Optimization-based parameter computation for nonnegative systems to achieve prescribed dynamic behaviour

Balázs Csutak[1] **and Gábor Szederkényi**[1,2]

[1]Pázmány Péter Catholic University, Práter u. 50/a, 1083 Budapest, Hungary
{csutak.balazs, szederkenyi}@itk.ppke.hu

[2]HUN-REN Institute For Computer Science And Control (SZTAKI), Kende u. 13-17, 1111 Budapest, Hungary

Abstract: In this paper, an optimization-based approach is proposed for finding parameter values that guarantee a predefined qualitative behavior for a nonlinear nonnegative dynamical system. Similarly to a model predictive control setup, we consider the model states and parameters as free decision variables and incorporate the system dynamics as well as the desired behavioral properties into the optimization constraints. To show the feasibility of our idea, we carried out three case studies on models with increasing state dimension: the Brusselator, a three-dimensional Lotka-Volterra, and a 4-dimensional epidemic model, where the goal is to induce oscillatory behavior.

Keywords: dynamical systems, nonnegative systems, kinetic systems, oscillations, nonlinear programming

1 Introduction

Nonnegative (also called positive) dynamical systems are widely applied in different fields of science where the modeled quantities (e.g., mass, concentration, pressure, population number or density, object count) are nonnegative [1]. In the essential book [2], the authors write: *"One is tempted to assert that positive systems are the most often encountered systems in almost all areas of science and technology, except electro mechanics ... "*.

An important subset of nonnegative systems is the class of kinetic models (also called chemical reaction networks) the dynamics of which can be formally represented by chemical reactions assuming certain reaction rates. The state variables of such models are the (generalized) concentrations. Chemical reaction network theory (CRNT) is mathematically well-founded and has been continuously developing since the 1970s [3]. It is often possible to transform originally non-kinetic models

This work is dedicated to Prof. Dr. Imre J. Rudas on the occassion of his 75^{th} birthday.

into kinetic form, therefore, reaction networks form a really wide system class which is sometimes called a "prototype of nonlinear science" [4]. This implies that it might be beneficial to write non-chemical models into reaction network form [5], and apply CRNT for analysis or control design [6]. One recent example for that is the kinetic modeling of vehicle traffic flows, where the reacting 'molecules' are vehicles and units of free space on highways [7].

The topic of oscillating (bio)chemical reactions has been an intensively studied field since the discovery of the Belousov-Zhabotinsky reaction [8]. It is now known that chemical oscillators together with other switching subsystems are widely present in living organisms, regulating physiological processes and influencing important cell decisions [9]. The possibility of complex dynamical behaviour is also a central problem in CRNT, where there exist several strong mathematical conditions for this (see, e.g. [10, 11]) However, finding the parameter values that guarantee the desired dynamic properties of the solutions of such a system still remains a complex problem. Numerous classical and novel computational approaches exist for solving these kinds of problems. In [12] a mathematical and chemical approach is presented on the Belousov-Zhabotinskii reaction, aiming to formalize the connection between oscillating reactions and the properties of nonlinear differential equations, and promising a systematic approach for synthesizing new chemical reactions showing oscillatory behavior. Similarly, the authors of [13] study analytically and numerically the transition of two-variable chemical models from stable steady states to oscillatory states.

Due to the rapid development of complex and efficient numerical solvers, several recent approaches for setting the behavior of biochemical systems have been oriented toward simulations and optimizations. One of the most functionally rich solutions is [14], where a multi-level optimization solution is proposed for the automated design of synthetic biological circuits from predefined components. Here, the model parameters belong to the decision variables, and several simulation-based optimization runs are needed to compute a feasible solution.

Another possibility is to introduce compact formalisms, such as different versions of Temporal Logic (Linear Temporal Logic, Signal Temporal Logic, Computational Tree Logic, etc.) for describing the required dynamical behavior. Such solutions are mostly used for model checking, and not as part of the synthesis process itself (see, e.g. [15–19]). There are even machine learning-based approaches for completing or correcting biological models semi-automatically starting from temporal logic formulae [20]. Optimization, however, can be directly used for controller and parameter synthesis as well. Derived manually or translated algorithmically from temporal logic formulae [21], optimization constraints describing the desired model behavior can be directly applied to find the correct parameter values. A temporal logic-based model synthesis was shown for reaction-diffusion networks in [22, 23].

In this paper, we propose an optimization-based method for finding parameter values for nonlinear models that guarantee a predefined complex behavior on a finite horizon. The structure of the paper is the following. In section 2, we give a step-by-step description of this method, with special attention given to manually deriving a constraint set prescribing the desired behavior. In section 3, we present three case

studies, by trying out the algorithm on three models with biochemical backgrounds, and reviewing its capabilities and boundaries as the model complexity increases.

2 Methodology

Our aim is to compute the parameters providing the desired qualitative dynamical properties by creating an appropriate nonlinear optimization problem, having the parameters as decision variables. To achieve this, we apply a strategy motivated by model predictive control (MPC). Along the parameters, all future states of the time-discretized system up to a predefined finite horizon of N steps are considered decision variables, while the system dynamics is introduced in the form of constraints between these variables. The desired behavior is also described using appropriate constraints on the state and input variables, either derived by hand or translated algorithmically from Signal Temporal Logic (STL) formulae. Similarly to the classical MPC framework, we can use the objective function of the optimization problem (typically given in a quadratic form involving the decision variables) to choose the preferred solution from the set of feasible solutions. However, we are mainly interested in finding a feasible solution.

To formalize the approach, let us consider a continuous dynamical system in the form:

$$\dot{x}(t) = f(x(t), p) \tag{1}$$

where the state function f has parameters collected into a vector p. We assume that the parameters are constant in time, which leads to a time-invariant system.

2.1 Discretization

As the first step of the computations, we want to transform the system model into discrete time with sampling time dt which gives the following difference equation:

$$x_{k+1} = F(x_k, p) \tag{2a}$$

$$x_k = x(k \cdot dt) \tag{2b}$$

We consider two well-known classical methods for discretization: the Euler, and the 4-th order Runge-Kutta method.

