HAL open science

Formal methods for modelling and validation of biological models

Alexandre Rocca

To cite this version:

Alexandre Rocca. Formal methods for modelling and validation of biological models. Mathematical Physics [math-ph]. Université Grenoble Alpes, 2018. English. NNT : 2018GREAM028 . tel-01913296

HAL Id: tel-01913296

https://theses.hal.science/tel-01913296v1

Submitted on 6 Nov 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Communauté UNIVERSITÉ Grenoble Alpes

THÈSE

Pour obtenir le grade de

DOCTEUR DE LA COMMUNAUTÉ UNIVERSITÉ GRENOBLE ALPES

Spécialité : **Mathématiques appliquées**

Arrêté ministériel : 25 mai 2016

Présentée par

Alexandre Rocca

Thèse dirigée par **Thao Dang,** Directeur de Recherche CNRS,
et codirigée par **Éric Fanchon,** Chargé de Recherche CNRS
préparée au sein des **laboratoires VERIMAG et TIMC-IMAG**
dans **l'Ecole Doctorale Mathématiques, Sciences et Technologies de l'Information, Informatique**

Formal methods for modelling and validation of biological models

Méthodes formelles pour la modélisation et la validation de modèles biologiques

Thèse soutenue publiquement le **07 mai 2018**,
devant le jury composé de :

Mme Thao Dang
DIRECTEUR DE RECHERCHE,CNRS DÉLÉGATION ALPES, Directeur de thèse

M. François Fages
DIRECTEUR DE RECHERCHE, INRIA CENTRE SACLAY-ÎLE-DE-FRANCE, Rapporteur

M. Ovidiu Radulescu
PROFESSEUR, UNIVERSITÉ DE MONTPELIER, Rapporteur

M. Éric Fanchon
CHARGÉ DE RECHERCHE, CNRS DÉLÉGATION ALPES, Examinateur

M. Hidde de Jong
DIRECTEUR DE RECHECHE, INRIA CENTRE GRENOBLE-RHÔNE-ALPES, Président

M. DavidŠafránek
ASSISTANT PROFESSOR, BRNO-MASARYK UNIVERSITY, MLDP DEPARTMENT, CZECH REPUBLIC, Examinateur

Acknowledgements

I want to thank my supervisors Thao Dang and Eric Fanchon for this opportunity, and great experience. I also want to thanks my co-workers from the Verimag and TIMC-IMAG laboratories who supported me during all these years and all these great coffee breaks. Especially, I want to thanks Marcelo and Victor for putting me back on the tracks at some time when I was feeling lost in my research. Finally, I want to thank my family and my girlfriend for always finding the good words and encouraging me in the times I was feeling down.

Abstract

The focus of this thesis is the modelling and analysis of biological systems using formal methods. The dynamics of biological systems exhibit continuous behaviours but also abrupt changes. Ordinary differential equations and hybrid dynamical systems are two mathematical formalisms that naturally model such dynamics.

A crucial aspect of modelling is the determination of valid parameter values that enable to simulate the behaviour and reproduce experimental data sets. If no valid parameter values are found it becomes necessary to revise the model. An option is to replace one or several lumped parameters (parameters which represent a set of processes) by functions of time. In this thesis we first study the model revision problem on hybrid dynamical systems. To this aim we propose a greedy scheme of optimal control methods based on occupation measures and convex relaxations.

Then, we study how to characterize dynamical properties of a model using set-based simulations and reachability analysis. For this purpose, we propose two methods: the first one, which relies on Bernstein expansion, is an extension for hybrid dynamical systems of the reachability tool `Sapo` [1], while the other one uses Krivine-Stengle representations [2] to perform the reachability analysis of polynomial ODEs. Finally, We also propose a methodology to generate hybrid dynamical systems modelling a class of experimental protocols.

The proposed methods are applied to different case studies. We first propose a model of haemoglobin production during the differentiation of an erythrocyte in the bone marrow [3]. To develop this model, we first apply the Monte-Carlo based parameters synthesis, followed by the model revision to correctly fit to the experimental data [4]. We also propose a preliminary study of the effect of low dose Cadmium on glucose response at different steps of a rat growth. Finally, we apply the reachability analysis techniques for the validation on large parameters set of the existing iron homoeostasis model [5], [6]. We note the haemoglobin production process, as well as the glucose response system can be formalised, with their experimental context, as hybrid dynamical systems. Thus, they serve as proof of concept for the methodology of biological experimental protocols modelling.

Résumé

L'objectif de cette thèse est la modélisation et l'étude de systèmes biologiques par l'intermédiaire de méthodes formelles. Les systèmes biologiques démontrent des comportements continues mais sont aussi susceptibles de montrer des changements abruptes dans leur dynamiques. Les équations différentielles ordinaires, ainsi que les systèmes dynamiques hybrides, sont deux formalismes mathématiques utilisés pour modéliser clairement de tels comportements.

Un point critique de la modélisation de systèmes biologiques est la recherche des valeurs des paramètres du modèle afin de reproduire de manière précise un ensemble de données expérimentales. Si aucun jeux de paramètres valides n'est trouvé, il est nécessaire de réviser le modèle. Une possibilité est alors de remplacer un paramètre, ou un ensemble de paramètres, définissant un processus biologique par une fonction dépendante du temps.

Dans le cadre de cette thèse, nous exposons tout d'abord une méthode pour la révision de modèles hybrides. Pour cela, nous proposons une approche gloutonne appliquée à une méthode de contrôle optimal utilisant les mesures d'occupations et la relaxation convexe. Ensuite, nous étudions comment analyser les propriétés dynamiques d'un modèle à temps discret en utilisant la simulation ensembliste. Dans cet objectif, nous proposons deux méthodes basées sur deux outils mathématiques. La première méthode, qui se repose sur les polynômes de Bernstein, est une extension aux systèmes dynamiques hybrides, de l'outil de calcul ensembliste `Sapo` [1]. La seconde méthode utilise les représentations de Krivine-Stengle [2] pour permettre l'analyse d'atteignaiblité de systèmes dynamiques polynomiaux. Enfin, nous proposons aussi une méthodologie pour générer des systèmes dynamiques hybrides modélisant des protocoles biologiques expérimentaux.

Les méthodes précédemment proposées sont appliquées sur divers études biologiques. Nous étudions tout d'abord un modèle de la production d'hémoglobine durant la différentiation des érythrocytes dans la moelle [3]. Pour permettre la construction de ce modèle, nous avons dans un premier temps généré un ensemble de jeux de paramètres valides à l'aide d'une méthode de type Monte-Carlo. Dans un second temps, nous avons appliqué la méthode de révision de modèle afin de reproduire plus précisément les données expérimentales [4]. Nous proposons aussi un modèle préliminaire des effets à faibles doses du Cadmium sur la réponse du métabolisme à différentes étapes de la vie d'un rat. Enfin, nous appliquons les

techniques d'analyse ensembliste pour la validation d'hypothèses sur un modèle d'homéostasie du fer [6] dans le cas où des paramètres varient dans de larges intervalles. Dans cette thèse, nous montrons aussi que le protocole associé à l'étude de la production d'hémoglobine, ainsi que le protocole étudiant l'intégration du Cadmium durant la vie d'un rat, peuvent être formalisés comme des systèmes dynamiques hybrides, et servent ainsi de preuves de concepts pour notre méthode de modélisation de protocoles expérimentaux.

Contents

Abstract **v**

Résumé **vii**

1 Introduction **1**
1.1 Context 1
1.2 State of the Art 6
1.3 Contributions 9

2 Preliminaries **13**
2.1 Ordinary differential equations 13
2.2 Hybrid dynamical systems 19

3 Occupation measures and optimal control for model revision **29**
3.1 Hybrid Optimal Control using Occupation measures 32
3.1.1 Introduction to occupations measures 32
3.1.2 Solving the hybrid optimal control problem (HOCP) . . . 35
3.2 Fitting time varying parameters 39
3.2.1 The model revision problem 39
3.2.2 Algorithm and implementation details 41
3.3 Perspectives 42

4 Set-based simulation for biological models validation **45**
4.1 Set-based simulation 48
4.1.1 Discrete time models 48
4.1.2 Template reachability analysis 49
4.1.3 Approximated dynamics over partitions 51
4.2 The Bernstein expansion based method 57
4.2.1 Polynomial optimization using the Bernstein expansion . . 57
4.2.2 Validation of piecewise polynomial ODE systems using the Bernstein expansion 64
4.3 The Krivine-Stengle representation based method 69
4.3.1 Polynomial Optimization using K.S representations 70
4.3.2 Set-based simulation using K.S representations 73

4.4 Discussion and perspectives 76

5 MOEPLA: a language for experimental protocol modelling 83
5.1 MOEPLA description 85
5.2 MOEPLA specification to hybrid automata 89
5.3 Discussion and future work 93

6 Applications and case studies 99
6.1 Erythropoiesis haemoglobin production model 99
6.1.1 Parameter study 100
6.1.2 Model revision 107
6.2 Mammalian cellular iron homoeostasis (MCIH) model 115
6.2.1 Context 115
6.2.2 Set-based analysis 117
6.3 Cadmium impact on glucose response model 122

7 Conclusion 131
7.1 Summary 131
7.2 Future work 132

Bibliography 133

Chapter 1

Introduction

Essentially, all models are wrong, but some are useful.

George E.P. Box

1.1 Context

Computational models in systems biology. Any science in interaction with the real world needs models. In Figure 1.1, we give a possible point of view on the modelling approach. The real world forms a whole: the environment in which we measure observations exhibiting a particular phenomenon. The phenomenon results from interactions of entities that are part of the environment. This environment may be set or constrained by an experimentation, or it can be free and uncontrolled. The observations are the measures performed that allow to observe, at least partially, the phenomenon. Then, a model is an abstraction of the observable world using a mathematical or computational formalism. Given inputs representing a possible state of a considered environment, a model provides outputs associated to some observations of a real phenomenon (see Figure 1.1).

The purposes of models are numerous. They can be used to replicate in-silico real-life experiments, reducing the time and cost of multiple real-life tests. Similarly, they can provide prediction, monitoring and diagnosis capabilities for a given phenomenon and for a known environment. This is primarily achieved by black-box models (also called operational models) which are built directly from observations, using inference-based techniques, machine learning, or model identification. In the biological context, this is especially useful when the environment is repetitive, well-sampled and with numerous observations. For example, any statistical model performing disease diagnostics is of this kind: given a set of well determined biological markers a model provides the probability of a good or bad evolution of

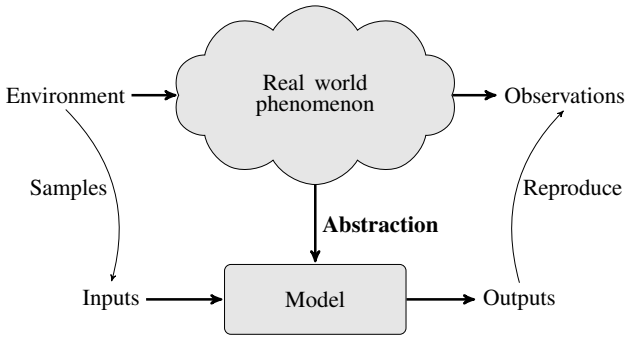

Figure 1.1: Models as an abstraction of the real world.

the disease. More advanced techniques, under the name of model identification, try to infer the complete structure of a model in an attempt to obtain a representation of the actual mechanism leading to the observed phenomenon. An example of such methods using hybrid systems as models is [7]. While black-box models are a very efficient way to represent a phenomenon for the previous applications, they are clearly limited on, at least, three points:

1. They are only useful when the inputs are taken inside a well defined environment.

2. It is hard to integrate new results, any real-life observations, which are not reproduced by the model, without reconsidering the whole model.

3. No actual knowledge or understanding of the underlying mechanism can be easily obtained from such models.

Therefore, in addition to black-box model there are models whose purpose is to represent and abstract the knowledge about the mechanism leading to the observed phenomenon. In the following, we will call these models mechanistic models. However, in the literature they are also named white-box models or denotational models. In black-box models the structure of the mathematical functions describing a phenomenon is directly inferred from observations. This is in contrast with mechanistic models where the structure of the laws defining the inputs-outputs relationship results from hypotheses and/or previous knowledge on the mechanism associated to the studied phenomenon.

Mechanistic models are employed to understand and represent the existing knowledge about a particular mechanism. Therefore, another purpose is to test hypotheses on unknown components of the mechanism Otherwise, up to our current observations, our understanding of the mechanism is more likely to be true: there is yet no guarantee. Consequently, a model can only disprove a hypothesis. Finally, mechanistic models can be used, to a certain extent, to design more faithful black-box models or to provide prediction and monitoring abilities in some applications [8, 9]. Unlike black-box models, which are useful only when they

are coherent with the observations, the mechanistic models have a greater panel of applications. However, they require in general more time to be designed and result from numerous loops between experimentation and validation with respect to the observations. Moreover, they are often more computationally expensive than black-box models.

We remark that mechanistic models can be studied with qualitative and/or quantitative methods. Qualitative methods seek to represent and describe the general behaviours of an observed phenomenon, for example the number of attractors or cycles, the stability or bifurcation conditions, etc. They often rely on strongly theoretical approaches and do not necessarily need numerical methods and computations for other purposes than displaying results. On the other hand, quantitative methods do not necessarily provide strong theoretical results on the behaviours. However, they seek to reproduce the observations as faithfully as possible, by studying the behaviours and the numerical values associated to both the observations and the variables describing the environment.

In this thesis, we will mainly address the problem of modelling biological phenomena and observations using mechanistic models with quantitative methods.

Mathematical formalisms. There exists a large choice of mathematical formalisms such as process algebra, Boolean networks, Petri nets, Thomas networks, stochastic models, difference equations, ordinary differential equation (ODEs), partial differential equations (PDEs), or hybrid dynamical systems. The choice of the formalism strongly depends on the types of available data and the questions asked.

Along this thesis, we use three from these mathematical formalisms. The first one is the ODE formalism. ODEs describe the variations of the entities under study over an infinitely small time instant dt. Hence, instead of directly modelling the evolution of each entity as function of time, we represent how the entities interact with each other. The actual evolution is then obtained by integrating the ODEs over the time.

In the ODE formalism, both time and state variables are continuous. Moreover, the ODE is a deterministic formalism and describes in general average and macroscopic quantities. Therefore stochastic effects are neglected: the ODE's solution must be similar to the average observations of numerous experimental runs with the same environment settings. In system biology, the relevance of the ODEs can be discussed, in particular for the study of molecular reactions with small or inhomogeneous entities concentrations. In this thesis we do not consider stochastic models.

The second formalism is discrete time systems and in particular difference equations. Difference equations are similar to the ODE formalism in the sense that we assume determinism and continuous state variables. However, they operate on discrete time instead of continuous time. Consequently, discrete time system computation is more direct than ODE numerical integration. It is also interesting to note that solutions of ODEs can be approximated by discrete time systems.

The third formalism is hybrid dynamical systems. Using this formalism, one can combine discrete and continuous dynamics. It is important to note that the term "hybrid system" in system biology refers to a wide range of formalisms as shown in [10]. Thus, we emphasize that in this thesis we use this term to denote hybrid dynamical systems. Additionally, as we do not consider stochastic hybrid systems, we refer the interested reader to a recent review [11] for information about them.

Problems arising in biological systems modelling. We recall that we are considering the problem of modelling of biological systems defined by: a phenomenon under study (for example, metabolic response to glucose intake), an environment (for example, a study on rats, or a particular diet), and a set of observations (for example the plasma glucose and insulin measurements).

The first step of the modelling process is the definition of the involved entities and the model structure describing the interactions between the entities. The entities (represented by state variables) to consider can be determined from the observations, the perturbations performed during the experiment, or they can be defined an abstract representation of a more complex mechanism. Similarly, the model structure (network topology) can be established using a priori knowledge, or from hypotheses and insights on the experimentations. Then, kinetics laws, such as the mass action laws, provide the equations governing the dynamics. We remark that at these equations are defined using parameters whose values may be uncertain. We explain the need of parameter estimation a bit further.

The first challenge is to determine the appropriate approximation level of the model. One attempt would be to integrate all previous modelling efforts and knowledge into the new model. This results in an overly complex model spanning over multiple scales which is hard to understand, validate and/or simulate. Another attempt is to design a model as simple as possible by representing solely the observable entities and by approximating or inferring other mechanisms. This leads to models which either cannot reproduce the existing observations, or cannot reproduce future observations of the same phenomenon but in slightly different environment settings. To know about methods for formally handle this first step, we refer to some work on model integrations [12], or model reduction techniques [13, 14, 15]. These results provide mathematical frameworks to approximate and integrate previous knowledge, or to reduce multi-scale models arising from knowledge integration, while ensuring the correctness of the resulting approximations.

The values of the parameters are especially important in quantitative modelling as we seek not only to reproduce the general behaviours of the observations, but also to replicate the same numerical results.

In classical physics these parameters are often well-defined physical constants which can be directly measured through experiments. In engineering, the amount and the accuracy of observations are often very high, leading to accurate parameter identification. Moreover, a priori ranges for the parameter values are easily estimated from physical laws, or parameters are often related to real world components

and have fixed values by construction.

In biology, parameters may represent physical or chemical constants, but they are often an abstraction of an underlying mechanism which is approximated by a constant for a given environment setting. Consequently, parameters may not be directly observable as in classical physics or engineering, and their actual ranges may be completely unknown at first. Moreover, actual experimental observations are scarce, and/or with high variance leading to poor parameter estimation quality. The parameter estimation and validation are critical issues in biological systems modelling. In the next section, Section 1.2, we will discuss briefly some work on parameter estimation for the considered formalisms. Once a set of parameter values (also called parametrization of the model, or parameter set) enable us to reproduce experimental observations, we need to determine the level of confidence in the quality of this parameter set, and to validate the model.

Parameter uncertainty analysis (sensitivity analysis, identifiability, and robustness) is a way to assess the quality of a parameter estimation with respect to experimental observations. In this work, we do not develop new methods to address these issues and we refer the interested reader to a recent review on the subject [16] for ODEs.

In addition to these methods, it is also possible to apply set-based methods from the field of formal verification to ensure the validity of the model with respect to the observations. Formal methods are the set of computational methods which provide guarantees and proofs, which can be achieved by convergence properties, conservative or certifiable results. Formal methods originally come from the field of automated theorem proving and algorithmic verification. In these topics, a model is considered as an accurate representation of the reality, such as a mathematical theorem, or a system/program execution and a major problem is to provide a binary answer to the question whether the model satisfies some property. In biology, a model is a hypothesis or an abstraction of the reality, and it is important to find a counter-example disproving the model, but proving that there is no counter-example is not critical, since experiments themselves may be inaccurate or incomplete. We remark that when dealing with the particular field of synthetic biology, therapy on patients and experimental protocol design, the issue of proving the correctness of the model may become critical in the future. Indeed, an error in the model would lead to patient death or failure of an expensive experimentation. However, formal methods are efficient in proving qualitative properties of the models, which eases considerably the theoretical work. Moreover, set-based methods and methods from formal verification give an exhaustive abstraction of all the possible solutions for a given set of inputs. They are also efficient to find counter-examples, or to deal with dense sets of possible inputs. When dealing with parameter uncertainties over multiple orders of magnitude, methods based on numerical simulations and large number of samplings may be cost inefficient due to the exponential volume of the parameter space to explore. Using set-based analysis from formal methods one can provide conservative results in the form of solution sets which can be more efficient than simulations. We refer the reader to three in-

teresting articles on the use of formal methods for systems biology [17, 18, 19]. In addition, for more details on the general process of biological systems mathematical modelling we recommend the following work [20, 21, 22].

Finally, we note that if there is no parameter valuation that allows reproducing the experimental observations, either the parameter estimation method is lacking, or the proposed model of the mechanism is incorrect. In the latter case, we need to update the model: we call this step model revision. This step can be performed in two manners. First, a model can be revised by adding additional knowledge, which were not previously considered, into the model. Second, in absence of insight and additional information, model revision can be done by assuming that a parameter modelling an underlying mechanism needs to be studied more in depth: in this case, one can resort to functional optimization techniques similar to model identification, to infer a better approximation for the underlying mechanism.

In this thesis, we address three of the above issues (designing biological model, model validation, and model revision) using techniques from the field of formal methods.

In the next section we provide an overview on dynamical hybrid system modelling for systems biology, a short review on parameter synthesis and parameter estimation, and finally on formal methods for validation of biological systems. An additional discussion on related work is provided in each chapter.

1.2 State of the Art

Hybrid dynamical systems modelling. In this thesis we use hybrid dynamical systems as a mathematical formalism to represent biological systems. Models of biological systems can be directly designed using hybrid dynamics. Indeed, this formalism allows us to clearly define and combine the continuous and homogeneous behaviours with discrete or discontinuous ones. One can model the different phases of a biological process, or cyclic behaviours, as a hybrid system. For example, in [23] the hybrid formalism was applied to the cell cycle modelling, while [24] and [25] are concerned with modelling cell behavioural changes under a stimulus. In [26], this model is used to specify multi-step protocols[1], while in [27] hybrid systems describe the spiking models and multi-phases behaviour of neuron activity.

In addition to the modelling purposes, hybrid systems can be used to reduce the complexity of non-linear dynamical systems.

Such approach was first considered for engineering goals as in [28] on plane control systems. More recent work addresses the problem of hybridization, that is approximation of complex non-linear dynamics by a hybrid system which is easier to compute and/or analyze. The resulting hybrid system can have dynamics described by linear ODEs in which case we speak of piecewise linearisation

[1] In a similar approach to the work described further in this thesis

techniques[2]. These linearisations can either be decided initially from the system equations [29, 30, 31] or on the fly [32, 33]. It is also possible to approximate a non-linear continuous model by a hybrid system with simpler non-linear dynamics [34].

For biological models, another way to reduce the model complexity is to propose hybridization methods approximating multi time scales dynamics and explicitly separate fast and slow modes inherent to biological processes [35, 36, 37]. We note that the latter work recovers, using formal methods, the previous hybrid modelling proposed in [23]. Activation processes in gene regulatory networks can also be formally represented by hybrid systems (piecewise linear) as demonstrated in [38].

Finally, we invite the reader to consult [18, 39] which review the aims and methods for hybrid systems modelling as an application to systems biology [18, 39]. More details on the hybrid system formalism itself can be found in Chapter 2-Section 2.2, or in introductory books [40, 41].

Parameter estimation techniques. In this thesis we do not directly address the problem of parameter estimation (also called parameter synthesis), and only provide simple exploratory schemes that we use in our case studies. However, we still provide here a short view of different tools and methods which handle this issue.

First, we remark that there are a number of methods and tools based on numerous simulation and optimization methods as explained in the previously cited review on ODE parameter uncertainty analysis [16]. The search of valid parameters is equivalent to minimizing a cost function modelling either the distance to some experimental time series [42, 43] or some temporal logic formulas describing expected realistic behaviours [44] or [45, 46] (based on Breach toolbox [47, 48]). While some of this work uses a purely optimization based approach [45, 46, 42, 43, 49] (see [50] for a review on optimization methods in systems biology), others consider statistical or probabilistic methods [51, 52] and [53] (based on [54] Bayesian modelling). Among this work, only [43, 47] are designed to handle hybrid systems or black-box simulations[3], while all the other methods are specific to ODEs or stochastic models. The work in [26] proposes, for a case study, an adaptation of the particle swarm technique for parameter estimation of non-linear hybrid systems. Only [43] handles multi-objective optimization problems, which is a useful when simultaneously considering observations from various experiments.

In addition to these simulation-based techniques, some other works consider the problem of searching for a set of valid parameters using techniques from formal verification. In [5] the author proposes a method to search for clusters of valid parameter sets using the robustness-based algorithm from [47]. Gene regulatory networks are an important topic in systems biology, which has led to intensive

[2]The hybrid dynamical system formalism may not be explicitly used but the resulting behaviours can be considered considered as hybrid.

[3]The tool only knows the trajectory solution and not the actual structure of the model.

modelling effort [55, 56, 57, 58, 59]. Gene regulatory networks can be modelled as multi-affine hybrid systems with box domains. Using previous theoretical work [60, 61, 62] on sigmoid approximations, it is possible to exploit the convexity of the flow at the corner of the domains. The previously cited works [55, 56, 57, 58, 59] use this to perform reachability analysis and parameter estimations. In [55, 56] the authors over-approximate multi-affine hybrid systems by linear hybrid automata [63] for which the reachable sets are easier to compute: parameter estimation is performed in parallel. Similarly, [57] approximates the flow on the faces of the domains and use model-checking techniques to compute the reachable set and perform parameter synthesis. The work [58, 59] uses the approximation from [61] to approximate the regulation networks by a discrete interaction graph on which constraint and satisfiability solvers can be efficiently applied to obtain a valid parameter set.

Finally, we mention some work which aims at handling directly dense parameter sets. On discrete time polynomial population models, the authors of [64, 65] propose a method to compute an under-approximation of a valid parameter space satisfying a set of temporal constraints (in Signal Temporal Logic). A similar approach by invalidation is investigated in [66] and the toolbox ADMIT [67]. Finally, the semi-definite programming methods usually applied for control problems can be used for parameter estimation of biological systems: we refer to [68] for discrete time biochemical systems, and [69, 70] for more general work on ODEs and non-linear hybrid dynamical systems, respectively.

Methods for model validation and verification. Once a parameter valuation is estimated, we want to gain confidence or validate in the determined model. This confidence can be obtained by computing with statistical methods the robustness of the current model with respect to some perturbations [16]. It is also possible to validate the model using either model-checking methods or set reachability analysis. Model checking can be roughly described as checking the satisfaction of a logic formula (describing a desired property) on a mathematical model. With the development of SAT solvers and SMT (SAT-modulo theory) the model-checking approach is an efficient way to validate a model. However, this method is in general more adapted to the validation of qualitative properties. The are numerous work applying the model-checking approach to systems biology. The work of [71] proposes to apply abstract interpretation to perform model checking on models defined in different formalisms: continuous Markov chains, Petri nets, boolean networks and in differential equations. Through the definition of type structures for multiple interactions arising in biochemical reactions, they can generate a satisfiability problem to validate qualitative properties. This approach is integrated into the toolbox Biocham [44]. We note this toolbox also includes a monitoring engine to perform robustness analysis over simulation traces of ODE systems. In addition to this tool, there are numerous other tools for verification of gene regulation networks using convex approximation of the flow as defined in [61]. The genetic

network analyser (GNA) [72, 73] provides a qualitative validation of a discrete abstraction for piecewise affine models of gene regulation networks. An extension for quantitative analysis of piecewise multi-affine models provided in RoVerGeNe [74, 75, 76]. A similar approach is also considered in more recent work [57]. Apart from gene regulation networks, we mention [77] which provides an algorithm for model checking of hybrid automata with explicit flow and algebraic constraints. We remark that outside of the biological context, the theorem prover toolbox Keymaera [78, 79, 80] also handles verification of temporal logic formula on hybrid dynamical systems whose trajectories can be explicitly described. Finally, we recommend the reader to consult an overview of the model-checking approach on discrete models of gene regulation [81] and broader reviews of the application of model-checking and their tools for biochemicals systems [19, 82].

In addition to model-checking based approaches there are a number of tools which can compute reachable sets of differential equation models by set-based simulation methods. However, there are only few which are dedicated to system biology [83, 84]. We note the toolbox Marco [85] which uses the previously cited works on flow convexity for piece-wise multi-affine systems [60, 61] to provide reachability analysis of biological systems. Indeed, most of these results are either applicable only to linear ODE system, or do not handle efficiently models with uncertain parameters. The works which handle both non-linear dynamical systems and uncertain parameters are [29, 34, 86, 83]. The first two tools [29, 34] rely on linearisation and non-linear hybridization, respectively. The last two tools [86, 83] use parametric Bernstein expansion of polynomials to compute over-approximations of reachable sets. Note that the last two tools are the starting steps that lead to the work we propose in Chapter 4. We also mention in the following some results which seem to be promising in their applications for systems biology, even if they do not consider uncertain parameters, but only uncertainty on the initial conditions. The toolbox [87] provides an efficient method to validate hybrid automata with linear dynamics. This method scales with the state-space dimension and accepts uncertain inputs. However, in its current state it does not accept non-linear dynamics. The toolbox [88] accepts non-linear hybrid systems, but do not efficiently handles uncertain parameters. Additionally, we mention some results for reachability of non-linear systems without uncertain parameters [89, 33, 90, 91, 92]. Finally, we recommend a recent review [93] on modelling and validation of biological systems using formal methods.

1.3 Contributions

In this thesis, we present our contributions in addressing three issues in biological systems modelling: the design of a model, its validation, and the model revision problem. In addition to the contributions described in Chapters 3-6, we first give a short mathematical introduction to differential equations, optimal control and hybrid dynamical systems in the Chapter 2.

Model revision. In Chapter 3, we propose an original method to revise a hybrid dynamical system with respect to some observations in the form of intermediate time measures. More concretely, we search for a time varying law better approximating an underlying mechanism (previously described by a constant parameter). We want to achieve this goal without predefined functional form for the sought law. Indeed, we consider the case where we do not possess any additional insight or knowledge. Moreover, we want to avoid classical optimization methods relying on an extensive sampling of a parameter space and numerous simulations. For this purpose, we propose an algorithm based on the optimal control method for hybrid systems proposed in [94]. Experimental results are later exposed in Section 6.1.2 in the chapter describing case studies. This work was accepted in the conference ADHS 2018 [4] (Analysis and Design of Hybrid Systems).

Set-based simulation. In Chapter 4, we extend the previous work of [83]. This work allows to perform reachability analysis of discrete time polynomial system with uncertain parameters. To do so, it relies on the Bernstein expansion, a mathematical tool which can be used for multivariate polynomial optimization over box domain. We contribute in this problem by extending [83] to discrete time piecewise-rational functions, allowing handling a larger panel of biological applications. Moreover, we also propose a few improvements to speed up the actual reachability analysis in some particular cases. In addition, we propose another set-based simulation method using Krivine-Stengle representations, which are another mathematical tool for polynomial optimization. We show that it can be adapted for an efficient application to reachability analysis of discrete-time polynomial systems with uncertain parameters in box domains. Finally, we discuss the complexity of both methods for polynomial optimization, and devise a policy for a more efficient reachability algorithm. The Bernstein reachability method is applied on a case study in Section 6.2. The Bernstein reachability approach and its associated case study were published in the conference HSB 2016 [6] (Hybrid System Biology). The Krivine -Stengle approach has been published in conference ARITH 24 [2] (IEEE Symposium on Computer Arithmetic) as an application to compute upper-bounds of the floating point round-off error. An extended version of this publication has been submitted to the journal IEEE Transaction on Computer [95].

MOEPLA: modelling oriented experimental protocol language. In Chapter 5, we propose a preliminary work on a language to formally specify an experimental protocol while taking into account a model of a mechanism. This language aims to facilitate the interactions between the experimentalists and the modelling team. It also allows automatic generation of a formal framework using the hybrid automaton formalism. Taking advantage of the non-determinism inherent to the hybrid automaton definition, we can either validate a model while taking into account an existing protocol (with its uncertainties), or verify that a future protocol (or therapy) will always be correctly executed. This can be achieved by using the

existing tools for hybrid automaton verification [87, 88, 89], or at least semi-formal validation with simulations [47]. As this work in still under development, it is not yet published.

Case studies. In Chapter 6, we describe three modelling studies which were done along the thesis in parallel to the methodological work. We first give a model of haemoglobin production during the differentiation stages of the an erythroblast into an erythrocyte (also called red blood cells). In the first part of this study, we use a simple exploratory scheme to perform parameter estimation with respect to multiple experimental data sets. The associated results constitute a part of the following reference [3]. In the second part, we propose to use this model as a proof of concept of the model revision method. This model revision with a time varying parameter enables us to better reproduce a considered dataset. We also note that from the inferred solution we derive multiple hypothesis which lead to a meaningful biological interpretation of the time varying parameter as an activation function. This model revision study is accepted in the conference paper [4].

The second case study use the iron homoeostasis model designed in [45]. In this second work, we applied the Bernstein reachability analysis from Chapter 4 to confirm a hypothesis formulated in [45] using exhaustive methods for uncertain parameters and initial sets. These results are published in [6].

The last case study is a preliminary modelling of a recent study of generational effect of low dose and chronic Cadmium intake on the metabolism [96]. In this case study, we propose a first simple model of the oral glucose tolerance test (OGTT) adapted from a previous glucose response model [97]. We also provide multiple parameter estimation associated to different data sets. This work is still on-going, and has yet to be published.

CHAPTER 2

Preliminaries

2.1 Ordinary differential equations 13
2.2 Hybrid dynamical systems 19

In this first chapter we give necessary basic definitions and insights on the two dynamical system mathematical formalisms we use throughout the thesis: *ordinary differential equations* (ODEs) and *hybrid dynamical systems*. The first Section defines multivariate calculus notations, introduces ODEs notations, and describes their main properties. Then, we extend these definitions and properties to parametric ODEs with inputs. Finally, we will use these definitions to formalize the optimal control problem for ODEs. The second Section defines hybrid dynamical systems, that is dynamical system mixing discrete and continuous trajectories. We also provide the definition of an instance of this formalism, namely the hybrid automaton.

2.1 Ordinary differential equations

Multivariate calculus notations. We first recall useful notation on multivariate calculus. For $\mathbf{x} = (x_1, \ldots, x_n) \in \mathbb{R}^n$ and the multi-index $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_n) \in \mathbb{N}^n$, we denote by $\mathbf{x}^{\boldsymbol{\alpha}}$ the product $\prod_{i=1}^n x_i^{\alpha_i}$. We also define $|\boldsymbol{\alpha}| = |\alpha_1| + \ldots + |\alpha_n|$, $\mathbf{0} = (0, \ldots, 0)$ and $\mathbf{1} = (1, \ldots, 1)$.
The notation $\sum_{\boldsymbol{\alpha}}$ is the nested sum $\sum_{\alpha_1} \ldots \sum_{\alpha_n}$. Equivalently $\prod_{\boldsymbol{\alpha}}$ is equal to the nested product $\prod_{\alpha_1} \ldots \prod_{\alpha_n}$.
Given another multi-index $\mathbf{d} = (d_1, \ldots, d_n) \in \mathbb{N}^n$, the inequality $\boldsymbol{\alpha} < \mathbf{d}$ (resp. $\boldsymbol{\alpha} \leq \mathbf{d}$) means that the inequality holds for each sub-index: $\alpha_1 < d_1; \ldots; \alpha_n < d_n$ (resp. $\alpha_1 \leq d_1, \ldots, \alpha_n \leq d_n$). Moreover, the binomial coefficient $\binom{\mathbf{d}}{\boldsymbol{\alpha}}$ is the prod-

uct $\prod_{i=1}^{n} \binom{d_i}{\alpha_i}$.
Let $\mathbf{f} : \mathbb{R}^n \to \mathbb{R}^n$ be a vector field on $\mathbb{R}^n$, then $\max(\mathbf{f}(\mathbf{x}))$ denotes the vector of optima $(\max f_1(\mathbf{x}), \ldots, \max f_n(\mathbf{x}))$.
Let $\mathbb{R}[\mathbf{x}]$ be the vector space of multivariate polynomials. Given $f \in \mathbb{R}[\mathbf{x}]$, we associate a *multi-degree* $\mathbf{d} = (d_1, \ldots, d_n)$ to f, with each d_i standing for the degree of f with respect to the variable x_i. Then, we can write $f(\mathbf{x}) = \sum_{\boldsymbol{\gamma} \leq \mathbf{d}} a_{\boldsymbol{\gamma}} \mathbf{x}^{\boldsymbol{\gamma}}$, with $a_{\boldsymbol{\gamma}}$ (also denoted by $(f)_{\boldsymbol{\gamma}}$) being the coefficients of f in the monomial basis and each $\boldsymbol{\gamma} \in \mathbb{N}^n$ is a multi-index. The degree d of f is given by $d := \max_{\{\boldsymbol{\gamma} : a_{\boldsymbol{\gamma}} \neq 0\}} |\boldsymbol{\gamma}|$.

Example 2.1 (Multivariate polynomial degrees)**.** As an example, if $f(x_1, x_2) = x_1^4 x_2 + x_1 x_2^3$ then $\mathbf{d} = (4, 3)$ and $d = 5$. For the polynomial $f(x, k_1, k_2, k_3) = (2x^2 - x)k_1 + x^2 k_2 + (x^2 - x)k_3$ used later in Section 4.2, one has $\mathbf{d} = (2, 1, 1, 1)$ and $d = 3$.

Ordinary differential equations. Apart from some particular cases, the dynamical systems studied we consider evolve in continuous time. Let t be the variable representing the time. Without loss of generality we can denote $\mathcal{T} = [0, T] \subseteq \mathbb{R}^+$ the time interval, or time domain of the dynamical system under study, such that $t \in \mathcal{T}$.

Let $\mathbf{x} \in X \subseteq \mathbb{R}^n$ be a vector where n is the number of components and the i-th element x_i denotes the i-th variable. The set X is the set which constraints the possible values of the variables $\mathbf{x}$. Then, n-dimensional ODEs are defined by:

$$\dot{\mathbf{x}} = \mathbf{f}(t, \mathbf{x}) \,, \tag{2.1}$$

where $\mathbf{f} : \mathcal{T} \times X \to \mathbb{R}^n$. We call $\mathbf{x}$ the state variables, and X the state space of the system. We recall in the following some main results on ODEs. However, we omit the proofs which can be found in a more in-depth textbook on the subject [98].

We first introduce the notion of continuous trajectory: A trajectory $\xi_{x_0}(t) : \mathcal{T} \times X \to X$ is a function which describes, for a *given initial point* $\mathbf{x}_0$, the evolution of the state variables as function of time. The Cauchy-Lipschitz theorem ensures the existence of a unique trajectory solution $\xi_{x_0}(t)$ of the ODEs (2.1) when $\mathbf{x}(0) = \mathbf{x}_0$. We note that Theorem 2.3 is a *sufficient condition* for the existence and uniqueness of a (maximal) solution. Let us recall the notion of locally Lipschitz for a function f.

Definition 2.2 (Locally Lipschitz)**.** A function $\mathbf{f} : \mathcal{T} \times X \to \mathbb{R}^n$ is locally Lipschitz w.r.t. $\mathbf{x}$ iff for all $(t, \mathbf{x}) \in \mathcal{T} \times X$ there exists a neighbourhood $N(t, \mathbf{x})$, and $L > 0$ such that for all $(t', \mathbf{x}_1)$ and $(t', \mathbf{x}_2) \in N(t, x)$:

$$\left\| \mathbf{f}(t', \mathbf{x}_1) - \mathbf{f}(t', \mathbf{x}_2) \right\| \leq L \left\| \mathbf{x}_1 - \mathbf{x}_2 \right\|$$

A sufficient condition for $\mathbf{f}$ to be locally Lipschitz is to be differentiable on X. We note that if L is a constant valid for all $\mathbf{x} \in X$, then $\mathbf{f}$ is globally Lipschitz. A sufficient condition for $\mathbf{f}$ to be globally Lipschitz is that its derivative is uniformly bounded by L.

Theorem 2.3 (Cauchy-Lipschitz)**.** *If $\mathbf{f} : \mathcal{T} \times X \to \mathbb{R}^n$ is a continuous function locally Lipschitz w.r.t. $\mathbf{x}$. Then, for all $(t_0, \mathbf{x}_0) \in \mathcal{T} \times X$ there exists a unique maximal*[1] *solution of* (2.1) *denoted by $\xi_{\mathbf{x}_0} : [t_0, T[\to \mathbb{R}^n$ with $t_0 < T \leq +\infty$ such that $\xi_{\mathbf{x}_0}(t_0) = \mathbf{x}_0$ and*

$$\forall t \in [t_0, T[, \quad \frac{d\xi_{\mathbf{x}_0}(t)}{dt} = \mathbf{f}(t, \xi_{\mathbf{x}_0}(t))$$

We note that Theorem 2.3 is a *sufficient condition* for the existence and uniqueness of a (maximal) solution.

Example 2.4 (Simple ODE)**.** As a simple example let us take the following ODE:

$$\dot{x} = -4x. \tag{2.2}$$

Then, there exists a unique solution $\xi_{x_0}(t) = x_0 e^{-4t}$ for $t \in [0, +\infty[$ such that $\xi_{x_0}(0) = x_0$. Let us now substitute the constant value 4 with the letter k. The example (2.2) becomes

$$\dot{x} = -kx\,, \tag{2.3}$$

where k is a constant parameter. Consequently, given $k \in \mathbb{R}$ there is still a unique solution to (2.3), $\xi_{x_0,k}(t) = x_0 e^{-kt}$ for $t \in [0, +\infty[$, such that $\xi_{x_0,k}(0) = x_0$.

Parameters and inputs functions. We can generalize Example 2.3 and define *parametric ODEs* as

$$\dot{\mathbf{x}} = \mathbf{f}(t, \mathbf{x}, \mathbf{k}), \tag{2.4}$$

where $\mathbf{k}$ is a vector of m parameters, $K \in \mathbb{R}^m$ is the parameter space, and $\mathbf{f}$ is a function $\mathbf{f} : \mathcal{T} \times X \times K \to \mathbb{R}^n$. Then, for every constant value of $\mathbf{k} \in K$ and a given initial condition $\mathbf{x}_0$, in a similar manner to Theorem 2.3, if $\mathbf{f}$ is a continuous function locally Lipschitz w.r.t. $\mathbf{x}$ and continuous Lipschitz w.r.t. $\mathbf{k}$, then there exists a unique solution. One way to understand it is to represent the parameters as constant state variables $\mathbf{y}$, adding m new equations $\dot{\mathbf{y}} = 0$, and choosing an initial condition $\mathbf{y}_0 = \mathbf{k}$.

We recall that in systems biology, parameters may have multiple interpretations. They can represent a physical constant, for example the Avogadro constant, or is used to approximate an underlying mechanism by a constant for a specific

[1] The solution is said to be maximal if it cannot be continued any further than T

experimental environment, As an example, kinetic rates are often modelled by constants for a given temperature.

In addition to constant parameters, one can model external influences, approximations of internal mechanisms or uncertainties, through an input function $\mathbf{u}(\cdot)$. Inputs can be put into two categories given their interpretations in the model.

– The uncontrolled inputs or perturbations: they can represent the environment, an influence external to the modelled system and not driven by human strategies, or an approximation of some underlying mechanism. Inputs can also be employed to obtain more accurate approximations of subsystems than a constant parameters.

  Some examples of uncontrolled inputs are a fluctuating temperature during an experiment, a noise function representing some uncertainty on the exact behaviour of a system (in which case we may lose determinism).

– The controlled inputs or controls: they model an external action applied to reach a target state given by a human being. The control inputs reflect strategies to achieve some objective. In some cases, the strategy is not only to reach a target, but also to minimise some function, such as a cost function along the trajectories, we then speak of optimal control.

  An example of controlled inputs are the concentration of a drug given to a patient during therapy. Another example would be the necessary number of ill people to treat at a given time to avoid a disease propagation.

Structurally, the input functions can be put into two families: the open loop inputs, and the feedback inputs. An open loop input $\mathbf{u}(\cdot)$ is a function $\mathbf{u} : \mathcal{T} \to U$, where $U \subseteq \mathbb{R}^p$ is the input space, and p the dimension of $\mathbf{u}$. The set of all the functions $\mathbf{u}$ such that $\mathbf{u}(t) \in U$ for $t \in \mathcal{T}$ is also called the *accepted input set* and we denote by it $\mathcal{U}^{\mathcal{T}}$. Then, parametric ODEs (2.4) can be extended to parametric ODEs with inputs

$$\dot{\mathbf{x}} = \mathbf{f}(t, \mathbf{x}, \mathbf{k}, \mathbf{u}(t)) \,, \tag{2.5}$$

and for a fixed function $\mathbf{u}$, (2.5) can be re-written as parametric ODEs $\dot{\mathbf{x}} = \mathbf{F}(t, \mathbf{x}, \mathbf{k})$ with $\mathbf{F}(t, \mathbf{x}, \mathbf{k}) = \mathbf{f}(t, \mathbf{x}, \mathbf{k}, \mathbf{u}(t))$ for all $t \in \mathcal{T}$. Consequently, Theorem 2.3 can be extended to the case with an open loop input function when $\mathbf{F}$ respects the condition from Theorem 2.3.

Theorem 2.5. *Let* $\mathbf{f}$ *be a function defined by* $\mathbf{f} : \mathcal{T} \times X \times K \times U \to \mathbb{R}^n$. *Let* $\mathbf{f}$ *be locally Lipschitz w.r.t.* $\mathbf{x}$*, continuous w.r.t* $\mathbf{u}$*, and let* $\mathbf{u} : \mathcal{T} \to U$ *be continuous. Then, given an initial condition* $(t_0, \mathbf{x}_0) \in \mathcal{T} \times X$ *and a parametrization* $\mathbf{k} \in K$*, there exists a unique maximal trajectory solution* $\xi_{\mathbf{x}_0,\mathbf{k},\mathbf{u}}(t)$ *for* $t \in [t_0, T[$*, with* $t_0 < T \leq +\infty$*, such that* $\xi_{\mathbf{x}_0,\mathbf{k},\mathbf{u}}(t_0) = \mathbf{x}_0$ *and*

$$\forall t \in [t_0, T[\quad , \frac{d\xi_{\mathbf{x}_0,\mathbf{k},\mathbf{u}}(t)}{dt} = f(t, \xi_{\mathbf{x}_0,\mathbf{k},\mathbf{u}}(t), \mathbf{k}, \mathbf{u}(t))$$

A feedback input is a function $\mathbf{u} : X \to U$, where the output depends of the state variables of the system. The set of all the function $\mathbf{u}$ such that $\mathbf{u}(\mathbf{x}) \in U$ for $\mathbf{x} \in X$ is denoted by $\mathcal{U}^X$. An extension of (2.4) for feedback input is:

$$\dot{\mathbf{x}} = \mathbf{f}(t, \mathbf{x}, \mathbf{k}, \mathbf{u}(\mathbf{x}))\,, \tag{2.6}$$

and in a similar way we did for open loop inputs, given an input function $\mathbf{u} : X \to U$, (2.6) can be re-written

$$\dot{\mathbf{x}} = \mathbf{F}(t, \mathbf{x}, \mathbf{k})\,,$$

where $\mathbf{F}(t, \mathbf{x}, \mathbf{k}) = \mathbf{f}(t, \mathbf{x}, \mathbf{k}, \mathbf{u}(\mathbf{x}))$ for all $\mathbf{x} \in X$ and $t \in \mathcal{T}$. Then an extension of Theorem 2.3 to feedback input is as follows.

Theorem 2.6. *Let* $\mathbf{f}$ *be a function defined by* $\mathbf{f} : \mathcal{T} \times X \times K \times U \to \mathbb{R}^n$. *Let* $\mathbf{f}$ *be locally Lipschitz w.r.t.* $\mathbf{x}$ and $\mathbf{u}$. *We recall this means that for all* $(t, \mathbf{x}, \mathbf{u}) \in \mathcal{T} \times X \times K$, $(\mathbf{x}_1, \mathbf{u}_2)$ *and* $(\mathbf{x}_1, \mathbf{u}_2)$ *in the neighbourhood of* $(\mathbf{x}, \mathbf{u})$, *there exists some* $L > 0$ *such that:*

$$\|\mathbf{f}(t, \mathbf{x}_1, \mathbf{u}_1) - \mathbf{f}(t, \mathbf{x}_2, \mathbf{u}_2)\| \leq L(\|\mathbf{x}_1 - \mathbf{x}_2\| + \|\mathbf{u}_1 - \mathbf{u}_2\|)$$

Moreover, let $\mathbf{u} : X \to U$ *be continuous locally Lipschitz w.r.t.* $\mathbf{x}$. *Then, given an initial condition* $(t_0, \mathbf{x}_0) \in \mathcal{T} \times X$ *and a parametrization* $\mathbf{k} \in K$, *there exists a unique maximal trajectory solution* $\xi_{\mathbf{x}_0,\mathbf{k},\mathbf{u}}(t)$ *for* $t \in [t_0, T[$, *with* $t_0 < T \leq +\infty$, *such that* $\xi_{\mathbf{x}_0,\mathbf{k},\mathbf{u}}(t_0) = \mathbf{x}_0$ *and*

$$\forall t \in [t_0, T[\quad \frac{d\xi_{\mathbf{x}_0,\mathbf{k},\mathbf{u}}(t)}{dt} = f(t, \xi_{\mathbf{x}_0,\mathbf{k},\mathbf{u}}(t), \mathbf{k}, \mathbf{u}(\xi_{\mathbf{x}_0,\mathbf{k},\mathbf{u}}(t))).$$

Optimal control problem. As seen previously, input functions can also represent an external control applied on the systems such that it reaches a final target set. Let X_T be that target set at the final time T.

In the open loop case, given a continuous control input $\mathbf{u} \in \mathcal{U}^{\mathcal{T}}$, $\mathbf{x}_0 \in X$, $\mathbf{k} \in K$, and $\mathbf{f} : \mathcal{T} \times X \times K \times U \to \mathbb{R}^n$ a function locally Lipschitz w.r.t. $\mathbf{x}$, continuous in $\mathbf{u}$, then $\xi_{\mathbf{x}_0,\mathbf{k},\mathbf{u}}(\cdot)$ is an admissible trajectory, and $\mathbf{u}(\cdot)$ is its associated admissible control, if $\xi_{\mathbf{x}_0,\mathbf{k},\mathbf{u}}(T) \in X_T$. The set of possible admissible controls can be very large, and two different admissible trajectories can reach the target set. To restrain the set of possible trajectories one can add a cost to minimize along the trajectory or at the final time. For example, we may want to minimize the amplitude of the control. Let us define the cost associated to a given admissible trajectory $\xi_{\mathbf{x}_0,\mathbf{k},\mathbf{u}}$ and control $\mathbf{u}$ as

$$J(t, \xi_{\mathbf{x}_0,\mathbf{k},\mathbf{u}}, \mathbf{u}) = \int_0^T h(t, \xi_{\mathbf{x}_0,\mathbf{k},\mathbf{u}}(t), \mathbf{u}(t))dt + H(T, \xi_{\mathbf{x}_0,\mathbf{k},\mathbf{u}}(T), \mathbf{u}(T))\,, \tag{2.7}$$

where $h(t, \xi_{\mathbf{x}_0,\mathbf{k},\mathbf{u}}(t), \mathbf{u}(t))$ is the running cost on the $[0, T]$ interval, and $H(T, \xi_{\mathbf{x}_0,\mathbf{k},\mathbf{u}}(T), \mathbf{u}(T))$ the cost at final time T. From this cost function we can define the optimal control problem (OCP) in the open loop case.

Definition 2.7 (Open loop OCP)**.** Let $\mathbf{f} : \mathcal{T} \times X \times K \times U \to \mathbb{R}^n$ be a function locally Lipschitz w.r.t. $\mathbf{x}$ and continuous in $\mathbf{u}$. Given $\mathbf{x}_0 \in X$, $\mathbf{k} \in K$, and X_T a compact subset of X the optimal control problem is:

$$\begin{aligned} J^* := \inf_{\mathbf{u}} \quad & \int_0^T h(t, \xi_{\mathbf{x}_0,\mathbf{k},\mathbf{u}}(t), \mathbf{u}(t))dt + H(T, \xi_{\mathbf{x}_0,\mathbf{k},\mathbf{u}}(T), \mathbf{u}(T)) \\ \text{s.t.} \quad & \frac{d\xi_{\mathbf{x}_0,\mathbf{k},\mathbf{u}}(t)}{dt} = f(t, \xi_{\mathbf{x}_0,\mathbf{k},\mathbf{u}}(t), \mathbf{k}, \mathbf{u}(t)) \,, \\ & \xi_{\mathbf{x}_0,\mathbf{k},\mathbf{u}}(t) \in X, \quad \forall t \in \mathcal{T} \,, \\ & \xi_{\mathbf{x}_0,\mathbf{k},\mathbf{u}}(0) = x_0 \,, \\ & \xi_{\mathbf{x}_0,\mathbf{k},\mathbf{u}}(T) \in X_T \,, \\ & \mathbf{u} : T \to U \text{ a continuous function.} \end{aligned} \tag{2.8}$$

We denote respectively by $\mathbf{u}^*$ the optimal control and by $\xi_{\mathbf{x}_0,\mathbf{k},\mathbf{u}^*}$ its associated optimal trajectory.

Similar to the open loop case, we can define the optimal control problem for feedback input $\mathbf{u} : X \to U$.

Definition 2.8 (Feedback OCP)**.** Let $\mathbf{f} : \mathcal{T} \times X \times K \times U \to \mathbb{R}^n$ be a function locally Lipschitz w.r.t. $\mathbf{x}$ and $\mathbf{u}$. Given $\mathbf{x}_0 \in X$, $\mathbf{k} \in K$, and X_T a compact subset of X the optimal control problem is:

$$\begin{aligned} J^* := \inf_{\mathbf{u}} \quad & \int_0^T h(t, \xi_{\mathbf{x}_0,\mathbf{k},\mathbf{u}}(t), \mathbf{u}(\xi_{\mathbf{x}_0,\mathbf{k},\mathbf{u}}(t)))dt \\ & \qquad + H(T, \xi_{\mathbf{x}_0,\mathbf{k},\mathbf{u}}(T), \mathbf{u}(\xi_{\mathbf{x}_0,\mathbf{k},\mathbf{u}}(T))) \,, \\ \text{s.t.} \quad & \frac{d\xi_{\mathbf{x}_0,\mathbf{k},\mathbf{u}}(t)}{dt} = f(t, \xi_{\mathbf{x}_0,\mathbf{k},\mathbf{u}}(t), \mathbf{k}, \mathbf{u}(\xi_{\mathbf{x}_0,\mathbf{k},\mathbf{u}}(t))) \,, \\ & \xi_{\mathbf{x}_0,\mathbf{k},\mathbf{u}}(t) \in X, \quad \forall t \in \mathcal{T} \,, \\ & \xi_{\mathbf{x}_0,\mathbf{k},\mathbf{u}}(0) = x_0 \,, \\ & \xi_{\mathbf{x}_0,\mathbf{k},\mathbf{u}}(T) \in X_T \,, \\ & \mathbf{u} : X \to U \text{ a function continuous locally Lipschitz w.r.t. } \mathbf{x}. \end{aligned}$$

Again, we denote respectively by $\mathbf{u}^*$ the optimal control and by $\xi_{\mathbf{x}_0,\mathbf{k},\mathbf{u}^*}$ its associated optimal trajectory.

The existence of an optimal control is not an easy problem. For a detailed introduction to optimal control (from a variation theory point of view) the reader is referred to the textbook [99] and in particular [99, Section 4.5] for a discussion and review on the existence problem.

Numerical integration. From now on for simplicity of notation, a trajectory is denoted by $\mathbf{x}(t)$ instead of $\xi_{\mathbf{x}_0,\mathbf{k},\mathbf{u}}(t)$, when the initial condition $\mathbf{x}_0$, the parameters $\mathbf{k}$, and the input function $\mathbf{u}$ are not ambiguous.

While for linear ODEs it is possible to determine their exact solution, in most non-linear cases the analytic solution is not available. In these cases, one can compute approximate solutions using numerical simulation methods. Let us now say a few words on numerical simulation. Most methods compute a discrete time approximation $\mathbf{x}_q$ of the continuous solution $\mathbf{x}(t)$ of the ODEs $\dot{x} = f(t, x)$. Among the simplest methods are the explicit numerical integration schemes, such as the forward Euler or Runge-Kutta schemes. They are in general designed from the Taylor expansion of the trajectory $x(t)$ around a given time t. For example, the numerical integration scheme using the forward Euler scheme and a fixed time step Δt and $\mathbf{x}_{q+1}$ denotes the approximate solution at time $t_{q+1} = (q+1)\Delta t$:

$$\mathbf{x}_{q+1} = \mathbf{x}_q + \Delta t \cdot \mathbf{f}(t_q, \mathbf{x}_q) \tag{2.9}$$

The accuracy of these methods is measured by the distance between the exact solution $\mathbf{x}(t + \Delta t)$ at time $t + \Delta t$ and its discrete approximation $\mathbf{x}_{q+1} = \mathbf{x}(t) + \Delta t \cdot \mathbf{f}(t, \mathbf{x}(t))$. In the forward Euler case, $\|\mathbf{x}(t + \Delta t) - \mathbf{x}_{q+1}\| = O(\Delta t)$ and the method is said to be of order 1. Given a sufficiently small time step Δt, the approximate solution converges toward the exact solution. However, finding a correct time step to ensure the stability of these simple algorithms results in inefficient simulations even with very small time step. For these reasons, it is necessary to use more recent algorithms which rely on an adaptive time step which produce an approximation valid for a given accuracy. In practice, when possible one can use the advanced algorithms detailed in [100] to perform the numerical simulation of ODEs.

2.2 Hybrid dynamical systems

Context. Hybrid dynamical systems describe the behaviours of both discrete and continuous components. They are useful to model systems with jumps, fast changes in dynamics, or multiple step processes where dynamics can vary depending on external conditions. While there exists a general notion of hybrid dynamical systems constituted of discrete states with associated continuous dynamics, and a transition function between these states, there is no unique formalism under the name of dynamical hybrid systems. Indeed, depending on the application context the models can be different. The computer science community tends to use complex discrete mechanisms and simpler continuous processes, which reflect the particularities of their applications such as programs or communication protocols. The control theory community deals with more complex continuous part to stay as close as possible to the classical laws modelling the physical world. A PhD thesis of MIT from 1995 [101] give a classification of hybrid dynamical systems and a general definition of hybrid dynamical systems. Another coarser classification of dynamical hybrid systems can be found in a more recent study on the existence of solutions of hybrid dynamical systems [102]. Finally, the 2009 review article from IEEE on hybrid dynamical systems contains an in-depth introduction to the recent

problems and solutions in the field of hybrid dynamical systems from a control theory point of view [40].

