berman [3, 61], namely optimizing the Mixing-time of a Markov chain [62, 63, 64]. Unfortunately, our problem is not a Markov chain. Our model is a continuous-time Markov process, with nonlinear rates. Actually, it is possible to represent it as a Markov chain, an approach taken by Klavins for his optimization technique [14]: we have to consider every possible macro-states or marking of the system starting from an initial condition, by discretizing all possible values taken by the products, and then creating or measuring the probabilities to jump from one macro-state to another. Obviously, this approach runs into a combinatorial explosion when the number of species and their populations increase, which forbids us from using it. Other approaches using Markov chains [65, 66] are also abandoned because of this problem.

Therefore, we looked into the chemical reaction networks literature for an optimization scheme taking advantage of the control we have on the stochastic constant rates.

We found papers focusing on the reachability of networks, namely knowing what parts of the state space could be attained knowing the system dynamics [67, 68, 69, 70, 71]. These works apply more to bioreactors and show little interest designing the reaction rates and more

on controlling already present reactions. Other works, on the other hand, rely too much on a control theory methodology, for example by adding a feedback to the system [72, 73, 74, 75]. As we want to modify only the rates, these were not helpful.

Eventually, we found an interesting study in [76]. The authors propose an analysis and optimization of the rates of a chain of reactions, with respect to several objectives. Those objectives closely resembles the two propositions we are trying to solve, we will therefore use ideas from their work, although our approach is original. However, they are working with reaction chains, which offers simplifications and results that are not available for our more complex reaction network.

6.2 Methodology

6.2.1 Changes on our models

Starting from the chemical reaction network (5.1), we make several modifications in order to satisfy our methodology:

1. We remove the robots from the system. We assume that the number of robots is bigger or equal to the number of pieces and that the robots find and carry the piece very quickly with respect to the time between assemblies. Under those assumptions, we can remove the robots in a quasi steady-state assumption, looking only at the dynamics of the products.
2. We add backward reactions to every existing assembling reaction. They correspond to the degradation of a product into its reactants. Such reactions are needed when we will prove that our new system has only one globally asymptotically stable equilibrium.

The new chemical reaction network is:

$$\begin{aligned}
X_1 + X_2 &\underset{k_1^-}{\overset{k_1^+}{\rightleftharpoons}} X_5 & \qquad X_2 + X_7 &\underset{k_4^-}{\overset{k_4^+}{\rightleftharpoons}} X_{F1} \\
X_3 + X_4 &\underset{k_2^-}{\overset{k_2^+}{\rightleftharpoons}} X_6 & \qquad X_2 + X_5 &\underset{k_5^-}{\overset{k_5^+}{\rightleftharpoons}} X_8 \\
X_5 + X_6 &\underset{k_3^-}{\overset{k_3^+}{\rightleftharpoons}} X_7 & \qquad X_6 + X_8 &\underset{k_6^-}{\overset{k_6^+}{\rightleftharpoons}} X_{F2}
\end{aligned} \tag{6.1}$$

In the thermodynamic limit [28], the physical system represented by (6.1) can be abstracted to a linear ordinary differential equation (ODE) model:

$$\begin{cases} \dot{x}_1 &= -k_1^+ x_1 x_2 + k_1^- x_5 \\ \dot{x}_2 &= -k_1^+ x_1 x_2 - k_5^+ x_2 x_5 - k_4^+ x_2 x_7 + k_1^- x_5 + k_5^- x_8 + k_4^- x_{F1} \\ \dot{x}_3 &= -k_2^+ x_3 x_4 + k_2^- x_6 \\ \dot{x}_4 &= -k_2^+ x_3 x_4 + k_2^- x_6 \\ \dot{x}_5 &= k_1^+ x_1 x_2 - k_1^- x_5 - k_3^+ x_5 x_6 + k_3^- x_7 - k_5^+ x_2 x_5 + k_5^- x_8 \\ \dot{x}_6 &= k_2^+ x_3 x_4 - k_2^- x_6 - k_3^+ x_5 x_6 + k_3^- x_7 - k_6^+ x_6 x_8 + k_6^- x_{F2} \\ \dot{x}_7 &= k_3^+ x_5 x_6 - k_3^- x_7 - k_4^+ x_2 x_7 + k_4^- x_{F1} \\ \dot{x}_8 &= k_5^+ x_2 x_5 - k_5^- x_8 - k_6^+ x_6 x_8 + k_6^- x_{F2} \\ \dot{x}_{F1} &= k_4^+ x_2 x_7 - k_4^- x_{F1} \\ \dot{x}_{F2} &= k_6^+ x_6 x_8 - k_6^- x_{F2} \end{cases} \tag{6.2}$$

Define a matrix $\mathbf{M} \in \mathbb{R}^{10\times 12}$ in which each entry m_{ji}, $j = 1, ..., 10$, of column $\mathbf{m}_i$ is defined as the coefficient of piece type j in complex i (0 if the piece type is absent). For instance, the column corresponding to the complex $X_1 + X_2$ is $[1\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0]^T$. Now represent the rate associated with reaction $(i, j) \in \mathcal{E}$ as k_{ij} and define a matrix $\mathbf{K} \in \mathbb{R}^{12\times 12}$ with entries:

$$K_{ij} = \begin{cases} k_{ji} & \text{if} \quad i \neq j\ , \quad (j,i) \in \mathcal{E}\ , \\ 0 & \text{if} \quad i \neq j\ , \quad (j,i) \notin \mathcal{E}\ , \\ -\sum_{(i,l)\in\mathcal{E}} k_{il} & \text{if} \quad i = j\ . \end{cases} \tag{6.3}$$

Finally, define a vector $\mathbf{y}(\mathbf{x}) \in \mathbb{R}^{12}$ in which entry y_i is the piece variable or products of variables associated with complex i:

$$\mathbf{y}(\mathbf{x}) = [x_1 x_2 \quad x_5 \quad x_3 x_4 \quad x_6 \quad x_2 x_7 \quad x_{F1} \quad x_5 x_6 \quad x_7 \quad x_2 x_5 \quad x_8 \quad x_6 x_8 \quad x_{F2}]^T\ . \tag{6.4}$$

Then the ODE model (6.2) can be written in the following form [77]:

$$\dot{\mathbf{x}} = \mathbf{M}\mathbf{K}\mathbf{y}(\mathbf{x})\ , \tag{6.5}$$

One set of conservation constraints on the piece quantities is:

$$\begin{cases} x_3 - x_4 &= N_1 \\ x_1 + x_5 + x_7 + x_8 + x_{F1} + x_{F2} &= N_2 \\ x_2 + x_5 + x_7 + 2(x_8 + x_{F1} + x_{F2}) &= N_3 \\ x_3 + x_6 + x_7 + x_{F1} + x_{F2} &= N_4 \end{cases} \tag{6.6}$$

where N_i, $i = 1, ..., 4$, are computed from the initial piece quantities.

It should be possible to find the equilibrium points as closed form formulas, but our attempts in that direction were unsuccessful. We thus propose another approach.

6.2.2 Approach

Our optimization relies on the following hypotheses:

1. The system (6.5) is deterministic and its dynamics depend only on the rates $\mathbf{K}$ and the initial conditions.

2. The system, while nonlinear in x_i, converge to an unique positive equilibrium given an initial condition and a matrix $\mathbf{K}$.
3. The system (6.5) is nonlinear in x_i but linear in k_i.
4. $\mathbf{K}$ governs the rate of convergence of the system.
5. $\mathbf{MKy}(\mathbf{x}) = 0$ only at the equilibrium point.

Hypothesis 1 derives from the definition of an ODE and on the fact that we keep the network fixed during the system evolution. Hypothesis 2 will be proved in Section 6.2.3, it is a main result in the success of our approach. Hypothesis 3 comes from the observation of the system (6.2). Hypothesis 4 will be discussed in Section 6.2.4. Hypothesis 5 is discussed in Section 6.2.4.

Then, under those hypothesis, it is possible to find an objective function linear in k_i, using $\mathbf{MKy}(\mathbf{x^d}) = 0$ as a set of constraints on the desired converged values of x^d and solve it using a convex optimization algorithm.

The complete derivation and discussion is presented in Section 6.2.4.

6.2.3 Convergence of chemical reaction networks

Theorem 1. *System* (6.2) *subject to* (6.6) *has a unique equilibrium* $\bar{\mathbf{x}} > \mathbf{0}$.

Proof. Each steady state of the system can be classified as either a *positive* steady state, $\{\bar{\mathbf{x}} \mid \mathbf{MKy}(\bar{\mathbf{x}}) = \mathbf{0},\ \bar{x}_i > 0\ \forall i\}$, or a *boundary* steady state, $\{\bar{\mathbf{x}} \mid \mathbf{MKy}(\bar{\mathbf{x}}) = \mathbf{0},\ \bar{x}_i = 0 \text{ for some } i\}$, which can be found by solving $\mathbf{y}(\bar{\mathbf{x}}) = \mathbf{0}$ [77].

From definition (6.4) of $\mathbf{y}(\mathbf{x})$, the boundary steady states for the system satisfy

$$\begin{aligned} \bar{x}_5 = \bar{x}_6 = \bar{x}_7 = \bar{x}_8 = \bar{x}_{F1} = \bar{x}_{F2} = 0\ , \\ \bar{x}_1\bar{x}_2 = \bar{x}_3\bar{x}_4 = \bar{x}_2\bar{x}_7 = \bar{x}_5\bar{x}_6 = \bar{x}_2\bar{x}_5 = \bar{x}_6\bar{x}_8 = 0\ . \end{aligned} \tag{6.7}$$

It can be concluded that in each boundary steady state, all $x_i = 0$ except for one of the four combinations of variables $(x_1, x_3), (x_1, x_4), (x_2, x_3), (x_2, x_4)$. Since we only consider systems that can produce x_{F1} and x_{F2}, it is not possible for the system to reach any of these steady states; each one lacks two piece types needed for the final assemblies.

The *deficiency* δ of a reaction network has been introduced as a measure giving insights and proofs on the stability and the existence of equilibrium of a class of reaction networks [78, 79, 80, 81, 82, 83, 84]. It is one of the two attempts at proving convergence and stability of chemical reaction network, the other one is based on so-called species-reaction graphs analysis [85, 86, 87, 88, 89].

δ is calculated has the number of complexes minus the number of *linkage classes*, each of which is a set of complexes connected by reactions, minus the network *rank*, which is the rank of the matrix whose rows are each obtained by subtracting a column of $\mathbf{M}$ associated with a reactant in a particular reaction from a column associated with a product [78]. It can be shown that each of the six linkage classes in our system has rank 1 and the rank of the entire network is 6; from this the deficiency of each linkage class is calculated to be 0

and the deficiency of the entire network is 0 as well. Also, we observe that no complex in any linkage class transforms into a complex in another linkage class. These properties, along with the fact that the system kinetics are mass-action, satisfy the criteria for applying the Deficiency One Theorem in [78], which gives the result that the system can have no more than one positive steady state.

Now we must show that there exists a positive steady state, which would be unique by the Deficiency One result. Since the network rank of the system is the sum of the ranks of the linkage classes, the linkage classes constitute a *direct partition* of the network [81]. Also, since each linkage class has deficiency 0 and is *weakly reversible*, meaning that there is a reaction pathway connecting each pair of complexes, each linkage class contains exactly one positive steady state by the Deficiency Zero Theorem [78]. By Lemma 8.2.3 of [81], these properties imply that the system admits a positive steady state.

□

We still lack a proof on the stability of this unique equilibrium. If the deficiency zero theorem [78] can be extended to support not only *weakly reversible* but also *block weakly reversible* networks, then we would get such a result. A remark in [88] support that fact, but we did not manage to get a formal proof as of today. However, the empirical experiments we are doing show that the equilibrium is stable, so we will assume this fact from now on.

6.2.4 Design of optimal rates

We consider the problem of designing the assembly system described by model (6.2) subject to (6.6) to produce desired quantities of pieces. The derivation is similar for other systems.

