Parameter Synthesis using Parallelotopic Enclosure and Applications to Epidemic Models[*]

Thao Dang[1], Tommaso Dreossi[1,2], and Carla Piazza[2]

[1] VERIMAG
2 avenue de Vignate
38610 Gieres, France
{thao.dang,tommaso.dreossi}@imag.fr
[2] Univerisity of Udine
via delle Scienze 206
33100 Udine, Italy
carla.piazza@uniud.it

Abstract. We consider the problem of refining a parameter set to ensure that the behaviors of a dynamical system satisfy a given property. The dynamics are defined through parametric polynomial difference equations and their Bernstein representations are exploited to enclose reachable sets into parallelotopes. This allows us to achieve more accurate reachable set approximations with respect to previous works based on axis-aligned boxes. Moreover, we introduce a symbolical precomputation that leads to a significant improvement on time performances. Finally, we apply our framework to some epidemic models verifying the strength of the proposed method.

Keywords: Parameter synthesis, Polynomial systems, Bernstein basis, Symbolic computation, Epidemic models.

1 Introduction

This work deals with the following problem: given a dynamical system with uncertain parameters, find a parameter set which guarantees that all the possible simulations of the model satisfy a desired property. The dynamics of the considered system are defined as discrete-time polynomials, the sets reachable by the system are represented with *parallelotopes* (the n-dimensional generalization of parallelograms), the parameter sets are represented by polytopes, and the desired property is specified in terms of a linear inequality.

The technique proposed in this paper advances our previous results on parameter synthesis [1] and reachability analysis of polynomial systems [2,3,4], both based on representation of polynomials in the Bernstein form [5]. Here we introduce a more precise representation of the state of the system and we develop a faster algorithm to synthesize the parameters and compute the reachable

[*] This work is partially supported by Istituto Nazionale di Alta Matematica (INdAM).

sets. The first contribution consists in changing the representation of reachable sets from axis-aligned boxes to parallelotopes. The second relies on the introduction of a precomputation on the dynamics of the system that allows to save calculations during the synthesis of the parameters and the simulation of the model.

We apply our technique on some epidemic models, a class of biological systems representing the evolution of infectious maladies. These models are helpful to make predictions on diseases spread and their study can aid the planning of strategies aimed to reduce the effects of possible future outbreaks. In this context, the parameter synthesis is of particular interest, since it allows to reason on specific plans and their effect on the population. For instance, as we will show in Section 5, particular isolation and treatment policies may have significant effects on the infection and mortality rates.

Parameter synthesis has been considered using various optimization based techniques. Model checking methods [6,7,8] and guided simulation techniques [9] have been proposed to analyze the parameters of stochastic biochemical models and identify parameter values that falsify a property. The closest work to ours are [10,11] in which the parameter sets are represented as boxes and the reachable sets are approximated via sensitivity analysis. The main difference here relies in a more compact representation based on polytopes and in the use of linear constraints in the refinement process.

The paper is organized as follows. In Section 2 we introduce some definitions and we state the problem. Sections 3 and 4 are dedicated to the description of the algorithms to compute the evolution of the system and to synthesize its parameters. In Section 5 we apply our technique on three epidemic models (SIR, SARS and Influenza), showing the improvements of the new technique with respect to the previous. Finally, we conclude in Section 6 with a brief discussion.

2 Preliminaries

We consider a parametric discrete-time dynamical system described by

$$\begin{aligned} \mathbf{x}(k+1) &= f(\mathbf{x}(k), \mathbf{p}) \\ \mathbf{x}(0) &\in X^0 \end{aligned} \tag{1}$$

where $\mathbf{x} \in \mathbb{R}^n$ is the vector of state variables ($\mathbb{R}$ denotes the set of reals), $\mathbf{p} \in P \subseteq \mathbb{R}^m$ is the vector of uncertain parameters, f is a vector of n multi-variate polynomials of the form $f_i : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$ for each $i \in \{1, \dots, n\}$. The set $X^0 \subseteq \mathbb{R}^n$ is called the *initial set.* The set P is called the *initial parameter set.*

Given an initial set X^0, at each step the set of all the states visited by the dynamical system (1) can be computed as the solution of the recursion $X^{j+1} = \{f(\mathbf{x}, \mathbf{p}) \mid \mathbf{x} \in X^j, \mathbf{p} \in P\}$, for $j = 0, 1, \dots, K$. With the notation $\mathcal{R}_P^K$ we emphasize that the reachable set is computed for the fixed parameter set P. This calculation at each step amounts to computing the parametric image of a set through the polynomial f. This is the core problem we address before proceeding to the parameter synthesis algorithm. Let us formally state such image computation problem.

Problem 1 (Image computation). Let $f : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$, $X \subseteq \mathbb{R}^n$ and $P \subseteq \mathbb{R}^m$. We are interested in computing the image of $X \times P$ through f, that is the set

$$f(X, P) = \{(f_1(\mathbf{x}, \mathbf{p}), \dots, f_n(\mathbf{x}, \mathbf{p})) \mid \mathbf{x} \in X, \mathbf{p} \in P\}.$$

We will base the iterative computation of X^j on the Bernstein representation of polynomials [12], which we recall in the following section. Once the image computation problem is solved, we will focus on the problem of constraining the parameter set P so that the resulting system satisfies a safety property.

Problem 2 (Parameters refinement). Let $\mathcal{F} = \{\mathbf{x} \mid s(\mathbf{x}) \geq 0\} \subseteq \mathbb{R}^n$ be an unsafe set where $s : \mathbb{R}^n \to \mathbb{R}$ is a linear constraint over the state variables. We are interested in finding the largest subset $P_s \subseteq P$ such that starting from the initial set X^0, the system does not enter the unsafe set $\mathcal{F}$ up to time K, that is

$$\forall \mathbf{p} \in P_s \; \forall j \in \{0, 1, \dots, K\} \; \forall \mathbf{x} \in \mathcal{R}^j_{P_s} : s(\mathbf{x}) < 0.$$

The Bernstein basis for polynomials

A multi-index is a vector $\mathbf{i} = (i_1, i_2, \dots, i_n)$ where each i_j is a non-negative integer. Given two multi-indexes $\mathbf{i}$ and $\mathbf{d}$, we write $\mathbf{i} \leq \mathbf{d}$ ($\mathbf{d}$ dominates $\mathbf{i}$) if for all $j \in \{1, \dots, n\}$, $i_j \leq d_j$. Also, we write $\mathbf{i}/\mathbf{d}$ for $(i_1/d_1, i_2/d_2, \dots, i_n/d_n)$ and $\binom{\mathbf{d}}{\mathbf{i}}$ for the product of binomial coefficients $\binom{d_1}{i_1}\binom{d_2}{i_2} \dots \binom{d_n}{i_n}$. Moreover, we use $\mathcal{B}^n$ to denote the n-dimensional unit box $[0, 1]^n \subseteq \mathbb{R}^n$.

