A Remark on Functional Sensitivity

Stanislaw Raczynski

Universidad Panamericana, Facultad de Ingeniería, Augusto Rodin 498, México, Ciudad de México, 03920, México

(Received November 24 2018, Accepted May 10 2019)

Abstract. Unlike the conventional definition of model sensitivity that uses the partial derivatives of model output with respect to the inputs, we define the sensitivity using the concepts of variational calculus. In the local approach, the sensitivity is defined using the variation of the perturbation, and the model output is treated as a functional, and not a function of the inputs. In non-local version, we do not require that the variations are small. This leads to the problem of the determination of reachable sets for certain differential inclusion. The relation to differential inclusions and control systems is discussed. Some examples are provided and compared to the conventional sensitivity procedures offered by the known system dynamics packages. In the research, results from the optimal control theory are used. The main tool used in the paper is the differential inclusion solver that provides images of the sensitivity sets for given dynamic models. The solver can also be used to analyze problems given in the form of differential inclusions. Main fields of applications include uncertainty treatment, robust control systems and any systems with perturbations that are not necessarily random variables with known probability properties.

Keywords: simulation, sensitivity, differential equations, uncertainty, differential inclusions

1 Introduction

The classical, local sensitivity analysis (basic local version) is based on the partial derivative of the model output Y, with respect to components of an input vector (model parameters) \( u = (u_1, u_2, \ldots, u_n) \), at a given point \( u_0 \):

\[
\left| \frac{\partial Y}{\partial u_i} \right|_{u_0}
\]

The derivative is taken at some fixed point in the space of the input (hence the ‘local’ in the name of the analysis mode). The use of partial derivatives suggests that we consider small perturbations of the input vector, around the point of interest \( u_0 \) [12]. The classical approach to model sensitivity (SA) and its modifications are not the topic of the present paper, so will not give here any extensive overview, making only some remarks. Scatter plots represent a useful tool in the SA. Plot of the output variable against individual input variables, after (randomly) sampling the model over its input distributions are made. This give us a graphical view of the model sensitivity [5].

Regression analysis is a powerful tool for sensitivity problems. It allows you to examine the relationship between two or more variables of interest. The method is used to model the relationship between a response variable and one or more predictor variables or perturbations [3, 4].

For non-linear models, one of useful tool are the variance-based models or Sobol methods. It decomposes the variance of the output of the model or system into fractions which can be attributed to the input or sets of inputs [13].

* Corresponding author. E-mail address: stanracz@stanr.com

Published by World Academic Press, World Academic Union
Screening method decomposes the variance of the output of the model or system into fractions which can be attributed to model inputs. Thus, we can see which variable is contributing significantly to the output uncertainty in high-dimensionality models. For more detail [7].

The logarithmic gain is a normalized sensitivity defined by the percentage response of a dependent variable to an infinitesimal percentage change in an independent variable. In dynamical systems, the logarithmic gain can vary with time, and this time-varying sensitivity is hereby called dynamic logarithmic gain. This concept is used in dynamic sensitivity analysis, where the core model is a dynamic system, described by ordinary differential equations [10].

The System Dynamics software offers tools for dynamic sensitivity analysis. Programs like Vensim or PowesSim include procedures that generate multiple model trajectories when the selected model parameters vary from one trajectory to another. However, in these packages the parameters are constant along the trajectory. Our approach is different. As explained in the following sections, we treat the perturbations as functions of time. The main tool used here is the differential inclusion.

This paper is focused on an application of some basic concepts of variational calculus to analyze the sensitivity of dynamic systems. It is pointed out that, according to our definition, the sensitivity task is equivalent to the problem of determination of reachable sets of perturbed model state. The main tool is the differential inclusion solver developed by the author. The images produced by the solver can hardly be found in other works on sensitivity. The analysis presented here is closely related to the differential inclusions [11]. This type of sensitivity should not be confused with the variational sensitivity discussed in various works, mainly in the field of structural design. For example, in [15] a variational method is applied to shape design sensitivity. The result presented in that paper is the design sensitivity expression, the main field of applications of these expressions is mechanical design, including problems of stress, strain, displacement, pressure and force fields.