The Euler method is known for its simplicity, using the first derivative at time instant k to estimate the next state:

$$F(x_k, p) = x_k + dt \cdot f(x_k, p) \tag{3}$$

While having the well-known drawback of quickly cumulating the error in case of larger derivatives, an appropriately small dt can be sufficiently reliable for integrating the differential equation and ensuring that the behavioral constraints are satisfied.

More importantly, due to its simplicity, the Euler discretization does not introduce high-order terms of previous state variables and parameters in a single step.

The 4-th order Runge-Kutta method divides the computation of the next state x_{k+1} into four consecutive steps, averaging the estimated current and future derivatives of state variables:

$$a_k = f(x_k, p) \tag{4a}$$
$$b_k = f(x_k + 0.5 \cdot dt \cdot a_k, p); \tag{4b}$$
$$c_k = f(x_k + 0.5 \cdot dt \cdot b_k, p); \tag{4c}$$
$$d_k = f(x_k + dt \cdot c_k, p); \tag{4d}$$
$$F(x_k, p) = x_k + dt \cdot (a_k + 2 \cdot b_k + 2 \cdot c_k + d_k)/6; \tag{4e}$$

While being a more precise approximation of the derivatives, and therefore generally working with longer dt timesteps, this discretization might be substantially harder to tackle by optimization solvers due to the increased number of dynamical constraints.

2.2 The optimization problem and its solution

To formalize the nonlinear optimization problem, we introduce the notation: $\mathbf{x} = \{x_0,\ldots,x_N\}$, collecting the future state variables for N steps. Using this, we are looking for the parameters p which satisfy:

$$p^* = \arg\min_p J(\mathbf{x}, p) \tag{5a}$$
$$\text{w.r.t. } x_{k+1} = F(x_k, p),\ k = 0 \ldots (N-1) \tag{5b}$$
$$G_x(\mathbf{x}) \leq h_x,\ G_p(p) \leq h_p, \tag{5c}$$

where $J(\mathbf{x}, p) = \mathbf{x}^\intercal P_x \mathbf{x} + p^\intercal P_p p$ is the quadratic cost function with positive definite weighting matrices P_x and P_p, (5b) are equality constraints for incorporating the system dynamics. Moreover, (5c) are additional constraints for the states, inputs, and parameters, respectively, including the nonnegativity constraint for the dynamics. Due to the nonlinear dynamics encoded in (5b), and the generality of the constraints, (5a) - (5c) is a nonlinear programming problem.

Behavioral constraints
There are several possibilities to encode the desired qualitative behavior into the additional constraints (5c). For a simple required behavior, it can be straightforward to manually define the constraints. For example, if the states should converge to 0 after a given finite time, we can define

$$G_x(x_0, x_1, \ldots, x_i, \ldots x_N) = (0\ 0 \ldots 0\ x_i \ldots x_N)^\intercal$$
$$h_x = (0\ 0\ \ldots \varepsilon\ \varepsilon \ldots \varepsilon)^\intercal$$

where ε is sufficiently small.

For more complex behaviors, mathematical formalisms like Signal Temporal Logic can be used. While STL is typically used for behavior checking and validation of dynamical models, there exist algorithms and toolboxes for transforming STL formulae into optimization constraints [21], and there exist applications of MPC-based nonlinear system control using these constraints [24, 25]. It must be noted, however, that the use of these toolboxes is far from being routine: the automatically generated constraint sets often have undue high dimensions, possibly resulting in a practically non-manageable computational problem for the solver. Additionally, STL specifications generally bring integer variables into the optimization task (5a) - (5c), making it a MINLP.

Enforcing oscillatory behavior
In this work, we chose to encode the oscillatory behavior we want to enforce for the examined systems. For this, we introduce the following notations. Let the ith state variable (x_i) at time step k be denoted by x_{ki}. Let the desired oscillation period for x_i be t_i, the oscillation threshold τ_i, and the constraint filling ratio be r_i. We prescribe that the system state variables go above and below the threshold at specific times, and remain there for a time period corresponding to the filling ratio. This dynamic constraint can be written as

$$l = r \cdot t/4 \tag{6a}$$

$$x_{ki} \leq \tau_i - \epsilon \;\; \forall k \;:\; k \cdot dt \in S_1,\; 0 \leq k \leq N \tag{6b}$$

$$S_1 = \bigcup_{j \in \mathbb{N}} \left[t_{0i} + j \cdot t_i - l_i; t_{0i} + j \cdot t_i + l_i \right] \tag{6c}$$

$$x_{ki} \geq \tau_i + \epsilon \;\; \forall k \;:\; k \cdot dt \in S_2,\; 0 \leq k \leq N \tag{6d}$$

$$S_2 = \bigcup_{j \in \mathbb{N}} \left[t_{0i} + \left(j + \tfrac{1}{2}\right) t_i - l_i; t_{0i} + \left(j + \tfrac{1}{2}\right) t_i + l_i \right] \tag{6e}$$

Numerical solution
In our setup, we used `MATLAB R2022b` as the base computation environment, and assembled the optimization problem using `YALMIP` (version 20210331 [26]), and solved it with `BARON 19.3.24` [27]. Baron is a proprietary MINLP solver which applies a polyhedral branch and bound approach, and shows outstanding solution capabilities and excellent performance in complex nonlinear optimization [28, 29]. The experiments were carried out on a Lenovo Thinkpad T590 notebook with an i7-8568U (4 cores, 1.8-4.0 GHz) processor and 16GB RAM.

The obtained results were validated in continuous time using the built-in Ode45 solver of Matlab (using the 4th and 5th order Runge-Kutta method) with default settings (except for the absolute tolerance `AbsTol`=10^{-10}) for simulating the system trajectory given the solver-computed values of the decision variables.

