

“On-the-fly” parameter identification for dynamic systems control, using interval computations and reduced-order modeling

Leobardo Valera, Angel Garcia Contreras, and Martine Ceberio

Abstract Computer simulations of dynamic systems are really important to better understand some processes or phenomena without having to physically execute them, and/or to make offline decisions, or decisions that do not need immediate, “on-the-fly” answers in general. However, given a set of equations describing a dynamic phenomenon, wouldn’t it be nice to be able to exploit them more? Instead of simulating a situation, could we gear it or even veer it to a predefined performance? This paper is concerned with the identification of parameters of dynamic systems that ensure a specific performance or behavior. We propose to carry such computations using intervals and constraint solving techniques. However, realistically, aiming to enable such identification and decision on an on-going process or phenomena requires being able to conduct very fast computations on possibly very large systems of equations. We further propose to combine interval and constraint solving techniques with reduced-order modeling techniques to guarantee results in a practical amount of time.

1 INTRODUCTION

Computer simulations of dynamic systems are really important to better understand some processes or phenomena without having to physically execute them, and/or to make offline decisions, or decisions that do not need immediate, “on-the-fly” answers in general. These simulations use models derived from observations of natural phenomena, which are often very complex and involving millions of variables. Running simulations on these high-fidelity models yields significant CPU time issues. However, given a set of equations describing a dynamic phenomenon, wouldn’t it

Leobardo Valera, Angel Garcia Contreras, and Martine Ceberio
Computer Science Department, College of Engineering, The University of Texas at El Paso,
500 W. University Ave, El Paso, TX 79969-0518, USA. e-mail: lvalera@utep.edu, afgarciacon-
treras@miners.utep.edu, mceberio@utep.edu

be nice to be able to exploit them more? Instead of simulating a situation, could we gear it or even veer it to a predefined performance?

We propose to carry such computations using interval computations and constraint solving techniques. By using intervals, we account for the uncertainty from observations that are never 100% accurate. However, realistically, aiming to enable such identification and decision on an on-going process or phenomena requires being able to conduct very fast computations on possibly very large systems of equations. We further propose to combine interval and constraint solving techniques with reduced-order modeling techniques to guarantee results in a practical amount of time.

2 BACKGROUND

Modeling real-life phenomena can result in very large (most likely) nonlinear systems of equations. One way to solve these problems is to find the zeroes of large-dimensional functions using some real-valued solvers, e.g. Newton’s method. The convergence of the real-valued solvers depends on several factors: selection of the initial point, continuity of the partial derivatives, condition on the Jacobian or the Hessian matrix, among others. To overcome these issues, the solution can be sought on a subspace where the convergence conditions are met, hence also reducing the size of the problem to be solved: such general approach is called Reduced-Order Modeling. We review it in what follows (Subsection 1).

Another challenge with solving dynamic systems is that we often assume that the models are 100% accurate. This is seldom the case. Moreover, in the specific case that we tackle in this article, where we aim to react to observations (possibly a disruption) of an unwinding dynamic phenomenon by identifying new parameters to adapt it “on the fly”, observations are not 100% accurate and such uncertainty needs to be taken into account. As a result, if we are to solve such problems, we need to be able to handle uncertainty, and to quantify it, to be able to assess the quality of our solutions. Techniques that allow handling and quantifying uncertainty are reviewed in Subsection 2.2.

2.1 *Reduced-Order Modeling (ROM)*

The models used in simulations of real-life phenomena often consist in very large (most likely) nonlinear systems of equations: $F(X) = 0$, where $F : \mathbb{R}^n \rightarrow \mathbb{R}^n$. Such systems are called the Full-Order Model (or FOM) of a given problem. As very large problems can yield significant solving time, a common approach consists in decreasing/reducing the size of FOM, while remaining truthful to its original aim: this process is called Model-Order Reduction (MOR) and results in a Reduced-Order Model (ROM).

The main idea of ROM is to find an approximation to a solution \tilde{X} such that $\|F(\tilde{x})\|$ is sufficiently small in a k -dimensional subspace W of \mathbb{R}^n , where $k \ll n$. Common techniques for Model-Order Reduction consist of two stages: 1) finding the referred subspace and its corresponding basis Φ , here represented as an $n \times k$ matrix [9, 10]¹; and 2) finding Y^* , the solution of the overdetermined ($n \times k$) system $F(\Phi \cdot Y) = 0$, i.e., $Y^* = \min_W \{Y : F(\Phi \cdot Y) = 0\}$, where W is the spanned subspace of the columns of Φ . Once Y^* is found, the approximation \tilde{X} of X such that $\|F(X)\| = 0$ is determined by $\tilde{X} = \Phi \cdot Y^*$, see Fig. 1 [11, 12].