A parametric polynomial $\rho : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$ can be represented using the power basis as follows:

$$\rho(\mathbf{x}, \mathbf{p}) = \sum_{\mathbf{i} \in I^\rho} \mathbf{a_i}(\mathbf{p}) \mathbf{x}^{\mathbf{i}}$$

where $\mathbf{i} = (i_1, i_2, \dots, i_n)$ is a multi-index of size n and $\mathbf{x}^{\mathbf{i}}$ denotes the monomial $x_1^{i_1} x_2^{i_2} \cdots x_n^{i_n}$. The set I^ρ is called the multi-index set of ρ. The *degree* $\mathbf{d}$ of ρ is the smallest multi-index which dominates all the multi-indexes of I^ρ (i.e., $\forall \mathbf{i} \in I^\rho : \mathbf{i} \leq \mathbf{d}$). The coefficients $\mathbf{a_i}(\mathbf{p})$ are functions of the parameters $\mathbf{p}$ of the form $\mathbb{R}^m \to \mathbb{R}$.

Bernstein basis polynomials of degree $\mathbf{d}$ is a basis for the space of polynomials of degree at most $\mathbf{d}$ over $\mathbb{R}^n$. In particular, for $\mathbf{x} = (x_1, \dots, x_n) \in \mathbb{R}^n$, the $\mathbf{i}^{th}$ Bernstein polynomial of degree $\mathbf{d}$ is defined as $\mathcal{B}_{\mathbf{d},\mathbf{i}}(\mathbf{x}) = \beta_{d_1,i_1}(x_1) \dots \beta_{d_n,i_n}(x_n)$ where for a real number y, $\beta_{d_j,i_j}(y) = \binom{d_j}{i_j} y^{i_j} (1-y)^{d_j - i_j}$. Hence, the polynomial ρ can also be represented using Bernstein basis and it can be written as

$$\rho(\mathbf{x}, \mathbf{p}) = \sum_{\mathbf{i} \in I^\rho} \mathbf{b_i}(\mathbf{p}) \mathcal{B}_{\mathbf{d},\mathbf{i}}(\mathbf{x})$$

where for each $\mathbf{i} \in I^\rho$ the Bernstein coefficient $\mathbf{b_i}(\mathbf{p})$ is

$$\mathbf{b_i}(\mathbf{p}) = \sum_{\mathbf{j} \leq \mathbf{i}} \frac{\binom{\mathbf{i}}{\mathbf{j}}}{\binom{\mathbf{d}}{\mathbf{j}}} \mathbf{a_j}(\mathbf{p}).$$

Bernstein representation is of particular interest due to useful geometric properties of its coefficients. If we refer to the unit box $\mathcal{B}^n$, Bernstein representation can be used to bound ρ, since:

$$\forall \mathbf{x} \in \mathcal{B}^n \ \forall \mathbf{p} \in P: \ \rho(\mathbf{x}, \mathbf{p}) \in [m^\rho, M^\rho] \tag{2}$$

where $m^\rho = min\{\mathbf{b_i}(\mathbf{p}) \mid \mathbf{i} \in I^\rho \ \wedge \ \mathbf{p} \in P\}$ and $M^\rho = max\{\mathbf{b_i}(\mathbf{p}) \mid \mathbf{i} \in I^\rho \ \wedge \ \mathbf{p} \in P\}$. We can see here the advantage of Bernstein representation in the analysis of parametric systems as it succinctly captures the bounds of the reachable set.

3 Image over-approximation

3.1 Parallelotope representations

In [1] a method to step-wise over-approximate the image computation through axis-aligned boxes has been proposed. Here we extend the method to parallelotopes, i.e., the n-dimensional generalizations of parallelepipeds. The use of parallelotopes makes the method more flexible as far as the choice of the initial set X^0 is concerned and it allows to obtain better approximations.

A parallelotope X is a centrally symmetric convex polyhedron that can be described using the *generator representation* as follows.

Definition 1 **($\mathcal{P}_{gen}(\mathbf{q}, G)$).** *Let $G = \{\mathbf{g}_1, \ldots, \mathbf{g}_n\}$ be a set of n linearly independent vectors in $\mathbb{R}^n$ and $\mathbf{q}$ be a point in $\mathbb{R}^n$. The parallelotope X* generated by *G* and *$\mathbf{q}$ is:*

$$X = \mathcal{P}_{gen}(\mathbf{q}, G) = \{\mathbf{q} + \sum_{j=1}^{n} \alpha_j \mathbf{g}_j \mid (\alpha_1, \ldots, \alpha_n) \in \mathcal{B}^n \wedge \mathbf{g}_j \in G\}.$$

The vectors $\mathbf{g}_j$ are called *generators* of the parallelotope and $\mathbf{q}$ is called *base vertex.* Given a set of generators $G = \{\mathbf{g}_1, \ldots, \mathbf{g}_n\}$ and a base vertex $\mathbf{q}$, we will also represent the parallelotope generated by G and $\mathbf{q}$ through the notation

$$X = \mathcal{P}_{gen}(\mathbf{q}, G) = \{\gamma_{(\mathbf{q},G)}(\alpha) \mid \alpha \in \mathcal{B}^n\}$$

where $\alpha = (\alpha_1, \ldots, \alpha_n)$ and $\gamma_{(\mathbf{q},G)}$ is the linear function defined as

$$\gamma_{(\mathbf{q},G)}(\alpha) = \mathbf{q} + \sum_{j=1}^{n} \alpha_j \mathbf{g}_j.$$

Such notation emphasizes the aspect that a parallelotope can be thought as the affine transformation of the unit box $\mathcal{B}^n$.

Definition 2 **($\mathcal{P}_{con}(\Lambda, \mathbf{d})$).** *Let Λ be a $2n \times n$ matrix such that $\Lambda = (\Lambda_j)_{j=1,\ldots,2n}$ and $\forall j \in \{1, \ldots, n\} \Lambda_j = -\Lambda_{j+n}$ and let $\mathbf{d} \in \mathbb{R}^{2n}$. The parallelotope X* generated by *Λ* and *$\mathbf{d}$ is:*

$$X = \mathcal{P}_{con}(\Lambda, \mathbf{d}) = \{\mathbf{x} \mid \Lambda\mathbf{x} \leq d\}.$$

The above representation is called *constraint representation.* The rows of the matrix Λ are called *directions* and the vector $\mathbf{d} = (d_1, \dots, d_{2n}) \in \mathbb{R}^{2n}$ is called *offset.* If the direction matrix Λ is fixed, Λ is called *template matrix.* In this case the paralellotopes are a special case of template polyhedra [13].

