10.24425/acs.2021.139730

Archives of Control Sciences
Volume 31(LXVII), 2021
No. 4*, pages 781–794*

Efficient choice of parameters on delta-reachability bounded hybrid systems

Eugénio Miguel Alexandre ROCHA and Kelly Patricia MURILLO

Hybrid systems (HS) are roughly described as a set of discrete state transitions and continuous dynamics modeled by differential equations. Parametric HS may be constructed by having parameters on the differential equations, initial conditions, jump conditions, or a combination of the previous ones. In real applications, the best solution is obtained by a set of metrics functional over the set of solutions generated from a finite set of parameters. This paper examines the choice of parameters on delta-reachability bounded hybrid systems. We present an efficient model based on the tool pHL-MT to benchmark the HS solutions (based on dReach), and a non-parametric frontier analysis approach, relying on multidirectional efficiency analysis (MEA). Three numerical examples of epidemic models with variable growth infectivity are presented, namely: when the variable of infected individuals oscillates around some endemic (non-autonomous) equilibrium; when there is an asymptotically stable non-trivial attractor; and in the presence of bump functions.

Key words: bounded hybrid system, safety property, delta-reachability, multidirectional efficiency analysis

1. Introduction

Hybrid systems occur frequently in safety-critical applications in various domains (health care, transportation, robotics, systems biology, etc). An interesting problem in hybrid systems' theory is the reachability analysis. In general terms, a reachability analysis problem consists in evaluating whether a given system

K. Murillo (corresponding author), e-mail: kellymurillo@ua.pt) and E. Rocha, e-mail: eugenio@ua.pt are with Center for Research and Development in Mathematics and Applications, and Department of Mathematics, University of Aveiro,3810-193 Aveiro, Portugal.

The authors were supported by The Center for Research and Development in Mathematics and Applications (CIDMA) through the Portuguese Foundation for Science and Technology (FCT – Fundação para a Ciência e a Tecnologia), references UIDB/04106/2020 and UIDP/04106/2020. Murillo was also supported by national funds (OE), through FCT, I.P., in the scope of the framework contract foreseen in the numbers 4, 5 and 6 of the article 23, of the Decree-Law 57/2016, of August 29, changed by Law 57/2017, of July 19.

Received 29.04.2021.

may reach certain unsafe states, starting from certain initial states [15]. Let H be a hybrid system where $X \subseteq R^k$ is a state space and Q is a finite set of modes. We write $[\![H]\!]$ to denote the set of all possible trajectories of H. In mathematical terms, the reachability analysis problem is translated in the following question: Given Unsafe $\subseteq R^k \times Q$, $[\![H]\!] \cap \mathit{Unsafe} = \emptyset$?. A system is safe (bounded) when the system does not violate the safety property within a bounded period of time and a bounded number of discrete mode changes. This work focuses on the study the δ-reachability problem. Given a bounded error δ, a system is δ-unsafe when the system would violate the safety property under some δ-bounded numerical perturbations.

Different methods have been proposed for solving the reachability problem: optimal control, deductive techniques, model checking techniques, and approximations, which works with all the above. In optimal control, approximate by sets and dynamics for which the problems are easier to solve; deductive techniques are inherently based on over-approximation, and in model checking, approximates by a system you can compute with; see [11]. In the literature, some techniques are available for the reachability analysis. SpaceEx is well-suited to analyze linear hybrid systems [6]. Flow* is specialized in non-linear systems but with a recent enhancement for dealing with linear systems [3]. Other prominent tools are KeYmaera [12], iSAT [5], HSolver [13], HyCreate [9] and Cora [1]. This work is a extension of the tool dReach for the reachability analysis, based on bounded model checking using constraint solving techniques. It encodes bounded reachability problems of hybrid systems as first-order formulas over the real numbers, and solves them using δ-decision procedures in the SMT solver dReal [7, 8].

In order to study the efficient choice of parameters in a parametric hybrid system, it is important to examine the fitting problem associated: what is the best algorithm to adjust a hybrid system to the data? In this sense, the notion of "best" can be approached in different contexts. In the setting of real optimization, maximizing a cost function with weights, in optimal control theory, maximizing a cost functional, or by finding a Pareto type solution, among others. Here, the notion of "best" is considered as the "most efficient". Efficiency allows to establish a relative ranking, using frontier analysis. The most common (non-parametric) technique in frontier analysis is the Data Envelopment Analysis (DEA) and an extension is the Multidirectional Efficiency Analysis (MEA), proposed in [2]. In contrast to DEA, the input reduction and output expansion benchmarks in the MEA approach are selected proportional to the potential improvements in efficiency identified, while considering the potential improvement separately in each variable.Thus, in addition to efficiency levels, MEA allows investigating changes in efficiency patterns.