System and equilibrium definition

The assembly system will be most productive if it yields the target quantities as quickly as possible. This objective will be posed as the design of a set of *optimal rates* k_i^+, k_i^-, $i = 1, ..., 6$, for model (6.2) that minimizes the convergence time of the resulting system to a target vector of piece quantities, $\mathbf{x^d}$.

Note that although the quantities of only the final assemblies, X_{F1} and X_{F2}, may need to be specified in practice, the optimization problem requires that target quantities of intermediate components be defined as well, as will be discussed later in this section.

The rates k_i^+, k_i^- can be chosen so that the assembly system yields a target piece distribution $\mathbf{x^d} > \mathbf{0}$ starting from *any* initial distribution of pieces:

1. First, specify target numbers of the intermediate and final assemblies, x_i^d, $i = 5, ..., 8$.
2. Then define piece quantities x_j^d, $j = 1, ..., 4$, in terms of these numbers according to the conservation equations (6.6).
3. Finally define $\mathbf{y^d} = \mathbf{y}(\mathbf{x^d})$ according to definition (6.21).

This vector y^d represents the steady-state quantities of the pieces and piece products associated with each complex. If the system described in the form (6.5) converges to $\mathbf{y^d}$, then it converges to the target quantities x_i^d, $i = 5, ..., 8$, since eight components of $\mathbf{y^d}$ are

equal to these variables. Convergence to x_j^d, $j = 1, ..., 4$ is also ensured since each of these quantities is a function of x_i^d, $i = 5, ..., 8$ by conservation laws.

By Theorem 1, model (6.2) always converges to a single positive equilibrium $\bar{\mathbf{y}}$. Therefore, system convergence to $\mathbf{y^d}$ can be guaranteed by specifying that $\bar{\mathbf{y}} \equiv \mathbf{y^d}$ through the following constraint on the rate matrix $\mathbf{K}$ in model (6.5):

$$\mathbf{MKy^d} = \mathbf{0} \ . \tag{6.8}$$

To control more easily the goal in term of final puzzles, we introduce the fraction α:

$$\alpha = \frac{x_{F1}}{x_{F1} + x_{F2}} \tag{6.9}$$

The target final assemblies can be defined in such a way that their sum, $x_{F1}+x_{F2}$, is constant and the fraction α is directly a tunable parameter. The sum $x_{F1} + x_{F2}$ is expressed in terms of a conservation law, which means that one of the target quantities x_i^d, $i \in \{1, ..., 4\}$, must be implicitly defined. For instance, using the third conservation equation in (6.6) and a target value for x_2^d:

$$x_{F1}^d = \tfrac{1}{2}\ \alpha\ \left(N_2 - (x_2^d + x_5^d + x_7^d + 2x_8^d)\right) \tag{6.10}$$

$$x_{F2}^d = \tfrac{1}{2}\ (1-\alpha)\ \left(N_2 - (x_2^d + x_5^d + x_7^d + 2x_8^d)\right) \ , \tag{6.11}$$

Whatever the selection of independent piece quantities, it must be ensured that the remaining piece quantities are positive to have a feasible $\mathbf{x^d}$ (positive equilibrium).

Convergence time

Now we consider the aspect of minimizing the convergence time of the system to the target equilibrium $\mathbf{x^d}$. We can define measures of this time by reformulating the system in terms of new variables. Define v_j, $j = 1, ..., 6$, as the difference between the forward and reverse fluxes associated with reaction j in system (6.1).

For instance, if we label reaction 1 as the one in which $X_1 + X_2$ is the reactant and X_5 is the product, then $v_1 = k_1 x_1 x_2 - k_2 x_5$. Let $\mathbf{v}(\mathbf{x}) = [v_1\ ...\ v_6]^T$. Denote the stoichiometric matrix of the system by $\mathbf{S} \in \mathbb{R}^{10\times 6}$, which is defined such that model (6.2) can be written as [90]:

$$\dot{\mathbf{x}} = \mathbf{S}\mathbf{v}(\mathbf{x}) \ . \tag{6.12}$$

Our assembly system is similar to a model of a biochemical network with mass action kinetics. The dynamical properties of such networks are often analyzed by linearizing the ODE model of the system about a steady state and studying the properties of the associated Jacobian matrix $\mathbf{J} = \mathbf{SG}$, where the entries of $\mathbf{G}$ are $g_{ij} = dv_i/dx_j$ (see [91] for an overview and further references).

Denoting the eigenvalues of $\mathbf{J}$ by λ_i, $\tau_i = 1/|Re(\lambda_i)|$ are measures of the characteristic times, referred to as *relaxation times*, in which different modes (dynamically independent variables) of the the system converge to a stable steady state after perturbation [92, 91].

Since the λ_i are negative at a stable steady state, one way to yield fast convergence is to choose rates k_i that minimize the largest negative λ_i, as was done for a linear chain of enzymatic reactions in [76].

However, in our system it is very difficult to find analytical expressions for the λ_i, so we must explore other ways of quantifying the convergence time.

Various measures of average relaxation times in biochemical networks have been defined, but they are applicable only under certain conditions, such as a linear reaction sequence [93]. For instance, one such measure was minimized in the optimization of rates for the linear chain in [76]. For our system, we can use a general estimate of the relaxation time for each reaction j that is given in [90]. It is derived by linearizing the system model around its equilibrium point, which in our case is the target distribution $\mathbf{x^d}$ enforced by equation (6.8):

$$\tau_j = \left(\sum_{i=1}^{12} (-s_{ij}) \frac{dv_j}{dx_i} \right)^{-1} \tag{6.13}$$

This expression is evaluated at $\mathbf{x^d}$. Since each reaction j in system (6.1) is of the form $X_k + X_l \rightleftharpoons_{k_j^-}^{k_j^+} X_m$, the net flux v_j is $k_j^+ x_k x_l - k_j^- x_m$, and the entries of column j in $\mathbf{S}$ are all 0 except for $s_{kj} = s_{lj} = -1$ and $s_{mj} = 1$. Thus, taking advantage of this fact and applying equation (6.13) at $\mathbf{x^d}$, we get that the relaxation time for each reaction is

$$\tau_j = (k_j^+ (x_k^d + x_l^d) + k_j^-)^{-1} \ . \tag{6.14}$$

Which gives us a measure of speed of convergence.

Objective functions

Define $\mathbf{k} \in \mathbb{R}^{12}$ as the vector of all rates k_i^+, k_i^-. Using characterization (6.14) of reaction relaxation times, we define two possible objective functions $f : \mathbb{R}^{12} \to \mathbb{R}$ to *maximize* in order to produce fast convergence to $\mathbf{x^d}$.

The first is the average inverse relaxation time,

$$f_{ave}(\mathbf{k}) = \tfrac{1}{10} \sum_{j=1}^{10} \tau_j^{-1} \ . \tag{6.15}$$

The second is the minimum inverse relaxation time,

$$f_{min}(\mathbf{k}) = \min\{\tau_1^{-1}, \ldots, \tau_{10}^{-1}\} \ . \tag{6.16}$$

Convex program definition

Finally, we map the rates onto actual adjustable parameters of the physical assembly system. Those adjustable parameters will be defined as the optimization variables.

Backward rates Each reverse rate k_j^- will be defined simply as a probability per unit time of breaking up an assembly, $p_j^- \in \{0, 1\}$, which can be adjusted:

$$k_j^- = p_j^- \tag{6.17}$$

Forward rates We use the formula (5.3) defined in Section 5.2.1. We extend it by adding a probability of starting the assembly $p_j^+ \in \{0, 1\}$ which can be adjusted:

$$k_j^+ = p_j^e \cdot p_j^a \cdot p_j^+ \tag{6.18}$$

p_j^e is the encountering probability defined by (5.4), dependent on some parameters and p_j^a is the assembly success rate, which we measure as explained in Section 5.3.

Let $\mathbf{p} \in \mathbb{R}^{12}$ be the vector of all adjustable probabilities, p_j^+ and p_j^-. Then we define an optimization Problem P as follows:

$$\begin{aligned} \textbf{P:} \quad & \text{maximize} \quad f(\mathbf{k}(\mathbf{p})) \\ & \text{subject to} \quad \mathbf{MK}(\mathbf{p})\mathbf{y^d} = \mathbf{0}, \quad \mathbf{0} \leq \mathbf{p} \leq \mathbf{1} \ . \end{aligned}$$

Depending on the objective function used, we define Problem P1 and Problem P2:

$$\begin{aligned} \textbf{P1:} \quad & \text{maximize} \quad f_{ave}(\mathbf{k}(\mathbf{p})) \\ & \text{subject to} \quad \mathbf{MK}(\mathbf{p})\mathbf{y^d} = \mathbf{0}, \quad \mathbf{0} \leq \mathbf{p} \leq \mathbf{1} \ . \end{aligned}$$

$$\begin{aligned} \textbf{P2:} \quad & \text{maximize} \quad f_{min}(\mathbf{k}(\mathbf{p})) \\ & \text{subject to} \quad \mathbf{MK}(\mathbf{p})\mathbf{y^d} = \mathbf{0}, \quad \mathbf{0} \leq \mathbf{p} \leq \mathbf{1} \ . \end{aligned}$$

Since this problem can be formulated as the minimization of a linear combination of the entries of $\mathbf{p}$ subject to a set of linear equality and inequality constraints on $\mathbf{p}$, it is a linear program, which can be solved efficiently.

6.2.5 Optimization implementation

In order to optimize the convex programs P1 and P2, we use a framework for Matlab, YALMIP[94]. It allows us to define our optimization variables, objective function and constraints and to apply a semidefinite programming solving algorithm, running in polynomial time [94].

6.3 Results and limitations

We start with problem P1, which uses the first objective function f_{ave}, optimizing the average inverse relaxation time. We take the system (6.1) and optimize it according to our methodology:

- As discussed in Section 5.3.3, and because our optimization rely on an underlying ODE approximation, we use an initial condition with a big copy number of pieces to avoid convergence problems. We use 60 pieces {1, 3, 4, 5}, 120 pieces 2 and no other mid-assemblies.
- For the forward rates, we assume that we can only optimize the probability of starting an assembly, p_j^+. This probability comes in addition to the probability of encounter and the probability of successful assembly, as shown in Equation (6.18). It can take values from 0 to 1.
- For the backward rates, we optimize the probability of breaking up an assembly p_j^-, as Equation (6.17). It can take values from 0 to 1.
- Our goal is to control the ratio α between the final number of F1 and F2 puzzles, Equation (6.9). The converged number of other pieces are set to the smaller values still consistent with conservation laws and non-negativity constraints.

We optimize the system for $\alpha \in \{0.01, 0.02, 0.03, \ldots, 0.99\}$ and store the obtained rates for all 6 reactions.

It turns out that nearly all rates keep constant values. All the forward rates are put to their maximal values. Backward rates all take constant positive values, except the backward rates of reactions 4 and 6, which vary continuously with respect to α. Table 6.1 displays the optimized probabilities.

Reaction j	**1**	**2**	**3**	**4**	**5**	**6**
Optimized $\mathbf{p_j^+}$	1.0					
Optimized $\mathbf{p_j^-}$	0.01885	0.00754	0.00377	*continuous*	0.00942	*continuous*

Table 6.1: Values of optimized rates for varying α, under objective function P1 for system (6.1). *Continuous* rates evolve continuously with respect to α.

The case of the continuously varying rates is very interesting. Those rates corresponds to the backward reaction coming from the two final puzzles F1 (reaction 4) and F2 (reaction 6). Their evolution with respect to α is shown in Figure 6.1.

Having constant rates for every reaction except those two give several informations:

- The network is highly flexible, allowing every ratio of final puzzles only by changing two specific backward rates. The non final reactions are kept to an activity low enough so as not to destroy to yield, but still high enough to be able to redistribute the materials when the final reactions are breaking up the final puzzles to converge to a desired ratio. Controlling only the disassembly rate of the two final puzzles is enough to attain any ratio α, while optimizing the yield and time to convergence.
- All the forward rates are put to their maximum value. This could show that using non-maximum forward rates decreases the convergence rate, but further tests about that hypothesis are necessary. In an informal discussion, G. Mermoud confirmed similar results in previous works of his with self-assembly processes.