For other works on variational sensitivity consult also Mordukhovich who presents applications of the generalized differentiation theory in variational analysis to sensitivity. The basic tools used in that paper involve coderivatives of set-valued mappings and second-order sub-differentiation of extended-real-valued functions. New conditions for robust stability are given including the problems in Banach spaces [6].

Note, however, that the methods and results discussed in above works are quite different from the topic on the present paper. Here, we are looking rather for the sensitivity sets for given dynamic systems, and focus rather on the software tools and not on analytical solutions. As stated before, the images produced by the software applied in this paper, can hardly be found in other works on sensitivity analysis. It should be also noted that the method used here consists in the deterministic, and not stochastic treatment of uncertainty.

The research direction in this research area should include both theoretic questions and the improvements of the software being developed. The theoretic background for the research is already available from the works on differential inclusions. However, the applicability to systems that do not satisfy some common regularity assumptions of the control theory should be investigated. As for the software, an important problem is the presentation of the results for multi-dimensional models, where the reachable sets are given as clouds of points in multi-dimensional space.

2 Functional sensitivity

Consider a dynamic model described by an ordinary differential equation

\[
\frac{dx}{dt} = f(x,u,t)
\]

where \(x = (x_1, x_2, ..., x_n)\) is the state vector, \(u = (u_1, u_2, u_m)\) is the perturbation (parameters, control) vector, \(t\) is the time. We have \(x \in X, u \in U, f : X \times U \times R \to X\). Here, \(X\) is the state space, \(U\) is the control space and \(R\) is the real number space. We restrict the considerations to the case \(X = R^n, \ U = R^m, \ R = R^1, \ R^k\) being the real Euclidean k-dimensional space. Let \(t \in I = [0, T]\), and \(G\) be the space of all measurable functions \(u : I \to R^m\).

Now, consider a variation \(\delta u\) of \(u\) and a perturbed control \(u'\). The variation is a function of time, so that \(u'(t) = u(t) + \delta u(t)\) for all \(t \in I\). The solution to Eq. (2) over \(I\), with given initial condition \(x = x_0\) and given
function $u(*)$ will be called a trajectory of Eq. (2). Thus, any component $x_k$ of the final value of $x(t)$ depends on the shape of the whole function $u(*)$. In other words, $x_k(t) = x_k(t)[u']$ (for any fixed $t$) is a functional (not a function) of $d(*)$. Unlike a function, in our case, a functional is a mapping from the space $G$ to $R$. Denote $\delta x_k = x_k[u + \delta u] - x_k[u] = x_k[u'] - x_k[u]$.

In this paper, the local functional sensitivity is defined as

$$S_k = \frac{\delta x_k}{\delta u_0}$$

(3)

Note the difference between the conventional local sensitivity Eq. (1) and the functional sensitivity Eq. (2). The notation $\delta u$ is nothing new, it denotes the variation of the function $u$, as defined in the calculus of variations $[8, 14]$.

The Eq. (2) defines a local property of the trajectory $x_k(t)$. Here, we are interested rather in the response of the model to perturbations that are not necessarily small. We will not enter in the methodology of the variational calculus. Our task is to define the functional sensitivity defined as the set of the graphs of all trajectories of Eq. (2), where $u = u_0 + \Delta u$. Here, $\Delta u(t)$ is a limited perturbation, not necessarily small. This is equivalent to say that $u(t)$ belongs to a set of restrictions $C(t)$, $u(t) \in C(t)$, for all $t \in I$. Here, $C(t)$ is a subset of $R^m$. The functional sensitivity defined this way is non-local. We do not use the term “global”, because this is not a global property of the model. We only do not require the perturbation to be small.