Hybrid automaton definitions. In this thesis, we use the hybrid automaton formalism to model hybrid dynamical systems. This formalism has the advantage of being commonly used in many verification tools such as SpaceEx [87, 103], and general enough to be applied in a large set of applications. We will also consider some modifications to the original definition [104], and its generalisation given in the textbook [41, Chapter 1]. Indeed, in the original definition, the continuous trajectories in each discrete states are explicitly given by a function $\mathbf{x}(t)$. However, with the exception of the linear ODEs and few non-linear ODEs, such analytic expressions are hard to obtain. Thus, the continuous trajectories are represented implicitly by the ODEs in each discrete state as in [41]. In our particular case, we use hybrid automata where continuous dynamics are specified by parametric ODEs with input, as defined in previous Section 2.1. The definition of such parametric hybrid automata is as follows.

Definition 2.9 (Parametric hybrid automaton)**.** Let $\mathbf{x} \in \mathbb{R}^n$ be a set of continuous state variables, $\mathbf{k} \in \mathbb{R}^m$ a set of parameters and $\mathbf{u}$ a set of input functions with value in $\mathbb{R}^p$. Given T a final execution time, possibly $+\infty$. We define a parametric n-dimensional hybrid automaton by the tuple $\mathcal{H} := (I, \mathcal{E}, L, X, U, K, S, R, \mathcal{F})$ where:

- $I \subset \mathbb{N}$ is a finite set of indices, used to index discrete states, also called *locations* or *modes*.
- $\mathcal{E} \subseteq I \times I$ is the set of *transitions* $e = (i, j)$ between two modes: i is the source mode, and j the destination mode.
- $L : e \mapsto L_e$ is a labelling function which associates to transition $e = (i, j) \in \mathcal{E}$ its *synchronisation label* L_e. Labels are necessary for the synchronous parallel composition in definition 2.11.
- $X \subseteq \mathbb{R}^n$ is the state space of the continuous variables $\mathbf{x}$. Here, $X = \bigcup_{i \in I} X_i$ where X_i, a compact subset of $\mathbb{R}^n$, is called the *invariant set*, or domain, associated to the mode i.
- $U \subseteq \mathbb{R}^p$ is the set of input values of $\mathcal{H}$. Similarly, $U = \bigcup_{i \in I} U_i$ where U_i, a compact subset of $\mathbb{R}^p$, is the set of input values associated to the mode i.
- $K \subseteq \mathbb{R}^m$ is the set of parameter values of $\mathcal{H}$. Again, $K = \bigcup_{i \in I} K_i$ where $K_i \subseteq \mathbb{R}^m$ is the set of parameter values associated to the mode i.
- $S : e \mapsto S_e$ is a labelling function which associates to transition $e = (i, j) \in \mathcal{E}$ its *guard* $S_e \subseteq X_i$. The guard $S_{(i,j)}$ defines the transition condition from mode i to mode j: for $\mathbf{x} \in X_i$, if $\mathbf{x} \in S_{(i,j)}$ then the system at $\mathbf{x}$ can take transition $e = (i, j)$ from mode i to mode j.

- $R : e \mapsto R_e$ is a labelling function which associates to a transition $e = (i, j) \in \mathcal{E}$ its *reset function* $R_e : S_e \to X_j$. It defines how the continuous variables change after the discrete transition from mode i to mode j.
- $\mathcal{F} : i \mapsto \mathcal{F}_i$ is a labelling function which associates to a mode $i \in I$ its *activity* $\mathcal{F}_i$ defining the continuous behaviour in this mode. In this work, we define $\mathcal{F}_i$ as a continuous dynamical system described by parametric ODEs with inputs as in (2.5) or (2.6). Consequently we define $\mathcal{F}_i$ as:

$$\mathcal{F}_i := ([0, T], X, K_i, U_i, \mathbf{f}_i)\,, \tag{2.10}$$

  with $\mathbf{f}_i : [0, T] \times X \times K_i \times U_i \to \mathbb{R}^n$ a function satisfying the conditions of Theorem 2.3, 2.5, or 2.6 depending of the form of $\mathbf{u}$ (open or close loop).

One can see the discrete structure of a hybrid automaton as a labelled directed graph, in which case we can define a set of labelled edges $E = \{(i, S_{(i,j)}, L_{(i,j)}, R_{(i,j)}, j) \in I \times S(\mathcal{E}) \times L(\mathcal{E}) \times R(\mathcal{E}) \times I \mid e = (i, j) \in \mathcal{E}\}$ in a similar manner to the original paper [104].

We now assume that the input functions $\mathbf{u}_i$ are feedback inputs $\mathbf{u}_i : X_i \to U_i$. The definitions for the open loop case are similar. We now define the trajectories accepted by a hybrid automaton.

Definition 2.10 (Hybrid automaton trajectory)**.** An accepted trajectory or execution of a hybrid automaton is defined by the pair of temporal functions λ and $\mathbf{x}$ for the time domain $[0, T]$, denoted by $(\lambda(\cdot), \mathbf{x}(\cdot))$, with $i = \lambda(t) \in \mathcal{I}$ being the mode[2] at time t, and $\mathbf{x}(t) \in X_i$ being the values of the continuous variables at t. The pair $(\lambda(t), \mathbf{x}(t))$ is also called the hybrid state at time t. From an initial condition $(i_0, \mathbf{x}_0)$, a hybrid automaton trajectory $(\lambda(\cdot), \mathbf{x}(\cdot))$ satisfies the following condition for every time point $t \in [0, T]$:

- Continuous behaviour. If $\lambda(t) = i \in I$, then $\mathbf{x}(t) \in X_i$. Let $\mathbf{u}_i : X_i \to U_i$ be a given input function and $\mathbf{k}_i \in K_i$ a parametrization. Let t_0 be the time point such that either $t_0 = 0$, or $t_0 \leq t$ is the latest time point at which the function λ jumps from some value j to i (corresponding to a transition $e = (j, i) \in \mathcal{E}$). Then, the continuous part $\mathbf{x}(t)$ is determined by the trajectory $\xi_{\mathbf{x}(t_0),\mathbf{u}_i,\mathbf{k}_i}(t)$, which is the solution to

$$\frac{d\xi_{\mathbf{x}(t_0),\mathbf{k}_i,\mathbf{u}_i}(t)}{dt} = \mathbf{f}_i(t, \xi_{\mathbf{x}(t_0),\mathbf{k}_i,\mathbf{u}_i}(t), \mathbf{k}_i, \mathbf{u}_i(\xi_{\mathbf{x}(t_0),\mathbf{k}_i,\mathbf{u}_i}(t)))\,,$$

  for all $t \in [t_0, T[$ such that $\xi_{\mathbf{x}(t_0),\mathbf{u}_i,\mathbf{k}_i}(t) \in X_i$.
- Discrete behaviour. If $\lambda(t) = i \in I$ and $\mathbf{x}(t) \in S_{(i,j)}$, then the system *can* take the discrete transition $e = (j, i) \in \mathcal{E}$, in which case, the trajectory at the right limit time t^+ of t becomes $(\lambda(t^+), \mathbf{x}(t^+)) = (j, R_{(i,j)}(\mathbf{x}(t)))$. If $\mathbf{x}(t) \in S_{(i,j)} \cap \partial X_i$ and $\mathbf{x}(t^+) = \xi_{\mathbf{x}(t_0),\mathbf{u}_i,\mathbf{k}_i}(t^+) \notin X_i$ at the limit right

[2] Note that λ here is a piecewise constant function over time.

instant t^+ of t, then the system *must* take the discrete transition $e = (j, i)$, since it goes outside the invariant set of mode i and thus is no longer allowed to stay in this mode. We say that the guard *enables* the transition, while the invariant *enforces* it.

It is possible for a trajectory such that $(\lambda(t) = i, \mathbf{x}(t))$ to leave invariant X_i at time t^+ without having the possibility to take a transition $i \to j$: $\mathbf{x}(t) \in \partial X_i$ and there is no $j \in I$ s.t. $\mathbf{x}(t) \in S_{(i,j)}$. In this condition the trajectory cannot continue neither by the continuous dynamics, nor by the discrete dynamics: we say that the system reaches a *deadlock*, and that the trajectory is called a *blocking trajectory*. Some results on condition for a hybrid automaton to be non-blocking can be found in [105].

Let us define $Reach(\mathcal{H}) \subset I \times X$ the set of all the possible points reachable in time at most T from any initial condition $(i_0, \mathbf{x}_0) \in I \times X_{i_0}$.

$$\begin{aligned} Reach(\mathcal{H}) = \{&(i, \mathbf{x}) \in I \times X_i \text{ s.t.} \\ &\exists (i_0, \mathbf{x}_0) \in I \times X_{i_0} \text{ and } t \leq T\,, (\lambda(t), \mathbf{x}(t)) = (i, \mathbf{x}) \\ &\text{and } (\lambda(\cdot), \mathbf{x}(\cdot)) \text{ is a hybrid trajectory starting at } (i_0, \mathbf{x}_0) \text{ over } [0, T]\}. \end{aligned} \tag{2.11}$$

Special Behaviours. As discrete transitions happen instantaneously, a hybrid automaton execution can exhibit some non trivial behaviours. Among them we will mention two types: the *Zeno behaviours*, and the *multiple event behaviours*.

A Zeno behaviour happens when the trajectory performs an infinite number of discrete transitions in a finite time interval $[t_1, t_2[$. This behaviour can happen even on simple models such as the bouncing ball [105] or, in the biological context, for gene regulation networks [106]. Singular perturbation theory [106] or Filippov theory [60, 107, 108] are possible ways to solve this problem.

A multiple event behaviour happens when a trajectory can take multiple discrete transitions (possibly an infinity) without letting time elapse. Indeed, a system may take discrete transitions all the time without evolving according to the continuous dynamics.

As one can observe in Definition 2.10, for a given initial condition $(i_0, \mathbf{x}_0) \in I \times X_{i_0}$ a hybrid automaton $\mathcal{H}$ may accept more than one trajectory. Indeed, let the continuous dynamics be deterministic by enforcing the Cauchy-Lipschitz conditions from Theorem 2.6 for the activities defined in Definition 2.9. Then, non-determinism can still be caused by the discrete dynamics, since at a given hybrid state the system can satisfy multiple transition guard conditions.

In the remainder of the thesis, our optimal control applications will only be considered for hybrid automata with deterministic discrete dynamics, and non-determinism in continuous inputs will be resolved by picking a solution to the involved optimization problems.

It is easy to see the following sufficient conditions for a hybrid automaton to be deterministic.

– If $(i, j) \in \mathcal{E}$ and $(i, j') \in \mathcal{E}$ with $j \neq j'$, then $S_{(i,j)} \cap S_{(i,j')} = \emptyset$.

– Given $e = (i, j) \in \mathcal{E}$ and $\mathbf{x} \in S_e$, then $\exists! \mathbf{x}' \in X_j$ s.t. $R_e(\mathbf{x}) = \mathbf{x}'$

For more "semantic" necessary and sufficient condition, the reader is referred to [105].

Synchronous parallel composition. We now define the composition of two hybrid automata. We consider here the simple case without common state variables, input functions, or parameters.

Definition 2.11 (Synchronous parallel composition)**.** We consider two hybrid automata $\mathcal{H}^{(a)} = (I^{(a)}, \mathcal{E}^{(a)}, L^{(a)}, X^{(a)}, U^{(a)}, K^{(a)}, S^{(a)}, R^{(a)}, \mathcal{F}^{(a)})$, and $\mathcal{H}^{(b)} = (I^{(b)}, \mathcal{E}^{(b)}, L^{(b)}, X^{(b)}, U^{(b)}, K^{(b)}, S^{(b)}, R^{(b)}, \mathcal{F}^{(b)})$, with $\mathbf{x}^{(a)} \cap \mathbf{x}^{(b)} = \emptyset$, where $\mathbf{x}^{(a)}$ (resp. $\mathbf{x}^{(b)}$) the continuous variables associated to $\mathcal{H}^{(a)}$ (resp. $\mathcal{H}^{(b)}$). Let t be the common time variable. Then, the synchronous parallel composition of two hybrid automata $\mathcal{H}^{(a)}$ and $\mathcal{H}^{(b)}$ is the hybrid automaton $\mathcal{H}^{(a\|b)} := \mathcal{H}^{(a)} \otimes \mathcal{H}^{(b)}$, where $\otimes$ denotes the parallel composition operator, and defined by:

– The continuous variables of $\mathcal{H}^{(a\|b)}$ are $\mathbf{x}^{(a\|b)} = \mathbf{x}^{(a)} \cdot \mathbf{x}^{(b)}$, where $\cdot$ is the concatenation operation.

– The dimension of $\mathcal{H}^{(a\|b)}$ is $n^{(a\|b)} = n^{(a)} + n^{(b)}$, with $n^{(a)}$ (resp. $n^{(b)}$) the dimension of $\mathcal{H}^{(a)}$ (resp. $\mathcal{H}^{(b)}$).

– The modes of $\mathcal{H}^{(a\|b)}$ are defined by:

$$I^{(a\|b)} := I^{(a)} \times I^{(b)}.$$

– The set of transitions $\mathcal{E}^{(a\|b)} \subset \mathcal{E}^{(a)} \times \mathcal{E}^{(b)}$ of $\mathcal{H}^{(a\|b)}$ as well as their associated synchronisation labels $L^{(a\|b)}(\mathcal{E}^{(a\|b)})$ are defined by the following rules. Let $(j^{(a)}, j^{(b)}) \in I^{(a\|b)}$, then given a discrete state $(i^{(a)}, i^{(b)}) \in I^{(a\|b)}$, its possible transitions and their associated synchronisation labels are:

  – $e^{(a\|b)} = \big((i^{(a)}, i^{(b)}), (j^{(a)}, j^{(b)})\big) \in \mathcal{E}^{(a\|b)}$ if:
  $e^{(a)} = (i^{(a)}, j^{(a)}) \in \mathcal{E}^{(a)}$, and $e^{(b)} = (i^{(b)}, j^{(b)}) \in \mathcal{E}^{(b)}$.
  Then, $L^{(a\|b)}(e^{(a\|b)}) = L^{(a)}(e^{(a)}) \cup L^{(b)}(e^{(b)})$.
  – $e^{(a\|b)} = \big((i^{(a)}, i^{(b)}), (j^{(a)}, i^{(b)})\big) \in \mathcal{E}^{(a\|b)}$ if:
  $e^{(a)} = (i^{(a)}, j^{(a)}) \in \mathcal{E}^{(a)}$, and
  $\forall j^{(b)} \in I^{(b)}$ s.t. $e^{(b)} = (i^{(b)}, j^{(b)}) \in \mathcal{E}^{(b)}$, $L^{(a)}(e^{(a)}) \cap L^{(b)}(e^{(b)}) = \emptyset$.
  Then, $L^{(a\|b)}(e^{(a\|b)}) = L^{(a)}(e^{(a)})$.
  – Similarly for $e^{(a\|b)} = \big((i^{(a)}, i^{(b)}), (i^{(a)}, j^{(b)})\big) \in \mathcal{E}^{(a\|b)}$.

We observe that common synchronisation labels enforce simultaneous transitions, meaning synchronisation, between two hybrid automata[3].

[3] We also note that for synchronised transitions, the guard conditions on the continuous variables will have be to be satisfied simultaneously to enable the transition.

- The state space of $\mathcal{H}^{(a\|b)}$ is defined by $X^{(a\|b)} := X^{(a)} \times X^{(b)}$. Moreover, we associate to each mode $i^{(a\|b)} = (i^{(a)}, i^{(b)}) \in I^{(a\|b)}$ the invariant $X^{(a\|b)}_{i^{(a\|b)}} = X^{(a)}_{i^{(a)}} \times X^{(b)}_{i^{(b)}}$
- The set of possible input values of $\mathcal{H}^{(a\|b)}$ for each mode is defined by:

$$U^{(a\|b)}_{i^{(a\|b)}} := U^{(a)}_{i^{(a)}} \times U^{(b)}_{i^{(b)}}, \quad i^{(a\|b)} = (i^{(a)}, i^{(b)}).$$

- The set of possible parameter values of $\mathcal{H}^{(a\|b)}$ for each mode is defined by:

$$K^{(a\|b)}_{i^{(a\|b)}} := K^{(a)}_{i^{(a)}} \times K^{(b)}_{i^{(b)}}, \quad i^{(a\|b)} = (i^{(a)}, i^{(b)}).$$

- The guard $S^{(a\|b)}_e \subseteq X^{(a)}_{i^{(a)}} \times X^{(b)}_{i^{(b)}}$ of $\mathcal{H}^{(a\|b)}$ associated to the transition:
  - $e = \big((i^{(a)}, i^{(b)}), (j^{(a)}, j^{(b)})\big) \in \mathcal{E}^{(a\|b)}$ is defined by : $S^{(a\|b)}_e = S^{(a)}_{(i^{(a)}, j^{(a)})} \times S^{(b)}_{(i^{(b)}, j^{(b)})}$.
  - $e = \big((i^{(a)}, i^{(b)}), (j^{(a)}, i^{(b)})\big) \in \mathcal{E}^{(a\|b)}$ is defined by : $S^{(a\|b)}_e = S^{(a)}_{(i^{(a)}, j^{(a)})} \times X^{(b)}_{i^{(b)}}$.
  - Similarly for $e = \big((i^{(a)}, i^{(b)}), (i^{(a)}, j^{(b)})\big) \in \mathcal{E}^{(a\|b)}$.
- The reset map $R^{(a\|b)}_e$ of $\mathcal{H}^{(a\|b)}$ associated to the transition:
  - $e = \big((i^{(a)}, i^{(b)}), (j^{(a)}, j^{(b)})\big) \in \mathcal{E}^{(a\|b)}$ is defined by : $R^{(a\|b)}_e = \left(R^{(a)}_{(i^{(a)}, j^{(a)})}, R^{(b)}_{(i^{(b)}, j^{(b)})}\right)$.
  - $e = \big((i^{(a)}, i^{(b)}), (j^{(a)}, i^{(b)})\big) \in \mathcal{E}^{(a\|b)}$ is defined by : $R^{(a\|b)}_e = \left(R^{(a)}_{(i^{(a)}, j^{(a)})}, Id^{(b)}_{i^{(b)}}\right)$, where $Id^{(b)}_{i^{(b)}} : X^{(b)}_{i^{(b)}} \to X^{(b)}_{i^{(b)}}$ is the identity function.
  - Similarly for $e = \big((i^{(a)}, i^{(b)}), (i^{(a)}, j^{(b)})\big) \in \mathcal{E}^{(a\|b)}$.
- The continuous dynamical systems of $\mathcal{H}^{(a\|b)}$ in mode $(i^{(a)}, i^{(b)})$ are $\mathcal{F}^{(a\|b)}_{(i^{(a)}, i^{(b)})}$ defined by:

$$\mathcal{F}^{(a\|b)}_{(i^{(a)}, i^{(b)})} := \left([0, T], X^{(a\|b)}, U^{(a\|b)}_{(i^{(a)}, i^{(b)})}, K^{(a\|b)}_{(i^{(a)}, i^{(b)})}, (\mathbf{f}^{(a)}_{i^{(a)}}, \mathbf{f}^{(b)}_{i^{(b)}})\right).$$

Example 2.12 (Synchronous parallel composition)**.** In this example we perform the synchronous composition (A⊗B)⊗C of the (hybrid) automata A, B, C given on the left side of the Figure 2.1. The automaton A can take its transition in parallel of the automata B and C as it shares no common label. The automata B and C share the label L_2 and thus must take their transition synchronously. The terms c_1, c_2, c_3 represent the conditions (guards in Definition 2.9) associated to the transitions $A_0 \to A_1$, $B_0 \to B_1$, $C_0 \to C_1$, respectively. We recall that we assume the composed hybrid automata do not share any common variables. Consequently, the condition c_1, c_2, c_3 are incomparable.
As A and B share no common label we must consider all the possible combination of transitions. This leads to a 4 states automaton A⊗B. The transitions involving a change on B involve the label L_2. Consequently, the size of (A⊗B)⊗C does not change as all the transitions on B are equivalent (synchronous) to a transition on C. The final automaton of (A⊗B)⊗C is given in the right side of Figure 2.1.

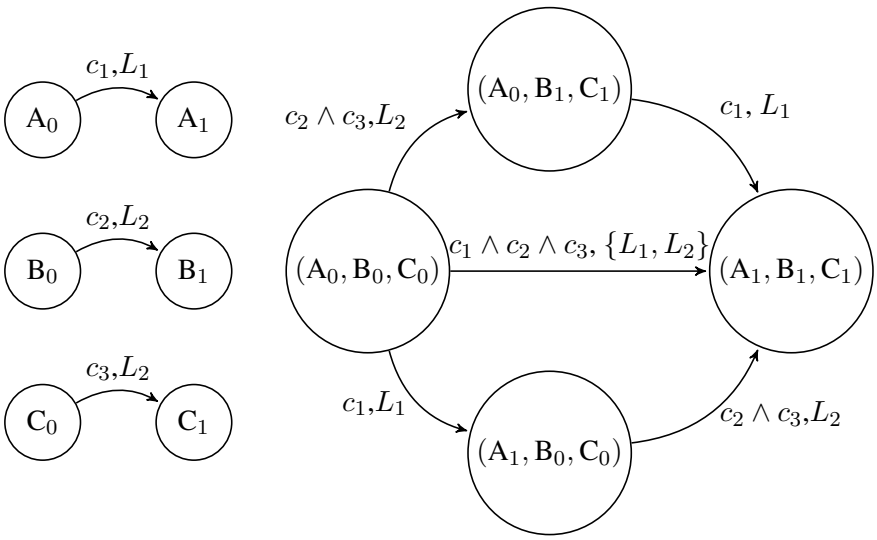

Figure 2.1: Composition of three (hybrid) automata A, B, and C given on the left. This pictures corresponds to Example 2.12.

Optimal control problem on hybrid automata. We now define the control problem for general hybrid automata. Let first define a target set $\mathcal{X}_T$ such that:

$$\mathcal{X}_T = \coprod_{i \in I} X_{T,i}\,, \tag{2.12}$$

where $X_{T,i}$ is a compact subset of X_i and $\coprod$ is the disjoint union operator. The disjoint union can simply be considered as a labelling operation on the set $\mathcal{I}$ of modes.

Let us first consider the open loop control case. Let $(i_0, x_0) \in I \times X_{i_0}$ be an initial condition and $k_i \in K_i$ in mode $i \in I$ a set of parameter values. Then, for a given input function $\mathbf{u}_i : [0,T] \to U_i$ for each mode $i \in I$ and $(\lambda(t), \mathbf{x}(t)) \in I \times X_{\lambda(t)}$ for $t \in [0,T]$ be a trajectory accepted by Definition 2.10, we define a control input function[4] $\mathbf{u}(\cdot)$ as $\mathbf{u}(t) = \mathbf{u}_{\lambda(t)}(t)$ for all $t \in [0,T]$. If $(\lambda(T), \mathbf{x}(T)) \in \mathcal{X}_T$ then $\mathbf{u}$ is an admissible control, and $(\lambda(t), \mathbf{x}(t))\,, t \in [0,T]$ is its associated admissible trajectory. The optimal control problem is then to find a control input $\mathbf{u}(\cdot)$ and a valid trajectory $(\lambda(\cdot), \mathbf{x}(\cdot))$ minimizing a given cost function $J\left((\lambda, \mathbf{x}), \mathbf{u}\right)$, similar to 2.7:

$$J\left((\lambda, \mathbf{x}), \mathbf{u}\right) = \int_0^T h_{\lambda(t)}(t, \mathbf{x}(t), \mathbf{u}(t)) + H_{\lambda(T)}(T, \mathbf{x}(T), \mathbf{u}(T))$$

The hybrid optimal control problem is then:

$$\begin{aligned} J^* := \quad & \inf_{(\lambda,\mathbf{x}),\mathbf{u}} \quad J\left((\lambda, \mathbf{x}), \mathbf{u}\right) \\ & \text{s.t.} \quad (\lambda(t), \mathbf{x}(t))\,, \forall t \in [0,T], \text{ is accepted by Definition 2.10}\,, \\ & \qquad (\lambda(t), \mathbf{x}(t)) \in I \times X_{\lambda(t)}\,, \forall t \in [0,T]\,, \\ & \qquad (\lambda(0), \mathbf{x}(0)) = (i_0, \mathbf{x}_0) \in I \times X_{i_0}\,, \\ & \qquad (\lambda(T), \mathbf{x}(T)) \in \mathcal{X}_T\,, \\ & \qquad \mathbf{u}_i : T \to U_i \text{ continuous functions}\,, i \in \mathcal{I} \\ & \qquad \mathbf{u}(t) = \mathbf{u}_{\lambda(t)}(t), \forall t \in [0,T]. \end{aligned} \tag{2.13}$$

For the feedback control case: we assume that we are given a set of input functions where $\mathbf{u}_i : X_i \to U_i$ for each mode $i \in I$ and $(\lambda(t), \mathbf{x}(t)) \in I \times X_{\lambda(t)}$ is its associated trajectory over $t \in [0,T]$, then we define the control $\mathbf{u}$ as $\mathbf{u}(\mathbf{x}(t)) = \mathbf{u}_{\lambda(t)}(\mathbf{x}(t))$ for all $t \in [0,T]$, and $\mathbf{x}(t) \in X_{\lambda(t)}$. Again, if $(\lambda(T), \mathbf{x}(T)) \in \mathcal{X}_T$ then $\mathbf{u}$ is one admissible control, and $(\lambda(t), \mathbf{x}(t))\,, t \in [0,T]$ is its associated admissible trajectory.

Similarly, the cost function and the hybrid optimal control problem in the feedback case are respectively defined by:

$$J\left((\lambda, \mathbf{x}), \mathbf{u}\right) = \int_0^T h_{\lambda(t)}(t, \mathbf{x}(t), \mathbf{u}(\mathbf{x}(t))) + H_{\lambda(T)}(T, \mathbf{x}(T), \mathbf{u}(\mathbf{x}(T))),$$

[4]Consequently, the function $\mathbf{u}$ is a piecewise function constituted from the input functions $\mathbf{u}_i$ in each mode.

and

$$
\begin{aligned}
J^* := \quad & \inf_{(\lambda,\mathbf{x}),\mathbf{u}} \quad J\left((\lambda,\mathbf{x}),\mathbf{u}\right) \\
& \text{s.t.} \quad (\lambda(t),\mathbf{x}(t))\,, \forall t \in [0,T], \text{ is accepted by Definition 2.10}\,, \\
& \qquad (\lambda(t),\mathbf{x}(t)) \in I \times X_{\lambda(t)}\,, \forall t \in [0,T]\,, \\
& \qquad (\lambda(0),\mathbf{x}(0)) = (i_0,\mathbf{x}_0) \in I \times X_{i_0}\,, \\
& \qquad (\lambda(T),\mathbf{x}(T)) \in \mathcal{X}_T\,, \\
& \qquad \mathbf{u}_i : X_i \to U_i \text{ Lipschitz continuous in } \mathbf{x}\,, i \in \mathcal{I} \\
& \qquad \mathbf{u}(\mathbf{x}(t)) = \mathbf{u}_{\lambda(t)}(\mathbf{x}(t)), \forall t \in [0,T].
\end{aligned}
\tag{2.14}
$$

CHAPTER 3

Occupation measures and optimal control for model revision

3.1 Hybrid Optimal Control using Occupation measures 32
3.1.1 Introduction to occupations measures 32
3.1.2 Solving the hybrid optimal control problem (HOCP) . 35
3.2 Fitting time varying parameters 39
3.2.1 The model revision problem 39
3.2.2 Algorithm and implementation details 41
3.3 Perspectives 42

Context. Mechanistic models in biology generally involve many parameters. The value of a given parameter can be either measured directly in a dedicated experiment (e.g. measurement of a kinetic parameter of a biochemical reaction in enzymology), or inferred from data which provide relationships between parameters and other known biological entities.

As seen in Chapter 1, a basic issue in biological systems modelling is the determination of numerical values for the parameters, or more generally a subset of the parameter space, under which the model agrees to some extent with the available data. We focus on multiple-step experiments, in which a biological system is perturbed or measured during its evolution.

In the biological modelling literature, it is common to synthesise parameters using a Monte-Carlo sampling of the parameter space, which is validated then by numerous simulations. An important effort to formalize and validate the parameter synthesis of biological models has been made in [47, 5, 83, 66]. Other articles

such as [109] or [110] design ODE models satisfying sets of temporal constraints. When model simulation does not reproduce satisfactorily available experimental data, to a degree which depends on data quality, for any admissible parameter value, the model has to be revised. One way of revising the model is to change some parameters into different types of functions of time, reflecting underlying biological mechanisms. We introduce a systematic way, based on formal methods, to study mechanistic biological models in their experimental context and revise parameters to produce conservative results with respect to experimental data. In this work, we consider a problem of model revision, defined as finding time varying laws of parameter evolution that minimizes the error in matching experimental measurements. Informally, it is the following optimization problem:

$$\min_{(\mathbf{x},\mathbf{u})} \quad \sum_{j=1}^{n_{exp}} \mathrm{dist}(\mathbf{m}(\mathbf{x}(T_j)), \mathbf{z}_j) \tag{3.1}$$

where $\mathbf{x}$ is a vector of biological variables, such as concentrations, whose evolution is modelled by trajectories of a biological dynamical system: in our particular context a hybrid automaton as defined Chapter 2. Time varying parameters are represented by the input variables $\mathbf{u}$ (modelling biological parameters) such that $\forall t \in [0,T], \mathbf{u}(t) \in U$. X_0 is the set of initial values of the variables, and the set of pairs $\{(T_j, \mathbf{z}_j)\}_j$ is the set of data points, for $1 \leq j \leq n_{exp}$, in the time frame $[0,T]$. An experimental measurement is a function of the variables $\mathbf{x}$ and is modelled via the function $\mathbf{m}(\mathbf{x})$.

Contributions. The framework of our approach is a mathematical formalization of experimental protocols as hybrid automata, describing biological systems of interest and experiments which are performed on them. An example of such protocol will be studied in Chapter 6, and a general way to generate hybrid automata from such protocol is given Chapter 5. However, the algorithm we provide can be applied to any biological hybrid systems with similar model revision problems.

In this chapter, we address the model revision problem (3.1) by formulating a particular instance of the optimal control problem with intermediate points, which means that the objective function depends on the system trajectory and control inputs at a given set of time points. This problem is then approximated by multiple optimal control problems on hybrid automata. Then, each problem is solved through a reformulation as a hybrid optimal control problem (HOCP) with one final cost. To this end, we apply a recently developed method [94] from the field of certified convex optimization to globally solve these HOCP. The method described in [94] produces piecewise optimal control functions which either may not correspond to biological knowledge of parameter variations or may be difficult to yield coherent and meaningful biological interpretations. Consequently, in order to satisfy realistic constraints on parameters, we use smooth approximations of the generated control input, in order to revise the given model while maintaining good data fitting accuracy. The method is demonstrated later in Chapter 6 on a hybrid

system modelling haemoglobin production whose parameter estimation is studied Section 6.1.1 of Chapter 6.

Related works. We recall that the hybrid formalism has previously been used as an abstraction method to simplify complex mechanisms which are hard to analyse [111, 6], or to represent "jump" evolution such as activation processes in genes regulatory networks for example using the stochastic formalism as in [11].

Optimal control theory and variation theory have been applied to biological systems in several works. Most of them address the classical problem, given in Chapter 2, of finding a correct input such that the system reaches a desired state. For example, one can control drug input such that a patient reaches a healthy state [112], or [8]. Another example is the control of some input in population studies [113]. A detailed review on the use of optimal control in systems biology can be found in [114]. The problem of parameter estimation (as a constant) in presence of multiple data, also called data assimilation, is stated in [114, Chapter 26]. However, none of these techniques for parameter estimation have been applied to the hybrid automata formalism.

The optimal control problem for specific classes of hybrid systems has been investigated in several domains, such as mechanical systems [115] and switched-mode systems [116]. More generally, [117] relies on Dynamic Programming and an extension of Pontryagin's Maximum Principle. However, these approaches need a priori knowledge either on the sequence of discrete transitions, or on the number of visited subsystems. To perform optimal control on hybrid systems, we build our work on the techniques from [94], which proposes a method to obtain a global solution for hybrid systems with state-dependent transitions, without any a priori knowledge on the execution and the sequence of transitions. We refer to [94, Section 1.1] and references therein for more details on optimal control of hybrid systems.

Semidefinite programming (SDP) eases the resolution of hard optimization problems and yields conservative results ensured by positivity certificates. In [118], hierarchies of semidefinite relaxations were introduced for static polynomial optimization. The definition of an infinite-dimensional linear program (LP) over occupation measures, for optimal control problems, was first introduced in [119]. From this infinite-dimensional LP, [120] defines hierarchies of Linear Matrix Inequalities (LMI) relaxations, to synthesise a sequence of polynomial controls converging to the solutions of the optimal control problem. In [121] the authors propose an extension to piecewise affine systems. Our underlying idea of constructing a sub-optimal control with an iterative algorithm is similar to [121, Section 4]. However, we use this scheme to find input functions allowing to reproduce data not only at a final time point but also at intermediate time points.

We make use of the recent method proposed in [94], which relies on occupation measures and a sequence semidefinite relaxations to produce a sequence of polynomial controls converging to the optimal solution of a hybrid optimal con-

trol problem (HOCP). There exists other methods which use occupation measures and LMI relaxations to produce both admissible controls and converging outer-approximations of the backward reachable set[1] (BRS) [122, 70], or the region of attraction[2] (ROA) [123]. Finally, we note that finding a sequence of converging outer-approximations for all valid parameters sets, such as in [68, 69, 83], is another crucial issue in the context of systems biology. When dealing with hybrid systems, an extension of the BRS computation method [124] can be applied to solve this problem.

3.1 Hybrid Optimal Control using Occupation measures

Before presenting our approach for model revision of biological hybrid systems, we give in Section 3.1.1 a short mathematical background on occupation measures inspired from the introductions to occupation measures in both [125] and [122]. In Section 3.1.2 we present recent results from [94] on optimal control for hybrid systems. These results will be used in our own method for model revision.

3.1.1 Introduction to occupations measures

Let first consider the following ODE system:

$$\dot{\mathbf{x}} = \mathbf{f}(t, \mathbf{x}) \,, \tag{3.2}$$

where $\mathbf{x} \in X$ compact subset of $\mathbb{R}^n$, $t \in [0, T]$, and f a Lipschitz continuous non-linear function on $[0, T] \times X$ with values in $\mathbb{R}^n$.

Definition 3.1 (Measures)**.** Let $X \subseteq \mathbb{R}^n$ and A its the Borel σ-algebra built over the subsets P_j of X. We call μ a measure on A the function which assigns to each subset $P_j \in X$ a real scalar such that:

- $\mu(\emptyset) = 0$
- $\forall P_i$ s.t. $P_i \cap P_j = \emptyset$, $(i \neq j)$:

$$\mu(\bigcup_i P_i) = \sum_i \mu(P_i)$$

Additionally, we say that μ is a Radon measure if μ is locally finite and inner regular, meaning for all Borel σ-algebra A of X, $\mu(A) = \sup(\mu(K))$, with K a compact subset of A .

For a compact set $X \subset \mathbb{R}^n$, let $\mathcal{M}(X)$ (resp. $\mathcal{M}_+(X)$) denote the space of unsigned (resp. signed) Radon measures supported on X. Elements of $\mathcal{M}(X)$ can also be seen as bounded linear functionals μ belonging to the dual space $C(X)'$ and

[1] The set of points reachable from a target set while going back in time.
[2] The set of initial conditions leading to an attractor.

acting on continuous function $v \in C(X)$, also called test functions. The image of a test function $v \in C(X)$ by a measure $\mu \in \mathcal{M}(X)$ is given by the following relation:

$$\langle \mu, v \rangle := \int_X v \, d\mu \,. \tag{3.3}$$

Given an initial condition $\mathbf{x}_0 \in X$, we note $\mathbf{x}(t|\mathbf{x}_0)$ the solution of (3.2) for $t \in [0, T]$ such that $\mathbf{x}(0) = \mathbf{x}_0$. We define the *occupation measure* μ as:

$$\mu(A \times B|\mathbf{x}_0) := \int_0^T I_{A\times B}(t, \mathbf{x}(t|\mathbf{x}_0)dt \,,$$

where $A \times B$ is in the Borel σ-algebra of $[0, T] \times X$ and $I_{A\times B}(\cdot)$ being the indicator function, equal to 1 on $A \times B$ and 0 outside. We note that the occupation measure $\mu(A \times B|\mathbf{x}_0)$ provides the total time the trajectory $\mathbf{x}(\cdot|\mathbf{x}_0)$ stays in the subset B for $t \in A$, while the support of $\mu(A \times B|\mathbf{x}_0)$ is the trajectory. We also note that $\frac{1}{T}\mu(A \times B|\mathbf{x}_0)$ is the probability measure of the time the trajectory stays in B. One can makes the relation with the concept of occupation time distribution in stochastic reachability analysis such as defined in [126, Chapter 5].

If the initial condition is unknown we can define the *initial measure* $\mu_0 \in \mathcal{M}(X)$ associated to the initial distribution of $\mathbf{x}_0$ in X. Then, we can define the *average occupation measure* as the Lebesgue integral:

$$\mu(A \times B) := \int_X \mu(A \times B|\mathbf{x}_0) d\mu_0(\mathbf{x}_0) \,. \tag{3.4}$$

Again, we note that the support of $\mu(A \times B)$ is the reachable set on the interval $[0, T]$ for all $\mathbf{x}_0$ in the support of μ_0. Given final set at time T defined as a compact set $X_T \subset X$, we can also define the final measure $\mu_T \in \mathcal{M}(X_T)$ as:

$$\mu_T(B) := \int_X I_B(x(T|\mathbf{x}_0)) d\mu_0(\mathbf{x}_0) \,. \tag{3.5}$$

From (3.3), the image of a test function $v \in C([0, T] \times X)$ by a measure $\mu \in \mathcal{M}([0, T] \times X)$ is given by the following relation:

$$\langle \mu, v \rangle := \int_{[0,T]\times X} v(t, \mathbf{x}) d\mu(t, \mathbf{x}) \,.$$

Moreover, given a test function smooth enough $v \in C^1([0, T] \times X)$ and an initial condition $\mathbf{x}(0) = \mathbf{x}_0 \in X$, we can also write[3] from (3.2):

$$v(T, \mathbf{x}(T|\mathbf{x}_0)) = v(0, \mathbf{x}(0)) + \int_0^T \dot{v}(t, \mathbf{x}(t|\mathbf{x}_0))dt \tag{3.6}$$

[3] We note $\dot{v}$ the total derivative of v: dv/dt.

Let now define a linear operator $\mathcal{L}_f : C^1([0,T] \times X) \to C([0,T] \times X)$ also called *Liouville operator* associated to a dynamic f, applied on a test function v:

$$\mathcal{L}_f v := \frac{\partial v}{\partial t} + \sum_{k=1}^{n} \frac{\partial v}{\partial x_k} f_k(t, \mathbf{x}) \,,$$

where x_k is the k-th element of $\mathbf{x}$. We note that $\dot{v} = \mathcal{L}_f v$ leading from (3.6) to:

$$v(T, \mathbf{x}(T|\mathbf{x}_0)) = v(0, \mathbf{x}(0)) + \int_0^T \mathcal{L}_f v(t, \mathbf{x}(t|\mathbf{x}_0))dt \tag{3.7}$$

Let $\mathcal{L}'_f : C([0,T] \times X)' \to C^1([0,T] \times X)'$ be the adjoint operator of $\mathcal{L}_f$ defined by the relation:

$$\langle \mathcal{L}'_f \mu, v \rangle = \langle \mu, \mathcal{L}_f v \rangle = \int_{[0,T]\times X} \mathcal{L}_f v(t, \mathbf{x}) d\mu(t, \mathbf{x}) \,.$$

Integrating (3.7) with respect to μ_0 accordingly to (3.4) and (3.5), we obtain for all test functions $v \in C^1([0,T] \times X)$:

$$\begin{aligned}
\int_X v(T, \mathbf{x}(T)) d\mu_T(\mathbf{x}) &= \int_X v(0, \mathbf{x}(0)) d\mu_0(\mathbf{x}) + \int_X \int_0^T \mathcal{L}_f v(t, \mathbf{x}(t|\mathbf{x}_0)) d\mu_0(\mathbf{x}) dt \\
\int_X v(T, \mathbf{x}(T)) d\mu_T(\mathbf{x}) &= \int_X v(0, \mathbf{x}(0)) d\mu_0(\mathbf{x}) + \int_X \int_0^T \mathcal{L}_f v(t, \mathbf{x}(t)) d\mu(t, \mathbf{x}) \,.
\end{aligned} \tag{3.8}$$

Using the Dirac measures δ_0 and δ_T, respectively at $t = 0$ and $t = T$, we can transform (3.8) into:

$$\begin{aligned}
\langle \delta_T \otimes \mu_T, v \rangle &= \langle \delta_0 \otimes \mu_0, v \rangle + \langle \mu, \mathcal{L}_f v \rangle \\
\langle \delta_T \otimes \mu_T, v \rangle &= \langle \delta_0 \otimes \mu_0, v \rangle + \langle \mathcal{L}'_f \mu, v \rangle \,,
\end{aligned} \tag{3.9}$$

where $\otimes$ is the product of measures. Finally, as we worked for all test functions $v \in C^1([0,T], X)$ we can write from (3.9) the following *Liouville equation* in the space of the measures:

$$\delta_0 \otimes \mu_0 + \mathcal{L}'_f \mu = \delta_T \otimes \mu_T \,. \tag{3.10}$$

Liouville equation (3.10) is also called continuity equation in statical physics and describes the evolution of a density of particles within a fluid [127]. We note that we transformed a *non-linear* ordinary differential equation over state variables $\mathbf{x} \in \mathbb{R}^n$ into a *linear* partial differential equation in the space of measures. Moreover, using this equation one can express evolution of a family of trajectories with initial conditions described by μ_0. We refer to prior works from Henrion & al. establishing this equation and demonstrating its use in the context of reachability analysis, parameters and controller synthesis [128, 69, 121]. For more details on the recent effort on the application of measures theory and the generalized problem of moments we refer the readers to the books from G. B. Folland [129] and J.B.Lassere [130].

3.1.2 Solving the hybrid optimal control problem (HOCP)

In the previous section we provided a short introduction to the use of measures to transform non-linear ODEs into the linear Liouville equation. In this section, we present a recent work from Zhao & al [94] which provides through the writing of the *hybrid Liouville equations* an approach to address the optimal control problem of hybrid system. This approach described is valid of a given class of hybrid system called *Controlled Hybrid System* (CHS) in [94]. This class differs from the hybrid automata by some change in the formalism and assumptions. These differences ensure that any CHS is deterministic. It is possible to construct a hybrid automaton from a CHS by simply adding to each modes the missing continuous variables. Noting $\phi(\mathbf{x}(t))$ the function which associates to an instantaneous state $\mathbf{x}(t)$ its corresponding mode, Definition 3.2 also ensures that the mode corresponding to $\mathbf{x}(t)$ is unique. Moreover, we obtain the relation $\lambda(t) = \phi(\mathbf{x}(t))$.

Definition 3.2. (Controlled Hybrid System) A controlled hybrid system can be considered as a variation of the hybrid automaton formalism (see Definition 2.9) with the following differences:

- Each mode i has it own dimension n_i such that $X_i \subseteq \mathbb{R}^{n_i}$. Consequently, we denote $\mathbf{x}_i \in \mathbb{R}^{n_i}$ the continuous variables associated to the mode $i \in I$.
- All the guards $S_{(i,\cdot)}$ are disjoint, and $S_{(i,j)} \subseteq \partial X_i$, for each pair of modes i and j, with ∂X_i being the border of the invariant X_i
- The input set $U_i = U$ for each mode i, and the parameters are fixed (all the sets K_i are reduced to singletons).

Moreover, we assume that:

- The initial set is restricted to a single point $\mathbf{x}_0$, with an associated mode i_0.
- The vector fields $\mathbf{f}_i$ are polynomials in $\mathbf{x}_i \in \mathbb{R}^{n_i}$, affine in $\mathbf{u}$, and have a nonzero normal component on the boundary of X_i.

Given measurable functions $\{h_i : [0,T] \times \mathbb{R}^{n_i} \times \mathbb{R}^m \to \mathbb{R}\}_{i\in I}$ and $\{H_i : \mathbb{R}^{n_i} \to \mathbb{R}\}_{i\in I}$, respectively representing the running costs and final costs associated to each mode $i \in I$, a *hybrid optimal control problem* (HOCP) is defined by:

$$\begin{aligned} J^*_{chs} := \inf_{(\mathbf{x},\mathbf{u})} \quad & \int_0^T h_{\lambda(t)}\left(t, \mathbf{x}_{\lambda(t)}(t), \mathbf{u}(t)\right) dt + H_{\lambda(T)}\left(\mathbf{x}_{\lambda(T)}(T)\right). \\ \text{s.t.} \quad & (\lambda(0), \mathbf{x}(0)) = (i_0, \mathbf{x}_0)\,, \\ & (\lambda(T), \mathbf{x}(T)) \in \mathcal{X}_T\,, \\ & \mathbf{u}(t) \in U \;\; \forall t \in [0,T], \\ & (\lambda(t), \mathbf{x}(t))\,, t \in [0,T]\,, \text{ a trajectory of a CHS.} \end{aligned} \tag{3.11}$$

Here $\mathcal{X}_T$ denotes the target set as defined by (2.12). This problem is solved using a hierarchy of semidefinite relaxations as described in [94], and for which

we recall the main ideas. Let first redefine occupation measures and Liouville operator accordingly to our hybrid context.

Given an hybrid trajectory $\xi = (\lambda, \mathbf{x}) : [0,T] \to I \times X$ and a control input $\mathbf{u} : [0,T] \to U$, let define the *occupation measure* in mode $i \in I$ as:

$$\mu^i(A \times B \times C|\mathbf{x}, \mathbf{u}) := \int_0^T I_{A\times B\times C}(t, \mathbf{x}_i(t), \mathbf{u}(t))dt \tag{3.12}$$

for all subsets $A \times B \times C$ in the Borel σ-algebra of subsets of $[0,T] \times X_i \times U$. Similarly to Section 3.1.1, we denote by μ_0^i, μ_T^i the initial and the final occupation measures and respectively, and for all $e \in \mathcal{E}$, the guard occupation measures $\mu^{S_e} \in \mathcal{M}_+([0,T] \times S_e)$ describing the measures associated to distribution of continuous variables in a given guard on $[0,T]$. We recall that if $\mathbf{x}_i(t) \in S_{(i,j)} \subset \partial X_i$, the trajectory have to take the transition $i \to j$: then $\mu^{S_{(i,j)}}$ is also a measure of what leaves[4] a given mode i on the interval $[0,T]$.

Let $\Gamma = (\mu^I, \mu_T^I, \mu^S) \in \mathcal{M}_+([0,T] \times X \times U) \times \mathcal{M}_+(X_T) \times \mathcal{M}_+([0,T] \times S)$. For each mode $i \in \mathcal{I}$, let μ_0^i be defined using the Dirac $\delta_{\mathbf{x}_0}$ if $\mathbf{x}_0 \in X_i$ and 0 otherwise.

The occupation measures technique allows to transform the optimal control problem (3.11) into a linear (but infinite-dimensional) problem (3.15) in the vector space of measures in a similar manner we did for the ODEs in Section 3.1.1. In terms of occupation measures, the cost function in the HOCP (3.11) can be expressed as [94, Lemma 5]

$$J(\mathbf{x}, \mathbf{u}) = \sum_{i\in I} \langle \mu^i(\cdot|\mathbf{x}, \mathbf{u}), h_i \rangle + \sum_{i\in I} \langle \mu_T^i(\cdot|\mathbf{x}), H_i \rangle. \tag{3.13}$$

For each mode i, we define a Liouville operator $\mathcal{L}_i : C^1([0,T] \times X_i) \to C([0,T] \times X_i \times U)$ which acts on test functions v as:

$$\mathcal{L}_i v = \frac{\partial v}{\partial t} + \sum_k \frac{\partial v}{\partial x_k} [\mathbf{f}(t, \mathbf{x}, \mathbf{u})]_k. \tag{3.14}$$

Again, we note $\mathcal{L}_i' : C^1([0,T] \times X_i \times U)' \to C([0,T] \times X_i)'$ the adjoint operator of $\mathcal{L}_i$. Let $R_{*,(i',i)}$ be the pushforward measure associated to the reset map $R_{(i,i')}$ as in [94, Lemma 6], we can write the *hybrid Liouville equation*:

$$\delta_0 \otimes \mu_0^i + \mathcal{L}_i' \mu^i + \sum_{(i',i)\in\mathcal{E}} R_{*,(i',i)} \mu^{S_{(i',i)}} = \delta_T \otimes \mu_T^i + \sum_{(i,i')\in\mathcal{E}} \mu^{S_{(i,i')}}.$$

Consequently, (3.11) can be reformulated as the infinite-dimensional LP [94, Sec-

[4] or enter the mode i for a transition (j, i).

tion 4]:

$$
\begin{aligned}
p^* := \inf_{\Gamma} \quad & \sum_{i\in\mathcal{I}} \langle \mu^i, h_i\rangle + \sum_{i\in\mathcal{I}} \langle \mu_T^i, H_i\rangle \\
\text{s.t.} \quad & \delta_0 \otimes \mu_0^i + \mathcal{L}_i' \mu^i + \sum_{(i',i)\in\mathcal{E}} R_{*,(i',i)} \mu^{\mathcal{S}_{(i',i)}} \\
& = \delta_T \otimes \mu_T^i + \sum_{(i,i')\in\mathcal{E}} \mu^{\mathcal{S}_{(i,i')}}, \quad \forall i \in \mathcal{I} \\
& \mu^i, \mu_T^i, \mu^{S_e} \geq 0, \quad \forall i \in \mathcal{I},\ e \in \mathcal{E}\,,
\end{aligned}
\tag{3.15}
$$

where the infimum is taken over the tuple of measures Γ defined above. Let Γ^* denote the optimal value of the measures associated to the solution p^*.

The optimal solution p^* can be approximated from below through a converging sequence of relaxed problems [94, Theorem 17]. In this particular case, we will focus on the SDP relaxation of the infinite-dimensional primal formulation (3.15).

Before continuing, we first introduce a few definitions and notations on moments, moment matrix, and localizing matrix of a measure. Given a multi-index $\boldsymbol{\alpha} \in \mathbb{N}^n$, let $\mathbf{y}_\mu$ be the moments of a measure μ,

$$
[\mathbf{y}_\mu]_{\boldsymbol{\alpha}} := \int \mathbf{x}^{\boldsymbol{\alpha}} d\mu(\mathbf{x})\,.
$$

Given $r \in \mathbb{N}$, and $p \in \mathbb{R}_r[\mathbf{x}]$, let

$$
\mathrm{L}_{\mathbf{y}_\mu}(p) := \langle \mu, p\rangle = \int \Big(\sum_{|\boldsymbol{\alpha}|\leq r} p_{\boldsymbol{\alpha}} \mathbf{x}^{\boldsymbol{\alpha}} \Big) d\mu.
$$

Given multi-indices $\boldsymbol{\alpha}$ and $\boldsymbol{\beta} \in \mathbb{N}^n$, the moment matrix, $M_r(\mathbf{y}_\mu)$, is defined as: $[M_r(\mathbf{y}_\mu)]_{(\boldsymbol{\alpha},\boldsymbol{\beta})} := [\mathbf{y}_\mu]_{(\boldsymbol{\alpha}+\boldsymbol{\beta})}$, where $|\boldsymbol{\alpha}+\boldsymbol{\beta}| \leq 2r$. Then, let $g \in \mathbb{R}_l[\mathbf{x}]$ be any polynomial with $l < r$, the localizing matrix, $M_r(g, \mathbf{y}_\mu)$, is defined[5] as: $[M_r(g, \mathbf{y}_\mu)]_{(\boldsymbol{\alpha},\boldsymbol{\beta})} := \sum_{|\gamma|\leq l} g_\gamma [\mathbf{y}_\mu]_{(\boldsymbol{\alpha}+\boldsymbol{\beta})}$.

Now, we assume that the sets X_i are semialgebraic, i.e. $X_i := \{\mathbf{x} \in \mathbb{R}^{n_i} : \boldsymbol{g}_{X_i}(\mathbf{x}) \geq 0\}$ for each mode $i \in I$, where $\boldsymbol{g}_{X_i}(\mathbf{x})$ is a vector of polynomials $(\boldsymbol{g}_{X_i})_k \in \mathbb{R}[\mathbf{x}]$ for all $k \in \{1, \ldots, \kappa(X_i)\}$, with $\kappa(\mathcal{X}_i)$ the number of polynomials defining X_i.

Similarly for $X_{T,i} := \{\mathbf{x} \in \mathbb{R}^{n_i} : \boldsymbol{g}_{T,i}(\mathbf{x}) \geq 0\}$, where $g_{T,i}(\mathbf{x})$ is a vector with components $(\boldsymbol{g}_{T,i})_k \in \mathbb{R}[\mathbf{x}]$ for all $k \in \{1, \ldots, \kappa(X_{T,i})\}$. For each transition $e = (i, i') \in \mathcal{E}$, let $S_{(i,i')} := \{\mathbf{x} \in \partial X_i : \boldsymbol{g}_{(i,i')}(\mathbf{x}) \geq 0\}$.

Let the input set be $U := \{\mathbf{u} \in \mathbb{R}^m : \boldsymbol{g}_U(\mathbf{x}) \geq 0\}$, where $\boldsymbol{g}_U(\mathbf{x})$ is a vector with components $(\boldsymbol{g}_U)_k \in \mathbb{R}[\mathbf{u}]$ for each $k \in \{1, \ldots, \kappa(U)\}$. Finally we define $g_\tau = t(T - t)$.

By re-writing, in (3.15), the positivity constraints as semidefinite constraints on moments, and localizing matrices, and then truncating the degree of the moments to $2r$, we obtain a finite dimensional semidefinite program [94, Section 5.1].

[5] We recall that g_γ denotes the coefficient of g at the multi-index γ for the monomial basis.

Now, using $\amalg$ to define the disjoint union of measures, let us denote $\Xi_I := \amalg_{i\in I}\mu^i$, $\Xi_{\mathcal{E}} := \amalg_{e\in\mathcal{E}}\mu^{S_e}$, $\Xi_T := \amalg_{i\in I}\mu^i_T$, and $\Xi := \Xi_I \bigcup \Xi_{\mathcal{E}} \bigcup \Xi_T$ the set of all the measures associated to the problem. Let ψ_i refers to a measure in Ξ, associated to the mode $i \in I$, so either μ^i, $\mu^{S_{(i',i)}}$ $\forall i' \in I$, $\mu^{S_{(i,i')}}$ $\forall i' \in I$, or μ^i_T. Let $\{\mathbf{y}_{r,\psi_i}\}_{\psi_i}$ be the sequence of moments of degree $2r$ for each $\psi_i \in \Xi$, and $\mathbf{y}_r := \{\mathbf{y}_{r,\psi_i}\}_{\psi_i}$. Then, the equality constraints in (3.15) can be approximated as a finite dimensional linear system (3.16), by taking the truncated moments, and localizing matrices:

$$A_r(\mathbf{y}_r) = \mathbf{b}_r \tag{3.16}$$

Then, the relaxed *primal problem* (3.17) is defined by:

$$\begin{aligned}
p^*_r := \inf_{\Xi} \quad & \sum_{i\in\mathcal{I}} \mathrm{L}_{y_{r,\mu^i}}(h_i) + \sum_{i\in\mathcal{I}} \mathrm{L}_{y_{r,\mu^i_T}}(H_i) \\
\text{s.t.} \quad & A_r(\mathbf{y}_r) = \mathbf{b}_r\,, \\
& M_r(\mathbf{y}_{r,\psi_i}) \succeq 0\,,\ \forall \psi_i \in \Xi\,, \\
& M_{r_{X_{ik}}}((\boldsymbol{g}_{X_i})_k, \mathbf{y}_{r,\mu^i}) \succeq 0\,, \\
& \qquad\qquad \forall (k,i) \in \{1,\dots,\kappa(\mathcal{X}_i)\} \times I\,, \\
& M_{r_{U_{ik}}}((\boldsymbol{g}_U)_k, \mathbf{y}_{r,\mu^i}) \succeq 0\,, \\
& \qquad\qquad \forall (k,i) \in \{1,\dots,\kappa(U)\} \times I\,, \\
& M_{r_{S_{ek}}}((\boldsymbol{g}_{S_e})_k, \mathbf{y}_{r,\mu^{S_e}}) \succeq 0\,, \\
& \qquad\qquad \forall (k,\mu^{S_e}) \in \{1,\dots,\kappa(S_e)\} \times \Xi_{\mathcal{E}}\,, \\
& M_{r_{(T,i)_k}}((\boldsymbol{g}_{T,i})_k, \mathbf{y}_{r,\mu^i_T}) \succeq 0\,, \\
& \qquad\qquad \forall (k,\mu^i_T) \in \{1,\dots,\kappa(\mathcal{X}_{T,i})\} \times \Xi_T\,, \\
& M_{r-1}(\boldsymbol{g}_\tau, \mathbf{y}_{r,\psi_i}) \succeq 0\,,\ \forall \psi_i \in \Xi_I \cup \Xi_{\mathcal{E}}\,,
\end{aligned} \tag{3.17}$$

where $r_{X_{ik}} := r - \deg((\boldsymbol{g}_{X_i})_k)/2$, $r_{U_{ik}} := r - \deg((\boldsymbol{g}_U)_k)/2$, $r_{S_{ek}} := r - \deg((\boldsymbol{g}_{S_e})_k)/2$, and $r_{(T,i)_k} := r - \deg((\boldsymbol{g}_{T,i})_k)/2$. Above, $M \succeq 0$ defines a matrix M as positive semidefinite.