3 Computation results

In order to find the boundaries and illustrate the capabilities of the proposed approach, we carried out three case studies for models with increasing state space di-

mension and number of parameters. Each of the presented nonnegative and kinetic models is capable of showing some kind of oscillatory behavior, and our aim is to find the approprite parameters for it.

3.1 Brussellator

The Brussellator is a classic two-dimensional model, proposed initially by Prigogine and Lefever for describing chemical reaction networks capable of showing complex behavioral patterns [30]. As a starting point, it is widely used for modeling (bio)chemical systems with oscillatory behavior, e.g. the Belousov-Zhabotinsky reaction, chemical reactions regulating circadian clocks, or certein neuron models.

Since many theoretical results are known on the oscillation conditions of general two-dimensional models and specifically on the Brusselator, this case is intended to be an introductory illustration and testing of our approach.

Model and nominal parameters
The model shows the trajectory of the concentrations of two reagents ($\mathbf{B_1}, \mathbf{B_2}$) present in a chemical reaction (the concentration of the remaining materials being constant), and is formally given by the following nonlinear ordinary differential equation (ODE) system in dimensionless form:

$$\dot{\mathbf{B}}_\mathbf{1} = k_1\, a - k_2\, b\, \mathbf{B_1} + k_3\, \mathbf{B_1}^2\, \mathbf{B_2} - k_4\, \mathbf{B_1} \tag{7a}$$

$$\dot{\mathbf{B}}_\mathbf{2} = k_2\, b\, \mathbf{B_1} - k_3\, \mathbf{B_1}^2\, \mathbf{B_2}. \tag{7b}$$

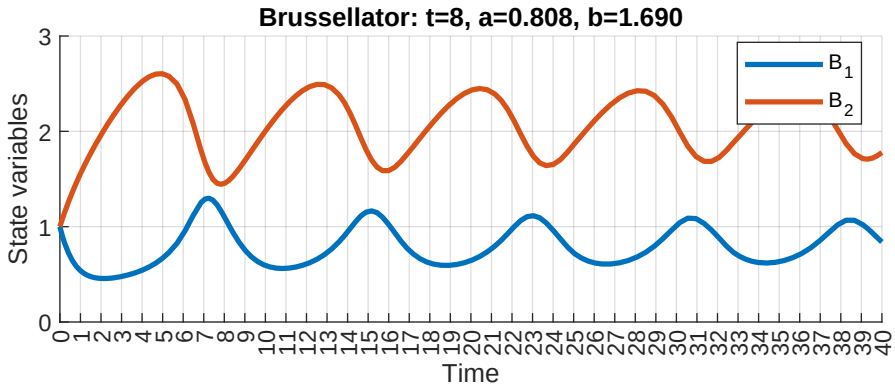
In our study, we use the Brussellator with the simplest nominal set of parameter values, choosing $k_1 = k_2 = k_3 = k_4 = 1$, and consider a and b as unknown parameters to be determined.

Our goal is that the system produces sustained oscillations with predefined period lengths, starting from initial state $x_0 = [1\ 1]^\intercal$. We remark that the necessary and sufficient condition of oscillations is well-known as

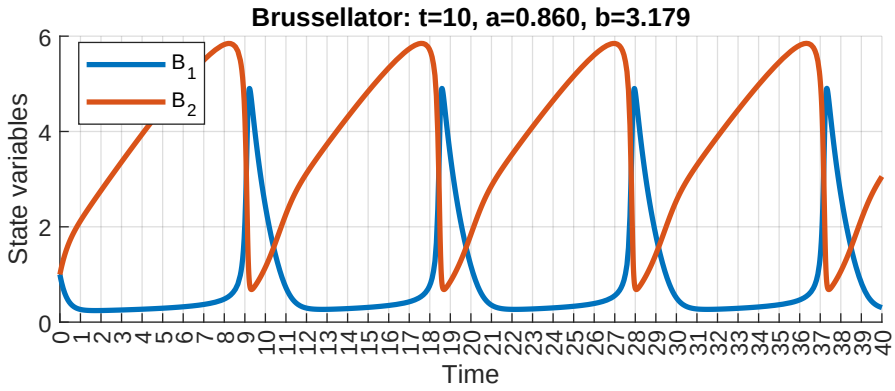
$$a^2 + 1 \leq b. \tag{8}$$

However, we won't use this condition among the constraints.

Optimization parameters and results
In our first experiment, we chose the oscillatory period to be $t = 8$ time units. We discretized the system using the Euler method using $dt = 0.2$ units and chose a time horizon of $T = 40$ units resulting in $N = 200$ steps. We introduced the following periodicity constraints (see, eq. (6)): $t_1 = t_2 = 8$, $t_{01} = 4, t_{02} = 8$, $\epsilon = 0.1$, $r_1 = r_2 = 0.1$. As we did not want to overspecify the problem and decide on a suitable oscillation threshold $\tau_1$and τ_2, we also introduced these as decision variables (i.e., we only want each state variable to have a threshold that periodically outgrows and then goes back below, but we do not specify where that threshold is between the realistic bounds $\epsilon \leq \tau_1, \tau_2 \leq 10$). As we are only looking for a feasible solution (and do not want to choose from them any specific one), we set the cost function J to a



(a)

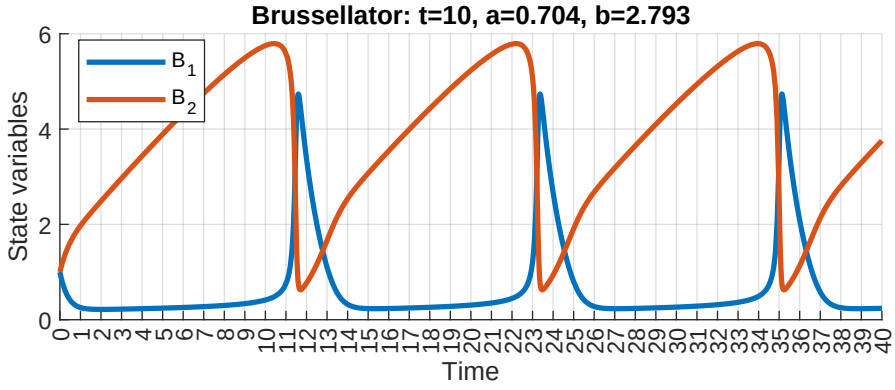


(b)