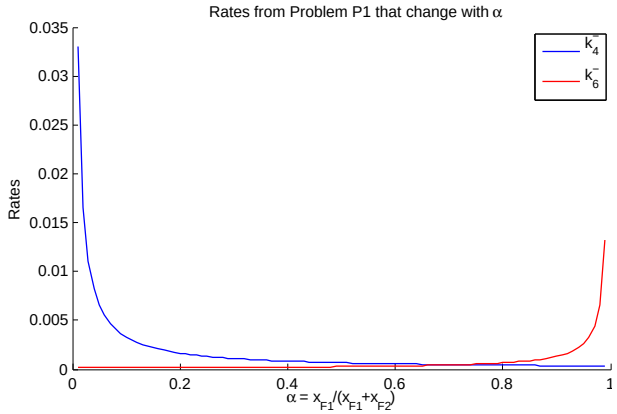


Figure 6.1: Backward rates changing continuously with respect to α under objective function P1 for system (6.1).

Doing the optimization of problem P2, under objective function f_{min} leads to slightly different results. This time the forward rates and backwards rates of reactions 4 and 6 are varying continuously. The forwards rate are not all maximum. Table 6.2 presents the results for P2.

Reaction j	**1**	**2**	**3**	**4**	**5**	**6**
Optimized $\mathbf{p_j^+}$	0.36	0.666	1.0	*continuous*	0.4705	*continuous*
Optimized $\mathbf{p_j^-}$	0.006855	0.005027	0.00377	*continuous*	0.00443	*continuous*

Table 6.2: Values of optimized rates for varying α, under objective function P2 for system (6.1). *Continuous* rates evolve continuously with respect to α.

Again with that objective function, the real control of the ratio α is carried on by the last two reactions in the construction of the final assemblies. This time the forward rates have a stronger role in that, but this has yet to be verified that this method is better. See Figure 6.2 for the continuous variation of rates with respect to α.

6.3.1 Comparison between objective functions and strategies

To compare the two functions, we plot the time-evolution of the ratio of the final puzzles F1 and F2 for three different α: 0.1, 0.5 and 0.9, see Figure 6.3 a), c) and e). The horizontal dotted lines show the target ratios and both objective function are shown (plain for P1, dashed for P2).

We see that P1 is converging quicker than P2 to the equilibrium for $\alpha = 0.1$ and $\alpha = 0.9$, but is slower for $\alpha = 0.5$. It is still unclear if it is only due to the values of the forward rates, as hypothesized. The overall speed of convergence is actually pretty slow, especially when trying to produce 50% of both puzzles. This could be problematic, but is in fact linked to the global behavior of the system, as we show now.

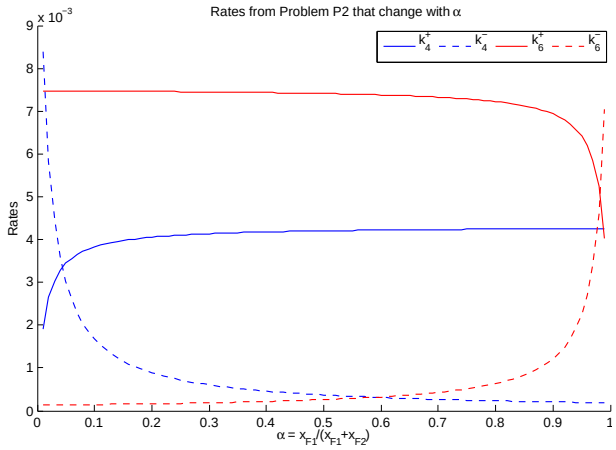


Figure 6.2: Forward and backward rates changing continuously with respect to α under objective function P2 for system (6.1).

Looking at semi-logarithmic plots of the same data, Figure 6.3 b), d) and f), we clearly see two regime of convergence:

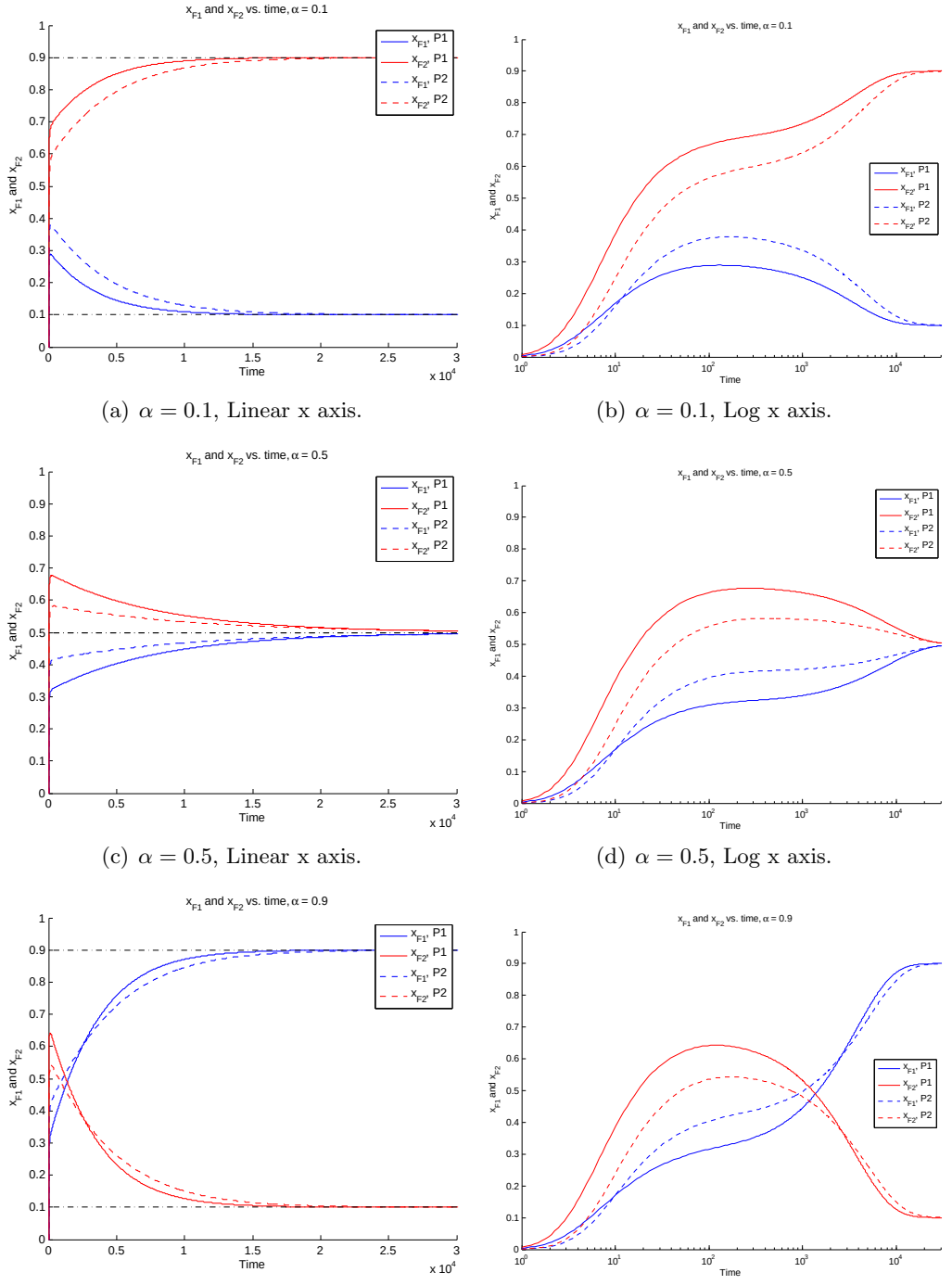
- A quick convergence to an unique equilibria, independent of the chosen α, till $t = 10^2 s$. We hypothesize that this equilibrium depends on the network topology and forward rates.
- A "redistribution" regime starting from there, which makes the system converge to the final α desired. This regime is much slower than the first one. The final reactions with optimized rates only act during that regime.

If we think in the state space of final assemblies, it means that the system first converge quickly to its intrinsic equilibrium, and then moves around slowly due to the redistribution of final puzzles until it arrives to the desired equilibrium.

This is a very interesting and quite counterintuitive result, as one might think that it would be best to go directly towards the desired equilibrium. The optimization process we use seems to point out that going quickly to the intrinsic equilibrium and then moving from there is a better strategy. Of course, this might be an artifact of the linear optimization we do, so a verification with another optimization process should be performed. This will be done in further works.

6.3.2 Online adaptation of the desired final puzzles ratio

We finish by showing the application of the rate optimization to the "green manufacturing" process shortly presented in Section 4.1. A "green manufacturing" process is a direct application of the flexibility offered by a non-specific assembly task. In that process, we reuse finished products (in our case, final puzzles) in order to create new products, depending on the current demand. We "recycle" the products into new ones. This is possible because



(a) $\alpha = 0.1$, Linear x axis.

(b) $\alpha = 0.1$, Log x axis.

(c) $\alpha = 0.5$, Linear x axis.

(d) $\alpha = 0.5$, Log x axis.

(e) $\alpha = 0.9$, Linear x axis.

(f) $\alpha = 0.9$, Log x axis.

Figure 6.3: Comparison of convergence of final assemblies over time after optimizing P1 and P2. Time unit is seconds.

our system possess the capabilities to create both products, and because we fix what it is supposed to create by optimizing the rates of assembly as shown in this chapter.

We show such an example of online adaptation to a new goal, by simulating the following experiment:

- Through all this experiment, we use the optimized goals under objective function P1, see Table 6.1 and Figure 6.1.
- We initialize the system with a first goal of a 60% ratio of final puzzle F2 ($\alpha = 0.4$). We let the system run till $t_1 = 1000s$.
- We change the rates to the one optimized to create a ratio of 99% final puzzle F1 ($\alpha = 0.99$). We let the system run till $t_2 = 6000s$.
- We change the rates to a goal of 99% of final puzzle F2 ($\alpha = 0.01$). We let the system run till $t_3 = 11000s$.
- We change for the last time the rates for a goal of 50% of each final puzzle ($\alpha = 0.5$). We let the system run till $t_4 = 21000s$.

The result is shown in Figure 6.4. We see that the system is capable of adapting smoothly to abrupt new commands in the desired targets. It attains the desired ratio when converged. The convergence time is pretty slow when approaching the desired ratio, yet the disruption of the previous steady state occurs quickly. The target of 50% of both puzzles is slower to attain.

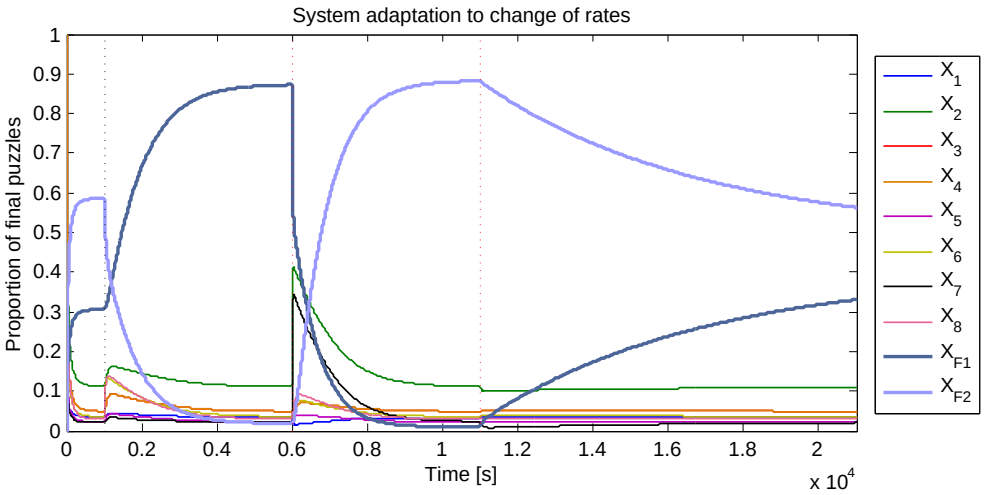


Figure 6.4: Change of reaction rates during an experiment. System adapts smoothly to the new equilibrium. Rates are changed at the times indicated by the dotted vertical lines. First goal is 60% of F2, second is 100% of F1, third is 100% of F2 and fourth is 50% of each.

