Predicting the Behavior of Dynamical Systems using Reduced-Order Modeling and Interval Computations

Journées de l'Optimisation, 7-9 Mai 2018

Martine Ceberio, Angel Garcia Contreras, and Leobardo Valera

Computer Science Department
The University of Texas at El Paso
mceberio@utep.edu

GENERAL OBJECTIVE

We are interested in dynamical phenomena: how to best make use of our knowledge of them?

GENERAL OBJECTIVE

We are interested in dynamical phenomena: how to best make use of our knowledge of them?

This is relevant to many areas:

- from understanding how a vehicle can withstand an underbody blast
- to understanding how a disease spreads depending on the number of affected people and the policies put in place for instance,
- to understanding how efficient a combustion system is, what performance different mixes of fuel yield
- *etc.*

GENERAL OBJECTIVE

We are interested in dynamical phenomena: how to best make use of our knowledge of them?

This is relevant to many areas:

- from understanding how a vehicle can withstand an underbody blast
- to understanding how a disease spreads depending on the number of affected people and the policies put in place for instance,
- to understanding how efficient a combustion system is, what performance different mixes of fuel yield
- *etc.*

In other words: wouldn't it be nice to be able to predict what could happen?

WHAT CAN WE DO?

Given a dynamical system, we use it to make **decisions**.

WHAT CAN WE DO?

Given a dynamical system, we use it to make **decisions**.

- What **types of decisions**?

WHAT CAN WE DO?

Given a dynamical system, we use it to make **decisions**.

- What **types of decisions**?
  - **Understanding** how a dynamic phenomenon unfolds under different input parameters: **simulations** $\longrightarrow$ e.g., design decisions

WHAT CAN WE DO?

Given a dynamical system, we use it to make **decisions**.

- What **types of decisions**?
  - **Understanding** how a dynamic phenomenon unfolds under different input parameters: **simulations** $\longrightarrow$ e.g., design decisions
  - Based on some knowledge of an unfolding phenomenon, **predicting** its behavior $\longrightarrow$ e.g., to allow preventive actions

WHAT CAN WE DO?

Given a dynamical system, we use it to make **decisions**.

- What **types of decisions**?
  - **Understanding** how a dynamic phenomenon unfolds under different input parameters: **simulations** $\longrightarrow$ e.g., design decisions
  - Based on some knowledge of an unfolding phenomenon, **predicting** its behavior $\longrightarrow$ e.g., to allow preventive actions
  - **Enforcing some behavior**, when control of input or other parameters is possible, and/or recomputing parameters on the fly $\longrightarrow$ e.g., to address an unexpected event and still guarantee an acceptable outcome of the situation

CHALLENGES

- **Size:** even if the original problem is not always very large, discretizing it potentially leads to large systems of equations

CHALLENGES

- **Size:** even if the original problem is not always very large, discretizing it potentially leads to large systems of equations
- **Complexity:** such problems are likely nonlinear, possibly non-smooth, and yet need to be solved

CHALLENGES

- **Size:** even if the original problem is not always very large, discretizing it potentially leads to large systems of equations
- **Complexity:** such problems are likely nonlinear, possibly non-smooth, and yet need to be solved
- **Uncertainty** in the dynamical systems we study / observe

CHALLENGES

- **Size:** even if the original problem is not always very large, discretizing it potentially leads to large systems of equations
- **Complexity:** such problems are likely nonlinear, possibly non-smooth, and yet need to be solved
- **Uncertainty** in the dynamical systems we study / observe
  - **Fuel combustion**: e.g., what decision can be made about the best nozzle geometry if the fuel mix is not known with certainty? Under fuel mix uncertainty, what design could limit pollutant emissions during training but maximize performance on the field?

CHALLENGES

- **Size:** even if the original problem is not always very large, discretizing it potentially leads to large systems of equations
- **Complexity:** such problems are likely nonlinear, possibly non-smooth, and yet need to be solved
- **Uncertainty** in the dynamical systems we study / observe
  - **Fuel combustion**: e.g., what decision can be made about the best nozzle geometry if the fuel mix is not known with certainty? Under fuel mix uncertainty, what design could limit pollutant emissions during training but maximize performance on the field?
  - **Trajectories:** e.g., of missiles. What if we could provide an envelope of a missile's trajectory under uncertainty of outside conditions (e.g., weather)?

