Towards Predictions of Large Dynamic Systems' Behavior using Reduced-Order Modeling and Interval Computations*

Leobardo Valera[1], Angel Garcia[1], Afshin Gholamy[2] Martine Ceberio[1] and Horacio Florez[1]

Abstract— **The ability to conduct fast and reliable simulations of dynamic systems is of special interest to many fields of operations. Such simulations can be very complex and, to be thorough, involve millions of variables, making it prohibitive in CPU time to run repeatedly for many different configurations. Reduced-Order Modeling (ROM) provides a concrete way to handle such complex simulations using a realistic amount of resources. However, uncertainty is hardly taken into account. Changes in the definition of a model, for instance, could have dramatic effects on the outcome of simulations. Therefore, neither reduced models nor initial conclusions could be 100% relied upon. In this research, Interval Constraint Solving Techniques (ICST) are employed to handle and quantify uncertainty. The goal is to identify key features of a given dynamical phenomenon in order to be able to propagate the characteristics of the model forward and predict its future behavior to obtain 100% guaranteed results. This is specifically important in applications, as a reliable understanding of a developing situation could allow for preventative or palliative measures before a situation aggravates.**

I. INTRODUCTION

The ability to make observations of natural phenomena has played a fundamental role in our world. From what we observe, we simulate models to help us understand how these phenomena vary over time and predict their future behavior or characteristics. Differential equations (PDEs or ODEs) help us represent the time-dependance of these dynamic systems. Numerical approaches to solving these differential equations require discretization, which could involve millions of variables, and as a result, so-called high-fidelity models yield significant CPU time issues. To overcome these issues, Reduced-Order Modelling (ROM) strategies are employed, which allow searching for the solutions of a given problem in a subspace whose dimension is much smaller than the dimension of the high-fidelity original model.

In addition, when we do not initiate simulations but rather we are simply making an observation (or observations) of an unfolding phenomenon, it is valuable to be able to understand it "on the fly" and predict its future behavior. In such situations, observations provide values of the variables of the high-fidelity model (or Full-Order Model, FOM) while we wish we could use ROM to truly conduct "on-the-fly" computations. In addition, things can also get challenging as observations are never 100% accurate and therefore we must also deal with uncertainty. This uncertainty propagates over time in dynamic systems, which could impact their future behavior and characteristics.

In this article, we show how understanding a dynamic phenomenon "on the fly" is possible, despite its possibly large size and embedded uncertainty. We demonstrate how we can translate FOM data into ROM data and combine Interval Constraint Solving Techniques (ICST) with Reduced-Order Modeling techniques to properly account for both challenges: size and uncertainty.

II. BACKGROUND

Modelling real-life phenomena can result in very large (most likely) nonlinear systems of equations that need to be solved. 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 [14], [15]. We review it in what follows (subsection II-A).

Another challenge with solving dynamical 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 want to identify the type of dynamical phenomenon that we observe, we have to deal with the inherent inaccuracy of observations. 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 Subsections II-A.1 and II-A.2.

A. Reduced-Order Modeling (ROM)

Let us consider a system of equations:

$$F(x) = 0 \tag{1}$$

where, $F : \mathbb{R}^n \to \mathbb{R}^n$. The main idea of ROM is to find a solution x in a subspace $W \subset \mathbb{R}^n$, whose dimension $k \ll n$.

There exists a $n \times k$ matrix Φ, whose columns are the basis vectors of W. This matrix can be redefined as an operator

$$T : \mathbb{R}^k \to \mathbb{R}^n$$

*This work was supported by The Army Research Lab and The National Science Foundation, CCF award 0953339

[1] Leobardo Valera, Angel Garcia and Martine Ceberio are with the Computer Science Department. [2]Afshin Gholamy is with the Civil Engineering Department, College of Engineering, The University of Texas at El Paso, 500 W. University Ave, El Paso, TX 79969-0518, USA {lvalera,mceberio}@utep.edu, afgarciacontreras@miners.utep.edu, agholamysalehabady@miners.utep.edu

where $T(y) = \Phi y$. Since $x \in W$, it can be defined as the linear combination of the columns of Φ:

$$x = \Phi y \tag{2}$$

Substituting (2) in (1), the nonlinear system becomes overdetermined.

$$\bar{F}(y) = F(\Phi y) = 0$$

$$\begin{tikzcd} \mathbb{R}^k \arrow[r, "T"] \arrow[dr, "\bar{F}"'] & \mathbb{R}^n \arrow[d, "F"] \\ & \mathbb{R}^n \end{tikzcd}$$

Let us illustrate the ROM through the following example.