A good result is the speed of adaptation to new commands, and the strong redistribution effect of commands asking for near 100% ratios. We think it is also possible to design a control policy stabilizing quicker to a slow target (50% for example) by switching between

a 100% F1 and a 100% F2 command, slowly damping the oscillations till we attain the desired distribution. This is a similar approach to the one shown in [95, 96] for the control of switching in *E. Coli*. More tests are still necessary to assess the validity of this process, but the behavior we obtain already offers us a large variety of adaptable behaviors.

6.4 Beyond control, direct optimization of the plan?

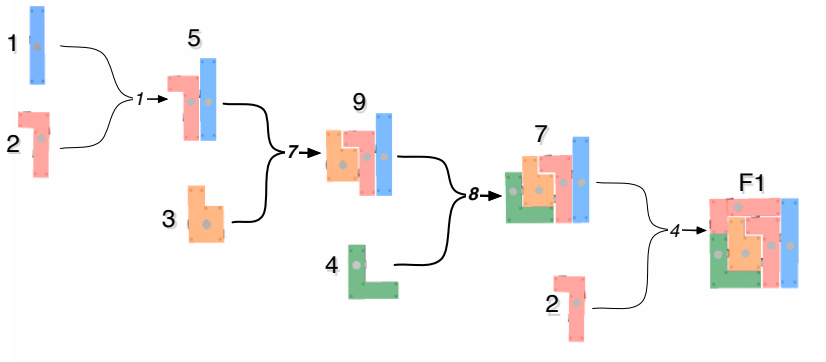
Can our optimization scheme optimize directly a set of assembly plans? We addressed that problem in Section 3.3, while speaking about the optimal plans. In his work, Klavins [14] constructs an optimal assembly plan using graph grammars. This is a discrete optimization, i.e. choosing what reactions to use to build an optimal plan. On the other hand, given optimized rates, it is possible to see if some reactions are promoted or deactivated. This corresponds to a continuous optimization of the assembly plan, effective while the system is behaving. We think our optimization is able to perform such a continuous optimization. We will alter our model in order to test that hypothesis.

We add new assembly steps to create additional pathways to the final assemblies. In addition to plans shown in Figures 4.3, we add two new plans, that reuse parts of the old ones, see Figures 6.5. The new assembly steps are shown in boldface. We call those new plans "sequential plans", as they assemble one piece at a time without parallel processes. Our goal is to see how the algorithm treats these new pathways, more precisely if it "shuts down" the ones which are not optimal or useful. Remark that we do not consider the case of all possible types of reactions in the system, but only a selected sub-set of them. Using the complete set for the optimization of the plan is let for further work.

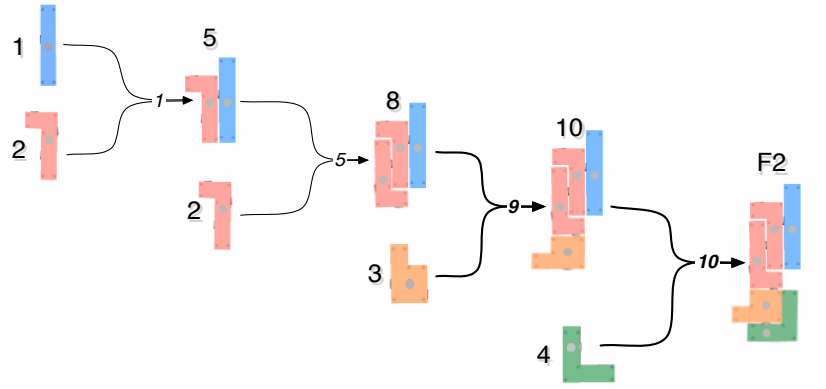
The 4 added assembly steps translate into the following reactions to be added to the set (6.1):

$$\begin{aligned} X_3 + X_5 &\underset{k_7^-}{\overset{k_7^+}{\rightleftharpoons}} X_9 \qquad & X_3 + X_8 &\underset{k_9^-}{\overset{k_9^+}{\rightleftharpoons}} X_{10} \\ X_4 + X_9 &\underset{k_8^-}{\overset{k_8^+}{\rightleftharpoons}} X_7 \qquad & X_4 + X_{10} &\underset{k_{10}^-}{\overset{k_{10}^+}{\rightleftharpoons}} X_{F2} \end{aligned} \tag{6.19}$$

With these added reactions, the ODE equations for the system are:



(a) New added first final puzzle plan



(b) New added second final puzzle plan

Figure 6.5: New plans created by adding 4 new assembly steps, written in boldface. We call those plans the "sequential plans", as they act by assembling one piece after another without parallel processes.

$$
\begin{cases}
\dot{x}_1 &= -k_1^+ x_1 x_2 + k_1^- x_5 \\
\dot{x}_2 &= -k_1^+ x_1 x_2 - k_5^+ x_2 x_5 - k_4^+ x_2 x_7 + k_1^- x_5 + k_5^- x_8 + k_4^- x_{F1} \\
\dot{x}_3 &= -k_2^+ x_3 x_4 + k_2^- x_6 - k_7^+ x_3 x_5 + k_7^- x_9 - k_9^+ x_3 x_8 + k_9^- x_{10} \\
\dot{x}_4 &= -k_2^+ x_3 x_4 + k_2^- x_6 - k_8^+ x_4 x_9 + k_8^- x_7 - k_{10}^+ x_4 x_{10} + k_{10}^- x_{F2} \\
\dot{x}_5 &= k_1^+ x_1 x_2 - k_1^- x_5 - k_3^+ x_5 x_6 + k_3^- x_7 - k_5^+ x_2 x_5 \\
&\quad + k_5^- x_8 - k_7^+ x_3 x_5 + k_7^- x_9 \\
\dot{x}_6 &= k_2^+ x_3 x_4 - k_2^- x_6 - k_3^+ x_5 x_6 + k_3^- x_7 - k_6^+ x_6 x_8 + k_6^- x_{F2} \\
\dot{x}_7 &= k_3^+ x_5 x_6 - k_3^- x_7 - k_4^+ x_2 x_7 + k_4^- x_{F1} + k_8^+ x_4 x_9 - k_8^- x_7 \\
\dot{x}_8 &= k_5^+ x_2 x_5 - k_5^- x_8 - k_6^+ x_6 x_8 + k_6^- x_{F2} - k_9^+ x_3 x_8 + k_9^- x_{10} \\
\dot{x}_9 &= k_7^+ x_3 x_5 - k_7^- x_9 - k_8^+ x_4 x_9 + k_8^- x_7 \\
\dot{x}_{10} &= k_9^+ x_3 x_8 - k_9^- x_{10} - k_{10}^+ x_4 x_{10} + k_{10}^- x_{F2} \\
\dot{x}_{F1} &= k_4^+ x_2 x_7 - k_4^- x_{F1} \\
\dot{x}_{F2} &= k_6^+ x_6 x_8 - k_6^- x_{F2} + k_{10}^+ x_4 x_{10} - k_{10}^- x_{F2}
\end{cases}
\tag{6.20}
$$

Like system (6.2), this system can be written in the form (6.5). In this case, the vector of complexes is defined as

$$\begin{aligned}\mathbf{y}(\mathbf{x}) = \quad & [x_1x_2,\ x_5,\ x_3x_4,\ x_6,\ x_2x_7,\ x_{F1},\ x_5x_6,\ x_7,\ x_2x_5,\\ & x_8,\ x_6x_8,\ x_{F2},\ x_3x_5,\ x_9,\ x_4x_9,\ x_3x_8,\ x_{10},\ x_4x_{10}]^T\ . \end{aligned} \tag{6.21}$$

One set of conservation constraints on the piece quantities in this system is:

$$\begin{cases} x_1 + x_5 + x_7 + x_8 + x_{F1} + x_{F2} + x_9 + x_{10} &= N_5 \\ x_2 + x_5 + x_7 + 2x_8 + 2x_{F1} + 2x_{F2} + x_9 + 2x_{10} &= N_6 \\ x_3 + x_6 + x_7 + x_{F1} + x_{F2} + x_9 + x_{10} &= N_7 \\ x_4 + x_6 + x_7 + x_{F1} + x_{F2} &= N_8 \end{cases} \tag{6.22}$$

where N_i, $i = 5, ..., 8$, are computed from the initial piece quantities.

6.4.1 Optimized rates and induced effective plan

We apply our optimization procedure with Problem P1 and P2 as before. The results for the convergence are shown in Figure 6.6. As Problem P2 is quicker in general and shows more interest dynamics, we will only study it for the rest of this section.

The resulting probabilities p_i^+, p_i^- for problem P2 are nearly all continuously varying. The only exceptions are p_3^+, p_6^+ and p_9^+ which are all at 1.0. The variation of the rates for the values of α are shown in Figure 6.7.

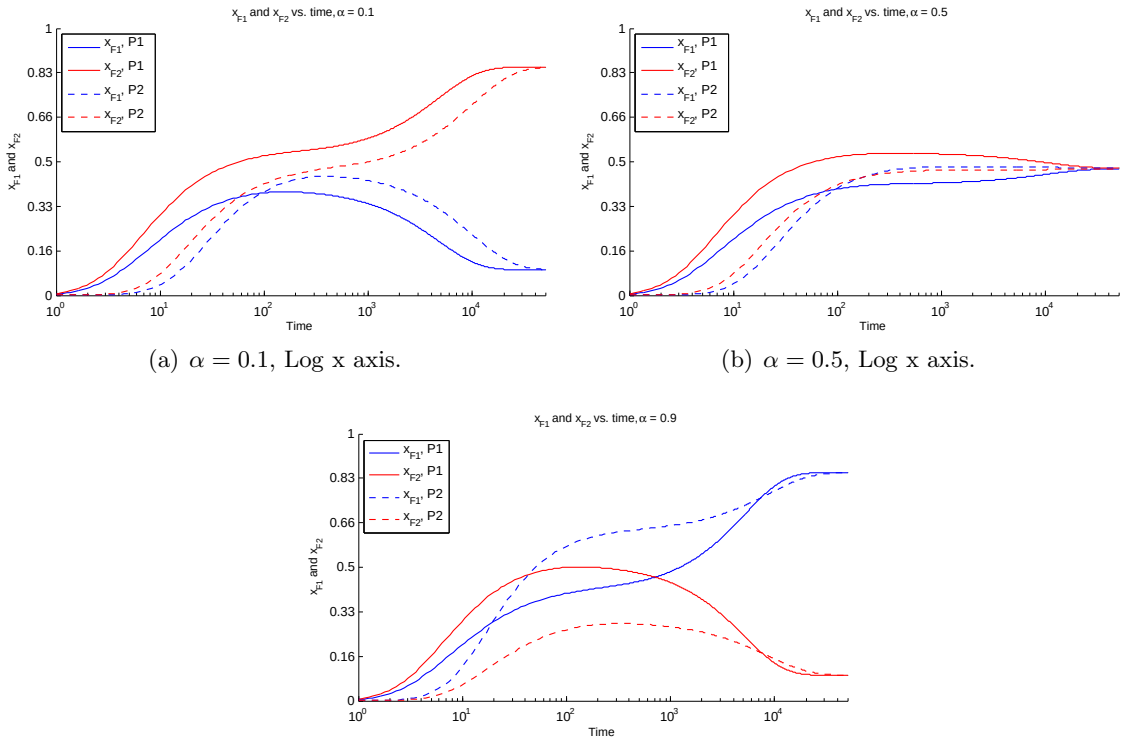
Looking at the values of the rates at specific positions, we can make observation on the actual plan promoted by the optimization. The reactions, though still present, will be promoted or deactivated when their corresponding forward and backward rates are modified. This corresponds to a continuous optimization of the plan, in contrast with a discrete optimization performed by Klavins[14].