Note that Bernstein representation presented in the previous section allows to easily bound the values of a polynomial over the unit-box (see Equation 2). Hence, the generator representation of X, which can be interpreted as a function of $\alpha \in \mathcal{B}^n$, is suitable to over-approximate $f(X, P)$. On the other hand, we will exploit the constraint representation to compute a new parallelotope which over-approximates such image.

Let us now focus on the image computation problem. Let X be a parallelotope represented through its generator representation $X = \mathcal{P}_{gen}(\mathbf{q}, G)$ and Λ be a template matrix. We are interested in computing a parallelotope $X' = \mathcal{P}_{con}(\Lambda, \mathbf{d})$ such that $f(X, P) \subseteq X'$. More concretely, we want to determine the offset $\mathbf{d} \in \mathbb{R}^{2n}$ such that $f(\mathcal{P}_{gen}(\mathbf{q}, G), P) \subseteq \mathcal{P}_{con}(\Lambda, \mathbf{d})$. The following Lemma shows how to determine such an offset $\mathbf{d}$.

Lemma 1. *Let $\mathbf{d} = (d_1, \dots, d_{2n})$ be such that, for each $j \in \{1, \dots, 2n\}$, the inequality $d_j \geq max\{\Lambda_j f(\mathbf{y}, \mathbf{p}) \mid \mathbf{y} \in \mathcal{P}_{gen}(\mathbf{q}, G) \ \wedge \ \mathbf{p} \in P\}$ holds. The inclusion $f(\mathcal{P}_{gen}(\mathbf{q}, G), P) \subseteq \mathcal{P}_{con}(\Lambda, \mathbf{d})$ is guaranteed.*

Exploiting the generator representation, the above condition can be rewritten as $d_j \geq max\{h^j(\alpha, \mathbf{p}) \mid \alpha \in \mathcal{B}^n \ \wedge \ \mathbf{p} \in P\}$, where $h_j(\alpha, \mathbf{p}) = \Lambda_j f(\gamma_{(\mathbf{q},G)}(\alpha), \mathbf{p})$. It is not hard to see that $h_j(\alpha, \mathbf{p})$ is a polynomial function of α and its coefficients are linear functions of the parameters $\mathbf{p}$. Furthermore, the domain we are interested in is exactly the unit box; therefore we can straightforwardly apply Bernstein representation to compute an upper bound of the function $h_j(\alpha, \mathbf{p})$ with $\alpha \in \mathcal{B}^n$.

We denote the set of the Bernstein coefficients of $h_j(\alpha, \mathbf{p})$ as $B^{h_j}(\mathbf{p}) = \{\mathbf{b}_{\mathbf{i}}^{h_j}(\mathbf{p}) \mid \mathbf{i} \in I^{h_j}\}$. Here we write each Bernstein coefficient as a function of $\mathbf{p}$ because they are computed from monomial coefficients which are linearly dependent on the parameters $\mathbf{p}$.

Theorem 1. *Let $\mathbf{d} = (d_1, \dots, d_{2n})$ be such that for each $j \in \{1, \dots, 2n\}$ the component d_j is defined as $d_j = max\{\mathbf{b}_{\mathbf{i}}^{h_j}(\mathbf{p}) \mid \mathbf{i} \in I^{h_j} \ \wedge \ \mathbf{p} \in P\}$. The vector $\mathbf{d}$ satisfies the inclusion $f(\mathcal{P}_{gen}(\mathbf{q}, G), P) \subseteq \mathcal{P}_{con}(\Lambda, \mathbf{d})$.*

3.2 Bounding reachable sets

In order to formalize the algorithm for bounding the reachable set, we rewrite the generator representation of parallelotopes explicitly distinguishing between the directions of the generators and their lengths.

Let $G = \{\mathbf{g}_1, \mathbf{g}_2, \dots, \mathbf{g}_n\}$ be a set of generators. Let $\beta_i \in \mathbb{R}$ be the euclidian norm of $\mathbf{g}_i$ and $\mathbf{u}_i$ be the versor (vector of norm 1) of $\mathbf{g}_i$, i.e., $\mathbf{g}_i = \beta_i \mathbf{u}_i$. Let $\beta = (\beta_1, \beta_2, \dots, \beta_n)$ and $U = \{\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_n\}$. With a slight abuse of notation, the generator representation can be rewritten as $\mathcal{P}_{gen}(\mathbf{q}, \beta, U) = \{\gamma_U(\alpha, \mathbf{q}, \beta) \mid \alpha \in$

$\mathcal{B}^n\}$, where $\gamma_U(\alpha, \mathbf{q}, \beta)$ is the linear function in α defined as

$$\gamma_U(\alpha, \mathbf{q}, \beta) = \mathbf{q} + \sum_{j=1}^{n} \alpha_j \beta_j \mathbf{u}_j.$$

When we work on parallelotopes using the constraint representation we can fix a template matrix Λ and let the offset $\mathbf{d}$ free. In this way we symbolically denote an infinite set of parallelotopes. On the generator representation this corresponds to the choice of a set U of n versors, while the norms β are free.

We focus on a single reachability step: given a parallelotope $X = \mathcal{P}_{gen}(\mathbf{q}, \beta, U)$, we want to compute $\mathbf{q}'$ and β' such that $\mathcal{P}_{gen}(\mathbf{q}', \beta', U)$ over-approximates the set $f(X, P)$. The set $f(X, P)$ can be characterized as

$$f(X, P) = f(\gamma_U(\mathcal{B}^n, \mathbf{q}, \beta), P) = \{f(\gamma_U(\alpha, \mathbf{q}, \beta), \mathbf{p}) \mid \alpha \in \mathcal{B}^n \wedge \mathbf{p} \in P\}.$$

Hence, if we find $\mathbf{q}'$ and β' such that $f(\gamma_U(\mathcal{B}^n, \mathbf{q}, \beta), P) \subseteq \gamma_U(\mathcal{B}^n, \mathbf{q}', \beta')$, we obtain an over-approximation in generator representation of $f(X, P)$. Such $\mathbf{q}'$ and β' can be found passing through an intermediate constraint representation. Let Λ be the template matrix which corresponds to the versor generators U (see Section 3.3), we try to find the offset $\mathbf{d}$ such that $f(\gamma_U(\mathcal{B}^n, \mathbf{q}, \beta), P) \subseteq \mathcal{P}_{con}(\Lambda, \mathbf{d})$ and then we convert the constraint representation $\mathcal{P}_{con}(\Lambda, \mathbf{d})$ to its generator representation $\mathcal{P}_{gen}(\mathbf{q}', \beta', U)$. The offset $\mathbf{d}$ can be calculated exploiting Theorem 1, i.e., for each $j \in \{1, \ldots, 2n\}$, over-approximating $\max\{\Lambda_j(f(\gamma_U(\alpha, \mathbf{q}, \beta), \mathbf{p})) \mid \alpha \in \mathcal{B}^n \wedge \mathbf{p} \in P\}$, task that can be carried out taking advantage of Bernstein representation. Such approach, similarly to the technique described in [1], would require the recomputation of Bernstein coefficients at each reachability step. However, three important aspects have to be pointed out:

1. by definition of generator representation, once that $\mathbf{q}, \beta$, and $\mathbf{p}$ have been chosen, the domain of $\Lambda_j(f(\gamma_U(\alpha, \mathbf{q}, \beta), \mathbf{p}))$ is the unit box $\mathcal{B}^n$, that it is exactly the domain on which Bernstein coefficients satisfy Equation 2;
2. the Bernstein coefficients of the function $\Lambda_j(f(\gamma_U(\alpha, \mathbf{q}, \beta), \mathbf{p}))$ are functions of the form $\mathbf{b}_\mathbf{i}(\mathbf{q}, \beta, \mathbf{p})$ linear in $\mathbf{p}$;
3. both the template matrix Λ and U are fixed, i.e., at each reachability step the directions of the edges of the parallelotopes are the same.

Since the template matrix Λ and U are fixed, we do not need to recompute the Bernestein coefficients of $\Lambda_j(f(\gamma_U(\alpha, \mathbf{q}, \beta), \mathbf{p}))$ at each reachability step but, keeping symbolically the parameters $\mathbf{q}, \beta$, and $\mathbf{p}$, we can compute them only once obtaining a template of Bernstein coefficients that we evaluate at each reachability step. In the following we formalize such idea.

Given the template matrix $\Lambda \in \mathbb{R}^{2n \times n}$, the set of generator versors $U = \{\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_n\} \subseteq \mathbb{R}^n$ and the dynamics $f : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}^n$, Algorithm 1 produces a template of Bernstein coefficients $(\Upsilon_j)_{j=1}^{2n}$ (also called control points), that is a $2n$-dimensional vector of vectors of parametrized Bernstein coefficients of the form $\mathbf{b}_{j,\mathbf{i}}(\mathbf{q}, \beta, \mathbf{p}) : \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}$, where $j \in \{1, \ldots, 2n\}$, and $\mathbf{i} \in I^{f(\gamma_U(\alpha, \mathbf{q}, \beta), \mathbf{p})}$.

Algorithm 1 Building the Bernstein coefficients Template

```
function BuildTemplate($\Lambda, U, f$)
    for $j \in \{1, \ldots, 2n\}$ do
        $h_j \leftarrow \Lambda_j(f(\gamma_U(\alpha, \mathbf{q}, \beta), \mathbf{p}))$
        $\Upsilon_j \leftarrow$BernCoeff($h_i$)
    end for
    return $\Upsilon$
end function
```

Notice that U depends on Λ, i.e., it can be computed from Λ. In the next section we will see how this can be done. However, in Algorithm 1 we are not interested in this technical detail and we pass to the function both Λ and U.

Example 1. Let us consider an example of the well known predator-prey Lotka-Volterra model whose two dynamics are $f_1(\mathbf{x}, \mathbf{p}) = x_1 + x_1(a - x_2)$ and $f_2(\mathbf{x}, \mathbf{p}) = x_2 - x_2(c - 2x_1)$. Choosing the generator versors $U = \{(1.0, 0.0), (0.55, 0.83)\}$ and the symbolic base vertex $\mathbf{q} = (q_1, q_2)$, we obtain the generator function and template matrix

$$\gamma_U(\alpha, \mathbf{q}, \beta) = \begin{pmatrix} q_1 + \beta_1\alpha_1 + \frac{2}{13}(\sqrt{13}\alpha_1\beta_2) \\ q_2 + \frac{3}{13}(\sqrt{13}\alpha_2\beta_2) \end{pmatrix} \qquad \Lambda = \begin{pmatrix} -0.83 & 0.55 \\ 0.00 & -1.00 \\ 0.83 & -0.55 \\ 0.00 & 1.00 \end{pmatrix}$$

that lead to a collection of functions h_j, with $j \in \{1, \ldots, 4\}$, where for instance

$$h_2 = -(q_2 + \frac{3}{13}(\sqrt{13}\alpha_2\beta_2))(2q_1 - c + 2\alpha_1\beta_1 + \frac{4}{13}(\sqrt{13}\alpha_2\beta_2)).$$

Finally, from h_1, h_2, h_3, and h_4 we compute the template coefficients Υ of which, for brevity, we report the first element:

$$\Upsilon_{1,1} = \mathbf{b}_{1,(0,0)}(\mathbf{q}, \beta, p) = \frac{2\sqrt{13}}{13} q_1(a - q_2)(-\frac{3}{13} - \sqrt{13}q_2(c - q_1 2)).$$

At this point, fixed the base vertex $\mathbf{q}$ and the versor norms β, in order to compute an over-approximation of the reachability step $f(X, P) = f(\gamma_U(\mathcal{B}^n, \mathbf{q}, \beta), P)$, it is sufficient to find the maximum of each row ($j = 1, \ldots, 2n$) of the Bernstein coefficients template over the parameter set P. Algorithm 2 formalizes such computation. Each offset d_j of the constraint representation of $f(X, P)$ is derived from the maximum of the j-th row of the Bernstein coefficients template Υ over the parameter set P (Line 3). Finally, the constraint representation $\mathcal{P}_{con}(\Lambda, \mathbf{d})$ of the over-approximation of $f(X, P)$ is converted in generator representation (Line 5). Such conversion (discussed in the next section) computes the essential information to reconstruct the new parallelotope: the new base vertex $\mathbf{q}'$ and the new generator amplitudes β'.

Algorithm 2 Bounding the reachable set from X

```
1: function ReachStep($\mathbf{q}, \beta, \Upsilon, P$)
2:     for $j \in \{1, \ldots, 2n\}$ do
3:         $d_j \leftarrow$ Max($\mathbf{q}, \beta, \Upsilon_j, P$)
4:     end for
5:     return $[\mathbf{q}', \beta'] \leftarrow$ con2gen($\mathcal{P}_{con}(\Lambda, \mathbf{d})$)
6: end function
```

3.3 Representation conversion

We now see how to convert the generator representation of a parallelotope into its constraint representation and vice versa. The switch from generator to constrain representation is useful to compute the best template matrix given a set versors and norms, while the inverse conversion represents the last task in our single step reachability computation (see Algorithm 2, Line 5). The efficiency with which such conversions are performed influences the reachability algorithm performances and thus indirectly the whole parameter synthesis procedure.