WHAT WE CAN DO

- **Size:** $\longrightarrow$ this can be addressed by Reduced-Order Modeling (ROM) techniques

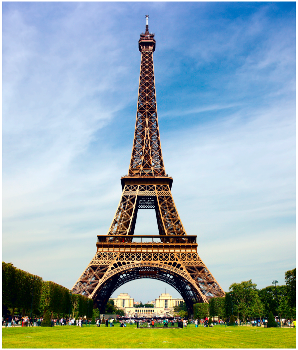
WHAT WE CAN DO

- **Size:** $\longrightarrow$ this can be addressed by Reduced-Order Modeling (ROM) techniques
- **Complexity and uncertainty:** we used interval computations and constraint solving techniques

PROBLEMS ARE TOO LARGE?

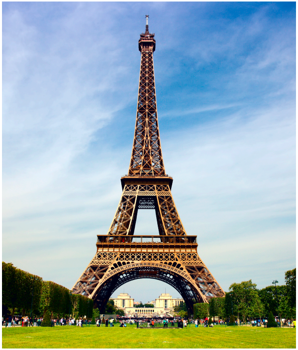
PROBLEMS ARE TOO LARGE?

If we can not take this home:



PROBLEMS ARE TOO LARGE?

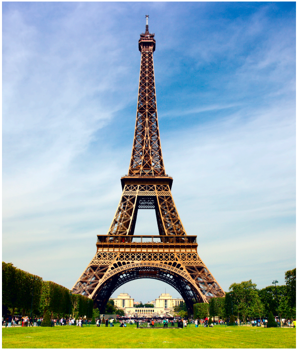
If we can not take this home:



At least, we can take this:

PROBLEMS ARE TOO LARGE?

If we can not take this home:

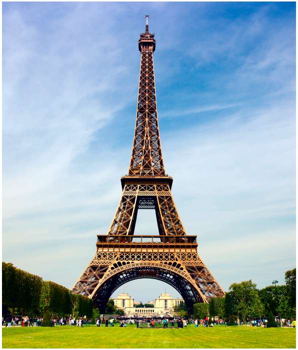


At least, we can take this:

This is what we call Model-Order Reduction (MOR).

PROBLEMS ARE TOO LARGE?

If we can not take this home:



At least, we can take this:

This is what we call Model-Order Reduction (MOR).
And from MOR, we get a Reduced-Order Model (ROM).

MODEL-ORDER REDUCTION (MOR)

Given the function

$$F : \mathbb{R}^n \longrightarrow \mathbb{R}^n$$

a nonlinear system of equations consists in finding x such that:

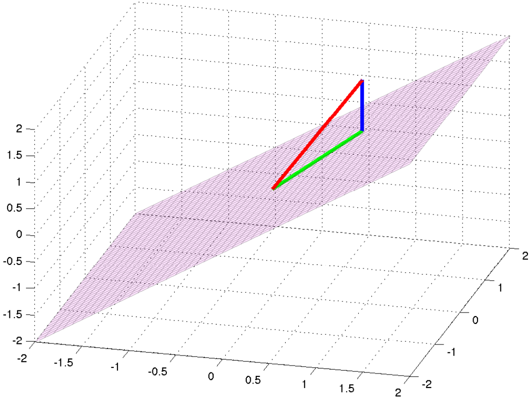
$$F(x) = 0$$

MOR is typically performed on the premise that the solution x belongs to an affine subspace, W, of $\mathbb{R}^n$ whose dimension k is orders of magnitude smaller than n, i.e.