In general, the value of the rates show the following hierarchy between reactions and pathways:

- k_2^+ is small and k_2^- is big compared to other rates. This deactivates Reaction 2.
- To compensate the absence of piece 6, the sequential pathways represented by reactions 7 and 8 for F1 and reactions 9 and 10 for F2 are really active.
- The small amount of pieces 6 that would result from breaking of assemblies are quickly reuse via the reactions 3 and 6, which have of full forward rate, or destroyed by the backward rate of reaction 2.

We see three different regimes for the rates, and the resulting continuous assembly plans:

1. Near $\alpha = 0.1$, the only changing rates are k_4^+ and k_4^-. They are changing toward a deactivation of Reaction 4, which is the only one creating F1 puzzles. The network then automatically tends to create more F2 puzzles, which seems to be a constant characteristic of the system we are studying.



(a) $\alpha = 0.1$, Log x axis.

(b) $\alpha = 0.5$, Log x axis.

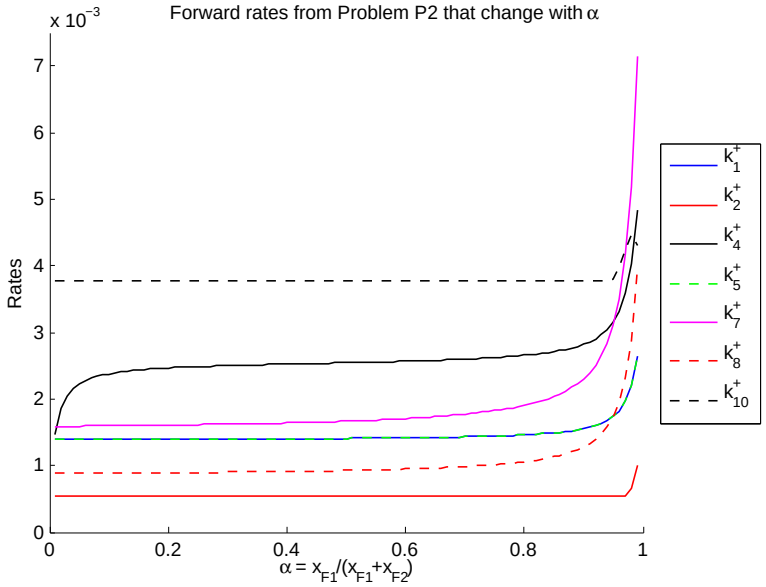
(c) $\alpha = 0.9$, Log x axis.

Figure 6.6: Expanded system. Comparison between the two objective functions P1 and P2 when showed with semi-logarithmic x axis.

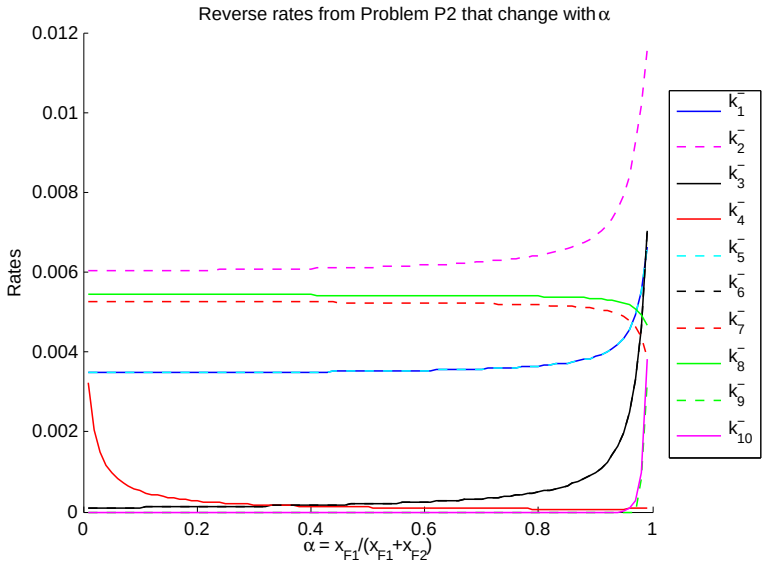
2. Between $\alpha = 0.2$ and $\alpha = 0.8$, the rates k_4^+ and k_6^- seems to have to most effect. They act by promoting the creation of F1 and slowing the creation of F2. During that period, the networks tends to produce a 50% ratio by itself, as shown by the convergence profile in Figure 6.6(b): the first convergence arrives close to a 50%, no redistribution is needed (to compare with Figure 6.3(d) of the previous system)
3. Near $\alpha = 0.9$, the behavior changes dramatically. Reactions 7, 8 and 4, leading to the creation of F1 are promoted extensively (increase in forward rate, decrease in backward rate). At the same moment, reactions 6 and 10, creating F2, are deactivated. This promotes the creation of F1 puzzles.

We see then that this optimization works with these general tendencies:

- Remove assembly 6 from the possible pool of pieces, to liberate all initial pieces for other reactions. This means removing the initial plans we were using an use the new introduced one, see Figure 6.5
- Use the sequential steps to create the final puzzles. This is a counter-intuitive results, but it may be linked to an added flexibility with having a large pool of initial pieces,



(a) Forward rates varying continuously



(b) Backward rates varying continuously

Figure 6.7: Optimized rates of expanded system under problem P2, for $\alpha \in 0.01, 0.99$. Remark: k_6^- curve is the same as k_3^-.

that can be used at precise assembly steps without temporal dependences between reactions.

- Modify the reactions leading to the final assemblies to change the ratio, and do not touch the "general" part of the system. This is in agreement with the behavior we observed in last section.

- As the system is more prone to create F2 than F1, it is easier to control the system in one direction than the other. To create a big ratio of F1, the overall behavior of the system is changed, leading to a nearly new plan. This plan correspond to the sequential one we added in Figure 6.5(a).

So in conclusion, it is possible to get insights into the kind of behaviors and continuous plans promoted by the optimization. The algorithm promoted the use of sequential plans, with the addition of "rewiring" the system when the goal to attain is in contradiction with the intrinsic behavior of the system (e.g. to create the big ratio of F1).

These results are not that intuitive, i.e. one would think that using parallel assembly steps is more efficient for the convergence time. But comparing the results, we see that the "sequential plan" is much quicker for $\alpha = 0.5$, and slightly worse for the two others ($4 \cdot 10^4$ *seconds* instead of $3 \cdot 10^4$ *seconds*). However, it has to be checked that such an optimization would give similar results for a more complicated set of assembly plans, especially the "full set assembly plan" consisting of all possibles reactions of the system.

This page intentionally left blank.

Chapter 7 Augmented assembly implementation

7.1 Top-bottom approach

This chapter will cover the last step in our methodology: going back from the modified model to the physical system, while applying the introduced modifications.

This is a Top-bottom approach, which is of growing interests nowadays. This method becomes interesting when the behavior of the low-level parts is hard to create, when the global behavior does not simply follow from the behavior of the low-level parts or when we want to low-level parts to follow directives given at a higher level.

In our case, our problem is that we do not know how to modify the behavior of the low-level parts (i.e. the robots and pieces and their interactions) so as to create a high-level goal. When working on the mathematical model level, we can define our high-level goals more easily.

The main difficulty of this approach is to create a mapping from the high-level to the low-level.

7.2 Rates mapping

In our case, this problem is actually simpler, because we have a direct dependance between rates in the mathematical model and behavior of the robots.

Especially, we created a bidirectional relation between the stochastic constant rates of our mathematical model and physical capacities or behavior of our robots and pieces. The mapping from the model to the physical system is thus straightforward:

Backward probability p_i^-: A robot carrying an assembly draws a random number at each timestep and compare it to the p_i^- corresponding to its carried assembly i. Precisely:

if $U(0,1) < p_i^- \cdot T \rightarrow$ *disasssemble assembly* i.

where T is the timestep of the physical simulation. This is needed because p_i^- is a backward rate per second.

Forward probability p_i^+: As shown in Equation (6.18), this probability is used when an assembly is starting. Before actually doing the assembly, a random number is drawn and compared to it:

if $U(0,1) < p_i^+ \cdot T \rightarrow$ *perform assembly step* i.

Reaction j		**1**	**2**	**3**	**4**	**5**	**6**
$\mathbf{p_j^+}$	$\alpha = 0.01$ $\alpha = 0.5$ $\alpha = 0.99$	1.0					
$\mathbf{p_j^-}$	$\alpha = 0.01$ $\alpha = 0.5$ $\alpha = 0.99$	0.018852	0.007541	0.003770	0.033074 0.000661 0.000334	0.009426	0.000134 0.000265 0.013229

Table 7.1: Set of optimized probabilities used for the Top-down mapping. Reactions from system (6.1).

When a disassembly is triggered, the carrying robot drops one of the assemblies on the ground and resume searching with the remaining carried assembly. This dropped assembly should then be grabbed by another robot in order to be attached again.

These behavior should directly create the change of rates needed for the modified mathematical model.

When such a bidirectional mapping is not that easily available, we have to discover that mapping. A possibility would be to create an iterative process between a modification of low-level parameters and a measure of the resulting rate modification in the model.

7.3 Augmentation results and implications

We choose to implement the set of optimized probabilities shown in Table 7.1. It has been found by solving problem P1 in Section 6.3.

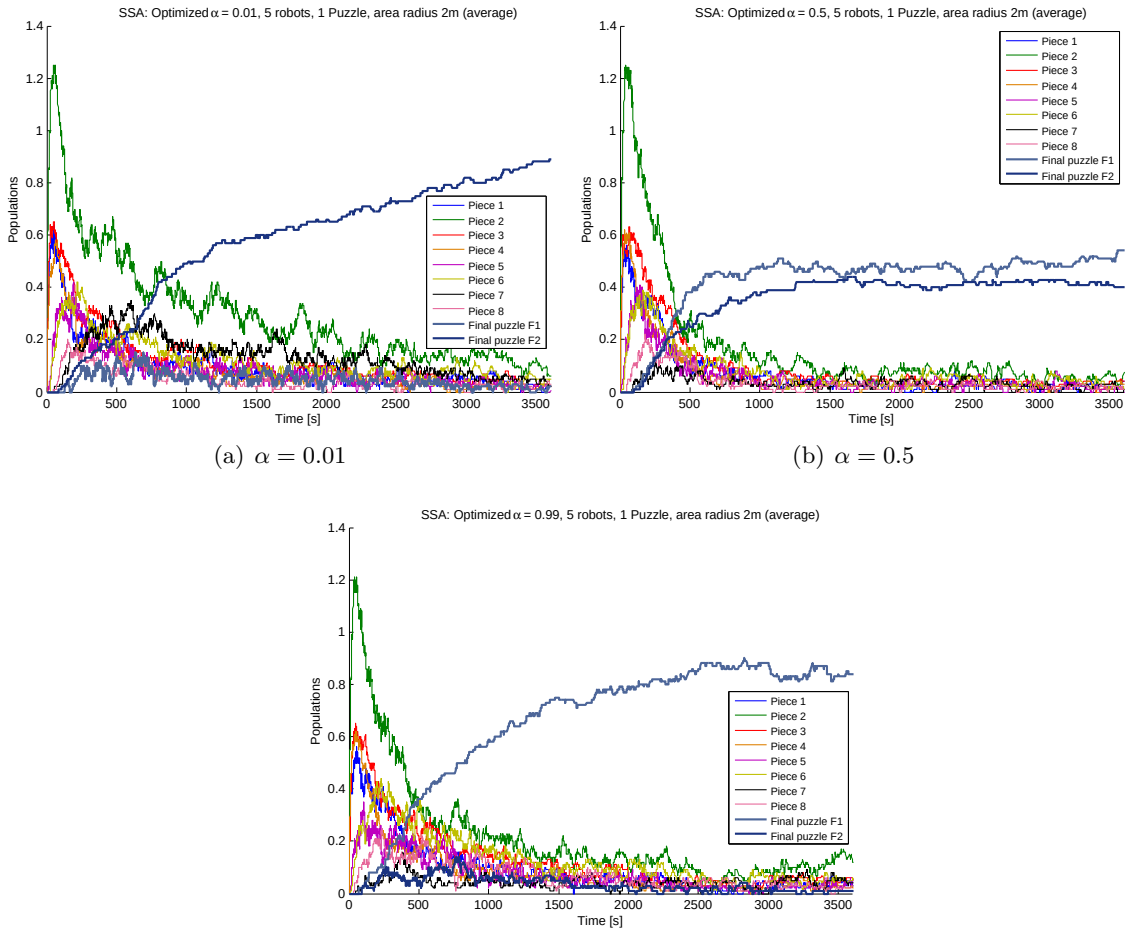
7.3.1 Stochastic model

We start by verifying the behavior with the stochastic simulation of our model with backward rates. We modify Equation (5.2) to add the backward reactions while still modeling the robots and free pieces. Our model takes into account the dropped pieces when a disassembly occurs. The results for 1 puzzle are shown in Figure 7.1. The results are similar for 3 puzzles, so we do not show them here.