In this work, a methodology is presented for efficient choice of parameters based on the called pHL-MT tool, (parallel hybrid language-model translator), to benchmark the hybrid system solutions, based on dReach [10] and a non-parametric frontier analysis approach relying on MEA [2]. In general terms, the problem to study is represented in the Figure 1. We have an ordinary differential equation that describes a phenomenon for a parameter α given. We want to change the parameter α to a new parameter $\bar{\alpha}$. Note that for different values of $\alpha \in \{\bar{\alpha_1}, \bar{\alpha_2}, \bar{\alpha_3}\}$, the differential equation will show different scenarios: solution $S1$, solution $S2$ and solution $S3$. Therefore, the parameter change will generate costs and will have some impact on the result. Then, we need to find an efficient way to generate all scenarios and measure the best one possible. In this sense, the model proposed here allows us to answer the following three questions:

Question A: How we can measure the cost of using $\bar{\alpha}$; instead of α?

Question B: How do measure the final impact of using $\bar{\alpha}$; instead of α?

Question C: If we have several parameters how we rank them?

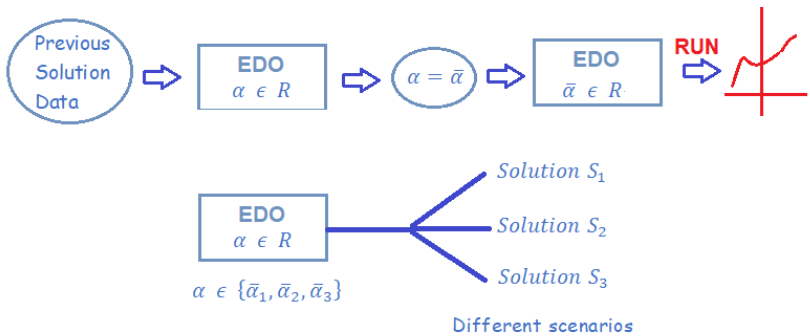

Figure 1: Fitting of parameters based on the previous data

The principal contribution of this work is considering the problem of finding an efficient choice of parameters of a parameterized hybrid system that will provide the best solution from a (given) set of solutions of the system generated from a finite set of parameters' values. The broad approach is to generate a set of solutions for each value of parameters using the tool dReach. Then, a set of functions measuring the characteristic of solutions is used, and for each solution, the pair of solution and its measurable value is stored in a database. Finally, a multidirectional efficiency analysis is used to establish the relative ranking of the solutions. In this sense, once the solutions are generated, the functional input will allow us to study the costs generated (answering question A); the functional output allows to study the impact of the results, (answering question B), and the application of the MEA model, will allow to identify the best scenario (answering question C). A detailed description of how the proposed model works is provided in Section 2.

The remainder of the paper is laid out as follows. The next section exposes the approach to the reachability problem and introduces the methodology proposed. A brief description of the main related packages and an overview of the multidirectional efficiency analysis, are given. Section 3 shows three numerical examples of a SIR model with variable growth infectivity. To end, in Section 4, some concluding remarks are formulated.

2. Methodology for an efficient choice of parameters on bounded hybrid systems

There are many natural systems that show a hybrid behavior characterized by a set of continuous laws that are modified by discrete events and can be described very naturally by hybrid automata. Hybrid automata is a formal model that combines discrete control graphs, usually called finite state automata, with continuously evolving variables, in mathematical terms:

Definition 1 *Hybrid automata is a tuple $H = (X, Q, Init, Flow, Jump, Inv)$ where: $X \subseteq R^k$ is a state space; Q is a finite set of modes; $Init\colon \subseteq Q \times X$ is a set of initial configurations; $Flow\colon \subseteq Q \times X \to TX$ are continuos flows (where $T \in R^+$ is an upper bound on the time duration); $Jump\colon \subseteq Q \times X \to 2^{Q \times X}$ are discrete jumps; $Inv\colon \subseteq Q \times X$ is an invariant set in each mode.*

Let δ be a numerical error bound and the δ-perturbation $H^\delta = (X, Q, Init^\delta, Flow^\delta, Jump^\delta, Inv^\delta)$ of H. Choose $n \in N$ to be a bound on the number of discrete mode changes. Let unsafe encode a subset of $X \times Q$, the state space of H, the δ-reachability problem asks for one of the following answers: safe (H cannot reach unsafe in n steps within time T) or δ-unsafe (H^δ can reach *unsafe*$^\delta$ in n steps within time T). When safe is the answer, H does not reach the unsafe region (no δ is involved); when δ-unsafe is the answer, there exists some δ-bounded perturbation of the system that can render it unsafe. The reachability problem is associated with the safety verification problem, which proves that the system can never reach any unsafe state. δ-reachability analysis checks robustness which implies safety. If a system is δ-reachable under a reasonably small δ, then a small error can lead the system to an unsafe state (robustness), see [7, 8].