Now, from the occupation measures solutions of the relaxed primal (3.17) and their associated moments $\mathbf{y}_\mu$, it is possible to synthesise the control. Given a polynomial $u \in \mathbb{R}[(t,\mathbf{x})]$, and a measure μ we define the vector $b_l(\mathbf{y}_\mu)$ as:

$$[b_l(\mathbf{y}_\mu)]_{\boldsymbol{\alpha}} := \mathrm{L}_{\mathbf{y}_\mu}(\phi(t,\mathbf{x})_{\boldsymbol{\alpha}} \cdot u)\,, \tag{3.18}$$

with $\phi(t,\mathbf{x})_{\boldsymbol{\alpha}}$ a monomial of $\mathbb{R}_l[(t,\mathbf{x})]$ of degree $\boldsymbol{\alpha}$, and $|\boldsymbol{\alpha}| \le l$. Given moment sequences truncated to degree $2d_u$, the optimal control law $\boldsymbol{u}_{d_u,i}$ for each mode $i \in I$ is approximated by a d_u-th order polynomial by solving the following linear system of equations:

$$\left([\mathbf{u}^*_{d_u,i}]_k\right) = \left(M_{d_u}(\mathbf{y}^*_{r,\mu^i})\right)^{-1} b_{d_u,k}(\mathbf{y}^*_{r,\mu^i}), \quad 1 \le k \le m \tag{3.19}$$

for all modes $i \in I$, and where k denotes the control's component. Here $M_{d_u}(\mathbf{y}^*_{r,\mu^i})$ is the truncated moment matrix of order d_u associated to the measure μ^i solution of the primal problem (3.17). This matrix is in general not invertible. However, it is positive semidefinite, and in consequence there exists a pseudoinverse, known as the Moore-Penrose generalized inverse [131, Chapter 7].

3.2 Fitting time varying parameters

3.2.1 The model revision problem

In this section, we solve the model revision problem of a hybrid automaton modelling a biological system together with a set of experiments. Therefore, we provide a method to find time varying parameters of biological hybrid automata, modelled as input functions $\mathbf{u}(t)$, in order to fit the hybrid automaton model to a set of experimental data. Thus, we write our problem as an optimal control problem where desired input functions are the optimal controls which minimize the distance of the results produced by the model and these experimental data (3.1).

We first formulate (3.1) as a particular instance of the optimal control problem on hybrid automata with intermediate points. Then, we propose a first approximation as a set of instances of the optimal control problem on hybrid automata 2.13 defined Chapter 2. However, instead of solving the possibly non-deterministic problem on hybrid automata, we restrict ourselves to a subset of deterministic hybrid automata using the controlled hybrid system formalism from [94] and defined previously. Consequently, we need to solve the hybrid optimal control problem presented Section 3.1.2. The solution is obtained using the previous results from [94, Section 4], of which we summarized the key points in Section 3.1.2. Finally, in Section 3.2.2 we explain the complete algorithm addressing our initial problem.

Let first give a few definitions and notations: experimental measurements, represented by a function $\mathbf{m}(\mathbf{x})$, are performed at given specific times T_j, $1 \leq j \leq n_{exp}$. Let $\mathbf{z}_j$ be the observed value of the experimental measurement at time T_j, then n_{exp} is the number of experimental data points.

Let $X_{T_j,i}$ be compact subsets of X_i, and $X_{T_j} := \coprod_{i \in I} X_{T_j,i}$. As in (2.13), let $(i_0, \mathbf{x}(0)) \in I \times X_0$, and suppose that we are given a set of time values $\{T_j\}$, with $1 \leq j \leq n_{exp}$, and $T_{n_{exp}} = T$.

We say that $((\lambda(\cdot), \mathbf{x}(\cdot)), \mathbf{u}(\cdot))$ is an admissible pair for a problem with intermediate points, if $(\lambda(t), \mathbf{x}(t)) \in I \times X$ is a trajectory of $\mathcal{H}$ accepted by the Definition 2.10, and $(\lambda(T_j), \mathbf{x}(T_j)) \in X_{T_j}$ for all j.

Let $H(\mathbf{x}(T_j))$ be an intermediate cost at time T_j, and $h(t, \mathbf{x}(t), \mathbf{u}(t))$ a running cost for the whole $[0, T]$ interval. The optimal control problem with intermediate

points for the hybrid automaton $\mathcal{H}$ is then:

$$\begin{aligned} J^* := \inf_{(\lambda,\mathbf{x}),\mathbf{u}} \quad & \int_0^T h(t,\mathbf{x}(t),\mathbf{u}(t))dt + \sum_{0\leq j\leq n_{exp}} H\left(\mathbf{x}(T_j)\right) \\ \text{s.t.} \quad & (\lambda(t),\mathbf{x}(t))\,, \forall t\in[0,T] \text{ a trajectory of } \mathcal{H}\,, \\ & (\lambda(t),\mathbf{x}(t))\in I\times X_{\lambda(t)}\,, \forall t\in[0,T]\,, \\ & (\lambda(0),\mathbf{x}(0)) = (i_0,\mathbf{x}_0)\in I\times X_{i_0}\,, \\ & (\lambda(T_j),\mathbf{x}(T_j))\in \mathcal{X}_{T_j}\,,\ \forall 1\leq j\leq n_{exp}\,, \\ & \mathbf{u}: T\to U \text{ continuous functions}\,, \end{aligned} \tag{3.20}$$

In our biological context we search in general to minimize the least square residual:

$$H\left(\mathbf{x}(T_j)\right) = ||\mathbf{m}(\mathbf{x}(T_j)) - \mathbf{z}_j||_2^2.$$

Solving the above problem can entail an excessive computational cost on a large hybrid model (see implementation results on the haemoglobin production model studied in Chapter 6). To reduce this cost, we propose an optimization scheme where we iteratively compute the control for each intermediate time in a greedy way.

Given $1\leq j\leq n_{exp}$, let:

$$J_j(t,\mathbf{x}(t),\mathbf{u}(t)) := \int_{T_{j-1}}^{T_j} h(t,\mathbf{x}(t),\mathbf{u}(t))dt + H\left(\mathbf{x}(T_j)\right),$$

with $T_0 = 0$, and $T_{n_{exp}} = T$, such that

$$J(t,\mathbf{x}(t),\mathbf{u}(t)) = \sum_{1\leq j\leq n_{exp}} J_j(t,\mathbf{x}(t),\mathbf{u}(t)).$$

Noting $(\lambda^{(j)}(t),\mathbf{x}^{(j)}(t))$ a trajectory of a hybrid automaton $\mathcal{H}$ on the interval $\mathcal{T}_j := [T_{j-1},T_j]$, and similarly $\tilde{\mathbf{u}}^{(j)}(t)$ the control on $\mathcal{T}_j$, we consider the following problem as particular instance of (2.13):

$$\begin{aligned} J_j^* := \inf_{(\mathbf{x}^{(j)},\tilde{\mathbf{u}}^{(j)})} & J_j(t,\mathbf{x}^{(j)}(t),\tilde{\mathbf{u}}^{(j)}(t)) \\ \text{s.t.} & \\ & (\lambda^{(j)},\mathbf{x}^{(j)}) \text{ a trajectory of } \mathcal{H} \text{ on } \mathcal{T}_j, \\ & \tilde{\mathbf{u}}^{(j)}(t)\in U\,, \quad \forall t\in\mathcal{T}_j\,, \\ & (\lambda^{(j)}(t),\mathbf{x}^{(j)}(t))\in\mathcal{X}\,, \quad \forall t\in\mathcal{T}_j\,, \\ & \text{if } j=1\,, \\ & \quad (\lambda^{(1)}(0),\mathbf{x}^{(1)}(0)) = (i_0,\mathbf{x}_0)\in I\times X_{i_0}\,, \\ & \text{if } j\geq 2. \\ & \quad (\lambda^{(j)}(T_{j-1}),\mathbf{x}^{(j)}(T_{j-1})) = (\lambda^{(j-1)}(T_{j-1}),\mathbf{x}^{(j-1)}(T_{j-1}))\,, \\ & (\lambda^{(j)}(T_j),\mathbf{x}^{(j)}(T_j))\in X_{T_j}. \end{aligned} \tag{3.21}$$

We note that if a transition $i \to i'$ occurs at the time T_j of the interval $[T_{j-1}, T_j]$, we retain only the left part in the mode i for the next optimization on the interval $[T_j, T_{j+1}]$.

Let $\tilde{\mathbf{u}}(t)$ and $(\lambda(t), \mathbf{x}(t))$ be respectively the control and the trajectory, for $t \in [0, T]$. They are respectively defined by the concatenation of all the controls $\tilde{\mathbf{u}}^{(j)}(t)$ and the trajectories $(\lambda^{(j)}(t), \mathbf{x}^{(j)}(t))$ on the sub-intervals $[T_{j-1}, T_j]$. By construction, $((\lambda(t), \mathbf{x}(t)), \tilde{\mathbf{u}}(t))$ is an admissible pair for (3.20), as $(\lambda(T_j), \mathbf{x}(T_j)) = (\lambda^{(j)}(T_j), \mathbf{x}^{(j)}(T_j)) \in X_{T_j}$.

Remark 3.3. We emphasize that $(\mathbf{x}(t), \tilde{\mathbf{u}}(t))$ is not necessary an optimal solution for (3.20). Moreover, as the optimization problem (3.21) is obtained through a greedy scheme, we have no guarantee that its optimal cost J_j^* is inferior to a given ε. However, our goal is only to find parameter functions satisfying desired error bounds, thus this approximate solution provides a good trade-off between optimality and computation cost. We note that we consider in this work the case of experimental measurements with partial information: if the measurements provide information on the state of all the variables at a time point, then the greedy scheme converge to the optimal solution.

3.2.2 Algorithm and implementation details

Let $(T_j, \mathbf{z}_j)$, $0 \leq j \leq n_{exp}$ be pairs of experimental data points and their measurement time, and we also note i_0, and $\mathbf{x}_0$ the initial mode and initial conditions of the studied hybrid automata $\mathcal{H}$ respectively. Let r be a given starting relaxation degree.

Algorithm 1 finds an admissible solution to (3.20), by solving the reformulation of the optimization problem (3.21) into a HOCP (3.11) for each experimental data point $(T_j, \mathbf{z}_j)$. For each j, the degree of the polynomial control $\tilde{\mathbf{u}}^{(j)}(\mathbf{x}(t), t)$ is determined as the smallest degree such that $||\mathbf{m}(\mathbf{x}(T_j)) - \mathbf{z}_j||_2^2 \leq \varepsilon$. Indeed, in the context of biological system modelling we desire to obtain a control of degree as small as possible to avoid overfitting. Then, for each iteration over j, Algorithm 1 is decomposed in three steps.

The first step is the procedure `HOCP`, associated to an instance of the HOCP (3.11) for j-th pairs $(T_j, \mathbf{z}_j)$. Given a relaxation order $d_r \geq r$, we solve the relaxed primal (3.17). It returns $M_{d_r}(\mathbf{y}_{\mu_i})$, the sequence moment matrices of degree d_r associated to the occupation measure μ_i of each mode $i \in I$. We also obtain $\underline{J}_j^{(d_r)}$ an under approximation of the optimum of (3.21).

The second step is the procedure `Synth`, which returns the admissible control $\tilde{\mathbf{u}}^{(j)}(t, \mathbf{x})$ of degree $d_u \leq d_r$ using a truncated moment matrix $M_{d_u}(\mathbf{y}_{\mu_i})$ as in (3.19).

The third and last step is the procedure `Simu`. It performs the validation that the synthesised control $\tilde{\mathbf{u}}^{(j)}$ yields $||\mathbf{m}(\mathbf{x}(T_j)) - \mathbf{z}_j||_2^2 \leq \varepsilon$. This step is done by approximating the trajectory of the controlled hybrid system using a solver of ODE with discrete events to produce numerical simulations.

If in iteration j, $||\mathbf{m}(\mathbf{x}(T_j)) - \mathbf{z}_j||_2^2 \leq \varepsilon$, then $\mathbf{x}^{(j)}(T_j)$ and the corresponding mode i_f reached at $t = T_j$ by the numerical simulations are the initial conditions for the next iteration $j + 1$.

Otherwise, the `Synth` and `Simu` procedures are repeated while increasing the degree of the synthesised polynomial control until $d_u = d_r$. In case the condition $||\mathbf{m}(\mathbf{x}(T_j)) - \mathbf{z}_j||_2^2 \leq \varepsilon$ is still not satisfied, the relaxation order d_r is increased, and the three steps are repeated.

If $\varepsilon \leq \underline{J}_j^{(d_r)}$ then we are sure that for the given initial condition at step j, there is no control such that $||\mathbf{m}(\mathbf{x}(T_j)) - \mathbf{z}_j||_2^2 \leq \varepsilon$. Consequently, we keep our previous result $\tilde{\mathbf{u}}^{(j)}$ and the corresponding mode i_f reached at $t = T_j$ by the numerical simulations are the initial conditions for the next iteration $j + 1$.

Algorithm 1 hybrid systems model revision algorithm

procedure REVISION($\mathcal{H}, \{(T_j, \mathbf{z}_j)\}_j, i_0, \mathbf{x}_0, \varepsilon, r$)
  $T_{init} = 0$
  for all experimental data $(T_j, \mathbf{z}_j)$ **do**
    $d_u = \mathbf{0}, d_r = r, \texttt{err} = +\infty$
    while $\texttt{err} \geq \varepsilon \wedge \underline{J}_j^{(d_r)} \leq \varepsilon$ **do**
      $\underline{J}_j^{(d_r)}, \mathbf{M}_{d_r}(\mathbf{y}_\mu)$ = `HOCP`($\mathcal{H}, i_0$,...
        ...$\mathbf{x}_0, T_{init}, T_j, \mathbf{z}_j, d_r$)
      while $\texttt{err} \geq \varepsilon$ **and** $d_u \leq d_r$ **do**
        $\tilde{\mathbf{u}}^{(j)}(\mathbf{x}(t), t) = \texttt{Synth}(\mathbf{M}_{d_r}(\mathbf{y}_\mu), d_u)$
        $(i_f, \mathbf{x}^{(j)}(t)) = \texttt{Simu}(\mathcal{H}, \tilde{\mathbf{u}}^{(j)}(\mathbf{x}(t), t)$,...
          ... $i_0, \mathbf{x}_0, T_{init}, T_j)$
        $\texttt{err} = H(x^{(j)}(T_j), \mathbf{z}_j)$
        `increase` d_u
      end while
      `increase` d_r
    end while
    $i_0 = i_f$
    $\mathbf{x}_0 = \mathbf{x}^{(j)}(T_j)$
    $T_{init} = T_j$
  end for
end procedure

3.3 Perspectives

In this chapter we proposed a method for fast model revision of hybrid dynamical system. Even if the result is not optimal, and convergence is not guaranteed using the scheme of Algorithm 1, we observe a good accuracy on the particular example of Section 6.1.2 in Chapter 6. We now propose two ideas to extend this work. The first idea is to search for a infinite LP on occupation measure which directly address

the problem with intermediate costs, which is not possible with the current method from [94]. Such rewriting would provide a method to obtain a solution converging to the optimal control. To do so we propose to extend the formulation of [69] on ODEs using multiple the hybrid Liouville equations restricted to each sub time in $[T_{j-1}, T_j]$, $1 \leq j \leq n_{exp}$. This leads to the following infinite LP:

$$
\begin{aligned}
p^* := \inf_{\{\Gamma_j\}} \quad & \sum_{j \leq n_{exp}} \sum_{i \in \mathcal{I}} \langle \mu^{i,j}, h_{i,j} \rangle + \sum_{j \leq n_{exp}} \sum_{i \in \mathcal{I}} \langle \mu^{i,j}_{T_j}, H_{i,j} \rangle \\
\text{s.t.} \quad & \delta_{T_{(j-1)}} \otimes \mu^{i,j}_{(j-1)} + \mathcal{L}'_{i,j} \mu^{i,j} + \sum_{(i',i) \in \mathcal{E}} R_{*,(i',i),j} \mu^{\mathcal{S}_{(i',i),j}} \\
& = \delta_{T_j} \otimes \mu^{i,j}_{T_j} + \sum_{(i,i') \in \mathcal{E}} \mu^{\mathcal{S}_{(i,i'),j}}, \quad \forall i \in \mathcal{I},\ 1 \leq j \leq n_{exp} \\
& \mu^{i,j}, \mu^{i,j}_{T_j}, \mu^{S_{e,j}} \geq 0, \quad \forall i \in \mathcal{I},\ e \in \mathcal{E},\ 1 \leq j \leq n_{exp}\,,
\end{aligned}
\tag{3.22}
$$

where $1 \leq j \leq n_{exp}$ is the index associated the time interval $[T_{j-1}, T_j]$. We note the number of constraints and occupation measures of the LP (3.22) is multiplied by n_{exp} which entails a great computational cost. In the example studied in Section 6.1.2, such formulation would be computationally expensive, which justifies our approach in Section 3.2.1.

The second idea is to use the scheme of Algorithm 1 to perform parameter synthesis. Indeed in the work [124] the authors propose a method for parameters synthesis using Backward Reachable Set (BRS) computation of hybrid systems. In a similar fashion to [94] they use the hybrid Liouville equation when the system needs to reach a single target at a final time. Using the method of [124] computing BRS for each intermediate points, and then intersecting the results we would obtain a converging over-approximation of the valid parameter set when there are constraints at intermediate time. However, currently the method from [124] is more expensive than [94], and cannot be applied on the example studied in Section 6.1.2.

CHAPTER 4

Set-based simulation for biological models validation

4.1 Set-based simulation 48
4.1.1 Discrete time models 48
4.1.2 Template reachability analysis 49
4.1.3 Approximated dynamics over partitions 51
4.2 The Bernstein expansion based method 57
4.2.1 Polynomial optimization using the Bernstein expansion 57
4.2.2 Validation of piecewise polynomial ODE systems using the Bernstein expansion 64
4.3 The Krivine-Stengle representation based method 69
4.3.1 Polynomial Optimization using K.S representations . . 70
4.3.2 Set-based simulation using K.S representations 73
4.4 Discussion and perspectives 76

Context. In cellular biology, models are often based on the elementary law of chemical reactions or empirical laws for lumped reactions, and expressed in terms of Ordinary Differential Equations (ODEs). However, as we have seen in the previous chapters, unlike models in classical chemistry, most of the parameters in biological models are uncertain, or can greatly vary from one sample (or one individual) to another. For these reasons, modelling in biology involves many round-trips between experimentation and validation of a hypothesis about a biological mechanism formulated by a model. Because these models are uncertain, hypothesis validation is often done with numerous numerical simulations. However, this

simulation-based method is both costly for large parameter spaces and, in addition, not exhaustive. Note that a particularity of biological models is that the sets of parameter values (as well as the set of initial conditions in some cases) can be large. Our objective is to validate a hypothesis on some properties of a given biological model which contains uncertain parameters. Furthermore, the initial conditions, such as the initial concentrations of most species, are not accurately known but lie within identifiable intervals. Therefore, the hypothesis should be validated for all the behaviours generated by such uncertainty, for which reachability analysis is an appropriate tool. Formal verification techniques allow proving properties, with set-based reachability computation techniques, by replacing simulation runs with conservative sets of trajectories. The result of this analysis is the validation of a proposed parameters space for which the model satisfies a set of constraints, proposed by the biologist or coming from experimental results. Thus, in this chapter we propose an approach which can be seen as a complement to the approach based on simulations. It uses discrete time reachability analysis (that is set-based simulation) to formally validate a hypothesis on the model. For polynomials systems, we propose two different methods for reachability analysis: using the Bernstein expansion and the Krivine-Stengle (K.S.) representation.

Contributions. In the first part, we propose an extension of the Bernstein-based method previously developed in [83] which allow tackling uncertain parameters at a small cost. Furthermore, we propose an extension of this Bernstein reachability analysis method to handle polynomial fractions. Another extension, useful to tackle the complex case studies such as the iron homeostasis model developed latter in Chapter 6, is a method for piecewise polynomial approximations of the dynamics and a reachability method for the resulting hybrid dynamics. These approximations and adaptations will be demonstrated in Chapter 6 on the concrete iron homeostasis model, allowing us to validate a hypothesis stated in [5], with an exhaustive analysis over uncertain parameters and initial conditions. In the second part, we propose an alternative method to the Bernstein expansion, namely the K.S. representation, to perform template reachability analysis of polynomial systems. The K.S. representation can be used as a relaxation method to approximate a non-linear optimization problem by a linear optimization problem (LP). We show that in the particular case of parameter space restricted to a box (or a linear transformation of a box), K.S. is another efficient method to perform template reachability analysis. Finally, we discuss its pros and cons compared with the Bernstein-based method.

This chapter is organized as follows. We first formulate, in Section 4.1, the discrete time parametric reachability analysis problem. We introduce, in Section 4.1.3, our method to handle piece-wise polynomial dynamics modelling the approximation of a more complex system over a fixed partition. In Section 4.2, we describe the reachability analysis method using the Bernstein expansion. Then, in Section 4.3, we give details on the second approach based on the K.S. represen-

tation. Finally, in Section 4.4 we discuss the differences between the Bernstein expansion and the K.S. representation approaches and their perspectives.

Related work. To study biochemical systems, such as the Mammalian Cellular Iron Homeostasis (MCIH) model presented in Chapter 6, we base our framework on the `C++` tool `Sapo` [1] for polynomial systems. The work [1] shows that the Bernstein technique is efficient for computing reachable sets of polynomial parametric ODE models. As polynomials arising in biological models are often sparse, using the implicit formulation of the Bernstein expansion in [132] allows us to avoid the explosion of Bernstein coefficients with the dimension and thus improving reachability computation for biological models. Adding our improved computation of the Bernstein coefficients and approximation by piece-wise non-linear models, the current framework can perform reachability analysis on a large class of biological models with switching behaviours. This work can be compared to other work focusing on reachability analysis of non-linear biological models.

There is similar previous work using the Bernstein expansion, such as [133] and [134]. The work [133] allows performing reachability analysis over polytopic sets, instead of bundles of parallelotopes. However, this approach does not directly handle parametric models and is much slower than the current approach due to the conversion from polytopes to boxes. The work [134] uses the Bernstein expansion to compute an LP-relaxation of a polynomial optimization problem (POP), which is then solved over a polytopic set. The technique proposed in [134] can compute reachable sets with high precision using polyhedral templates, but is more expensive than [1] which only needs to compute the parametric Bernstein coefficients once. The work of [134] is also related to the K.S. approach, since it uses a relaxation of a polynomial optimization problem as a LP. Our approach differs from [134] by the ability of our approach to efficiently handle parameters in the optimization. We also mention a recent method using SDP relaxation [128] for discrete time reachability analysis of polynomials.

We also point out the tools, such as [85], [57], [135], which have been developed mainly for reachability analysis of biological models. The work in [85] and [57] is dedicated to piecewise multi-affine models with either conical representations of reachable sets or rectangular abstractions, while [135] focuses on parameters synthesis of piecewise multi-affine models such as gene networks. We also note that the piecewise approximation in our work is similar to the hybridization in [32].

Finally, we can mention well-known tools such as `Flow*` [89] and `Keymaera` [80], for the reachability analysis of non-linear systems. `Flow*` is an efficient tool based on Taylor models for approximating flowpipes in form of unions of boxes, while the Bernstein and K.S. methods computes flowpipes as unions of polytopes. `Flow*` can be used for more general non-linear hybrid models but it does not seem to extend easily to parametric analysis. The tool `Keymaera` [80] uses a different approach: it is a theorem prover based on differential logic. It requires

knowing solutions to differential equations or solving them numerically. It can compute invariants; however in the context of systems biology, it is very useful when interacting with the biologist to provide explicit reachable sets as temporal flowpipes.

Before continuing, it is important to note that the technical contents of this chapter use a lot of multivariate calculus notations defined in Section 2.1.

4.1 Set-based simulation

4.1.1 Discrete time models

In this chapter we consider biological systems modelled by discrete time parametric dynamical systems. The discrete time parametric dynamical systems we study are defined by a set of difference equations:

$$\mathbf{x}_{\tau+1} = \mathbf{f}(\mathbf{x}_\tau, \mathbf{k}) \tag{4.1}$$

where $\tau \in \mathbb{N}$, $\mathbf{x}_\tau \in \mathbb{R}^n$ is the value of the variables $\mathbf{x}$ at the iteration τ, and $\mathbf{k} \in K \subseteq \mathbb{R}^m$ is a set of parameters. A trajectory solution is a sequence $\xi_{\mathbf{x}_0,\mathbf{k}} : \mathbb{N} \to X$ defined for an initial condition $\mathbf{x}_0 \in \mathbb{R}^n$ and a parametrization $\mathbf{k} \in K$. We note that the iteration index $\tau \in \mathbb{N}$ replaces the time variable t. However, as seen in Chapter 2 for the particular case of numerical integration, one can retrieve the value of the continuous time variable t for a given time step Δt.

Discrete time is often used in models from computer science, such as programs where the notions of clock and iteration are natural. However, it can also be used as an approximation of continuous time systems, such as gene regulatory networks (see for example the work of [136] or [72]). Discrete time models can also be obtained though explicit numerical integration of ODEs system as seen in Chapter 2. In the following, we assume that we are given a model as a discrete time parametric dynamical system.

Example 4.1 (2D discrete time system of Lotka-Volterra)**.** A discretization of the Lotka-Volterra continuous time model (LV-ODE) taken from [137, 138] describing the oscillations of a prey-predator interaction system is given in (LV-discrete). We assume the following values for the parameters: $(\alpha, \beta, \delta, \gamma) = (2/3, 4/3, 1, 1)$. To obtain this discretization we apply the forward Euler numerical integration scheme with a time step $\Delta t = 0.01$. Figure 4.1 shows 1600 iterations of the resulting discrete time dynamical system (LV-discrete). In Section 4.4 we demonstrate a method for reachability analysis on the model (LV-ODE).

$$\begin{aligned} \dot{x} &= x(\alpha - \beta y) \\ \dot{y} &= -y(\gamma - \delta x) \end{aligned} \quad \text{(LV-ODE)} \qquad \begin{aligned} x_{\tau+1} &= x_\tau(1 + (\alpha - \beta y_\tau)\Delta t) \\ y_{\tau+1} &= y_\tau(1 - (\gamma - \delta x_\tau)\Delta t) \end{aligned} \quad \text{(LV-discrete)}$$

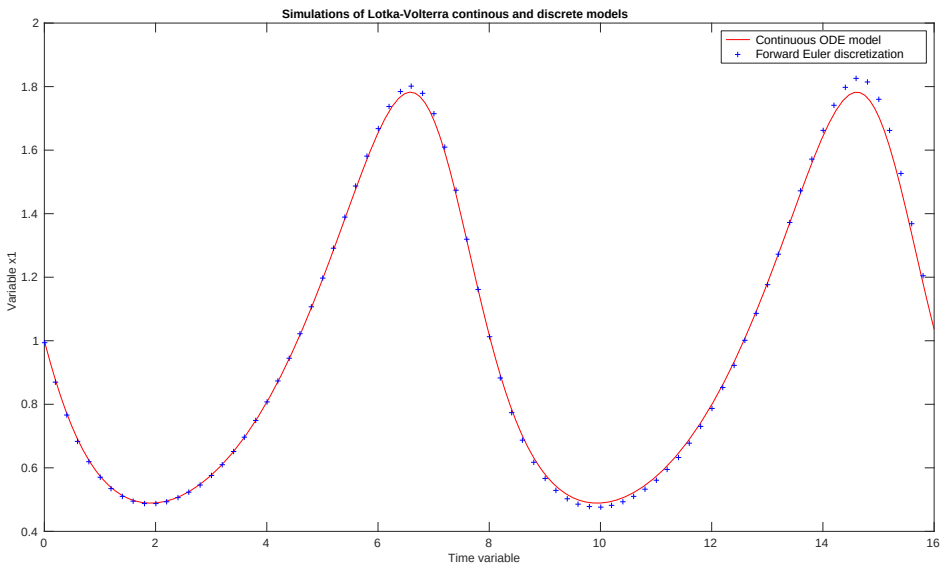

Figure 4.1: Simulations of the ODE model (LV-ODE) in red, and the discrete time model (LV-discrete) in blue (one out of 20 points are plotted).

4.1.2 Template reachability analysis

Reachability analysis. In this chapter, we extend the previous work of [83] on the computation of flowpipe and reachable sets for parametric discrete time dynamical systems. We first recall the general definition of the reachable set $\mathcal{R}(\mathbf{x}_0, T, \mathbf{k})$ of the dynamical system (4.1), up to iteration $T > 0$ (possibly $+\infty$), for given an initial condition $\mathbf{x}_0 \in \mathbb{R}^n$ and a parametrization $\mathbf{k} \in K$:

$$\mathcal{R}(\mathbf{x}_0, T, \mathbf{k}) := \{\mathbf{x} \mid \mathbf{x} = \xi_{\mathbf{x}_0,\mathbf{k}}(\tau)\,, \forall \tau \text{ s.t. } 0 \leq \tau \leq T\}\,. \tag{4.2}$$

For a given initial set $X_0 \subset \mathbb{R}^n$ and for all parameter values in K, we obtain the reachable set:

$$\mathcal{R}(X_0, T, K) := \bigcup_{\mathbf{k} \in K} \bigcup_{\mathbf{x}_0 \in X_0} \mathcal{R}(\mathbf{x}_0, T, \mathbf{k}). \tag{4.3}$$

We can put the reachable set computation methods into two categories:

– *Depth-first methods*: they compute the reachable set up to iteration T for each sample point $(\mathbf{x}_0, \mathbf{k}) \in X_0 \times K$ independently, and then iterate over the set of all the possible points. These methods are also called trajectory-based reachability analysis. As they rely on numerical integration tools they are very fast. In practice these techniques are often used when searching for a counter-example [139], as one counter-example is sufficient, or in combination with sensibility analysis [47] or flow abstraction to produce a flow-pipe approximation [140]. A non-exhaustive list of trajectory based methods is [141, 47, 139].

– *Breadth-first methods*: Instead of computing the reachable set for each initial condition and parameter set, the breadth-first (also called set-based) approach uses set operations to compute the set $X_{\tau+1}$ (that is the image of X_τ for each iteration τ), given an initial set X_0 and a parameter space K. The set operations are more expensive than operations on points and for non-linear vector fields $\mathbf{f}$ a precise approximation of $X_{\tau+1}$ has in general a computational cost exponential in dimension. However, the set-based methods produce conservative results which correspond to an exhaustive simulation. A non-exhaustive list of set-based reachability tools is [87, 89, 83, 88].

In the general case, the problem of computing the reachable set is undecidable [63]. However, we note that in the particular case of polynomial discrete-time dynamical system, with X_0 and K two semi-algebraic compact sets, the computation of the reachable set at a finite time T is a decidable problem. Indeed, it can be formulated as a logic formula which is decidable [142] and exactly solved by quantifier elimination methods as in [143]. However, this solution using quantifier elimination remains too expensive and for this reason we use template reachability, a method to compute an over-approximation of the reachable set.

Template reachability analysis. We now assume that X_0 is a compact subset of $\mathbb{R}^n$. Exactly computing $X_{\tau+1} = \mathbf{f}(X_\tau, K)$, that is the image of X_τ by $\mathbf{f}$, can be challenging, since an exact representation of the image is not known. We thus apply the following compact over-approximation:

$$\{\mathbf{x} \in \mathbb{R}^n \mid \mathbf{g}(\mathbf{x}) \leq \mathbf{c}\,, \quad \forall \mathbf{x} \in X_{\tau+1}\}\,, \tag{4.4}$$

where $\mathbf{g}(\mathbf{x})$ is a vector of polynomials $\mathbf{g}_i(\mathbf{x}) \in \mathbb{R}[\mathbf{x}]$ for all $1 \leq i \leq p$. Hence these p polynomial constraints define a compact semi-algebraic subset of $\mathbb{R}^n$. We call $\mathbf{c} \in \mathbb{R}^p$ the offset vector associated to the constraint system. The vector of polynomials $\mathbf{g}$ defines a fixed template if the coefficients of each polynomial $\mathbf{g}_i$ are fixed. Thus, by fixing the template $\mathbf{g}(\mathbf{x})$ and defining $\mathbf{c}_{\tau+1} \in \mathbb{R}^p$ as the optimum of the polynomial optimization problem:

$$\begin{aligned} \mathbf{c}_{\tau+1} &= \max_{\mathbf{x} \in X_{\tau+1}} (\mathbf{g}(\mathbf{x})) \\ &= \max_{(\mathbf{y},\mathbf{k}) \in X_\tau \times K} (\mathbf{g}(\mathbf{f}(\mathbf{y},\mathbf{k}))\,, \end{aligned} \tag{4.5}$$

we can obtain the following $\hat{X}_{\tau+1}$ as a tight template over-approximation of $X_{\tau+1}$:

$$\hat{X}_{\tau+1} = \{\mathbf{x} \in \mathbb{R}^n \mid \mathbf{g}(\mathbf{x}) \leq \mathbf{c}_{\tau+1}\}\,. \tag{4.6}$$

Given a semi-algebraic template $\mathbf{g}$ as defined in (4.5), let $\mathcal{O}_{\mathbf{g}}(X)$ be the operator computing a tight over-approximation $\hat{X}$ of a set X. An over-approximation $\hat{\mathcal{R}}(X_0, T, K)$ of the reachable set is obtained by the following recurrence: $\hat{X}_0 = \mathcal{O}_{\mathbf{g}}(X_0)$ and $\hat{X}_{\tau+1} = \mathcal{O}_{\mathbf{g}}(\mathbf{f}(\hat{X}_\tau, K))$. The approximated reachable set at a fixed time T is then :

$$\hat{\mathcal{R}}(X_0, T, K) := \bigcup_{0 \leq \tau \leq T} \hat{X}_\tau.$$

Polyhedral template. We now define a particular type of semi-algebraic templates we use in the remainder of this chapter: polyhedral templates. A template $\mathbf{g}$ is called polyhedral template[1] if it is constituted of a system of p linear constraints. Let $\Gamma \in M_{p,n}(\mathbb{R})$ denote the template matrix such that $\mathbf{g}(\mathbf{x}) = \Gamma\mathbf{x}$. In a similar way to (4.4), given $\mathbf{c} \in \mathbb{R}^p$ we say that $(\Gamma, \mathbf{c})$ is a template over-approximation of a set X if for all $\mathbf{x} \in X$, $\Gamma\mathbf{x} \leq \mathbf{c}$. It follows from (4.5) that given a template matrix Γ, a tight template over-approximation of a set X can be the polyhedral set:

$$\hat{X} := \{\, \mathbf{x} \in \mathbb{R}^n \mid \Gamma\mathbf{x} \leq \max_{\mathbf{x}\in X}(\Gamma\mathbf{x}) \,\}. \tag{4.7}$$

Finally, using the above-described template reachability scheme we obtain a particular case of (4.6) for a polyhedral template $(\Gamma, \mathbf{c}_{\tau+1})$:

$$\hat{X}_{\tau+1} = \{\, \mathbf{x} \in \mathbb{R}^n \mid \Gamma\mathbf{x} \leq \mathbf{c}_{\tau+1} = \max_{(\mathbf{y},\mathbf{k})\in X_\tau \times K} \Gamma\mathbf{f}(\mathbf{y},\mathbf{k}) \,\}. \tag{4.8}$$

It is clear that the complexity of the reachability algorithm will depend on the resolution of the optimization problem $\max_{(\mathbf{y},\mathbf{k})\in X_\tau \times K} \Gamma\mathbf{f}(\mathbf{y},\mathbf{k})$. If X_τ is a polyhedron and $\mathbf{f}$ is a linear vector field, such problems can efficiently be solved using linear programming. Otherwise, if $\mathbf{f}$ is non-linear the complexity of the optimization increases drastically. Indeed, one cannot rely on local optimization method, since the result may be inferior to the maximum, leading to a non-conservative approximation of the set. In the following, we propose two approaches based on two different global optimization techniques providing upper-bounds of the optimum, and thus a conservative over-approximation.

Example 4.2 (Box template)**.** Let us consider X a compact subset of $\mathbb{R}^2$ as shown in Figure 4.2. The template matrix Γ_{box} defines axis parallel constraints, and the template over-approximation $(\Gamma_{\text{box}}, \mathbf{c})$ as shown in Figure 4.2 is the box template over-approximation of X.

$$\Gamma_{\text{box}} = \begin{pmatrix} \Gamma_1 \\ \Gamma_2 \\ \Gamma_3 \\ \Gamma_4 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ -1 & 0 \\ 0 & -1 \end{pmatrix} \tag{Γ_{box}}$$

4.1.3 Approximated dynamics over partitions

State-space partition. In this section we consider how to handle discrete time dynamical systems defined using piecewise continuous functions. Such models can be obtained through local approximation of the dynamics over a fixed partition of the state space. For example, in the particular case of the Mammalian

[1] We assume that $\Gamma\mathbf{x} \leq \mathbf{c}$ defines a compact subset of $\mathbb{R}^n$.

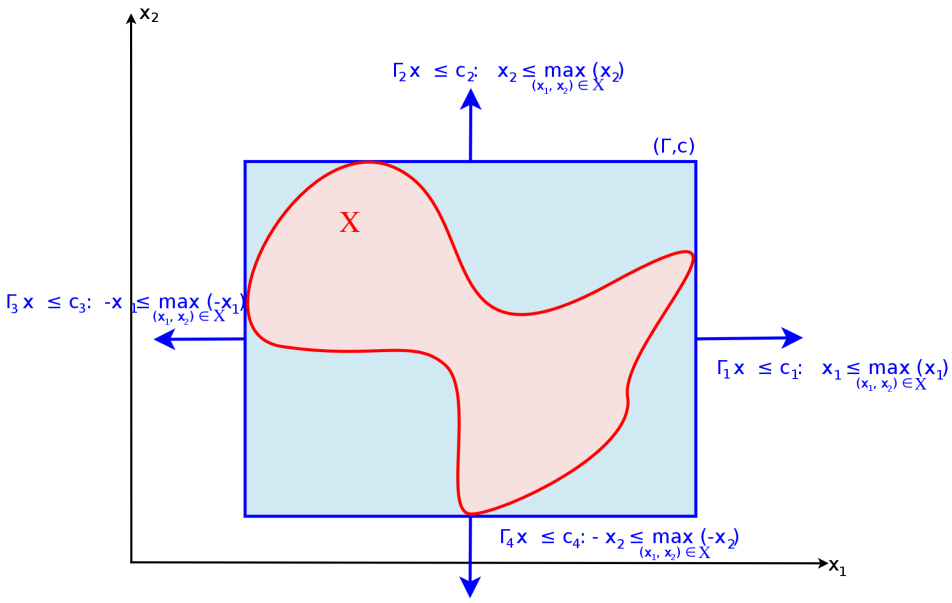

Figure 4.2: Tight over-approximation $(\Gamma_{\text{box}}, \mathbf{c})$ (in blue) of a set X (in red) by a box template Γ_{box} given in Example 4.2.

Cellular Iron Homeostasis (MCIH) model studied in Chapter 6, we approximate sigmoid functions in three affine pieces. Piecewise models can also be obtained using model reduction techniques such as [111] where models of multi-scale biochemical reaction networks are decomposed in piecewise polynomial systems.

Let h be a linear constraint of the form $\mathbf{a} \cdot \mathbf{x} \leq c$ where $\mathbf{a} \in \mathbb{R}^n$ and $c \in \mathbb{R}$. We denote by $\neg h$ the constraint $\mathbf{a} \cdot \mathbf{x} > c$, the negation of the constraint[2] h. For simplicity of notation and presentation we use h to denote both the constraint and the half-space defined by this constraint. We also assume that our state-space is a compact subset of $\mathbb{R}^n$: for example a box of $\mathbb{R}^n$ large enough such that all the trajectories stay inside for the considered time interval.

Let $H = \{h_j(\mathbf{x}) \mid j \in \{1, \ldots, n_H\}\}$ be a set of n_H linear constraints. The constraints of H partition the state space into q non-empty compact subsets of $\mathbb{R}^n$, $\{\mathcal{N}_i\}_{1 \leq i \leq q}$, called domains of the partition.

Remark 4.3. We recall that we are considering models with discrete time dynamics. Under such dynamics, it is possible, in one discrete step, to jump across multiple domains of the partition (see Figure 4.3). In this work, we accept such behaviours as we are considering the general case of discrete time dynamics. However, while the discrete time model is produced through an approximation of a continuous time model, such "jump" are considered as a bad behaviours and must be detected. At the end of this section we present how to detect such behaviours using our tree representation.

[2] In practice we perform computations with non-strict inequalities to keep compact sets.

We note that under the assumption of the existence of the behaviours described in Remark 4.3, we can provide the following translation of a partition into a hybrid system with discrete time dynamics as in Definition 4.4. We define q modes, one for each domain $\mathcal{N}_i$, $i \in [1, q]$: the invariant associated to a mode i is the whole state space. Through Remark 4.3, we note that a mode i has an urgent[3] transition to any other mode $i' \neq i$, with an associated guard being the domain $\mathcal{N}_{i'}$. The reset associated to each transition is the identity.

Definition 4.4 (Approximation map)**.** Given a continuous function $f : \mathbb{R}^n \to \mathbb{R}$, we define as its approximation map associated to the list of constraint H the piecewise continuous function $\mathcal{A}_f : \mathbb{R}^n \to \mathbb{R}$ such that:

$$\mathcal{A}^{(f)}(\mathbf{x}) := \mathcal{A}_i^{(f)}(\mathbf{x}), \quad \forall \mathbf{x} \in \mathcal{N}_i\,,$$

with each $\mathcal{A}_i^{(f)} : \mathbb{R}^n \to \mathbb{R}$ a polynomial (or rational function) approximating f for $\mathbf{x} \in \mathcal{N}_i$.

We first mention that in this work we do not focus on how the approximation is obtained (be it by Taylor approximation, or by global interpolation methods for example): we assume that it is given. We also do not take into consideration the approximation error, as it can be handled latter as a parameter of the system.

Binary space partition trees (BSPT). To deal with these hybrid dynamics, we need an efficient way to encode this partition of the state space, such that it is easy to locate a set on the partition during its evolution under the dynamics. To this end, we use the Binary space partition tree (BSPT) techniques [144]. Each node of a BSPT is associated with a non-empty set (also called domain of the node) that is defined by a conjunction of linear constraints. For simplicity of notation and presentation, instead of saying that the domain of a node intersects with some set, we simply that say a node intersects with some set and we use the notation $\mathcal{N}$ to denote both the nodes and its associated domain.

We recall that $H = \{h_i \mid i \in \{1, \ldots, n_H\}\}$ is the set of constraints involved in the piecewise approximations of the dynamics. The domain associated with the root node $\top$ of the tree is the whole state space. For each leaf node, if adding one constraint from H splits the corresponding domain into two non-empty sub-domains, we create two child nodes from it, each corresponds to a sub-domain. This constraint is called the *splitting constraint* of the node. We repeat the same procedure until all the constraints in H are added.

We associate with the root node the highest rank n_H, and a child node has a rank smaller than its parent by 1. Once the BSPT is constructed, each leaf of this tree corresponds to a mode i associated to each subset $\mathcal{N}_i$ of the partition of the state space by H.

[3]The transition *must* be taken if the conditions are satisfied.

If a model involves multiple approximation maps, it is possible to statically build a single BSPT representing all the possible discrete modes. Two BSPT can be fused by replacing each leaves of the first tree by the top node of the second tree. Then, starting from the node of smallest rank of the first tree (previously the leaves) we go downward and eliminate the branches corresponding to redundant constraints and the nodes with empty domains. Such a method corresponds to the simple parallel composition of the two associated hybrid systems. In this work, the composition is performed statically at the beginning. In the recent work [145], a similar approach has been applied to approximate non-linear dynamics using piecewise linear ODEs. Unlike our work, they perform the composition of each approximation map on the fly in the hybrid automata tool SpaceEx [87].

Example 4.5 (Partition of $\mathbb{R}^2$)**.** In this example we consider a partition of the $\mathbb{R}^2$ by three constraints h_1, h_2, h_3 defined by $x_1 \leq c_1$, $x_1 \leq c_2$ and $x_2 \leq c_3$ respectively. We also assume that $c_1 < c_2$. Then the state-space is partitioned in 6 domains $\mathcal{N}_1$ to $\mathcal{N}_6$. We associate to a domain $\mathcal{N}_i$ a vector filed $\mathcal{A}_i^{(f)}(\mathbf{x})$. This partition of $\mathbb{R}^2$ is represented in Figure 4.3. Its associated BSPT is given in Figure 4.4.

Set localization. We now explain how to locate a given polyhedral compact set X on the partition, and identify the nodes that intersect with X. Note that such sets X can be sets generated by the dynamics of the system. Indeed, even considering the worst case of Remark 4.3 we expect some continuity in the flow to speed up this localization operation in the general case. Thus, we can expect that the successor set $X_{\tau+1}$ is in the same locations or in the locations that are adjacent to the positions of the current set X_τ. Consequently, during the reachability process, instead of starting the search from the root of the tree, we can start from the current or adjacent locations. We call these starting nodes the guess nodes. However, at the initial step $\tau = 0$, since there is no previous information, we start our search from the root of the tree. The search algorithm, named `locating`, consists of the following two steps.

In the first step, we search for the node $\mathcal{N}$ of the lowest rank which strictly contains X, that is $X \subseteq \mathcal{N}$. To do so, we test if there is no node in the guess list satisfying this condition. If this is the case, we go upward in the tree and test their parent nodes, until the condition is satisfied. If the node $\mathcal{N}$ found this way is a leaf, $\mathcal{N}$ is the only node containing X, the algorithm returns $(\mathcal{N}, X)$ and stops. Otherwise, it proceeds to the second step starting from $\mathcal{N}$. One can easily see that the efficiency of this algorithm depends on the ordering of the constraints when building the BSPT. In any case using guess lists is not worse than starting from the root. A good heuristic when building the tree, is to have the constraints the most susceptible to be crossed at the lowest ranks. This way, the previous scheme avoid going too much upward in the tree before finding the first strictly containing node.

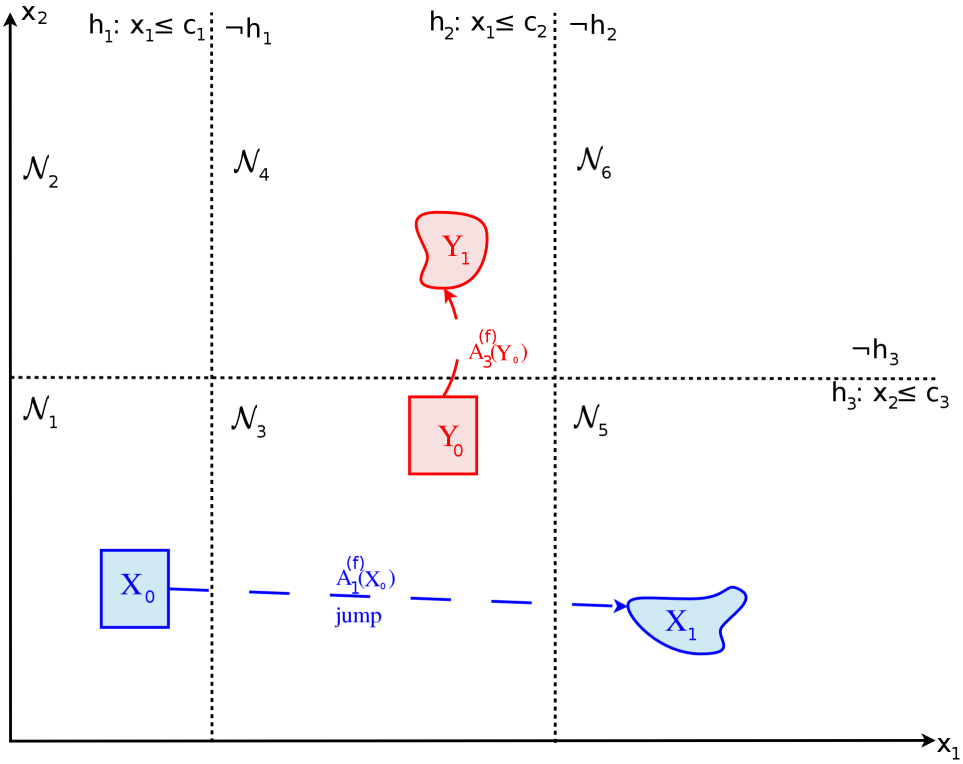

Figure 4.3: Partition of the state space by three constraints h_1, h_2, and h_3 as defined in Example 4.5. The associated BSPT is given in Figure 4.4. The set X_0 is located in the domain $\mathcal{N}_1$. Its image by $\mathcal{A}_1^{(f)}$ is X_1 located in domain $\mathcal{N}_5$ which is not adjacent to $\mathcal{N}_1$: we say that X_0 performed a jump.

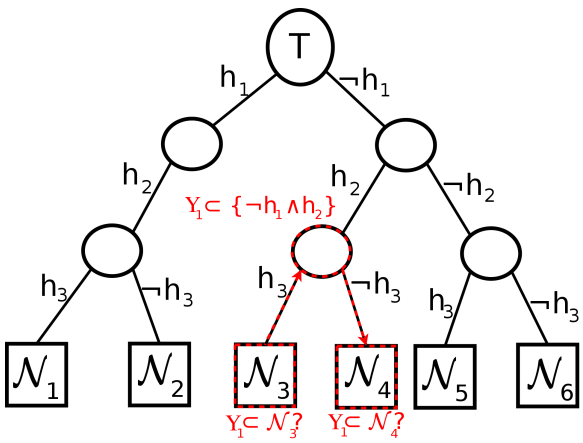

Figure 4.4: BSPT associated to the partition proposed in Example 4.5 and shown in Figure 4.3. In red we show the step associated to the localization of the set Y_1 as described in Example 4.6

This ordering can be preprocessed using the traces of a few simulations to have an estimation of the number of time each constraint is crossed.

The second step of the algorithm is a breadth-first search starting from the node $\mathcal{N}$, in order to obtain $(\mathcal{L}, \mathcal{X})$ where $\mathcal{L}$ is the set of nodes with non-empty intersections, and $\mathcal{X}$ is the set of corresponding intersections (stored as a set of convex polyhedra). This step is a recursive procedure applied to a set $\mathcal{L}$ of nodes intersecting with X, until we reach the leaves. Initially, $\mathcal{L} = \{\mathcal{N}\}$, and the set $\mathcal{X}$ contains only $\{X\}$. Then, until $\mathcal{L}$ contains only leaves when the algorithm returns $(\mathcal{L}, \mathcal{X})$ and then stops, the following procedure is iterated. By construction, at any iteration, all the nodes in $\mathcal{L}$ have the same rank and therefore have the same splitting constraint, denoted by h:

- If $X \subseteq h$, then X intersects the left child of every nodes in $\mathcal{L}$, then $\mathcal{L} = \texttt{left_children}(\mathcal{L})$.
- If $X \subseteq \neg h$, then X intersects the right child of every nodes in $\mathcal{L}$, then $\mathcal{L} = \texttt{right_children}(\mathcal{L})$.
- If X satisfies none of the two above conditions, then X intersects both all the right and left children of the nodes in $\mathcal{L}$. Then, $\mathcal{L} = \texttt{left_children}(\mathcal{L}) \cup \texttt{right_children}(\mathcal{L})$, and the algorithm updates $\mathcal{X} = \{\mathcal{X} \cap h\} \cup \{\mathcal{X} \cap \neg h\}$, where $\{\mathcal{X} \cap h\}$ and $\{\mathcal{X} \cap \neg h\}$ are polyhedra resulting from intersecting each set in $\mathcal{X}$ with the half-spaces corresponding to the constraints h and $\neg h$ respectively.

Example 4.6 (Locating a set)**.** In this example, we continue with the partition of $\mathbb{R}^2$ defined in Example 4.5 and represented in Figure 4.3. We want to locate Y_1 the image of Y_0 by the vector field $\mathcal{A}_3^{(f)}$ (see Figure 4.3). Let consider the BSPT given in Figure 4.4 as a representation of the partition of Example 4.5. We know that $\mathrm{Y}_0 \subset \mathcal{N}_3$, therefore we first guess that $\mathrm{Y}_1 \subseteq \mathcal{N}_3$. As it is false, we go upward in the tree and test if $\mathrm{Y}_1 \subseteq \{\neg h_1 \wedge h_2\}$ which is true. Consequently, we know that either $\mathrm{Y}_1 \subseteq \mathcal{N}_4$ or $\mathcal{Y}_1$ is intersecting both $\mathcal{N}_3$ and $\mathcal{N}_4$. As $\mathrm{Y}_1 \subset \neg\{h_3\}$ we know that $\mathrm{Y}_1 \subseteq \mathcal{N}_4$ which terminates the localisation search.

Detecting jumps. We now want to define a procedure to test if a set $X_{\tau+1}$ is in a domain adjacent to the position of the previous set X_τ, or if it has "jumped" across multiple domains as shown in Figure 4.3. In case a jump is detected, a roll-back to iteration τ may be performed and, for example in the case of discrete time systems coming from numerical integration, the time step may be reduced.

First we define a constraint h in the form $\mathbf{a}\mathbf{x} \leq c$, as tangent to a compact set $\mathcal{N}$ if $\max_{x \in \mathcal{N}}(\mathbf{a} \cdot \mathbf{x}) = c$. We recall that we defined a node $\mathcal{N}$ by a conjunction of constraints, possibly redundant. For each leaf $\mathcal{N}_i$ we identify all its constituting constraints that are not tangent.

Let us assume that X_τ is located in the node $\mathcal{N}_i$ at step τ. Let us also assume that $X_{\tau+1}$ intersects a node $\mathcal{N}_{i'}$ at step $\tau+1$ such that there is a constraint h that is crossed: meaning h appears in the definition of $\mathcal{N}_i$ and $\neg h$ in the one of $\mathcal{N}_{i'}$. If h is not tangent to $\mathcal{N}_i$ then $X_{\tau+1}$ has performed a "jump" (see Figure 4.3).

4.2 The Bernstein expansion based method

As seen in the previous section, one can compute an over-approximation of a reachable set using template reachability analysis. One of the main difficulties is the optimization problem $\max_{(\mathbf{y},\mathbf{k})\in X\times K} \Gamma\mathbf{f}(\mathbf{y},\mathbf{k})$ (in the particular case of polyhedral templates) to compute a tight template over-approximation.

In this section, we present the first method for reachability analysis based on the Bernstein expansion of polynomials. This approach has been previously studied in [83] and we propose to extend it to a larger class of dynamical systems.

We first present the necessary background on the Bernstein expansion and its application to non-linear optimization. Then we introduce our contribution for the reachability analysis of discrete time piece-wise polynomial dynamical systems.

4.2.1 Polynomial optimization using the Bernstein expansion

Bernstein expansion. Bernstein expansion is the reformulation of a polynomial from its expression in the canonical basis to its expression in the Bernstein basis. It was first proposed by S.N. Bernstein, at the beginning of the XXth century, in a proof of the Weirstrass theorem [146]. Its use for the enclosure of univariate polynomials was later proposed in [147] and extended in [148] to multivariate polynomials. We recommend two recent surveys [149, 150] on the Bernstein expansion, its computation, and applications. We recall from [148, Theorem 2] the definition of the multivariate Bernstein expansion:

Definition 4.7 (Multivariate Bernstein expansion)**.** Given a multivariate polynomial f and a degree $\mathbf{l} \geq \mathbf{d}$ with $\mathbf{d}$ being the multi-degree of f, then for $\mathbf{x} \in [0,1]^n$, the Bernstein expansion of multi-degree $\mathbf{l}$ of f is given by:

$$f(\mathbf{x}) = \sum_{\gamma} a_{\gamma}\mathbf{x}^{\gamma} = \sum_{\boldsymbol{\alpha}\leq\mathbf{l}} b_{\boldsymbol{\alpha}}^{(f)}\mathbf{B}_{\mathbf{l},\boldsymbol{\alpha}}(\mathbf{x}). \tag{4.9}$$

where $b_{\boldsymbol{\alpha}}^{(f)}$ (also denoted by $b_{\boldsymbol{\alpha}}$ when there is no confusion) are the Bernstein coefficients (of multi-degree $\mathbf{l}$) of f, and $\mathbf{B}_{\mathbf{l},\boldsymbol{\alpha}}(\mathbf{x})$ are the Bernstein basis polynomials defined by $\mathbf{B}_{\mathbf{l},\boldsymbol{\alpha}}(\mathbf{x}) := \prod_{i=1}^{n} B_{l_i,\alpha_i}(x_i)$ and $B_{l_i,\alpha_i}(x_i) := \binom{l_i}{\alpha_i} x_i^{\alpha_i}(1-x_i)^{l_i-\alpha_i}$. The Bernstein coefficients are given by the following formulas:

$$b_{\boldsymbol{\alpha}} = \sum_{\boldsymbol{\beta}\leq\boldsymbol{\alpha}} \frac{\binom{\boldsymbol{\alpha}}{\boldsymbol{\beta}}}{\binom{\mathbf{l}}{\boldsymbol{\beta}}} a_{\boldsymbol{\beta}}, \quad \mathbf{0} \leq \boldsymbol{\alpha} \leq \mathbf{l}. \tag{4.10}$$

We call the set of points $(\boldsymbol{\alpha}/\mathbf{l}, b_{\boldsymbol{\alpha}}) \in \mathbb{R}^{n+1}$ the Bernstein control points associated to the Bernstein expansion of multi-degree $\mathbf{l}$.

Example 4.8 (Bernstein expansion)**.** The Bernstein coefficients up to degree $l = 10$ of the univariate polynomial $f(x) = x^5 + 3x^2$ for $x \in [-1, 1]$ are:

$$\mathbf{b} = \{2, \frac{9}{5}, \frac{44}{45}, \frac{1}{5}, \frac{-26}{105}, \frac{-1}{3}, \frac{-16}{105}, \frac{1}{5}, \frac{34}{45}, \frac{9}{5}, 4\}.$$

The associated Bernstein control points are represented in the Figure 4.5. We note that the maximum of the coefficients is 4. As it is achieved in $x = 1$, we know the solution is optimal, which can be observed in Figure 4.5. However, the minimum of the Bernstein coefficients $\mathbf{b}$, which is $\frac{-1}{3}$, is not the minimum of $f(x)$ over $[-1, 1]$.