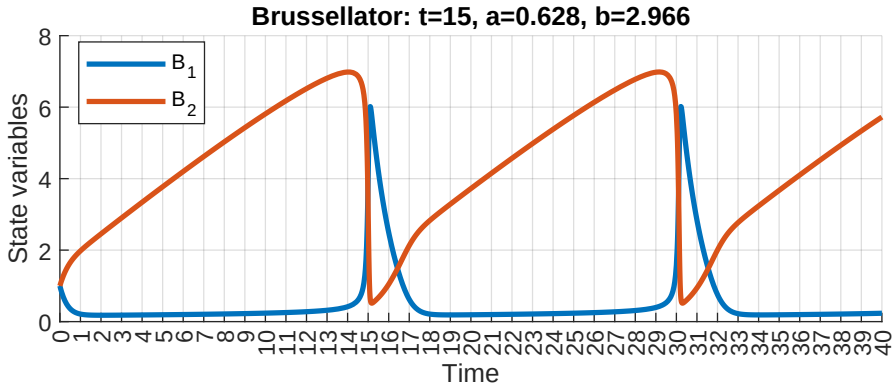
constant value. The problem took approximately 6 seconds for the solver, resulting in parameter values $a = 0.808, b = 1.690$, and trajectories illustrated in Fig. 1a.

Consequently, we repeated the computation for a series of growing periods $t = \{10, 12, 15, 20\}$, using the same approach to derive the constraints, resulting in parameters and trajectories shown in Figs. 1b-1e. The solver times needed for the computation are shown in Table 1.

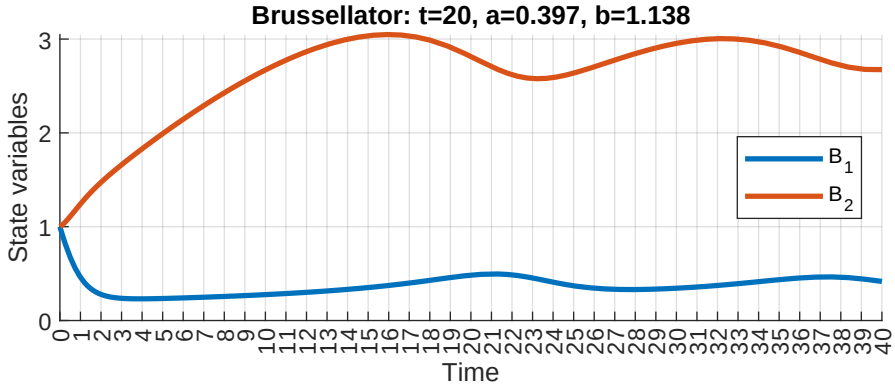
Looking at the results, it can be checked that for cases shown in Figs. 1b-1d, the condition (8) holds, meaning that the system indeed produces stable oscillations with these parameters. In the last case in Fig. 1e, however, the condition is not satisfied, although the prescribed oscillations are fulfilled. Here, we can observe a limitation imposed by the finite horizon $T = 40$ of the optimization being barely able to include two full periods of $t = 20$. Thus, while the system satisfies each constraint and produces oscillations, the trajectories converge to a stable equilibrium in the long term. Essentially the same results were obtained using the Runge-Kutta 4 discretization.



(c)



(d)



(e)

Figure 1
Computed parameters and their simulated trajectories for the Brussellator model

3.2 A food chain model in Lotka-Volterra form

Lotka-Volterra-type (LV) models were originally derived independently for two purposes: modeling (then only hypothetical) chemical reactions in which the concentrations of reagents oscillate, and describing population dynamics of interacting species in a habitat [8, 31]. It is important to mention that generalized Lotka-Volterra models can be considered as universal descriptors of nonlinear dynamics, therefore, their analysis and application reach far beyond population models [32].

Model and nominal parameters
We analyze the system proposed in [33], which models a linear three-species food chain consisting of a low-level prey ($\mathbf{L_1}$), its mid-level predator ($\mathbf{L_2}$), and a high-level predator ($\mathbf{L_3}$) hunting for $\mathbf{L_2}$. The interaction between the species is modeled by the differential equation system below

$$\dot{\mathbf{L}}_1 = a\,\mathbf{L}_1 - b\,\mathbf{L}_1\,\mathbf{L}_2, \tag{9a}$$

$$\dot{\mathbf{L}}_2 = -c\,\mathbf{L}_2 + d\,\mathbf{L}_1\,\mathbf{L}_2 - e\,\mathbf{L}_2\,\mathbf{L}_3, \tag{9b}$$

$$\dot{\mathbf{L}}_3 = -f\,\mathbf{L}_3 + g\,\mathbf{L}_2\,\mathbf{L}_3. \tag{9c}$$

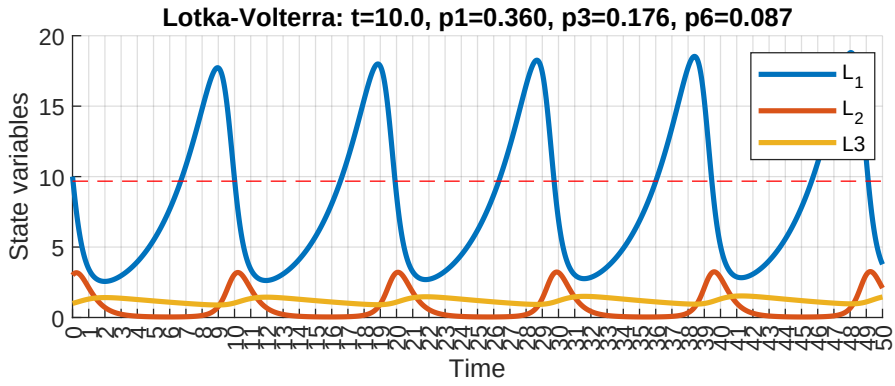
with the following parameters:

- a, c, f representing the natural growth and decay rate of populations $\mathbf{L_1}, \mathbf{L_2}$, and $\mathbf{L_3}$, respectively (without being hunted and without prey being available);
- b, e representing the negative effect of being hunted on populations $\mathbf{L_1}, \mathbf{L_2}$; and
- d, g representing the positive effect of prey being available for populations $\mathbf{L_2}, \mathbf{L_3}$.