Example 1: Consider the following nonlinear system of equations:

$$\begin{cases} (x_1^2 + x_1 - 2)(x_2^2 + 1) &= 0 \\ (x_2^2 - 5x_2 + 6)(x_1^2 + 1) &= 0 \\ (x_3^2 - 2x_3 - 3)(x_4^2 + 1) &= 0 \\ (x_4^2 - 4)(x_3^2 + 1) &= 0 \end{cases} \tag{3}$$

We can reduce (3) searching a solution in the subspace W spanned by $\Phi = \{(2,4,-2,0)^T; (0,0,0,-4)^T\}$.

$$\begin{cases} (16y_1^2 + 1)(4y_1^2 + 2y_1 - 2) &= 0 \\ (4y_1^2 + 1)(16y_1^2 + 20y_1 + 6) &= 0 \\ (16y_2^2 + 1)(4y_1^2 + 4y_1 - 3) &= 0 \\ (4y_1^2 + 1)(16y_2^2 - 4) &= 0 \end{cases} \tag{4}$$

The system (4) has two solutions $Y_1 = (0.5, 0.5)^T$ and $Y_2 = (0.5, -0.5)^T$. We can obtain the solutions of (3) by plugging Y_1 and Y_2 into (2). Note that the nonlinear system (3) has 16 solutions, but only two of those is given by ROM.

For more information on how to obtain a reduced basis on a nonlinear problems refer to [9], [10], [11], [12].

1) Computations with Intervals: Let us make a couple of adjustments. Hereafter, we will be using intervals to refer to *closed real-value bounded intervals* as opposed to the more commonly used floating-point-bounded intervals.

In this work, an interval X is defined as follows:

$$X = [\underline{X}, \overline{X}] = \{x \in \mathbb{R} : \underline{X} \leq x \leq \overline{X}\}. \tag{5}$$

Since $x \in X$ is equal to $\underline{X} \leq x \leq \overline{X}$, and $y \in Y$ is defined as $\underline{Y} \leq y \leq \overline{Y}$ the following operations are inferred based on its infimum and supremum:

- **Addition:** $X + Y = [\underline{X} + \underline{Y}, \overline{X} + \overline{Y}]$
- **Substraction:** $X - Y = [\underline{X} - \overline{Y}, \overline{X} - \underline{Y}]$
- **Multiplication:** $X \cdot Y = [\min S, \max S]$, where $S = \{\underline{X}\underline{Y}, \underline{X}\overline{Y}, \overline{X}\underline{Y}, \overline{X}\overline{Y}\}$

As it is shown, combining intervals with addition, subtraction, and multiplication, results in another interval. However, this is not always the case. For instance, the division of an interval by another one that contains 0 results in two disjoint intervals. To avoid such cases that compromise the nature of traditional interval computations (according to which combining intervals should result in an interval), we generalize the operations of two intervals as follows: $\forall X, Y$ intervals,

$$X \diamond Y = \Box\{x \diamond y, \text{ where } x \in X \text{ and } y \in Y\} \tag{6}$$

where $\diamond$ stands for any arithmetic operator, including division, and $\Box$ represents the hull operator.

Essentially, when carrying out more general computations involving intervals, e.g., computing the interval value of a given function $f : \mathbb{R}^n \to \mathbb{R}$ on interval parameters (or a mix of interval and real-valued parameters), we have the following property:

$$\begin{aligned} &f(X_1, \ldots, X_n) \subseteq \Box\{f(x_1, \ldots, x_n), \text{where} \\ &x_1 \in X_1, \ldots, x_n \in X_n\} \end{aligned} \tag{7}$$

where $f(X_1, \ldots, X_n)$ represents the range of function f over the domain $X_1 \times \cdots \times X_n$ and $\Box\{f(x_1, \ldots, x_n),$ where $x_1 \in X_1, \ldots, x_n \in X_n\}$ represents the smallest closed interval enclosing this range. Computing the exact range of f over intervals is therefore a very hard problem and instead, we approximate the range of f over domains using what we call an interval extension of f, which is in fact a surrogate interval function F.