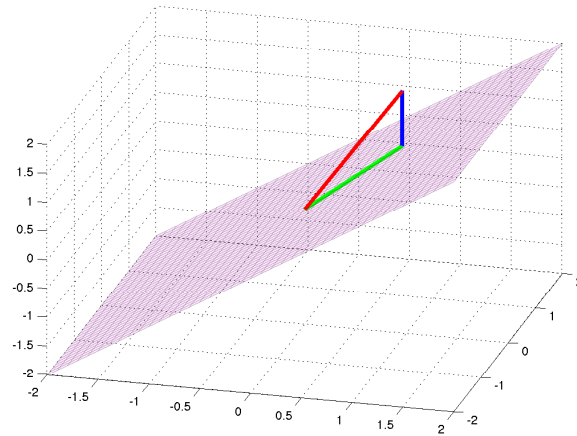


Fig. 1 Representation of ROM. The red vector represents the solution X . The approximation \tilde{X} , in green color, lies on the spanned subspace W of Φ , here represented by a plane

2.2 Interval Computations

In this article, we aim to address situations in which an unfolding dynamic phenomenon, for which we know F as well as all input parameters and other properties, is perturbed and requires recomputation of some parameters so as to ensure that some properties be satisfied (e.g., the below helicopter example where the landing zone is guaranteed even after perturbation, see Fig. 2). In general, if we priori restrict ourselves to a lower- dimensional space, we only get an approximation solution.

In the event of a perturbation, observations are essential to understanding the perturbation but observations are inherently inaccurate. As a result, if we

¹ In this paper, we assume that the basis Φ is given.

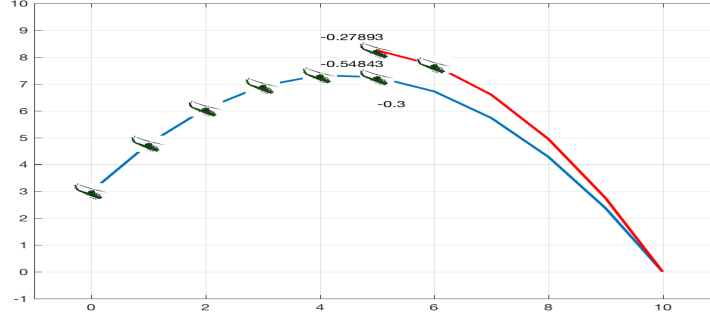


Fig. 2 Parameters of the flight are reliably recomputed to reach the landing zone after perturbation

are to solve such problems, we need to handle and quantify uncertainty to assess the quality of our solutions. We use interval computations to handle uncertainty.

In this paper, we propose to handle uncertainty as intervals. Whenever a quantity is not known for sure, e.g., observed value $v \pm \varepsilon$, we will represent this uncertainty as a closed interval: $[v - \varepsilon, v + \varepsilon]$, where given any real value r , \underline{r} is the largest floating-point number $\leq r$, and \bar{r} is the smallest floating-point number $\geq r$. Such floating-point-bounded intervals are carried in any computation originally involving seemingly 100% accurate real values, following interval arithmetic rules, generally described as follows:

$$\forall \text{ interval } X, Y, \forall \bowtie \in \{+, -, *, /\}, X \bowtie Y = Z \supseteq \square\{x \bowtie y \mid x \in X, y \in Y\} \quad (1)$$

where $\square A$ stands for the hull of set A , and Z is the smallest floating-point-bounded interval including $\square\{x \bowtie y \mid x \in X, y \in Y\}$.

How to Solve Nonlinear Equations with Intervals? The premise of our approach is to replace all real-valued computations with interval-based computations by abstracting real-valued parameters into interval parameters, and using interval constraint solving techniques to find solutions [3]. In our case, each of our nonlinear equations $f_i(x_1, \dots, x_n) = 0$ of the system to be solved is a constraint and our system of nonlinear equations a system of constraints. Our goal is to find values of its variables $\{x_1, \dots, x_n\} \in R$ that are such that: $\forall i, f_i(x_1, \dots, x_n) = 0$.