$$x = z + \Phi p$$

where Φ is a basis of a subspace of $\mathbb{R}^n$ associated to W

ILLUSTRATION OF MOR



$$F : \mathbb{R}^n \longrightarrow \mathbb{R}^n$$

$$\begin{aligned} F(X) &= 0 \\ F(\Phi p + z) &= 0 \end{aligned}$$

Assuming

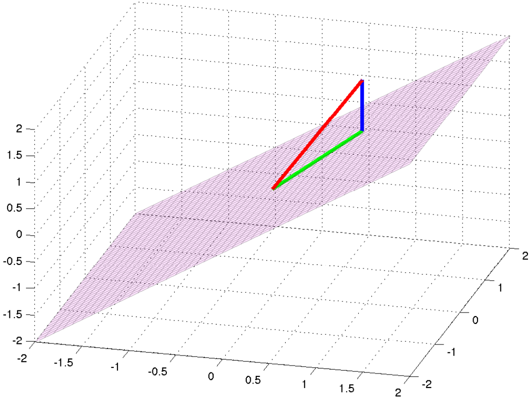
$$z \approx 0$$

we have to solve:

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

$$p = \arg\min \frac{1}{2} ||F(\Phi p)||^2$$

ILLUSTRATION OF MOR



$$F : \mathbb{R}^n \longrightarrow \mathbb{R}^n$$

$$\begin{aligned} F(X) &= 0 \\ F(\Phi p + z) &= 0 \end{aligned}$$

Assuming

$$z \approx 0$$

we have to solve:

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

$$p = \arg\min \frac{1}{2} ||F(\Phi p)||^2$$

$$\begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} \phi_{11} & \phi_{12} \\ \phi_{21} & \phi_{22} \\ \phi_{31} & \phi_{32} \end{pmatrix} \begin{pmatrix} p_1 \\ p_2 \end{pmatrix} + \begin{pmatrix} z_1 \\ z_2 \\ z_3 \end{pmatrix}$$

MOR BASICS

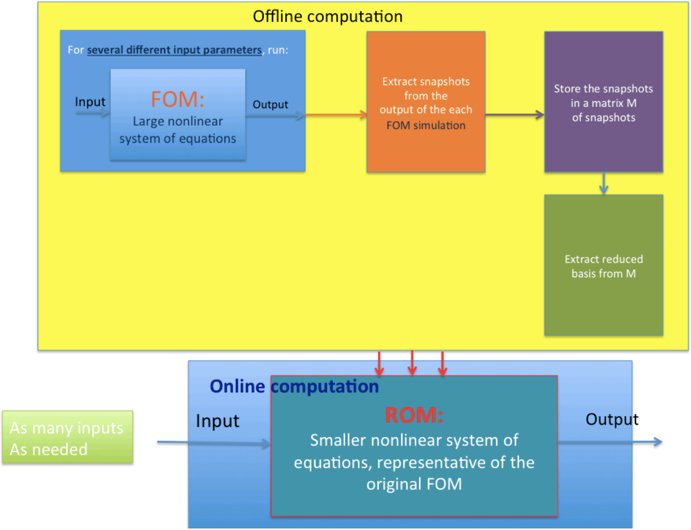
- The goal of Model Order Reduction (MOR) is to:
  - Reduce complexity
  - Maintain (input-output) accuracy
  - Maintain relevant physical properties
- A good Reduction methodology must be:
  - Accurate, efficient, numerically robust, and generate useful models

HOW TO GO ABOUT MOR?

The most common technique currently used to conduct Model-Order Reduction is **Proper Orthogonal Decomposition (POD)** **Chapman et al. 2016**

It is based on Principal Component Analysis (PCA), which is a procedure for identifying a smaller number of uncorrelated variables, called “principal components”, from a large set of data. The goal of PCA is to describe the maximum amount of variance with the fewest number of principal components.

HOW DOES MOR WORK?



- **Our approach.**

UNCERTAINTY

- **Our approach.**
  - We modeled uncertainty using **intervals** $\longrightarrow$ interval computations

UNCERTAINTY

- **Our approach.**
  - We modeled uncertainty using **intervals** $\longrightarrow$ interval computations

UNCERTAINTY

- **Our approach.**
  - We modeled uncertainty using **intervals** $\longrightarrow$ interval computations
- **Why?**

UNCERTAINTY

- **Our approach.**
  - We modeled uncertainty using **intervals** $\longrightarrow$ interval computations
- **Why?** Because interval algorithms are **reliable**: no solution is lost.