Given a parametric hybrid system, a domain of parameters and a set of functional to measure the characteristics of the solutions, we establish a methodology, for determining the efficient choice of parameters. The model is main based on the software tool pHL-MT, which involves packages of logic and statics (see Introduction Section dReach [10] and MEA [2]).

The system architecture of pHL-MT tool is given in Figure 2. The scheme comprises the following five stages:

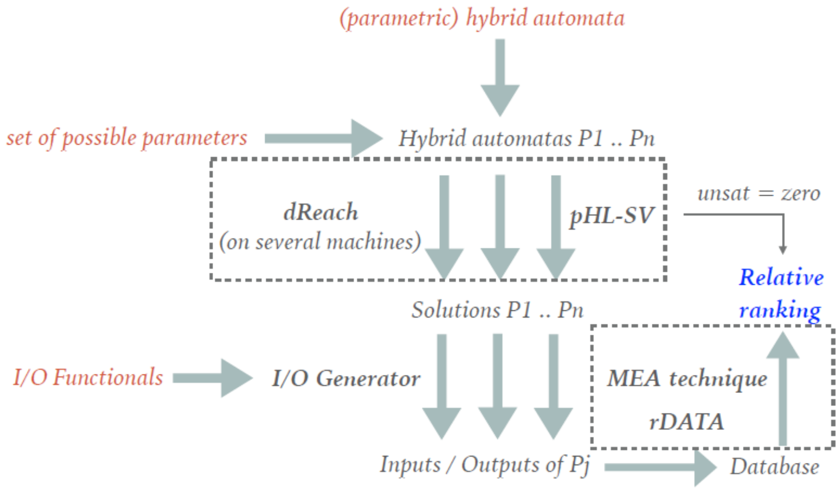

Figure 2: Efficient squem with parallel hybrid languaje

First stage: input the file of pHL with a parametric hybrid automata H.

Second stage: input a domain of possible parameters, generating hybrid automata Pj, $1 < j < n$.

Third stage: from the inputs, dReach generates a logical encoding that involves the quantification on the time variables. The tool then makes iterative calls to the package pHL-SV, which is based on solver dReach, to decide the reachability properties. When the answer is δ-reachable, dReach generates the solutions for each hybrid automata Pj. When the answer is unreachable, no numerical error is involved and the answer is unsat equal to zero.

Fourth stage: insert a set of input/output functional ($I_{i_{(S,I,R)}}$, $i \in N$ and $O_{k_{(S,I,R)}}$, $k \in N$) to measure the characteristics of the solutions.

Fifth stage: a set of inputs and outputs is generated for each Pj, which make up a database. Once the database has been created, the MEA technique is applied to establish a relative ranking of the different scenarios.