### 3 Differential inclusions

It might appear that to calculate the set of functional sensitivity we could integrate a series of trajectories, each of them with input $u$ charged with limited random fluctuations applied in each integration step, and then see the set reached by the trajectories. Unfortunately, this is not the case. Trajectories with perturbations generated in such way concentrate in a narrow spot inside the reachable set, providing wrong results. On the other hand, if we generate trajectories with perturbations constant over each one, the results are also poor and do not provide the true reachable set. These observations will be discussed in the following sections.

#### 3.1 Basic properties and definitions

In this section you can find some remarks on the early works on differential inclusions. More detailed assumptions and a comprehensive survey can be found in [11]. Differential Inclusion (DI) is defined by the following statement.

$$\frac{dx}{dt} \in F(t, x(t)), \quad x(0) \in X_0$$

(4)

where $x \in R^n$, $t$ is a real variable representing the time, $F$ is a mapping from $RxR^n$ to subsets of $R^n$ and $X_0 \subset R^n$ is the initial set. $R^n$ is the real n-dimensional Euclidean space. In the following, $R$ is equal to $R^1$. $F$ is also called the set of admissible directions. Note then here we delimitate the discussion to the real Euclidean space. In more advanced research on differential inclusions, the state variable may be considered an element of a more abstract space. In the following, the abbreviation a.e. on interval $I$ means “almost everywhere,” i.e. everywhere in $I$, except a set of measure zero. A set of measure zero defined over an interval of $R$ is a set which can be covered by a finite or enumerable sequence of open intervals whose total length (i.e., the sum of the individual lengths) is arbitrary small. A function $y : R^n \rightarrow R^n$ satisfies the Lipschitz condition if such real constant $L$ exists that for all $z_1, z_2 \in R^n$

$$|y(z_1) - y(z_2)| \leq L |z_1 - z_2|$$

(5)

where the real constant $L$ is independent on $z_1$ and $z_2$. The Lipschitz condition plays important role in the problems of existence and uniqueness of the solutions to differential equations. Such function $y$ is said to be Lipschitzian.
A function \( y : [a, b] \to \mathbb{R}^n \) is said to be absolutely continuous if for every \( \varepsilon > 0 \) a number \( \delta \) exists, such that for any finite collection of disjoint sub-intervals \([\alpha_k, \beta_k]\) of \([\alpha, \beta]\) satisfying the inequality
\[
\sum (\beta_k - \alpha_k) < \delta, \quad \text{we have} \quad \sum |y(\beta_k) - y(\alpha_k)| < \varepsilon.
\]
An absolutely continuous function is continuous.

Consider a differential equation
\[
\frac{dx}{dt} = f(t, x(t)), \quad t \in [t_o, t_1], t_o < t_1
\] (6)

with Lipschitz-continuous right-hand side. A solution to this equation is a function \( x(t) \) that is absolutely continuous, measurable and differentiable almost everywhere and fulfills the equation a.e. over a given time interval \( I \).

In the case of a differential inclusion Eq. (4), we will call the function \( x(t) \) a trajectory of the DI, if it satisfies Eq. (4) a.e. over the interval under consideration. The trajectory must be absolutely continuous and almost everywhere differentiable function.

Roughly speaking, the requirement that \( f \) is Lipschitzian is needed to guarantee that there exists the unique solution to Eq. (5).

The reachable set of a DI is the union of the graphs of all trajectories of the DI. The term emission zone has been used in early works. Here, we will rather use the term reachable or attainable set. In many works on the DIs, the mapping \( F \) is called a field of permissible directions. A trajectory of the DI is also called a trajectory of the field \( F \).

Let us comment on the term “solution to the DI”. It is commonly understood that a trajectory of the DI is a solution. Observe that a DI normally has an infinite number of trajectories, so the trajectory cannot be just called “the solution”. Our point is that THE solution to a differential inclusion is given by its reachable set. If we consider a sequence of DIs with shrinking right-hand side that, in the limit, degenerates to a single-valued function, then the corresponding sequence of reachable sets tends to the graph of the solution of the resulting differential equation. This is an argument to call the reachable set the solution to the DI. However, to avoid any ambiguity and conflict of terms, the term “solution to a DI” will not be used in following sections. Instead, we will discuss trajectories and reachable or attainable sets.