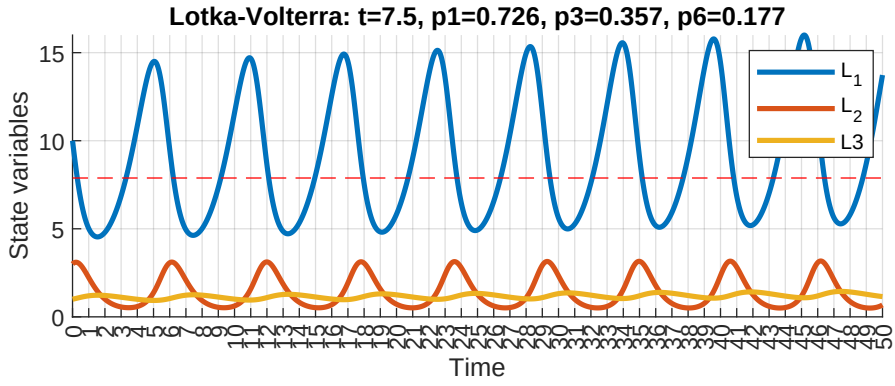
The model is known to show oscillations and periodic behavior for certain parameter combinations, specific behaviors being shown in the case of different parameter regions [33]. For our study, we opted for nominal parameter values: $b = 1/2, e = 1, d = 1/6, g = 1/8$, fixing the effect of interaction between the species, and let the solver find the natural growth and decay rates a, c, f for which the task specification holds. We prescribed (stable) oscillations for the system with predefined period lengths, starting from the initial state $x_0 = [10\ 3\ 1]^\intercal$

Optimization parameters and results
As the Lotka-Volterra model showed serious approximation errors in case of the Euler method, we used different dt values and time horizons in the three case studies, as detailed in Table 1. Similarly to the Brussellator experiment, we manually set the periodicity constraint for the first state variable (substituting Eq. (6) with $t_1 = 10, t_{01} = 4, \epsilon = 0.1, r_1 = 0.1, 1 \leq \tau_1 \leq 20$ and setting J constant. For the second and third state variables no constraints were set.

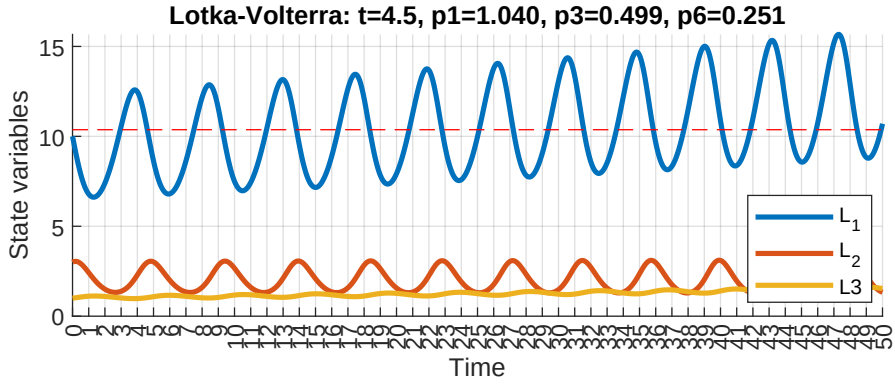
We carried out three computations, with results shown in Figures 2a - 2c. It is visible that the oscillations fulfill the prescribed periods. It can also be seen from Table 1 that the average solution time was the highest in the case of this model.



(a)



(b)



(c)

Figure 2
Computed parameters and corresponding trajectories for the Lotka-Volterra model. The red dashed line shows the computed threshold τ_1.

3.3 An SEIR epidemic model with immunity waning

The SEIR (*Susceptible-Exposed-Infected-Recovered*) is a four-dimensional nonlinear compartmental model widely used for describing various epidemic processes [34]. It divides the population into four compartments, representing the different states of the disease, and people transition from one compartment to another based on their current health condition. Susceptible individuals (in compartment **S**) are assumed to have no protection against the virus and may become ill if exposed to the pathogen. Those in the exposed compartment (**E**) already carry the disease, but can not spread it yet, and transition automatically to the infected group (**I**) after the incubation period. Infected people (**I**) can infect those in **S** and are assumed to recover over time. Recovered individuals (**R**) are assumed to have perfect immunity against the pathogen. Complementing the basic SEIR model, in our case recovered people lose their immunity after a given time, and are moved back into the susceptible compartment. Compartments of the model and the possible transitions can be observed in Fig. 3.

Formally, the dynamics of the model is given by the following differential equations:

$$\dot{\mathbf{S}} = -\beta\,\mathbf{S}\,\mathbf{I}\,/\,\mathbf{N} + \omega\,\mathbf{R}, \tag{10a}$$

$$\dot{\mathbf{E}} = \beta\,\mathbf{S}\,\mathbf{I}\,/\,\mathbf{N} - k_2\,\mathbf{E}, \tag{10b}$$

$$\dot{\mathbf{I}} = k_2\,\mathbf{E} - k_3\,\mathbf{I}, \tag{10c}$$

$$\dot{\mathbf{R}} = k_3\,\mathbf{I} - \omega\,\mathbf{R}. \tag{10d}$$

where **N** is the total population size which is assumed to be constant.