Interval extensions of a given function f have to satisfy the following property:

$$f(X_1, \ldots, X_n) \subseteq F(X_1, \ldots, X_n) \tag{8}$$

which to some extent would allow F to be the function that maps any input to the interval $[-\infty, +\infty]$. More pragmatically, the aim is to identify a function F that does not dramatically overestimate the range of our original function f (the closer to the range the better of course, but cost of achieving better range is also an issue).

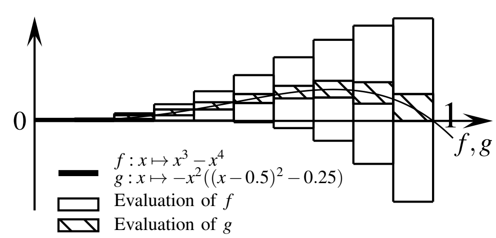

Fig. 1. Evaluation of the natural extensions of two expressions of the same real function f.

There are many interval extensions. The most common is the so-called natural extension, which is a simple interval extension of the syntactical expression of f: arithmetic operations are evaluated using interval rules as shown above, and any other single operator – e.g., power – has its own interval extension; see [1] for more details. Other extensions include Trombettoni et Al.'s occurrence grouping approach.

In general, two different interval extensions of the same real function f are different Fig. 1.

Important note: in practice, we conduct interval computations with floating-point-bounded intervals. What this changes is that each time an interval computation is carried, is the bounds of the resulting interval are not floating points, they are outward rounded (to the closest outward floating point), to guarantee that the range of the interval computation still be enclosed.

In this work, we use interval computations provided in RealPaver [3] and the natural extensions that this software provides.

2) 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. In this subsection, we give the reader an overview of the way we solve a nonlinear system of equations using interval computations. In particular, it does not involve picking a starting point, and therefore, is much more robust. It actually is guaranteed to be complete: all solutions will be retrieved, but most importantly, if there is no solution (our solver does not return solutions), we know for sure that it is because there is no solution, and it is not because our solver may or may not have failed to retrieve a solution.

We choose to solve nonlinear equations using interval constraint solving techniques. Constraint solving techniques allow to solve systems of constraints. Generally speaking, a constraint describes a relationship that its variables need to satisfy. A solution of a constraint is an assignment of values to its variables such that the relationship is satisfied.

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: it establishes a relationship that the values of the variables should satisfy, in this case so that $f_i(x_1,\dots,x_n)$ be equal to 0. Our system of nonlinear equations is therefore a system of constraints and our goal is to find values of the variables of this system 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).

The guarantee of completeness provided by interval constraint solving techniques comes from the underlying solving mode: a branch-and-bound [6] (or branch-and-prune for faster convergence [7]) approach that uses the whole search space as a starting point and successively assess the likeliness 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.

Example 2: Let us consider the following equation:

$$2x = y^2, \text{ with } x \in [-5,5],\, y \in [-2,2] \tag{9}$$

In order to solve this problem we have to following a process that can be divided in two stages: *Contraction* and *Bisection*.

- *Contraction:*
  - $x = [-5,5]$, then $2[-5,5] = [-10,10]$
  - $y = [-2,2]$, then $[-2,2]^2 = [0,4]$
  - since $2x = y^2$, we have to take the intersection: $[0,4] \cap [-10,10] = [0,4]$. Finally, using the inverse functions we determinate which the new values of x and y are.
  - $[0,4]/2 = [0,2]$, The new value of $x = [0,2]$.
  - $\sqrt{[0,4]} = [-2,2]$, then the new value of $y = [-2,2]$. In this case, there was not contraction in the variable y.
- *Bisection:* In this example, since there was not contraction in the variable y, we can consider y as a disjoint union, for example, $y = y_1 \cup y_2 = [-2,0] \cup [0,2]$, and repeat the process for x and y_1, and for x and y_2. A good criteria of bisecting can be found in [8].
- We repeat the whole process discarding the intervals no containing any solution and stopping when the width of the intervals are less that a tolerance previously defined.

Fig. 2 illustrate one step of contraction using the HC4-revise contractor [3].