According to these simulations, the optimized rates for the simplified system (6.1) translates into the same global optimized behavior when used on the complete system with robots and free pieces. It manages to create the target ratio α quite successfully. This is a good thing, as it shows that we have a full loop between the physical system and the optimization of the model. We see that, because of the low number of pieces available, there are still quite a lot of non final puzzles in the system.

7.3.2 Physical simulation

As explained earlier, we augmented the system by adding the capacity to break assemblies and grab mid-assemblies lying on the floor. We then use the optimized probabilities of



(a) $\alpha = 0.01$

(b) $\alpha = 0.5$

(c) $\alpha = 0.99$

Figure 7.1: Stochastic simulation of the Augmented system, for 1 puzzle and 5 robots.

Table 7.1 before running our simulations.

The carried pieces discrepancy

Our first result is an experiment with $\alpha = 0.01$, using 5 robots and 5 pieces. We run 100 experiments, using the same methodology as explained in Section 4.4.1, for a maximum of 10 minutes. The results are shown in Figure 7.2.

This is quite disappointing, as it does not converge to our desired values, even if we do not run the system for a long time. The biggest problem is the amount of free piece 2: they are more abundant than final assemblies.

Studying visually the behavior in simulation, we discovered that this problem is due to the re-carrying of pieces dropped during a disassembly. When few robots are available, the time till a piece is carried again can be in the same order of magnitude than the time between

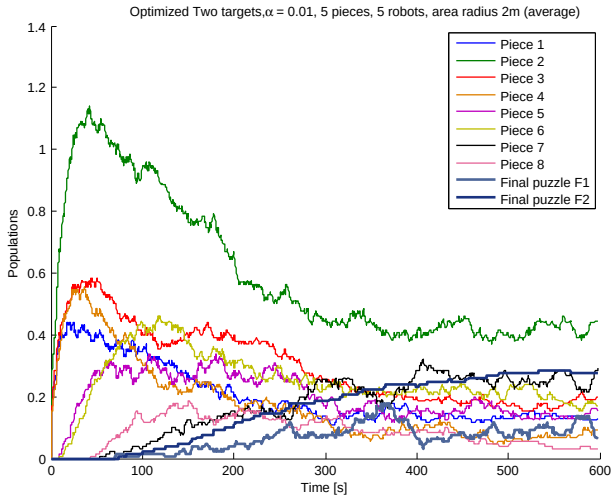


Figure 7.2: Results of the augmented system with optimized rates for $\alpha = 0.01$. Problem of carrying of pieces.

two assemblies. Yet when constructing our model for the optimization, we assumed that the pieces would be carried very quickly (Section 6.2.1). This hypothesis is thus pretty important as its effect shows it here.

In order to correct that, we add 3 robots to the arena, which ensure that the time to re-carrying of pieces is small. We then compare the results of this new physical simulation to the stochastic simulation of the same experiment in Figure 7.3. Again we do 100 experiments, of 10 minutes each for a target value $\alpha = 0.01$.

The results between the physical and the stochastic simulations are comparable but differ slightly:

- The rates of convergence up to 9 minutes are of the same order. The stochastic model fits correctly the physical simulation.
- After 9 minutes, the stochastic model continues to converge towards its equilibrium, as expected from Figure 7.1(a), but the physical simulation saturates to a sub-optimal value. The amount of free pieces 2 is kept pretty high, without assembling to create the desired final puzzle F2.

When observing visually the behavior of the system, it appears that two scenarios are occurring:

1. The system builds a final puzzle F2. As the backward rate from this puzzle is very slow, it is kept complete till the end of the simulation.
2. The system builds a final puzzle F1. According to our rates, it should disassemble back till a convergence to F2. Unfortunately this disassembly does not work that well. Several problem arise:

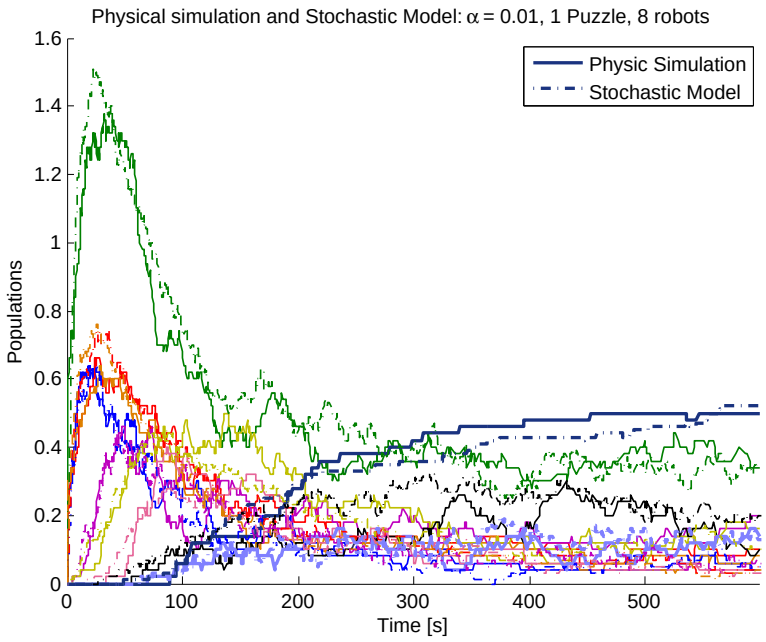


Figure 7.3: Comparison between physical augmented system and stochastic model for $\alpha = 0.01$.

- When F1 disassemble, it is very likely that the lying free piece 2 is taken and assembled back into a new F1. The problem comes from spatial constraints: the piece is dropped near the current robot.
- If F1 disassembles, we end up with an assembly 7 and an initial piece 2. The backward rate to disassemble 7 is low, which posses a problem in our case. We think that because 7 stays alive for some time, it has more chances to assemble back with the piece 2 than disassemble further, because of the spatial problem we presented.

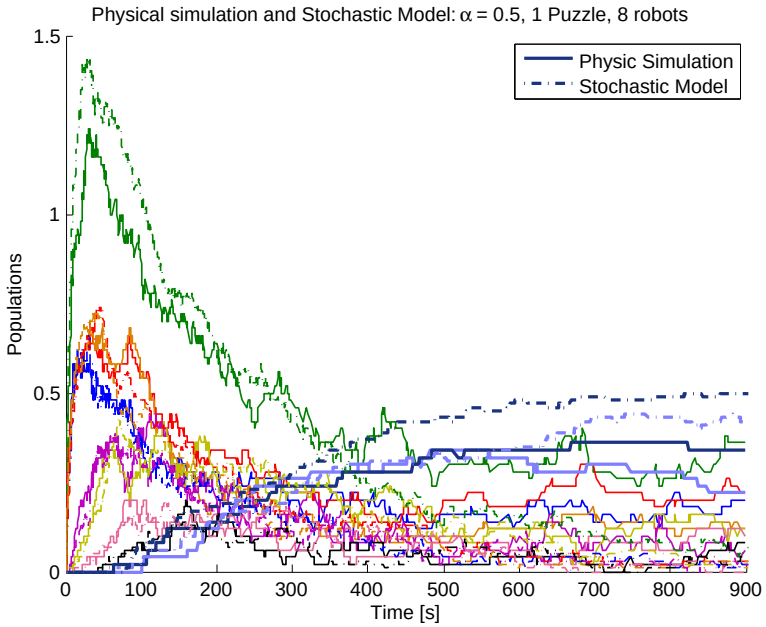
There seems to be an even distribution between the two scenarios, the second of them impeding the capacity of the system to converge to the target ratio. We would need to do an iterative process to bring back this difference to the model level, in order to get optimized rates taking it into account.

Other target ratios α

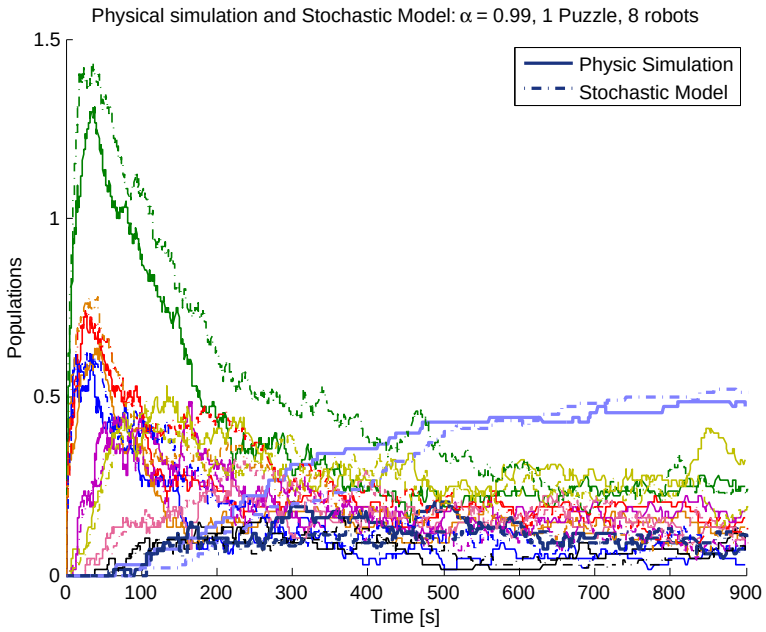
See Figure 7.4 for the two other target ratio and their comparison with the stochastic model. The methodology is again the same, we perform 100 experiments of 20 minutes.

We see that we closely fit the predicted data, but that the physical simulations converge to sub-optimal values in both cases. Again the "mobility" between assemblies in the real system is much smaller than in the model. This prevents a good convergence to theoretical values over time.

Even though the yield is bad, it has to be remarked that the ratios α are successfully



(a) $\alpha = 0.5$



(b) $\alpha = 0.99$

Figure 7.4: Augmented system results for $\alpha = 0.5$ and $\alpha = 0.99$. Comparison with the stochastic model.

enforced by the optimized rates. This might point out that the problem does not come from the dynamics directly but from the small amount of pieces available. Further tests with

bigger amount of pieces would be needed, but we are limited by simulation constraints for the time being.

7.3.3 Implications

We saw that it was possible to use the rates we optimized in a simplified mathematical model to map onto the physical system. It successfully created the desired ratios of final puzzles, even though the dynamics of interactions between robots is quite complex and random. The Top-down approach of our framework is thus valid and promising even at this early stage.

However, some differences can have big impacts. Small copy numbers can disturb the convergence, and the non-spatiality assumption can have bad effects when our physical system does not manage to satisfy it. We think that our approach is still worthy of interest and leads to insightful results, especially theoretically. It would need some tuning in order to map correctly the theoretical optimized probabilities onto the physical system. Unfortunately, this would have to be done in a further work.

This page intentionally left blank.

Chapter 8 Conclusion and outlook

8.1 Conclusion

Through this project, we presented a framework to perform a Top-down control over an existing system. It has been tested on a specific assembly task for robots.