From generators to constraints. Given the generator representation of a parallelotope $\mathcal{P}_{gen}(\mathbf{q}, \beta, U)$ we want to find its equivalent constraint representation $\mathcal{P}_{con}(\Lambda, \mathbf{d})$, i.e., we want to define a function such that given $\mathbf{q}$, β, and U allows to compute Λ and d such that $\mathcal{P}_{gen}(\mathbf{q}, \beta, U) = \mathcal{P}_{con}(\Lambda, \mathbf{d})$. Let $\mathbf{q} \in \mathbb{R}^n$ be the base vertex, $U = \{\mathbf{u}_1, \ldots, \mathbf{u}_n\}$ be the generator versors set, where for $i = 1, \ldots, n$ it holds $\mathbf{u}_i \in \mathcal{B}^n$, and $\beta = (\beta_1, \beta_2, \ldots, \beta_n) \in \mathbb{R}^n$. Moreover, let $\mathbf{g}_i = \beta_i \mathbf{u}_i$, for $i = 1, \ldots, n$. As first step we calculate the points $\mathbf{p}_1, \ldots, \mathbf{p}_n$ through which we will traverse the hyperplanes of the constraint representation. Such $\mathbf{p}_i$ are obtained adding to the base vertex the unit vectors $\mathbf{u}_i$, that is $\mathbf{p}_i = \mathbf{q} + \mathbf{u}_i$, for $i = 1, \ldots, n$. Let $\pi_i = \mathbf{a}_i \mathbf{x} + d_i$ be the equation of the hyperplane passing through the points $\mathbf{q}, \mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_{i-1}, \mathbf{p}_{i+1}, \ldots, \mathbf{p}_n$. The equation π_i represents the hyperplane on which lies the i-th facet of the parallelotope. The equation $\pi_{i+n} = \mathbf{a}_{i+n}\mathbf{x} + d_{i+n}$ of the hyperplane parallel to π_i can be found translating the vertexes used to compute π_i by the vector $\mathbf{g}_i$, i.e., π_{i+n} is the hyperplane passing through the points $\mathbf{q} + \mathbf{g}_i, \mathbf{p}_1 + \mathbf{g}_i, \mathbf{p}_2 + \mathbf{g}_i, \ldots, \mathbf{p}_{i-1} + \mathbf{g}_i, \mathbf{p}_{i+1} + \mathbf{g}_i, \ldots, \mathbf{p}_n + \mathbf{g}_i$. Let $\underline{d_i}$ and $\overline{d_i}$ be defined as $\underline{d_i} = \min\{d_i, d_{i+n}\}$ and $\overline{d_i} = \max\{d_i, d_{i+n}\}$. Since π_i and π_{i+n} are parallel, it must hold that $\mathbf{a}_i = \mathbf{a}_{i+n}$. Hence, the portion of the parallelotope included between π_i and π_{i+n} can be characterized by the inequality $\underline{d_i} \leq \mathbf{a}_i \mathbf{x} \leq \overline{d_i}$ which means that the i-th and $(i+n)$-th rows of the template matrix Λ are $\Lambda_i = \mathbf{a}_i$ and $\Lambda_{i+n} = -\mathbf{a}_i$, while the i-th and $(i+n)$-th directions are $d_i = \overline{d_i}$ and $d_{i+n} = -\underline{d_i}$.

From constraints to generators. We now see how to compute the opposite conversion. We first rewrite the inequalities given by the template matrix Λ and the direction $\mathbf{d}$ in from $-d_{n+i} \leq \Lambda_i \leq d_i$, for $i = 1, \ldots, n$. The base vertex $\mathbf{q}$ and the coordinates of the vertex $\mathbf{v}_i$, for $i = 1, \ldots, n$, that lies on the straight line passing trough the i-th generator vector applied to the vertex $\mathbf{q}$, are the solution of the linear systems:

$$\begin{pmatrix} \Lambda_1 \\ \vdots \\ \Lambda_n \end{pmatrix} \mathbf{x} = \begin{pmatrix} -d_{n+1} \\ \vdots \\ -d_{2n} \end{pmatrix} \qquad \begin{pmatrix} \Lambda_1 \\ \vdots \\ \Lambda_i \\ \vdots \\ \Lambda_n \end{pmatrix} \mathbf{x} = \begin{pmatrix} -d_{n+1} \\ \vdots \\ d_i \\ \vdots \\ -d_{2n} \end{pmatrix}$$

Hence, the i-th generator $\mathbf{g}_i$ is the difference between the vertex $\mathbf{v}_i$ and the base vertex $\mathbf{q}$, i.e., $\mathbf{g}_i = \mathbf{v}_i - \mathbf{q}$. Finally, the versor $\mathbf{u}_i$ and the generator norm β_i such that $\mathbf{g}_i = \beta_i \mathbf{u}_i$ are given by $\beta_i = \|\mathbf{g}_i\|$ and $\mathbf{u}_i = \frac{\mathbf{g}_i}{\|\mathbf{g}_i\|}$.

4 Parameter Synthesis

In this work we assume that the safety constraint s is linear in $\mathbf{x}$ and that all the coefficients $\mathbf{a_i}$ of the dynamics f are linear in the parameters $\mathbf{p}$. This assumption allows us to reduce the synthesis problem to a set of linear programs.

To check whether the system does not reach the unsafe set $\mathcal{F}$ we can consider the safety function $\sigma = s(f(\gamma_U(\alpha, \mathbf{q}, \beta), \mathbf{p}))$ and its set of Bernstein coefficients $B^\sigma(\mathbf{p}) = \{\mathbf{b}^\sigma_\mathbf{i}(\mathbf{p}) \mid \forall \mathbf{i} \in I^\sigma\}$. The following is a sufficient condition for the system f to satisfy the safety property s after one step starting from the set represented by $\gamma_U(\mathcal{B}^n, \mathbf{q}, \beta)$:

$$\forall \mathbf{p} \in P \; \forall \mathbf{i} \in I^\sigma : \; \mathbf{b}^\sigma_\mathbf{i}(\mathbf{p}) < 0. \tag{3}$$

Note that since s is a linear function and the parameters $\mathbf{p}$ appear linearly in the dynamics of f, the coefficients in the monomial representation of $sigma$ remain linear in $\mathbf{p}$. This means that the constraints of Equation 3 are linear inequalities over $\mathbf{p}$. Such observation allows us to translate the synthesis problem in the resolution of a linear system of inequalities.