The calculation in pHL-MT tool is run in parallel. The execution of the five steps allows getting parameters' value corresponding to the best solution. These parameters correspond to the solution with the best position (first position) within the ranking, showing that it is the most efficient in the sample. Efficient, in the sense of minimize inputs and maximize outputs.

```
//%OPS CAS=MAPLE
//%OPS RUNSEQ=CODE TEXT
```

```
//%SCOPE Hybrid-beta SIR model for I_b > I_s and T^\ast=+infty

// ****************************************************
// PARAMETERS FOR SPECIAL SYSTEM VARS
// ****************************************************

define(_TTIME,  30)     // Total Sym Time

// CONSTANTS
define(_TS, 0.8)
define(_IB, 0.3)
define(_IS, 0.01)

define(_B0, 1.4)
define(_BP, 0.2)
define(_BN, -1.3)
define(_BX, EVAL(_TTIME * (_BP + 1)))

define(_I0, 0.200)
define(_R0, 0.150)
define(_S0, EVAL(1 - _I0 - _R0))
define(_alpha, 0.1)
define(_zeta, 0.1)
define(_gamma, 0.1)
define(_lambdaZ, EVAL(_B0 - _alpha - _zeta - _gamma))

define(_C1,1)
define(_C2,1)
define(_h,3)

#define _FLOWINIT  d/dt[T] = 1;  d/dt[nT] = 1;
#define _FLOWVARS [0, _TTIME] time; [0, _TTIME] T; [0, _DTIME] nT;
#define _JUMP0 (T:M4S=T)(nT:M4S=0)
#define _INITVARS0 (nT=0)(T=0)

define(_DTIME,  _TTIME) // default  DTIME = TTIME / 1

// --------- main ODEs --------------------
define(_dtVS, ‘d/dt[VS] = _alpha - _alpha * VS + _zeta * VI - VB * VI * VS’)
define(_dtVI, ‘d/dt[VI] = VB * VI * VS - (_alpha + _zeta + _gamma) * VI’)
define(_dtVR, ‘d/dt[VR] = _gamma * VI - _alpha * VR’)

// --------- ODE jump conditions --------------------
define(_JUMP, (VS’ = VS)(VI’ = VI)(VR’ = VR)(VB’ = VB)(VT’ = 0))

// --------- Initial conditions --------------------
define(_INITVARS, (VS = _S0)(VI = _I0)(VR = _R0)(VB = _B0)(VT = 0))

// --------- Goal conditions --------------------
define(_NODE, 1)
define(_GOAL, (and (T >= _TTIME)))

BEGIN-CODE [ pHL-MT/3.16 | -l=2 -k=3 -html ]
{ config;
 param:
   _B0: 1.4,0.4,1.4;
   _BN: -1.3,-0.2,-1.3;
   _BP: 0.2,0.2,0.2;
   _IB: 0.6,0.3,0.1;
   _IS: 0.5,0.285,0.05;
   _alpha: 0.1,0.1,0.1;
```

```
   _zeta: 0.1,0.2,0.1;
   _gamma: 0.1,0.1,0.1;
 inputs:
   I1: _C1*sum(i,1,_h,2*(_h+1-i)*(_h*(_h+1))*VI[(_h-1)*i]);
   I2: _C2*int(t,0,_TTIME,ind_mode(3,4));
 outputs:
   O1: 1-max(t,0,_TTIME,I[t]);
   O2: 1-1/(_TTIME)*int(t,0,_TTIME,VI[t]);
}

_FLOWVARS
[0, 1] VS;              // susceptible
[0, 1] VI;             // infected and infectious
[0, 1] VR;              // recovery
[-_BX, _BX] VB;         // beta
[0,_TS] VT;             // max intervention time

{ mode 1;  // terminal node
 invt:
 flow: _FLOWINIT
    d/dt[VS] = 0;
    d/dt[VI] = 0;
    d/dt[VR] = 0;
    d/dt[VB] = 0;
    d/dt[VT] = 0;
 jump:
}

{ mode 2;  // [off_- & VI <= IB]
 invt:
    VI <= _IB;
    VB >= 0.0;
 flow: _FLOWINIT
    _dtVS;
    _dtVI;
    _dtVR;
    d/dt[VB] = _BP;
    d/dt[VT] = 0;
 jump:
    (and (VI=_IB)) ==> @3 (and _JUMP0 _JUMP);
    (and (T=_TTIME)) ==> @1 (and _JUMP0 _JUMP);
}

{ mode 3; // [on_+]
 invt:
    VI >= _IS;
    VT <= _TS;
    VB >= 0.0;
 flow: _FLOWINIT
    _dtVS;
    _dtVI;
    _dtVR;
    d/dt[VB] = _BN * VB;
    d/dt[VT] = 1;
 jump:
    (and (VB=0.0)) ==> @4 (and _JUMP0 _JUMP);
    (and (VI<=_IB)(or (VI=_IS)(VT=_TS))) ==> @2 (and _JUMP0 _JUMP);
    (and (VI>=_IB)(or (VI=_IS)(VT=_TS))) ==> @5 (and _JUMP0 _JUMP);
    (and (T=_TTIME)) ==> @1 (and _JUMP0 _JUMP);
}
```

```
{ mode 4; // [on_+ and VB=0]
 invt:
    VI >= _IS;
    VT <= _TS;
    VB >= 0.0;
 flow: _FLOWINIT
    _dtVS;
    _dtVI;
    _dtVR;
    d/dt[VB] = 0;
    d/dt[VT] = 1;
 jump:
    (and (VI<=_IB)(or (VI=_IS)(VT=_TS))) ==> @2 (and _JUMP0 _JUMP);
    (and (VI>=_IB)(or (VI=_IS)(VT=_TS))) ==> @5 (and _JUMP0 _JUMP);
    (and (T=_TTIME)) ==> @1 (and _JUMP0 _JUMP);
}

{ mode 5;  // [off_- & VI>= IB]
 invt:
    VI >= _IB;
    VB >= 0.0;
 flow: _FLOWINIT
    _dtVS;
    _dtVI;
    _dtVR;
    d/dt[VB] = _BP;
    d/dt[VT] = 0;
 jump:
    (and (VI=_IB)) ==> @3 (and _JUMP0 _JUMP);
    (and (T=_TTIME)) ==> @1 (and _JUMP0 _JUMP);
}

// --------- Target Conditions --------------------
init:
  @2 (and _INITVARS0 _INITVARS);

goal:
  @_NODE  (and _GOAL);
END-CODE
```