UNCERTAINTY

- **Our approach.**
  - We modeled uncertainty using **intervals** $\longrightarrow$ interval computations
- **Why?** Because interval algorithms are **reliable**: no solution is lost.
- **What this allowed us to do:**

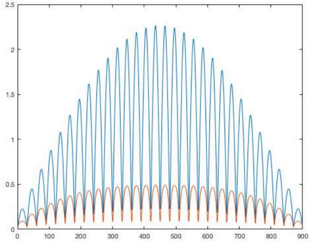
UNCERTAINTY

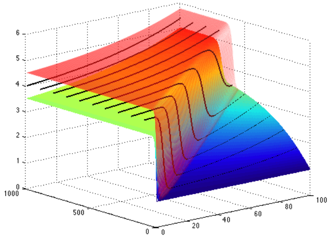
- **Our approach.**
  - We modeled uncertainty using **intervals** $\longrightarrow$ interval computations
- **Why?** Because interval algorithms are **reliable**: no solution is lost.
- **What this allowed us to do:**
  - **Simulations** with intervals: e.g., uncertainty in initial conditions, in input parameters, etc.
  - **Reduced-Order Modeling using interval computations**: to handle both the many snapshots and the possible uncertainty in other parameters / constants $\longrightarrow$ I-POD technique

UNCERTAINTY

- **Our approach.**
  - We modeled uncertainty using **intervals** $\longrightarrow$ interval computations
- **Why?** Because interval algorithms are **reliable**: no solution is lost.
- **What this allowed us to do:**
  - **Simulations** with intervals: e.g., uncertainty in initial conditions, in input parameters, etc.
  - **Reduced-Order Modeling using interval computations**: to handle both the many snapshots and the possible uncertainty in other parameters / constants $\longrightarrow$ I-POD technique

UNCERTAINTY: INTERVAL TECHNIQUES





USING INTERVAL IN SOLVING TECHNIQUES

Interval Branch-and-Bound and variations of it

It is the underlying principle of search in interval constraint solving techniques and it allows to guarantee completeness of the search.

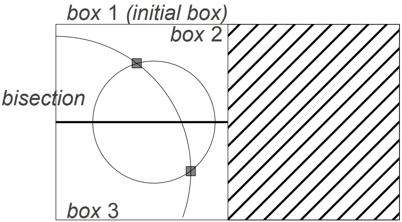


Fig.: L. Granvilliers, RealPaver User's Manual.

Algorithm

Input: System of constraints $C = \{c_1, \ldots, c_k\}$, a search space D_0.
Output: A set Sol of interval solutions

Set Sol to empty
If $\forall i$, $0 \in F_i(D_0)$ then:
  Store D_0 in some storage S
  While (S is not empty) do:
    Take D out of S
    If ($\forall i$, $0 \in F_i(D)$) then:
      If (D is still too large) then:
        Split D in D_1 and D_2
        Store D_1 and D_2 in S
      Else:
        Store D in Sol
Return Sol

PREDATOR-PREY EXAMPLE

The Lotka-Volterra problem models a predator-prey situation: e.g., foxes and rabbits

$$\begin{cases} y_1' = \theta_1 y_1 (1 - y_2), & y_1(0) = 1.2 & \theta_1 = 3, \\ y_2' = \theta_2 y_2 (y_1 - 1), & y_2(0) = 1.1 & \theta_2 = 1 \end{cases}$$

No analytic solution is available.