At time $j = 1, \ldots, K$, the parameter set P^j is represented as the solution of the linear system of the form $A^j \mathbf{p} < \mathbf{b}^j$. Before starting the parameter synthesis, we collect in a vector Φ the Bernstein coefficients of the safety function $\sigma = s(f(\gamma_U(\alpha, \mathbf{q}, \beta), \mathbf{p}))$, keeping symbolically the base vertex $\mathbf{q}$, the generator vector amplitudes β, and the parameters $\mathbf{p}$. Each element of Φ is a function $\mathbf{b}^\sigma_\mathbf{i}(\mathbf{q}, \beta, \mathbf{p}) : \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$, for all $\mathbf{i} \in I^\sigma$.

Suppose that the state of the system at the j-th step is described by the set X^j whose base vertex and generator amplitudes are $\mathbf{q}^j$ and β^j. The refinement of the parameter set P^{j-1}, represent by the system $A^{j-1}\mathbf{p} < \mathbf{b}^{j-1}$, with respect to the constraint s consists of the following steps:

1. for all $\mathbf{b}^\sigma_\mathbf{i}(\mathbf{q}, \beta, \mathbf{p}) \in \Phi$ substitute $\mathbf{q}$ and β with $\mathbf{q}^j$ and β^j, respectively. All the functions $\mathbf{b}^\sigma_\mathbf{i}(\mathbf{q}^j, \beta^j, \mathbf{p})$ are now linear in $\mathbf{p}$;
2. build the linear system merging $A^{j-1}\mathbf{p} < b^{j-1}$ with the constraints $\mathbf{b}^\sigma_\mathbf{i}(\mathbf{q}^j, \beta^j, \mathbf{p}) < 0$, for all $\mathbf{i} \in I^\sigma$. We will refer with P^j to such new linear system;
3. check whether P^j has solutions.

If P^j has solutions, i.e., the parameter set is not empty, then the set $X^{j+1} = f(X^j, P^j)$ is safe with respect to the constraint s. If P^j has no solution, then there do not exists parameters values in P^j such that the system can safely evolve, i.e., $\forall \mathbf{p} \in P^j : s(f(X^j, P^j)) \geq 0$.

The whole parameter synthesis is summarized in Algorithm 3. First, the procedure computes the Bernstein coefficient template $\varUpsilon$ and the Bernstein coefficients $\varPhi$ of the safety function (Lines 3,4). Then the algorithm enters in a loop that is iterated until either the maximum number of steps K is reached or the parameter set P^j is empty. Each iteration refines the $(j-1)$-th parameter set thanks to the procedure RefParams which exploits the symbolic coefficients stored in $\varPhi$ (Line 7). Then, if the refined set P^j is not empty, the algorithm performs a safe single reachability step from the state set X^j with the parameter values P^j (Line 9). As result, the function ReachStep returns the base vertex $\mathbf{q}^j$ and the generator amplitudes β^j that are the data needed to represent the new reached set X^j.

Algorithm 3 Parameter synthesis w.r.t. s

```
 1: function ParaSynth($\mathbf{q}^0, \beta^0, U, P^0, K$)
 2:     $\varLambda \leftarrow$ConstraintDirections($\mathbf{q}^0, \beta^0, U$)
 3:     $\varUpsilon \leftarrow$BuildTemplate($\varLambda, U, f$)
 4:     $\varPhi \leftarrow$BernCoeffs($s \circ f \circ \gamma_G$)
 5:     $j \leftarrow 1$
 6:     repeat
 7:         $P^j \leftarrow$RefParams($\mathbf{q}^{j-1}, \beta^{j-1}, P^{j-1}, \varPhi$)
 8:         if $P^j \neq \emptyset$ then
 9:             $[\mathbf{q}^j, \beta^j] \leftarrow$ReachStep($\mathbf{q}^{j-1}, \beta^{j-1}, P^j, \varUpsilon$)
10:         end if
11:         $j \leftarrow j + 1$
12:     until $(j = K) \vee (P^j = \emptyset)$
13:     return $(j, \mathbf{q}^j, \beta^j, P^j)$
14: end function
```

5 Experimental Results

Our case of study focuses on epidemiological models. Such systems are useful to understand the dynamics of infectious diseases and to plan strategies that counter their proliferation. The recent arousal of the influenza strain A(H1NA) in the United States and Mexico [14], or the Severe Acute Respiratory Syndrome (SARS) in southern China [15], are some examples that show the impact of the spread of diseases on our society. Often, vaccines are not available for the entire population, because of disproportioned demand or elevated costs that many countries cannot afford. Therefore, a good contrast strategy based on faithful mathematical models can bring benefits to both the population health and country economics.

5.1 SIR and SARS

Let us consider the basic epidemic SIR model [16] in its continuous and Euler discretized (with time step h) versions:

$$\begin{aligned} \dot{S} &= \ -\beta S I & \qquad S_{k+1} &= \ S_k - (\beta S_k I_k)h \\ \dot{I} &= \ \beta S I - \gamma I & \qquad I_{k+1} &= \ I_k + (\beta S_k I_k - \gamma I_k)h \\ \dot{R} &= \ \gamma I & \qquad R_{k+1} &= \ R_k + (\gamma I)h \end{aligned}$$

In this model, a fixed population $N = S(t) + I(t) + R(t)$ is grouped in three classes: $S(t)$ is the number of individuals not yet infected and susceptible to the disease, $I(t)$ are the individuals who have been infected and who could infect healthy individuals, $R(t)$ are those who have been infected and removed from the system. The parameters β and $1/\gamma$ are the contraction rate of the disease and the mean infective period, respectively.

We now perform two tests choosing different generator sets but with the same contraction rate $\beta = 0.34$, initial mean infective period $\gamma \in [0.05, 0.07]$, time step $h = 1.0$, and maximum number of reachability steps $K = 30$.