3.2 Possible applications: uncertainty, differential games and optimal control

There exists a strange conviction among many simulationists who deal with continuous simulation that everything that happens in the real world can be described by differential equations. This approach to modeling is somewhat dangerous because it forces the modelers to look for something (differential equation model) that might not exist at all. There are many theoretical results in the field of DI-based models, but little has been done in the area of numerical methods and practical applications. The solution to a DI (the reachable set) needs hundreds of single system trajectory evaluations, which makes the whole task computationally expensive. Another challenge is the representation of the results (N-dimensional sets and the boundary surfaces).

The absence of reliable data in computer simulation is an important obstacle in many simulation projects. In the past, a common way to treat this lack of exact data was to suppose that some model parameters or input variables are random ones. This approach results in a stochastic model, where every realization of the system trajectory is different, and the problem is to determine the probability density function in the system space for certain time-sections, the variance, confidence intervals, etc.

Such stochastic analysis is interesting and sometimes useful, but not always possible. Some parameters of the model have “uncertain” values, and the model user may have no idea about their probabilistic behavior. More likely we are given an interval the uncertain parameter belongs to, instead of its probability distribution. Some external disturbances can fluctuate within certain intervals, and the task consists in obtaining the interval for some output variables. Frequently, the user wants to know a possible extreme value rather than the probability to reach them (recall the law of Murphy!). The uncertainty treatment in this paper has nothing, or very little, to do with “Monte Carlo” or stochastic simulation. The intervals we are looking for are not confidence intervals or other statistics.
The need of a new approach, different from the stochastic uncertainty treatment has recently been commented by some authors. The non-stochastic approach to uncertainty treatment is referred to as tychastic [1, 2]. The authors observe that the “tychastic viability measure of risk” is an evolutionary alternative to statistical measures [1]. As the example, the price intervals are used instead of probability distribution in the solvency capital requirement problem, dealing with evolution under uncertainty. The tychastic variables are used to obtain the corresponding properties for portfolios, the evaluation of the capital and the transaction rule [2].

Some problems of differential games can also be treated using DIs. A missile that follows a moving target will fail when the target trajectory escapes from the attainable set of the missile. This can be used to verify different strategies of both players involved in the game.

While modeling systems with uncertainties, an input signal or a disturbance is often an unknown function that belongs to a given restriction set. In this case, the system can be modelled by a corresponding differential inclusion (deterministic instead of stochastic model). Also, note that the uncertain parameter may represent erroneous information, intentionally inserted to the real system, which means that it cannot be treated as a random one. Other difficulties appear while generating multidimensional random variables with arbitrary density functions. On the other hand, using a DI model, we only need to know the restrictions for the uncertain signals. As the result, we get the system attainable set that can provide important information on a possible system behavior. While simulating robust control systems, we often want the system to remain in some permissible region of states, regardless the possible disturbances. The reachable sets of the corresponding DIs can provide solutions to such problems.

As for the control systems, DIs are closely related to the optimal control problem. From the theory of optimal control it is known that the optimal trajectories are those that belong to the boundary of the system reachable set. This relation to control systems and the application of some methods taken from the optimal control theory are discussed in the following section.

4 Differential inclusion solver

4.1 General remarks

While working with differential equations, one can find a huge number of numerical methods and software. On the other hand, for the DIs there is nearly nothing that could help a simulationist. The Differential Inclusion Solver is the result of an attempt to fill this gap in the simulation software. The basic version of the DI solver is not new. It was published in [9]. Here, we only recall the algorithm. In this paper, a new version of the solver and some new applications to the functional sensitivity are described.