UNCERTAINTY: LOTKA-VOLTERRA

Let's assume that we know the initial populations: $y_1(0) = 1.2$ and $y_2(0) = 1.1$ but we do not know exactly θ_1 and θ_2: all we have are intervals of their potential values for instance.

$$\theta_1 = [2.95, 3.05] \quad \text{and} \quad \theta_2 = [0.95, 1.05]$$

UNCERTAINTY: LOTKA-VOLTERRA

Let's assume that we know the initial populations: $y_1(0) = 1.2$ and $y_2(0) = 1.1$ but we do not know exactly θ_1 and θ_2: all we have are intervals of their potential values for instance.

$$\theta_1 = [2.95, 3.05] \quad \text{and} \quad \theta_2 = [0.95, 1.05]$$

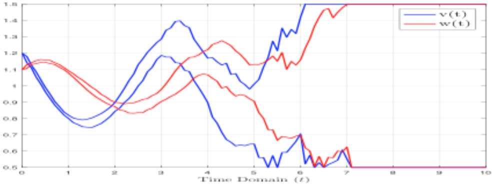
Let's see what we obtain when we run the simulations with FOM (size 200) and with ROM (size 3).

ADVANTAGE OF INTERVAL ROM OVER INTERVAL FOM

- We observed that simulations on ROM yield less uncertainty than simulations on FOM

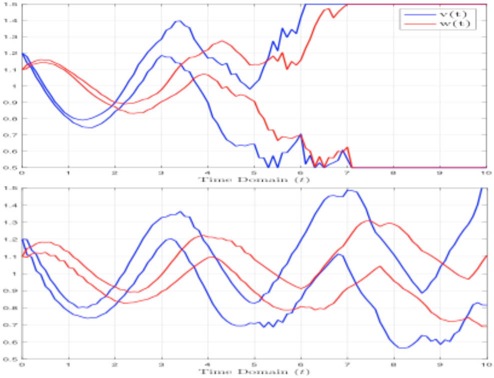
ADVANTAGE OF INTERVAL ROM OVER INTERVAL FOM

- We observed that simulations on ROM yield less uncertainty than simulations on FOM



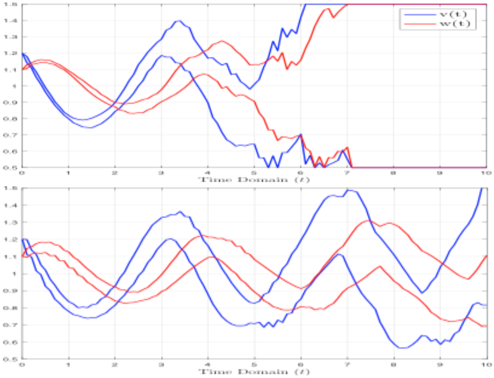
ADVANTAGE OF INTERVAL ROM OVER INTERVAL FOM

- We observed that simulations on ROM yield less uncertainty than simulations on FOM



ADVANTAGE OF INTERVAL ROM OVER INTERVAL FOM

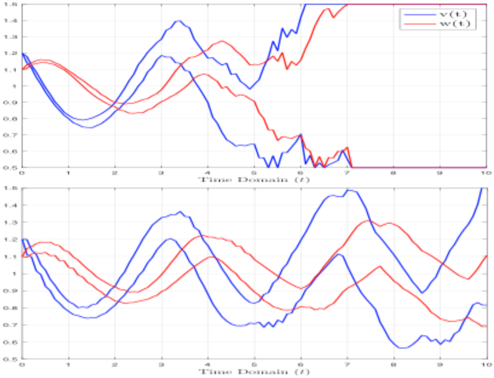
- We observed that simulations on ROM yield less uncertainty than simulations on FOM



The runtime is 74,596ms for FOM and 4,616ms for ROM.

ADVANTAGE OF INTERVAL ROM OVER INTERVAL FOM

- We observed that simulations on ROM yield less uncertainty than simulations on FOM



The runtime is 74,596ms for FOM and 4,616ms for ROM.

PREDICTIONS

- **Our original problem.** We want to solve: $F_\lambda(x) = 0$, where $x \in \mathbb{R}^n$ with n too large.

PREDICTIONS

- **Our original problem.** We want to solve: $F_\lambda(x) = 0$, where $x \in \mathbb{R}^n$ with n too large.

- **Our new problem.** We know what type of phenomenom is happening, we observe it, but we do not know λ.

PREDICTIONS

- **Important note.** We have access to F.