Constraint solving techniques allow us to identify all values of the parameters that satisfy the constraints. Interval constraint solving techniques [4, 5] produce a solution set (set of the solutions of the constraint system) that is interval in nature: it is a set of multi-dimensional intervals (or boxes whose dimension is n , the number of variables): this set is guaranteed to contain all the solutions of the constraint problem (in our case, of the nonlinear system of equations).

Most importantly, if the solving process returns no solution, we know for sure that it is because there is no solution. This guarantee of completeness provided by interval constraint solving techniques comes from the underlying solving mode: a branch-and-bound [6] approach (or branch-and-prune for faster convergence [7]) that uses the whole search space as a starting point and successively assess the likelihood of finding solutions in the given domain (via interval computations) and possibly (if Branch and Prune) reduce it, and discard domains that are guaranteed not to contain any solution.

Using ICST, it is possible to determine if $F(\Phi \cdot P) = 0$ has no solution, a unique solution, or many solutions in the subspace W defined by Φ (see [2, 3]).

3 Problem Statement and Proposed Approach

Let us recall the problem we want to solve. Using the model of a dynamic system, we aim to find certain parameter values that guarantee a specific outcome of the modeled dynamic phenomenon.

Assuming this phenomenon is modeled as a parametric differential equation (ODE / PDE), the parameters that lead to a certain outcome can be found as follows:

1. Discretize the ODE/PDE equation leading to a parametric system of equations $F(X, P) = 0$, where $X = (x_1, x_2, \dots, x_n)$ is the approximation of the solution and P are the parameters of the ODE/PDE equation, and $F : \mathbb{R}^n \rightarrow \mathbb{R}^n$.
2. Let $[i_1, i_2, \dots, i_m]$ a subset of $[1, 2, \dots, n]$ where n is the dimension. Fix the values of $x_{i_j} = [\underline{x}_{i_j}, \overline{x}_{i_j}]$, with $j = 1, 2, \dots, m$ representing the expected outcomes. Solve for P using ICST the following system:

$$\begin{aligned} \Phi(i_1, :)Y &= x_{i_1} = [\underline{x}_{i_1}, \overline{x}_{i_1}] \\ \Phi(i_2, :)Y &= x_{i_2} = [\underline{x}_{i_2}, \overline{x}_{i_2}] \\ &\dots \quad \dots \\ \Phi(i_m, :)Y &= x_{i_m} = [\underline{x}_{i_m}, \overline{x}_{i_m}] \\ F(\Phi Y, P) &= 0 \end{aligned}$$

Where $\Phi(i_j, :)$ is the i_j -row of Φ . The solutions P correspond to the sought parameters.

4 Experimental Results and Analysis

In this section, we report on preliminary experiments of our approach on one well-known problem: a particular case of the Lotka-Volterra problem. The Lotka-Volterra problem models a predator-prey system. We use the following equations to describe

this problem:

$$\begin{cases} v' = \theta_1 v(1-w), & v(0) = v_0 = 1.2 \\ w' = \theta_2 w(v-1), & w(0) = w_0 = 1.1 \end{cases} \quad (2)$$

where v and w respectively represent the number of preys and predators represented in thousands. The system was integrated from time $t_0 = 0$ to $t_m = 10$ with a constant step size $h = 0.1$. We used $\theta_1 = 3$ and $\theta_2 = 1$. Let us assume that at $t = 5$, a perturbation occurs, which changes the number of predators and preys. Since, it is not possible to know the new real number of animals of each species, the new number of both species is handled with uncertainty, i.e. $v(t = 6) = [0.8062, 0.8116]$, $w(t = 6) = [1.0834, 1.0884]$. Using ICST, it is possible to determine that, with $\theta_1 = [2.964, 3.039]$ and $\theta_2 = [0.9863, 1.014]$, we reach a balance of the two species at time $t = 10$, $v(t = 10) = [0.7675, 0.7738]$, $w(t = 10) = [0.9903, 1.0086]$, see Fig. 3.

In a similar setting, we were able to conclude that it is impossible to reach 1000 animal of each species by time $t = 10$, i.e. $v(t = 10) = w(t = 10) = 1000$.