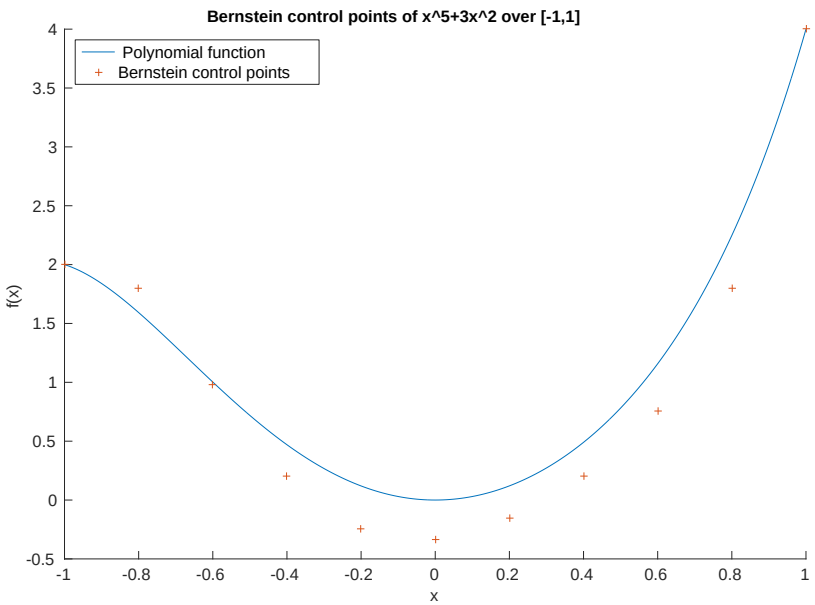

Figure 4.5: Representation in blue of the polynomial $f(x) = x^5 + 3x^2$ given in Example 4.8, for $x \in [-1, 1]$. In red, we show its Bernstein control points for $l = 10$.

The Bernstein expansion having numerous properties, we give only the ones necessary for our reachability purpose in Section 4.2.2. For a more exhaustive introduction to the Bernstein expansion, as well as some proof of the basic properties, we refer the interested reader to [151].

Property 4.9 (Cardinality [151, (3.14)])**.** The number of Bernstein coefficients in the Bernstein expansion of multi-degree $\mathbf{l}$ is equal to $(\mathbf{l} + \mathbf{1})^{\mathbf{1}} = \prod_{i=1}^{n}(l_i + 1)$.

Property 4.10 (Linearity [151, (3.2.3)])**.** Given two polynomials f_1 and f_2,

$$b_{\boldsymbol{\alpha}}^{(cf_1+f_2)} = cb_{\boldsymbol{\alpha}}^{(f_1)} + b_{\boldsymbol{\alpha}}^{(f_2)} \ , \quad \forall c \in \mathbb{R},$$

where the Bernstein expansions with same multi-degrees are considered.

Property 4.11 (Enclosure [151, (3.2.4)])**.** The minimum (resp. maximum) of a polynomial f over $[0,1]^n$ can be lower bounded (resp. upper bounded) by the minimum (resp. maximum) of its Bernstein coefficients:

$$\min_{\boldsymbol{\alpha}\leq \mathbf{l}} b_{\boldsymbol{\alpha}} \leq f(\mathbf{x}) \leq \max_{\boldsymbol{\alpha}\leq \mathbf{l}} b_{\boldsymbol{\alpha}}, \ \ \forall \mathbf{x} \in [0,1]^n \ .$$

Property 4.12 (Sharpness [151, (3.2.5)])**.** If the minimum (resp. maximum) of the $b_{\boldsymbol{\alpha}}$ is reached for $\boldsymbol{\alpha}$ in a corner of the box $[0,l_1] \times \cdots \times [0,l_n]$, then $b_{\boldsymbol{\alpha}}$ is the minimum (resp. maximum) of f over $[0,1]^n$.

Property 4.9 gives the maximal computational cost needed to find a lower or a upper bound of $f(\mathbf{x})$, $\forall \mathbf{x} \in [0,1]^n$ and for a Bernstein expansion of fixed multi-degree $\mathbf{l}$. Property 4.11 is used to provide a lower (resp. upper) bound of $\min_{\mathbf{x}\in[0,1]^n} f(\mathbf{x})$ (resp. $\max$), while Property 4.12 allows us to determine if the given bound is optimal. The convergence toward the optimum can be obtained either through subdivision of the domain $[0,1]^n$, or through an increase of the multi-degree $\mathbf{l}$. The work of [152, 153, 154] proposes multiple subdivision schemes and gives their associated convergence rates. The convergence rate in degree elevation can be found in the original work on the multivariate Bernstein expansion [148] or more recently in [155]. Finally, we recall from [133] an upper bound on the distance between a polynomial f and its Bernstein enclosure:

Lemma 4.13. *Let $\pi_{\mathbf{l}}^{(f)} : \mathbb{R}^n \to \mathbb{R}$ be the piecewise linear function defined by the Bernstein control point of a polynomial f at a given degree $\mathbf{l}$. Then, the following inequation holds for all $\mathbf{x} \in [0,1]^n$:*

$$\|f(\mathbf{x}) - \pi_{\mathbf{l}}^{(f)}(\mathbf{x})\|_\infty \leq \max_{\mathbf{x}\in[0,1]^n; i,j\in\{1\ldots n\}} |\partial_i \partial_j f(\mathbf{x})| \tag{4.11}$$

In the recent years multiple methods have been developed to compute efficiently the Bernstein coefficients. In this work we use two of them depending of the situation: the matrix computation [65] and the implicit form [132]. The first method is the method originally implemented in the software `sapo` [65] and efficiently computes *all the Bernstein coefficients* using matrix operations. The second method uses an implicit representation of the Bernstein coefficients. Indeed, it can be observed, in [132], that the Bernstein coefficients of a multivariate monomial $\mathbf{x}^{\boldsymbol{\gamma}}$ up to a degree $\mathbf{l}$ can be expressed as the product of the coefficient of each univariate monomial $x_i^{\gamma_i}$ constituting $\mathbf{x}^{\boldsymbol{\gamma}}$:

$$b_{\boldsymbol{\alpha}}^{(\mathbf{x}^{\boldsymbol{\gamma}})} = \prod_{i\leq n} b_{\boldsymbol{\alpha}_i}^{(x_i^{\gamma_i})} \tag{4.12}$$

Consequently, instead of storing all the Bernstein coefficients of $\mathbf{x}^\gamma$ it suffices to compute the Bernstein coefficients of the univariate monomials which are less numerous[4]. If needed, the Bernstein coefficients of $\mathbf{x}^\gamma$ (and any polynomial thanks to Property 4.10) can be computed independently and on the fly at a lesser cost.

Example 4.14 (Implicit representation)**.** Let us consider the 2 dimensional polynomial $\mathbf{f}(x_1, x_2) = x_1^3 x_2 - 3x_1 x_2^2$ for $(x_1, x_2) \in [-2, 0.5]^2$. Then, the implicit representation for $\mathbf{l} = (3, 3)$ is given by:

$$1 \begin{pmatrix} -2 \\ -7/6 \\ -1/3 \\ 1/2 \end{pmatrix} \begin{pmatrix} -8 & 2 & -1/2 & 1/8 \\ 16 & -4 & 1 & -1/4 \\ 28/3 & -7/3 & 7/12 & -7/48 \\ 8/3 & -2/3 & 1/6 & -1/24 \\ -4 & 1 & -1/4 & 1/16 \end{pmatrix} + -3 \begin{pmatrix} 4 \\ 2/3 \\ -7/12 \\ 1/4 \end{pmatrix} \begin{pmatrix} -2 & -7/6 & -1/3 & 1/2 \\ -8 & -14/3 & -4/3 & 2 \\ -4/3 & -7/9 & -2/9 & 1/3 \\ 7/6 & 49/72 & 7/36 & -7/24 \\ -1/2 & -7/24 & -1/12 & 1/8 \end{pmatrix}$$

Finally, the Bernstein coefficients are obtained by combing both matrices with the coefficients 1 and -3:

$$\begin{pmatrix} 40 & 10 & 5 & -25/4 \\ 40/3 & 0 & 5/4 & -55/48 \\ -5/6 & -65/24 & -5/12 & 5/6 \\ -5/2 & 15/8 & 0 & -5/16 \end{pmatrix}$$

This representation is especially useful to determine the enclosure as one does not always need to compute explicitly all the Bernstein coefficients to find their maximum or minimum. Indeed, the work of A.P. Smith [132] provides a set of rules based on the sparsity of the polynomial to determine a subset of coefficients containing the enclosure $\min_{\boldsymbol{\alpha}}(b_{\boldsymbol{\alpha}})$ and $\max_{\boldsymbol{\alpha}}(b_{\boldsymbol{\alpha}})$. We also note that the complete set of the Bernstein coefficients of $\mathbf{x}^\gamma$ can be obtain as the successive Kronecker product [156] of the coefficients of each constituting univariate monomial $x_i^{\gamma_i}$. As the set of coefficient as defined in Definition 4.7 can also be represented as a tensor, this Bernstein form is also called tensorial Bernstein expansion over boxes (in contrast with the simplicial Bernstein expansion over simplices [157]).

In practice biochemical reaction networks are often described by sparse polynomials: the dynamic of each species may not depend of all the species of the systems. This sparsity characteristic makes the implicit form very efficient for template reachability analysis of biochemical systems. However, when one cannot take advantage of any sparsity pattern or when all the coefficients have to be computed, it is preferable to use the matrix computation method from [65].

Parametric Bernstein expansion. The previous work of [65] introduces a parametric Bernstein expansion for functions $f(\mathbf{x}, \mathbf{k}) : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$, which are

[4] We recall that the number of Bernstein coefficients is exponential in dimension.

multivariate polynomial in $\mathbf{x}$ and linear in $\mathbf{k}$. In a similar manner to Definition 4.7 we can write a parametric Bernstein expansion of f up to multi-degree $\mathbf{l}$ as:

$$f(\mathbf{x},\mathbf{k}) = \sum_{\gamma} a_{\boldsymbol{\gamma}}(\mathbf{k})\mathbf{x}^{\gamma} = \sum_{\boldsymbol{\alpha}\leq\mathbf{l}} b_{\boldsymbol{\alpha}}^{(f)}(\mathbf{k})\mathbf{B}_{\mathbf{l},\boldsymbol{\alpha}}(\mathbf{x})\,, \tag{4.13}$$

where each $a_{\boldsymbol{\gamma}}(\mathbf{k}) : \mathbb{R}^m \to \mathbb{R}$ is a linear functions in $\mathbf{k}$. Similarly, each Bernstein coefficient is formulated as a linear function in the parameters $\mathbf{k}$:

$$b_{\boldsymbol{\alpha}}(\mathbf{k}) = \sum_{\boldsymbol{\beta}\leq\boldsymbol{\alpha}} \frac{\binom{\boldsymbol{\alpha}}{\boldsymbol{\beta}}}{\binom{\mathbf{l}}{\boldsymbol{\beta}}} a_{\boldsymbol{\beta}}(\mathbf{k}), \quad \mathbf{0} \leq \boldsymbol{\alpha} \leq \mathbf{l}. \tag{4.14}$$

The previous work of [65] demonstrates that the enclosure and sharpness properties hold for the parametric Bernstein expansion:

$$\begin{aligned} \min_{\boldsymbol{\alpha}\leq\mathbf{l}} \min_{\mathbf{k}\in K} b_{\boldsymbol{\alpha}}(\mathbf{k}) &\leq \min_{(\mathbf{x},\mathbf{k})\in[0,1]^n\times K} f(\mathbf{x},\mathbf{k}) \\ \max_{(\mathbf{x},\mathbf{k})\in[0,1]^n\times K} f(\mathbf{x},\mathbf{k}) &\leq \max_{\boldsymbol{\alpha}\leq\mathbf{l}} \max_{\mathbf{k}\in K} b_{\boldsymbol{\alpha}}(\mathbf{k})\,. \end{aligned} \tag{4.15}$$

Moreover, we note that the methods and implementation to compute the Bernstein coefficients in the non-parametric case can still be applied in the parametric case. Indeed, one can write by linearity (Property 4.10):

$$b_{\boldsymbol{\alpha}}^{(f)}(\mathbf{k}) = \sum_{\gamma} a_{\boldsymbol{\gamma}}(\mathbf{k}) b_{\boldsymbol{\alpha}}^{(\mathbf{x}^{\gamma})}\,. \tag{4.16}$$

We also note that the number of *parametric Bernstein coefficients* is still $\prod_{i=1}^{n}(l_i+1)$. Finally, if the set K is a polyhedral set then each optimization $\min_{\mathbf{k}\in K} b_{\boldsymbol{\alpha}}(\mathbf{k})$ can be solved using linear programming.

Example 4.15 (Parametric Bernstein expansion)**.** Let define $f'(x,\mathbf{k}) = (2x^2 - x)k_1 + x^2k_2 + (x^2 - x)k_3$, with $\mathbf{k} \in [-1,1]^3$. Applying the above described methods for the parametric Bernstein expansion with $\mathbf{l} = \mathbf{d} = 2$, we consider the following parametric Bernstein coefficients:

$$b_0^{(f'_{\mathbf{k}})} = 0, \quad b_1^{(f'_{\mathbf{k}})} = -\frac{k_1}{2} - \frac{k_3}{2}, \quad b_2^{(f'_{\mathbf{k}})} = k_1 + kk_2.$$

We note that the number of Bernstein coefficients w.r.t. x is 3, which is much lower than the one w.r.t. $(x,\mathbf{k})$, which is equal to 24. One can obtain an upper bound (resp. lower bound) by taking the maximum (resp. minimum) of the Bernstein coefficients. In this case, $\max_{\mathbf{k}\in[-1,1]^3} b_1^{(f'_{\mathbf{k}})} = 0$, $\max_{\mathbf{k}\in[-1,1]^3} b_2^{(f'_{\mathbf{k}})} = 1$ and $\max_{\mathbf{k}\in[-1,1]^3} b_3^{(f'_{\mathbf{k}})} = 2$. Thus, one obtains $\overline{f'_{\mathbf{l}}} = 2$ as an upper bound of f'.

In this work we also propose an additional method to compute an upper bound of the optimum $\max_{(\mathbf{x},\mathbf{k})\in[0,1]^n\times K} f(\mathbf{x},\mathbf{k})$ in the particular case of parameters varying inside a box $K = [\overline{\mathbf{k}}, \underline{\mathbf{k}}]$. As each $b_{\boldsymbol{\alpha}}^{(f)}(\mathbf{k})$ is linear in $\mathbf{k}$, by monotonicity of linear applications, the maximum and minimum of each $b_{\boldsymbol{\alpha}}^{(f)}$ are obtained at the corners of K. Through a linear transformation to the box $[-1,1]^m$ it is possible to further speed up this computation.

Let $\phi(\mathbf{k}) : \mathbb{R}^m \to \mathbb{R}^m$ be the linear transformation that associates to each $\mathbf{k} \in K$ a vector $\mathbf{e} \in [-1,1]^m$, noting $\mathbf{e} = \phi(\mathbf{k})$. The optimization problem transformed to $\mathbf{e} \in [-1,1]^m$ is now $\max_{(\mathbf{x},\mathbf{e})\in[0,1]^n\times[-1,1]^m} f'(\mathbf{x},\mathbf{e}) = f(\mathbf{x},\phi^{-1}(\mathbf{e}))$. Then, by linearity of $f'(\mathbf{x},\mathbf{e})$ in $\mathbf{e}$ we can write:

$$f'(\mathbf{x},\mathbf{e}) = \sum_{j=1}^{m} \mathbf{e}_j s_j(\mathbf{x}) \,, \tag{4.17}$$

where $s_j(\mathbf{x}) = \frac{\partial f'(\mathbf{x},\mathbf{e})}{\partial e_j}$ is a polynomial in $\mathbf{x}$.

Finally, for each $\mathbf{l} \geq \mathbf{d}$[5], let us note $\overline{f'_{\mathbf{l}}} := \max_{\boldsymbol{\alpha}\leq\mathbf{k}} \sum_{j=1}^{m} |b_{\boldsymbol{\alpha}}^{(s_j)}|$ and $\underline{f'_{\mathbf{l}}} := -\overline{f'_{\mathbf{l}}}$. Our procedure is based on the following lemma:

Lemma 4.16. *For each* $\mathbf{l} \geq \mathbf{d}$*, the polynomial* $f'(\mathbf{x},\mathbf{e})$ *can be bounded as follows:*

$$\underline{f'_{\mathbf{l}}} \leq f'(\mathbf{x},\mathbf{e}) \leq \overline{f'_{\mathbf{l}}} \,, \quad \forall(\mathbf{x},\mathbf{e}) \in [0,1]^n \times [-1,1]^m \,. \tag{4.18}$$

Proof. We write $f'_{\mathbf{e}} \in \mathbb{R}[\mathbf{x}]$ the polynomial $f'(\mathbf{x},\mathbf{e})$ for a given $\mathbf{e} \in [-1,1]^m$. Property 4.11 provides the enclosure of $f'_{\mathbf{e}}(\mathbf{x})$ w.r.t. $\mathbf{x}$ for a given $\mathbf{e} \in [-1,1]^m$:

$$\min_{\boldsymbol{\alpha}\leq\mathbf{k}} b_{\boldsymbol{\alpha}}^{(f'_{\mathbf{e}})} \leq f'_{\mathbf{e}}(\mathbf{x}) \leq \max_{\boldsymbol{\alpha}\leq\mathbf{k}} b_{\boldsymbol{\alpha}}^{(f'_{\mathbf{e}})} \,, \quad \forall\mathbf{x} \in [0,1]^n \,, \tag{4.19}$$

where each Bernstein coefficient satisfies $b_{\boldsymbol{\alpha}}^{(f'_{\mathbf{e}})} = \sum_{j=1}^{m} e_j b_{\boldsymbol{\alpha}}^{(s_j)}$ by Property 4.10 (each e_j being a scalar in $[-1,1]$). The proof of the left inequality comes from:

$$\begin{aligned}\min_{\mathbf{e}\in[-1,1]^m} \big(\min_{\boldsymbol{\alpha}\leq\mathbf{k}}(\sum_{j=1}^{m} e_j b_{\boldsymbol{\alpha}}^{(s_j)})\big) &= \min_{\boldsymbol{\alpha}\leq\mathbf{k}}\big(\min_{\mathbf{e}\in[-1,1]^m}(\sum_{j=1}^{m} e_j b_{\boldsymbol{\alpha}}^{(s_j)})\big) \\ &= \min_{\boldsymbol{\alpha}\leq\mathbf{k}} \sum_{j=1}^{m} -|b_{\boldsymbol{\alpha}}^{(s_j)}| = -\max_{\boldsymbol{\alpha}\leq\mathbf{k}} \sum_{j=1}^{m} |b_{\boldsymbol{\alpha}}^{(s_j)}| \,.\end{aligned}$$

The proof of the right inequality is similar. □

Remark 4.17. The computational cost of $\underline{f'_{\mathbf{l}}}$ is now $m(\mathbf{l}+\mathbf{1})^{\mathbf{1}}$ since we need to compute the Bernstein coefficients for each $s_j(\mathbf{x})$. This cost is polynomial in the degree and exponential in n but is linear in m the number of parameters. In the implementation, we first compute each $b_{\boldsymbol{\alpha}}^{(f'_{\mathbf{e}})}$ as a function of $\mathbf{e}$ and then optimize

[5] We recall that $\mathbf{d}$ is the multi-degree of f' (and f) in $\mathbf{x}$.

afterwards using either optimization as in [65] or Lemma 4.16 depending of the situation. As described further in Section 4.2.2, we show how [1] gets around the limitation to boxes in Property 4.11 and handle $\mathbf{x}$ in polyhedral domains through the definition of bundles of parallelotopes. A similar approach can be taken if one wants to extend Lemma 4.16 to polyhedral sets.

Parametric Bernstein expansion of rational functions. To handle a larger class of biological models we introduce an additional method, based on the original results of [158], to optimize rational function using the Bernstein expansion. Let us now assume that $f(\mathbf{x},\mathbf{k}) : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$ is a rational function in $\mathbf{x}$ and linear in $\mathbf{k}$:

$$f(\mathbf{x},\mathbf{k}) = \frac{f_1(\mathbf{x},\mathbf{k})}{f_2(\mathbf{x})}\,,$$

where $f_1 \in \mathbb{R}[\mathbf{x}]$ linear in $\mathbf{k}$ and $f_2 \in \mathbb{R}[\mathbf{x}]$ with $f_2(\mathbf{x}) \neq 0\,,\ \forall \mathbf{x} \in [0,1]^n$. Then, from the previous results on the parametric Bernstein expansion (4.13),(4.16) and by linearity in $\mathbf{k}$ we can extend the theorem from [158] to the parametric case.

Theorem 4.18. *Let $f_1(\mathbf{x},\mathbf{k})$ be a polynomial in $\mathbf{x}$ of multi-degree $\mathbf{d}^{(f_1)}$ and linear in $\mathbf{k}$ and $f_2(\mathbf{x})$ be a polynomial in $\mathbf{x}$ of multi-degree $\mathbf{d}^{(f_2)}$ such that $f_2(\mathbf{x}) \neq 0\,,\ \forall \mathbf{x} \in [0,1]^n$. Given $\mathbf{l} \geq \max(\mathbf{d}^{(f_1)}, \mathbf{d}^{(f_2)})$, we note $\{b^{f_1}_{\boldsymbol{\alpha}}(\mathbf{k})\}$ and $\{b^{f_2}_{\boldsymbol{\alpha}}\}$ the Bernstein coefficients up to degree $\mathbf{l}$ of f_1 and f_2 respectively.*

Then $f(\mathbf{x},\mathbf{k}) = f_1(\mathbf{x},\mathbf{k})/f_2(\mathbf{x})$ is bounded for $(\mathbf{x},\mathbf{k}) \in [0,1]^n \times K$ by:

$$\min_{\boldsymbol{\alpha}\leq \mathbf{l}} \min_{\mathbf{k}\in K} \frac{b^{f_1}_{\boldsymbol{\alpha}}(\mathbf{k})}{b^{f_2}_{\boldsymbol{\alpha}}} \leq f(\mathbf{x},\mathbf{k}) \leq \max_{\boldsymbol{\alpha}\leq \mathbf{l}} \max_{\mathbf{k}\in K} \frac{b^{f_1}_{\boldsymbol{\alpha}}(\mathbf{k})}{b^{f_2}_{\boldsymbol{\alpha}}} \tag{4.20}$$

Proof. For a given fixed $\mathbf{k} \in K$ we have thank to [158]:

$$f(\mathbf{x},\cdot|\mathbf{k}) \leq \max_{\boldsymbol{\alpha}\leq \mathbf{l}} \frac{b^{f_1}_{\boldsymbol{\alpha}}(\mathbf{k})}{b^{f_2}_{\boldsymbol{\alpha}}} = \max_{\boldsymbol{\alpha}\leq \mathbf{l}} \sum_{\boldsymbol{\gamma}\leq \mathbf{d}^{(f_1)}} a_{\boldsymbol{\gamma}}(\mathbf{k}) \frac{b^{\mathbf{x}^{\boldsymbol{\gamma}}}_{\boldsymbol{\alpha}}}{b^{f_2}_{\boldsymbol{\alpha}}}$$

Thus, it yields for all $\mathbf{k} \in K$ by linearity in $a_{\boldsymbol{\gamma}}(\mathbf{k})$

$$\begin{aligned} f(\mathbf{x},\mathbf{k}) &\leq \max_{\mathbf{k}\in K} \max_{\boldsymbol{\alpha}\leq \mathbf{l}} \sum_{\boldsymbol{\gamma}\leq \mathbf{d}^{(f_1)}} a_{\boldsymbol{\gamma}}(\mathbf{k}) \frac{b^{\mathbf{x}^{\boldsymbol{\gamma}}}_{\boldsymbol{\alpha}}}{b^{f_2}_{\boldsymbol{\alpha}}} \\ &\leq \max_{\boldsymbol{\alpha}\leq \mathbf{l}} \max_{\mathbf{k}\in K} \sum_{\boldsymbol{\gamma}\leq \mathbf{d}^{(f_1)}} a_{\boldsymbol{\gamma}}(\mathbf{k}) \frac{b^{\mathbf{x}^{\boldsymbol{\gamma}}}_{\boldsymbol{\alpha}}}{b^{f_2}_{\boldsymbol{\alpha}}} = \max_{\boldsymbol{\alpha}\leq \mathbf{l}} \max_{\mathbf{k}\in K} \frac{b^{f_1}_{\boldsymbol{\alpha}}(\mathbf{k})}{b^{f_2}_{\boldsymbol{\alpha}}} \end{aligned}$$

The proof is similar for the other inequality. □

The main advantage is that this enclosure is more accurate than the naive enclosure:

$$\frac{\min_{\mathbf{x},\mathbf{k}} f_1(\mathbf{x},\mathbf{k})}{\max_{\mathbf{x}} f_2(\mathbf{x})} \leq f(\mathbf{x},\mathbf{k}) \leq \frac{\max_{\mathbf{x},\mathbf{k}} f_1(\mathbf{x},\mathbf{k})}{\min_{\mathbf{x}} f_2(\mathbf{x})}\,.$$

Indeed, in the formulation of Theorem 4.18 we avoid decorrelating the optimization on the numerator and denominator. However, this property requires generating the complete Bernstein expansion of both functions f_1 and f_2 up to the closest common multi-degree $\mathbf{l} \geq \max(\mathbf{d}^{(f_1)}, \mathbf{d}^{(f_2)})$. We refer the reader to [159] for the convergence rates in degree elevation and subdivision for the Bernstein expansion of rational functions. We will call in the following a rational vector field, a vector field constituted of polynomial functions and at least one rational function.

4.2.2 Validation of piecewise polynomial ODE systems using the Bernstein expansion

In the previous sections, we defined the necessary background on discrete time reachability analysis using polyhedral templates. We also introduced the parametric Bernstein expansion as a method to solve parametric polynomial optimization problems. We exhibited our contribution for an efficient representation of discrete time system approximation maps. Finally, we showed how the work of [65] can be extended to the optimization of parametric rational function, and we proposed an alternate method for parametric polynomial optimization when parameters lie in a box.

In this section, we introduce the algorithm for discrete time reachability analysis by extending the previous implementation of [83]. To this aim, we propose in Algorithm 2 an algorithm to compute the intersection of parallelotope bundles with linear constraints. Then, we extend in Algorithm 3 the work of [83] to discrete time reachability analysis of piecewise rational functions.

We recall that we are considering dynamics defined[6] by a polynomial (or rational) vector field $\mathbf{f}$ in $\mathbf{x}$ and linear in the parameters $\mathbf{k}$. We remind from Section 4.1.2 (4.8) that given a template matrix the template-based reachability algorithm can be summarized to the following optimization problem at each iteration τ:

$$\mathbf{c}_{\tau+1} = \max_{(\mathbf{x},\mathbf{k})\in X_\tau \times K} \Gamma \mathbf{f}(\mathbf{x},\mathbf{k}) = \begin{pmatrix} \max\limits_{(\mathbf{x},\mathbf{k})\in X_\tau \times K} \Gamma_1 \mathbf{f}(\mathbf{x},\mathbf{k}) \\ \dots \\ \max\limits_{(\mathbf{x},\mathbf{k})\in X_\tau \times K} \Gamma_p \mathbf{f}(\mathbf{x},\mathbf{k}) \end{pmatrix} \tag{4.21}$$

In the previous section we showed how the Bernstein expansion can be used to efficiently solve this optimization problem for $\mathbf{x} \in [0,1]^n$. We note such optimization method can be extended for $\mathbf{x}$ in any linear transformation of the unit box $[0,1]^n$. The previous implementation [83] focused on the parallelotope: a set representation that can be expressed as a linear transformation of $[0,1]^n$.

Definition 4.19 (Parallelotope generator representation)**.** Let $\mathbf{q} \in \mathbb{R}^n$ be a point, and $\{\boldsymbol{\sigma}_, \dots, \boldsymbol{\sigma}_n\}$ a set of n linearly independent vectors in $\mathbb{R}^n$. We associate to $\mathbf{q}$

[6] Locally defined by rational or polynomial vector field in each domain, for the particular case of piecewise approximated dynamics

and $\{\boldsymbol{\sigma}_j\}$ the parallelotope Π such that for all $\mathbf{x} \in \Pi$ there exists $\mathbf{y} \in [0,1]^n$ with:

$$\mathbf{x} = \pi(\mathbf{y}) = \mathbf{q} + \sum_{j=1}^{n} y_j * \boldsymbol{\sigma}_j \tag{4.22}$$

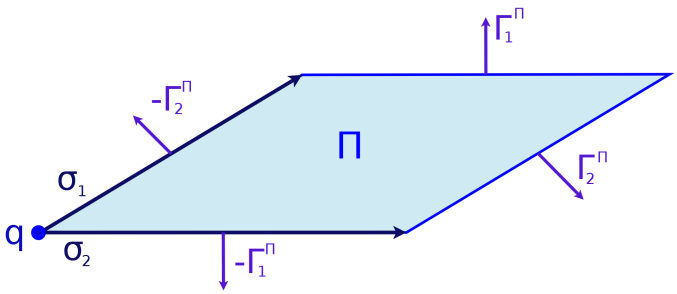

Figure 4.6: We represent a 2 dimensional parallelotope Π defined by its generator form: a linear combination of the base vertex q, and the generators $\boldsymbol{\sigma}_1$, $\boldsymbol{\sigma}_2$. We also represent its associated constraint from Γ^Π defined by the facets normal $\Gamma_1^\Pi, \Gamma_2^\Pi, -\Gamma_1^\Pi, -\Gamma_2^\Pi$

A parallelotope Π can also be described by a set of linear constraints that we denote by the linear system $\Gamma^\Pi \mathbf{x} \leq \mathbf{c}^\Pi$ (see Figure 4.6). The representation change from the generators to the constraints representation, or inverse, is defined in the original work on parallelotopes [65]. Using the linear transformation from (4.22), then in the particular case where X_τ is a parallelotope each optimization $\max_{X_\tau \times K} \Gamma_i \mathbf{f}(\mathbf{x}, \mathbf{k})$ from (4.21) becomes:

$$\begin{aligned} c_{\tau+1,i} &= \max_{(\mathbf{x},\mathbf{k}) \in X_\tau \times K} \Gamma_i \mathbf{f}(\mathbf{x}, \mathbf{k}) \\ &= \max_{(\mathbf{y},\mathbf{k}) \in [0,1]^n \times K} \Gamma_i \mathbf{f}(\pi_\tau(\mathbf{y}), \mathbf{k}) \\ &= \max_{(\mathbf{y},\mathbf{k}) \in [0,1]^n \times K} \eta_i(\mathbf{y}, \mathbf{k}) \,, \end{aligned} \tag{4.23}$$

where π_τ is the linear transformation from the unit box to the parallelotope X_τ, and $\eta_i(\mathbf{y}, \mathbf{k}) = \Gamma_i \mathbf{f}(\pi_\tau(\mathbf{y}), \mathbf{k})$ at iteration τ. This optimization can now be addressed using the Bernstein expansion. However, after this single step we may still obtain a polyhedral set $\hat{X}_{\tau+1}$, which is not necessarily a parallelotope[7]. The work of [1] develops further [65] and introduces the parallelotope bundle to approximate polyhedral compact sets. In [1] it is proved that a polyhedron $\mathcal{P}$ can be exactly described by the intersection of at least $n_\mathcal{P} = \texttt{round} \uparrow (p/n)$ parallelotopes, where $\texttt{round} \uparrow (\cdot)$ is the rounding to the superior natural integer, n is the dimension and p the number of non-symmetric constraints (two constraints are symmetric if their normal vectors are opposed) defining in the constraint matrix of $\mathcal{P}$.

$$\mathcal{P} = \Pi_1 \cap \Pi_2 \cap \cdots \cap \Pi_{n_\mathcal{P}} \tag{4.24}$$

[7]The polyhedral template can be chosen to be a parallelotope however we may lose in accuracy.

We note from [1] that the constraint representation of each parallelotope Π_j, $\imath \in [1, n_{\mathcal{P}}]$ is constituted from n constraints of $\mathcal{P}$ and their symmetric.

Example 4.20 (Bundle of parallelotopes)**.** Let $\mathcal{P}$ be a polyhedron defined by a template matrix Γ of 5 constraints: $\Gamma_1, \Gamma_2, \Gamma_3, -\Gamma_1, -\Gamma_2$: we note that $p = 3$ as only 3 constraints are non-symmetric. We represent this polyhedron in the Figure 4.7. We can defined $\texttt{round} \uparrow (3/2) = 2$ parallelotopes Π_1 and Π_2 such that $\mathcal{P} = \Pi_1 \cap \Pi_2$. They are represented in Figure 4.7.
The parallelotope Π_1 is defined by the constraint system $\Gamma^{\Pi_1} = \{\Gamma_1, \Gamma_3, -\Gamma_1, -\Gamma_3\}$, while the parallelotope Π_2 is defined by the constraint system $\Gamma^{\Pi_1} = \{\Gamma_1, \Gamma_2, -\Gamma_1, -\Gamma_2\}$. We note that the constraint Γ_1 is redundant in Π_1 and Π_2: this redundancy will be used later in Example 4.21.

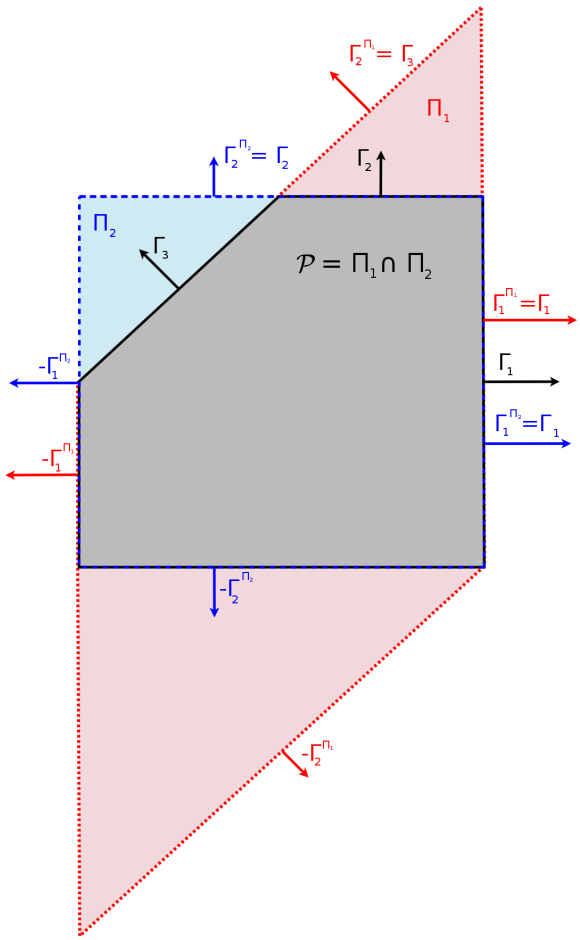

Figure 4.7: In this figure we show a possible bundle of parallelotopes $\{\Pi_1, \Pi_2\}$ which represents the polyhedron $\mathcal{P}$ as defined in Example 4.20

It is also proved that the image of $\mathcal{P}$ by a non-linear transformation is included in the intersection of the image of all its associated parallelotopes. Let $\mathbf{f}$ be a non-linear vector field:

$$\mathbf{f}(\mathcal{P}) \subseteq \mathbf{f}(\Pi_1) \cap \mathbf{f}(\Pi_2) \cap \cdots \cap \mathbf{f}(\Pi_{n_{\mathcal{P}}}) \tag{4.25}$$

If at iteration τ the set X_τ is a polyhedron that is described by the intersection of a set of parallelotopes $\{\Pi_j\}_{j\in[1,n_{X_\tau}]}$, then we need to solve an optimization problem similar to (4.23) for each parallelotope and keep the tightest result:

$$\begin{aligned} c_{\tau+1,i} &= \min_{j\in[1,n_{X_\tau}]} \max_{(\mathbf{y},\mathbf{k})\in[0,1]^n\times K} \Gamma_i \mathbf{f}(\pi_{j,\tau}(\mathbf{y}),\mathbf{k}) \\ &= \min_{j\in[1,n_{X_\tau}]} \max_{(\mathbf{y},\mathbf{k})\in[0,1]^n\times K} \eta_{i,j}(\mathbf{y},\mathbf{k}) \end{aligned} \tag{4.26}$$

where $\eta_{i,j}(\mathbf{y},\mathbf{k}) = \Gamma_i \mathbf{f}(\pi_{j,\tau}(\mathbf{y}),\mathbf{k})$ at iteration τ. This procedure is called *all for one* (AFO) in [1] as the position of each constraint is the tightest among all the ones resulting from the optimization over each parallelotope. Using the formulation of (4.26) we can now use the Bernstein expansion to compute the over-approximation using polyhedral template of the image of a polyhedron by a polynomial (or rational) vector field.

In this work we want to perform reachability analysis of piecewise continuous vector fields. To this aim, we defined in Section 4.1.3 approximation maps and their associated representation as BSPT. To locate the position of a polyhedron in the tree, and its intersection with the different nodes we use the intersection operator between a polyhedron X and a linear constraint h. The work of [1] does not introduce an equivalent operator for parallelotope bundles.

The previous work of [1] presented a procedure, called `decompose`, to produce a bundle of parallelotopes representing a template polyhedron X. One can compute the intersection of a bundle associated to the polyhedron X with a linear constraint h by applying the procedure `decompose` on the resulting intersection $X \cap h$. In [1], the authors note that finding the most accurate parallelotope decomposition representing a polyhedron is NP-hard. To this aim, they include into the procedure `decompose` an heuristic to obtain an accurate decomposition.

In this work we propose a different approach which focuses less on accuracy and more on performances. A method to compute this intersection without having to recompute the bundle representation from scratch using the `decompose` procedure [1] is defined in the Algorithm 2. This algorithm is divided in two steps: the first is to add the intersecting constraint to the bundle definition, and the second is to tighten the redundant constraints in consequence. Indeed, as we work with a fixed template, some constraints of the template may become redundant at some point of the analysis. We note that if the intersecting constraint does not appear in the template, and there is no constraint in common between two parallelotopes of the bundle, unlike in Figure 4.8 for example, then we need to add a new parallelotope: this occurs when $n_{\mathcal{P}} = p/n$.

In view of application to systems biology, we already noted that the polynomials modelling the dynamics are often sparse. This implies that the optimization of the constraint parallel to the axis (defining a box) are in general cheaper to compute. A good heuristic is to always add such axis parallel constraints in the template, and keep a complete box in the bundle (see Figure 4.8). This box can then be duplicated when one needs to add new parallelotope to the bundle.

Example 4.21 (Bundle intersection)**.** We consider the bundle defined in the previous Example 4.20. We now compute its intersection with a constraint h as shown in Figure 4.8. As seen in Example 4.20, $p/n = 1.5 \leq 2$ and there is a redundant constraint: $\Gamma_1^{\Pi_1} = \Gamma_1^{\Pi_2} = \Gamma_1$. Following Algorithm 2, we need to update the parallelotope Π_1 into Π_1' by changing the constraint $\Gamma_1^{\Pi_1}$ into $\Gamma_1^{\Pi_1'} = h$. Consequently, the parallelotope Π_1' is defined by the constraint system $h, \Gamma_3, -h, -\Gamma_3$.
Finally, to obtain a representing bundle we tighten, using linear optimization, the remaining constraints $-\Gamma_2^{\Pi_1}$ and $-\Gamma_2^{\Pi_2}$ such that they are tangent to the set $\mathcal{P} \cap h$.

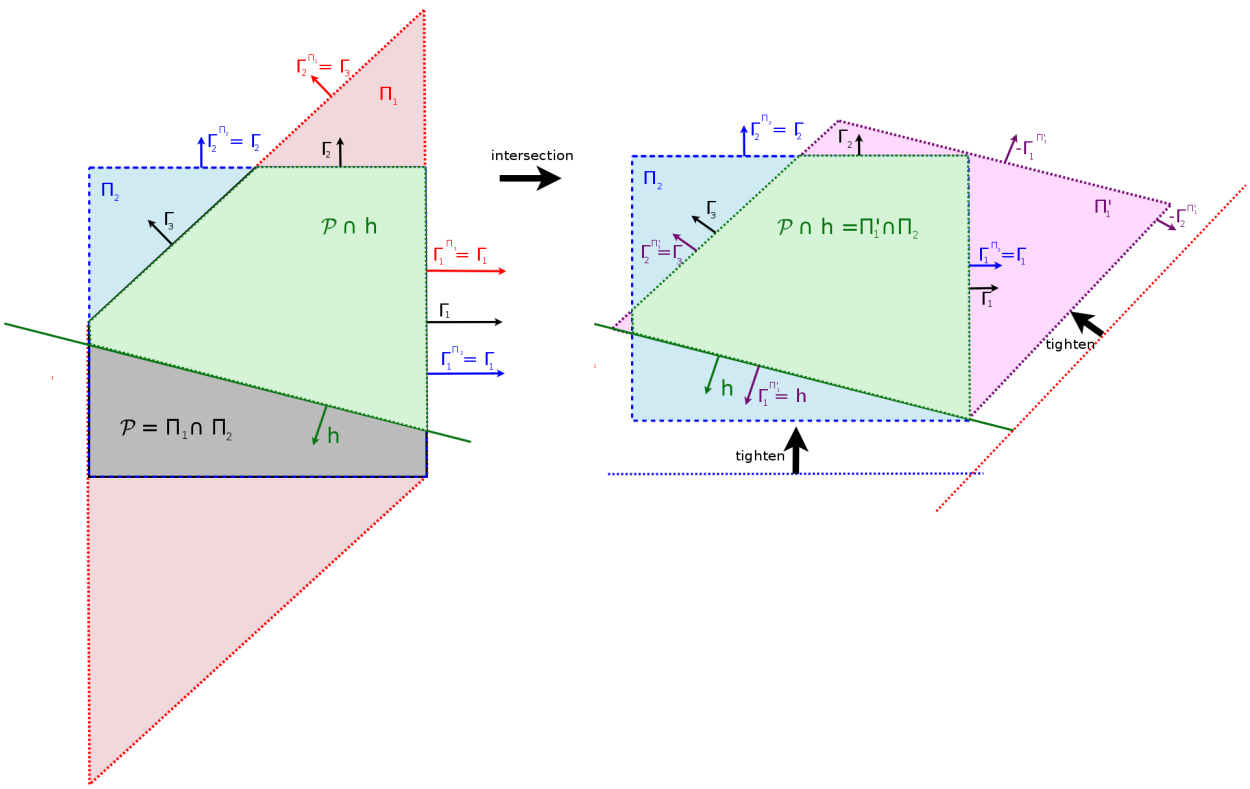

Figure 4.8: On the left we show a parallelotope bundle of $\mathcal{P}$ and the intersecting constraint h. On the right we show a bundle generated by Algorithm 2 representing the intersection $\mathcal{P} \cap h$.

Algorithm 2 Intersection(X,$\mathcal{B}$,h)

INPUT: $X = (\Gamma, \mathbf{c})$, current set (template polyhedron) .
INPUT: $\mathcal{B} = \{\Pi_i\}_{i\in[1,n_\mathcal{P}]}$, associated bundle constituted of $n_\mathcal{P}$ parallelotopes.
We note $\Gamma^{(\Pi_i)}$, $\mathbf{c}^{(\Pi_i)}$ their associated constraint representations.
INPUT: h, intersecting constraint of the form $\mathbf{a}\mathbf{x} \leq c_h$

*/*If* $\mathbf{a}$ *is already in the template matrix* Γ**/*
if $\mathbf{a} \in \Gamma$ **then**
    $l = \texttt{position_in}(\mathbf{a}, \Gamma)$
    $\mathbf{c}[l] = c_h$
    Update accordingly $\mathbf{c}^{(\Pi_i)}$ for$\Pi_i \in \mathcal{B}$ associated to $\Gamma[l]$
else
    if $p/n == \texttt{round} \uparrow (p/n)$ **then**
        / The new constraint* h *is represented by a new parallelotope */*
        $\Pi_{new} = \Pi_1$ */* Create a duplicate */*
        $\Gamma^{(\Pi_{new})}[1] = \mathbf{a}$, $\mathbf{c}^{(\Pi_{new})}[1] = c_h$
        / Add* Π_{new} *to* $\mathcal{B}$ *and add a new line to* Γ *and* $\mathbf{c}$ **/*
        $\Gamma[\texttt{last}] = \mathbf{a}$, $\mathbf{c}[\texttt{last}] = c_h$
    else
        $\exists i \neq i' \in [1, n_\mathcal{P}]$ s.t. $l = \texttt{common_cstr}(\Pi_i, \Pi_{i'})$
        / A constraint appears multiple times: we replace it by* h **/*
        $\Gamma^{(\Pi_i)}[l] = \mathbf{a}$, $\mathbf{c}^{(\Pi_i)}[l] = c_h$
    end if
end if
for $\Gamma[l] \in \Gamma$ **do**
    / Tighten the constraints of the polyhedron */*
    $\mathbf{c}[l] = \max_{\mathbf{x}\in(\Gamma,\mathbf{c})} \Gamma[l]\mathbf{x}$
    Update each parallelotope constraint representation accordingly.
    Update each parallelotope generator representation.
end for
return X, $\mathcal{B}$

Finally, we describe in Algorithm 3 the complete algorithm for one step of polyhedral template reachability analysis of piecewise dynamics such as defined in Definition 4.4 using Bernstein expansion. The complete reachability analysis is done by iterating over Algorithm 3.

4.3 The Krivine-Stengle representation based method

In this section we provide a method for reachability analysis of parametric polynomial system using sparse Krivine-Stengle (K.S.) representations of positive polynomials. This method is an alternative to the Bernstein expansion based method as it is also designed to solve parametric polynomial optimization problems. However, unlike Bernstein expansion, which is limited to variables constrained in a box, K.S. can be used for polynomial optimization with semi-algebraic constraints. In practice we propose an algorithm for polyhedral template reachability analysis of

Algorithm 3 Bernstein_One_Step_Reach(X,Ψ,Γ,K)

INPUT: X: Current set (template polyhedron)
INPUT: Ψ: BSPT and its associated piecewise approximation of the dynamics.
INPUT: Γ: template which is used to over-approximate the reachable set.
INPUT: K: parameters set
OUTPUT: an over-approximation of the reachable set of X after one step.

$\mathcal{G}$: set of current nodes, or the tree root $\top$

($\mathcal{L}$,$\mathcal{X}$) = Ψ.`locating`$(X, \mathcal{G})$ /* *finding intersecting nodes, and associated intersections* */

for $\Gamma_i \in \Gamma$ **do**
  /* *computing offset bound c_i for each constraint Γ_i in the template* */

  for $(\mathcal{L}_\kappa, X_\kappa) \in \mathcal{L}$ **do**
    Compute the approximate dynamics f_κ associated with the node $\mathcal{L}_\kappa$
    for $\Pi_j \in$ `Bundle`(X_κ) **do**
      /* *Get parallelotopic bundle associated to polyhedron X_κ* */

      Construct the polynomial $\eta_{i,\kappa,j}$ from polynomial f_κ of the dynamics,
      template constraint Γ_i and paralletopic domain Π_j, as defined in (**??**)

      $c_{i,j,\kappa} = \max\{\eta_{i,j,\kappa}(\mathbf{y}, \mathbf{k})) \mid \mathbf{y} \in [0,1]^n \wedge \mathbf{k} \in K\}$
      /* *using the Bernstein expansion for polynomial $\eta_{i,\kappa,j}$* */

    end for

    $c_{i,\kappa} = \min_j(c_{i,\kappa,j})$
    /* *smallest bound over all parallelotopes Π_j in the bundle* */

  end for

  $c_i = \max_\kappa(c_{i,\kappa})$
  /* *largest bound by all approximate dynamics of intersecting nodes* */
  /* *We keep one set to avoid cost explosion due to successive intersections* */

end for
return $(\Gamma, \mathbf{c})$ /* *the result is the template polyhedron with offsets* **c** */

discrete time polynomial dynamics with parameters constrained in a box.

We first give, in Section 4.3.1, the necessary background on Krivine-Stengle representations, used in the context of polynomial optimization. Then, we present a sparse version based on [160]. These notions are later applied in Section 4.3.2 to parametric polynomial optimization and template discrete time reachability analysis.

4.3.1 Polynomial Optimization using K.S representations

Dense Krivine-Stengle representations. Krivine-Stengle certificates for positive polynomials can first be found in [161, 162] (see also [163, Theorem 1(b)]). Such certificates give representations of positive polynomials over a compact set $\mathcal{X} = \{\mathbf{x} \in \mathbb{R}^n : 0 \leq g_i(\mathbf{x}) \leq 1,\ i = 1, \ldots, p\}$, with $g_1, \ldots, g_p \in \mathbb{R}[\mathbf{x}]$. We note $d^{\mathbf{g}} = \max_i(\deg(g_i))$. The compact set $\mathcal{X}$ is a basic semi-algebraic set, that is a set defined by a conjunction of finitely many polynomial inequalities. In the sequel, we assume without loss of generality that $\mathcal{X} \subseteq [0,1]^n$ and that $\mathcal{X}$ involves the

polynomials x_i for all $i = 1, \ldots, n$ in its definition. This implies that the family $\{1, g_i\}_{i \leq p}$ generates $\mathbb{R}[\mathbf{x}]$ as an $\mathbb{R}$-algebra, which is a mandatory assumption for Theorem 4.23. Given $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_p)$ and $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_p)$, let us define the polynomial $h_{\boldsymbol{\alpha},\boldsymbol{\beta}}(\mathbf{x}) = \mathbf{g}^{\boldsymbol{\alpha}}(\mathbf{1} - \mathbf{g})^{\boldsymbol{\beta}} = \prod_{i=1}^{p} g_i^{\alpha_i}(1 - g_i)^{\beta_i}$.

Example 4.22 (K.S. representations)**.** For instance on the two-dimensional unit box, one has $n = p = 2$, $\mathcal{X} = [0,1]^2 = \{\mathbf{x} \in \mathbb{R}^2 : 0 \leq x_1 \leq 1\,,\ 0 \leq x_2 \leq 1\}$. For $\boldsymbol{\alpha} = (2,1)$ and $\boldsymbol{\beta} = (1,3)$, one has $h_{\boldsymbol{\alpha},\boldsymbol{\beta}}(\mathbf{x}) = x_1^2 x_2 (1 - x_1)(1 - x_2)^3$.

Theorem 4.23 (Dense Krivine-Stengle representations)**.** *Let $\psi \in \mathbb{R}[\mathbf{x}]$ be a positive polynomial over $\mathcal{X}$. Then there exist $l \in \mathbb{N}$ and a finite number of nonnegative weights $\lambda_{\boldsymbol{\alpha},\boldsymbol{\beta}} \geq 0$ such that:*

$$\psi(\mathbf{x}) = \sum_{|\boldsymbol{\alpha}+\boldsymbol{\beta}| \leq l} \lambda_{\boldsymbol{\alpha},\boldsymbol{\beta}} h_{\boldsymbol{\alpha},\boldsymbol{\beta}}(\mathbf{x}), \quad \forall \mathbf{x} \in \mathbb{R}^n. \tag{4.27}$$

It is possible to compute the weights $\lambda_{\boldsymbol{\alpha},\boldsymbol{\beta}}$ by identifying in the monomial basis the coefficients of the polynomials in the left and right sides of (4.27). Denoting by $(\psi)_{\boldsymbol{\gamma}}$ the monomial coefficients of ψ, with $\boldsymbol{\gamma} \in \mathbb{N}^n_{l'} := \{\boldsymbol{\gamma} \in \mathbb{N}^n : |\boldsymbol{\gamma}| \leq l' = kd^{\mathbf{g}}\}$, the $\lambda_{\boldsymbol{\alpha},\boldsymbol{\beta}}$ fulfill the following equalities:

$$\psi_{\boldsymbol{\gamma}} = \sum_{|\boldsymbol{\alpha}+\boldsymbol{\beta}| \leq l} \lambda_{\boldsymbol{\alpha},\boldsymbol{\beta}} (h_{\boldsymbol{\alpha},\boldsymbol{\beta}})_{\boldsymbol{\gamma}}, \quad \forall \boldsymbol{\gamma} \in \mathbb{N}^n_{l'}. \tag{4.28}$$

Global optimization using the dense Krivine-Stengle representations. Here we consider the polynomial maximization problem $\overline{f}^* := \max_{\mathbf{x} \in \mathcal{X}} f(\mathbf{x})$, with f a polynomial of degree d. We can rewrite this problem as the following infinite dimensional problem[8]:

$$\begin{aligned} \overline{f}^* := \min_{t \in \mathbb{R}} \quad & t, \\ \text{s.t.} \quad & t - f(\mathbf{x}) \geq 0\,, \quad \forall \mathbf{x} \in \mathcal{X}. \end{aligned} \tag{4.29}$$

The idea is to look for a hierarchy of finite dimensional linear programming (LP) relaxations by using Krivine-Stengle representations of the positive polynomial $\psi = t - f$ involved in Problem (4.29). Applying Theorem 4.23 to this polynomial, we obtain the following LP problem for each $l \geq d$:

$$\begin{aligned} p_l^* := \min_{t, \lambda_{\boldsymbol{\alpha},\boldsymbol{\beta}}} \quad & t, \\ \text{s.t} \quad & (t - f)_{\boldsymbol{\gamma}} = \sum_{|\boldsymbol{\alpha}+\boldsymbol{\beta}| \leq l} \lambda_{\boldsymbol{\alpha},\boldsymbol{\beta}} (h_{\boldsymbol{\alpha},\boldsymbol{\beta}})_{\boldsymbol{\gamma}}\,, \quad \forall \boldsymbol{\gamma} \in \mathbb{N}^n_{l'}\,, \\ & \lambda_{\boldsymbol{\alpha},\boldsymbol{\beta}} \geq 0. \end{aligned} \tag{4.30}$$

As in [163, (4)], one has the following convergence theorem:

[8] the minimization problem resolution is analogous.

Theorem 4.24 (Dense Krivine-Stengle LP relaxations)**.** *The sequence of optimal values* (p_l^*) *satisfies* $p_l^* \to \overline{f}^*$ *as* $l \to +\infty$*. Moreover each* p_l^* *is a upper bound of* $\overline{f}^*$*.*

At fixed l, the total number of variables of Problem (4.30) is given by the number of $\lambda_{\boldsymbol{\alpha},\boldsymbol{\beta}}$ and t, that is $\binom{2p+l}{l} + 1$, where p is the dimension of $\mathbf{g}$. The number of constraints is equal to the cardinality of $\mathbb{N}^n_{l'}$, which is $\binom{n+l'}{l'}$. We recall that $l' = l\, d^{\mathbf{g}}$. In the particular case where $\mathcal{X}$ is an hypercube, the LP has $\binom{2n+l}{l}+1$ variables and $\binom{n+l}{l}$ constraints.

Sparse Krivine-Stengle representations. We now explain how to derive less computationally expensive LP relaxations, by relying on sparse Krivine-Stengle representations. For $I \subseteq \{1,\ldots,n\}$, let $\mathbb{R}[\mathbf{x}, I]$ be the ring of polynomials restricted to the variables $\{x_i \,:\, i \in I\}$. We borrow the notion of a sparsity pattern from [164, Assumption 1]:

Definition 4.25 (Sparsity Pattern)**.** Given $m \in \mathbb{N}$, $I_j \subseteq \{1,\ldots,n\}$, and $J_j \subseteq \{1,\ldots,p\}$ for all $j = 1,\ldots,m$, a sparsity pattern is defined by the four following conditions:

- f can be written as: $f = \sum_{j=1}^m f_j$ with $f_j \in \mathbb{R}[\mathbf{x}, I_j]$,
- $g_i \in \mathbb{R}[\mathbf{x}, I_j]$ for all $i \in J_j$, for all $j = 1,\ldots,m$,
- $\bigcup_{j=1}^m I_j = \{1,\ldots,n\}$ and $\bigcup_{j=1}^m J_j = \{1,\ldots,p\}$,
- (Running Intersection Property) for all $j = 1,\ldots,m-1$, there exists $s \leq j$ s.t. $I_{j+1} \cap \bigcup_{i=1}^j I_i \subseteq I_s$.

Example 4.26 (Sparsity Pattern)**.** As an example, the four conditions stated in Definition 4.25 are satisfied while considering $f(\mathbf{x}) = x_1x_2 + x_1^2x_3$ on the hypercube $\mathcal{X} = [0,1]^3$. Indeed, one has $f_1(\mathbf{x}) = x_1x_2 \in \mathbb{R}[\mathbf{x}, I_1]$, $f_2(\mathbf{x}) = x_1^2x_3 \in \mathbb{R}[\mathbf{x}, I_2]$ with $I_1 = \{1,2\}$, $I_2 = \{1,3\}$. Taking $J_1 = I_1$ and $J_2 = I_2$, one has $g_i = x_i \in \mathbb{R}[\mathbf{x}, I_j]$ for all $i \in I_j$, $j = 1,2$.

Let us consider a given sparsity pattern as stated above. By noting $n_j = |I_j|$, $p_j = |J_j|$, then the set $\mathcal{X} = \{\mathbf{x} \in \mathbb{R}^n : 0 \leq g_i(\mathbf{x}) \leq 1,\ i = 1,\ldots,p\}$ yields subsets $\mathcal{X}_j = \{\mathbf{x} \in \mathbb{R}^{n_j} : 0 \leq g_i(\mathbf{x}) \leq 1,\ i \in J_j\}$, with $j = 1,\ldots,m$. If $\mathcal{X}$ is a compact subset of $\mathbb{R}^n$ then each $\mathcal{X}_j$ is a compact subset of $\mathbb{R}^{n_j}$. As in the dense case, let us note $h_{\boldsymbol{\alpha}_j,\boldsymbol{\beta}_j} := \mathbf{g}^{\boldsymbol{\alpha}_j}(\mathbf{1}-\mathbf{g})^{\boldsymbol{\beta}_j}$, for given $\boldsymbol{\alpha}_j, \boldsymbol{\beta}_j \in \mathbb{N}^{n_j}$.

The following result, a sparse variant of Theorem 4.23, can be retrieved from [164, Theorem 1] but we also provide here a shorter alternative proof by using [160].