2.1. MEA model

To continue, the main idea of a nonparametric and deterministic MEA model is introduced. Let $[m]$ denotes the set $\{1, \ldots, m\}$; for notation convenience. From what was discussed above, to any given simulation (ρ) it is possible to associate $J \in N$ outputs $y_j(\rho)$, $j \in [J]$ and $I \in N$ inputs $x_i(\rho)$, $i \in [I]$.Some input variables may be discretionary (i.e. their values can be changed) but others may be non-discretionary (i.e. they are fixed). From now on, the discretionary variables are represented by the first indices from 1 to $d \in [1, I]$. So, $x(\rho)$ is the vector of all the inputs and $y(\rho)$ is the vector of all the outputs. Furthermore, the model change with respect to a chosen set of complementary variables. Here is used the so-called variable returns to scale (VRS) model, by defining the set

$$\Lambda^N = \left\{ \lambda \in R^N\colon \sum_{n=1}^{N} \lambda_n = 1 \wedge \lambda_n \geqslant 0 \right\}, \tag{1}$$

where N is the number of sequences under study. Alternative definitions of Λ^N allow other models, not relevant in this work. Then, the MEA score is found by solving the following linear optimization problems.

$$\begin{array}{ll}
\underline{\text{Problem } P_m^{\alpha}(\bar{\rho})}: & \underline{\text{Problem } P_j^{\beta}(\bar{\rho})}: \\
\min \alpha_m(\bar{\rho}) \text{ such that} & \max \beta_j(\bar{\rho}) \text{ such that} \\
\sum_n \lambda_n x_m(\rho) \leqslant \alpha_m(\bar{\rho}) & \sum_n \lambda_n x_i(\rho) \leqslant x_i(\bar{\rho}),\ \ i \in [I] \\
\sum_n \lambda_n x_i(\rho) \leqslant x_i(\bar{\rho}),\ i \in [I],\ \ i \neq m & \sum_n \lambda_n y_s(\rho) \leqslant \beta_j(\bar{\rho}),\ \ s \in [J] \\
\sum_n \lambda_n y_l(\rho) \leqslant y_l(\bar{\rho}),\ \ l \in [J] & \sum_n \lambda_n y_l(\rho) \leqslant y_l(\bar{\rho}),\ l \in [J],\ \ l \neq j,
\end{array} \tag{2}$$

$$\begin{array}{l}
\underline{\text{Problem } P^{\gamma}(\alpha, \beta, \bar{\rho})}: \\
\max \gamma(\bar{\rho}) \text{ such that} \\
\sum_n \lambda_n x_i(\rho) \leqslant x_i(\bar{\rho}) - \gamma(\bar{\rho})(x_i(\bar{\rho}) - \alpha_i^*(\bar{\rho})),\ \ i \in [I] \\
\sum_n \lambda_n x_i(\rho) \leqslant x_i(\bar{\rho}),\ \ i \in [I] \setminus \{m\} \\
\sum_n \lambda_n y_l(\rho) \geqslant y_l(\bar{\rho}) + \gamma(\bar{\rho})(\beta_l^*(\bar{\rho}) - y_l(\bar{\rho})),\ \ l \in [J] \\
\sum_n \lambda_\rho \geqslant \bar{\rho},\ \ \forall n.
\end{array} \tag{3}$$

where $\lambda \in \Lambda^n$, $\alpha_m^*(\bar{\rho})$ and $\beta_j^*(\bar{\rho})$ are the optimal solutions to the problems $P_m^{\alpha}(z, \bar{\rho})$ and $P_j^{\beta}(z, \bar{\rho})$ respectively.

Definition 2 *The MEA score obtained by the directional contribution of each input and each output variable, is defined by*

$$MEA(\rho) = \frac{\dfrac{1}{\gamma^*(n)} - \dfrac{1}{D}\displaystyle\sum_{i=1}^{D} \frac{x_i(\rho) - \alpha_i^*(\rho)}{x_i(n)}}{\dfrac{1}{\gamma^*(\rho)} + \dfrac{1}{J}\displaystyle\sum_{j=1}^{J} \frac{\beta_j^*(\rho) - y_j(\rho)}{y_j(\rho)}} \in [0, 1], \tag{4}$$

where α^*, β^* and ρ^* are the optimal solutions to the problems $P_m^{\alpha}(z, \bar{\rho})$, $P_j^{\beta}(z, \bar{\rho})$ and $P^{\gamma}(\alpha, \beta, \bar{\rho})$ respectively.