The solver has been coded in the Embarcadero® Delphi. A limited stand-alone “.exe” version of the solver is available. It should be emphasized that our main goal is the RS determination and not optimization. The DI solver and the present problem statement should not be confused with the differential inclusions method used in the optimal control problems.

To avoid repetitions from earlier articles, we will not discuss here the algorithm of the DI Solver. The basic version of the algorithm has been published in [9]. In few words, the DI Solver generates a series of DI trajectories that scan the boundary, and not the interior of the reachable set. The differential inclusion is derived from the model state Eq. (6) that include the vector of uncertain parameters $u = (u_1, u_2, \ldots, u_m)$.

$$\frac{dx}{dt} = f(t, x(t), u(t)), \quad t \in I = [t_o, t_1], t_o < t_1$$  \hspace{1cm} (7)

The value of $u$ belongs to a given set of restrictions $C$. When $u$ scans the interior of the set $C$, the right-hand-side of Eq. (6) scans the set $F$, being the right-hand-side of the corresponding differential inclusion Eq. (4).

Shortly speaking, the solver algorithm uses some results from the optimal control theory. From the Pontryagins’ Principle of Maximum it is known that each model trajectory that reaches a point on the boundary of the reachable set at the final simulation time must belong to the boundary of this set for all earlier time instants.

WJMS email for contribution: submit@wjms.org.uk
Moreover, such trajectory must satisfy the Jacobi-Hamilton equations (the necessary condition). These equations involve a vector of auxiliary variables \( p = (p_1, p_2, p_3, \ldots, p_n) \). In the optimal control, the problem is that we have the initial conditions for the state vector \( x \), but the conditions for the vector \( p \) are given at the end of the trajectory, \( t = T \). If no analytic solution is available, then we must use an iterative procedure for so-called two-point-boundary-value (TPBV) problem. In our case, we are in better situation. Observe that starting with the given initial condition for \( x \) and with any initial condition for \( p \), we obtain a trajectory that scans the boundary of the reachable set. This means that we do not have to solve the TPBV problem. The algorithm generates a series of trajectories with randomly generated initial conditions \( p(0) \). After integrating a sufficient number of trajectories, we can see the shape of the reachable set boundary. In other words, we are looking for the mapping from \( MP : p(0) \rightarrow RS(T) \), where \( RS(T) \) is the boundary of the reachable set at \( t = T \). The problem is that the mapping \( MP \) may be extremely irregular even for a simple linear model. The solver algorithm uses certain heuristic procedure to avoid the “holes” of the final image [9].

4.2 Implementation, new version of di solver

There are two versions of the solver. To run the first, basic version (S1), the user must have the Delphi package installed on his/her computer. The solver has the form of a Delphi project that must be compiled and run from the Delphi package. This dependence on Delphi occurs because a part of the source code is the model to be analyzed. Namely, the user must provide a procedure with expressions of the right-hand-sides of Eq. (7), where the control \( u \) intervenes as an m-dimensional array. While defining the way the control array appears, the use also defines the limits for its components. Then, the whole project is compiled and executed. The use of S1 is somewhat complicated, and requires some basic skills in Delphi coding.

New version of the solver (S2) is a stand-alone independent application. S2 includes an editor for the right-hand-sides of the model equations. The user introduces the code for the expressions, and the program compiles them and runs. Thus, the S2 version includes its own compiler for arithmetic expressions. The compiler converts the model code in an internal data structure that is used to quickly calculate the model right-hand sides. Obviously, such way to integrate the model equations is rather slow, compared with the code generated by the S1 version that uses the original Delphi compiler. The examples shown in the following section have been obtained from S1 version. The benchmarks show that the final results (reachable sets) provided by S1 and S2 versions are identical to each other.

5 Examples

The reachable sets of differential inclusions may have various applications. In this paper, we focus on the non-local functional sensitivity. As stated before, the functional sensitivity is defined as the attainable set of model states, according to the parameter uncertainty. Thus, the terms reachable or attainable set (RS) and functional sensitivity are treated here as synonyms.