For the first test we fix as safety condition the constraint $s(I) = I - 0.64$, that corresponds asking whether or not there are values of γ such that the maximum number of infected individuals stays always below 0.64. We consider a generator set whose vectors are mutually perpendicular and that leads to a box. The generator versor set $U = \{u_1, u_2, u_3\}$, template matrix Λ, and vector amplitudes β are:

$$\begin{aligned} u_1 &= (1,0,0) \\ u_2 &= (0,1,0) \\ u_3 &= (0,0,1) \end{aligned} \qquad \Lambda = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{pmatrix} \qquad \beta = \begin{pmatrix} 0.001 \\ 0.001 \\ 0.001 \end{pmatrix}.$$

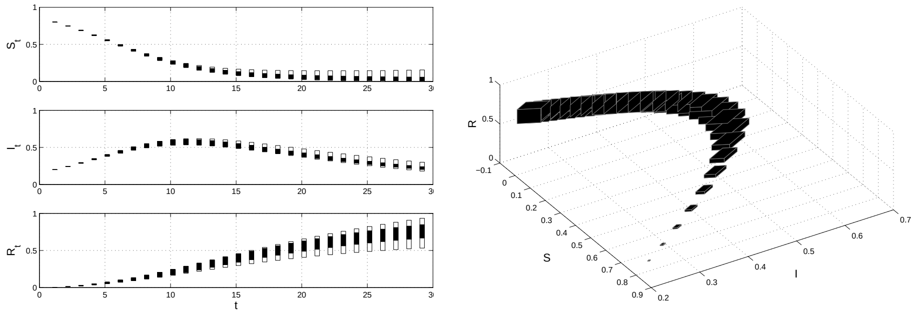
In the second test we strengthen the safety condition down to $s(I) = I - 0.62$. In such a case we try to keep the number of infected individuals always lower than 0.62. Choosing the versor set $U = \{u_1, u_2, u_3\}$, template matrix Λ, and vector amplitudes β with values

$$\begin{aligned} u_1 &= (0.7071, 0.7071, 0) \\ u_2 &= (-0.7071, 0.7071, 0) \\ u_3 &= (0,0,1) \end{aligned} \qquad \Lambda = \begin{pmatrix} 0.7071 & 0.7071 & 0 \\ -0.7071 & 0.7071 & 0 \\ 0 & 0 & 1 \\ -0.7071 & -0.7071 & 0 \\ 0.7071 & -0.7071 & 0 \\ 0 & 0 & -1 \end{pmatrix} \qquad \beta = \begin{pmatrix} 0.0014 \\ 0.0014 \\ 0.0010 \end{pmatrix}$$

we obtain a parallelotope with only two faces parallel to the axis R (those generated by u_3). All the other are not parallel to any axis. In both the tests the initial sets are anchored to the base vertex $\mathbf{q} = (0.8, 0.2, 0)$.

In the first test our tool computed the template Bernstein coefficients in 6.77 seconds and synthesized the safe parameter set $P_s = [0.661, 0.675]$ in 27.95

seconds against the 45.83 seconds of our previous technique. The second test required 13.59 seconds to compute the template control points and 39.70 seconds to synthesized the safe parameter set $P_s = [0.670, 0.675]$.



(a) Time evolution.

(b) Space evolution.

Fig. 1: SIR synthesis with parallelotopes. Black and white boxes represent the constrained and unconstrained evolution of the system, respectively. Note how the edges of the parallelotopes are not parallel to the axis S and I.

Figure 1 depicts the results of the second test. Figure 1a shows the time evolution of the system and Figure 1b its rechability set.

In the spirit of verifying the improvements with respect to our previous approach, we now take into account a more complex and realistic epidemic model that describes the *Severe Acute Respiratory Syndrome*, commonly know as SARS [17]. The dynamics of this model are nonlinear and involve six variables and fifteen parameters. For a detailed description of the model the reader may refer to [1]. As in the previous comparison, we choose a generator versor set $U = \{u_1, u_2, \ldots, u_6\}$, template matrix Λ and generator amplitudes β that produce a box, and we fix the base vertex $\mathbf{q} = (6.5, 124.0, 0.0, 1.0, 0.0, 0.0)$. The simulation parameter values are the same as in [1], that is we try to synthesis the four parameters γ_1, γ_2, k_1, and k_2 imposing the safety constraint $s(I) = I - 300$. The precomputation of the Bernstein coefficients took 14.45 seconds, while the parameter synthesis 907.54 seconds against the 2012.82 seconds of our previous technique. Note that for both the SIR and SARS models we have significantly reduced the computational times.

5.2 Influenza

In this section we consider a simplification of the influenza model described in [18]. Such model is a variation of the standard SIR model where two controllable parameters, the antiviral treatment τ and the social distancing d, i.e., the infected individuals who receive the antiviral treatment and the number of contacts per unit time between individuals, are taken into account. The considered population is composed by N individuals grouped in four classes: S is the number of individuals susceptible to the influenza and not infected, I are the individuals infected by the disease, T are those who are under treatment, and R are the recovered patients. The model is defined by the following system of difference equations:

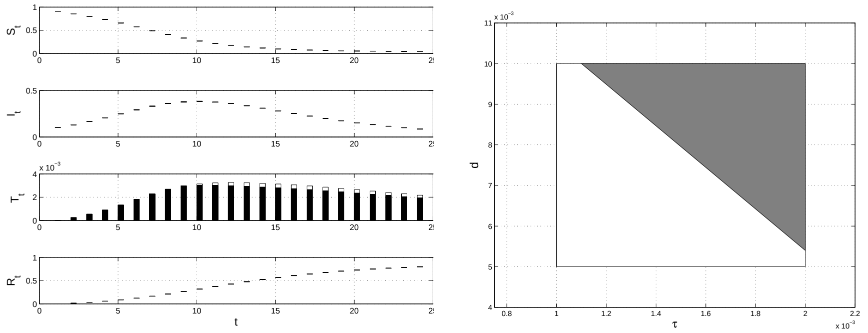
$$\begin{aligned} S_{t+1} =& S_t(1-G_t) \\ I_{t+1} =& (1-\tau)(1-\sigma_1)(1-\delta)I_t + S_t G_t \\ T_{t+1} =& (1-\sigma_2)T_t + \tau(1-\sigma_1)(1-\delta)I_t \\ R_{t+1} =& R_t + \sigma_1(1-\delta)I_t + \sigma_2 T_t \end{aligned}$$

where $G_t = \rho(1-d)(I_t + \varepsilon T_t)/(N_t)$. Variable G_t represents the number of susceptible people that at time t remains so also at time $t+1$. The dynamics of the model involve seven parameters: τ characterizes the fraction of individuals who get the treatment; σ_1 and σ_2 are the probabilities of recovering individuals thanks to natural causes and treatment, respectively; δ is the ratio of induced deaths while β is the disease transmission rate; d represents the social distancing, that is the number of contacts between individuals by unit time, and ρ is the reduction in transmissibility for the treated compartment. The controllable parameters in which we are interested are the antiviral treatment τ and the social distancing d.