3. Numerical examples in epidemic modelling

In this section, we present three examples of an epidemic model which allows show the importance of the proposed method. The examples are showed following the same structure of scheme 1.

First stage. Considere the SIR epidemic model together with a piecewise linear continuous coefficient β'_λ, described by

$$(a)\quad \begin{aligned} S' &= \alpha - \alpha S + \zeta I - \beta_\lambda IS \\ I' &= \beta_\lambda IS - (\alpha + \zeta + \theta_\gamma)I \\ R' &= \theta(\gamma I - \alpha R), \end{aligned} \quad \text{and} \quad (b)\quad \beta'_\lambda = \lambda\zeta,\ \ \zeta \in (\beta_-,\ \beta_+) \tag{5}$$

where the values $S(t)$, $I(t)$, $R(t)$ are respectively, the number of healthy individuals (susceptible), infected individuals and recovered individuals; $\alpha \geqslant 0$ is a parameter of birth and death, $\gamma > 0$ is a recovery rate without possibility of re-infection, ζ accounts for the rate of individuals that become healthy but may be re-infected in the future with $\zeta \geqslant -\alpha$, and $\lambda \in R$ is a bifurcation parameter.

The hybrid model associated to (5) is described in Fig. 3. In this case, is established the maximum time of intervention $T^* > 0$ and two threshold values

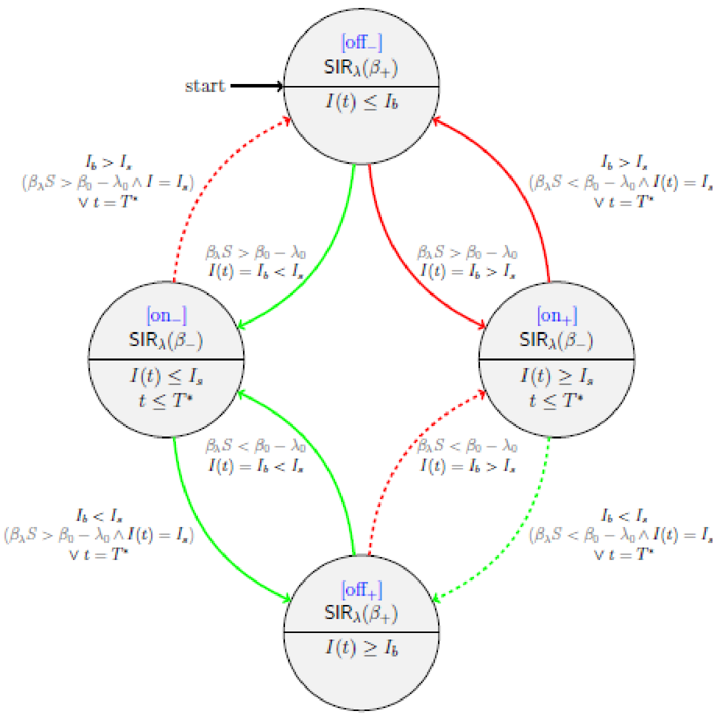

Figure 3: Hybrid system associated to (5) describing the agent police

as triggers to the on/off states: $I_b > 0$: when the government answer starts; and $I_s > 0$: when the government answer stops. We also consider three government strategies: (S_0) ... stops with only T^*; (S_1) ... stops with only I_s; and (S_2) ... stops by either.

Second stage. Three numerical examples (E1)–(E3) are studied. Table 1 presents the parameters considered in each case.

Table 1: Parameters and initials conditions

Ex.	α	ζ	γ	β_-	β_+	I_b	I_s	$S(0)$	$I(0)$	$R(0)$	$\beta(0)$
E1	0.1	0.1	0,1	–1.3	0.2	0.6	0.500	0.550	0.300	0.150	1.400
E2	0.1	0.2	0,1	–0.2	0.2	0.3	0.285	0.910	0.060	0.030	0.400
E3	0.1	0.1	0,1	–1.3	0.2	0.1	0.050	0.925	0.050	0.025	1.400

Third stage. The model results allow to characterize the examples studied. Example (E1) where a nontrivial asymptotically stable attractor on $I(t)$ appears, see Figure 4. Example (E2): with a set of parameters for which the infected individuals variable $I(t)$ oscillates around some endemic equilibrium which in the limit tends to $(\bar{S}, \bar{I}) = \left(0, \dfrac{\alpha}{\alpha+\gamma}\right) = (0, 0.3)$, see Figure 5. Example (E3): with a succession of bump behaviors a long time, although in the intervals between bumps $I(t)$ is coming near to zero, see Figure 6.