Theorem 4.27 (Sparse Krivine-Stengle representations)**.** *Let $f, g_1, \ldots, g_p \in \mathbb{R}[\mathbf{x}]$ be given and assume that there exist I_j and J_j, $j = 1, \ldots, m$, which satisfy the four conditions stated in Definition 4.25. If f is positive over $\mathcal{X}$, then there exist $\phi_j \in \mathbb{R}[\mathbf{x}, I_j]$, $j = 1, \ldots, m$ such that $f = \sum_{j=1}^m \phi_j$ and $\phi_j > 0$ over $\mathcal{X}_j$. In addition, there exist $l \in \mathbb{N}$ and finitely many nonnegative weights $\lambda_{\boldsymbol{\alpha}_j, \boldsymbol{\beta}_j}$, $j = 1, \ldots, m$, such that:*

$$\phi_j = \sum_{|\boldsymbol{\alpha}_j + \boldsymbol{\beta}_j| \leq l} \lambda_{\boldsymbol{\alpha}_j, \boldsymbol{\beta}_j} h_{\boldsymbol{\alpha}_j, \boldsymbol{\beta}_j}\,, \quad j = 1, \ldots, m. \tag{4.31}$$

Proof. From [160, Lemma 3], there exist $\phi_j \in \mathbb{R}[\mathbf{x}, I_j]$ such that $f = \sum_{j=1}^m \phi_j$ and $\phi_j > 0$ on $\mathcal{X}_j$. Applying Theorem 4.23 on each ϕ_j, there exist $l_j \in \mathbb{N}$ and finitely many nonnegative weights $\lambda_{\boldsymbol{\alpha}_j, \boldsymbol{\beta}_j}$ such that $\phi_j = \sum_{|\boldsymbol{\alpha}_j + \boldsymbol{\beta}_j| \leq l_j} \lambda_{\boldsymbol{\alpha}_j, \boldsymbol{\beta}_j} h_{\boldsymbol{\alpha}_j, \boldsymbol{\beta}_j}$. With $l = \max_{1 \leq j \leq m}\{l_j\}$, we complete the representations with as many zero λ as necessary and obtain the desired result. □

In Theorem 4.27, one assumes that f can be written as the sum $f = \sum_{j=1}^m f_j$, where each f_j is not necessarily positive. The first result of the theorem states that that f can be written as another sum $f = \sum_{j=1}^m \phi^j$, where each ϕ_j is now positive. As in the dense case, the $\lambda_{\boldsymbol{\alpha}_j, \boldsymbol{\beta}_j}$ can be computed by equalizing the coefficients in the monomial basis. We also obtain a hierarchy of LP relaxations to approximate the solution of polynomial optimization problems. We now provide these relaxations as well as their computational costs in the particular context of parametric polynomial optimization for reachability analysis in Section **??**.

4.3.2 Set-based simulation using K.S representations

We recall from (4.5) and (4.8) that template reachability analysis with polyhedral templates can be summarized into solving the optimization problem:

$$c_{\tau+1,i} = \max_{(\mathbf{x},\mathbf{k}) \in \hat{X}_\tau \times K} \Gamma_i \mathbf{f}(\mathbf{x}, \mathbf{k})$$

For the sake of keeping close notations to Section 4.3.1, we note $f'(\mathbf{x}, \mathbf{k}) = \Gamma_i \mathbf{f}(\mathbf{x}, \mathbf{k})$ which is polynomial in $\mathbf{x}$ and linear in $\mathbf{k}$. In this particular section, we assume that $\hat{X}_\tau$ (also noted X in the following) is a compact semi-algebraic set include in $[0, 1]^n$. We also assume that K is the hypercube[9] $[-1, 1]^m$.

Here we explain how to compute upper bounds of $\overline{f'} := \max_{(\mathbf{x},\mathbf{k}) \in X \times K} f'(\mathbf{x}, \mathbf{k})$ by using sparse Krivine-Stengle representations. If necessary, we can obtain lower bounds of $\underline{f'} := \min_{(\mathbf{x},\mathbf{k}) \in X \times K} f'(\mathbf{x}, \mathbf{k})$ in a similar way.

Let $\mathbf{g}^X$ be the vector of p polynomial constraints whose conjunction defines the semi-algebraic set X. For the sake of consistency with Section 4.3.1, we introduce the variable $\mathbf{y} \in \mathbb{R}^{n+m}$ defined by $y_i := x_i$, $j = 1, \ldots, n$ and $y_i := k_{i-n}$, $i = n+1, \ldots, n+m$. Then, one can write the set $\mathcal{X} = X \times K$ as follows:

$$\mathcal{X} = \{\mathbf{y} \in \mathbb{R}^{n+m} : 0 \leq g_j(\mathbf{y}) \leq 1\,, \quad j = 1, \ldots, p+m\}\,, \tag{4.32}$$

[9] Or obtained though a linear transformation of $[-1, 1]^m$.

with $g_j(\mathbf{y}) := g_j^X(\mathbf{x})$, for each $j = 1, \ldots, n$ and $g_j(\mathbf{y}) := \frac{1}{2} + \frac{k_j}{2}$, for each $j = p+1, \ldots, p+m$.

Lemma 4.28. *For each $j = 1, \ldots, m$, let us define $I_j := \{1, \ldots, n, n+j\}$ and $J_j := \{1, \ldots, p, p+j\}$. Then the sets I_j and J_j satisfy the four conditions stated in Definition 4.25.*

Proof. The first condition holds as $f'(\mathbf{y}) = f'(\mathbf{x}, \mathbf{k}) = \sum_{j=1}^m s_j(\mathbf{x}, \mathbf{k}) k_j = \sum_{j=1}^m s_j(\mathbf{y}) k_j$, with $s_j(\mathbf{y}) \in \mathbb{R}[\mathbf{y}, I_j]$. The second and third condition are obvious. The running intersection property comes from $I_{j+1} \cap I_j = \{1, \ldots, n\} \subseteq I_j$. $\square$

Given $\boldsymbol{\alpha}_j, \boldsymbol{\beta}_j \in \mathbb{N}^{p+1}$, one can write $\boldsymbol{\alpha}_j = (\boldsymbol{\alpha}'_j, \omega_j)$ and $\boldsymbol{\beta}_j = (\boldsymbol{\beta}'_j, \delta_j)$, for $\boldsymbol{\alpha}'_j, \boldsymbol{\beta}'_j \in \mathbb{N}^p$, $\omega_j, \delta_j \in \mathbb{N}$. In our case, this gives the following formulation for the polynomial $h_{\boldsymbol{\alpha}_j, \boldsymbol{\beta}_j}(\mathbf{y}) = \mathbf{g}^{\boldsymbol{\alpha}_j}(\mathbf{1} - \mathbf{g})^{\boldsymbol{\beta}_j}$:

$$\begin{aligned} h_{\boldsymbol{\alpha}_j, \boldsymbol{\beta}_j}(\mathbf{y}) &= h_{\boldsymbol{\alpha}'_j, \boldsymbol{\beta}'_j, \omega_j, \delta_j}(\mathbf{x}, \mathbf{k}) \\ &= \mathbf{g}^X(\mathbf{x})^{\boldsymbol{\alpha}'_j} (\mathbf{1} - \mathbf{g}^X(\mathbf{x}))^{\boldsymbol{\beta}'_j} (\frac{1}{2} + \frac{k_j}{2})^{\omega_j} (\frac{1}{2} - \frac{k_j}{2})^{\delta_j} . \end{aligned}$$

Example 4.29 (Sparse representation in the parameters)**.** For instance, with the polynomial $f' = (2x^2 - x)k_1 + x^2 k_2 + (x^2 - x)k_3$ depending on x, k_1, k_2, k_3, where $x \in [0, 1] = \{x \in \mathbb{R} \mid 0 \leq x \leq 1\}$ and $\mathbf{k} \in [-1, 1]^3$, one can consider the multi-indices $\boldsymbol{\alpha}_1 = (1, 2)$, $\boldsymbol{\beta}_1 = (2, 3)$ associated to the scaled parameter k_1. Then $h_{\boldsymbol{\alpha}_1, \boldsymbol{\beta}_1}(\mathbf{y}) = x(1-x)^2 (\frac{1}{2} + \frac{k_1}{2})^2 (\frac{1}{2} - \frac{k_1}{2})^3$.

Now, we consider the following hierarchy of LP relaxations, for each $l \geq d$:

$$\begin{aligned} \overline{f'_l} := \min_{t, \lambda_{\boldsymbol{\alpha}_j, \boldsymbol{\beta}_j}} \quad & t , \\ \text{s.t} \quad & t - f' = \sum_{j=1}^m \phi_j , \\ & \phi_j = \sum_{|\boldsymbol{\alpha}_j + \boldsymbol{\beta}_j| \leq l} \lambda_{\boldsymbol{\alpha}_j, \boldsymbol{\beta}_j} h_{\boldsymbol{\alpha}_j, \boldsymbol{\beta}_j} , \quad j = 1, \ldots, m , \\ & \lambda_{\boldsymbol{\alpha}_j, \boldsymbol{\beta}_j} \geq 0 , \quad j = 1, \ldots, m . \end{aligned} \tag{4.33}$$

Similarly, we obtain $\underline{f'_l}$ while replacing $\min$ by $\max$ and $t - f'$ by $f' - t$ in LP (4.33).

Lemma 4.30. *The sequence of optimal values $(\underline{f'_l})$ (resp. $(\overline{f'_l})$) satisfies $\underline{f'_l} \uparrow \underline{f'}$ (resp. $\overline{f'_l} \downarrow \overline{f'}$) as $l \to +\infty$.*

Proof. By construction $(\underline{f'_l})$ is monotone nondecreasing. For a given arbitrary $\varepsilon' > 0$, the polynomial $f' - \underline{f'} + \varepsilon'$ is positive over $\mathcal{X}$. By Lemma 4.28, the subsets I_j and

J_j satisfy the four conditions stated in Definition 4.25, so we can apply Theorem 4.27 to $f' - \underline{f'} + \varepsilon'$. This yields the existence of ϕ_j, $j = 1, \ldots, m$, such that $f' - \underline{f'} + \varepsilon' = \sum_{j=1}^{m} \phi_j$ and $\phi_j = \sum_{|\boldsymbol{\alpha}_j + \boldsymbol{\beta}_j| \leq l} \lambda_{\boldsymbol{\alpha}_j, \boldsymbol{\beta}_j} h_{\boldsymbol{\alpha}_j, \boldsymbol{\beta}_j}$, $j = 1, \ldots, m$. Hence, $(\underline{f'} - \varepsilon', \phi_j, \lambda_{\boldsymbol{\alpha}_j, \boldsymbol{\beta}_j})$ is feasible for LP (4.33). It follows that there exists l such that $\underline{f'_l} \geq f' - \varepsilon'$. Since $\underline{f'_l} \leq f'$, and ε' has been arbitrary chosen, we obtain the convergence result for the sequence $(\underline{f'_l})$. The proof is analogous for $(\overline{f'_l})$. □

Remark 4.31. In the special case of parametric reachability analysis of polynomial systems, one can prove that the number of variables of LP (4.33) is $m\binom{2(p+1)+l}{l}+1$ with a number of constraints equal to $[\frac{ml'}{n+1} + 1]\binom{n+l'}{l'}$. This is in contrast with the dense case where the number of LP variables is $\binom{2(p+m)+l}{l} + 1$ with a number of constraints equal to $\binom{n+m+l'}{l'}$.

Proof of Remark 4.30. We replace the representation of a function ϕ of dimension $(n+m)$ on the set $\mathcal{X}$ by a sum of m functions ϕ_j of dimension $(n+1)$ defined on their associated subsets $\mathcal{X}_j$. From Section 4.3.1, the number of coefficients $\lambda_{\boldsymbol{\alpha}_j, \boldsymbol{\beta}_j}$ for the K.S. representation of a ϕ_j over $\mathcal{X}_j$ is $\binom{2(p+1)+l}{l}$. This leads to a total of $m\binom{2(p+1)+l}{l}$ for all the ϕ_j and $m\binom{2(p+1)+l}{l} + 1$ variables when adding t.

The number of equality constraints is the number of monomials involved in $\sum_{j=1}^{m} \phi_j$. Each ϕ_j has $\binom{(n+1)+l'}{l'}$ monomials. However there are redundant monomials between all the ϕ_j: the ones depending of only $\mathbf{x}$, and not $\mathbf{e}$. These $\binom{n+l'}{l'}$ monomials should appear only once. This leads to a final number of $m\binom{(n+1)+l'}{l'} - (m-1)\binom{n+l'}{l'}$ monomials which is equal to $[\frac{ml'}{n+1} + 1]\binom{n+l'}{l'}$. □

Example 4.32 (K.S. parametric optimization)**.** Continuing Example 4.15 and 4.29, for the polynomial $f' = (2x^2 - x)k_1 + x^2 k_2 + (x^2 - x)k_3$, we consider LP (4.33) at the relaxation order $l = d = 3$ over $\mathcal{X} = [0,1] \times [-1,1]^3$. This problem involves $3\binom{2\times(1+1)+3}{3} + 1 = 106$ variables and $[\frac{3\times 3}{2} + 1]\binom{4}{3} = 22$ constraints. This is in contrast with a dense Krivine-Stengle representation, where the corresponding LP involves 35 linear equalities and 166 variables. Computing the values of $\overline{f'_l}$ provides an upper bound of 2, yielding $f'(\mathbf{x}, \mathbf{k}) \leq 2\varepsilon \quad \forall (\mathbf{x}, \mathbf{k}) \in [0,1] \times [-1,1]^3$.

The algorithm for reachability analysis of discrete time polynomial systems using K.S. representations is more straightforward than Algorithm 3. Indeed, using K.S. representations we avoid the decomposition of the polyhedrons into parallelotopes bundles. In previous Algorithm 3, these decompositions were not only costly as they multiplied the number of necessary optimizations at each step by the number parallelotopes, but also less accurate as Bernstein method does not directly handles variables in polyhedron.

Algorithm 4 KS_One_Step_Reach(X,Ψ,Γ,K)

INPUT: X: Current set (template polyhedron)
INPUT: Ψ: BSPT and its associated piecewise approximation of the dynamics.
INPUT: Γ: template which is used to over-approximate the reachable set.
INPUT: K: parameters set
OUTPUT: an over-approximation of the reachable set of X after one step.

$\mathcal{G}$: set of current nodes, or the tree root $\top$

$(\mathcal{L},\mathcal{X})$ = Ψ.`locating`$(X,\mathcal{G})$ /* *finding intersecting nodes, and associated intersections (still polyhedron)* */

for $\Gamma_i \in \Gamma$ **do**
    /* *computing offset bound c_i for each constraint Γ_i in the template* */

    for $(\mathcal{L}_\kappa, X_\kappa) \in \mathcal{L}$ **do**
        Compute the approximate dynamics f_κ associated with the node $\mathcal{L}_\kappa$
        Construct the polynomial $\eta_{i,\kappa}$ from polynomial f_κ of the dynamics and
        template constraint Γ_i, as defined in (**??**).

        $c_{i,\kappa} = \max\{\eta_{i,\kappa}(\mathbf{x},\mathbf{k})) \mid \mathbf{x} \in X_\kappa \wedge \mathbf{k} \in K\}$
        /* *using Sparse K.S representation for polynomial $\eta_{i,\kappa}$* */

    end for

    $c_i = \max_\kappa(c_{i,\kappa})$
    /* *largest bound by all approximate dynamics of intersecting nodes* */

end for
return $(\Gamma, \mathbf{c})$ /* *the result is the template polyhedron with offsets* $\mathbf{c}$ */

In Algorithm 4, we present our algorithm for one step of the discrete time reachability analysis. We note that we did not introduce a method to apply K.S. representations to rational functions, but this can be perform through variable changes, in a similar fashion to [1].

Finally, we do not provide any cases study applying K.S. for reachability analysis as an efficient implementation is still on-going. However, we propose Section. 4.4 a discussion on our preliminary results on the comparison between Bernstein expansion and K.S. representation applied to parametric polynomial optimization problems. This results are from outside the field of system biology, and were published in [2] for bounding the floating point roundoff error. They still give a groundwork for a choosing policy between these two methods for polynomial optimization.

4.4 Discussion and perspectives

Theoretical complexity. In this section we provide a preliminary comparison between Bernstein expansion and K.S representations as two methods to solve parametric polynomial optimizations problems. We recall that we search $\overline{f}$ an upper bound of the optimization problem:

$$f^* := \max_{(\mathbf{x},\mathbf{k})\in X\times K} f(\mathbf{x},\mathbf{k}) \leq \overline{f}\,,$$

where f is polynomial in $\mathbf{x}$ and linear in $\mathbf{k}$.

The Bernstein expansion method to provide an upper bound $\overline{f}$ when $X \subset \mathbb{R}^n$ is a box, and $K \subset \mathbb{R}^m$ is either a box or a polyhedron. If K is a box then we recall from Section 4.3.1 that the theoretical complexity is at most equal to the number of Bernstein coefficients, which is $m(\mathbf{l}+\mathbf{1})^{\mathbf{1}}$. The cost is linear in the number of parameters m, polynomial in the degree $\mathbf{l}$ (at fixed dimension n), and exponential in dimension n (at fixed degree $\mathbf{l}$). If K is a (compact) polyhedron then we have to solve at most $(\mathbf{l}+\mathbf{1})^{\mathbf{1}}$ linear programs of m variables and p constraints, where p is the number of constraints defining K.

From this theoretical cost we note that the Bernstein based method greatly suffers when dimension increases. Methods such as [132] can reduce this cost when there is some sparsity in f, but in general Bernstein perform badly in high dimension. However when dimension is small, as the Bernstein expansion is polynomial in degree it is possible to obtain efficiently a precise bound of a high degree polynomial.

We note that if X is a polyhedron we have to define an associated bundle of parallelotopes $\{\Pi_i\}_{i \leq n_{\mathcal{P}}}$. However, we have no guarantee of convergence to the optimal solution by optimizing over the bundle as described in Section 4.2.2. Moreover, as we have to perform one optimization for each parallelotope, the computational cost is now linear in the number of parallelotopes constituting the bundle: $n_{\mathcal{P}} m(\mathbf{l}+\mathbf{1})^{\mathbf{1}}$.

The K.S. representations based method provides bounds when X is a semi-algebraic compact set, and K either a box or a semi-algebraic compact set. In the case where K is a box, we can use the sparse K.S. representation as described in Section 4.3.1. This yields a relaxation of the polynomial optimization problem into an linear programs of $m\binom{2(p+1)+l}{l}+1$ variables and $[\frac{ml'}{n+1}+1]\binom{n+l'}{l'}$ constraints, where n is the dimension of $\mathbf{x}$, p the number of constraints defining X, m the dimension of the parameters $\mathbf{k}$, and the adjusted degree[10] $l' = ld^{\mathbf{g}}$.

From [165],[166] we know that the computation cost of an LP resolution is polynomial in the number of constraints and variables. Consequently, when the the degree k is fixed this yields a polynomial cost in the dimension n or p the number of constraints defining X. We note that the complexity is linear in m allowing to efficiently handle parameters as the Bernstein-based method. Experimentally (see Table 4.1), the K.S. based method seems to have a large cost when the degree increase, and the dimension is fixed.

When K is a semi-algebraic compact set, we have to use the dense K.S. representation over the whole set $X \times K$, leading to the important computational cost: variables and constraints. To reduce the cost, it may be possible to provide a problem specific sparsity pattern: however finding an optimal sparsity pattern is an NP-hard problem [167].

In our reachability analysis application, we always compute linear relaxation of order $l = d$, or Bernstein expansion of multi-degree $\mathbf{l} = \mathbf{d}$: thus by fixing the

[10] Note that $l' = l$ in the particular case of polytopes.

Benchmark	n	m	d	Bern (`double`)	Bern (`exact`)	K.S.
`ex-2-2-5`	2	9	3	**3e–4**	**3e–4**	0.13(0.02)
`ex-2-2-10`	2	14	3	**4e–4**	**4e–4**	0.18(0.02)
`ex-2-2-15`	2	19	3	**5e–4**	**5e–4**	0.24(0.03)
`ex-2-2-20`	2	24	3	**5e–4**	8e–4	0.30(0.03)
`ex-2-5-2`	2	9	6	**2e–3**	3e–3	1.08(0.14)
`ex-2-10-2`	2	14	11	**2e–2**	4e–2	90.1(53.1)
`ex-5-2-2`	5	12	3	**7e–3**	4e–2	0.63(0.05)
`ex-10-2-2`	10	22	3	**2.48**	1242	5.5(0.3)

Table 4.1: Comparison of execution times (in seconds) for examples generated from (**??**). First and second column are the execution times using Bernstein optimization with either double precision or rational arithmetic (exact precision). The last column details execution times for Krivine-Stengle method. For Krivine-Stengle the CPLEX solving time is given between parentheses. For each polynomial, the best results are emphasized using **bold fonts**.

degree we know that for high dimension K.S. is more efficient that Bernstein. If this relaxation order, or this multi-degree is not high enough to provides accurate bounds, we prefer to split the set than increase the degree. From these theoretical consideration we argue that K.S. is a good complement to Bernstein expansion when optimizing high dimensional polynomial with low degree. K.S. can also handle more complex representation for X than Bernstein when accuracy is a key factor of the analysis. Thus, we argue that a policy choosing at each step the optimization method to use depending of the context would be more efficient.

Experimental results. Finally, we provide a table with experimental results comparing K.S. and Bernstein for polynomial optimization of boxes. This is a subset of the benchmarks performed in our paper [2]. The studied examples are of the form:

$$\texttt{ex-n-sum-d}(\mathbf{x}) := \sum_{j=0}^{\texttt{m}}(\prod_{l=1}^{\texttt{d}}(\sum_{i=1}^{\texttt{n}} x_i))\,. \tag{4.34}$$

with parameters taken in $[-1,1]^m$. Their dimension m is determined separately in [2]. The dimension of $\mathbf{x}$ is n and $\mathbf{x}$ is also taken in $[-1,1]^n$. Finally, and d is the degree of the polynomial. We also note that the associated multi-degree is $\mathbf{d} = (d,\ldots,d)$. To perform Bernstein expansion we re-use parts of the `C++` implementation from `sapo` [83] based on the matrix method [65]. The optimization using sparse K.S. representations is currently implemented in interpreted `Matlab 2015a` as a modification of the previous toolbox SBSOS implemented for [164]. Experimentations were performed on an Intel Core i7-5600U (2.60Ghz, 16GB) with Ubuntu 14.04LTS, using GINAC 1.7.1 version in association to `sapo`, and CPLEX 12.63 to solve linear programs from K.S. representation method. In Table 4.1 we compare the performances of the two methods. Taking into account that

Benchmark	n	m	d	Bernstein	Krivine-Stengle
`ex-2-2-5`	2	9	3	201	201
`ex-2-2-10`	2	14	3	480	480
`ex-2-2-15`	2	19	3	860	860
`ex-2-2-20`	2	24	3	1342	1342
`ex-2-5-2`	2	9	6	1504	1504
`ex-2-10-2`	2	14	11	94576	94576
`ex-5-2-2`	5	12	3	770	770
`ex-10-2-2`	10	22	3	4648	4648

Table 4.2: Comparison of the accuracy for examples generated from (**??**). The first column contains the upper bounds using Bernstein optimization with either double precision or rational arithmetic (exact precision). The second column details the upper bounds given by Krivine-Stengle method.

the LP generation for K.S. is implemented in interpreted `Matlab` we provide in parenthesis the solving time of the LP by `Cplex`. From the results of examples `ex-2-2-5` to `ex-2-2-20`, we observe that accordingly to the theoretical cost both methods are linearly affected by the number of parameters. Comparing examples `ex-2-5-2` and `ex-2-10-2`, we note that K.S. is strongly affected by degree elevation, while in examples `ex-5-2-2` and `ex-10-2-2`, Bernstein is greatly affected by dimension increase. In particular in example `ex-10-2-2`, we note that the LP solving time related to K.S. method is small, which agree with the polynomial cost of K.S. at fixed degree.

Finally we note that the current implementation of K.S. is limited by the LP generation time in `Matlab`, and a new implementation in a compiled language must provided in order to obtain more results, and a reliable implementation for reachability analysis. In Table 4.2, we compare the accuracy of the two method at the smallest relaxation order $l = d$, and the smallest multi-degree for the Bernstein expansion, $\mathbf{l} = \mathbf{d}$. This provides a similar context to an application to reachability analysis where we focus on first approximation results. In all examples both methods have the same accuracy, further benchmarks are provided in [2] which yield a similar conclusion. Consequently, there is apparently no loss in accuracy by using K.S. instead of Bernstein when it is theoretically more cost efficient.

In the following we describe our perspectives and future work for both K.S. and Bernstein based methods. The future works associated to this chapter can be divided in three: the further development of the current implementation, and two different approaches to perform reachability analysis.

Future implementations. First we need to improve the performances of the current implementation of the K.S. based optimization method and to integrate it inside a toolbox for reachability analysis. Indeed, the current implementation in

Matlab (based on [164]) cannot efficiently be applied on large problems (see Table 4.1).

As previously stated, our results on the Bernstein-based method are already an improvement of the reachability toolbox `sapo`. However, the current implementation relies on an old version of `sapo`, and it is now important to merge our new features into a more recent version.

Both algorithms 3 and 4 have been detailed for reachability analysis of discrete time systems. If they produce safe and conservative results in this context, they are not guaranteed to be conservative for continuous time systems. To this aim we need to implement a method to efficiently handle numerical integration error, and we need to compute the continuous time corrected initial set, [87], $\texttt{correct}(X_0)$ defined as:

$$\texttt{correct}(X_0) := X_0 \oplus \mathbf{f}(X_0, K)\,, \tag{4.35}$$

where $\oplus$ is the Minkowski sum operator.

Finally, it is important to provide an implementation for hybrid automata similar to [87], to handle hybrid systems with polynomial continuous dynamics, as well as experimental protocol models we define further in Section 5.

Fixed time reachability analysis. In addition to future improvements of the current implementation we identify some new ideas which seems to be promising. The first idea is the use of simulation and approximation methods to compute reachable set at fixed instant. Given the ODE system:

$$\dot{\mathbf{x}}(t) = \mathbf{f}(t, \mathbf{x}), \quad \mathbf{x} \in \mathbb{R}^n\,. \tag{4.36}$$

Let us consider a closed time interval $[0, T] \subset \mathbb{R}$. For all $t \in [0, T]$, $\xi_{\mathbf{x}_0}(t)$ denotes the trajectory solution of (4.36), with initial condition $\mathbf{x}(0) = \mathbf{x}_0$. By extension, $\forall t \in [0, T]$, $\xi_{X_0}(t)$ is the set of trajectories for all $x_0 \in \mathcal{X}_0$.

We want to address the following problem: for a given time $t \in [0, T]$, we consider $\phi_t(x) : \mathbb{R}^n \to \mathbb{R}^n$, the function which gives the value at time t of the trajectory $\xi_{\mathbf{x}}(\cdot)$. Instead of computing the whole flow pipe, we only want to find an over-approximation of the set $X_t = \phi_t(x)$, $\forall x \in X_0$. Results from biological experiments are often time series, with sometimes samples separated by hours. We argue that interpolating some continuous temporal properties from these time series is already a strong assumption on the model. Thus we seek to ensure that given some ODEs modelling our system, all the trajectories associated to the model reach the desired data points, without assuming anything in between.

To compute X_t we ideally need the exact analytical function $\phi_t(X_0)$. In simple cases, for example with linear autonomous ODE systems, we have this function. For the ODE system, $\dot{\mathbf{x}} = A\mathbf{x}, \mathbf{x}(0) \in X_0$, we have $\phi_t(X_0) = X_0 e^{-At}$.

However, in the general case we may not know this analytical solution and in particular for $\mathbf{f} \in \mathbb{R}[\mathbf{x}]$. For this reason we will search an approximation $\hat{\phi}_t(x)$ of $\phi_t(x)$ and the associated approximation error defined by $\|\phi_t(x) - \hat{\phi}_t(x)\| \leq$

ε_{abs}, $\forall \mathbf{x} \in X_0$. This way we can build a new over-approximation of X_t by:

$$\hat{X}_T = \mathcal{O}(\hat{\phi}_t(X_0)) \oplus \varepsilon_{abs} \tag{4.37}$$

Assuming that $\phi_t(x)$ is at least $\mathcal{C}^1(\mathbb{R}^n)$, we propose to use polynomial interpolation over the trajectories from an initial sample of X_0 obtained by numerical simulation, to determine this approximation. For two or three dimensional systems we can use interpolation over a Padua-Grid and Chebychev polynomials [168] to obtain a precise polynomial approximation $\hat{\phi}_t(\cdot)$ of $\phi_t(\cdot)$. The template over-approximation $\mathcal{O}\cdot$ can be performed using the Bernstein or K.S. based methods described in this chapter. In Fig 4.9, one can see the results of box approximations using the Bernstein expansion at fixed time points of the Lotka-Volterra continuous model (LV-ODE) from Example 4.1. If this method performs well in two dimen-

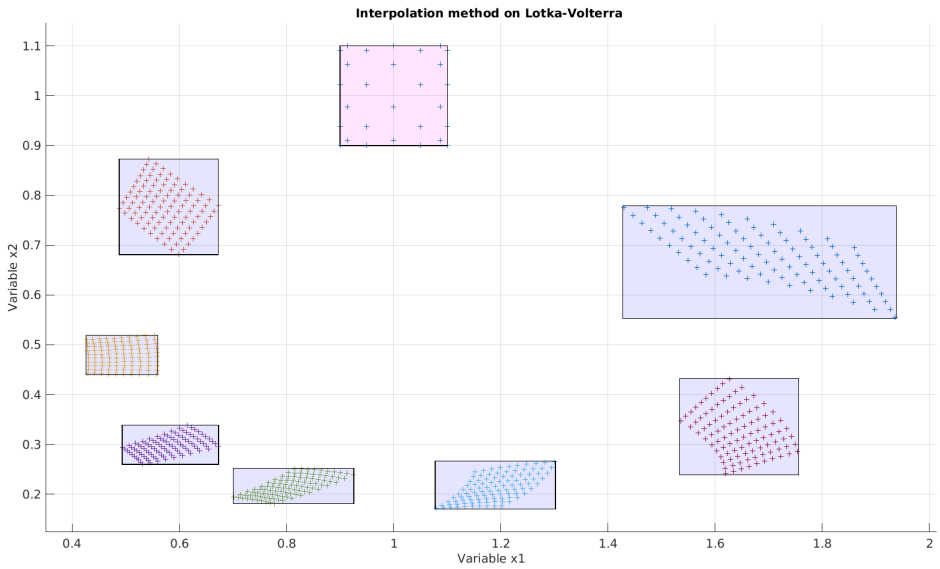

Figure 4.9: Interpolation method for reachability analysis applied on the Lotka-Volterra ODE system. The '+' dots are the numerical simulations of the 2D Padua grid of the initial set: the box $[0.9, 1.1]^2$. The other boxes are the over-approximation the reachable set using polynomial approximations $\hat{\phi}_t(X_0)$ interpolated from simulations on the initial set grid.

sion, it is not viable without modification for dimension higher than 3. Indeed, using dense grid of the state-space results in an exponential number of samples, and thus simulations. Moreover, the theoretical accuracy of the polynomial approximation quickly decreases if the degree of the approximation $\hat{\phi}_t(\cdot)$ remains fixed while the dimension increases. To tackle the problem of the size of the grid exponential in the dimension we plan to use sparse grid such as describes in [169]. The work of [169] also provide a scheme to converge to an accurate polynomial approximation up to a given error ε_{abs} using sparse grids. However, it would be

a great improvement to obtain an even more accurate value for ε_{abs} than the one provided by their bounds. For this, one idea would be to obtain an upper bound of $\max_{\mathbf{x}\in X} |(\phi_t(x) - \hat{\phi}_t(x))|$, using occupation measure relaxation methods similar to the one described in Chapter 3 (we recall that $\phi_t(x)$ is not analytically known).

Polynomial lift using Bernstein expansion. Carleman linearisation is a method to approximate non-linear vector field by a high dimensional linear transformation. The previous work [170] provides an error bound on the approximation of a non-linear ODE by its Carleman linearisation. It is possible to extend [170] to reachability analysis in three steps: first we lift the initial set into the higher dimensional space. Then, we perform reachability analysis in this high dimensional space using linear reachability analysis tools such as [87] or [171]. Finally, we project the result into the original low dimensional space while taking into account some linearisation error given by [170]. We propose to approximate the polynomial lift described in [170] using the Bernstein expansion to compute a convex over-approximation using the convex hull of the coefficients. For performance purposes in the high dimensional cases, we want to use the linear reachability tool [171]. To this aim, we propose to compute an over-approximation of the lift in Cartesian product format, which directly usable by [171]. In Figure 4.10, we show the polynomial lift $\mathcal{L} : \mathbb{R} \to \mathbb{R}^3$, $\mathcal{L}(x) = (x, x^2, x^3)$ as well as two possible over-approximations using the Bernstein expansion.

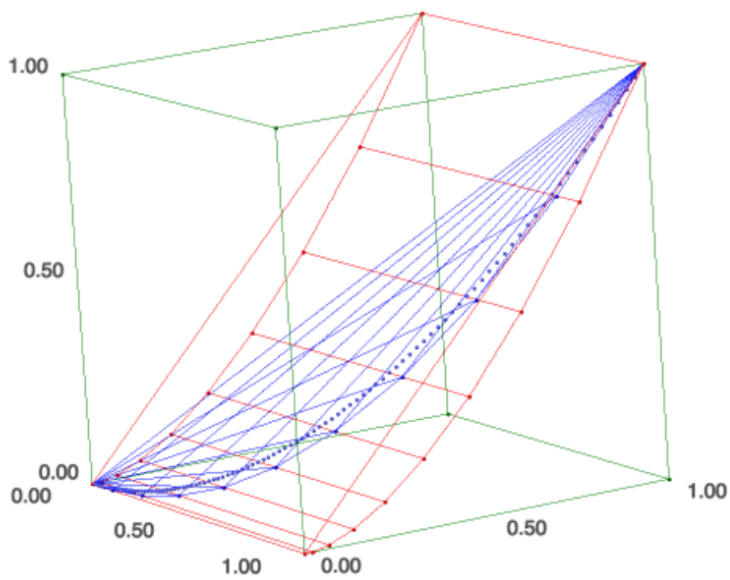

Figure 4.10: Polynomial lift (x, x^2, x^3) of the interval $[0, 1]$ (blue dotted-line) approximated by either the convex hull of the Bernstein control points (blue enclosure), or its Cartesian product decomposition (red enclosure). The green box would be a box over-approximation of the polynomial lift.

CHAPTER 5

MOEPLA: a language for experimental protocol modelling

5.1 MOEPLA description 85
5.2 MOEPLA specification to hybrid automata 89
5.3 Discussion and future work 93

In this chapter, we focus on biological mechanism modelled using ordinary differential equations (ODEs). They can represent mechanisms on multiple scales including the molecular scale, cell scale or physiological scale. The motivation for such models is not only to give quantitative predictions, when sufficient data is available to validate the model, but also to provide a mathematical representation of a biological system. Such formalization then provides a framework to incorporate new data and knowledge of various types in a consistent way.

We address the models corresponding to a class of biological experiments in which the system, in a given initial state, is perturbed in some way, evolves, and then at a later time some measurements are performed. To represent such behaviours, we provide a mathematical formalization of experimental protocols. A model in this sense describes the system under study and the experiments which have been performed on it. To achieve this, a formalization as hybrid automata is proposed for this class of experimental protocols. With the hybrid automata formalism, we introduce a systematic way to study mechanistic biological models, in their experimental context, using formal methods which produce conservative or certifiable results.

In this chapter, we first propose, in Section 5.1, a high level "action based" specification language of experimental protocols, named Modelling-Oriented Experimental Protocol Language (MOEPLA). This action-based language serves as

an interface between the experimental view and the mathematical modelling of a given mechanism. Using this description, an abstract experimental protocol and its execution are formally represented in MOEPLA using an explicit semantic. Moreover, it is expressive enough to be applied in numerous biological contexts such as therapeutic modelling, pharmacology, or molecular modelling. It can be seen as the first step of a pipeline to study in a systematic manner the experimental protocols. Other protocol specification languages, such as EXACT [172], focus on expressing all the stages and details of an experiment to enable its real life reproducibility in another laboratory. In particular, our specification language expresses experimental protocols while taking into account insight on specific mechanism models. Of course, this may lead to neglecting some particular stages or details of the experiments since they are either without any influence, or too complex, for the chosen mechanistic modelling purposes.

There are multiple results [172, 173, 174, 175] on formal languages for the description of experimental protocols in some particular context (experiments over genes, or proteins). The development of the Systems Biology Markup Language (SBML) format [176] allows representing biological data and models using process algebra. These efforts are motivated by the inherent ambiguity in natural language for describing real-life experiments, which can result in the lack of repeatability of a given biological laboratory protocol.

In Section 5.2, we provide a preliminary translation from this specification language into a hybrid automaton, an expressive mathematical formalism on which a panel of analysing tools already exist [87, 47, 88]. Using hybrid automata, discrete changes describe the different stages of the protocol, while ODEs describe biological mechanisms. This hybrid automaton representation can be analyzed using formal validation, providing conservative or certified results that a proposed biological mechanism is coherent with experimental observations. The general workflow defined by the conjoint use of MOEPLA, the hybrid automaton representation and an analysis tool, is described in Figure 5.1

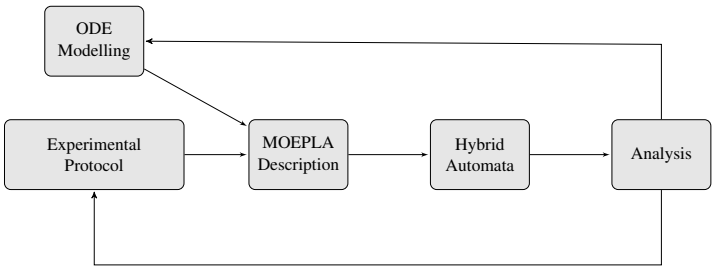

Figure 5.1: Biological experiment modelling workflow.

The hybrid formalism has previously been used as an abstraction method to simplify mechanistic models which are complex and hard to analyse [111], or to represent activation and switch processes such as in the genes regulatory networks [11]. To our best knowledge, except for some general guidelines [177], the closest

work on the formalization of biological protocol incorporating mechanistic models is [178]. This recent work proposes a formalization as piecewise Markov-process of biological protocol. This work differs from ours on the mathematical formalization and seems to be designed to address models from biochemistry protocol. In our case, we want to handle a larger panel of possible experimental protocols: from biochemistry to therapy or animal study. In this chapter, the model of haemoglobin production in erythroblast from [3] is used as a running example and will be further analyzed in Chapter 6.

5.1 MOEPLA description

In this section, we define the formal specification language MOEPLA. We aim to mix experimental protocol specifications with modelling hypotheses on specific biological mechanisms. We note that this specification language is useful when the experimental protocol exhibits multiple step influences on the behaviour of the entities involved in the biological mechanism. This is especially true for particular evolving systems such as differentiating cells, or different stages of an organism's life. On the downside, if it is possible to express short protocols constituted only of one stage (one initial state and a final measurement), this is not the case for which MOEPLA is the most useful. We now define the main notions used to describe a protocol.

The protocol we construct is organized around two main notions: experimental objects and experimental actions, respectively denoted by $\mathcal{O}$ and $\mathcal{A}$. We call experimental object, an object (in a physical sense) on which an experimental protocol action is applied at a given time. Following the context, it can represent objects of multiple scales, such as the animals or cells. A given protocol can have different experimental objects, for different stages of the protocol. As we focus on the modelling of the effect of a protocol on a given biological system, we need information on the associated biological mechanism. This mechanism is represented by a parametric dynamical system $\mathcal{F} = (\mathcal{T}, X, U, K, \mathbf{f})$. Thus each experimental object $\mathcal{O}$ is connected to a biological mechanism $\mathcal{F}$.

Definition 5.1 (Experimental object)**.** Let $\mathcal{F}_\mathcal{O}$ be a parametric dynamical system:

$$\mathcal{F}_\mathcal{O} := (\mathcal{T}_\mathcal{O}, X_\mathcal{O}, U_\mathcal{O}, K_\mathcal{O}, \mathbf{f}_\mathcal{O}).$$

We define by:

$$\mathcal{O} := \mathcal{O}(\mathcal{F}_\mathcal{O}) \,, \tag{5.1}$$

an experimental object $\mathcal{O}$ associated to the parametric dynamical system $\mathcal{F}_\mathcal{O}$. In this context, $\mathcal{F}_\mathcal{O}$ represents various mechanisms related to this particular object the biologist wants to study. Then, $\mathcal{T}_\mathcal{O}$, $X_\mathcal{O}, U_\mathcal{O}, K_\mathcal{O})$ respectively are the default time interval associated to $\mathcal{O}$, its default state space, input space, and parameters space; and $\mathbf{f}_\mathcal{O}$ are parametric ODEs with input, which models the dynamics of the

variables of $\mathcal{O}$. In the following, we assume that $\mathcal{T}_{\mathcal{O}} = \mathcal{T} = [0, T]$ for all the objects.

Remark 5.2 (Notations)**.** For a given experimental object $\mathcal{O}$, let $X_{\mathcal{O}} \subseteq \mathbb{R}^{n_{\mathcal{O}}}$, then $\mathbf{x}^{\mathcal{O}}$ is the set of state variables associated to $\mathcal{O}$, with $\mathbf{x}^{\mathcal{O}} \in X_{\mathcal{O}}$. Let $\{\mathcal{O}^{(\alpha)}\}_{\alpha}$ be the set of all the experimental objects involved in a protocol. Given $n = \sum_{\alpha} n_{\mathcal{O}^{(\alpha)}}$, then let $\mathbf{y} \in \mathbb{R}^n$ be the concatenation of all the variables $\mathbf{x}^{\mathcal{O}^{(\alpha)}}$, that is the set of all variables associated to the whole experimental protocol.

The notations $\mathbf{k} \leftarrow K_{new}$ and $\mathbf{u} \leftarrow U_{new}$) define the redefinition operators which associate respectively a new parameters space K_{new} and input space U_{new}, to respectively the subset of parameters $\mathbf{k}$ and subset of inputs $\mathbf{u}$.

And $\mathbf{x} \leftarrow R(\mathbf{z})$ defines the operation which associates to a subset $\mathbf{z}$ of variables $\mathbf{x}$ a new value $R(\mathbf{z})$, via the reset map R.

Definition 5.3 (Conditional Statement)**.** A conditional statement c is a function $c : \mathcal{T} \times \mathbb{R}^n \to \mathbb{B}$ such that for a given pair $(t, \mathbf{y})$, $c(t, \mathbf{y})$ decides if a condition on $\mathbf{y}$ is satisfied or not. Such conditional statements will be used to define the condition under which a particular step of the protocol must be applied.

We define experimental actions as actions which are performed by the biologist during the experiment and are relevant for the modelling. They include: actions which *create* an experimental object $\mathcal{O}$ at a given step of the experiment; actions performed on an experimental object $\mathcal{O}$, *which have an influence* on its associated dynamics; and finally, the measurement performed on a given experimental object. We associate to an experimental action, a conditional statement $c(t, \mathbf{y})$. In the general case, we define 7 kinds of actions which can be separated in 3 classes: creation ($I_0, \mathcal{A}_{create}, \mathcal{A}_{cr_from}$), perturbation ($\mathcal{A}_x, \mathcal{A}_k, \mathcal{A}_u$), measurement ($\mathcal{A}_{\mathcal{M}}$). We say that an action is executed when the effect of this action on the associated object takes place. We say that an object $\mathcal{O}$ is created when it is part of the initialisation $I_0(\mathcal{O})$, or when one of the actions $\mathcal{A}_{create}(\mathcal{O})$ or $\mathcal{A}_{cr_from}(\mathcal{O}, \dots)$ are executed. We assume that an experimental protocol always starts at time $t = 0$.
First we define the actions creating a new experimental object. These actions include the initialisation step $I_0(\mathcal{O})$ which defines the initial state of the experiment for a given object $\mathcal{O}$. In the following definitions, the symbol L represents a synchronisation label associated to a particular action.

Definition 5.4 (Experimental actions - Creation)**.**

- $I_0(\mathcal{O}, \mathbf{x}^{\mathcal{O}}(0) := \mathbf{x}_0^{\mathcal{O}})$: this creates an initial object $\mathcal{O}$, and initialises its state variables to $\mathbf{x}_0^{\mathcal{O}}$.
- $\mathcal{A}_{create}(\mathcal{O}, L, \mathbf{x}^{\mathcal{O}}(t) := \mathbf{x}_t^{\mathcal{O}}, c(t))$: this action defines the creation of a new object $\mathcal{O}$ under the condition $c(t) = \texttt{true}$, with some initial value $\mathbf{x}_t^{\mathcal{O}}$. The condition c only depends on t in this particular case. This action allows creating a new object at a given time independently of another experimental object.

– $\mathcal{A}_{cr_from}(\mathcal{O}, \{\mathcal{O}^{(\beta)}\}, L, \mathbf{x}^{\mathcal{O}}(t) \leftarrow R(\mathbf{x}(t)), c(t, \mathbf{x}))$: this action defines the creation of $\mathcal{O}$ from a given set of experimental objects $\{\mathcal{O}^{(\beta)}\}$, under the condition $c(t, \mathbf{x}(t)) = \texttt{true}$. Here, $\mathbf{x}$ is the concatenation of the variables $\mathbf{x}^{\mathcal{O}^{(\beta)}}$. The initial condition $\mathbf{x}^{\mathcal{O}}(t)$ of $\mathcal{O}$ is determined by a reset map on the state variables $\mathbf{x}(t)$ of the set of objects $\{\mathcal{O}^{(\beta)}\}$.

Secondly, we define the actions which perturb a given experimental object $\mathcal{O}$, by changing its state variables, its parameters space, or its input space. In real life, if an action has multiple effects, then in the specification language, this is translated by multiple actions under the same condition.

Definition 5.5 (Experimental actions - Perturbation)**.**

– $\mathcal{A}_k(\mathcal{O}, L, \mathbf{k} \leftarrow K_{new}, c(t, \mathbf{x}^{\mathcal{O}}))$: the action performed on $\mathcal{O}$ affects the parameters $\mathbf{k}$, changing the associated parameters space to K_{new}, under the condition $c(t, \mathbf{x}^{\mathcal{O}}(t)) = \texttt{true}$.

– $\mathcal{A}_x(\mathcal{O}, L, \mathbf{x}^{\mathcal{O}} \leftarrow R(\mathbf{x}^{\mathcal{O}}), c(t, \mathbf{x}^{\mathcal{O}}))$: the action performed on $\mathcal{O}$ affects the state variables $\mathbf{x}^{\mathcal{O}}$, changing their values to $R(\mathbf{x}^{\mathcal{O}})$, under the condition $c(t, \mathbf{x}^{\mathcal{O}}(t)) = \texttt{true}$.

– $\mathcal{A}_u(\mathcal{O}, L, \mathbf{u} \leftarrow U_{new}, c(t, \mathbf{x}^{\mathcal{O}}))$: the action performed on $\mathcal{O}$ affects the input functions $\mathbf{u}$, changing the associated input space to U_{new}, under the condition $c(t, \mathbf{x}^{\mathcal{O}}(t)) = \texttt{true}$.

Example 5.6 (Perturbation action)**.** An example of experimental action is as follows: at exactly 3 hours after the start of the experiment, the biologist increases the temperature to $37°C$. It has the effect on a production rate k_{prod} associated to the experimental object $\mathcal{O}_1$. Then the experimental action is: temperature increase to $37°C$ on $\mathcal{O}_1$ affecting k_{prod} under condition $t == 3$h, and expressed as

$$\mathcal{A}_k(\mathcal{O}_1, L_{T\uparrow}, k_{prod} \leftarrow \{k_{prod}(37°\text{C})\}, t == 3h).$$

In this example, the synchronization $L_{T\uparrow}$ allows to gather all the perturbation associated to this perturbation: T increases to $37°C$. One of the perturbation of this actions is described by $\mathcal{A}_k$: the parameter space of k_{prod} is changed to the singleton $\{k_{prod}(37°\text{C})\}$.

Finally, we define the action corresponding to an experimental measurement.

Definition 5.7 (Experimental actions - Measurement)**.**

– $\mathcal{A}_{\mathcal{M}}(\mathcal{O}, L, \mathbf{x}_{meas}(t) := m(\mathbf{x}^{\mathcal{O}}(t)), c(t, \mathbf{x}^{\mathcal{O}}))$: the action performed on $\mathcal{O}$ is a measurement of the state variables of $\mathcal{O}$, under the condition $c(t, \mathbf{x}^{\mathcal{O}}(t)) = \texttt{true}$. The information of the measurement is preserved in a variable $x_{meas}(t)$, for all time instants t which satisfy the condition.

We use measurement actions to define possible (or actual) observation windows during the experiment. These action are parallel to actual experiment, and do not influence it directly. If an experimental measurement does have in reality an effect of the observed object, then it is represented as a perturbation action with the same condition.

The set of possible experimental actions, with the exception of the initialisation I_0, is summarized as follows:

$$\mathcal{A} := \mathcal{A}_{create}, \mathcal{A}_{cr_from}, \mathcal{A}_k, \mathcal{A}_x, \mathcal{A}_u, \mathcal{A}_{\mathcal{M}}$$

It is important to notice that we expect the biologist to have some insight on how an action will affect the model. However, we do not actually expect him to always provide explicit mathematical functions, or numerical values, when specifying the protocol in our language. In the following we may use dots, as in $\mathcal{A}_k(\mathcal{O}, L, \ldots, c)$, to ease the redaction of non-relevant information in the context of a given explanation. These dots are not part of the language.

Definition 5.8 (Experimental Protocol Syntax)**.** We call experimental protocol a set of actions $\{\mathcal{A}\}$ performed on objects $\{\mathcal{O}\}$, the objective of which is to highlight a biological mechanism, and to measure directly or indirectly its evolution, in time and/or after some events. The set $\{I_0\}$ provides the initial conditions of a protocol. A protocol must end with a final measurement $\mathcal{A}_{\mathcal{M}}$. After defining the variables, dynamical systems, protocol maximal duration t_{end}, experimental objects $\mathcal{O}(\mathcal{F})$ that will appear in the experiment, as well as their (possible) initial conditions $I_0(\mathcal{O})$, we express the protocol as a sequence:

$$\mathcal{P} := I_0^* \mathcal{A}^* \mathcal{A}_{\mathcal{M}} \,,$$

where I_0^* and $\mathcal{A}^*$ are a finite sequence of initial conditions and a finite sequence of any actions $\mathcal{A}$, respectively.

Definition 5.9 (Experimental Protocol Rules)**.** To ensure a correct description of protocols using the specification language, we add two rules on the definition of an experimental action:

- There can be only one action of creation among $\{I_0, \mathcal{A}_{create}, \mathcal{A}_{cr_from}\}$ for a given object $\mathcal{O}$.
- All the actions *must have* a synchronisation label denoting the associated step in the protocol progress.
- Two creation or perturbation actions $\mathcal{A}(\mathcal{O}, L, \ldots, c)$ and $\mathcal{A}'(\mathcal{O}, L, \ldots, c)$ on the same object $\mathcal{O}$ can share the same label L; in this case they share the same condition c as they should represent the same step in the protocol.
- Similarly, if multiple perturbation (or creation) actions on the same object $\mathcal{O}$, expressed as *successive* in $\mathcal{P}$, are performed under the *same condition*

$c(t, \mathbf{y})$, they indeed represent multiple effects of the same step in the experiment. Consequently they must have the same synchronization label and cannot affect the same variables, parameters, or inputs.

– Perturbation (or creation) actions and measurement actions *must not* share the same synchronisation labels since measurement actions are just a means to represent a possible observation window on a given object (but they can share the same conditions).

Definition 5.10 (Experimental Protocol Execution Semantics)**.** Let us first consider the case without synchronization. The initialisations are executed first and without conditions, in which the action $\mathcal{A}_{create}(\mathcal{O}, \ldots, c(t))$ is executed under the only condition $c(t) == \texttt{true}$. We abuse the notations here by denoting $\mathcal{A}(\mathcal{O}, c(t, \mathbf{x}^{\mathcal{O}}(t)))$ the actions $\mathcal{A}_x, \mathcal{A}_k, \mathcal{A}_u$, and $\mathcal{A}(\mathcal{O}, \{\mathcal{O}^{(i)}\}, c(t, \mathbf{x}(t)))$ the action $\mathcal{A}_{cr_from}$; and finally $\mathcal{A}_{\mathcal{M}}(\mathcal{O}, c(t, \mathbf{x}^{\mathcal{O}}(t)))$ the measurement action. The action $\mathcal{A}(\mathcal{O}, c(t, \mathbf{x}^{\mathcal{O}}(t)))$ can be executed at t such that:

– $\mathcal{O}$ has been created, all the previous actions $\mathcal{A}(\mathcal{O})$ in the sequence $\mathcal{P}$ have been executed, **and** t is a solution of $c(t, \mathbf{x}^{\mathcal{O}}(t)) == \texttt{true}$.

An action $\mathcal{A}(\mathcal{O}, \{\mathcal{O}^{(i)}\}, c(t, \mathbf{x}(t)))$ can be executed at t such that:

– All the objects $\mathcal{O}^{(i)}$ have been created, and t is a solution of $c(t, \mathbf{x}(t)) == \texttt{true}$, with $\mathbf{x}$ as in Definition 5.5. Thus, the execution of this action is independent of the other actions performed on all the objects in $\{\mathcal{O}^{(i)}\}$ except for the creation one.

An action $\mathcal{A}_{\mathcal{M}}(\mathcal{O}, c(t, \mathbf{x}^{\mathcal{O}}(t)))$ is executed at time t such that:

– $\mathcal{O}$ has been created, and t is a solution of $c(t, \mathbf{x}^{\mathcal{O}}(t)) == \texttt{true}$.

In absence of synchronization labels the execution of a protocol can be interpreted as a set of concurrent processes for each object $\mathcal{O}$, each set of actions $\mathcal{A}(\mathcal{O})$ on $\mathcal{O}$ being a process. The only possible interaction between two or more objects is the creation action $\mathcal{A}(\mathcal{O}, \{\mathcal{O}^{(i)}\}, c(t, \mathbf{x}(t)))$. When using synchronisation labels we must add the following rule:

– All the actions with the same label must be executed at the same time: this implies that the conditions of all these actions must be satisfied simultaneously. If it is not possible, this implies an inconsistency in the protocol definition (see Example 5.11).

This additional rule allows multiple experimental objects to move from one step of the experiment to another in a synchronized way.

5.2 MOEPLA specification to hybrid automata

We propose to generate a hybrid automaton from a protocol specified using the above-described specification language for experimental protocols. This automa-

Example 5.11 (Synchronized actions)**.** A good example of synchronized evolution can be observed in animal studies. When one considers the evolution of a pregnant mother, the perturbations or different evolution stages occur simultaneously for the mother and the foetus. Following the semantic rule for synchronized actions, it is indeed inconsistent to consider transitions for the foetus and the mother as asynchronous. If a delivery is performed on the mother, then, since its environment is modified, the model of the "foetus" must change: the baby is in the foetus state only if the mother is in the pregnant state.
We propose an example of sequence of actions describing the start of a pregnancy process. Let us first consider two experimental objects. We defined $\mathcal{O}_{\text{mother}}$ the experimental object representing the mother and its associated dynamical system $\mathcal{F}_{\text{mother}}$. Similarly, $\mathcal{O}_{\text{baby}}(\mathcal{F}_{\text{baby}})$ which models the baby. We now show how to write the sequence:
First we model the initial condition as a non-pregnant mother by the initialisation $I_0(\mathcal{O}_{\text{mother}}, \dots)$.
Then, we model the start of the pregnancy under the condition c_1 by a creation action for the baby from the mother, for example: $\mathcal{A}_{cr_from}(\mathcal{O}_{\text{baby}}, \mathcal{O}_{\text{mother}}, L_{\text{preg}}, \dots, c_1)$.
If the start of the pregnancy modifies the model of the mother we can add some perturbations actions $\mathcal{A}(\mathcal{O}_{\text{mother}}, L_{\text{preg}}, \dots, c_1)$ which must be synchronous (using the label L_{preg}) to the creation of the baby (in foetus stage).

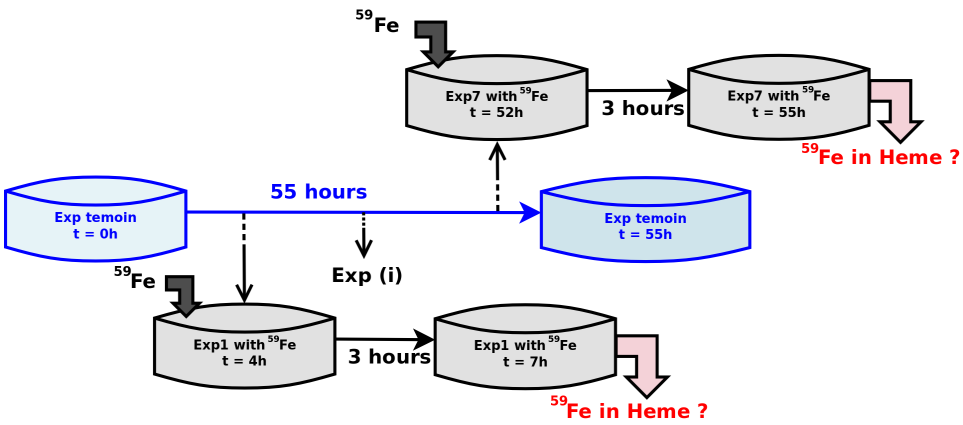

Figure 5.2: Representation of the haemoglobin protocol described in Example 5.12 and later in Section 6.1

ton should accept the trajectories corresponding to the execution of the protocol following our previously defined semantic.