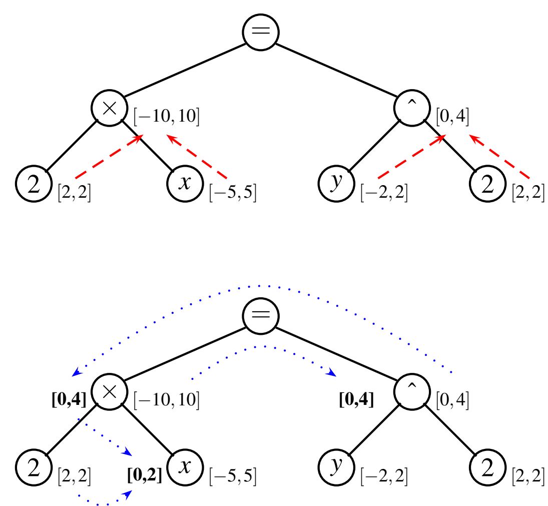

Fig. 2. One contraction step for (9) using ICST's HC4 contractor

III. Handling Uncertainty in Dynamical Systems

In this section, we do not set ourselves to solve simulations of dynamical systems whose input parameters and other features we know. We set ourselves to solve the corollary problem of identifying the "flavor" (features, parameters) of a known dynamical phenomenon, as it is unfolding. Such identifications are based on observations of the said phenomenon. Therefore, as observations are expected to be inaccurate, we aim to solve large dynamical systems involving uncertainty, on the fly.

In order to do this, we combine ROM techniques with ICST and observe how efficient our approach is to solve dynamic systems with some degree of uncertainty. We show how we can translate FOM observations into working ROM data. We also show that since ICST are based on solving constraints “independently of each other” (in the sense t we do not need a jacobian of the whole system to be a to run the solving techniques), we do not need to kn the initial conditions or the input parameters to solve system: we just need to have access to some informati which could be an observation, or observations, of the act phenomenon.

As a result, we show that our ability to handle uncertainty further allows us to make predictions. More specifically, we prove that even in the presence of uncertainty in some part of the data we are able to identify the values of the input parameters and unfold the dynamic behavior further in time.

A. *Uncertainty in observations*

Here, we simulate having observations (with uncertainty) about a given type of dynamical phenomenon, and we will quantify the relationship between the quality of data and the quality of the prediction. We set values $v(t_i) = [\underline{v_i}, \overline{v_i}]$ to simulate the uncertainty in some observations in times t_i. We then set to figure out the relationship between the number of data points with uncertainty and the accuracy of the prediction of the parameters. We illustrate our work on a set of examples.

1) Lotka-Volterra: Consider a particular case of Lotka-Volterra problem, which involves a model of predator-prey system. We use the following equations describe this problem:

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

where v and w respectively represent the amount of pr and predators. In this particular example, the growth rate of the first species reflects the effect the second species has on the population of the first species (θ_1). Similarly, the growth rate of the second species reflects the effect the first species has on the population of the second species (θ_2). The system was integrated from $t_0 = 0$ to $t_m = 10$. Numerical experiments were carried out with a constant step size $h = 0.1$. Ranges for the parameters $\theta_1 = [2.95, 3.05]$ and $\theta_2 = [0.95, 1.05]$ were used as input.

After discretization, the nonlinear system of differential equations (10) can be written as the function $F : \mathbb{R}^{2n} \to \mathbb{R}^{2n}$, with $n = 100$

$$F\begin{pmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \\ w_1 \\ w_2 \\ \vdots \\ w_n \end{pmatrix} = \begin{pmatrix} v_2 - v_0 - 2h\theta_1 v_1(1-w_1) \\ w_2 - w_0 - 2h\theta_2 w_1(v_1 - 1) \\ v_3 - v_3 - 2h\theta_1 v_2(1-w_2) \\ w_3 - w_1 - 2h\theta_2 w_2(v_2 - 1) \\ \vdots \\ v_n - v_{n-2} - 2h\theta_1 v_{n-1}(1-w_{n-1}) \\ w_n - w_{n-2} - 2h\theta_2 w_{n-1}(v_{n-1} - 1) \\ v_n - v_{n-1} - h\theta_1 v_n(1-w_n) \\ w_n - w_{n-1} - h\theta_2 w_n(v_n - 1) \end{pmatrix} \qquad (11)$$

To simplify the notation, let us denote $V = (v_1, v_2, \cdots, v_n, w_1, w_2, \cdots, w_n)^T$.