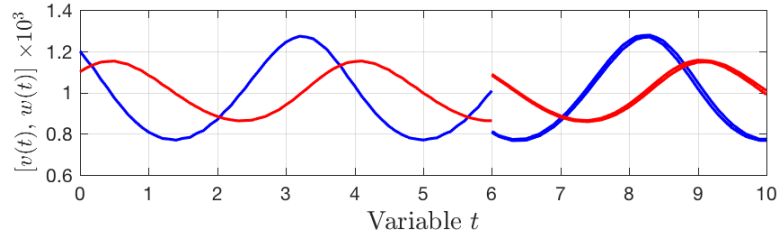


Fig. 3 Recomputation of θ_1 and θ_2 after perturbation at time $t = 6$

5 CONCLUSIONS

In this article, we aimed to design a technique that allows to identify parameters of a given (known and observed) dynamic system that has been perturbed, in such a way that some final conditions still hold. We used Reduced-Order Modeling and interval constraint solving techniques to determine such values of the phenomenon's parameters.

We were able to identify reliable intervals in which the desired parameters' values lie. We improved the runtime of this method by using ROM.

Future work includes taking into account the recomputation time and not assuming that new behavior can be "plugged" directly from where perturbation happened. As a result, more uncertainty needs to be taken into account, which includes time uncertainty. Additionally, we plan to consider perturbations as fuzzy numbers. This will require us to consider our dynamic system with uncertainty differently (e.g.,

with fuzzy derivatives). But most importantly, this is expected to help us make more informed decision: if we can label our input uncertainty with fuzzy values, how does this inform us about labels on uncertain solutions to focus on the best ones? [13].

ACKNOWLEDGMENT

This work was supported by Stanford's Army High-Performance Computing Research Center funded by the Army Research Lab, and by the National Science Foundation award #0953339.

References

1. Moore, R. E., Kearfott, R. B., and Cloud, M. J., *Introduction to Interval Analysis, 1st edition*, SIAM, Philadelphia, 2009.
2. Benhamou, F., Goualard, F., Granvilliers, V., and Puget, J., *Revising Hull and Box Consistency*. Proceedings of the 1999 International Conference on Logic Programming. MIT Press, 1999
3. Granvilliers, L. and Benhamou, F., *RealPaver: An Interval Solver using Constraint Satisfaction Techniques*. ACM Trans. on Mathematical Software 32(1), 138–156, 2006.
4. Mackworth, A. K., *Consistency in Networks of Relations* Artificial Intelligence, 8, 1, 99–118, 1977.
5. Jaffar, J. and Maher, M., *Constraint Logic Programming: a Survey* The Journal of Logic Programming, 19/20, 503–58, 1994.
6. Kearfott, R. B., *Verified Branch and Bound for Singular Linear and Nonlinear programs: An epsilon-inflation process*, April 2007. Available from the author.
7. Caroa, S., Chablata, S., Goldsztejn, A., Ishii, D., and Jermann, C., *A branch and Prune Algorithm for the Computation of Generalized Aspects of Parallel Robots* Artificial Intelligence, 211, 34, 2014.
8. Kearfott, R. B. and Kreinovich, V., *Where to bisect a box? A Theoretical Explanation of the Experimental Results*, In: G. Alefeld and R.A. Trejo (eds.), Proc. MEXICON'98, Workshop on Interval Computations, 4th World Congress on Expert Systems, Mexico City, México, 1998.
9. Schilders, W. H. and Vorst, H. A., *Model Order Reduction: Theory, Research Aspects and Applications*. Springer Science & Business Media, 2008.
10. White, J., *A Trajectory Piecewise-Linear Approach to Model Order Reduction and Fast Simulation of Nonlinear Circuits and Micromachined Devices* IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 22(2), 155–170, 2003.
11. Kelly, C. T., *Reduction of Model Order Based on Proper Orthogonal Decomposition for Lithium-Ion Battery Simulations*, Journal of the Electrochemical Society, 156 (2009), pp. A154–A161.
12. Willcox, K. B. and Peraire, J., *Balanced Model Reduction Via the Proper Orthogonal Decomposition*. AIAA journal, 40(11):2323–2330, 2002.
13. Lodwick, A., Bassanezy, R. C., and de Barros, L., *A First Course in Fuzzy Logic, Fuzzy Dynamical Systems, and Biomathematics. Theory and Applications*, Studies in Fuzziness and Soft Computing, Springer, Vol 347, Germany, 2017.