The automaton representing the experimental protocol can be defined by the

Example 5.12 (Haemogolbin production protocol)**.** In Chapter 6 we study more in detail a model of haemoglobin production in erythroblast: a particular kind of cells which differentiates into blood cells inside the bone marrow. In this example we show how the associated experimental procedure [179] can be described in MOEPLA.
The experimental paper [179] proposes a protocol to observe Heme production in erythroblast (among other experimental results). This measure is done through the introduction of a radioactive entity $^{59}\mathrm{Fe}$. The incorporation of $^{59}\mathrm{Fe}$ in Heme on a 3 hours intervals is measured multiple time along the 52 hours of the experiment: this gives an approximation of the Heme production speed at multiple instants of the erythroblast differentiation process (see Figure 5.2)
We can highlight two different mechanisms. First, the mechanism $\mathcal{F}_{rad}$ modelling the cells in presence of $^{59}\mathrm{Fe}$ in the batches where we measure are performed. Second, the mechanism $\mathcal{F}_{ctrl}$ without the radioactive entity associated to the control batch. The ODE systems ($\mathbf{f}_{\text{ctrl}}$) and ($\mathbf{f}_{\text{rad}}$), as well as the associated parameters (see Table 6.6) can be found in Chapter 6 dedicated to the biological case studies. As this protocol implies multiple steps with variation in the dynamics under study, and intermediate measures, its formalisation in MOEPLA is relevant. More details on the analysis of this model can be found in the associated Section 6.1 in Chapter 6.
First we define all the experimental objects, as well as their associated dynamical system, and parameters instantiation:

$$\mathcal{O}(\mathcal{O}_{ctrl}, \mathcal{F}_{ctrl}), \mathcal{O}(\mathcal{O}_{4h}, \mathcal{F}_{rad}), \mathcal{O}(\mathcal{O}_{8h}, \mathcal{F}_{rad}), \mathcal{O}(\mathcal{O}_{16h}, \mathcal{F}_{rad}),$$
$$\mathcal{O}(\mathcal{O}_{24h}, \mathcal{F}_{rad}), \mathcal{O}(\mathcal{O}_{32h}, \mathcal{F}_{rad}), \mathcal{O}(\mathcal{O}_{42h}, \mathcal{F}_{rad}), \mathcal{O}(\mathcal{O}_{52h}, \mathcal{F}_{rad}).$$

Then we explicit the experimental protocol. We note the absence of synchronisation label since this particular example is simple enough.

1. $I_0(\mathcal{O}_{ctrl}, \mathbf{x}_{ctrl}(0) \leftarrow [\mathbf{x}]_0)$
2. $\mathcal{A}_{cr_from}(\mathcal{O}_{4h}, \mathcal{O}_{ctrl}, \mathbf{x}_{rad} \leftarrow \mathcal{R}(\mathbf{x}_{ctrl}), t == 4h)$
3. $\mathcal{A}_{\mathcal{M}}(\mathcal{O}_{4h}, {}^{59}\mathrm{H}(t) + 4{}^{59}\mathrm{Hb}(t), t == 7h)$
4. $\mathcal{A}_{cr_from}(\mathcal{O}_{8h}, \mathcal{O}_{ctrl}, \mathbf{x}_{rad} \leftarrow \mathcal{R}(\mathbf{x}_{ctrl}), t == 8h)$
5. ... (Here it is a compacted version as it is a repeating scheme)

15. $\mathcal{A}_{\mathcal{M}}(\mathcal{O}_{52h}, {}^{59}\mathrm{H}(t) + 4{}^{59}\mathrm{Hb}(t), t == 55h)$

composition of the hybrid automata modelling each concurrent process, i.e. the evolution of each experimental object considered separately.

First we define the hybrid automaton created by the set of experimental actions performed *on a single* object $\mathcal{O}$ with dynamical system $\mathcal{F}_{\mathcal{O}}$. We recall these actions are constituted by exactly one action of creation among $I_0(\mathcal{O})$, $\mathcal{A}_{create}(\mathcal{O},L_{\mathcal{O},0},c_0(t))$, $\mathcal{A}_{cr_from}(\mathcal{O}, L_{\mathcal{O},0}, \ldots, c_0(t,\mathbf{y}))$, a set of perturbation[1] actions $\mathcal{A}(\mathcal{O},L_{\mathcal{O},i},c_i(t,\mathbf{x}^{\mathcal{O}}))$ of different condition c_i and associated label $L_i^{\mathcal{O}}$, $i \geq 0$ ordered following the sequence $\mathcal{P}$, and at least one measurement $\mathcal{A}_{\mathcal{M}}(\mathcal{O}, L_{\mathcal{O},j}, c_j(t,\mathbf{x}^{\mathcal{O}}))$. We recall that successive perturbations actions in $\mathcal{P}$ under the same conditions share the same labels and represent the same step in the protocol progress.

We now proceed to define the hybrid automaton representing the different stages of the evolution of $\mathcal{O}$. Let us denote $\mathcal{H}_{\mathcal{O}} = (\mathcal{I}_{\mathcal{O}}, \mathcal{E}_{\mathcal{O}}, L_{\mathcal{O}}, X_{\mathcal{O}}, K_{\mathcal{O}}, U_{\mathcal{O}}, S_{\mathcal{O}}, R_{\mathcal{O}}, \mathcal{F}_{\mathcal{O}})$ the hybrid automaton associated to the single object $\mathcal{O}$. We remark that in Section 2.2 we introduced the set of labels, guards, reset maps, and activities through the definition of functions associating to these respective sets to each mode or transition. Here we surcharge the notation using $L_{\mathcal{O}}$, $S_{\mathcal{O}}$, $R_{\mathcal{O}}$ and $\mathcal{F}_{\mathcal{O}}$ to denote either the set of possible labels, guards, reset maps, and activities respectively, or the associated labelling functions as in Definition 2.9.

Let us denote $\mathcal{O}_i$, $i \geq 0$, the modes following the creation of $\mathcal{O}$. We also call $\mathcal{O}_\emptyset$ a mode where the object $\mathcal{O}$ does not exist, and $\mathcal{O}_{\texttt{STOP}}$ a mode where measurements are stored in constant variables. The creation action on $\mathcal{O}_i$ determines the initial mode and the initial condition of $\mathcal{H}_{\mathcal{O}}$:

- If the creation action is $I_0(\mathcal{O})$ then the initial mode is $\mathcal{O}_0$ with the parametric dynamical system $\mathcal{F}_{\mathcal{O}}$ as in Definition 5.1. The initial condition of the variables are defined by $I_0(\mathcal{O})$.
- If the creation action is $\mathcal{A}_{create}(\mathcal{O}, L_{\mathcal{O},0}, c_0(t))$ or $\mathcal{A}_{cr_from}(\mathcal{O}, \ldots, L_{\mathcal{O},0}, \ldots, c_0(t,\mathbf{y}))$, the initial mode is the mode $\mathcal{O}_\emptyset$ when $\mathcal{O}$ does not exist, and we define a transition from $\mathcal{O}_\emptyset$ to $\mathcal{O}_0$ with the associated guard $S_{\emptyset,0}$ associated with the condition c_0 and a reset $R_{\emptyset,0}$. We also associate the synchronisation label to the transition $\mathcal{O}_\emptyset \to \mathcal{O}_0$.

Remark 5.13. We note that in the particular case of the action $\mathcal{A}_{cr_from}(\mathcal{O}, \ldots, c_0(t,\mathbf{y}))$ we have a condition and assignment (as seen in Definition 5.4) possibly depending on variables associated to another object. As the definition of $\mathcal{H}_{\mathcal{O}}$ is only an intermediate step to formalize the global hybrid automaton of the whole protocol, we consider at this point that only for the definitions of the guards $S_{\mathcal{O}}$ the variables of the hybrid automaton $\mathcal{H}_{\mathcal{O}}$ is extended from $\mathbf{x}^{\mathcal{O}} \in \mathbb{R}^{n_{\mathcal{O}}}$ to the whole set of variables $y \in \mathbb{R}^n$.

The i-th set of perturbation actions[2] $\mathcal{A}(\mathcal{O}, L_{\mathcal{O},i}, c_i(t,\mathbf{x}^{\mathcal{O}}))$ defines a transition from the mode $\mathcal{O}_{i-1}$ to $\mathcal{O}_i$, with $i \geq 1$. The guard $S_{(i-1),i}$ associated to this

[1] We recall that perturbation actions are the actions $\mathcal{A}_x$, $\mathcal{A}_k$, $\mathcal{A}_u$.

[2] We recall that i is the index of the different stages of the object $\mathcal{O}$.

transition is defined by $c_i(t, \mathbf{x}^{\mathcal{O}})$ and the synchronisation label associated to the transition $(i-1, i)$ is $L_{\mathcal{O},i}$. The continuous dynamics of the mode $\mathcal{O}_i$ is the one of the mode $\mathcal{O}_{i-1}$ with the effects of the action (in case of $\mathcal{A}_k$ or $\mathcal{A}_u$). In case of multiple simultaneous actions as in Remark 5.9, multiple effects are applied if they share the same label and represent the same stage in the protocol progress. In a similar manner, the reset map $R_{(i-1),i}$ is defined by the effect of the action $\mathcal{A}_x$ (if applied), and by default it is the identity map.

The invariants of the modes $\mathcal{O}_\emptyset, \mathcal{O}_0, \mathcal{O}_i$, $i \geq 1$, and $\mathcal{O}_{\texttt{STOP}}$ are defined by the Cartesian product of the state space $X_{\mathcal{O}} \subseteq \mathbb{R}^{n_{\mathcal{O}}}$ of the dynamical system $\mathcal{F}_{\mathcal{O}}$ and the time interval $[0, t_{end}]$, with t_{end} the maximum duration of the protocol as in Definition 5.8.

Finally, we handle each measurement action $\mathcal{A}_{\mathcal{M}}(\mathcal{O}, L_{\mathcal{O},j}, c_j(t, \mathbf{x}^{\mathcal{O}}))$ by creating a transition from all the modes $\mathcal{O}_i$, $i \geq 0$ to a mode $\mathcal{O}_{\texttt{STOP}}$ with frozen dynamics (with derivatives equal to zero), which has for the only purpose of holding the measurement information. The guards of these transitions are defined by the condition $c_j(t, \mathbf{x}^{\mathcal{O}})$.

Remark 5.14. We highlight the particular case of the measurement actions: the $\mathcal{O}_{\texttt{STOP}}$ mode defined here is only used as a way to integrate the measurement in the hybrid automaton. In practice, the measurement actions need to be handled case by case for each tool selected for the analysis as we will see in the Section 6.1.2 of Chapter 6.

Methodology 5.16 (Hybrid automaton generation)**.** The hybrid automaton $\mathcal{H}$ of the experimental protocol using a set of objects $\{\mathcal{O}^{(i)}\}$,with $0 \leq i \leq p$ the number of experimental objects, can be generated, using Definition 2.11, by the synchronous parallel compositions of each hybrid automata $\mathcal{H}_{\mathcal{O}^{(i)}}$ defined for each object $\mathcal{O}^{(i)}$ as described above.

After each composition, we delete all the resulting transitions leaving a composite mode defined by at least one mode $\mathcal{O}_{\texttt{STOP}}$. This last step ensures that experimental measurements are separated from the actual mechanism in the protocol and do not perturb the simulation.

We refer to the Examples 5.15 and 5.17 for two examples of the application of Methodology 5.16.

5.3 Discussion and future work

In its current state the specification language MOEPLA allows describing the evolution of a mechanistic model during a multi-stage experiment. It is expressive enough to represent a large panel of possible experiments either coming from the

Example 5.15 (Hybrid automaton of protocol)**.** Given the following experimental objects with their associated dynamical systems: $\mathcal{O}(\mathcal{F}_{\mathcal{O}})$, $\mathcal{O}'(\mathcal{F}_{\mathcal{O}'})$
We propose an example of experimental protocol, in a simplified form:

1. $I_0(\mathcal{O}, \mathbf{x}^{\mathcal{O}}(0) = \mathbf{x}_0^{\mathcal{O}})$: We start with $\mathcal{O}$.
2. $\mathcal{A}_k(\mathcal{O}, L_1, K^{\mathcal{O}} \leftarrow K_{\text{new}}, c_1(t, \mathbf{x}^{\mathcal{O}}))$: Object $\mathcal{O}$ may be perturbed and its parameter space changed.
3. $\mathcal{A}_{cr_from}(\mathcal{O}', \mathcal{O}, L_2, \mathbf{x}^{\mathcal{O}'} \leftarrow R(\mathbf{x}^{\mathcal{O}}), c_2(t, \mathbf{x}^{\mathcal{O}}))$: Object $\mathcal{O}'$ is created from $\mathcal{O}$.
4. $\mathcal{A}_{\mathcal{M}}(\mathcal{O}', L_3, x_{meas} \leftarrow R(\mathbf{x}^{\mathcal{O}'}), c_3(t, \mathbf{x}^{\mathcal{O}'}))$: A measure on $\mathcal{O}'$ of the variables $\mathbf{x}^{\mathcal{O}'}$ is stored in x_{meas}.

The two hybrid automata corresponding to the evolution of each object $\mathcal{O}$ and $\mathcal{O}'$ are represented in Figure 5.3. The hybrid automaton of above protocol and resulting from the composition of the two automata of the Figure 5.3, can be found in Figure 5.4. In the Figure 5.4 we assume there is no synchronization. If $L_1 = L_2$ then only the transition with the condition $c_1 \wedge c_2$ would remain.

fields of molecular biology, or biochemical reactions, or medicine and therapy modelling.

Thanks to the formalization of a MOEPLA protocol as a hybrid automaton, we can provide a formal representation of all the possible executions of a protocol and its associated mechanisms. Indeed, hybrid automata allow non-determinism and through the synchronous parallel composition of automata we can represent the evolution of multiple processes involved a given protocol, and their interactions. This non-determinism represents all the possible executions of a protocol assuming a model for each experimental object mechanism. Then, depending on the sparsity of the knowledge about the mechanisms or the uncertainty (or ambiguities) in the protocol definitions, it is, for example, possible to search for bad behaviours and assert properties for critical procedures (as in therapy).

However, the current work still has multiple flaws that need to be addressed: they are related either to the hybrid system representation or to the specification of the protocol itself. The first list of comments concerns the hybrid automaton representation. Hybrid automata are very useful to represent non-deterministic systems. However, they are lacking on multiple points in the formalism that we use. In the currently used formalism for hybrid dynamical systems, we do not consider urgent transitions as they are not handled by most of the formal verification software such as SpaceEx [87]. Moreover, as seen in Definition 2.9 in Chapter 2, all the modes of a hybrid automaton have the same dimension, which in our case leads to a huge

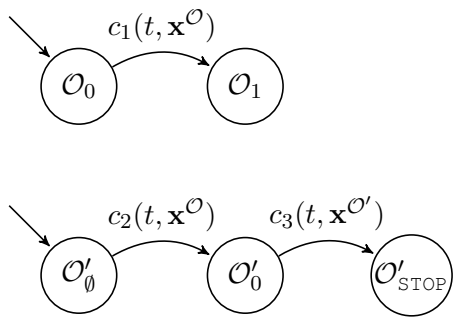

Figure 5.3: The hybrid automata of the objects defined in the protocol in Example 5.15.

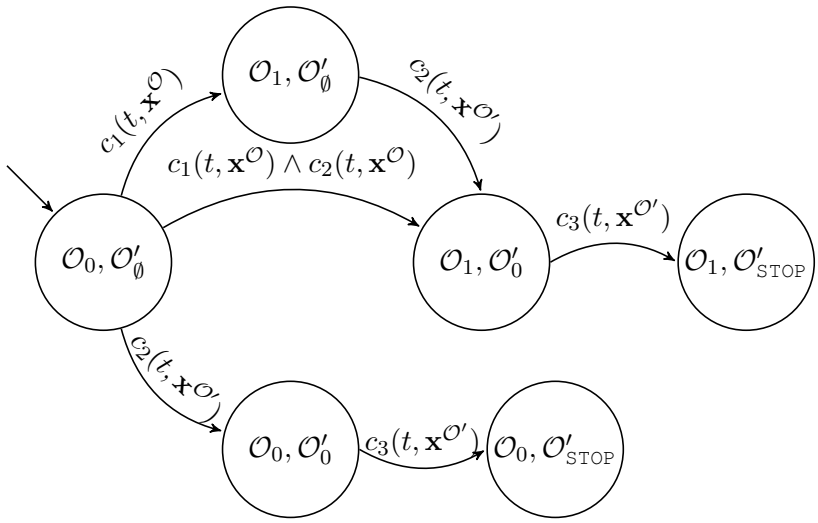

Figure 5.4: Parallel composition of the hybrid automata of the protocol in Example 5.15.

dimension increases. For example, in the hybrid automaton from Example 5.17 each mode has 46 state variables, while in practice when addressing this model using the Controlled Hybrid System[3] in Section 6.1.2, we have to consider at most 9 variables (including a clock variable). Similarly, the protocol of Example 5.17 could be modelled by a 2 modes hybrid system instead of the one we show in Figure 5.5 using in Methodology 5.16 to generate an hybrid automata. While our method avoids producing loops in the hybrid automaton representation and do not allow Zeno behaviours, it results in large hybrid systems. We note that one way to efficiently handle the exponential cost of the parallel composition would be to perform it on the fly during the analysis, as done in the reachability toolbox SpaceEx [87].

The second list of comments concern the expressiveness of the specification language itself. A first comment is that we assume that the different objects share

[3]See Definition 3.2

Example 5.17 (Hybrid automaton for Haemoglobin production)**.** In Figure 5.5, we show a shortened version of the hybrid automaton generated by the composition of all object automata as defined in the Methodology 5.16. We note that in each mode, there are 8 dynamical systems running in parallel. This method for hybrid automaton generation leads to a theoretically huge number of variables. In Section 6.1.2, we will work on an adapted and simplified version of Figure 5.5 automaton using the Controlled Hybrid System formalism defined in Chapter 3.

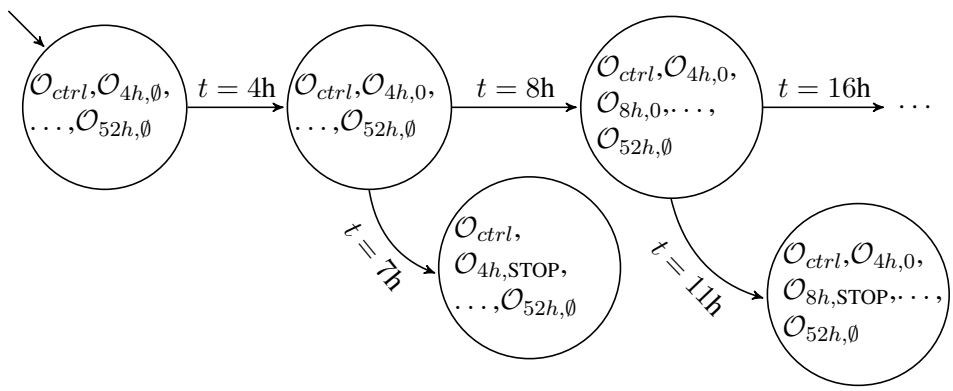

Figure 5.5: Hybrid automata (shortened) resulting from the application of Methodology 5.16 to the protocol of Example 5.12. We do not show the transitions resulting from the composition whose conditions can never be satisfied (as we have fixed time condition).

no common variables, parameters, or input, at the exception of the possible creation action $\mathcal{A}_{cr_from}$. This assumption was done to simplify the semantics in a first version of the language. A way to get around this current limitation is to define a single object modelling all the interaction or coupled variables. However, this limits the expressiveness of the language as we cannot define separate objects which share variable for only a few stages. Therefore, it is necessary to update the language (and the definition of the synchronous parallel composition) to handle the definition of common variables (parameters, or inputs) on a given set of actions, or stages.

Another comment is that time is currently a global variable separated from the other state variables. Such global time only allows us to easily define conditions on the absolute time, for example $t \geq 4$ hours. However, conditions on events or relative to a previous event are hard to express. To easily express them we need to introduce clock variables. Such variables can currently be introduced as state variables (see Section 6.1.2) but are not directly apparent in the protocol syntax. In addition, the actions are performed instantaneously and it should be an interesting addition to consider delays in the actions. In the current language this can be done

by defining a time (or clock) interval on the condition of each action: for example the first action is performed for $t \in [1\text{h}, 2\text{h}]$ and the second for $t \in [1.5\text{h}, 2.5\text{h}]$ means there is a delay of at most 1h on both actions. However this leads to spurious sequences of multiple instantaneous transitions in the hybrid automaton representation, without having the global time elapsing. Indeed, in the previous example, both actions can still occur without letting the global time t elapse: for all the instants $t \in [1.5\text{h}, 2\text{h}]$ both transitions can be taken successively.

Finally, the syntax of the language itself may change to facilitate the interactions between the biologist and the modelling team. However, the aim remains that the language must have a succinct definition with few rules while allowing its possible executions, defined by the language semantics, to be represented by a hybrid dynamical system.

CHAPTER 6

Applications and case studies

6.1 Erythropoiesis haemoglobin production model 99
6.1.1 Parameter study 100
6.1.2 Model revision 107
6.2 Mammalian cellular iron homoeostasis (MCIH) model . . . 115
6.2.1 Context 115
6.2.2 Set-based analysis 117
6.3 Cadmium impact on glucose response model 122

6.1 Erythropoiesis haemoglobin production model

In this first section we study a model of haemoglobin production during the differentiation of erythroblasts into erythrocytes. Erythrocytes (also named red blood cells) are produced inside the bone marrow. In this place, they go through multiple differentiation stages from stem cells (also called hemocytoblasts in this context) into erythroblasts and finally erythrocyte. This differentiation process is also called erythropoiesis.

During its differentiation, an erythroblast produces haemoglobin. At the final stages, the erythroblast forces out its nucleus and is released in the circulating blood. The haemoglobin stored in the erythrocyte will play the role of oxygen transport protein. Without entering into details, the haemoglobin is constituted from 8 sub-components: 4 hemes and 4 globins[1].

In this work, we model the haemoglobin production during the final stages of an erythroblast differentiation into erythrocyte. The proposed model is given by the

[1]Here we do not consider the multiple variations of haemoglobin, and globin.

ODEs system ($\mathbf{f}_{\text{ctrl}}$) which describes the dynamics of intracellular concentrations of: iron Fe, heme H, globin G and haemoglobin Hb.

$$\begin{aligned}
\frac{d\mathrm{Fe}}{dt} &= k_1\mathrm{Fe}_{ex} - k_2\mathrm{Fe} - k_3\mathrm{Fe} \\
\frac{d\mathrm{H}}{dt} &= k_3\mathrm{Fe} - k_4\mathrm{H} - 4k_5\mathrm{H\,G} \\
\frac{d\mathrm{G}}{dt} &= k_6\mathrm{H} - 4k_5\mathrm{H\,G} - k_7\mathrm{G} \\
\frac{d\mathrm{Hb}}{dt} &= k_5\mathrm{H\,G} - k_8\mathrm{Hb}
\end{aligned} \tag{$\mathbf{f}_{\text{ctrl}}$}$$

In the first equation of ($\mathbf{f}_{\text{ctrl}}$), the term $k_1\mathrm{Fe}_{ex}$ describes iron influx from the extracellular space, while the second term $k_2\mathrm{Fe}$ and third term $k_3\mathrm{Fe}$ respectively describe the iron efflux from the cell and its consumption for the production of heme. In the second equation, the terms $k_3\mathrm{Fe}$ and $k_4\mathrm{H}$ respectively describe the heme production and heme degradation and transport from the cell. The term $4k_5\mathrm{H\,G}$ models the heme and globin consumption for the production of haemoglobin in the second and third equations. Note that the factor 4 describes the need of 4 hemes for the production of 1 haemoglobin. In the third equation, globin synthesis is accelerated by heme, hence the production term depends on H (first term $k_6\mathrm{H}$). The heme is not consumed so this term does not appear in the second equation. The globin degradation term is $k_7\mathrm{G}$. Finally, in the last equation, the two terms are haemoglobin production $k_5\mathrm{H\,G}$ and degradation $k_8\mathrm{Hb}$.

In Section 6.1.1, we describe how the data obtained from mouse erythroid cell cultures are exploited to estimate the value of parameters occurring in ($\mathbf{f}_{\text{ctrl}}$). Then, in Section 6.1.2, we use this model as a proof of concept of the model revision method previously described in Chapter 3.

6.1.1 Parameter study

In modelling complex biological phenomena, the identification of parameter values is always a critical problem due to the scarcity of kinetic data. Here the situation is rather favourable. We have been able to use four datasets [180, 181, 179, 182] to determine the values of the 8 kinetic parameters contained in the model of haemoglobin production presented in ($\mathbf{f}_{\text{ctrl}}$).

These experiments were performed on cultures of erythroid cells from the spleens of mice during their differentiation. The general scheme is to add radiolabels (^{59}Fe or ^{3}H-leucine[2]) at different time points, continue the culture for a given time duration and then measure the quantity of ^{59}Fe incorporated in heme or ^{3}H-leucine incorporated in β-major globin.

In order to exploit this kind of experiment we use two extended systems of equations: the first contains additional equations for the evolution of ^{59}Fe-containing

[2]Note that the symbol H in ^{3}H-leucine stands for the isotope of hydrogen.

species (intracellular $^{59}\mathrm{Fe}$ itself and radioactive heme and haemoglobin); the second contains additional equations for the evolution of $^{3}\mathrm{H}$-leucine-containing globin and haemoglobin. Consequently, each specific type of experiment has to be simulated by the corresponding system of differential equations respectively ($\mathbf{f}_{\mathrm{rad}}$) for datasets #1 and #2 (see Tables 6.1 and 6.2), and (6.1) for dataset #3 (see Table 6.3). The system of equations representing the haemoglobin production in presence of $^{59}\mathrm{Fe}$ is given in ($\mathbf{f}_{\mathrm{rad}}$). We note that the dynamics of G and Hb are optional if one only observes the radioactive species.

$$\begin{aligned}
\frac{d\mathrm{Fe}}{dt} &= k_1\mathrm{Fe}_{ex} - k_2\mathrm{Fe} - k_3\mathrm{Fe} \\
\frac{d\mathrm{H}}{dt} &= k_3\mathrm{Fe} - k_4\mathrm{H} - 4k_5\mathrm{H}\,\mathrm{G} \\
\frac{d\mathrm{G}}{dt} &= k_6\mathrm{H} - 4k_5\mathrm{H}\,\mathrm{G} - k_7\mathrm{G} \\
\frac{d\mathrm{Hb}}{dt} &= k_5\mathrm{H}\,\mathrm{G} - k_8\mathrm{Hb} \\
\frac{d^{59}\mathrm{Fe}}{dt} &= k_1^{59}\mathrm{Fe}_{ex} - k_2^{59}\mathrm{Fe} - k_3^{59}\mathrm{Fe} \\
\frac{d^{59}\mathrm{H}}{dt} &= k_3^{59}\mathrm{Fe} - k_4^{59}\mathrm{H} - 4k_5^{59}\mathrm{H}\,\mathrm{G}_{tot} \\
\frac{d\mathrm{G}_{tot}}{dt} &= k_6(\mathrm{H} +^{59}\mathrm{H}) - 4k_5(\mathrm{H} +^{59}\mathrm{H})\,\mathrm{G}_{tot} - k_7\mathrm{G}_{tot} \\
\frac{d^{59}\mathrm{Hb}}{dt} &= k_5^{59}\mathrm{H}\,\mathrm{G}_{tot} - k_8^{59}\mathrm{Hb}
\end{aligned} \tag{$\mathbf{f}_{\mathrm{rad}}$}$$

We now provide the system of equation corresponding to the experimental protocol of dataset #3: we note G^r the $^{3}\mathrm{H}$-leucine-containing globin.

$$\begin{aligned}
\frac{d\mathrm{Fe}}{dt} &= k_1\mathrm{Fe}_{ex} - k_2\mathrm{Fe} - k_3\mathrm{Fe} \\
\frac{d\mathrm{H}_{tot}}{dt} &= k_3\mathrm{Fe} - k_4\mathrm{H} - 4k_5\mathrm{H}\,(\mathrm{G} + \mathrm{G}^r) \\
\frac{d\mathrm{G}}{dt} &= k_6\mathrm{H} - 4k_5\mathrm{H}\,\mathrm{G} - k_7\mathrm{G} \\
\frac{d\mathrm{Hb}}{dt} &= k_5\mathrm{H}\,\mathrm{G} - k_8\mathrm{Hb} \\
\frac{d\mathrm{G}^r}{dt} &= k_6\mathrm{H} - 4k_5\mathrm{H}\,\mathrm{G}^r - k_7\mathrm{G}^r \\
\frac{d\mathrm{Hb}^r}{dt} &= k_5\mathrm{H}\,\mathrm{G}^r - k_8\mathrm{Hb}^r
\end{aligned} \tag{6.1}$$

The experimental results from Tables 6.1, 6.2, and 6.3 are given in radioactivity units (cpm standing for counts per minute) since the experiments measure the activities of the radio-labels $^{59}\mathrm{Fe}$ and $^{3}\mathrm{H}$. However, these data are incomplete as we lack the actual conversion factor from the radioactivity measures in cpm to the

actual molecule quantity in mol. As we have mixed information, concentrations on one part and radioactivity measures on the other, we choose to work with normalized data points, by choosing a point in each dataset as a reference point, and dividing the other points by this value. The chosen reference data point is the one providing the smallest residual value (6.2) after scaling. However, we add a few constraints to ensure the simulations results stay in viable biological ranges.

The dataset [181, Table 2] (not reproduced here) allows us to estimate k_1 the iron intake rate into the cell. We make the approximation that the flux of iron intake is constant. With this k_1 is found equal to 250 atoms/fL/min. In the conditions of this experiment the concentration of ^{59}Fe:Tf in the culture medium is 200 μg/mL. We take the molecular mass of glycosylated, iron-saturated Tf (diferric) as being equal to 80 kDa, and we obtain $k_1 = 1.4\text{e}^{-3}$ s^{-1}.

Exposure time (h)	4	8	16	24	32	42	52
^{59}Fe in Heme (cpm/1e^{-7}L)	47	213	697	1020	1725	2379	2370

Table 6.1: Dataset #1: Accumulated ^{59}Fe in Heme. The radiolabel (transferrin-bound ^{59}Fe) is added to the medium at t = 0h. The first line indicate the time at which cells are collected.

Exposure time (h)	4-7	8-11	16-19	24-27	32-35	42-45	52-55
^{59}Fe in Heme (cpm/1e^{-7}L/h)	16	85	348	391	399	481	395

Table 6.2: Dataset #2: The cells are first cultured in presence of iron (normal, non-radioactive, isotope). Then the radiolabel (transferrin-bound ^{59}Fe) is added directly to the medium, and the culture continues for the period indicated in the first line (duration = 3h).

In addition to these datasets we consider a set of constraints to keep the results biologically viable.

- The quantity of ^{59}Fe incorporated in free heme at t=52h is at most 5% of the ^{59}Fe incorporated in all forms of heme (free heme and heme in haemoglobin).

- The quantity of ^{59}Fe incorporated in free heme at t=52h is at least 0.002% of the ^{59}Fe incorporated in all forms of heme (free heme and heme in haemoglobin).

- The concentration of haemoglobin at t=52h is at least 100 molecules/fL.

- The concentration of intracellular labile iron is less than 10 times the concentration of haemoglobin (all taken at t=52h).

Exposure time (h)	0-12	12-24	24-36	36-48
^{3}H-leu in β-major globin (cpm/1e^{-7}L/h)	103	255	771	942

Table 6.3: Dataset #3: Incorporation of ^{3}H-leucine in β-major globin. The cells are first cultured in a medium without radiolabel. Then the radiolabel (^{3}H-leucine) is added directly to the medium, and the culture continues for the period indicated in the first line (duration = 12h).

For the 7 kinetic parameters other than k_1 and the initial concentration in intracellular iron Fe_0, we have to resort to a systematic search. It is known from [183] that the IRP concentration decreases down as differentiation proceeds. Because IRP activity represses the biosynthesis of protoporphyrin IX, which is inversely related to internal Fe concentration, the rate of heme production increases with time. To take this into account we consider that parameter k_3 is time dependent. In this first study we consider the following polynomial function: $k_3(t) = a_{k3} + (b_{k3}t)^4$. With this function, the increase is slow during the first hours and becomes steeper in the later stages of differentiation. A linear time dependence was tried, but it did not provide a good adjustment to the experimental data. In Section 6.1.2 in the following, we propose to revise this parameter as a proof of concept of our method described in Chapter 3. Therefore, we consider in total 8 parameters related to kinetics. As we start this modelling without any prior knowledge on the parameter ranges (except physiological bounds), our initial parameter space where the search is performed is given in Table 6.4: for each unknown quantity we define a search interval (spanning several orders of magnitudes). In Table 6.4 we also provide the set of intracellular iron initial conditions Fe_0 we consider in our search.

Variables	k_2	a_{k3}	b_{k3}
Search intervals	$[1e^{-11}, 1e^{-02}]$	$[1e^{-10}, 1e^{-01}]$	$[1e^{-10}, 1e^{-04}]$
Variables	k_4	k_5	k_6
Search intervals	$[1e^{-06}, 1e^{+01}]$	$[1e^{-11}, 1e^{-02}]$	$[1e^{-07}, 1e^{+01}]$
Variables	k_7	k_8	Fe_0
Search intervals	$[1e^{-09}, 1e^{-01}]$	$[1e^{-11}, 1e^{-01}]$	$[6e^{+03}, 6e^{+06}]$

Table 6.4: This table provides the intervals considered for the search for parameters sets and initial conditions satisfying the experimental results of Tables $\mathbf{f}_{\mathrm{ctrl}}$,$\mathbf{f}_{\mathrm{rad}}$ and 6.1.

In this work we searched for parameters sets satisfying three experimental datasets. These datasets were obtained through three different experimental protocols and we simulate them accordingly with three different models. At the time of this work, we did not consider the formalization of experimental protocol as a hybrid system: we argue that this formalization would have eased our analysis.

Parameters	Enclosure	Units
k_2	$[1.0e^{-10}, 5.81e^{-9}]$	s^{-1}
a_{k3}	$[3.43e^{-9}, 1.47e^{-8}]$	s^{-1}
b_{k3}	$[6.89e^{-7}, 7.36e^{-7}]$	$s^{-\frac{5}{4}}$
k_4	$[3.82e^{-4}, 5.16e^{-4}]$	s^{-1}
k_5	$[2.87e^{-6}, 1.01e^{-5}]$	fL.molecules^{-1}.s^{-1}
k_6	$[3.94e^{-4}, 5.68e^{-4}]$	s^{-1}
k_7	$[2.48e^{-10}, 5.08e^{-7}]$	s^{-1}
k_8	$[1.01e^{-5}, 1.49e^{-5}]$	s^{-1}
Fe_0	$[2.99e^{+2}, 6.07e^{+2}]$	atoms/fL

Table 6.5: Results of the search procedure. The second column defines the box enclosing the set of valid points in parameter space. The following columns contains the values for 4 particular solutions. Fe_0 is the initial concentration of internal iron (not labelled). The concentration of external labelled iron ($^{59}Fe_{ex}$) is 3000atoms/fL, while the concentration of external iron Fe_{ex} (not labelled) is 4atoms/fL.

Parameters	p_1	p_2	p_3	p_{mean}
k_2	$1.61e^{-10}$	$1.22e^{-10}$	$6.0e^{-10}$	$3.78e^{-10}$
a_{k3}	$1.42e^{-8}$	$7.70e^{-9}$	$9.25e^{-9}$	$7.4e^{-9}$
b_{k3}	$7.2e^{-7}$	$7.13e^{-7}$	$7.15e^{-7}$	$7.2e^{-7}$
k_4	$3.95e^{-4}$	$4.63e^{-4}$	$4.30e^{-4}$	$4.47e^{-4}$
k_5	$5.59e^{-6}$	$6.48e^{-6}$	$5.71e^{-6}$	$7.27e^{-6}$
k_6	$4.74e^{-4}$	$5.10e^{-4}$	$4.31e^{-4}$	$4.47e^{-4}$
k_7	$4.46e^{-8}$	$2.88e^{-10}$	$3.01e^{-7}$	$4.97e^{-10}$
k_8	$1.3e^{-5}$	$1.243e^{-5}$	$1.17e^{-5}$	$1.14e^{-5}$
Fe_0	$3.10e^{+2}$	$3.46e^{+2}$	$3.27e^{+2}$	$3.21e^{+2}$
ε_1	0.220	0.235	0.239	0.238
ε_2	0.239	0.228	0.238	0.238
ε_3	0.225	0.238	0.199	0.200
$\varepsilon_{\text{mean}}$	0.228	0.234	0.226	0.226

Table 6.6: Results of the search procedure. Columns 2, 3 and 4 provide solutions corresponding to the minimum (within the set of found solutions) of ε_1, ε_2, and ε_3, respectively. The following column is the one with the lowest average residual $\varepsilon_{\text{mean}} = (\varepsilon_1+\varepsilon_2+\varepsilon_3)/3$.

However, as these protocols are simple enough, we can still simulate them as sequences[3] of classical ODE simulations using either the ODE system modelling the cell without radiolabels ($\mathbf{f}_{\text{ctrl}}$), the ones modelling the cell in presence of ^{59}Fe ($\mathbf{f}_{\text{rad}}$) or ^{3}H (6.1).

[3]The simulations we perform are conform to the experimental protocol of each particular dataset. In Section 6.1.2, we use the hybrid systems formalization to improve the analysis.

The Cartesian product of the parameter intervals (see Table 6.4) defines a multidimensional search box. To perform simulations and search for valid parameter sets, we draw a large number of random points in that box, and for each parameter point we perform simulations and compare accordingly with each experimental dataset. In order to sample equally all the orders of magnitude, the sampling is done on a logarithmic scale. We use the quasi-Monte Carlo method [184] with the Sobol sequence. For each dataset we quantify the agreement between simulated $\mathbf{y}$ and experimental $\mathbf{z}$ measured quantities by computing the following residual:

$$\varepsilon_q = \frac{\sum_i |\mathbf{y}_i - \mathbf{z}_i|}{\sum_i \mathbf{z}_i} , \tag{6.2}$$

where $q \in \{1, 2, 3\}$ denotes one of the three studied datasets and the index i identifies the measurements belonging to that dataset #i. For the experiment associated to the dataset of Table 6.1, we have $\mathbf{y} = {}^{59}\mathrm{H} + 4\,{}^{59}\mathrm{Hb}$. In the context of Table 6.2, $\mathbf{y} = ({}^{59}\mathrm{H} + 4\,{}^{59}\mathrm{Hb})/3$. Finally, the measurement observed in the dataset of Table 6.3 is given by $\mathbf{y} = (0.4\mathrm{G}^r + 1.6\mathrm{Hb}^r)/12$.

Each dataset q is associated with a threshold M_q, and we keep as potential solutions all parameter points which satisfy $\varepsilon_q \leq M_q$, where M_q. Since we have 3 datasets, we have 3 constraints of this type. It is, in general, not possible to minimize all 3 residuals simultaneously. In order to retain only physiologically relevant solutions we also add constraints bearing on the concentrations reached at the end of the differentiation process using the constraints we previously defined. This simple parameter search method is summarized in Algorithm 5. We note that we can iterate over Algorithm 5 by constructing new smaller parameter spaces over each previous results of Algorithm 5. Since there is no unique way to combine several experimental datasets and additional constraints, we present the results as a set of solutions (a cloud of points in parameter space). The 3 thresholds M_q are chosen equal to 0.25 as we want to equally verify all datasets. The box enclosing the cloud of valid points is given in Table 6.5. These results are given after a search over 35 millions samples (for around 48h of computation). This gives an idea of the spread of the set of solutions. It can be seen that most parameters are rather well defined, except k_2 and k_7. A scatter plot of the cloud of valid points on the plane $k_4 - k_5$ is displayed in Figure 6.1 in order to visualize the shape of the cloud on that projection. Table 6.6 shows 4 solutions we have selected in this set: the solutions p_1, p_2, p_3 and p_{mean} which minimizes ε_1, ε_2, ε_3 or $\varepsilon_{\text{mean}} = (\varepsilon_1 + \varepsilon_2 + \varepsilon_3)/3$, respectively. Figure 6.2 displays the evolution of the system with the p_{mean} values. This last solution is the one which has the lowest average residual $\varepsilon_{\text{mean}}$ within the solution set. It can be observed that internal iron concentration goes through a maximum and then decreases to low values. Haemoglobin concentration increases steadily to a value of about 10^2 molecules/fL. The agreement between the measurements and the corresponding quantities derived from the simulations is displayed in Figures 6.3, 6.4, and 6.5 for datasets #1, #2 and #3, respectively. In each figure the results obtained with parameter points p_1, p_2, p_3, and p_{mean} are shown.

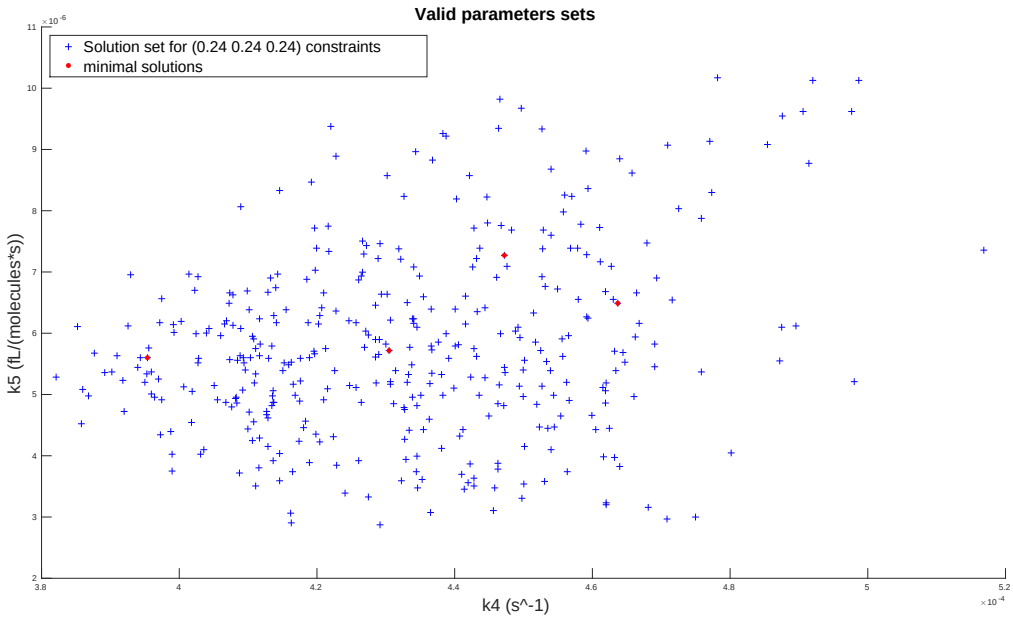

Figure 6.1: Projection of the cloud of valid points on the plane $k_4 - k_5$. The solutions p_1, p_2, p_3, p_{mean} are indicated in red.

Algorithm 5 Simple parameter search(K,M,E,n_{samples})

INPUT: K parameter space considered for search box.
INPUT: $M = (M_1, M_2, M_3)$, threshold considered for the experiment q.
INPUT: $E = \{E_1, E_2, E_3\}$, 3 piecewise ODE systems modelling the experiments associated to datasets #1, #2 and #3, respectively.
INPUT: n_{samples}, the number of samples draw in the parameter space K.
OUTPUT: r, set of samples which satisfy all the error thresholds M_q.

$\mathcal{S} = \texttt{sobol}(K, n_{\text{samples}})$
$r = \emptyset$
for $\mathbf{k} \in \mathcal{S}$ **do**
    for $q \in \{1, 2, 3\}$ **do**
        $\mathbf{y}^{[q]} = \texttt{simulate}(E_q, \mathbf{k})$
        $\varepsilon_q = \texttt{evaluate}(\mathbf{y}^{[q]}, \mathbf{z}^{[q]})$
    end for
    if $\forall q \in \{1, 2, 3\}$, $\varepsilon_q \leq M_q$ **then**
        $r = r \cup \mathbf{k}$
    end if
end for
return r

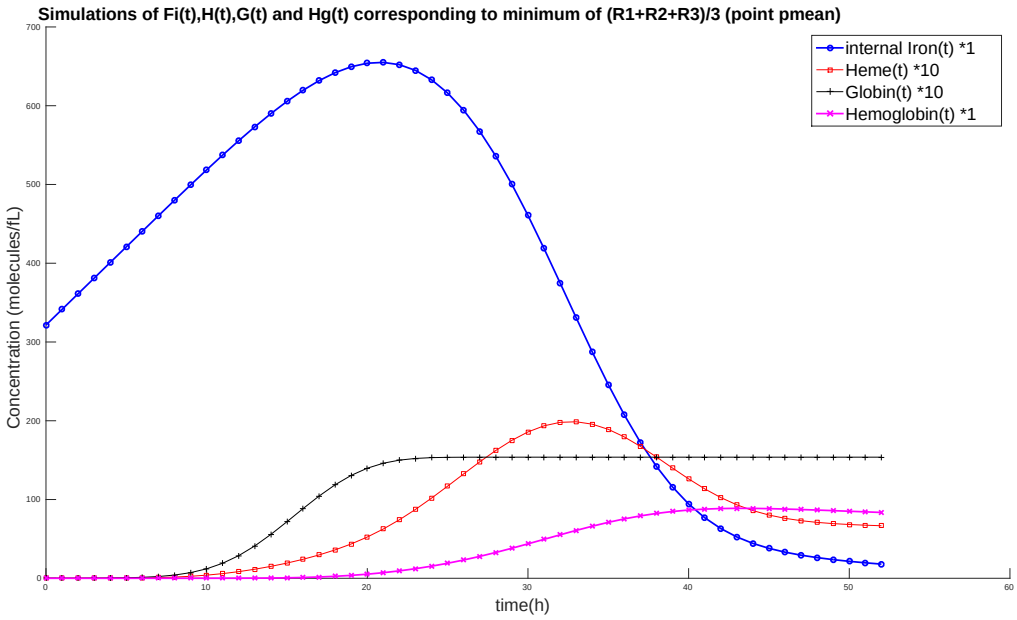

Figure 6.2: Evolution of the 4 concentrations computed with the parameter values identified as p_{mean} in Table 6.6.

6.1.2 Model revision

Model adjustment. In the previous section, we addressed the problem of refining the parameters space and fitting multiple datasets on the haemoglobin production model. In this section, we apply the model revision method proposed in Chapter 3 on the haemoglobin production model ($\mathbf{f}_{\text{rad}}$) for the protocol [179] corresponding to the dataset #2 in Table 6.2. Using the method defined in Chapter 3, we search for another modelling of the time varying parameter $k_3(t)$, which better fit the dataset #2, while being biologically interpretable. While in the previous section the experimental protocol was not explicitly formalized as a hybrid system, we will consider in this section a formulation close to the one proposed in the Chapter 3. We use the values given in the column p_{mean} from the Table 6.6 for any parameters other than k_3 .

We recall that the ODEs ($\mathbf{f}_{\text{ctrl}}$) model the evolution of the haemoglobin production in the differentiating erythrocyte cells situated in the bone marrow. In this section, to stay close to the notations of Chapter 3 we note x_1 to x_4 the variables that represent respectively the internal iron in the cell Fe, the heme H , the globin G, and the haemoglobin Hb. The hybrid dynamical system $\mathcal{H}$ models an experimental protocol designed to measure the integration of iron inside heme (H) at several steps of the cell differentiation. For example, we recall that the data point at time $t = 7$ hours in Table 6.9, is obtained through the following procedure: we

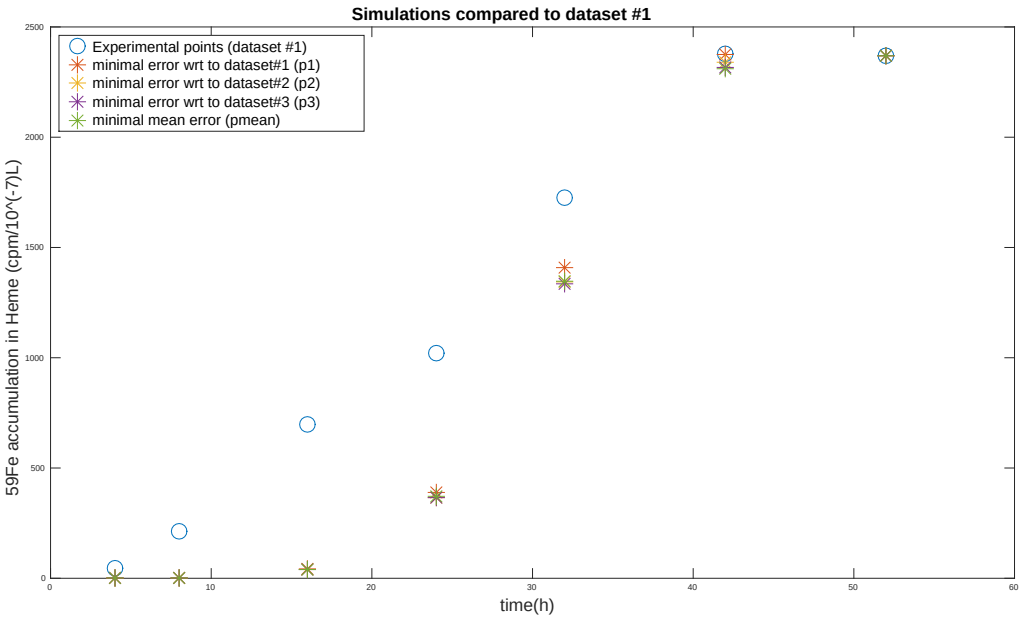

Figure 6.3: Visualization of the match between the measurements and the corresponding quantities computed from the simulations, for dataset #1. Measurements are represented by circles and computed quantities by asterisks. The 3 parameter points p_1, p_2, p_3, and p_{mean} defined above are shown.

first start with a control batch of cells, then at time $t = 4$ hours after the start of the experiment, the culture medium is perturbed with an injection of measurable radioactive iron $^{59}\mathrm{Fe}$ for a subset of the cells. This perturbation implies the new ODEs ($\mathbf{f}_{\mathrm{rad}}$) modelling the evolution of two interdependent models[4] : the model of non-radioactive haemoglobin production and the model of haemoglobin production with radioactive species. Three hours after the perturbation with radioactive iron, the total radioactive heme is measured, meaning the heme free in the cell and the one in the radioactive haemoglobin. This measurement is given by the formulas $^{59}\mathrm{H} + 4\,^{59}\mathrm{Hb}$.

This hybrid dynamical system $\mathcal{H}$ is close to the one proposed in Chapter 5, and differs from it on two points: the lack of `Stop` modes and a number of variables that changes between two modes. Indeed, in this work we do not model the measurement actions as they are represented in the cost function of the optimization problem.

We recall in Table 6.9 the observed radioactivity divided by three hours. Finally, these measurements provide results on the variation during the cell differentiation of the integration of iron in heme, which is associated to the parameter k_3.

[4]For this reason we kept G and Hb present in the ODE system ($\mathbf{f}_{\mathrm{rad}}$)

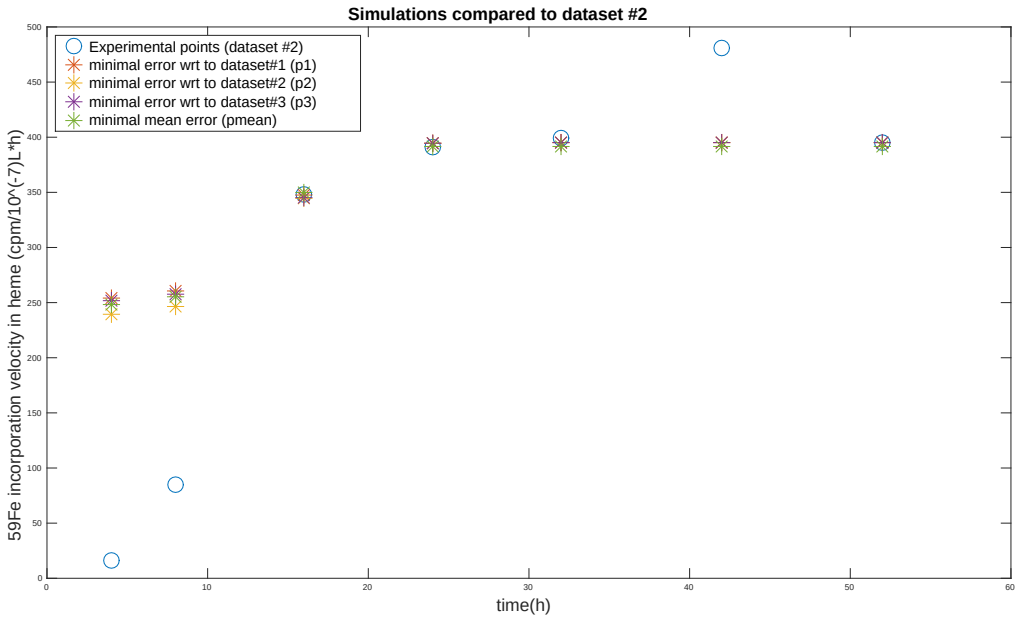

Figure 6.4: Same as previous for dataset #2.

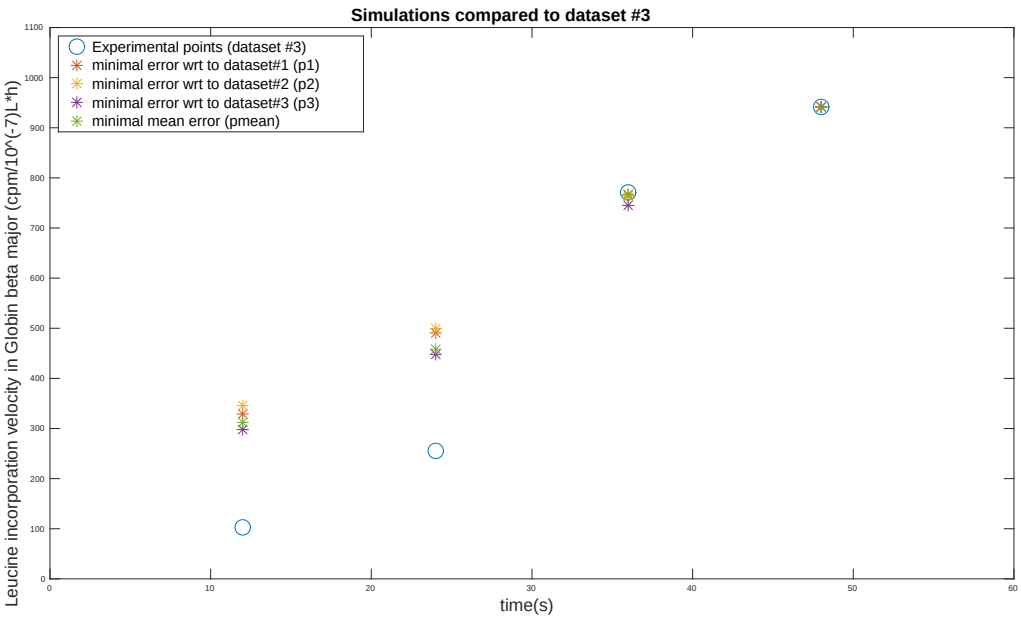

Figure 6.5: Same as previous for dataset #3.

Mode	n_i	$\mathbf{f}_i(t, \mathbf{x}, \mathbf{u})$	$\mathcal{X}_i$	U_i
$i = 1$	5	$\mathbf{f}_{ctrl}(t, \mathbf{x}, u)$	$[0, 4] \times [0, 1]^4$	$[0, 1]$
$i = 2$	9	$\mathbf{f}_{rad}(t, \mathbf{x}, u)$	$[4, 7] \times [0, 1]^8$	$[0, 1]$
$i = 3$	5	$\mathbf{f}_{ctrl}(t, \mathbf{x}, u)$	$[7, 8] \times [0, 1]^4$	$[0, 1]$
$i = \vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$i = 13$	5	$\mathbf{f}_{ctrl}(t, \mathbf{x}, u)$	$[45, 52] \times [0, 1]^4$	$[0, 1]$
$i = 14$	9	$\mathbf{f}_{rad}(t, \mathbf{x}, u)$	$[52, 55] \times [0, 1]^8$	$[0, 1]$

Table 6.7: Dimensions (with x_c), vector fields, domains, and input sets for the controlled hybrid system $\mathcal{H}$ of the haemoglobin production model.

Mode	$e = (i, j)$	S_e	R_e
$i = 1$	$(1, 2)$	$t == 4$	$\begin{bmatrix} I_{5,5} \\ O_{4,4} \end{bmatrix}$
$i = 2$	$(2, 3)$	$t == 7$	$\begin{bmatrix} I_{5,5}, O_{4,4} \end{bmatrix}$
$i = 3$	$(3, 4)$	$t == 8$	$\begin{bmatrix} I_{5,5} \\ O_{4,4} \end{bmatrix}$
$i = \vdots$	$\vdots$	$\vdots$	$\vdots$
$i = 13$	$(13, 14)$	$t == 52$	$\begin{bmatrix} I_{5,5} \\ O_{4,4} \end{bmatrix}$

Table 6.8: Transitions, guards, and reset maps of the controlled hybrid system $\mathcal{H}$.

The controlled hybrid system $\mathcal{H}$ associated to the experiment of dataset #2 studied in Section 6.1.1 is given, in a shortened version, in Tables 6.7 and 6.8. The ODEs ($\mathbf{f}_{\text{ctrl}}$) and ($\mathbf{f}_{\text{rad}}$) are given in the previous section 6.1.1, and we consider the parameter set p_{mean} for the values of the parameters other than k_3. In the implementation, we also introduce a variable x_c modelling time, whose derivative is equal to 1. We recall that we want to search for an optimal control $u(t)$ minimizing the distance of the simulated trajectory to the corresponding point in the dataset. In this particular study, we consider $u(t) = k_3(t)$ and we will keep this notation until further notice.

For numerical reasons, it is necessary to scale the parameters and state variables, making it easier for the solver to succeed in solving the relaxed problem. Similarly, to facilitate the numerical optimization we rewrite the control variable $u(t) \in U = [0, 1]$ as $u(t) = \zeta \hat{u}(t)$, with $\zeta \ll 1$ and $\hat{u}(t) \in [0, 1/\zeta]$. While the

scale factor ζ may take different values depending on the numerical optimization details, the objective control $u(t)$ always evolves in $[0, 1]$.