Firstly, we solve the nonlinear system of equations (11) $F(V) = 0$ using ICST. Due to the large number of unknowns and the uncertainty on θ_1 and θ_2, we obtain a solution with overestimation in each of the unknowns, see Fig. 3.

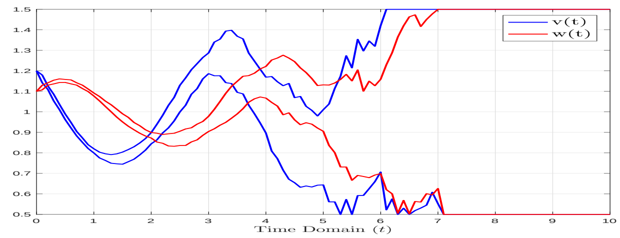

Fig. 3. ICST and Full-Order Modeling

To reduce the time it takes to run such predictions and to reduce at the same time the overestimation when (11) is solved, we search the solution of it on a reduced subspace W whose basis will be the columns of a matrix Φ, i.e., $V = \Phi p$.

$$F(\Phi p) = 0 \qquad (12)$$

Using ROM and ICST (12), the computing time is significantly reduced from 75ms to 17ms, which represents 22% of FOM runtime. This is due to smaller number of unknowns in ROM. see Fig. 4.

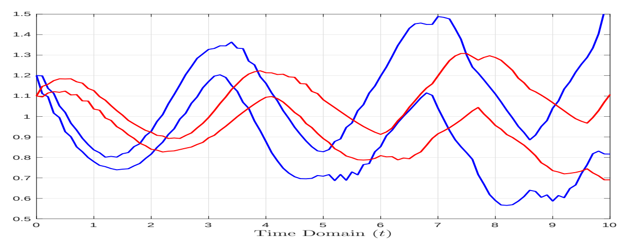

Fig. 4. ICST and Reduced-Order Modeling

We showed that we are able to obtain an enclosure of the solution of a nonlinear system of differential equations when we are dealing with uncertainty on one or more variables. The question that needs to be addressed is whether or not we could obtain the values of the parameters by knowing a basis Φ of a subspace W. In the following examples, let us set the parameters θ_1 and θ_2 as unknowns and v_{50} and w_{50} as known values:

$$\begin{aligned} \Phi(50,:)p &= v_{50} = [0.76, 0.78] \\ \Phi(150,:)p &= w_{50} = [0.96, 1.04] \\ F(\Phi p) &= 0 \end{aligned} \qquad (13)$$

where $\Phi(50,:)$ and $\Phi(150,:)$ are respectively rows 50 and 150 of the matrix Φ.

When (14) is solved the overestimation of the solution has been reduced, see Fig 5, and the values of parameters are close to the original ones $\theta_1 = [2.92, 3.05]$ and $\theta_2 = [1.048, 1.051]$.

Now we set two more data

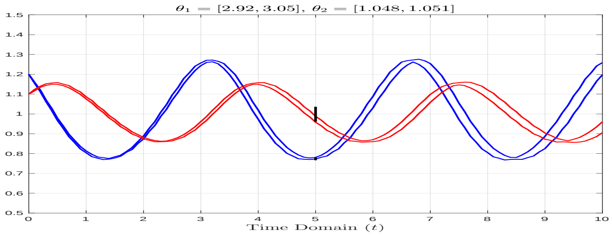

Fig. 5. Uncertainty associated with one observation

$$
\begin{array}{rcl}
\Phi(50,:)p & = & v_{50} = [0.76, 0.78] \\
\Phi(150,:)p & = & w_{50} = [0.96, 1.04] \\
\Phi(70,:)p & = & v_{70} = [1.20, 1.28] \\
\Phi(170,:)p & = & w_{70} = [0.97, 1.10] \\
F(\Phi p) & = & 0
\end{array} \quad (14)
$$

We see in Fig 6,

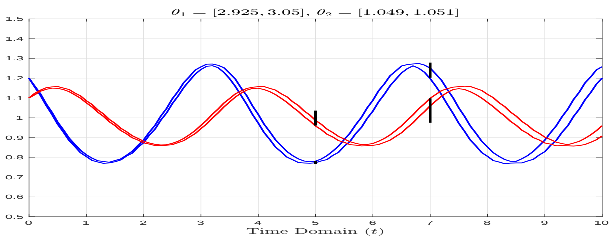

Fig. 6. Uncertainty associated with two observations

that the behavior of the solution does not change, and t value of the parameters barely change. $v_{50}, v_{70}, w_{50}, w_{70}$, now the value of the predicted values were $\theta_1 = [2.925, 3.05]$ and $\theta_2 = [1.049, 1.051]$.