We first stated the overall framework and its components. Particularly, we defined two separate systems connected together by the Top-down control approach, the *intrinsic* and the *augmented* system. Our other major choice is the use of a Chemical Reaction Network mathematical model for all the framework. It allows a description of many systems while being widely accepted and used in the scientific community, especially in the life science community.

We presented the test case on which we have applied our framework. This test case is a robotic assembly platform using a team of multiple robots. This has been completely developed and simulated using the 3D realistic physical simulation Webots. The platform has been created to verify a couple of properties, e.g. a well-mixed property of agents. The robotic platform allows us to assemble pieces following an arbitrary assembly plan, easily modifiable. We measured extensively the behavior of this platform under several conditions.

We introduced the mathematical representation in term of a chemical reaction network of our robotic platform. All its parameters where fitted by using first an heuristic guess based on geometric probabilities, behavior of the robots and measures of our platform. The theoretical parameters were then compared and adapted to the real measured ones, in order to closely fit to the physical simulation. This chemical reaction network has be simulated in two different ways: using an ODE approximation and with an exact stochastic simulation. Both simulations successfully captured the behavior of the physical system. The ODE approximation is wrong when few number of robots and pieces are considered, as one would hypothesize. On the other hand, the stochastic simulation captured especially well the quantitative global behavior of the physical system. Some characteristics still need to be accounted for, for example the irrecoverable errors in the physical simulation, or the divergence from a well-mixed system when the robots overcrowded certain parts of the arena.

We also successfully modeled a scenario where the robotic platform can create two different final assemblies. The assembly plans used, as well as geometric constraints and pieces availability then defined the probability to generate the first or the second final assembly.

We introduced our Top-down control goal as the capacity to accurately control the ratio between the two final assemblies, while converging quickly to that state.

In order to perform this control optimization, we introduced several results for the convergence of chemical networks and their dependence upon the reactions rate constants. We developed an algorithm representing the chemical reaction network and the goal of controlling the final ratio of final assemblies as a linear program function of the rate constants.

It allowed us to define any target ratio and to make the system converge to it only by using a specific set of reactions rate constants. Control of the ratio was achieved by modifying only a small subset of all rates, namely the one controlling the final building reactions. The obtained behavior gave insights into the flexibility shown by such chemical reaction networks. We studied extensively the evolution of the controlled system and the behavior it showed. We then presented a direct application of our findings, in the shape of a "green manufacturing" system, which change its target final assembly smoothly during one experiment. It showed that our robotic platform displayed a flexibility in its capacities that are harder to replicate using classical assembly lines process for example.

We finished by studying the effect of our optimization scheme on a set of assembly plans. The goal was to study if it would directly perform a discrete optimization of the plans, outputting the assembly plans most adapted to the assemblies being built. Interestingly, such a result took place, in a continuous fashion. A close analysis of the optimized rates showed several regimes of activity, corresponding to several dynamic plans, depending on the target assemblies. This is a surprising result, and research in that direction could lead to interesting discoveries.

Finally, we implemented our theoretical findings into our simulated robotic platform. This mapping was straightforward because of the strong relation built through the application of our whole framework. Stochastic simulations of the controlled system showed a behavior in accordance with the theoretical findings, even though several approximations were made to perform the optimization.

However, the physical simulations showed a bigger discrepancy. The number of final puzzle was small and the system was crowded by initial and mid-assemblies. We think this is due to physical characteristics of our system, and because of the non-spatiality assumption made in the model. The real system does not validate this non-spatiality assumption in general, especially when disassembling a piece. This leads to a sub-optimal behavior when the number of pieces is too small.

It showed that the last step in our framework, namely mapping back the model onto the physical system, is crucial, and very sensitive to hypotheses and real problems. It is still interesting to see that a full loop was successfully constructed, which would permit us to perform an iterative improving loop to more closely match the model to the physical system.

We think that our framework showed promising results, especially its model component. Chemical reaction networks are powerful explanatory tools, and they allow for a very precise yet very insightful representation of processes. While still being hard to manipulate and design because of their non linearity, we managed to propose an efficient optimization scheme, allowing a direct Top-down control scheme over the assembly process.

8.2 Outlook

Further work might go in several directions:

1. A more precise and general way to map the mathematical model on the physical system, the Top-down mapping, should be proposed. As of now, we take advantage of the simplicity and strong links between our simulation and our model, but this might not be true for other systems. Furthermore, we saw that, even in our simple case, small problems could cause big issues.
2. We created different assembly platforms, namely a self-assembly and a mixed-assembly platform. They showed interesting results, but were not studied mathematically and optimized in this current work. We think they might show new dynamics which would help improving our framework.
3. Our optimization process should be verified and compared more thoroughly against other optimization and searches in the parameter space of possible rate constants. This is the goal of an ongoing paper on our work.

Finally, we would really like to apply this framework to a completely different system, like a biological system or a microscale assembly process. Such systems show complex dynamics which requires a precise and flexible framework. As we first thought our framework to be applied to such systems, it would be only fair to eventually answer our claim that it is well adapted to such inherently complex systems.

If this comes to be true, we would be pleased to have developed with success a framework capable of handling systems so utterly different as a biological process and a team of multi robots. We think there will be a need for flexible frameworks allowing an easy transfer of knowledge and informations between scientists working on completely different fields. Scientific work done at the frontier of several fields will soon give rise to impressive new possibilities (e.g. nanoscale robots to deliver drugs directly in the body), so if a framework can help towards that goal, we hope our work will provide a step forward.

Chapter 9 Additional Material

9.1 Videos

Videos showing the behavior of the robotic platform, for different scenarios and initial conditions, are available in the project's electronic handout.

9.2 Acknowledgement

I would like to especially thanks and acknowledge:
Spring Berman for all her help, unbridled overnights and weekend hours of work and for abiding my unrestrained blattering.
Gregory Mermoud for the help, support and lengthy discussions on world-changing discoveries.
Prof. Vijay Kumar for his direction, availability and enthusiasm for this project.
Prof. Alcherio Martinoli for his support and the given opportunity to discover a new culture and work in a great university.

And all the great people I've met at International House, who made me like the whole world a bit more.

Bibliography

[1] O. Michel, "Webots: Professional mobile robot simulation," International Journal of Advanced Robotic Systems, vol. 1, Jan 2004.

[2] "Open dynamic engine documentation." http://www.ode.org.

[3] S. Berman, Á. Halász, M. A. Hsieh, and V. Kumar, "Optimal stochastic policies for task allocation in swarms of robots," IEEE Transactions on Robotics, 2008. Under review.

[4] M. Hucka, A. Finney, H. Sauro, H. Bolouri, and J. Doyle, "The systems biology markup language (sbml): a medium for representation and exchange of biochemical network models," Bioinformatics, Jan 2003.

[5] J. Halloy, G. Sempo, G. Caprari, and C. Rivault, "Social integration of robots into groups of cockroaches to control self-organized choices," Science, Jan 2007.

[6] B. Grzybowski and C. Campbell, "Complexity and dynamic self-assembly," Chemical Engineering Science, Jan 2004.

[7] S. Griffith, D. Goldwater, and J. Jacobson, "Self-replication from random parts," Nature, Jan 2005.

[8] M. Rechtsman, F. Stillinger, and S. Torquato, "Designed interaction potentials via inverse methods for self-assembly," Physical Review E, Jan 2006.

[9] A. Forster and G. Church, "Towards synthesis of a minimal cell," Mol. Syst. Biol, Jan 2006.

[10] W. B. Dunbar, N. A. Wilson, J. Schaeffer, E. Klavins, J. Bishop, and B. C. Tanner, "Dynamics and control of biomolecules: Research venues and opportunities," Decision and Control, 2007 46th IEEE Conference on, pp. 3297 – 3307, Nov 2007.

[11] X. Xiong, M. Lidstrom, and B. Parviz, "Microorganisms for mems," Microelectromechanical Systems, Journal of, vol. 16, pp. 429 – 444, Apr 2007.

[12] B. Berger, P. Shor, L. Tucker-Kellogg, and J. King, "Local rule-based theory of virus shell assembly.," Proceedings of the National Academy of Sciences, Jan 1994.

[13] Shen, B. Salemi, D. Rus, M. Moll, H. Lipson, and E. Klavins, "Modular self-reconfigurable robot systems," IEEE ROBOTICS AND AUTOMATION MAGAZINE, Jan 2007.

[14] E. Klavins, "Programmable self-assembly," Control Systems Magazine, Jan 2007.

[15] G. Whitesides and B. Grzybowski, "Self-assembly at all scales," Science, Jan 2002.

[16] M. Kassner, S. Nemat-Nasser, Z. Suo, G. Bao, and et al, "New directions in mechanics," Mechanics of Materials, Jan 2005.

[17] H. Onoe, K. Matsumoto, and I. Shimoyama, "Three-dimensional micro-self-assembly using hydrophobic interaction controlled by self-assembled monolayers," Microelectromechanical Systems, Jan 2004.

[18] W. Zheng and H. Jacobs, "Fabrication of multicomponent microsystems by directed three-dimensional self-assembly," Advanced Functional Materials, Jan 2005.

[19] S. Stauth and B. Parviz, "Self-assembled single-crystal silicon circuits on plastic," Proceedings of the National Academy of Sciences, Jan 2006.

[20] J. Werfel and R. Nagpal, "Extended stigmergy in collective construction," Intelligent Systems, IEEE, vol. 21, pp. 20 – 28, Mar 2006.

[21] J. Bishop, S. Burden, E. Klavins, and R. Kreisberg, "Programmable parts: a demonstration of the grammatical approach to self-organization," Intelligent Robots and Systems, Jan 2005.

[22] J. McNew, E. Klavins, and M. Egerstedt, "Solving coverage problems with embedded graph grammars," Hybrid Systems: Computation and Control, 2007.

[23] E. Klavins, R. Ghrist, and D. Lipsky, "A grammatical approach to self-organizing robotic systems," Urbana, 2005.

[24] M. Feinberg, "Lectures on chemical reaction networks," Notes of lectures given at the Mathematics Research Center, 1979.

[25] D. J. Wilkinson, "Stochastic modelling for systems biology," p. 254, Jan 2006.

[26] F. Horn and R. Jackson, "General mass action kinetics," Archive for Rational Mechanics and Analysis, Jan 1972.

[27] Y. Cao and L. Petzold, "Slow scale tau-leaping method," Computer Methods in Applied Mechanics and Engineering, 2008. to appear.

[28] D. T. Gillespie, "Stochastic simulation of chemical kinetics," Annual Review of Physical Chemistry, vol. 58, pp. 35–55, Jan 2007.

[29] D. Gillespie, "Exact stochastic simulation of coupled chemical reactions," The Journal of Physical Chemistry, Jan 1977.

[30] D. Gillespie, "A rigorous derivation of the chemical master equation," Physica A, Jan 1992.

[31] D. Gillespie, "The chemical langevin equation," The Journal of Chemical Physics, Jan 2000.

[32] D. Gillespie, "The multivariate langevin and fokker–planck equations," American Journal of Physics, Jan 1996.

[33] D. Gillespie, "Approximate accelerated stochastic simulation of chemically reacting systems," Journal of Chemical Physics, vol. 115, no. 4, 2001.

[34] Y. Cao, D. Gillespie, and L. Petzold, "Efficient step size selection for the tau-leaping simulation method," The Journal of Chemical Physics, Jan 2006.

[35] M. Rathinam, L. Petzold, Y. Cao, and D. Gillespie, "Stiffness in stochastic chemically reacting systems: The implicit tau-leaping method," The Journal of Chemical Physics, Jan 2003.

[36] Y. Cao, D. Gillespie, and L. Petzold, "Adaptive explicit-implicit tau-leaping method with automatic tau selection," The Journal of Chemical Physics, Jan 2007.

[37] M. Gibson and J. Bruck, "Efficient exact stochastic simulation of chemical systems with many species and many channels," J. Phys. Chem. A, Jan 2000.

[38] Y. Cao, D. Gillespie, and L. Petzold, "Multiscale stochastic simulation algorithm with stochastic partial equilibrium assumption for chemically reacting systems," Journal of Computational Physics, Jan 2005.