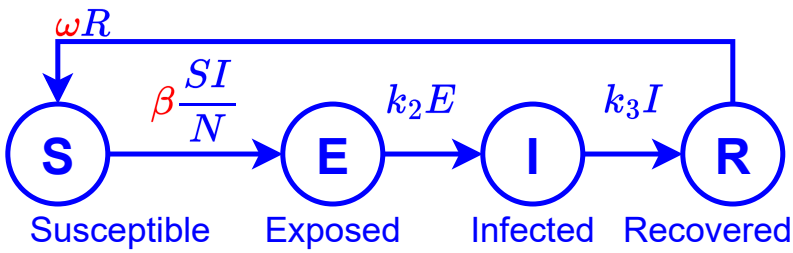
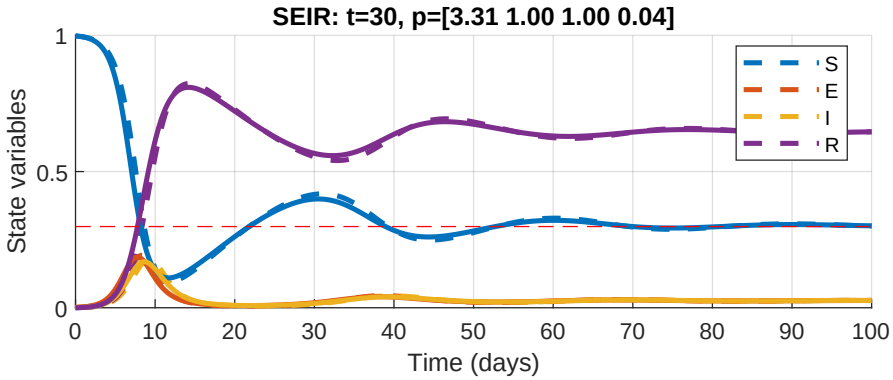


Figure 3
Compartmental transitions of SEIR epidemic model with waning of immunity

The model has four parameters: the transmission rate of the virus (β), the average incubation period (k_2^{-1}), the average time it takes to recover (k_3^{-1}) and the average time the immunity obtained by recovery lasts (ω^{-1}). Thus the parameter vector was defined as $p = [\beta\ k_2\ k_3\ \omega]^T$. Typical values of the parameters, roughly corresponding to the evolution of COVID-19 in Hungary between 15/08/2020 - 15-08-2021 are $\beta \approx 0.1 - 0.7, k_2 \approx 1/2.65, k_3 \approx 1/6.5, \omega \approx 1/150$.

As it has actually been observed during the COVID-19 epidemic, with appropriately selected parameters, the model can show damped oscillations even considering the



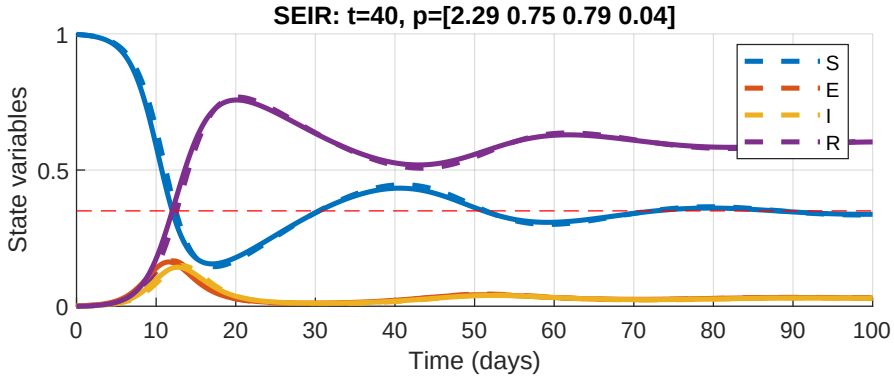
(a)

same virus variant: people moving back and forth between **S** and **R** cause multiple epidemic peaks. There exist proofs for similar systems (e.g., for SIRS models) to be globally asymptotically stable by the construction of appropriate Lyapunov functions [35]. Therefore, we cannot expect sustained oscillations in the solutions. Thus we prescribed that the solutions show oscillations with specific time period without altering the parameters or explicitly specifying the oscillation threshold. Moreover, we wanted to influence the rate of convergence, by prescribing a minimum for the oscillation amplitude at the end of the time horizon.

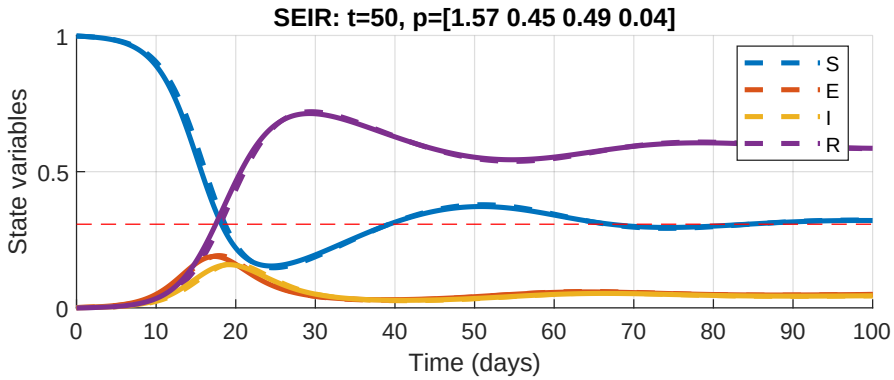
Optimization parameters and results

For each simulation, we discretized the system using $dt = 0.25$ (days). Periodicity constraints were derived for only the first compartment (**S**), by substituting into Eq. (6): $t_{01} = t_1/2, \epsilon_1 = 0.01, r_1 = 0.001, J(\mathbf{x}, p) = 0$, and $t_1 = \{30, 40, 50, 50\}$ (cases (a), (b), (c), (d)). Additionally, we used the following constraints: $0.01 \leq \tau_1 \leq 0.9$, $\beta \leq 5$, $0 \leq k_2, k_3, \omega \leq 1$ to ensure the optimizer remains within the selected model class. In the last simulation (d), to induce stronger oscillations, we increased the parameter value ϵ_1 to 0.025 and the time horizon to $T = 150$ (days).