The experiment is repeated several times. Each time a new observation is set. The first step is to set the observation corresponding with $t = 0.05$, next with $t = 0.05$ and $t = 0.1$, and so on, until having a total of 38 observations set. In TABLE I, it can be observed that the behavior of the values of the parameters θ_1 and θ_2 depends on the number of set observations. each row of the table corresponds to new observations that are added.

The following example illustrates how problems with uncertainty and highly sensitive parameters can still be handled with ICST.

2) Gaussian Model: Consider the σ-parameter differential equation

$$y' + \frac{x}{\sigma^2} y = 0, \; y(0) = \frac{1}{\sigma\sqrt{2\pi}} \quad (1$$

The analytical solutions of (15) is given for the family functions $f_\sigma(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{x^2}{2\sigma^2}}$. In Fig. 7 we have the gra of three members of the family of solutions correspondi to $\sigma = 1$, $\sigma = \sqrt{2}$, and $\sigma = \sqrt{3}$.

Although $\sqrt{2} \in [1, \sqrt{3}]$, the function $f_{\sqrt{2}}(x)$ is not bounded by $f_1(x)$ and $f_{\sqrt{3}}(x)$, so we cannot bound all the members of the family between the the graph of the solution corresponding to the lower bound and the upper bound of the parameters.

v_i, w_i	$[\theta_1, \overline{\theta_1}]$	$[\theta_2, \overline{\theta_2}]$
v_5, w_5	[2.949700,3.050500]	[1.049500,1.050500]
v_{10}, w_{10}	[2.949500,3.050800]	[1.049500,1.050500]
v_{15}, w_{15}	[2.949600,3.050300]	[1.049500,1.050500]
v_{20}, w_{20}	[2.950900,3.049000]	[1.049100,1.050900]
v_{25}, w_{25}	[2.951800,3.050000]	[1.049400,1.050600]
v_{30}, w_{30}	[2.951800,3.050000]	[1.049300,1.050800]
v_{35}, w_{35}	[2.952000,3.049700]	[1.049400,1.050600]
v_{40}, w_{40}	[2.949900,3.050500]	[1.049400,1.050600]
v_{45}, w_{45}	[2.950000,3.050400]	[1.049400,1.050600]
v_{50}, w_{50}	[2.950400,3.050000]	[1.049200,1.050800]
v_{55}, w_{55}	[2.950400,3.050000]	[1.049200,1.050800]
v_{60}, w_{60}	[2.951100,3.050200]	[1.049200,1.050800]
v_{65}, w_{65}	[2.951300,3.049300]	[1.049300,1.050700]
v_{70}, w_{70}	[2.951200,3.049300]	[1.049400,1.050600]
v_{75}, w_{75}	[2.951300,3.049900]	[1.049400,1.050600]
v_{80}, w_{80}	[2.951300,3.049900]	[1.049400,1.050600]
v_{85}, w_{85}	[2.950900,3.049900]	[1.049500,1.050500]
v_{90}, w_{90}	[2.950900,3.049900]	[1.049500,1.050500]
v_{95}, w_{95}	[2.950900,3.049800]	[1.049500,1.050500]

TABLE I

NUMBER OF VARIABLES WITH UNCERTAINTY VERSUS THE THE VALUE OF THE PARAMETERS θ_1 AND θ_2

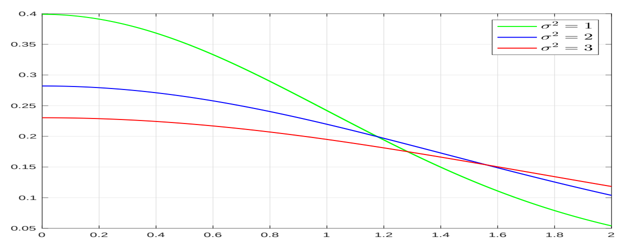

Fig. 7. Solution of (15) for $\sigma = 1$, $\sigma = \sqrt{2}$, and $\sigma = \sqrt{3}$

To know the behavior of all members of the family, we can solve the system of equations that comes from the discretization of (15) using ICST and taking $\sigma = [1, \sqrt{3}]$. We discretize the domain $[0, 2]$ in 200 observatiuons so we have to solve a linear system of 200 unknowns with 200 equations.