[39] Y. Cao, D. Gillespie, and L. Petzold, "Accelerated stochastic simulation of the stiff enzyme-substrate reaction," The Journal of Chemical Physics, Jan 2005. Proof of slow-scale approximation.

[40] D. Gillespie, L. Petzold, and Y. Cao, "Comment on "nested stochastic simulation algorithm for chemical kinetic systems with disparate rates"," The Journal of Chemical Physics, Jan 2007.

[41] J. Puchalka and A. Kierzek, "Bridging the gap between stochastic and deterministic regimes in the kinetic simulations of the biochemical reaction networks," Biophysical Journal, Jan 2004.

[42] E. Haseltine and J. Rawlings, "Approximate simulation of coupled fast and slow reactions for stochastic chemical kinetics," minds.wisconsin.edu, Jan 2002.

[43] E. Haseltine and J. Rawlings, "On the origins of approximations for stochastic chemical kinetics," The Journal of Chemical Physics, Jan 2005.

[44] T. Kiehl, R. Mattheyses, and M. Simmons, "Hybrid simulation of cellular behavior," Bioinformatics, Jan 2004.

[45] S. Isaacson and C. Peskin, "Incorporating diffusion in complex geometries into stochastic chemical kinetics simulations," SIAM J. Sci. Comput, Jan 2006.

[46] S. Andrews and D. Bray, "Stochastic simulation of chemical reactions with spatial resolution and single molecule detail," Physical Biology, Jan 2004.

[47] T. Lu, D. Volfson, L. Tsimring, and J. Hasty, "Cellular growth and division in the gillespie algorithm," Systems Biology, Jan 2004.

[48] T. Turner, S. Schnell, and K. Burrage, "Stochastic approaches for modelling in vivo reactions," Computational Biology and Chemistry, Jan 2004.

[49] T. Zhang, R. Rohlfs, and R. Schwartz, "Implementation of a discrete event simulator for biological self-assembly systems," Winter Simulation Conference, Jan 2005.

[50] H. Li, Y. Cao, and L. Petzold, "Stochkit: A stochastic simulation toolkit," cs.ucsb.edu.

[51] R. Alur, C. Belta, F. Ivancic, V. Kumar, and M. Mintz, "Hybrid modeling and simulation of biomolecular networks," Hybrid Systems: Computation and Control, Jan 2001.

[52] G. Pólya, "Kombinatorische anzahlbestimmungen für gruppen, graphen und chemische verbindungen," Acta Mathematica, Jan 1937.

[53] H. Berg, "Motile behavior of bacteria," Physics Today, Jan 2000.

[54] J. Adler, "Chemotaxis in bacteria," Annual Reviews in Biochemistry, Jan 1975.

[55] R. Macnab and D. Koshland, "The gradient-sensing mechanism in bacterial chemotaxis," Proceedings of the National Academy of Sciences, Jan 1972.

[56] A. Dhariwal, G. Sukhatme, and A. Requicha, "Bacterium-inspired robots for environmental monitoring," Robotics and Automation, Jan 1926.

[57] B. J. Bornstein, S. M. Keating, A. Jouraku, and M. Hucka, "Libsbml: an api library for sbml," Bioinformatics, vol. 24, pp. 880–1, Mar 2008.

[58] N. Correll and A. Martinoli, "Modeling self-organized aggregation in a swarm of miniature robots," IEEE 2007 International Conference on Robotics and ..., Jan 2007.

[59] N. Correll and A. Martinoli, "Robust distributed coverage using a swarm of miniature robots," Robotics and Automation, Jan 2007.

[60] N. Correll and A. Martinoli, "System identification of self-organizing robotic swarms," Proc. of the Eight Int. Symp. on Distributed Autonomous ..., Jan 2006.

[61] A. Halasz, M. Hsieh, S. Berman, and V. Kumar, "Dynamic redistribution of a swarm of robots among multiple sites," Intelligent Robots and Systems, Jan 2007.

[62] D. Randall, "Rapidly mixing markov chains with applications in computer science and physics," Computing in Science & Engineering, vol. 8, pp. 30 – 41, Mar 2006.

[63] S. N. Ethier and T. G. Kurtz, "Markov processes: Characterization and convergence," p. 544, Jan 1986.

[64] J. Sun, S. Boyd, L. Xiao, and P. Diaconis, "The fastest mixing markov process on a graph and a connection to a maximum variance unfolding problem," SIAM Review, Jan 2005.

[65] T. Runolfsson and Y. Ma, "Model reduction of nonreversible markov chains," Decision and Control, pp. 3739—3744, Jan 2007.

[66] L. Tierney, "Markov chains for exploring posterior distributions," Annals of Statistics, Jan 1994.

[67] S. Berman, A. Halasz, and V. Kumar, "Marco: A reachability algorithm for multi-affine systems with applications to biological systems," International Conference on Hybrid Systems: Computation and Control (HSCC'07), pp. 76–89, 2007.

[68] G. Bastin and J. Levine, "On state accessibility in reaction systems," Automatic Control, Jan 1993.

[69] M. Kloetzer and C. Belta, "Reachability analysis of multi-affine systems," Hybrid Systems: Computation and Control, Jan 2006.

[70] G. Bastin and D. Dochain, "On-line estimation and adaptative control of bioreactors," Laboratoire D'Automatique, Jan 1990.

[71] G. Batt, B. Yordanov, R. Weiss, and C. Belta, "Robustness analysis and tuning of synthetic gene networks," Bioinformatics, Jan 2007.

[72] I. Otero-Muras, G. Szederkényi, K. Hangos, and A. Alonso, "Dynamic analysis and control of biochemical reaction networks," Mathematics and Computers in Simulation, Jan 2008.

[73] P. Iglesias, M. Khammash, B. Munsky, E. Sontag, and D. D. Vecchio, "Systems biology and control — a tutorial," Decision and Control, 2007 46th IEEE Conference on, pp. 1 – 12, Nov 2007.

[74] C. Belta, L. Habets, and V. Kumar, "Control of multi-affine systems on rectangles with applications to hybrid biomolecular networks," Decision and Control, Jan 2002.

[75] C. Belta and L. Habets, "Controlling a class of nonlinear systems on rectangles," IEEE Trans. Aut. Control, Jan 2006.

[76] S. Schuster and R. Heinrich, "Time hierarchy in enzymatic reaction chains resulting from optimality principles," Journal of theoretical biology, vol. 129, pp. 189–209, Nov 1987.

[77] M. Chaves, E. D. Sontag, and R. J. Dinerstein, "Steady-states of receptor-ligand dynamics: a theoretical framework," Journal of theoretical biology, vol. 227, pp. 413–28, Apr 2004.

[78] M. Feinberg, "Chemical reaction network structure and the stability of complex isothermal reactors. i: The deficiency zero and deficiency one theorems," Chemical Engineering Science, Jan 1987.

[79] G. Craciun and M. Feinberg, "Multiple equilibria in complex chemical reaction networks: I. the injectivity property," SIAM J. Appl. Math, Jan 2005.

[80] G. Craciun and M. Feinberg, "Multiple equilibria in complex chemical reaction networks: Ii. the species-reaction graph," SIAM J. Appl. Math, Jan 2006.

[81] M. Feinberg, "The existence and uniqueness of steady states for a class of chemical reaction networks," Archive for Rational Mechanics and Analysis, Jan 1995.

[82] M. Feinberg and D. Hildebrandt, "Optimal reactor design from a geometric viewpoint—i. universal properties of the attainable region," Chemical Engineering Science, Jan 1997.

[83] D. Bernstein and S. Bhat, "Nonnegativity, reducibility, and semistability of mass actionkinetics," Decision and Control, Jan 1999.

[84] G. Craciun, Y. Tang, and M. Feinberg, "Understanding bistability in complex enzyme-driven reaction networks," Proceedings of the National Academy of Sciences, Jan 2006.

[85] E. Sontag, "Monotone and near-monotone biochemical networks," Systems and Synthetic Biology, Jan 2007.

[86] D. Angeli and E. Sontag, "Translation-invariant monotone systems, and a global convergence result for enzymatic futile cycles," Nonlinear Analysis: Real World Applications, Jan 2008.

[87] E. D. Sontag, "Some new directions in control theory inspired by systems biology," Systems Biology, vol. 1, pp. 9–18, Jun 2004.

[88] M. Chaves, E. Sontag, and R. Dinerstein, "Steady-states of receptor–ligand dynamics: a theoretical framework," Journal of theoretical biology, Jan 2004.

[89] E. Sontag, "Structure and stability of certain chemical networks and applications to the kinetic proofreading model of t-cell receptor signal transduction," Automatic Control, Jan 2001.

[90] R. Heinrich and S. Schuster, The Regulation of Cellular Systems. 1996.

[91] N. Jamshidi and B. O. Palsson, "Formulating genome-scale kinetic models in the post-genome era," Molecular Systems Biology, vol. 4, no. 171, pp. 1–10, 2008.

[92] R. Heinrich and S. M. Rapoport, "Metabolic regulation and mathematical models," Prog. Biophys. Molec. Biol., vol. 32, pp. 1–82, 1977.

[93] R. Heinrich, S. Schuster, and H.-G. Holzhutter, "Mathematical analysis of enzymic reaction systems using optimization principles," Eur. J. Biochem., vol. 201, pp. 1–21, 1991.

[94] J. Lofberg, "Yalmip: a toolbox for modeling and optimization in matlab," Computer Aided Control Systems Design, Jan 2004.

[95] A. Julius, A. Halasz, V. Kumar, and G. Pappas, "Controlling biological systems: the lactose regulation system of escherichia coli," American Control Conference, Jan 2007.

[96] A. Julius, A. Halasz, M. Sakar, H. Rubin, and V. Kumar, "Stochastic modeling and control of biological systems: The lactose regulation system of escherichi coli," Automatic Control, Jan 2008.

[97] J.-Y. L. Boudec, "Modelling the immune system toolbox: Stochastic reaction models." 2006.

[98] T. Tian and K. Burrage, "Stochastic models for regulatory networks of the genetic toggle switch," Proceedings of the National Academy of Sciences, Jan 2006.

[99] B. Sweeney, T. Zhang, and R. Schwartz, "Exploring the parameter space of complex self-assembly through virus capsid models," Biophys J, vol. 94, pp. 772–783, Jan 2008.

[100] A. L. Tournier, P. W. Fitzjohn, and P. A. Bates, "Probability-based model of protein-protein interactions on biological timescales," Algorithms for molecular biology : AMB, vol. 1, p. 25, Jan 2006.

[101] J. Halloy, G. Sempo, G. Caprari, and C. Rivault, "Social integration of robots into groups of cockroaches to control self-organized choices," Science, Jan 2007.

[102] I. Smets, J. Claes, E. November, G. Bastin, and J. V. . . . , "Optimal adaptive control of (bio) chemical reactors: past, present and future," Journal of Process Control, Jan 2004.

[103] R. Gunawan, Y. Cao, L. Petzold, and F. Doyle, "Sensitivity analysis of discrete stochastic systems," Biophysical Journal, Jan 2005.

[104] S. Berman, A. Halasz, V. Kumar, and S. Pratt, "Algorithms for the analysis and synthesis of a bio-inspired swarm robotic system," Swarm Robotics SAB 2006 International Workshop, Jan 2006.

[105] F. Jamalyaria, R. Rohlfs, and R. Schwartz, "Queue-based method for efficient simulation of biological self-assembly systems," Journal of Computational Physics, Jan 2005.

[106] M. Branicky, V. Borkar, and S. Mitter, "A unified framework for hybrid control: model and optimal controltheory," Automatic Control, Jan 1998.

[107] S. Berman, A. Halasz, V. Kumar, and S. Pratt, "Bio-inspired group behaviors for the deployment of a swarm of robots to multiple destinations," Proceedings of the 2007 IEEE International Conference on Robotics and Automation (ICRA'07), pp. 2318–2323, Jan 2007.