We now simulate and study the model trying to synthesize the two controllable parameters. The recovering probabilities without and with treatment are $\sigma_1 = 1/7$ and $\sigma_2 = 1/5$; the transmissibility coefficient of the treated class is $\epsilon = 0.7$; the mortality and susceptibility rates are fixed to $\delta = 8 \times 10^{-5}$ and $\rho = 0.5$. The controllable parameters, that are the antiviral treatment τ and the social distancing d, can vary inside the initial sets $\tau \in [0.001, 0.002]$ and $d \in [0.005, 0.010]$. The imposed safety constraint is $s(I) = I - 0.3964$, while the base vertex $\mathbf{q}$, the generator versos $U = \{u_1, \ldots, u_4\}$, and the vector amplitudes β that generate the initial set are

$$\mathbf{q} = \begin{pmatrix} 0.9 \\ 0.1 \\ 0.0 \\ 0.0 \\ 0.0 \end{pmatrix} \qquad \begin{aligned} u_1 &= (0.7053, 0.7053, 0.7053, 0.0) \\ u_2 &= (0.0, 0.9806, 0.1961, 0.0) \\ u_3 &= (0.0, 0.0, 1.0, 0.0) \\ u_4 &= (0.0, 0.7071, 0.0, 0.7071) \end{aligned} \qquad \beta = \begin{pmatrix} 0.1418 \\ 0.5099 \\ 0.100 \\ 0.1414 \end{pmatrix} \times 10^{-3}.$$

As maximum number of steps we fix $K = 30$. From the initial parameter set $P = [0, 0.001, 0.002] \times [0.005, 0.010]$, our tool found the safe parameter subset $P_s \subset P$ whose vertices are $(0.0011, 0.0100)$, $(0.0020, 0.0054)$, and $(0.0020, 0.0100)$. The graphical representation of P and P_s is depicted in Figure 2b. Figure 2a shows the unconstrained and constrained evolution of the influenza model. The template control points and the parameter synthesis were computed in 92.24 and 304.67 seconds, respectively.



(a) Time evolution.

(b) Synthesized parameter set.

Fig. 2: SITR synthesis.

6 Conclusion

In this work we have introduced an improved parameter synthesis algorithm for polynomial dynamical systems and shown its effectiveness by applying it to some epidemic models. The main advantage of our algorithm is that it can handle a large number of parameters since their refinement can be reduced to linear program solving.

The benefits brought form the proposed advancements are twofold. First, the parallelotope based representation allows a more precise and flexible over-approximation of the states of the considered system. Second, the introduction of the symbolical precomputation on the system dynamics and safety condition avoids the recalculation of the Bernstein coefficients at each synthesis and reachability step. We have seen how multiple evaluations of these precomputed formulæ halve the computational times with respect to our previous approach.

The results obtained from the studies on the epidemic models are encouraging and we intend to pursue this work in several directions. It is our intention to address more complex behavioral specifications in biological systems, expressing constraints that involve logical operators and time-dependent queries. Moreover, we will extend the set representation combining several parallelotopes in each step, hopefully obtaining better approximations and more precise parameter refinements.

References

1. Dreossi, T., Dang, T.: Parameter synthesis for polynomial biological models. In: Proceedings of the 17th International Conference on Hybrid Systems: Computation and Control. HSCC '14, New York, NY, USA, ACM (2014) 233–242
2. Dang, T., Testylier, R.: Reachability analysis for polynomial dynamical systems using the Bernstein expansion. Reliable Computing **17(2)** (2012) 128–152

3. Sassi, M., Testylier, R., Dang, T., Girard, A.: Reachability analysis of polynomial systems using linear programming relaxations. In: ATVA 2012. Volume 7561 of LNCS., Springer (2012) 137–151
4. Testylier, R., Dang, T.: Analysis of parametric biological models with non-linear dynamics. In: Int. Workshop on Hybrid Systems and Biology HSB. Volume 92 of EPTCS. (2012) 16–29
5. Garloff, J., Smith, A.: Rigorous affine lower bound functions for multivariate polynomials and their use in global optimisation. In: Proc. of the 1st Int. Conf. on Applied Operational Research. Volume 1 of Lecture Notes in Management Science. (2008) 199–211
6. Barnat, J., Brim, L., Krejci, A., Streck, A., Safranek, D., Vejnar, M., Vejpustek, T.: On parameter synthesis by parallel model checking. IEEE/ACM Trans. Comput. Biol. Bioinformatics **9**(3) (2012) 693–705
7. Jha, S.K., Langmead, C.J.: Synthesis and infeasibility analysis for stochastic models of biochemical systems using statistical model checking and abstraction refinement. Theor. Comput. Sci. **412**(21) (May 2011) 2162–2187
8. Bartocci, E., Bortolussi, L., Nenzi, L.: On the robustness of temporal properties for stochastic models. In: Hybrid Systems and Biology HSB. Volume 125 of EPTCS. (2013)
9. Dreossi, T., Dang, T.: Falsifying oscillation properties of parametric biological models. In: Int. Workshop on Hybrid Systems and Biology HSB. Volume 125 of EPTCS. (2013) 53–67
10. Donzé, A.: Breach, a toolbox for verification and parameter synthesis of hybrid systems. In: CAV. (2010) 167–170
11. Donzé, A., Clermont, G., Langmead, C.J.: Parameter synthesis in nonlinear dynamical systems: Application to systems biology. Journal of Computational Biology **17**(3) (2010) 325–336
12. Garloff, J., Smith, A.: A comparison of methods for the computation of affine lower bound functions for polynomials. In: Global Optimization and Constraint Satisfaction. LNCS, Springer (2005) 71–85
13. Sankaranarayanan, S., Sipma, H., Manna, Z.: Scalable analysis of linear systems using mathematical programming. In: Verification, Model-Checking and Abstract-Interpretation (VMCAI 2005). LNCS 3385, Springer (2005)
14. Organization, W.H.: Pandemic (h1n1) 2009 - update 103 (June 2010) http://www.who.int/csr/don/2010_06_04/en/.
15. Organization, W.H.: Summary of probable sars cases with onset of illness from 1 november 2002 to 31 july 2003 (December 2013) http://www.who.int/csr/sars/country/table2004_04_21/en/.
16. Kermack, W., McKendrick, A.: A contribution to the mathematical theory of epidemics. Proceedings of the Royal Society Of London: Series A, Physical and Mathematical Sciences **115** (1927) 700–721
17. Gumel, A., Ruan, S., Day, T., Watmough, J., Brauer, F., Driessche, V., Gabrielson, D., Bowman, C., Alexander, M., Ardal, S., et al.: Modelling strategies for controlling sars outbreaks. Proc. of the Royal Society of London. Series B: Biological Sciences **271**(1554) (2004) 2223–2232
18. Parra, P.A.G., Lee, S., Velzquez, L., Castillo-Chavez, C.: A note on the use of optimal control on a discrete time model of influenza dynamics. Mathematical Biosciences and Engineering **8(1)** (2011) 183–197