PREDICTIONS

- **Important note.** We have access to F.
  Instead of solving: $F_\lambda(x) = F(\lambda, x) = 0$, we now solve: $F_{Obs}(\lambda, x \setminus Obs) = 0$, where our observations are uncertain.

PREDICTIONS

- **Important note.** We have access to F.
  Instead of solving: $F_\lambda(x) = F(\lambda, x) = 0$, we now solve: $F_{Obs}(\lambda, x \setminus Obs) = 0$, where our observations are uncertain.

$$\begin{cases} F_{Obs}(\lambda, x \setminus Obs) = 0 \\ \forall x_k \in Obs,\ x_k = [\underline{Obs_k}, \overline{Obs_k}] \end{cases}$$

- **Our challenge.**

PREDICTIONS

- **Important note.** We have access to F.
  Instead of solving: $F_\lambda(x) = F(\lambda, x) = 0$, we now solve: $F_{Obs}(\lambda, x \setminus Obs) = 0$, where our observations are uncertain.

$$\begin{cases} F_{Obs}(\lambda, x \setminus Obs) = 0 \\ \forall x_k \in Obs,\ x_k = [\underline{Obs_k}, \overline{Obs_k}] \end{cases}$$

- **Our challenge.**
  - Instead of solving the above problem in the original space, we need to solve it in the reduced space.

PREDICTIONS

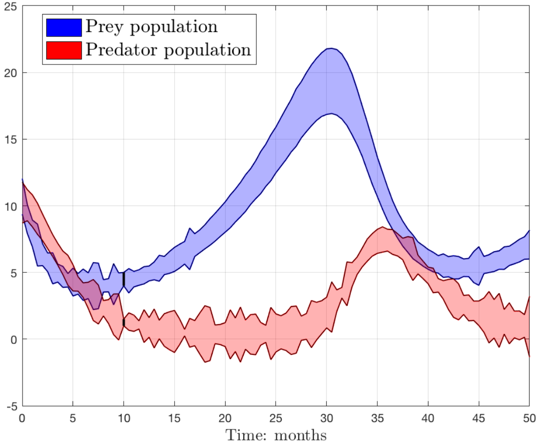
- **Important note.** We have access to F.
  Instead of solving: $F_\lambda(x) = F(\lambda, x) = 0$, we now solve: $F_{Obs}(\lambda, x \setminus Obs) = 0$, where our observations are uncertain.

$$\begin{cases} F_{Obs}(\lambda, x \setminus Obs) = 0 \\ \forall x_k \in Obs,\ x_k = [\underline{Obs_k}, \overline{Obs_k}] \end{cases}$$

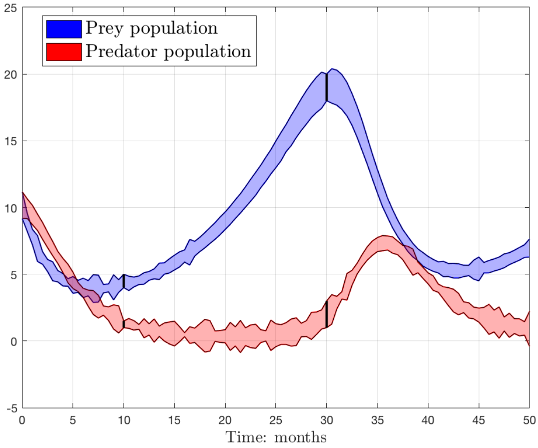
- **Our challenge.**
  - Instead of solving the above problem in the original space, we need to solve it in the reduced space. However, the observations do not correspond to any variable of the reduced space.

$$\begin{cases} F(\Phi\tilde{x}, \lambda) = 0 \\ \forall x_k \in Obs,\ x_k = \sum_{i=1}^{p} \Phi_{k,i}\tilde{x}_i \end{cases}$$