Implementation. Now that we have a valid controlled hybrid system, $\mathcal{H}$, we solve the optimal control problem with intermediate time points defined in (3.20), using the method from Section 3.2.1 and its implementation in Section 3.2.2. The experimental measurement is modelled by the function $m(\mathbf{x}) = {}^{59}\mathrm{H} + 4{}^{59}\mathrm{Hb} = x_6 + 4x_8$. Thus, we set

$$H(\mathbf{x}(T_j)) := (x_6(T_j) + 4x_8(T_j) - 3z_j)^2,$$

as we search to minimize the total residual error term:

$$\varepsilon_{total} = \sum_{1 \leq j \leq n_{exp}} \frac{\sqrt{H(\mathbf{x}(T_j))}}{\sum_{1 \leq j \leq n_{exp}} z_j} \tag{6.3}$$

We recall that the original experimental data points (T_j, z_j) are given in Table 6.9.

Time (h)	7	11	19	27	35	45	55
Measure ($\frac{cpm}{1e^{-7}\mathrm{L}\cdot h^{-1}}$)	16	85	348	391	399	481	395

Table 6.9: Experimental data points (T_j, z_j) used as references.

Here, the input control $k_3(t) = u(t)$ models some hidden mechanism resulting in an evolution of the iron integration rate k_3 with the differentiation of the cells. It should be the same function of time for both the control and the radioactive cells batch. However, as the control generated by Algorithm 1 is piecewise for each mode, and the fact that our data are on the radioactive species only, the solution of the optimization problem with only a final cost $H(\mathbf{x}(T_i))$ is not *balanced*, having a much stronger control in the modes where the radioactive species are evolving. A workaround for the balancing problem is the following. We add a small penalization cost $c_i^1(t) = (0.01u(t))^2$ to equilibrate the control when i corresponds to a mode with radioactive species, otherwise $c_i^1(t) = 0$. In a similar vein, we add another penalization cost $c_i^2(t) = (u(T_j) - u(t))^2$ to avoid when the control strongly varies between two iterations j on the interval $[T_{j-1}, T_j]$ and $j + 1$ on $[T_j, T_{j+1}]$ (with the exception of the first iteration). This leads to $h_i(t, \mathbf{x}(t), u(t)) = c_i^1(t) + c_i^2(t)$. Let us note that, even if these additional costs can eventually degrade the accuracy of the data fitting, we gain in terms of biological interpretation of the resulting traces.

Finally, by partitioning the computation in the time domain, we can greatly reduce the computational cost at each iteration. More technically, since the transitions of the hybrid system $\mathcal{H}$ are fully determined by the time t, we can precompute the function $\lambda : \mathbb{R}^+ \to \mathcal{I}$, which associates a mode $\lambda(t)$ to each time

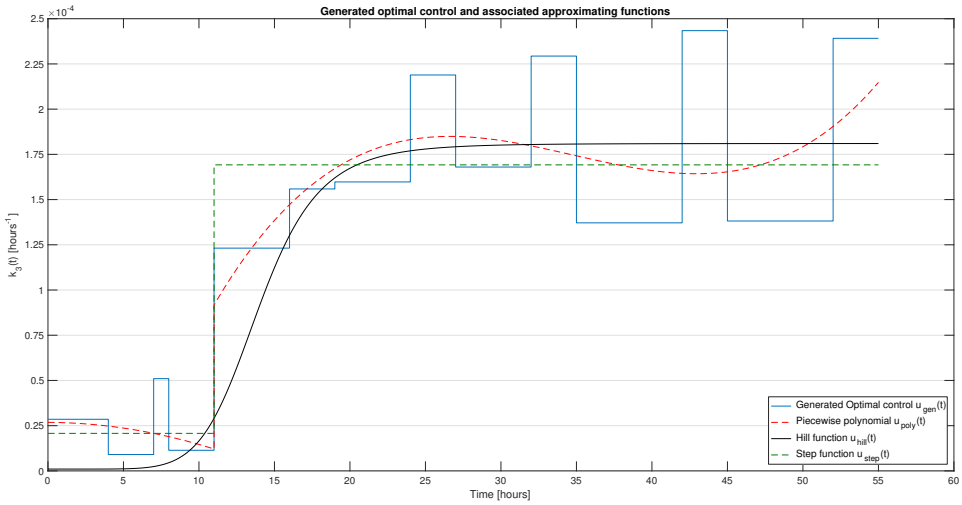

Figure 6.6: Synthesized optimal control and various approximations that yield a realistic interpretation.

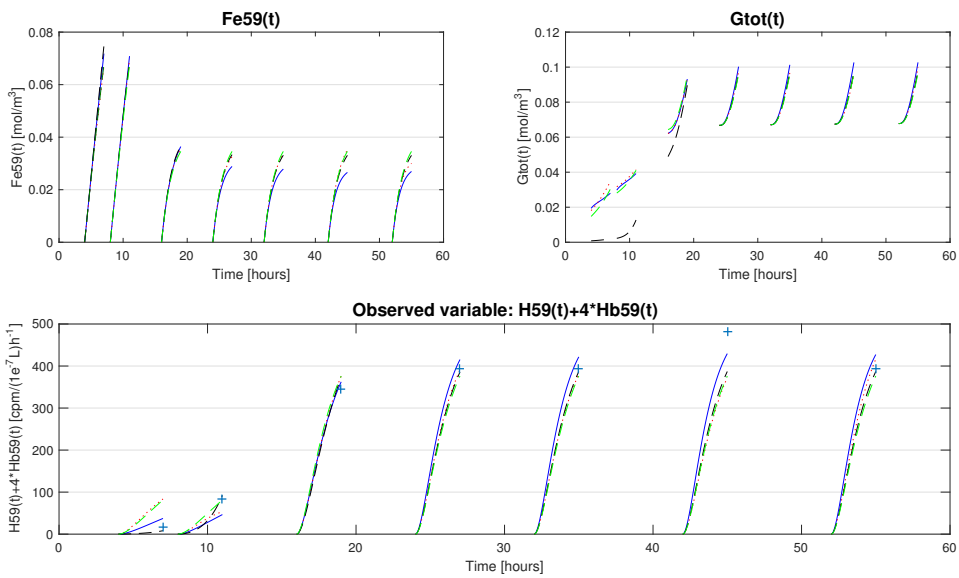

Figure 6.7: Radio-active variables ^{59}Fe, G_{tot} in ($\mathbf{f}_{\text{rad}}$), as well as, the comparison of the measurement function results to the dataset #2 (see Table 6.9).

instant t. Thus, each iteration j of Algorithm 1 can be restrained to the hybrid sub-system $\mathcal{H}^j$ of $\mathcal{H}$, constituted by the modes visited in the interval $[T_{j-1}, T_j]$.

For numerical implementation, the problem on measures is formulated in SPOTLESS[5], and then we extract the primal solution provided by a primal-dual SDP solver. To do so, we use the implementation from [94] to generate the dual problem of (3.17) defined in Chapter 3. As an SDP solver we used MOSEK [185] v.7.1. These tools are used in MATLAB v.9.0 (R2016a). Performance results are obtained with an Intel Core i7-5600U CPU (2.60 GHz) with 16Gb of RAM running on Debian 8.

Applying Algorithm 1 on the $\mathcal{H}$ hybrid system, as described above, we have to solve 7 times the optimization problem (3.17), on 2 mode hybrid systems of respectively 5 and 9 continuous variables in each mode. We only solve the problem for a relaxation order $r = 4$, as any higher order would be too memory expensive. We only synthesize a piecewise constant control, and to avoid oscillation in the resulting control we force $d_u = 0$ in Algorithm 1 from Chapter 3. Using this configuration, the total time taken by Algorithm 1 is 2107s, with 1700s spent in the `HOCP` procedure, and 390s in the `Synth` procedure.

In Figure 6.6, the control generated by Algorithm 1 is shown in blue. This control is piecewise, and clearly divided in two phases: before and after t equals 11 hours. However, the control synthesized is still difficult to interpret as a biological phenomenon. Consequently, we propose three additional fits of this control to ease interpretation by using functions closer to biological knowledge. In Table 6.10 one can find the total error associated to all the possible controls, as well as the previous result of Table 6.6. In Figure 6.7, we show a graphical representation of how closely each function can control the model to reach the desired data points.

Control Type	ε_{total}
Best ε_2 in Table 6.6	0.23
Results generated by Algo 1	0.096
Step function fit	0.12
Piecewise Polynomial fit	0.13
Hill function fit	0.075

Table 6.10: Total error ε_{total} associated to each proposed input.

Discussion. In a simulation-based approach, we have to propose for the desired time varying parameter, a template function to fit the data, e.g. a polynomial of given degree. If we want to fit a polynomial of higher degree, the simulations have to be run again multiple times. On the contrary, the proposed approach returns a

[5]The SPOTLESS implementation was taken from https://github.com/spot-toolbox/spotless

control signal, and since the fit to data points is performed a posteriori, there is no additional computation cost in refining the model.

From the form of the experimental data points, an usual hypothesis is that $k_3(t) = u(t)$ should be similar to a jump function, with a low value for the two first points, and a higher one for the following ones. However, even with such information a good fit is not easily achieved with simulations.

The control generated using Algorithm 1 returns the expected "jump" behaviour for $u(t)$, and even with a low relaxation degree, the total residual error for the generated control is 9.59% which is much lower than the 22.8% from the minimal value of ε_2 in Table 6.6.

We first fit a step function to the generated control, with a change at $t = 11$. The associated error of 12.24% is still lower than Section 6.1.1, yet being higher than the generated control mainly due to the second-to-last point.

The second fit is a piecewise polynomial function in two pieces. The first piece, for $t \in [0, 11]$, is a polynomial of degree 2 while the degree of the second, for $t \in [11, 55]$, is 4. This proposed input control allows to reproduce more accurately, than the step function, the third data point. However, its accuracy is worse on the first and two last points. The total error associated to this control is 13%, being overall the worst of the proposed fits.

Lastly, we fit a Hill function, a function used to model the kinetics of a class of biochemical reactions and which is a very common way to represent biological activation processes. The associated total error is 7.5%, which is the lowest, taking advantages from both the step function and the piecewise polynomial function. In this case, the inaccuracy also mainly comes from the second-to-last point, which is quite separated from trend of the other experimental points, and may be due to some experimental problems (no standard deviation results were available). Without taking this point into consideration for the error computation the error falls to 3% for the Hill function fit.

On this particular example, this method provided a way to generate a control satisfying intermediate points without any *a priori* on a particular form, avoiding the need for extensive numerical simulations. The generated control is accurate, and computed in a reasonable time ($\sim$35min), even for a large hybrid system of 14 modes with at most 9 continuous variables. Using some fitting functions afterwards, it is even possible to refine the results and obtain a more interpretable function for the desired time varying parameters.

Since in this model, the sequence of transitions is known in advance, the use of [94] to solve (3.11) at each iteration of Algorithm 1 is arguable, as other methods can handle this problem. If needed Algorithm 1 can easily be adapted with another method to solve the optimal control problem on hybrid automata. However, Algorithm 1 in its current form does not require any knowledge on the sequence of transition and can be applied to a larger set of biological models. We can also note a similar approach to our own in [26]. In this paper, the authors first search for parameter sets satisfying experimental data on an hybrid dynamical system using particle swarm techniques. Then, they revise the model by searching a parame-

ter as a time varying function using optimal control technique. While they have similar objectives, our work differs from theirs in the methods used for both the parameter fitting and the model revision.

6.2 Mammalian cellular iron homoeostasis (MCIH) model

6.2.1 Context

In this section we apply the set-based analysis method from Chapter 4 on a discretization of the mammalian cellular iron homoeostasis (MCIH) model previously characterized in [5]. Using this model as a proof of concept we show that the method proposed in Chapter 4 can be applied to ensure properties or hypothesis while taking into account uncertainty on both initial conditions and parameters (possibly spanning multiple orders of magnitude).

The ODE system proposed in [5] is built to study and represent the mechanism of iron homoeostasis for a large parameter space. This previous work [5] provides a method to characterize a large valid parameter space (19 parameters, spanning several orders of magnitude), by finding the parameters points which respect a set of temporal constraints and clustering them in multiple ellipsoids. Here we define our experiments based on both a discretization of this model and some previous results from [5].

This model describes the control of the iron concentration inside a cell, thanks to both an iron storage protein, ferritin, and regulatory proteins IRP (Iron Regulatory Proteins). Moreover, both the transferrin receptor TfR1(which influences the iron input in the cell) and the iron exporting protein FPN1a are influenced by the IRP concentration. $\mathrm{Tf}_{\mathrm{sat}}$ is the external saturated transferrin concentration, which is the iron transport protein outside the cell. The concentration of free iron in the cell that is not stored inside ferritin must be well controlled since too much or too little of it can have deleterious effects.

In the presence of a stable concentration of iron-loaded transferrin, outside the cell, the cell system converges to a steady state. When there is no more iron outside the cell ($\mathrm{Tf}_{\mathrm{sat}}$ is almost equal to 0) the non-ferritin bound iron quickly drops for a short time, but then increases again at the expense of ferritin iron and stabilizes for some time (around 10 hours) after the activation of the regulation mechanism. The low iron concentration stimulates the IRP activity which itself activates the release of the iron stored in the ferritin. This supply of iron from the ferritin leads to a pseudo-steady state for a few hours, until the ferritin concentration is too low to release enough iron to maintain the equilibrium. If no iron is added to the medium shortly thereafter, the cell dies.

The model contains 5 state variables (Fe, IRP, Ft, TfR1, FPN1a), and 19 parameters. The dynamics are defined in the ODE system (6.4).

The work [5] observed that for all the valid parameter points, the value of FPN1a is almost not influenced by the value of the other variables during the experiments, FPN1a being almost constant with this modelling. FPN1a, being the

iron exporting protein, should quickly decrease with the iron concentration Fe, as modelled by the IRP dependency. However, the FPN1a concentration stays stable for all the *valid* parameters points.

In this work, we propose to compute the reachable set of FPN1a for the parameters $[k_{\text{FPN1a}_{\text{deg}}}, k_{\text{FPN1a}_{\text{prod}}}, k_{\text{IRP-FPN1a}}, \theta_{\text{Fe}-\text{IRP}}]$ taken in the interval given by the valid parameter points. The computed reachable set must ensure that in presence of external iron, the system evolves to a steady state, and in absence of external iron, there is a plateau of at least 10 hours for the variable Fe, followed by a decrease in iron concentration.

$$\begin{aligned}
\frac{\mathrm{dFt}}{\mathrm{d}t} &= k_{\text{Ft}_{\text{prod}}} - k_{\text{IRP-Ft}}\,\mathtt{sig}(\text{IRP}, \theta_{\text{IRP-Ft}}, d_{\mathtt{sig}}) - k_{\text{Ft}_{\text{deg}}}\text{Ft} \\
\frac{\mathrm{dFe}}{\mathrm{d}t} &= k_{\text{Fe}_{\text{input}}}\text{Tf}_{\text{sat}}\,\text{TfR1} - \text{n}_{\text{Ft}}\,\frac{\mathrm{dFt}}{\mathrm{d}t} - k_{\text{Fe}_{\text{export}}}\text{Fe}\,\text{FPN1a} - k_{\text{Fe}_{\text{cons}}}\text{Fe} \\
\frac{\mathrm{dIRP}}{\mathrm{d}t} &= k_{\text{IRP}_{\text{prod}}} - k_{\text{Fe-IRP}}\,\mathtt{sig}(\text{Fe}, \theta_{\text{Fe-IRP}}, d_{\mathtt{sig}})\text{IRP} - k_{\text{IRP}_{\text{deg}}}\text{IRP} \\
\frac{\mathrm{dFPN1a}}{\mathrm{d}t} &= k_{\text{FPN1a}_{\text{prod}}} - k_{\text{IRP-FPN1a}}\,\mathtt{sig}(\text{IRP}, \theta_{\text{IRP-FPN1a}}, d_{\mathtt{sig}}) - k_{\text{FPN1a}_{\text{deg}}}\text{FPN1a} \\
\frac{\mathrm{dTfR1}}{\mathrm{d}t} &= k_{\text{TfR1}_{\text{prod}}} + k_{\text{IRP-TfR1}}\text{IRP} - k_{\text{TfR1}_{\text{deg}}}\text{TfR1}
\end{aligned} \tag{6.4}$$

where $\mathtt{sig}(x, \theta, d_{\mathtt{sig}}) = \dfrac{x^{d_{\mathtt{sig}}}}{x^{d_{\mathtt{sig}}} + \theta^{d_{\mathtt{sig}}}}$.

The original model of iron homoeostasis (6.4) leads to a huge number of Bernstein coefficients because of the high degree of the sigmoids which are rational functions. On the other hand, simply lowering the degree $d_{\mathtt{sig}}$ can cause significant errors, compared to the original model. To cope with this difficulty, each sigmoid has been approximated by a piecewise function. For a sigmoid function $x^{d_{\mathtt{sig}}}/(x^{d_{\mathtt{sig}}} + \theta^{d_{\mathtt{sig}}})$ (where x and θ are scalar variables), the associated piecewise function of x and θ ($d_{\mathtt{sig}}$ being a constant[6]) is:

$$\mathtt{sig}(x, \theta, d_{\mathtt{sig}}) = \begin{cases} 0, & \text{if } x \leq \dfrac{(d_{\mathtt{sig}} - 2)\theta}{d_{\mathtt{sig}}} \\ \dfrac{d_{\mathtt{sig}}(x - \theta)}{4\theta} + \dfrac{1}{2}, & \text{if } x > \dfrac{(d_{\mathtt{sig}} - 2)\theta}{d_{\mathtt{sig}}} \text{ and } x \leq \dfrac{(d_{\mathtt{sig}} + 2)\theta}{d_{\mathtt{sig}}} \\ 1, & \text{if } x > \dfrac{(d_{\mathtt{sig}} + 2)\theta}{d_{\mathtt{sig}}} \end{cases}$$

The new MCIH model contains 4 parameters $k_{\text{FPN1a}_{\text{deg}}}$, $k_{\text{FPN1a}_{\text{prod}}}$, $k_{\text{IRP-FPN1a}}$ and $\theta_{\text{Fe}-\text{IRP}}$ taken on large intervals. While the parameters $k_{\text{FPN1a}_{\text{deg}}}$, $k_{\text{FPN1a}_{\text{prod}}}$, and $k_{\text{IRP-FPN1a}}$ appear linearly in the dynamics, $\theta_{\text{IRP}-\text{FPN1a}}$ appears non-linearly. For this reason, we treat the parameter $\theta_{\text{IRP}-\text{FPN1a}}$ as a sixth variable, and thus the term $\text{IRP}^5/(\text{IRP}^5 + \theta^5_{\text{IRP}-\text{FPN1a}})$ is not approximated by a piecewise linear function but its approximation is non-linear (rational function) in $\theta_{\text{IRP}-\text{FPN1a}}$ in one of the pieces. These piecewise approximations lead to a new model where each

[6]Here we consider $d_{\mathtt{sig}} = 5$.

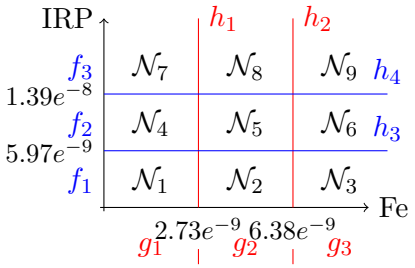

Figure 6.8: 9 modes generated by $[f_1; f_2; f_3] = [0; 1.25e^8 \cdot \text{IRP} - \frac{3}{4}; 1]$ the approximation of $\texttt{sig}(\text{IRP}, \theta_{\text{IRP}-\text{Ft}})$ and $[g_1; g_2; g_3] = [0; 2.74e^8 \cdot \text{Fe} - \frac{3}{4}; 1]$ the one of $\texttt{sig}(\text{Fe}, \theta_{\text{Fe}-\text{IRP}})$. The constraints h_i are of the form $\mathbf{a} \cdot x \leq c$, for example h_1: Fe$\leq 2.73e^{-9}$.

sigmoid is substituted by a 3-piece approximation. In place of one ODE system, the dynamics is now hybrid with 15 modes [7]. The 9 domains corresponding to the approximations of $\texttt{sig}(\text{IRP}, \theta_{\text{IRP}-\text{Ft}})$ and $\texttt{sig}(\text{Fe}, \theta_{\text{Fe}-\text{IRP}})$ are represented in Figure 6.8.

6.2.2 Set-based analysis

We recall that our goal is to validate the observations which were obtained in [5] using numerical simulations, about the regulation of FPN1a. These observations were made with parameter values chosen such that the system respects some properties:

- In presence of external iron input ($\text{Tf}_{\text{sat}} \neq 0$), the Fe, Ft, and IRP concentrations reached a steady state.
- In absence of external iron input ($\text{Tf}_{\text{sat}} = 0$), the iron concentration first stabilized on a plateau for at least 10h, then decreased to 0.

The set-based analysis produces an over-approximation of the reachable set. Because of accumulated error, this set may grow at each step in every directions. We thus do not impose strong constraints for the plateau definition, and currently restrict to a qualitative observation. For the same reason and because we are interested in the question whether the FPN1a concentration strongly decreases during the $\text{Tf}_{\text{sat}} = 0$ phase, we restrict to a qualitative observation on the lower bound of the reachable set of FPN1a. The reachability analysis of the adapted model was done using the following method: starting from initial conditions (taken from [5]) and a corresponding valid parameter set p, these initial conditions are bloated to a set. The following parameters $[k_{\text{FPN1a}_{\text{deg}}}, k_{\text{FPN1a}_{\text{prod}}}, k_{\text{IRP-FPN1a}}, \theta_{\text{IRP-FPN1a}}]$ are extended to cover a few orders of magnitude based on the results of [5]. The considered initial conditions and parameter space are given in Table 6.11.

From this starting initial set, we first let the system evolve to a steady state

[7] Two sigmoids are on IRP, and one on Fe

Variable	Interval	Unit
Fe	$[2.27e^{-8}\ \ 2.28e^{-8}]$	mol/L
IRP	$[6.646e^{-9}\ \ 6.647e^{-9}]$	mol/L
Ft	$[2.804e^{-7}\ \ 2.805e^{-7}]$	mol/L
FPN1a	$[2.35e^{-7}\ \ 3.61e^{-7}]$	mol/L
TfR1	$[9.8e^{-8}\ \ 10.2e^{-8}]$	mol/L

parameters	Value or Interval	Unit
$k_{\text{FPN1a}_{\text{deg}}}$	$[1e^{-7}\ \ 1e^{-6}]$	s^{-1}
$k_{\text{FPN1a}_{\text{prod}}}$	$[1e^{-17}\ \ 1e^{-13}]$	$\text{mol}(\text{L}\,s)^{-1}$
$k_{\text{IRP-FPN1a}}$	$[1e^{-17}\ \ 1e^{-13}]$	$\text{mol}(\text{L}\,s)^{-1}$
$\theta_{\text{IRP-FPN1a}}$	$[1e^{-8}\ \ 2.01e^{-6}]$	$\text{mol}\,\text{L}^{-1}$
$k_{\text{Fe}-\text{IRP}}$	$5.24e^{-5}$	s^{-1}
$k_{\text{Fe}_{\text{cons}}}$	$1.56e^{-1}$	s^{-1}
$k_{\text{Fe}_{\text{export}}}$	$2.191e^{3}$	$\text{L}(\text{mol}\,s)^{-1}$
$k_{\text{Fe}_{\text{input}}}$	$3.65ee^{-2}$	s^{-1}
$k_{\text{Ft}_{\text{deg}}}$	$2.92e^{-5}$	s^{-1}
$k_{\text{Ft}_{\text{prod}}}$	$8.93e^{-12}$	$\text{mol}(\text{L}\,s)^{-1}$
$k_{\text{IRP}-\text{Ft}}$	$8.71e^{-12}$	$\text{mol}(\text{L}\,s)^{-1}$
$k_{\text{IRP}-\text{TfR1}}$	$3.03e^{-4}$	s^{-1}
$k_{\text{IRP}_{\text{deg}}}$	$1.5e^{-5}$	s^{-1}
$k_{\text{IRP}_{\text{prod}}}$	$4.48e^{-13}$	$\text{mol}(\text{L}\,s)^{-1}$
$k_{\text{TfR1}_{\text{deg}}}$	$2.23e^{-5}$	s^{-1}
$k_{\text{TfR1}_{\text{prod}}}$	$1.78e^{-13}$	$\text{mol}\,(\text{L}\,s)^{-1}$
$\theta_{\text{Fe}-\text{IRP}}$	$9.89e^{-9}$	$\text{mol}\,\text{L}^{-1}$
θ_{IRP-Ft}	$4.56e^{-9}$	$\text{mol}\,\text{L}^{-1}$
nFt	177.4	–
$d_{\texttt{sig}}$	5	–

Table 6.11: On the left: set of initial conditions (after the stabilization phase). On the right: considered parameter space for the reachability analysis.

with $\text{Tf}_{\text{sat}} \neq 0$. This is the mode where the system should be stable. We let the system stabilize for a few hours. Some results of this stabilization are shown in Figures 6.9 and 6.10. It is clear that the system evolves towards an invariant set, and converges. Because this tool does not compute a precise invariant set, we will take, for the next part of the computation, this over-approximation as the initial set. The initial set for this part is given in Table 6.11.

In the second part of the analysis we reduce Tf_{sat} to 0: this is the mode where the external iron is depleted. Then, we simulate 32 hours (230400 iterations using a fixed time step of 0.5 seconds) of the iron depleted mode. In Figures 6.11,6.12, and 6.13, we can observe the different phases of the computations in different colors (blue, red, green, and purple).

– Phase 1 (blue): On the initial state previously computed, we apply the following change: Tf_{sat} drop from 1 to 0. Experimentally this corresponds to washing the external medium of the cell and replacing it by a medium without iron. This sudden change of Tf_{sat} leads to the very low iron concentration at t near 0 (see Figure 6.11). This very low iron concentration triggers the production of IRP, which itself activates the release of iron by the ferritin. The iron and the IRP concentrations quickly grow back until both the IRP and iron are around their respective thresholds $\theta_{\text{IRP}-\text{Ft}}$ and $\theta_{\text{Fe}-\text{IRP}}$. The IRP increase slows down while the iron concentration stabilizes. To compute precisely this blue part, reachability analysis was done using 15

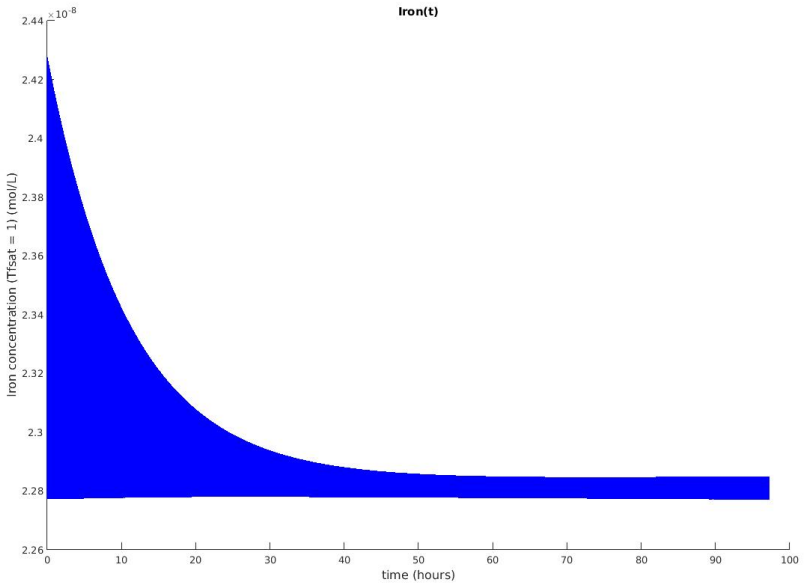

Figure 6.9: Iron stabilization

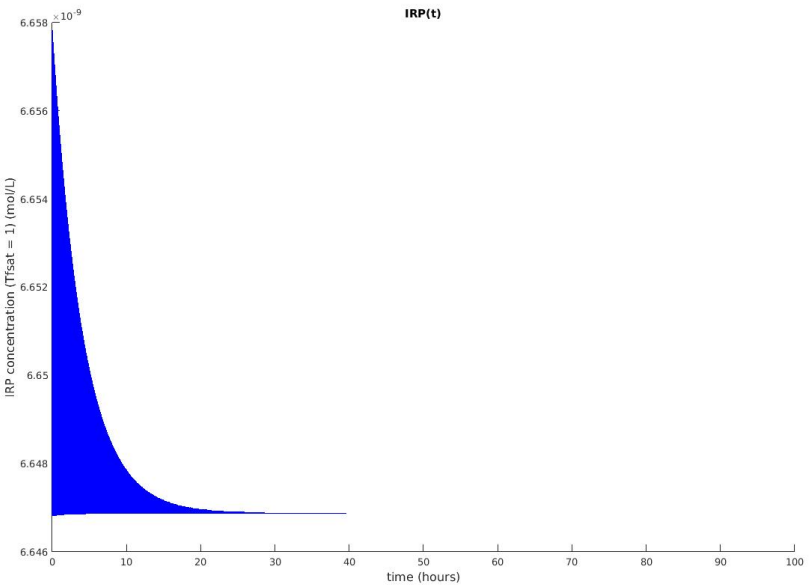

Figure 6.10: IRP stabilization

different directions to represent template polyhedral set. The reachability computation time for the blue part is around 2 hours.

- Phase 2 (red): Because the iron is now in the plateau, we need to ensure that the analysis is as precise as possible. Thus, we reduce the error by bisecting the set on the IRP axis, and perform reachability analysis with two smaller sets instead of one big set. Even with such a method, one can observe the fast growing accumulated error in the red phase. In the red part, the system overlaps two partitions of IRP: the one where $\mathtt{sig}(\mathrm{IRP}, \theta_{\mathrm{IRP-Ft}})$ is represented by an affine function, and the one where $\mathtt{sig}(\mathrm{IRP}, \theta_{\mathrm{IRP-Ft}}) = 1$. Overlapping two partitions increases the error during a short time, leading to the observed growth of the reachable set in red in Figure 6.11.
- Phase 3 (green): Once the reachable set has completely crossed the border between the two partitions, and $\mathtt{sig}(\mathrm{IRP}, \theta_{\mathrm{IRP-Ft}}) = 1$, the reachable set quickly contracts, and the iron concentration begins to decrease notably. Reciprocally, the IRP concentration increases trying to compensate the lack of iron. However at this moment, there is no longer enough ferritin to supply the cell in iron. The red part and the green part took around 3 hours to compute in total.
- Phase 4 (purple): The iron concentration is not stable in a plateau, but now decreases to 0. To compute this part we did not need as good precision as before and used a simple box over-approximation, and the computation time of the purple part is around 15 minutes.

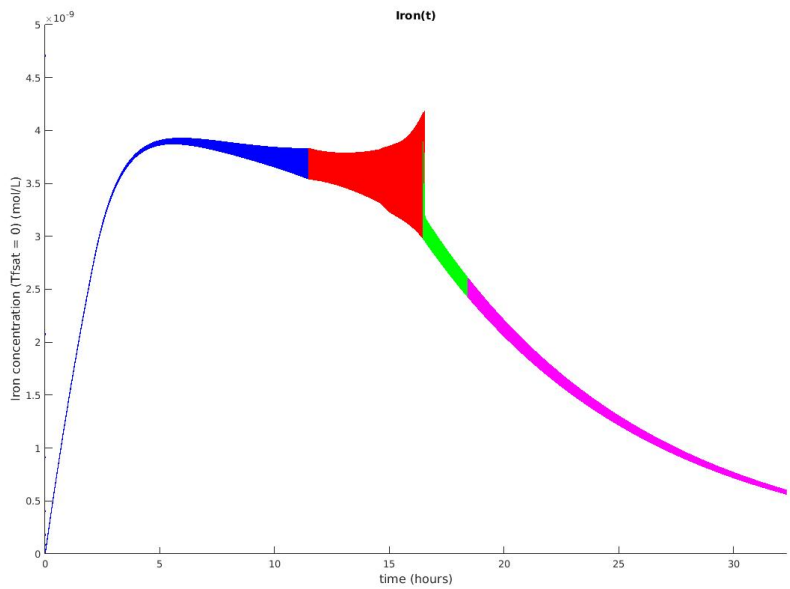

Figure 6.11: Iron reachable set for $\mathrm{Tf}_{\mathrm{sat}} = 0$

The reachability analysis of the system allows us to validate the previous observation made in [5] using point-based simulations: the regulation term of FPN1a by

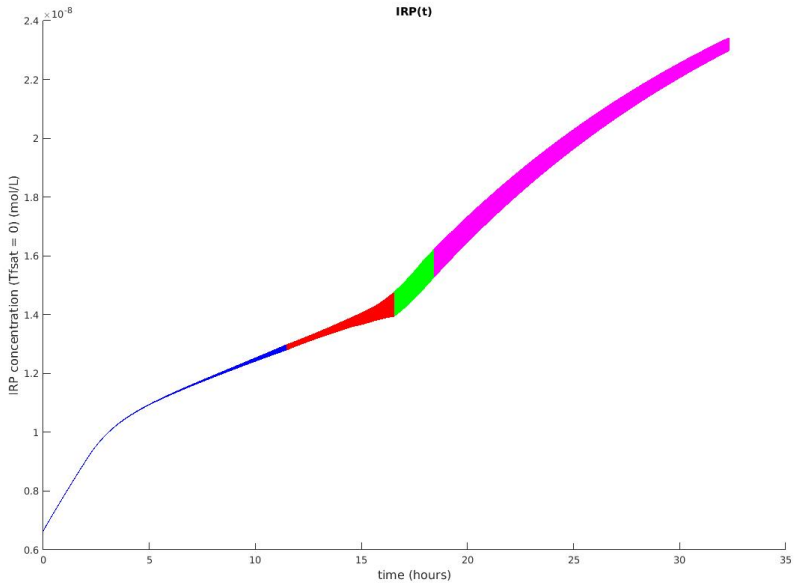

Figure 6.12: IRP reachable set for $\mathrm{Tf}_{\mathrm{sat}} = 0$

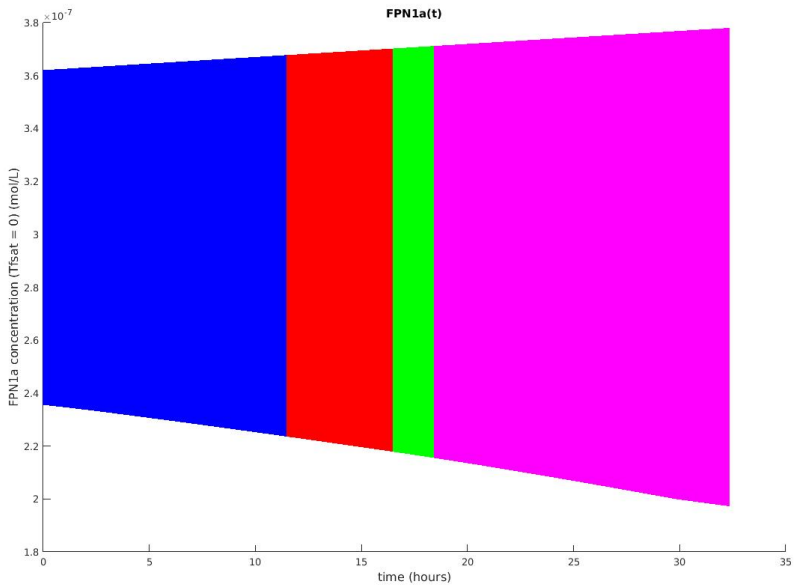

Figure 6.13: FPN1a reachable set for $\mathrm{Tf}_{\mathrm{sat}} = 0$

only IRP in this model and within these parameters intervals is not effective. This suggests that another actor is needed for the regulation of FPN1a. Indeed, even

with an initial set for FPN1a, and having the parameters influencing FPN1a spanning over large intervals (multiple orders of magnitude), the reachability analysis results show that the model satisfies the expected properties:

1. Fe, IRP, and Ft tend to a small invariant set when $\mathrm{Tf}_{\mathrm{sat}} \neq 0$.
2. The iron concentration reaches a "plateau" for at least 10 hours.
3. After reaching a plateau the iron concentration decreases to 0.
4. The IRP concentration first increases quickly then more slowly during the plateau and then increases quickly again.

However given all those conditions, the FPN1a concentration did not undergo any notable decrease. Indeed, in Figure 6.13, while the upper bound slowly increases due to the accumulated error, the lower bound, which is conservative, does not decrease notably unlike what we could expect.

This analysis shows that if the model efficiently represents the regulation of the iron concentration with the IRP proteins, it does not fully model the FPN1a regulation, and and IRP is not the main regulating factor in this regime on the FPN1a concentration.

6.3 Cadmium impact on glucose response model

Context. In this section we present a preliminary analysis of the glucose response mechanism and its evolution when exposed to cadmium (Cd). This work follows a thorough experimental study [96] about the effect of low-dose cadmium exposure on the glucose regulation, and its link to type 2 diabetes. Indeed it was observed that some populations exposed to low-dose[8] of cadmium for a long period developed type 2 diabetes syndromes [186, 187, 188]. The experiments in [96] are performed on both cultured cells and animals: in this section we study the experiments on the animals.

Glucose regulation mechanism mainly depends on an hormone called insulin which trigger the integration of plasma glucose into cells: especially muscles or adipocytes (fat holding cells). The insulin is produced by the β-cells of the pancreas in response to a glucose increase [189, 190]. As we search to link the cadmium exposure to type 2 diabetes, it was first hypothesized that the cadmium had a negative effect on insulin production mechanism in the β-cells. However, preliminary experiments from [96] on cells cultures show not clear results in this sense.

In parallel to the cells cultures, a second set of experiments were performed on rats. These experiments can be separated in two subsets: the experiments on adults animals directly exposed to the cadmium, and the experiments on pups (baby rats) indirectly exposed to the cadmium through the mother placenta before birth, or milk after birth. The last experiments on pups are summarized in Figure 6.14.

[8] Compared to usual cadmium toxicity studies.

During the experiments described in Figure 6.14 multiple oral glucose tolerance tests (OGTT) were performed. This test consists in measuring the evolution of plasma glucose concentration during the 2 hours following an oral glucose intake. The tests were performed on the pups at 21 days after birth, which is the end of the milk based diet, 26 days, and 60 days. We first propose a preliminary modelling

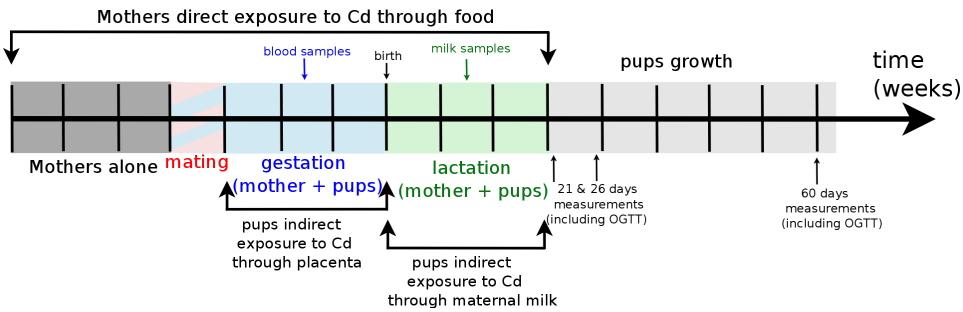

Figure 6.14: Protocol for indirect exposure of small rats to Cadmium through their mothers.

and parameter analysis of the OGTT results at 21, 26 and 60 days. Then, we discuss of the application of MOEPLA for the a further study of the experiments practised on the pups.

Parameter study. We first propose a model to reproduce the OGTT results obtained by [96] with the protocol described in Figure 6.14. To this aim, we use the MINMOD model [97], a small ODE model describing the evolution of glucose concentration after an initial glucose intake. We also refer to [191] for a review of glucose regulation models, and in particular to [192, 193, 194, 195] for a modelling of the OGTT. Finally, we highlight the work [196] which contains a very detailed model of glucose response after a meal. We may want to apply formal methods or computationally expensive techniques on our model, for this reason we use the minimal model MINMOD as a starting point.

The model MINMOD from [97] is not designed for OGTT, but for intravenous glucose tolerance tests (IvGTT). Therefore, we cannot consider that the plasma glucose is already at its maximum concentration at $t = 0$, as it is done for IvGTT studies. Complex OGTT models such as [196] use compartmental modelling to represent the multiple stages of the digestion, and to obtain the glucose rate of appearance in plasma after the meal. In first approximation we propose a simpler modelling using directly experimental results measuring the glucose rate of appearance in the winstar rat. From [197, Figure 4] we determine the maximum of the rate of appearance is obtained $\sim$ 30min after the meal[9]. Similarly, the initial value of the rate of appearance is $\sim 70\%$ of its maximum (see Figure 6.15). This curve

[9]We consider that the time food spends in stomach is close to zero as the glucose meal is liquid.

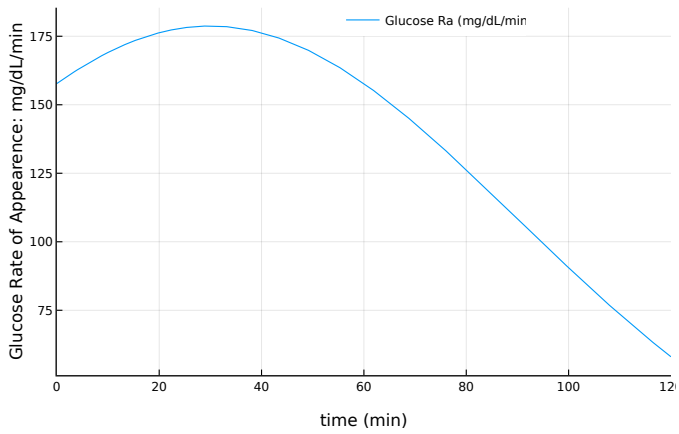

Figure 6.15: Approximation of the plasma glucose rate of appearance.

is modelled by the continuous function $\mathrm{G_{RA}}(t)$:

$$\mathrm{G_{RA}}(t) = K \frac{1}{\sigma\sqrt{2\pi}} \mathrm{e}^{\frac{-(t-\mu)^2}{2\sigma^2}} ,$$

where $\mu = 30$, and $\sigma = 60$. The value of the parameter K is determined by the actual quantity of glucose fed to the rats. Let $\mathrm{m_{Glc}}$ be the mass of glucose fed to the rats and V_{Blood} the rat blood volume as given in [198]. Then given an administrated concentration of glucose $\mathrm{m_{Glc}}/V_{Blood}$, the value of K is the solution of the following equation:

$$\int_0^{+\infty} K \frac{1}{\sigma\sqrt{2\pi}} \mathrm{e}^{\frac{-(t-\mu)^2}{2\sigma^2}} \mathrm{dt} = \frac{\mathrm{m_{Glc}}}{V_{Blood}} .$$

In presence of glucose, the release of insulin by the β-cells can be separated in two phases. The first phase is the exocytosis of the insulin already present and stored near to the cellular membrane. The second phase corresponds to the release of insulin whose production was triggered by the glucose increase. The MINMOD model correctly simulates the second phase of insulin production, but does not reproduce satisfactory the first phase. To address this problem we add an additional state variable representing the insulin already present and ready to be released in the blood circulation. Finally, the adapted MINMOD model is given by (6.5).

$$\begin{aligned}
\dot{\mathrm{G}} &= -p_1(\mathrm{G}(t) - \mathrm{G}_b) - r_{Cd}\,\mathrm{X}(t)\mathrm{G}(t) + \mathrm{G_{RA}}(t) \\
\dot{\mathrm{X}} &= -p_2\mathrm{X}(t) + p_3(\mathrm{I}(t) - \mathrm{I}_b) \\
\dot{\mathrm{I}} &= -n\,\mathrm{I}(t) + \gamma\,(\mathrm{G}(t) - h)\,t + p_4\mathrm{I}_s(t) \\
\dot{\mathrm{I}}_s &= -p_4\mathrm{I}_s(t)
\end{aligned} \tag{6.5}$$

In the ODE system (6.5), G is the glucose concentration in circulating blood, X the rate of glucose absorption in muscles and adipocytes due to insulin, I the insulin

concentration in circulating blood, and I_s the insulin concentration stored in the β-cells and ready to be released.

The experimental results are separated into 3 groups: the control group which was not exposed[10] to cadmium, the group Cd1 which shows a medium cadmium exposure, and the group Cd2 with a high cadmium exposure.

To estimate the goodness of fit of a given simulation compared to the experimental data, we use the root weighted least-square error:

$$\varepsilon(\mathbf{k}) = \sqrt{\sum_i W_i(\mathbf{x}_{exp,i} - \mathbf{x}_{simu}(t_i, \mathbf{k}))^2}\,, \tag{6.6}$$

where $\mathbf{k}$ is a parameter set, and $\mathbf{x}_{simu}(t, \mathbf{k})$ its associated simulation of the OGTT. The weight W_i is determined by the equation:

$$W_i = \frac{1}{\sigma^2(\sum_i \mathbf{x}^2_{exp,i})}\,,$$

where σ are the variance to the mean associated to the i-th data point.

The initial condition, and parameters are searched in the intervals proposed in [97] bloated by one order of magnitude. To ease the parameter search, when fitting the parameter to the datasets corresponding to group Cd1 and Cd2 we only consider a few hypotheses on the evolution of the parameters (compared to the one fitting the control group). At first for the experiments at 21 days we consider the following hypotheses:

- **Hypothesis 1:** Increase or reduction of the insulin sensibility.
  - **Hypothesis 1.1:** r_{Cd} varies: this shows the effect of cadmium on the glucose absorption in the cells. If $r_{Cd} < 1$ then the system has developed insulin resistance, otherwise the system is more sensitive to insulin.
  - **Hypothesis 1.2:** p_3 varies: this represents the effect of insulin on the rate of absorption dynamics.
- **Hypothesis 2:** n varies: this models an effect on the insulin degradation.
- **Hypothesis 3:** p_2 varies: this affects the degradation of the glucose absorption rate.
- **Hypothesis 4:** γ varies: this modifies the sensitivity to glucose of the phase 2 of insulin production.

A 21 days, the goodness of fit for each hypothesis and associated to group Cd1 dataset are shown in Table 6.13. The associated best parameter sets are given in Table 6.14. The initial conditions are given in Table 6.12. The simulation corresponding to the best parameter fits are shown in Figure 6.16. The goodness of fit

[10] Except the default food pollution as it is noted in [96].

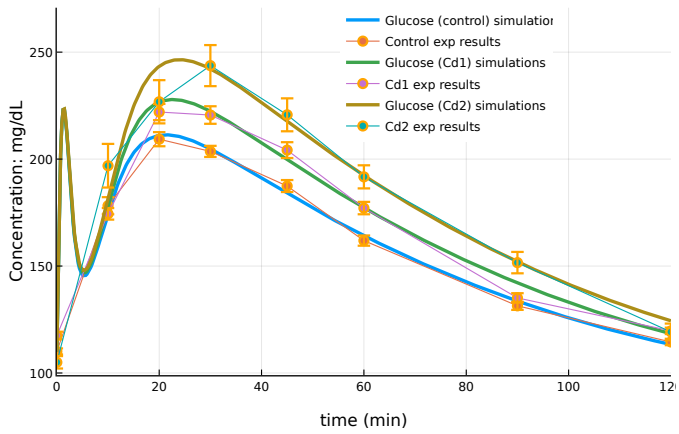

Figure 6.16: Simulations of the OGTT at 21 days for the control group, Cd1 and Cd2 groups.

associated to the control group data set is 0.00512. The best fits for group Cd1 are obtained considering the Hypothesis 4: a reduction of glucose sensibility of β-cells during the phase 2 of insulin production. We consider the same hypothesis when searching for a parameter fit of group Cd2 dataset: the best solution yields a goodness of 0.006. After 26 days, we consider an additional hypothesis. Indeed, from

Variable	Value	Unit
$G(0)$	110.0	mg/dL
$X(0)$	0.0	min^{-1}
$I(0)$	16.0	nU/dL
$I_s(0)$	5950.0	nU/dL

Table 6.12: Initial condition determined for the control group at 21 days. These initial conditions are conserved for the groups Cd1 and Cd2. Note that 1U = 0.0347mg of insulin.

Hypothesis	Cd1
Hyp 1.1	0.0203
Hyp 1.2	0.0185
Hyp 2	0.0178
Hyp 3	0.0202
Hyp 4	0.0117

Table 6.13: Goodness of fit (6.6) of each hypothesis applied to group Cd1 dataset at 21 days.

21 days to 26 days the pups went from a milk-based diet to a "normal" food diet. This change of diet induces important change on the regulation mechanism: this is observed by an evolution of the parameter set fitting the control group dataset.

– **Hypothesis 5:** $[p_2, \gamma, h, p_4]$ are all allowed to vary: these parameters are the ones which differ the most between the fits of the control group at 21 days and 26 days.

The goodness of fit for each hypothesis on group Cd1 and Cd2 are given in Table 6.16. The associated best parameters sets are given in Table 6.17. The initial

Parameters	Ctrl	Cd1	Cd2	Unit
p_1	0.01	–	–	min^{-1}
G_b	100.0	–	–	mg/dL
p_2	0.56	–	–	min^{-1}
p_3	0.0142	–	–	$(\text{dL/nU})\text{min}^{-2}$
I_b	10.0	–	–	nU/dL
n	5.93	–	–	min^{-1}
γ	0.0625	0.05	0.04	$(\text{nU/dL})\text{min}^{-2}$
h	90.0	–	–	mg/dL
p_4	0.07	–	–	min^{-1}
r_{Cd}	1.0	–	–	N.U.

Table 6.14: Parameters values fitted for the control group as well as the groups Cd1 and Cd2 at 21 days (considering hypothesis 4).

conditions are given in Table 6.15. The simulation associated to the best fits are shown in Figure 6.17. The goodness of fit associated to the control group data set is 0.011. We note that Hypothesis 5 yields the best results, and the associated parameter values are in-between the values of the control at 21 days and 26 days. One interpretation of this hypothesis would be that the cadmium affects the speed at which the organism adapts itself to a new diet. Finally the parameter set fitted to the datasets at 60 days are given in Table 6.20. The goodness of fit of the control group dataset is 0.006. As shown in Table 6.19, it is hard to distinguish a hypothesis for the fit of the dataset at 60 days of the group Cd2: all of the tested one yielded good results, but this is mainly due to the high variance on this dataset. We introduce an additional hypothesis to better fit the dataset of the group Cd1.

Variable	Value	Unit
G(0)	76.0	mg/dL
X(0)	0.0	min^{-1}
I(0)	34.0	nU/dL
$\text{I}_s(0)$	5950.0	nU/dL

Table 6.15: Initial condition determined for the control group at 26 days. These initial conditions are conserved for the groups Cd1 and Cd2. Note that 1U = 0.0347mg of insulin.

Hypothesis	Cd1	Cd2
Hyp 1.1	0.009	0.012
Hyp 1.2	0.009	0.012
Hyp 2	0.009	0.012
Hyp 3	0.009	0.012
Hyp 4	0.010	0.010
Hyp 5	0.005	0.005

Table 6.16: Goodness of fit (6.6) of each hypothesis applied to the datasets of groups Cd1 and Cd2 at 26 days.

- **Hypothesis 6:** h varies: this models an influence on the threshold to trigger the phase two of insulin production.

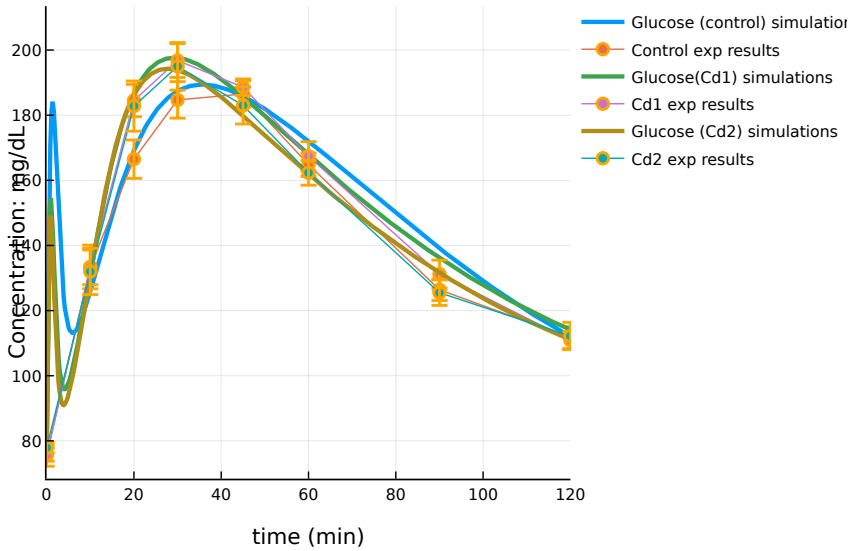

Figure 6.17: Simulations of the OGTT at 26 days for the control group, Cd1 and Cd2 groups.

Parameters	Ctrl	Cd1	Cd2	Unit
p_1	0.01	–	–	min^{-1}
G_b	100.0	–	–	mg/dL
p_2	0.50	0.805	0.86	min^{-1}
p_3	0.0312	–	–	$(\text{dL/nU})\text{min}^{-2}$
I_b	10.0	–	–	nU/dL
n	5.33	–	–	min^{-1}
γ	0.0165	0.0347	0.0410	$(\text{nU/dL})\text{min}^{-2}$
h	65.0	85.0	85.0	mg/dL
p_4	0.035	0.0585	0.065	min^{-1}
r_{Cd}	1.0	–	–	N.U.

Table 6.17: Parameter values fitted for the control group as well as the groups Cd1 and Cd2 at 26 days (considering Hypothesis 5).

Overall, we propose multiple parameter sets associated to each particular experiment. All of these fits are good and allow us to successfully reproduce the data. Additionally, we propose possible interpretations on the cadmium impact at each step of the pups growth.

However, the results are local and only represent one possible valid parameter set for each experiment. To go further, we need to provide a model of the cadmium absorption by the pups and we have to make explicit its influence on the dynamics

Variable	Value	Unit
G(0)	95.0	mg/dL
X(0)	0.0	min^{-1}
I(0)	34.0	nU/dL
$\text{I}_s(0)$	5950.0	nU/dL

Table 6.18: Initial condition determined for the control group and the groups Cd1 and Cd2 at 60 days.

Hypothesis	Cd1	Cd2
Hyp 1.1	0.008	0.005
Hyp 1.2	0.008	0.005
Hyp 2	0.008	0.005
Hyp 3	0.008	0.005
Hyp 4	0.007	0.005
Hyp 6	0.005	0.005

Table 6.19: Goodness of fit (6.6) of each hypothesis applied to the datasets of groups Cd1 and Cd2 at 60 days.

Parameters	Ctrl	Cd1	Cd2	Unit
p_1	0.01	–	–	min^{-1}
G_b	100.0	–	–	mg/dL
p_2	0.55	–	–	min^{-1}
p_3	0.0205	–	0.0210	$(\text{dL/nU})\text{min}^{-2}$
I_b	10.0	–	–	nU/dL
n	5.33	–	–	min^{-1}
γ	0.0265	–	–	$(\text{nU/dL})\text{min}^{-2}$
h	65.0	73.0	–	mg/dL
p_4	0.05	–	–	min^{-1}
r_{Cd}	1.0	–	–	N.U.

Table 6.20: Parameter values fitted for the control group as well as the groups Cd1 and Cd2 at 60 days (considering hypothesis 1.2 for Cd2 and hypothesis 6 for Cd1).

to confirm our hypotheses. Moreover, unless we ensure that there is no other valid parameter set, our interpretations are just one possible solution.

Future work. In addition to the modelling of the OGTT, we need to model the cadmium absorption into the mother and pups organs at each step of the protocol. In the following we propose a simple example of specification for the protocol from Figure 6.14. From the associated hybrid system we still have to find a parametrization of the cadmium absorption rate.

For a first example of specification of the protocol from Figure 6.14, we consider two experimental objects: the mother $\mathcal{O}_{\text{mother}}$ and the pups $\mathcal{O}_{\text{pup}}$. We consider two mechanisms of Cadmium absorption (they may differ for the mother and the pups): $\mathcal{F}_{\text{mother}}$ and $\mathcal{F}_{\text{pups}}$. We also note the default parameters space of the mother K_{mother}, and the one of the pups K_{pups}.

We remark that the mother and the pup could be merged into one single experimental object. Indeed, in the current formalization described in Chapter 5, we do

not accept common variables or parameters. As stated in Section 5.3, this is one of the points we want to address to improve the MOEPLA language in the future. If we want to make the pups depend of the mother, we currently need to define a single object including the coupled mechanisms of the mother and the pups. In the following, we give the sketch of the formalisation with two objects. However, the approach with one object is similar.

The time is counted in days, and the protocol last at most 105 days (15 weeks). At $t = 0$ day, we start with the mother alone: $I_0(\mathcal{O}_{\text{mother}})$. After 21 days the mothers enter in the mating period which leads to pregnancy: this period lasts at most 1 week. In MOEPLA, this can be expressed as a perturbation under the label L_{pregnacy} that occurs in the time interval $t \in [21, 28]$ days:

$$\mathcal{A}_k(\mathcal{O}_{\text{mother}}, L_{\text{pregnacy}}, K_{\text{mother}} \leftarrow K_{\text{pregnant}}, t \in [21, 28])\,.$$

Similarly, the pups come into being under this perturbation and we associate a creation action with the label L_{pregnacy}:

$$\mathcal{A}_{create}(\mathcal{O}_{\text{pup}}, L_{\text{pregnacy}}, \mathbf{x}_0^{\text{pup}}, t \in [21, 28])\,.$$

In a similar manner we can define a perturbation on a change of the parameter space of both the mother and the pups at the birth stage, and the switch to the normal food. Measurements are performed independently on either the pups (at time $t = 70$ days, 76 days and 105 days) or the mother (at $t = 39$ days and 60 days). All these measurements can be expressed in MOEPLA using the action $\mathcal{A}_{meas}$ on either the variables of the object $\mathcal{O}_{\text{mother}}$ or $\mathcal{O}_{\text{pup}}$.

CHAPTER 7

Conclusion

7.1 Summary 131
7.2 Future work 132

7.1 Summary

In this thesis, we addressed three issues in biological systems modelling: model design, model validation, and model revision. In addition, we investigated three biological case studies and used them as proof of concept for our different methods.