Fourth stage. Consider the following functionals (advised by healthcare professionals):

$$\begin{aligned}
I_{1(S,I,R)} &= c_1 \sum_{i=1}^{h} 2\frac{h+1-i}{h(h+1)} I((h-1)i),\\
I_{2(S,I,R)} &= c_2 \int_0^T \chi_{\{t\in \mathrm{on}\}}(t)\,\mathrm{d}t,\\
O_{1(S,I,R)} &= 1 - \max_{0\leqslant t\leqslant T} I(t),\\
O_{2f(S,I,R)} &= 1 - \frac{1}{T}\int_0^T I(t)\,\mathrm{d}t.
\end{aligned} \tag{6}$$

Fifth stage. Table 2 presents the sets of inputs and outputs for each example and the ranking results.

The best scenario in this SIR model, under the established conditions, corresponds to the Example (E3): a succession of bump behaviors a long time, which in the intervals between bumps $I(t)$ is coming near to zero.

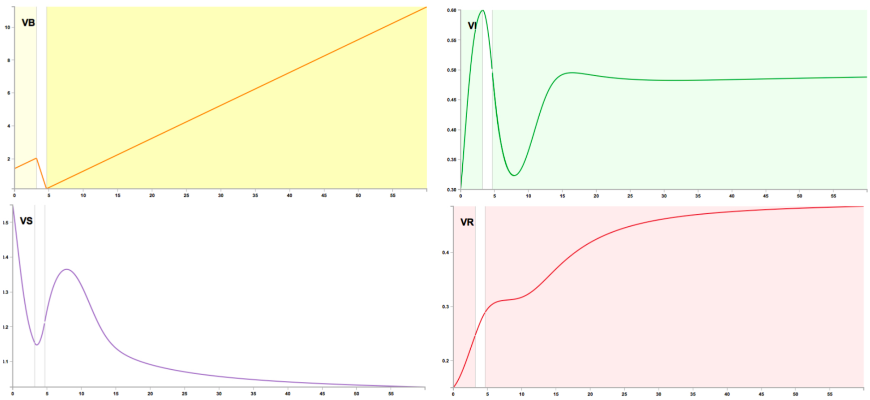

Figure 4: A nontrivial asymptotically stable attractor for $I(t)$

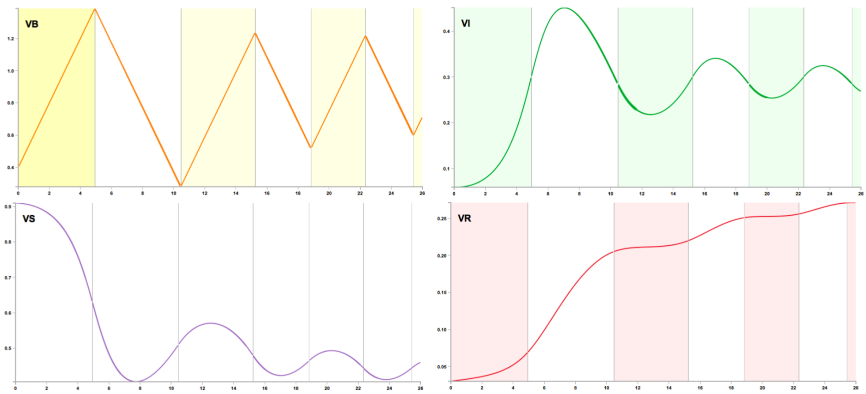

Figure 5: Oscillatory behaviour of $I(t)$

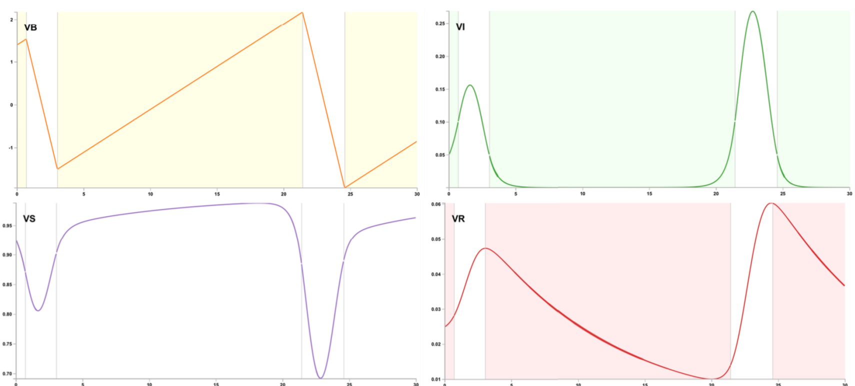

Figure 6: Bump behaviour of $I(t)$

Table 2: Inputs, outputs and MEA values

Ex	I_1	I_2	O_1	O_2	MEA
E1	0.493	1.759	0.403	0.527	0.673
E2	0.319	17.23	0.531	0.730	0.483
E3	0.031	9.581	0.726	0.825	1.000

4. Conclusions

We propose a methodology for determining the efficient choice of parameters in hybrid systems, where the optimization is performed over a given set of functional which measure specific features of the solutions. Although the set of parameters is finite and small, the optimization problem turns to be a real multi objective problem with inputs and outputs variables; for which benchmarking techniques seem to be the most adequate.