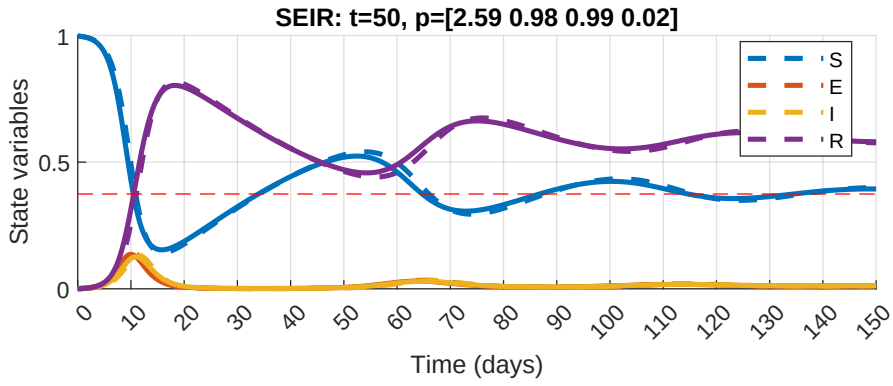
We carried out four experiments, as seen in Figures 4a-4d. In experiments (a)-(c) we computed the parameter values for oscillation periods 30,40 and 50 days respectively, for a time horizon of $T = 100$ days. For the last experiment (d), in order to produce stronger oscillations (with higher amplitude), we repeated case (c) with higher ϵ (i.e., higher deviation from the threshold was required), and a longer horizon. Although the dynamical constraints could be satisfied, the obtained transition rates were (sometimes unrealistically) high, partly caused by the short time horizon. It is also apparent from Table 1 that the computation times were significantly lower than in the case of the Lotka-Volterra model.


SEIR: t=40, p=[2.29 0.75 0.79 0.04]
State variables
Time (days)
S
E
I
R
0
0.5
1
0
10
20
30
40
50
60
70
80
90
100

(b)


SEIR: t=50, p=[1.57 0.45 0.49 0.04]
State variables
Time (days)
S
E
I
R
0
0.5
1
0
10
20
30
40
50
60
70
80
90
100

(c)



(d)

Figure 4
Computed parameters and their simulated trajectories for the SEIR model. Dashed lines represent the trajectories created by the optimizer (using the Euler method), while continuous lines show the simulated trajectories produced by the Ode45 solver using the previously computed parameters.

Table 1
Comparison of the case studies involving three models

Case study	oscillation period (t)	time horizon (T)	sampling time dt	no. of timesteps (N)	free parameters	no. of decision variables	computation time (s)
Brussellator (a)	8						5.23
Brussellator (b)	10						262.72
Brussellator (c)	12	40	0.2	200	a, b	403	21.17
Brussellator (d)	15						3.65
Brussellator (e)	20						29.89
Lotka-Volterra (a)	10	20	0.02	1000		3004	604.74
Lotka-Volterra (b)	7.5	16	0.02	800	a, c, f	2404	385.37
Lotka-Volterra (c)	5	12	0.015	800		2404	185.75
SEIR (a)	30	100		400		1605	4.50
SEIR (b)	40	100	0.25	400	β, k_2, k_3, ω	1605	6.72
SEIR (c)	50	100		400		1605	5.32
SEIR (d)	50	150		600		2405	12.88

Conclusions

An optimization-based method for finding parameter values ensuring a prescribed dynamical behaviour for nonlinear models was proposed in this paper. The approach was inspired by nonlinear model predictive control with complex constraints used e.g., in [25]. Therefore, the computations used the discrete-time dynamics of the models. Clearly, the choice of the discretization method is fundamentally important in balancing between computability and the satisfactory approximation of the nonlinear dynamics. In contrast to simulation-based global optimization approaches, the applied setup requires one optimization run. The price of this is the increased number of decision variables which are, however, strictly constrained. The methodology was illustrated using three kinetic nonlinear models where the goal was to induce some kind of oscillatory behaviour (sustained or damped) with a given frequency. The computations were successfully performed using the proprietary Baron solver. Future work will be focused on defining the qualitative dynamical requirements in STL form and automatically translating the constraints from those, and also incorporating other solvers for which the current solutions may possibly be used as feasible initial values.

Acknowledgements

B. Cs. acknowledges the support of the ÚNKP-23-3-II-PPKE-10 New National Excellence Program of the Ministry for Culture and Innovation from the source of the National Research, Development and Innovation Fund (NKFIH). G. Sz. acknowledges the support of the grants NKFIH-OTKA 145934 and RRF-2.3.1-21-2022-00006.

References

[1] W. M. Haddad, V. Chellaboina, and Q. Hui, *Nonnegative and Compartmental Dynamical Systems*. Princeton University Press, 2010.

[2] L. Farina and S. Rinaldi, *Positive Linear Systems: Theory and Applications*. John Wiley & Sons, 2000.

[3] M. Feinberg, *Foundations of Chemical Reaction Network Theory*. Springer, 2019.

[4] P. Érdi and J. Tóth, *Mathematical Models of Chemical Reactions: Theory and Applications of Deterministic and Stochastic models*. Manchester University Press, 1989.

[5] N. Samardzija, L. D. Greller, and E. Wasserman, "Nonlinear chemical kinetic schemes derived from mechanical and electrical dynamical systems," *The Journal of Chemical Physics*, vol. 90, no. 4, pp. 2296–2304, 1989.

[6] G. Lipták, G. Szederkényi, and K. M. Hangos, "Kinetic feedback design for polynomial systems," *Journal of Process Control*, vol. 41, p. 56–66, May 2016.

[7] M. Pereira, B. Kulcsár, G. Lipták, M. Kovács, and G. Szederkényi, "The traffic reaction model: A kinetic compartmental approach to road traffic modeling," *Transportation Research Part C: Emerging Technologies*, vol. 158, p. 104435, Jan. 2024.