The black external lines in Fig. 8 correspond to the lower bound and the upper bound of the interval solution. Observe how the lower and upper bound enclose the solutions for $\sigma = 1$, $\sigma = \sqrt{2}$, and $\sigma = \sqrt{3}$. We have guaranteed that there is not a $\sigma \in [1, \sqrt{3}]$ that is not bounded by the interval solution.

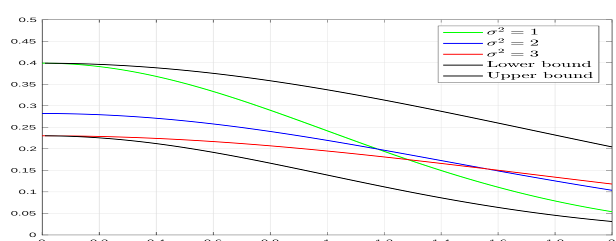

Fig. 8. interval Solution for $\sigma = [1, \sqrt{3}]$

Let us see how we can determine the parameters knowing the value of the function for some t. If we set the variable $y_{117} = [0.2003, 0.2036]$, which is close to the intersection of the members of the family corresponding to $\sigma = 1$ and $\sigma = \sqrt{2}$. When we solve the system of equations the parameter $\sigma = [0.8641975308641979, 2]$, which encloses $\sigma = \sqrt{2}$ and $\sigma = 1$, see Fig 9.

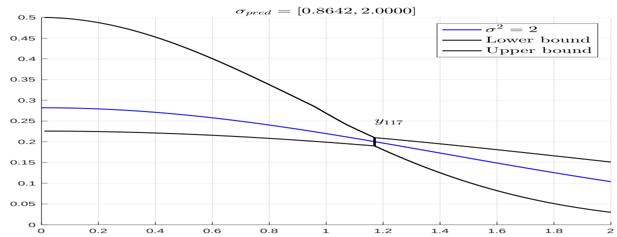

Fig. 9. Uncertainty associated with the observation $y_{117} = [0.2003, 0.2036]$

Then, if we also fix the variable $y_{156} = [0.1535, 01547]$, which is close to the intersection of the graphs of $f_{\sqrt{2}}(x)$ and $f_{\sqrt{3}}(x)$. In this case $\sigma = [1.372854139434509, 1.435761414626781]$, which encloses the value of $\sigma = \sqrt{2}$, see Fig 10.

Fig. 10. Uncertainty associated with the observations $y_{117} = [0.2003, 0.2036]$ and $y_{156} = [0.1535, 0.1547]$

Note: Because we used numerical constraint solving techniques [11], [12] to narrow down the possible values of the unknowns of the constraint systems we solves, our approach is sound regardless of whether the constraint system models a simulation (identifying the whole solution space vector); or for predictions (identifying the input parameters, the initial conditions, or any other value of interest as is our focus in this article).

IV. CONCLUSIONS

In this article, starting from the problem of solving large dynamical systems while handling uncertainty using Reduced-Order Modeling, we set to solving the corollary problem of identifying specific input parameters of unfolding dynamical systems under observation. We showed that we can, not only use ROM techniques, but also Interval Constraint Solving techniques to solve such problems. We demonstrated the ability of these techniques and the quality of the results in a few nonlinear problems chosen for the features they exhibit (nonlinearity, solving difficulty). We observed that non only we could translate observation data from a Full-Order Model into the solving process on a Reduced-Order Model and that uncertainty quantification did not suffer from this translation. Future work includes integrating outliers in observations and designing techniques to handle them.

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.

The authors would like to thank Professor Oscar Neira from the Universidad Metropolitana, Caracas, Venezuela, for his expert advice and suggestion of some examples presented in this paper.

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., Goldsztejnb, A., Ishiic, D., and Jermannd, 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. 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.

[14] Valera, L. Ceberio, M., *Model-Order Reduction Using Interval Constraint Solving Techniques*, Journal of Uncertainty Systems, Vol 11. 2017.

[15] Valera, L., Ceberio, M., *Model-Order Reduction Using Interval Constraint Solving Techniques*, 7th International Workshop on Reliable Engineering Computing (REC), 2016.