In this work, we extend the dReach tool, which verifies delta-reachability of hybrid systems, by adding the power of multidirectional efficiency analysis in order to do the solutions benchmark. We also apply the ideas to a model in epidemic mathematical modelling, where the functional where suggested by experts in healthcare. This methodology corresponds to a first approach and extensions will be the subject of future research.

References

[1] M. Althoff and J.M. Dolan: Online verification of automated road vehicles using reachability analysis. *IEEE Trans. on Robotics*, **30**(4), (2014), 903–918, DOI: 10.1109/TRO.2014.2312453.

[2] P. Bogetoft and J.L. Hougaard: Efficiency evaluations based on potential (non-proportional) improvements. *Journal of Productivity Analysis*, **12**(3), (1999), 233–247, DOI: 10.1023/A:1007848222681.

[3] X. Chen, E. Abraham and S. Sankaranarayanan: Flow*: An analyzer for non-linear hybrid systems. In: Proc. of CAV'13. LNCS, **8044**, 258–263, Springer, 2013.

[4] E.M. Clarke and S. Gao: Model checking hybrid systems. In: Margaria T., Steffen B. (eds): Leveraging Applications of Formal Methods, Verification and Validation. Specialized Techniques and Applications. ISoLA 2014. Lecture Notes in Computer Science, **8803**, 385–386, Springer, Berlin, Heidelberg, 2014.

[5] M. Franzle, C. Herde, S. Ratschan, T. Schubert and T. Teige: Efficient solving of large non-linear arithmetic constraint systems with complex Boolean structure. *Journal on Satisfiability, Boolean Modeling and Computation*, **1** (2007), 209–236, DOI: 10.3233/SAT190012.

[6] G. Frehse, C.L. Guernic, A. Donze, R. Ray, O. Lebeltel, R. Ripado, A. Girard, T. Dang and O. Maler: SpaceEx: Scalable verification of hybrid systems. In: Proc. ofCAV'11. LNCS, **6806**, 379–395, Springer, 2011.

[7] S. Gao: Computable analysis, decision procedures, and hybrid automata: A new framework for the formal verification of cyber-physical systems. Ph.D. thesis, Carnegie Mellon University, 2012.

[8] S. Gao, S. Kong and E.M. Clarke: dReal: An SMT solver for nonlinear theories over the reals. In: M.P. Bonacina (ed.) CADE 2013. LNCS (LNAI), **7898**, 208–214, Springer, Heidelberg (2013). DOI: 10.1007/978-3-642-38574-2.

[9] HyCreate: A tool for overapproximating reachability of hybrid automata, http://stanleybak.com/projects/hycreate/hycreate.html.

[10] S. Kong, S. Gao, W. Chen and E. Clarke: dReach: δ-reachability analysis for hybrid systems. In Proc. International Conference on Tools and Algorithms for the Construction and Analysis of Systems, 2015. Available: http://link.springer.com/10.1007/978-3-662-46681-0.

[11] J. Lygeros, C. Tomlin and S. Sastry: Hybrid systems: modeling analysis and control. Electronic Research Laboratory, University of California, Berkeley, CA, Tech. Rep. UCB/ERL M, 2008.

[12] A. Platzer and J. Quesel: Keymaera: A hybrid theorem prover for hybrid systems (system description). In: Proc. of IJCAR'08. LNCS, **5195**, 171–178, Springer, 2008.

[13] S. Ratschan and Z. She: Safety verification of hybrid systems by constraint propagation based abstraction refinement. In: Proc. of HSCC'05. LNCS, **3414**, 573–589, Springer, 2005.

[14] E.M. Rocha: Oscillatory behaviour on a non-autonomous hybrid SIR-model. In: M. Chaves and M. Martins (eds.), Molecular Logic and Computational Synthetic Biology. MLCSB 2018. Lecture Notes in Computer Science, **11415**, Springer, Cham, 2019.

[15] L. Zhang, Z. She, S. Ratschan, H. Hermanns and E. Hahn: Safety verification for probabilistic hybrid systems. In: Proc. International Conference on Computer Aided Verification, (2010), 196–211.