Model revision. In Chapter 3, we proposed a method to revise a hybrid dynamical system with respect to some observations in form of intermediate time measures. The method searches for time varying parameters which produce better approximations of an underlying mechanism, compared to constant parameters. For this purpose, it uses an algorithm based on the optimal control method for hybrid systems proposed in [94]. The model revision is achieved without imposing beforehand any structure on the sought law, since we consider the case where we do not possess any additional insight or knowledge.

Set-based simulation. In Chapter 4, we extended the previous work of [83] for reachability analysis of discrete time polynomial systems with uncertain parameters. We contributed in this problem an extension of [83] to discrete time piecewise-rational functions, allowing handling a larger panel of biological applications. Moreover, we also give a few improvements to speed up the actual reachability analysis in some particular cases. In addition, we proposed another set-based

simulation method using the Krivine-Stengle representations. We showed how to adapt the Krivine-Stengle representation based method for an efficient application to reachability analysis of discrete-time polynomial systems with uncertain parameters in box domains.

MOEPLA: Modelling Oriented Experimental Protocol Language. In Chapter 5, we proposed a preliminary work on a language to formally specify an experimental protocol while taking into account a model of a mechanism. In its current form, the proposed language allows modelling multi-stage evolutions of multiple experimental objects in parallel, with possible synchronization steps. It also allows automatic generation of a formal framework using the hybrid automaton formalism. Taking advantage of non-determinism inherent to hybrid automata that can model uncertainty in biological models, we can either validate a model while taking into account an existing protocol (with its uncertainties), or verify that a future protocol (or therapy) will always be correctly executed. We demonstrated the use of MOEPLA on two experimental protocols associated to the haemoglobin production model and the glucose response model, respectively.

Case studies. Finally, in Chapter 6 we described three modelling studies. We first investigated a model of haemoglobin production during the differentiation stages of the an erythroblast into an erythrocyte (also called red blood cells). In the first part of this study, we used a simple exploratory scheme to perform parameter estimation with respect to multiple experimental data sets. In the second part, we used this model to demonstrate the applicability of our model revision method. This model revision with a time varying parameter enabled us to better reproduce a considered dataset. We also note that from the inferred solution we derived multiple hypotheses which led to a meaningful biological interpretation of the time varying parameter as an activation function.

The second case study used the iron homoeostasis model designed in [45]. In this work, we applied the Bernstein reachability analysis from Chapter 4 to confirm a hypothesis formulated in [45] using exhaustive methods for uncertain parameters and initial sets.

The last case study was a preliminary modelling of a recent study of generational effect of low dose and chronic Cadmium intake on the metabolism [96]. In this case study, we proposed a first simple model of the oral glucose tolerance test (OGTT) adapted from a previous glucose response model [97]. We also provided multiple parameter estimations associated to different data sets.

7.2 Future work

The work described in this thesis still needs improvement on several points, but also suggests numerous ideas for future work.

Improvements and implementation. In addition to the theoretical work given in this thesis, we still need to provide an implementation in a user-friendly toolbox. The improvements provided in Chapter 4 can be integrated in the latest version of reachability toolbox `sapo` [83]. Similarly, the current implementation of sparse Krivine-Stengle representations must be re-written in a more efficient implementation than the current one in `Matlab`. The MOEPLA language is still under development, we need to implement an interface and a parser to check the validity of experimental protocols and automatically generate their hybrid automaton models for formal analysis purposes. In addition to the implementation work, we still need to provide a solution to handle common variables, relative time constraints, and a more efficient hybrid automaton generation.

Occupation measure methods. The theoretical results of Chapter 3 for model revision can be improved by providing a converging sequence of relaxations for the optimal control problem with intermediate points instead of using a greedy algorithm. In addition, it has been shown that occupation measure methods [124] allow computing converging over-approximations of the valid parameter sets. These results can be extended to our biological problem of fitting parameters with respect to intermediate time measures.

Non-linear reachability analysis. The work on the Bernstein expansion provided some ideas for its application for conservative approximations of polynomial lifts. Then, such a method can be used within an algorithm for non-linear reachability analysis using Carleman linearisation [170]. Another idea is to use multivariate polynomial interpolation to approximate the image of an initial set by a trajectory at a fixed time.

Glucose response model. The experimental results of Cadmium effect on the glucose response are recent, and in this thesis we presented a preliminary modelling work and parameter estimation of the glucose response model, independently of the Cadmium. We now need to ensure the robustness and validity of the estimated parameters using either statistical methods or set-based methods. Additionally, we still need to consider the whole protocol to design a model of Cadmium integration. The MOEPLA language will help to formally specify this protocol and generate an associated model.

Bibliography

[1] Tommaso Dreossi, Thao Dang, and Carla Piazza. Parallelotope bundles for polynomial reachability. In *Proceedings of the 19th International Conference on Hybrid Systems: Computation and Control*, pages 297–306. ACM, 2016. (*Cited on pages v, vii, 47, 63, 65, 66, 67, 76, and 156.*)

[2] Alexandre Rocca, Victor Magron, and Thao Dang. Certified roundoff error bounds using bernstein expansions and sparse krivine-stengle representations. In *Computer Arithmetic (ARITH), 2017 IEEE 24th Symposium on*, pages 74–81. IEEE, 2017. (*Cited on pages v, vii, 10, 76, 78, 79, and 156.*)

[3] A Bouchnita, A Rocca, E Fanchon, MJ Koury, JM Moulis, and V Volpert. Multi-scale modelling of erythropoiesis and hemoglobin production. *Journal of Inorganic and Organometallic Polymers and Materials*, 26(6):1362–1379, 2016. (*Cited on pages v, vii, 11, 85, and 156.*)

[4] Thao Dang, Eric Fanchon, Marcelo Forets, Victor Magron, and Alexandre Rocca. Occupation measure methods for modelling and analysis of biological hybrid automata. *arXiv preprint arXiv:1710.03158*, 2017. (*Cited on pages v, vii, 10, 11, and 156.*)

[5] Nicolas Mobilia. *Méthodologie semi-formelle pour l'étude de systèmes biologiques: Application à l'homéostasie du fer*. PhD thesis, Université Grenoble Alpes, 2015. (*Cited on pages v, 7, 29, 46, 115, 117, 120, and 156.*)

[6] Alexandre Rocca, Thao Dang, Eric Fanchon, and Jean-Marc Moulis. Application of the reachability analysis for the iron homeostasis study. In *International Workshop on Hybrid Systems Biology*, pages 67–84. Springer, 2016. (*Cited on pages v, viii, 10, 11, 31, and 156.*)

[7] Simone Paoletti, Aleksandar Lj Juloski, Giancarlo Ferrari-Trecate, and René Vidal. Identification of hybrid systems a tutorial. *European journal of control*, 13(2-3):242–260, 2007. (*Cited on page 2.*)

[8] Flavien Caraguel, Anne-Cécile Lesart, François Estève, Boudewijn van der Sanden, and Angélique Stéphanou. Towards the design of a patient-specific virtual tumour. *Computational and mathematical methods in medicine*, 2016, 2016. (*Cited on pages 2 and 31.*)

[9] Garry M Steil, Joseph Murray, Richard N Bergman, and Thomas A Buchanan. Repeatability of insulin sensitivity and glucose effectiveness from the minimal model: implications for study design. *Diabetes*, 43(11):1365–1371, 1994. (*Cited on page 2.*)

[10] A Stéphanou and Vitaly Volpert. Hybrid modelling in biology: a classification review. *Mathematical Modelling of Natural Phenomena*, 11(1):37–48, 2016. (*Cited on page 4.*)

[11] Xiangfang Li, Oluwaseyi Omotere, Lijun Qian, and Edward R Dougherty. Review of stochastic hybrid systems with applications in biological systems modeling and analysis. *EURASIP Journal on Bioinformatics and Systems Biology*, 2017(1):8, 2017. (*Cited on pages 4, 31, and 84.*)

[12] Katrin Kolczyk and Carsten Conradi. Challenges in horizontal model integration. *BMC systems biology*, 10(1):28, 2016. (*Cited on page 4.*)

[13] Kenneth A Johnson and Roger S Goody. The original michaelis constant: translation of the 1913 michaelis–menten paper. *Biochemistry*, 50(39):8264–8269, 2011. (*Cited on page 4.*)

[14] Ovidiu Radulescu, Alexander N Gorban, Andrei Zinovyev, and Alain Lilienbaum. Robust simplifications of multiscale biochemical networks. *BMC systems biology*, 2(1):86, 2008. (*Cited on page 4.*)

[15] Aleksej K Lopatin. *Oscillations and Dynamical Systems: Normalization Procedures and Averaging*. Weierstraß-Inst. für Angewandte Analysis und Stochastik, 1996. (*Cited on page 4.*)

[16] J Vanlier, CA Tiemann, PAJ Hilbers, and NAW Van Riel. Parameter uncertainty in biochemical models described by ordinary differential equations. *Mathematical biosciences*, 246(2):305–314, 2013. (*Cited on pages 5, 7, and 8.*)

[17] Nicola Bonzanni, K Anton Feenstra, Wan Fokkink, and Elzbieta Krepska. What can formal methods bring to systems biology? In *International Symposium on Formal Methods*, pages 16–22. Springer, 2009. (*Cited on page 6.*)

[18] Patrick Lincoln and Ashish Tiwari. Symbolic systems biology: Hybrid modeling and analysis of biological networks. In *International Workshop on Hybrid Systems: Computation and Control*, pages 660–672. Springer, 2004. (*Cited on pages 6 and 7.*)

[19] Luboš Brim, Milan Češka, and David Šafránek. Model checking of biological systems. In *Formal Methods for Dynamical Systems*, pages 63–112. Springer, 2013. (*Cited on pages 6 and 9.*)

[20] G Wayne Brodland. How computational models can help unlock biological systems. In *Seminars in cell & developmental biology*, volume 47, pages 62–73. Elsevier, 2015. (*Cited on page 6.*)

[21] Werner Dubitzky, Jennifer Southgate, and Hendrik Fuss. *Understanding the dynamics of biological systems: lessons learned from integrative systems biology*. Springer Science & Business Media, 2011. (*Cited on page 6.*)

[22] Kunihiko Kaneko. *Life: an introduction to complex systems biology*. Springer, 2006. (*Cited on page 6.*)

[23] Rajat Singhania, R Michael Sramkoski, James W Jacobberger, and John J Tyson. A hybrid model of mammalian cell cycle regulation. *PLoS computational biology*, 7(2):e1001077, 2011. (*Cited on pages 6 and 7.*)

[24] Thomas R Kiehl, Robert M Mattheyses, and Melvin K Simmons. Hybrid simulation of cellular behavior. *Bioinformatics*, 20(3):316–322, 2004. (*Cited on page 6.*)

[25] P Ye, E Entcheva, SA Smolka, and R Grosu. Modelling excitable cells using cycle-linear hybrid automata. *IET systems biology*, 2(1):24–32, 2008. (*Cited on page 6.*)

[26] Jianxiong Ye, Yuduo Zhang, Enmin Feng, Zhilong Xiu, and Hongchao Yin. Nonlinear hybrid system and parameter identification of microbial fed-batch culture with open loop glycerol input and ph logic control. *Applied Mathematical Modelling*, 36(1):357–369, 2012. (*Cited on pages 6, 7, and 114.*)

[27] Eugene M Izhikevich. Hybrid spiking models. *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 368(1930):5061–5070, 2010. (*Cited on page 6.*)

[28] Pieter J Mosterman and Gautam Biswas. Hybrid automata for modeling discrete transitions in complex dynamic systems. *IFAC Proceedings Volumes*, 31(29):43–48, 1998. (*Cited on page 6.*)

[29] Matthias Althoff, Olaf Stursberg, and Martin Buss. Reachability analysis of nonlinear systems with uncertain parameters using conservative linearization. In *Decision and Control, 2008. CDC 2008. 47th IEEE Conference on*, pages 4042–4048. IEEE, 2008. (*Cited on pages 7 and 9.*)

[30] Eugene Asarin, Thao Dang, and Antoine Girard. Hybridization methods for the analysis of nonlinear systems. *Acta Informatica*, 43(7):451–476, 2007. (*Cited on page 7.*)

[31] Jiri Petrzela. On the piecewise-linear approximation of the polynomial chaotic dynamics. In *Telecommunications and Signal Processing (TSP), 2011 34th International Conference on*, pages 319–323. IEEE, 2011. (*Cited on page 7.*)

[32] Thao Dang, Oded Maler, and Romain Testylier. Accurate hybridization of nonlinear systems. In *Proceedings of the 13th ACM international conference on Hybrid systems: computation and control*, pages 11–20. ACM, 2010. (*Cited on pages 7 and 47.*)

[33] Romain Testylier and Thao Dang. Nltoolbox: A library for reachability computation of nonlinear dynamical systems. In *International Symposium on Automated Technology for Verification and Analysis*, pages 469–473. Springer, 2013. (*Cited on pages 7 and 9.*)

[34] Nacim Ramdani, Nacim Meslem, and Yves Candau. Reachability of uncertain nonlinear systems using a nonlinear hybridization. *Lecture Notes in Computer Science*, 4981:415–428, 2008. (*Cited on pages 7 and 9.*)

[35] Vincent Noël, Sergey Vakulenko, and Ovidiu Radulescu. A hybrid mammalian cell cycle model. *EPTCS 125*, page 68. (*Cited on page 7.*)

[36] Vincent Noël, Dima Grigoriev, Sergey A Vakulenko, and Ovidiu Radulescu. Hybrid models of the cell cycle molecular machinery. In *Hybrid Systems and Biology (HSB 2012)*, volume 92, pages 88–105, 2012. (*Cited on page 7.*)

[37] Jasha Sommer-Simpson, John Reinitz, Leonid Fridlyand, Louis Philipson, and Ovidiu Radulescu. Hybrid reductions of computational models of ion channels coupled to cellular biochemistry. In *International Conference on Computational Methods in Systems Biology*, pages 273–288. Springer, 2016. (*Cited on page 7.*)

[38] Riccardo Porreca, Samuel Drulhe, Hidde de Jong, and Giancarlo Ferrari-Trecate. Structural identification of piecewise-linear models of genetic regulatory networks. *Journal of Computational Biology*, 15(10):1365–1380, 2008. (*Cited on page 7.*)

[39] Kazuyuki Aihara and Hideyuki Suzuki. Theory of hybrid dynamical systems and its applications to biological and medical systems, 2010. (*Cited on page 7.*)

[40] Rafal Goebel, Ricardo G Sanfelice, and Andrew R Teel. Hybrid dynamical systems. *IEEE Control Systems*, 29(2):28–93, 2009. (*Cited on pages 7 and 20.*)

[41] Arjan J Van Der Schaft and Johannes Maria Schumacher. *An introduction to hybrid dynamical systems*, volume 251. Springer London, 2000. (*Cited on pages 7 and 20.*)

[42] Marie Csete and John Doyle. The mathematician's control toolbox for management of type 1 diabetes. *Interface focus*, 4(5):20140042, 2014. (*Cited on page 7.*)

[43] Eva Balsa-Canto, David Henriques, Attila Gábor, and Julio R Banga. Amigo2, a toolbox for dynamic modeling, optimization and control in systems biology. *Bioinformatics*, 32(21):3357–3359, 2016. (*Cited on page 7.*)

[44] Laurence Calzone, François Fages, and Sylvain Soliman. Biocham: an environment for modeling biological systems and formalizing experimental knowledge. *Bioinformatics*, 22(14):1805–1807, 2006. (*Cited on pages 7 and 8.*)

[45] Nicolas Mobilia, Alexandre Donzé, Jean Marc Moulis, and Eric Fanchon. A model of the cellular iron homeostasis network using semi-formal methods for parameter space exploration. *arXiv preprint arXiv:1208.3851*, 2012. (*Cited on pages 7, 11, and 132.*)

[46] Szymon Stoma, Alexandre Donzé, François Bertaux, Oded Maler, and Gregory Batt. Stl-based analysis of trail-induced apoptosis challenges the notion of type i/type ii cell line classification. *PLoS computational biology*, 9(5):e1003056, 2013. (*Cited on page 7.*)

[47] Alexandre Donzé. Breach, a toolbox for verification and parameter synthesis of hybrid systems. In *CAV*, volume 10, pages 167–170. Springer, 2010. (*Cited on pages 7, 11, 29, 49, and 84.*)

[48] Alexandre Donzé. *Trajectoires pour la vérification et la commande de systémes continus et hybrides*. PhD thesis, Université Joseph Fourier (Grenoble), 2007. (*Cited on page 7.*)

[49] Konstantinos Koutroumpas, Eugenio Cinquemani, Panagiotis Kouretas, and John Lygeros. Parameter identification for stochastic hybrid systems using randomized optimization: A case study on subtilin production by bacillus subtilis. *Nonlinear Analysis: Hybrid Systems*, 2(3):786–802, 2008. (*Cited on page 7.*)

[50] Julio R Banga. Optimization in computational systems biology. *BMC systems biology*, 2(1):47, 2008. (*Cited on page 7.*)

[51] Faraz Hussain, Sumit K Jha, Susmit Jha, and Christopher J Langmead. Parameter discovery in stochastic biological models using simulated annealing and statistical model checking. *International Journal of Bioinformatics Research and Applications 2*, 10(4-5):519–539, 2014. (*Cited on page 7.*)

[52] Christian Schilling, Sergiy Bogomolov, Thomas A Henzinger, Andreas Podelski, and Jakob Ruess. Adaptive moment closure for parameter inference of biochemical reaction networks. *Biosystems*, 149:15–25, 2016. (*Cited on page 7.*)

[53] Tina Toni, David Welch, Natalja Strelkowa, Andreas Ipsen, and Michael PH Stumpf. Approximate bayesian computation scheme for parameter inference and model selection in dynamical systems. *Journal of the Royal Society Interface*, 6(31):187–202, 2009. (*Cited on page 7.*)

[54] Katalin Csilléry, Michael GB Blum, Oscar E Gaggiotti, and Olivier François. Approximate bayesian computation (abc) in practice. *Trends in ecology & evolution*, 25(7):410–418, 2010. (*Cited on page 7.*)

[55] Sergiy Bogomolov, Christian Schilling, Ezio Bartocci, Gregory Batt, Hui Kong, and Radu Grosu. Abstraction-based parameter synthesis for multiaffine systems. In *Haifa Verification Conference*, pages 19–35. Springer, 2015. (*Cited on page 8.*)

[56] Sergiy Bogomolov, Christian Schilling, Ezio Bartocci, Gregory Batt, Andreas Podelski, and Radu Grosu. Spacerover: Parameter synthesis for multiaffine systems beyond rovergene. *To appear*, 2015. (*Cited on page 8.*)

[57] Lubos Brim, Jana Fabriková, Sven Drazan, and David Safranek. Reachability in biochemical dynamical systems by quantitative discrete approximation. *arXiv preprint arXiv:1107.5924*, 2011. (*Cited on pages 8, 9, and 47.*)

[58] Estelle Dumas, Johannes Geiselmann, Regis Monte, Pedro T Monteiro, Michel Page, Francøois Rechenmann, and Delphine Ropers. Genetic network analyzer: A tool for the qualitative modeling and simulation of bacterial regulatory networks. (*Cited on page 8.*)

[59] Gregory Batt, Michel Page, Irene Cantone, Gregor Goessler, Pedro Monteiro, and Hidde De Jong. Efficient parameter search for qualitative models of regulatory networks using symbolic model checking. *Bioinformatics*, 26(18):i603–i610, 2010. (*Cited on page 8.*)

[60] Aleksei Fedorovich Filippov. Differential equations with discontinuous right-hand side. *Matematicheskii sbornik*, 93(1):99–128, 1960. (*Cited on pages 8, 9, and 22.*)

[61] Samuel Drulhe, Giancarlo Ferrari-Trecate, and Hidde De Jong. The switching threshold reconstruction problem for piecewise-affine models of genetic regulatory networks. *IEEE Transactions on Automatic Control*, 53(Special Issue):153–165, 2008. (*Cited on pages 8 and 9.*)

[62] Liliana Ironi, Luigi Panzeri, Erik Plahte, and Valeria Simoncini. Dynamics of actively regulated gene networks. *Physica D: Nonlinear Phenomena*, 240(8):779–794, 2011. (*Cited on page 8.*)

[63] Thomas A Henzinger, Peter W Kopke, Anuj Puri, and Pravin Varaiya. What's decidable about hybrid automata? In *Proceedings of the twenty-*

seventh annual ACM symposium on Theory of computing, pages 373–382. ACM, 1995. (*Cited on pages 8 and 50.*)

[64] Thao Dang, Tommaso Dreossi, and Carla Piazza. Parameter synthesis using parallelotopic enclosure and applications to epidemic models. In *International Workshop on Hybrid Systems Biology*, pages 67–82. Springer, 2014. (*Cited on page 8.*)

[65] Tommaso Dreossi and Thao Dang. Parameter synthesis for polynomial biological models. In *Proceedings of the 17th international conference on Hybrid systems: computation and control*, pages 233–242. ACM, 2014. (*Cited on pages 8, 59, 60, 61, 63, 64, 65, and 78.*)

[66] Philipp Rumschinski, Steffen Borchers, Sandro Bosio, Robert Weismantel, and Rolf Findeisen. Set-base dynamical parameter estimation and model invalidation for biochemical reaction networks. *BMC systems biology*, 4(1):69, 2010. (*Cited on pages 8 and 29.*)

[67] Stefan Streif, Anton Savchenko, Philipp Rumschinski, Steffen Borchers, and Rolf Findeisen. Admit: a toolbox for guaranteed model invalidation, estimation and qualitative–quantitative modeling. *Bioinformatics*, 28(9):1290–1291, 2012. (*Cited on page 8.*)

[68] J Hasenauer, Steffen Waldherr, K Wagner, and F Allgöwer. Parameter identification, experimental design and model falsification for biological network models using semidefinite programming. *IET systems biology*, 4(2):119–130, 2010. (*Cited on pages 8 and 32.*)

[69] Stefan Streif, Philipp Rumschinski, Didier Henrion, and Rolf Findeisen. Estimation of consistent parameter sets for continuous-time nonlinear systems using occupation measures and LMI relaxations. In *Decision and Control (CDC), 2013 IEEE 52nd Annual Conference on*, pages 6379–6384. IEEE, 2013. (*Cited on pages 8, 32, 34, and 43.*)

[70] Victor Shia, Ram Vasudevan, Ruzena Bajcsy, and Russ Tedrake. Convex computation of the reachable set for controlled polynomial hybrid systems. In *53rd IEEE Conference on Decision and Control*, pages 1499–1506. IEEE, 2014. (*Cited on pages 8 and 32.*)

[71] François Fages and Sylvain Soliman. Abstract interpretation and types for systems biology. *Theoretical Computer Science*, 403(1):52–70, 2008. (*Cited on page 8.*)

[72] Grégory Batt, Delphine Ropers, Hidde De Jong, Johannes Geiselmann, Radu Mateescu, Michel Page, Dominique Schneider, et al. Analysis and verification of qualitative models of genetic regulatory networks: A model-checking approach. In *IJCAI*, pages 370–375, 2005. (*Cited on pages 9 and 48.*)

[73] Grégory Batt, Hidde De Jong, Michel Page, and Johannes Geiselmann. Symbolic reachability analysis of genetic regulatory networks using discrete abstractions. *Automatica*, 44(4):982–989, 2008. (*Cited on page 9.*)

[74] Grégory Batt, Calin Belta, and Ron Weiss. Model checking genetic regulatory networks with parameter uncertainty. In *HSCC*, volume 7, pages 61–75. Springer, 2007. (*Cited on page 9.*)

[75] Grégory Batt, Calin Belta, and Ron Weiss. Model checking liveness properties of genetic regulatory networks. *Tools and Algorithms for the Construction and Analysis of Systems*, pages 323–338, 2007. (*Cited on page 9.*)

[76] Grégory Batt, Calin Belta, and Ron Weiss. Temporal logic analysis of gene networks under parameter uncertainty. *IEEE Transactions on Automatic Control*, 53(Special Issue):215–229, 2008. (*Cited on page 9.*)

[77] Carla Piazza, Marco Antoniotti, Venkatesh Mysore, Alberto Policriti, Franz Winkler, and Bud Mishra. Algorithmic algebraic model checking i: Challenges from systems biology. In *CAV*, volume 3576, pages 5–19. Springer, 2005. (*Cited on page 9.*)

[78] André Platzer. Differential dynamic logic for verifying parametric hybrid systems. In *Tableaux*, volume 4548, pages 216–232. Springer, 2007. (*Cited on page 9.*)

[79] André Platzer. Differential dynamic logic for hybrid systems. *Journal of Automated Reasoning*, 41(2):143–189, 2008. (*Cited on page 9.*)

[80] André Platzer and Jan-David Quesel. Keymaera: A hybrid theorem prover for hybrid systems (system description). In *International Joint Conference on Automated Reasoning*, pages 171–178. Springer, 2008. (*Cited on pages 9 and 47.*)

[81] Gilles Bernot, Jean-Paul Comet, and El Houssine Snoussi. Formal methods applied to gene network modelling, 2014. (*Cited on page 9.*)

[82] Miguel Carrillo, Pedro A Góngora, and David A Rosenblueth. An overview of existing modeling tools making use of model checking in the analysis of biochemical networks. *Frontiers in plant science*, 3, 2012. (*Cited on page 9.*)

[83] Tommaso Dreossi. Sapo: Reachability computation and parameter synthesis of polynomial dynamical systems. In *Proceedings of the 20th International Conference on Hybrid Systems: Computation and Control*, pages 29–34. ACM, 2017. (*Cited on pages 9, 10, 29, 32, 46, 49, 50, 57, 64, 78, 131, and 133.*)

[84] Thao Dang, Colas Le Guernic, and Oded Maler. Computing reachable states for nonlinear biological models. *Theoretical Computer Science*, 412(21):2095–2107, 2011. (*Cited on page 9.*)

[85] Spring Berman, Ádám Halász, and Vijay Kumar. Marco: a reachability algorithm for multi-affine systems with applications to biological systems. In *International Workshop on Hybrid Systems: Computation and Control*, pages 76–89. Springer, 2007. (*Cited on pages 9 and 47.*)

[86] Romain Testylier and Thao Dang. Analysis of parametric biological models with non-linear dynamics. *arXiv preprint arXiv:1208.3849*, 2012. (*Cited on page 9.*)

[87] Goran Frehse, Colas Le Guernic, Alexandre Donzé, Scott Cotton, Rajarshi Ray, Olivier Lebeltel, Rodolfo Ripado, Antoine Girard, Thao Dang, and Oded Maler. Spaceex: Scalable verification of hybrid systems. In *Computer Aided Verification*, pages 379–395. Springer, 2011. (*Cited on pages 9, 11, 20, 50, 54, 80, 82, 84, 94, and 95.*)

[88] Matthias Althoff. An introduction to cora 2015. In *ARCH@ CPSWeek*, pages 120–151, 2015. (*Cited on pages 9, 11, 50, and 84.*)

[89] Xin Chen, Erika Ábrahám, and Sriram Sankaranarayanan. Flow*: An analyzer for non-linear hybrid systems. In *International Conference on Computer Aided Verification*, pages 258–263. Springer, 2013. (*Cited on pages 9, 11, 47, and 50.*)

[90] Matthias Althoff and Bruce H Krogh. Reachability analysis of nonlinear differential-algebraic systems. *IEEE Transactions on Automatic Control*, 59(2):371–383, 2014. (*Cited on page 9.*)

[91] Olivier Bouissou, Samuel Mimram, Baptiste Strazzulla, and Alexandre Chapoutot. Set-based simulation for design and verification of simulink models. In *Embedded Real Time Software and Systems (ERTS2)*, 2014. (*Cited on page 9.*)

[92] John Maidens and Murat Arcak. Reachability analysis of nonlinear systems using matrix measures. *IEEE Transactions on Automatic Control*, 60(1):265–270, 2015. (*Cited on page 9.*)

[93] Ezio Bartocci and Pietro Lió. Computational modeling, formal analysis, and tools for systems biology. *PLoS computational biology*, 12(1):e1004591, 2016. (*Cited on page 9.*)

[94] Pengcheng Zhao, Shankar Mohan, and Ram Vasudevan. Optimal control for nonlinear hybrid systems via convex relaxations. *arXiv preprint arXiv:1702.04310*, 2017. (*Cited on pages 10, 30, 31, 32, 35, 36, 37, 39, 43, 113, 114, and 131.*)

[95] Victor Magron, Alexandre Rocca, and Thao Dang. Certified roundoff error bounds using bernstein expansions and sparse krivine-stengle representations. *arXiv preprint arXiv:1802.04385*, 2018. (*Cited on page 10.*)

[96] Adeline Jacquet. *Conséquences d'une exposition chronique à des doses modérées de cadmium sur le métabolisme du glucose de rats à différents stades de la vie*. PhD thesis, Grenoble Alpes, 2017. (*Cited on pages 11, 122, 123, 125, and 132.*)

[97] Aparna Nittala, Soumitra Ghosh, Darko Stefanovski, Richard Bergman, and Xujing Wang. Dimensional analysis of minmod leads to definition of the disposition index of glucose regulation and improved simulation algorithm. *Biomedical engineering online*, 5(1):44, 2006. (*Cited on pages 11, 123, 125, and 132.*)

[98] Morris W.. Hirsch and Stephen Smale. *Differential equations, dynamical systems and linear algebra*. Academic Press college division, 1973. (*Cited on page 14.*)

[99] Daniel Liberzon. *Calculus of variations and optimal control theory: a concise introduction*. Princeton University Press, 2012. (*Cited on page 18.*)

[100] Alan C Hindmarsh, Peter N Brown, Keith E Grant, Steven L Lee, Radu Serban, Dan E Shumaker, and Carol S Woodward. Sundials: Suite of nonlinear and differential/algebraic equation solvers. *ACM Transactions on Mathematical Software (TOMS)*, 31(3):363–396, 2005. (*Cited on page 19.*)

[101] Michael S Branicky. Studies in hybrid systems: Modeling, analysis, and control. Technical report, MASSACHUSETTS INST OF TECH CAMBRIDGE LAB FOR INFORMATION AND DECISION SYSTEMS, 1995. (*Cited on page 19.*)

[102] WPMH Heemels, MK Camlibel, AJ Van der Schaft, and JM Schumacher. On the existence and uniqueness of solution trajectories to hybrid dynamical systems. *Nonlinear and hybrid control in automotive applications*, pages 391–422, 2002. (*Cited on page 19.*)

[103] Xin Chen, Stefan Schupp, Ibtissem Ben Makhlouf, Erika Ábrahám, Goran Frehse, and Stefan Kowalewski. A benchmark suite for hybrid systems reachability analysis. In *NASA Formal Methods Symposium*, pages 408–414. Springer, 2015. (*Cited on page 20.*)

[104] Rajeev Alur, Costas Courcoubetis, Nicolas Halbwachs, Thomas A Henzinger, P-H Ho, Xavier Nicollin, Alfredo Olivero, Joseph Sifakis, and Sergio Yovine. The algorithmic analysis of hybrid systems. *Theoretical computer science*, 138(1):3–34, 1995. (*Cited on pages 20 and 21.*)

[105] John Lygeros, Karl Henrik Johansson, Shankar Sastry, and Magnus Egerstedt. On the existence of executions of hybrid automata. In *Decision and Control, 1999. Proceedings of the 38th IEEE Conference on*, volume 3, pages 2249–2254. IEEE, 1999. (*Cited on pages 22 and 23.*)

[106] A Machina, Roderick Edwards, and Pauline van den Driessche. Singular dynamics in gene network models. *SIAM Journal on Applied Dynamical Systems*, 12(1):95–125, 2013. (*Cited on page 22.*)

[107] Vincent Acary, Hidde De Jong, and Bernard Brogliato. Numerical simulation of piecewise-linear models of gene regulatory networks using complementarity systems. *Physica D: Nonlinear Phenomena*, 269:103–119, 2014. (*Cited on page 22.*)

[108] Anna Machina and Arcady Ponosov. Filippov solutions in the analysis of piecewise linear models describing gene regulatory networks. *Nonlinear Analysis: Theory, Methods & Applications*, 74(3):882–900, 2011. (*Cited on page 22.*)

[109] Luca Cardelli, Milan Češka, Martin Fränzle, Marta Kwiatkowska, Luca Laurenti, Nicola Paoletti, and Max Whitby. Syntax-guided optimal synthesis for chemical reaction networks. In *CAV*, 2017. (*Cited on page 30.*)

[110] Ezio Bartocci, Luca Bortolussi, and Laura Nenzi. A temporal logic approach to modular design of synthetic biological circuits. In *CMSB*, 2013. (*Cited on page 30.*)

[111] Vincent Noel, Sergei Vakulenko, and Ovidiu Radulescu. Algorithm for identification of piecewise smooth hybrid systems: application to eukaryotic cell cycle regulation. In *International Workshop on Algorithms in Bioinformatics*, pages 225–236. Springer, 2011. (*Cited on pages 31, 52, and 84.*)

[112] Urszula Ledzewicz and Heinz Schättler. Optimal controls for a model with pharmacokinetics maximizing bone marrow in cancer chemotherapy. *Mathematical biosciences*, 206(2):320–342, 2007. (*Cited on page 31.*)

[113] Erin N Bodine, Louis J Gross, and Suzanne Lenhart. Optimal control applied to a model for species augmentation. *Mathematical biosciences and engineering*, 4(5):669–680, 2008. (*Cited on page 31.*)

[114] Suzanne Lenhart and John T Workman. *Optimal control applied to biological models*. CRC Press, 2007. (*Cited on page 31.*)

[115] Andrew M Pace and Samuel A Burden. Piecewise-differentiable trajectory outcomes in mechanical systems subject to unilateral constraints. In *Proceedings of the 20th International Conference on Hybrid Systems: Computation and Control*, pages 243–252. ACM, 2017. (*Cited on page 31.*)

[116] Yorai Wardi, Magnus Egerstedt, and M Hale. Switched-mode systems: gradient-descent algorithms with armijo step sizes. *Discrete Event Dynamic Systems*, 25(4):571–599, 2015. (*Cited on page 31.*)

[117] Ali Pakniyat and Peter E Caines. On the relation between the minimum principle and dynamic programming for hybrid systems. In *Decision and Control (CDC), 2014 IEEE 53rd Annual Conference on*, pages 19–24. IEEE, 2014. (*Cited on page 31.*)

[118] Jean B Lasserre. Global optimization with polynomials and the problem of moments. *SIAM Journal on Optimization*, 11(3):796–817, 2001. (*Cited on page 31.*)

[119] Richard Vinter. Convex duality and nonlinear optimal control. *SIAM journal on control and optimization*, 31(2):518–538, 1993. (*Cited on page 31.*)

[120] Jean B Lasserre, Didier Henrion, Christophe Prieur, and Emmanuel Trélat. Nonlinear optimal control via occupation measures and lmi-relaxations. *SICON*, 2008. (*Cited on page 31.*)

[121] M Rasheed Abdalmoaty, Didier Henrion, and Luis Rodrigues. Measures and lmis for optimal control of piecewise-affine systems. In *Control Conference (ECC), 2013 European*, pages 3173–3178. IEEE, 2013. (*Cited on pages 31 and 34.*)

[122] Anirudha Majumdar, Ram Vasudevan, Mark M Tobenkin, and Russ Tedrake. Convex optimization of nonlinear feedback controllers via occupation measures. *The International Journal of Robotics Research*, 33(9):1209–1230, 2014. (*Cited on page 32.*)

[123] Milan Korda, Didier Henrion, and Colin N Jones. Controller design and region of attraction estimation for nonlinear dynamical systems. *IFAC*, 2014. (*Cited on page 32.*)

[124] Shankar Mohan and Ram Vasudevan. Convex computation of the reachable set for hybrid systems with parametric uncertainty. In *American Control Conference (ACC), 2016*, pages 5141–5147. IEEE, 2016. (*Cited on pages 32, 43, and 133.*)

[125] Didier Henrion and Milan Korda. Convex computation of the region of attraction of polynomial control systems. *IEEE Transactions on Automatic Control*, 59(2):297–312, 2014. (*Cited on page 32.*)

[126] Luminita Manuela Bujorianu. *Stochastic reachability analysis of hybrid systems*. Springer Science & Business Media, 2012. (*Cited on page 33.*)

[127] Vladimir Igorevich Arnol'd. *Mathematical methods of classical mechanics*, volume 60. Springer Science & Business Media, 2013. (*Cited on page 34.*)

[128] Victor Magron, Pierre-Loïc Garoche, Didier Henrion, and Xavier Thirioux. Semidefinite approximations of reachable sets for discrete-time polynomial systems. *arXiv preprint arXiv:1703.05085*, 2017. (*Cited on pages 34 and 47.*)

[129] Gerald B Folland. *Real analysis: modern techniques and their applications*. John Wiley & Sons, 2013. (*Cited on page 34.*)

[130] Jean Bernard Lasserre. *Moments, positive polynomials and their applications*. World Scientific, 2009. (*Cited on page 34.*)

[131] Roger A Horn and Charles R Johnson. *Matrix analysis*. Cambridge university press, 2012. (*Cited on page 39.*)

[132] Andrew Paul Smith. Fast construction of constant bound functions for sparse polynomials. *Journal of Global Optimization*, 43(2-3):445–458, 2009. (*Cited on pages 47, 59, 60, and 77.*)

[133] Thao Dang and Romain Testylier. Reachability analysis for polynomial dynamical systems using the bernstein expansion. *Reliable Computing*, 17(2):128–152, 2012. (*Cited on pages 47 and 59.*)

[134] Mohamed Amine Ben Sassi. *Analyse et contrôle des systèmes dynamiques polynomiaux*. PhD thesis, Université de Grenoble, 2013. (*Cited on page 47.*)

[135] Grégory Batt, Boyan Yordanov, Ron Weiss, and Calin Belta. Robustness analysis and tuning of synthetic gene networks. *Bioinformatics*, 23(18):2415–2422, 2007. (*Cited on page 47.*)

[136] R Sakthivel, K Mathiyalagan, S Lakshmanan, and Ju H Park. Robust state estimation for discrete-time genetic regulatory networks with randomly occurring uncertainties. *Nonlinear dynamics*, 74(4):1297–1315, 2013. (*Cited on page 48.*)

[137] Alfred J Lotka. Contribution to the theory of periodic reactions. *The Journal of Physical Chemistry*, 14(3):271–274, 1910. (*Cited on page 48.*)

[138] Vito Volterra. Fluctuations in the abundance of a species considered mathematically. *Nature*, 118(2972):558–560, 1926. (*Cited on page 48.*)

[139] Tommaso Dreossi, Thao Dang, Alexandre Donzé, James Kapinski, Xiaoqing Jin, and Jyotirmoy V Deshmukh. Efficient guiding strategies for testing of temporal properties of hybrid systems. In *NASA Formal Methods Symposium*, pages 127–142. Springer, 2015. (*Cited on page 49.*)

[140] Antoine Girard and George J Pappas. Verification using simulation. In *International Workshop on Hybrid Systems: Computation and Control*, pages 272–286. Springer, 2006. (*Cited on page 49.*)

[141] Yashwanth Annpureddy, Che Liu, Georgios Fainekos, and Sriram Sankaranarayanan. S-taliro: A tool for temporal logic falsification for hybrid systems. In *International Conference on Tools and Algorithms for the Construction and Analysis of Systems*, pages 254–257. Springer, 2011. (*Cited on page 49.*)

[142] Alfred Tarski. A decision method for elementary algebra and geometry. In *Quantifier elimination and cylindrical algebraic decomposition*, pages 24–84. Springer, 1998. (*Cited on page 50.*)

[143] George E Collins. Quantifier elimination for real closed fields by cylindrical algebraic decompostion. In *Automata Theory and Formal Languages 2nd GI Conference Kaiserslautern, May 20–23, 1975*, pages 134–183. Springer, 1975. (*Cited on page 50.*)

[144] Henry Fuchs, Zvi M Kedem, and Bruce F Naylor. On visible surface generation by a priori tree structures. In *ACM Siggraph Computer Graphics*, volume 14, pages 124–133. ACM, 1980. (*Cited on page 53.*)

[145] N. Kekatos, M. Forets, and G. Frehse. Constructing verification models of nonlinear simulink systems via syntactic hybridization. In *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, pages 1788–1795, Dec 2017. (*Cited on page 54.*)

[146] Sergei Natanovich Bernstein. Démonstration du théorème de weierstrass fondée sur le calcul des probabilités. *Kharkov Mathematical Society Communication*, 13:1–2, 1912. (*Cited on page 57.*)

[147] O Shisha. The bernstein form of a polynomial. *J Res Natl Bur Stand Math Math Phys B*, 70:79, 1966. (*Cited on page 57.*)

[148] Jürgen Garloff. Convergent bounds for the range of multivariate polynomials. In *Interval Mathematics 1985*, pages 37–56. Springer, 1986. (*Cited on pages 57 and 59.*)

[149] Rida T Farouki. The bernstein polynomial basis: A centennial retrospective. *Computer Aided Geometric Design*, 29(6):379–419, 2012. (*Cited on page 57.*)

[150] Jihad Titi and Jürgen Garloff. Fast determination of the tensorial and simplicial bernstein forms of multivariate polynomials and rational functions. 2017. (*Cited on page 57.*)

[151] Andrew Paul Smith. *Enclosure methods for systems of polynomial equations and inequalities*. PhD thesis, 2012. (*Cited on pages 58 and 59.*)

[152] Jürgen Garloff and Andrew P Smith. Investigation of a subdivision based algorithm for solving systems of polynomial equations. 2000. (*Cited on page 59.*)

[153] Paluri SV Nataraj and M Arounassalame. A new subdivision algorithm for the bernstein polynomial approach to global optimization. *International journal of automation and computing*, 4(4):342–352, 2007. (*Cited on page 59.*)

[154] Bernard Mourrain and Jean Pascal Pavone. Subdivision methods for solving polynomial equations. *Journal of Symbolic Computation*, 44(3):292–306, 2009. (*Cited on page 59.*)

[155] E. de Klerk and M. Laurent. Error Bounds for Some Semidefinite Programming Approaches to Polynomial Minimization on the Hypercube. *SIAM J. on Optimization*, 20(6):3104–3120, 2010. (*Cited on page 59.*)

[156] Willi-Hans Steeb and Yorick Hardy. *Matrix calculus and Kronecker product: a practical approach to linear and multilinear algebra*. World Scientific, 2011. (*Cited on page 60.*)

[157] Jihad Titi and Jürgen Garloff. Matrix methods for the simplicial bernstein representation and for the evaluation of multivariate polynomials. *Applied Mathematics and Computation*, 315:246–258, 2017. (*Cited on page 60.*)

[158] A. Narkawicz, J. Garloff, A. Smith, and C. Muñoz. Bounding the Range of a Rational Function over a Box. *Reliable Computing*, 17:34–39, 2012. (*Cited on page 63.*)

[159] Jürgen Garloff and Tareq Hamadneh. Convergence and inclusion isotonicity of the tensorial rational bernstein form. In Marco Nehmeier, Jürgen Wolff von Gudenberg, and Warwick Tucker, editors, *Scientific Computing, Computer Arithmetic, and Validated Numerics*, pages 171–179. Springer International Publishing, 2016. (*Cited on page 64.*)

[160] David Grimm, Tim Netzer, and Markus Schweighofer. A note on the representation of positive polynomials with structured sparsity. *Archiv der Mathematik*, 89(5):399–403, 2007. (*Cited on pages 70, 72, and 73.*)

[161] Jean-Louis Krivine. Anneaux préordonnés. *Journal d'analyse mathématique*, 12(1):307–326, 1964. (*Cited on page 70.*)

[162] Gilbert Stengle. A nullstellensatz and a positivstellensatz in semialgebraic geometry. *Mathematische Annalen*, 207(2):87–97, 1974. (*Cited on page 70.*)

[163] Jean Bernard Lasserre, Kim-Chuan Toh, and Shouguang Yang. A bounded degree sos hierarchy for polynomial optimization. *EURO J. on Comput. Opt.*, pages 1–31, 2015. (*Cited on pages 70 and 71.*)

[164] Tillmann Weisser, Jean Bernard Lasserre, and Kim-Chuan Toh. Sparse-bsos: a bounded degree sos hierarchy for large scale polynomial optimization with sparsity. *Mathematical Programming Computation*, pages 1–32, 2017. (*Cited on pages 72, 78, and 80.*)

[165] Ahron Ben-Tal and Arkadi Nemirovski. *Lectures on modern convex optimization: analysis, algorithms, and engineering applications*, volume 2. Siam, 2001. (*Cited on page 77.*)

[166] Lorant Porkolab and Leonid Khachiyan. On the complexity of semidefinite programs. *Journal of Global Optimization*, 10(4):351–365, 1997. (*Cited on page 77.*)

[167] Victor Magron, George Constantinides, and Alastair Donaldson. Certified roundoff error bounds using semidefinite programming. *ACM Transactions on Mathematical Software (TOMS)*, 43(4):34, 2017. (*Cited on page 77.*)

[168] Brett N Ryland and Hans Z Munthe-Kaas. On multivariate chebyshev polynomials and spectral approximations on triangles. In *Spectral and High Order Methods for Partial Differential Equations*, pages 19–41. Springer, 2011. (*Cited on page 81.*)

[169] Andreas Klimke. Sparse grid interpolation toolbox. *Universität Stuttgart*, 2008. (*Cited on page 81.*)

[170] Marcelo Forets and Amaury Pouly. Explicit error bounds for carleman linearization. *arXiv preprint arXiv:1711.02552*, 2017. (*Cited on pages 82 and 133.*)

[171] Sergiy Bogomolov, Marcelo Forets, Goran Frehse, Andreas Podelski, Christian Schilling, and Frédéric Viry. Reach set approximation through decomposition with low-dimensional sets and high-dimensional matrices. *arXiv preprint arXiv:1801.09526*, 2018. (*Cited on page 82.*)

[172] Larisa N Soldatova, Wayne Aubrey, Ross D King, and Amanda Clare. The exact description of biomedical protocols. *Bioinformatics*, 24(13):i295–i303, 2008. (*Cited on page 84.*)

[173] Alvis Brazma, Pascal Hingamp, John Quackenbush, Gavin Sherlock, Paul Spellman, Chris Stoeckert, John Aach, Wilhelm Ansorge, Catherine A Ball, Helen C Causton, et al. Minimum information about a microarray experiment (miame)—toward standards for microarray data. *Nature genetics*, 29(4):365–371, 2001. (*Cited on page 84.*)

[174] Marcel Schilling, Andrea C Pfeifer, Sebastian Bohl, and Ursula Klingmüller. Standardizing experimental protocols. *Current Opinion in Biotechnology*, 19(4):354–359, 2008. (*Cited on page 84.*)

[175] Vaishnavi Ananthanarayanan and William Thies. Biocoder: A programming language for standardizing and automating biology protocols. *Journal of biological engineering*, 4(1):13, 2010. (*Cited on page 84.*)

[176] Michael Hucka, Andrew Finney, Herbert M Sauro, Hamid Bolouri, John C Doyle, Hiroaki Kitano, Adam P Arkin, Benjamin J Bornstein, Dennis Bray, Athel Cornish-Bowden, et al. The systems biology markup language (sbml): a medium for representation and exchange of biochemical network models. *Bioinformatics*, 19(4):524–531, 2003. (*Cited on page 84.*)

[177] www.integrativebioinformatics.com/textbook/index.html, 2017. (*Cited on page 84.*)

[178] Alessandro Abate, Luca Cardelli, Marta Kwiatkowska, Luca Laurenti, and Boyan Yordanov. Experimental biological protocols with formal semantics. *arXiv preprint arXiv:1710.08016*, 2017. (*Cited on page 85.*)

[179] Mark J Koury and Maurice C Bondurant. Maintenance by erythropoietin of viability and maturation of murine erythroid precursor cells. *Journal of cellular physiology*, 137(1):65–74, 1988. (*Cited on pages 91, 100, and 107.*)

[180] MJ Koury, ST Sawyer, and MC Bondurant. Splenic erythroblasts in anemia-inducing friend disease: a source of cells for studies of erythropoietin-mediated differentiation. *Journal of cellular physiology*, 121(3):526–532, 1984. (*Cited on page 100.*)

[181] Stephen T Sawyer and Sanford B Krantz. Transferrin receptor number, synthesis, and endocytosis during erythropoietin-induced maturation of friend virus-infected erythroid cells. *Journal of Biological Chemistry*, 261(20):9187–9195, 1986. (*Cited on pages 100 and 102.*)

[182] HD Kim, MJ Koury, SANG JOON Lee, JH Im, and ST Sawyer. Metabolic adaptation during erythropoietin-mediated terminal differentiation of mouse erythroid cells. *Blood*, 77(2):387–392, 1991. (*Cited on page 100.*)

[183] Matthias Schranzhofer, Manfred Schifrer, Javier Antonio Cabrera, Stephan Kopp, Peter Chiba, Hartmut Beug, and Ernst W Müllner. Remodeling the regulation of iron metabolism during erythroid differentiation to ensure efficient heme biosynthesis. *Blood*, 107(10):4159–4167, 2006. (*Cited on page 103.*)

[184] Harald Niederreiter. *Random number generation and quasi-Monte Carlo methods*, volume 63. Siam, 1992. (*Cited on page 105.*)

[185] E. D. Andersen and K. D. Andersen. The Mosek Interior Point Optimizer for Linear Programming: An Implementation of the Homogeneous Algorithm. In Hans Frenk, Kees Roos, Tamás Terlaky, and Shuzhong Zhang, editors,

High Performance Optimization, volume 33 of *Applied Optimization*, pages 197–232. Springer US, 2000. (*Cited on page 113.*)

[186] Li-Jian Lei, Liang Chen, Tai-Yi Jin, Monica Nordberg, and Xiu-Li Chang. Estimation of benchmark dose for pancreatic damage in cadmium-exposed smelters. *Toxicological sciences*, 97(1):189–195, 2007. (*Cited on page 122.*)

[187] Nida Fatima Kolachi, Tasneem Gul Kazi, Hassan Imran Afridi, Naveed Kazi, Sumaira Khan, Ghulam Abbas Kandhro, Abdul Qadir Shah, Jameel Ahmed Baig, Sham Kumar Wadhwa, Faheem Shah, et al. Status of toxic metals in biological samples of diabetic mothers and their neonates. *Biological trace element research*, 143(1):196–212, 2011. (*Cited on page 122.*)

[188] Zhou Zhou, Yong-hui Lu, Hui-feng Pi, Peng Gao, Min Li, Lei Zhang, Li-ping Pei, Xiang Mei, Lin Liu, Qi Zhao, et al. Cadmium exposure is associated with the prevalence of dyslipidemia. *Cellular Physiology and Biochemistry*, 40(3-4):633–643, 2016. (*Cited on page 122.*)

[189] Manuela Salvucci, Zoltan Neufeld, and Philip Newsholme. Mathematical model of metabolism and electrophysiology of amino acid and glucose stimulated insulin secretion: in vitro validation using a β-cell line. *PloS one*, 8(3):e52611, 2013. (*Cited on page 122.*)

[190] Alessandro Bertuzzi, Serenella Salinari, and Geltrude Mingrone. Insulin granule trafficking in β-cells: mathematical model of glucose-induced insulin secretion. *American Journal of Physiology-Endocrinology and Metabolism*, 293(1):E396–E409, 2007. (*Cited on page 122.*)

[191] Hyuk Kang, Kyungreem Han, and MooYoung Choi. Mathematical model for glucose regulation in the whole-body system. *Islets*, 4(2):84–93, 2012. (*Cited on page 123.*)

[192] Elena Breda, Melissa K Cavaghan, Gianna Toffolo, Kenneth S Polonsky, and Claudio Cobelli. Oral glucose tolerance test minimal model indexes of β-cell function and insulin sensitivity. *Diabetes*, 50(1):150–158, 2001. (*Cited on page 123.*)

[193] Elena Breda, Gianna Toffolo, Kenneth S Polonsky, and Claudio Cobelli. Insulin release in impaired glucose tolerance. *Diabetes*, 51(suppl 1):S227–S233, 2002. (*Cited on page 123.*)

[194] Chiara Dalla Man, Marco Campioni, Kenneth S Polonsky, Rita Basu, Robert A Rizza, Gianna Toffolo, and Claudio Cobelli. Two-hour seven-sample oral glucose tolerance test and meal protocol. *Diabetes*, 54(11):3265–3273, 2005. (*Cited on page 123.*)

[195] Chiara Dalla Man, Andrea Caumo, Rita Basu, Robert Rizza, Gianna Toffolo, and Claudio Cobelli. Measurement of selective effect of insulin on glucose disposal from labeled glucose oral test minimal model. *American Journal of Physiology-Endocrinology and Metabolism*, 289(5):E909–E914, 2005. (*Cited on page 123.*)

[196] Chiara Dalla Man, Robert A Rizza, and Claudio Cobelli. Meal simulation model of the glucose-insulin system. *IEEE Transactions on biomedical engineering*, 54(10):1740–1749, 2007. (*Cited on page 123.*)

[197] Peter Y Wielinga, Renate E Wachters-Hagedoorn, Brenda Bouter, Theo H van Dijk, Frans Stellaard, Arie G Nieuwenhuizen, Henkjan J Verkade, and Anton JW Scheurink. Hydroxycitric acid delays intestinal glucose absorption in rats. *American Journal of Physiology-Gastrointestinal and Liver Physiology*, 288(6):G1144–G1149, 2005. (*Cited on page 123.*)

[198] BJ Constable. Changes in blood volume and blood picture during the life of the rat and guinea-pig from birth to maturity. *The Journal of physiology*, 167(2):229–238, 1963. (*Cited on page 124.*)

Formal methods for modelling and validation of biological models

Alexandre Rocca

Thesis Directed By **Thao Dang and Éric Fanchon**

The focus of this thesis is the modelling and analysis of biological systems using formal methods. The dynamics of biological systems exhibit continuous behaviours but also abrupt changes. Ordinary differential equations and hybrid dynamical systems are two mathematical formalisms that naturally model such dynamics.
A crucial aspect of modelling is the determination of valid parameter values that enable to simulate the behaviour and reproduce experimental data sets. If no valid parameter values are found it becomes necessary to revise the model. An option is to replace one or several lumped parameters (parameters which represent a set of processes) by functions of time. In this thesis we first study the model revision problem on hybrid dynamical systems. To this aim we propose a greedy scheme of optimal control methods based on occupation measures and convex relaxations.
Then, we study how to characterize dynamical properties of a model using set-based simulations and reachability analysis. For this purpose, we propose two methods: the first one, which relies on Bernstein expansion, is an extension for hybrid dynamical systems of the reachability tool `Sapo` [1], while the other one uses Krivine-Stengle representations [2] to perform the reachability analysis of polynomial ODEs. Finally, We also propose a methodology to generate hybrid dynamical systems modelling a class of experimental protocols.
The proposed methods are applied to different case studies. We first propose a model of haemoglobin production during the differentiation of an erythrocyte in the bone marrow [3]. To develop this model, we first apply the Monte-Carlo based parameters synthesis, followed by the model revision to correctly fit to the experimental data [4]. We also propose a preliminary study of the effect of low dose Cadmium on glucose response at different steps of a rat growth. Finally, we apply the reachability analysis techniques for the validation on large parameters set of the existing iron homoeostasis model [5], [6]. We note the haemoglobin production process, as well as the glucose response system can be formalised, with their experimental context, as hybrid dynamical systems. Thus, they serve as proof of concept for the methodology of biological experimental protocols modelling.

L'objectif de cette thèse est la modélisation et l'étude de systèmes biologiques par l'intermédiaire de méthodes formelles. Les systèmes biologiques démontrent des comportements continues mais sont aussi susceptibles de montrer des changements abruptes dans leur dynamiques. Les équations différentielles ordinaires, ainsi que les systèmes dynamiques hybrides, sont deux formalismes mathématiques utilisés pour modéliser clairement de tels comportements.
Un point critique de la modélisation de systèmes biologiques est la recherche des valeurs des paramètres du modèle afin de reproduire de manière précise un ensemble de données expérimentales. Si aucun jeux de paramètres valides n'est trouvé, il est nécessaire de réviser le modèle. Une possibilité est alors de remplacer un paramètre, ou un ensemble de paramètres, définissant un processus biologique par une fonction dépendante du temps.
Dans le cadre de cette thèse, nous exposons tout d'abord une méthode pour la révision de modèles hybrides. Pour cela, nous proposons une approche gloutonne appliquée à une méthode de contrôle optimal utilisant les mesures d'occupations et la relaxation convexe. Ensuite, nous étudions comment analyser les propriétés dynamiques d'un modèle à temps discret en utilisant la simulation ensembliste. Dans cet objectif, nous proposons deux méthodes basées sur deux outils mathématiques. La première méthode, qui se repose sur les polynômes de Bernstein, est une extension aux systèmes dynamiques hybrides, de l'outil de calcul ensembliste `Sapo` [1]. La seconde méthode utilise les représentations de Krivine-Stengle [2] pour permettre l'analyse d'atteignaiblité de systèmes dynamiques polynomiaux. Enfin, nous proposons aussi une méthodologie pour générer des systèmes dynamiques hybrides modélisant des protocoles biologiques expérimentaux.
Les méthodes précédemment proposées sont appliquées sur divers études biologiques. Nous étudions tout d'abord un modèle de la production d'hémoglobine durant la différentiation des érythrocytes dans la moelle [3]. Pour permettre la construction de ce modèle, nous avons dans un premier temps généré un ensemble de jeux de paramètres valides à l'aide d'une méthode de type Monte-Carlo. Dans un second temps, nous avons appliqué la méthode de révision de modèle afin de reproduire plus précisément les données expérimentales [4]. Nous proposons aussi un modèle préliminaire des effets à faibles doses du Cadmium sur la réponse du métabolisme à différentes étapes de la vie d'un rat. Enfin, nous appliquons les techniques d'analyse ensembliste pour la validation d'hypothèses sur un modèle d'homéostasie du fer [6] dans le cas où des paramètres varient dans de larges intervalles. Dans cette thèse, nous montrons aussi que le protocole associé à l'étude de la production d'hémoglobine, ainsi que le protocole étudiant l'intégration du Cadmium durant la vie d'un rat, peuvent être formalisés comme des systèmes dynamiques hybrides, et servent ainsi de preuves de concepts pour notre méthode de modélisation de protocoles expérimentaux.