5.1 Example 1

Consider a simple non-linear model of the second order:

\[
\begin{align*}
\frac{dx_1}{dt} &= x_2 \\
\frac{dx_2}{dt} &= 1 - 0.2u_1 - x_1 - 0.1u_2(x_2 + 2.3x_2^2)
\end{align*}
\]

(8)

where \( u_1 \) and \( u_2 \) are uncertain parameters. Let the parameter \( u_1 \) fluctuate between \(-0.2\) and \(+0.2\), and parameter \( u_2 \) fluctuate between \(0.025\) and \(0.175\). The initial conditions are \( x_1 = x_2 = 0 \), and the final simulation time is equal to 10.

Fig.1 shows the 3D image of the model RS. The three axes of the plot represent \( x_1, x_2 \) and the time. The image was generated by the DI solver.

It can be seen that even for relatively small perturbation \( u_1 \) and \( u_2 \), the deviation of the state vector may be quite big.

WJMS email for subscription: info@wjms.org.uk
Fig. 1: Functional sensitivity of model (8). The reachable set.

Fig. 2 A depicts the comparison of the functional sensitivity to the conventional sensitivity analysis provided in some system dynamics packages. The contour indicates the boundary of the reachable set for the model (8) at $t = 10$. These are end points of about 2000 boundary-scanning trajectories. A small black region marked with X is the result of the “Vensim-like” sensitivity analysis, where the parameters are constant along each trajectory. The region X was obtained by generating 50000 trajectories with the same limits of uncertain parameters. At the part B of the figure we can see the region Y, obtained by generating 50000 trajectories where the parameters can change the value within the same limits at each integration step.

Note a small part of the contour marked with Z. These are points generated by the solver that belong to the interior of the reachable set. Recall that the Maximum Principle provides the necessary and not sufficient conditions for the trajectory to be optimal.

Consequently, when the reachable set is more complicated and folds several times, we can obtain also such extra points.

Fig.3 shows a side view of the same reachable set, projected into the $x_1 - time$ plane. Note that the functional analysis region coincides with the conventional results for the initial time interval (0 - 35), but it is very different from the true reachable set for greater time interval.

5.2 Example 2

Fig.4 shows a simple closed loop control system. The controller is of of proportional-ingular action, and the process is of second order, oscillating. The set-point is denoted as $p$, $e$ is the control error, $v$ and $y$ are the process input and output, respectively. The control error $e$ is equal to $p - y$. We will simulate the response $y$ to the set point equal to the unit step function.

The equation of the controller is as follows. $K_R$ is the controller gain, and $T_i$ is the integrator parameter.

$$v = K_R \left( e + \frac{1}{T_i} \int_0^t e(\tau)d\tau \right), e = p - y \tag{9}$$

The process is described by the following equation.
Fig. 2: Final contour of the reachable set. Comparison with the simple shooting.

Fig. 3: Functional analysis projected into the x1-time plane.

Fig. 4: A control system with PI controller and a second order process.

\[ a \frac{d^2 y}{dt^2} + b \frac{dy}{dt} + cy = K_P v \]  

(10)
Calculating the derivative of both sides of Eq. (9) we obtain
\[
\frac{dv}{dt} = K_R \left( \frac{de}{dt} + \frac{e(t)}{T_i} \right)
\]

(11)

Deriving both sides of Eq. (11) and substituting the expression for \( dv/dt \) and \( e \), we have
\[
a \frac{d^3y}{dt^3} + b \frac{d^2y}{dt^2} + c \frac{dy}{dt} = K_P K_R \left( \frac{d(p - y)}{dt} + \frac{p - y}{T_i} \right)
\]

(12)

Recall that the time-derivative of the step function \( p(t) \) is equal to Dirac’s pulse function \( \delta(t) \). Now, we denote:

Thus, we have
\[
\begin{align*}
\frac{dx_1}{dt} &= x_2 \\
\frac{dx_2}{dt} &= x_3 \\
\frac{dx_3}{dt} &= \frac{1}{a} \left[ -bx_3 - cx_2 + K_P K_R \left( \delta(t) - x_2 + \frac{p - x_1}{T_i} \right) \right]
\end{align*}
\]

(13)

The third equation of Eq. (13) contains the function delta of Dirac. It might be difficult to simulate such function. Fortunately, we don’t need to do this. Recall that the integral of \( \delta(t) \) over any neighborhood of the point \( t=0 \) is equal to one, and the value of \( \delta(t) \) is equal to zero for any \( t \) different from zero. Now, consider the integral of this equation on the interval \([0, \varepsilon]\) with \( \varepsilon \) approaching zero. This integral, in the limit, will be equal to \( K_P K_R / a \) because all other terms in the expression are limited. Thus, if we integrate the equation over the (open on the left) interval \((0, T]\), then the initial condition for \( x_3 \) should be equal to \( K_P K_R / a \). This way, we eliminate the Dirac’s function from the corresponding simulation code.

Fig. 5 shows a simple simulation of our control system. The parameters are:
\[
K_R = 2, K_P = 1, T_l = 10, a = 10, b = 2, c = 1.
\]

Recall that \( x_1 = y \) is the process output, and \( x_2 \) is its’ derivative.

Now, suppose that the process parameters \( b \) and \( c \) are uncertain and can fluctuate by \( \pm 15\% \) around the original values. Such analysis may be useful while treating with robust control systems. The initial conditions are \( x_1(Q) = 0, x_2(0) = 0 \) and \( x_3(0) = K_P K_R / a \), and the final simulation time \( T = 100 \).

![Fig. 5: Simple simulation of control system of Fig.4, Eq. (13).](image-url)
In Fig. 6, we can see the projection of the reachable set into the plane $x_1, x_2$. Part A shows the cloud of end-points of the trajectories that belong to the boundary of the reachable set. It might appear that some points are inside the set, but remember that this is the projection of a 3-dimensional surface (like a balloon) onto a 2-dimensional plane. The reachable set was generated by the DI solver with approximately 2000 trajectories, where the parameters $b$ and $c$ fluctuated by ±15%.

Again, the part B shows the comparison of the functional sensitivity (reachable set) with the conventional Vensim-like analysis. The black region was obtained by simulating 50000 trajectories where the parameters $b$ and $c$ had random values within the same interval of ±15%, but constant along each trajectory.

![Fig. 6: Projection of the reachable set of Eq. (13) into the $x_1 - x_2$ plane. Comparison with the conventional sensitivity set. Images generated by the DI solver.](image)

6 Conclusion

The main point of this article is the definition of the local functional sensitivity in terms of the variational calculus rather than partial derivatives. This does not mean that we use the variational calculus methods. If we define this kind of sensitivity in non-local terms, as functional sensitivity, the concept leads to the reachable set of a differential inclusion. In this kind of sensitivity we permit that the perturbations are functions of model time, and not just parameters that are constant along each trajectory. To perform the functional sensitivity analysis, the DI solver must be used, to get the estimates of the reachable sets. The solver is based on some results from the optimal control theory, but our aim is sensitivity analysis and not optimization. The implementation of the solver is working satisfactory, but needs further improvements. The tool may fail for high-dimensional models, stiff equations and oscillatory models of order greater than three that involve two or more different proper frequencies. Comparing functional sensitivity with the conventional sensitivity analysis we can see that the obtained reachable sets almost always several times greater than that obtained using the classical approach.

More research on the reachable sets calculation should be done. This includes the applicability to distributed parameter systems, fluid dynamics and discrete time systems. The extension to distributed parameter systems seems to be difficult, mainly for the numerical tractability problems. Also the presentation of results for such systems may be rather difficult. As for the software, the new DI solver version that uses multi-processing, is being developed. Also, similar tools for discrete time and discrete state systems should be developed.
References


WJMS email for contribution: submit@wjms.org.uk