[8] I. Epstein and J. Pojman, *An Introduction to Nonlinear Chemical Dynamics: Oscillations, Waves, Patterns, and Chaos*. Topics in Physical Chemistry, Oxford University Press, 1998.

[9] J. J. Tyson, K. C. Chen, and B. Novak, "Sniffers, buzzers, toggles and blinkers: dynamics of regulatory and signaling pathways in the cell," *Current Opinion in Cell Biology*, vol. 15, no. 2, pp. 221–231, 2003.

[10] C. Conradi and C. Pantea, *Multistationarity in Biochemical Networks: Results, Analysis, and Examples*. Academic Press, 2019. Algebraic and Combinatorial Computational Biology, Ch. 9, Eds. Robeva R. and Macaulay M., Mathematics in Science and Computation.

[11] B. Boros and J. Hofbauer, "Some minimal bimolecular mass-action systems with limit cycles," *Nonlinear Analysis: Real World Applications*, vol. 72, p. 103839, Aug. 2023.

[12] R. J. Field and F. W. Schneider, "Oscillating chemical reactions and nonlinear dynamics," *Journal of Chemical Education*, vol. 66, no. 3, p. 195, 1989.

[13] K. Beutel and E. Peacock-Lopez, "Chemical oscillations: Two-variable models," *The Chemical Educator*, vol. 12, pp. 224–235, 01 2007.

[14] I. Otero-Muras and J. R. Banga, "Automated design framework for synthetic biology exploiting Pareto optimality," *ACS Synthetic Biology*, vol. 6, no. 7, pp. 1180–1193, 2017.

[15] B. Shults and B. Kuipers, "Qualitative simulation and temporal logic: proving properties of continuous systems," *University of Texas Artificial Intelligence Laboratory TR AI96-244*, 1996.

[16] P. Ballarini and M. L. Guerriero, "Query-based verification of qualitative trends and oscillations in biochemical systems," *Theoretical Computer Science*, vol. 411, no. 20, pp. 2019–2036, 2010. Hybrid Automata and Oscillatory Behaviour in Biological Systems.

[17] A. Rizk, G. Batt, F. Fages, and S. Soliman, "On a continuous degree of satisfaction of temporal logic formulae with applications to systems biology," in *Computational Methods in Systems Biology* (M. Heiner and A. M. Uhrmacher, eds.), (Berlin, Heidelberg), pp. 251–268, Springer Berlin Heidelberg, 2008.

[18] F. Fages and S. Soliman, *Model Revision from Temporal Logic Properties in Computational Systems Biology*, pp. 287–304. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008.

[19] C. Banks, *Spatio-Temporal Logic for the Analysis of Biochemical Models*. PhD thesis, 06 2015.

[20] L. Calzone, N. Chabrier-Rivier, F. Fages, and S. Soliman, "A machine learning approach to biochemical reaction rules discovery," in *Proceedings of Foundations of Systems Biology and Engineering (FOSBE'05)*, pp. 375–379, 2005.

[21] A. Donze and V. Raman, "BluSTL: Controller synthesis from signal temporal logic specifications," in *2nd International Workshop on Applied Verification for Continuous and Hybrid Systems (ARCH 2015)*, 2015.

[22] E. A. Gol, E. Bartocci, and C. Belta, "A formal methods approach to pattern synthesis in reaction diffusion systems," in *53rd IEEE Conference on Decision and Control*, pp. 108–113, 2014.

[23] E. Bartocci, E. Aydin Gol, I. Haghighi, and C. Belta, "A formal methods approach to pattern recognition and synthesis in reaction diffusion networks," *IEEE Transactions on Control of Network Systems*, vol. 5, no. 1, pp. 308–320, 2018.

[24] I. Haghighi, N. Mehdipour, E. Bartocci, and C. Belta, "Control from signal temporal logic specifications with smooth cumulative quantitative semantics," in *2019 IEEE 58th Conference on Decision and Control (CDC)*, IEEE, 2019.

[25] T. Péni, B. Csutak, G. Szederkényi, and G. Röst, "Nonlinear model predictive control with logic constraints for COVID-19 management," *Nonlinear Dynamics*, vol. 102, pp. 1965–1986, Dec. 2020.

[26] J. Lofberg, "YALMIP : a toolbox for modeling and optimization in MATLAB," in *2004 IEEE International Conference on Robotics and Automation (IEEE Cat. No.04CH37508)*, pp. 284–289, 2004.

[27] A. Khajavirad and N. V. Sahinidis, "A hybrid LP/NLP paradigm for global optimization relaxations," *Mathematical Programming Computation*, vol. 10, pp. 383 – 421, 2018.

[28] L. Alkhalifa, "Comparison between five MINLP solvers and new results related to trigonometric functions," *Fractals*, vol. 30, no. 10, p. 2240249, 2022.

[29] J. Kronqvist, D. E. Bernal, A. Lundell, and I. E. Grossmann, "A review and comparison of solvers for convex MINLP," *Optimization and Engineering*, vol. 20, pp. 397–455, 2019.

[30] R. Lefever, G. Nicolis, and P. Borckmans, "The Brusselator: it does oscillate all the same," *Journal of the Chemical Society, Faraday Transactions 1: Physical Chemistry in Condensed Phases*, vol. 84, no. 4, pp. 1013–1023, 1988.

[31] Y. Takeuchi, *Global dynamical properties of Lotka-Volterra systems*. World Scientific, 1996.

[32] L. Brenig, "Reducing nonlinear dynamical systems to canonical forms," *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 376, no. 2124, p. 20170384, 2018.

[33] E. Chauvet, J. E. Paullet, J. P. Previte, and Z. Walls, "A Lotka-Volterra three-species food chain," *Mathematics Magazine*, vol. 75, no. 4, pp. 243–255, 2002.

[34] D. J. Daley and J. Gani, *Epidemic Modelling: an Introduction*. No. 15, Cambridge University Press, 2001.

[35] J. Chen, "An SIRS epidemic model," *Applied Mathematics-A Journal of Chinese Universities*, vol. 19, no. 1, pp. 101–108, 2004.