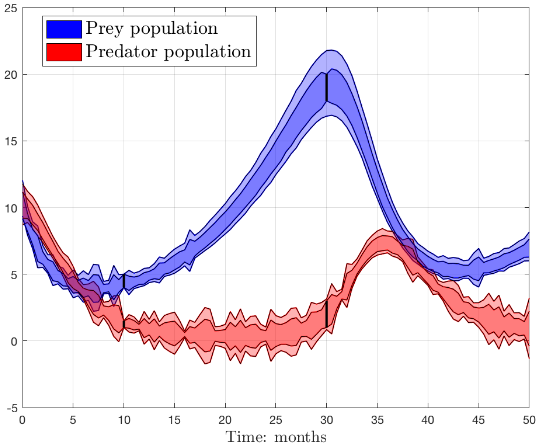
PREDICTIONS: SOME RESULTS



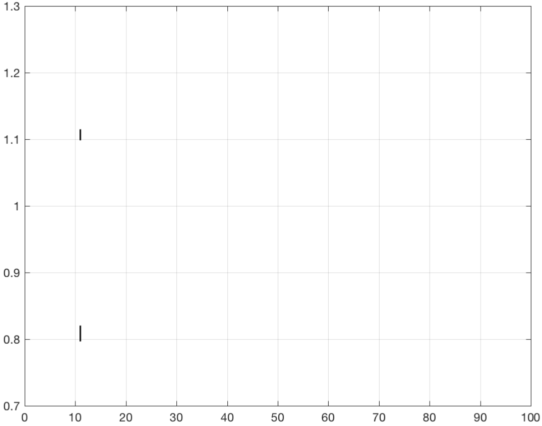
PREDICTIONS: SOME RESULTS



PREDICTIONS: SOME RESULTS

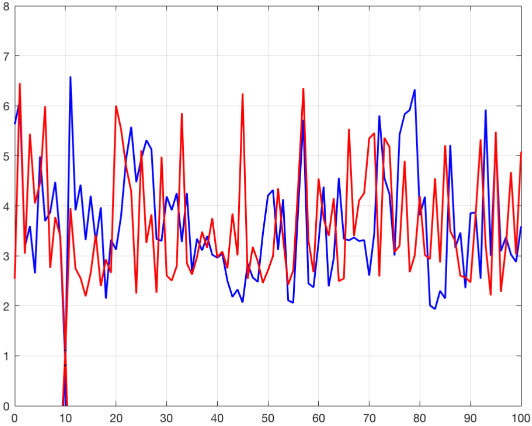


PREDICTIONS: SOME RESULTS

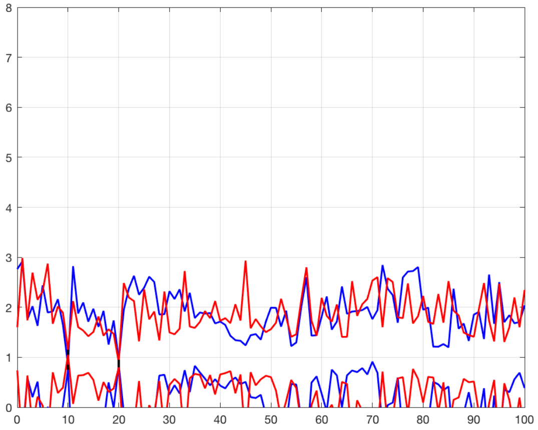


What we know: one observation set and $\theta_1 = \theta_2 = [0, 6]$.

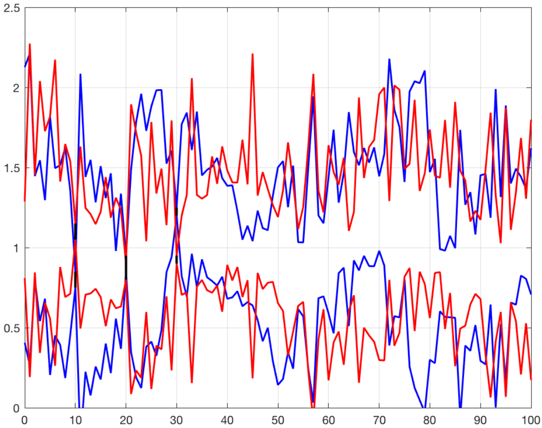
PREDICTIONS: SOME RESULTS



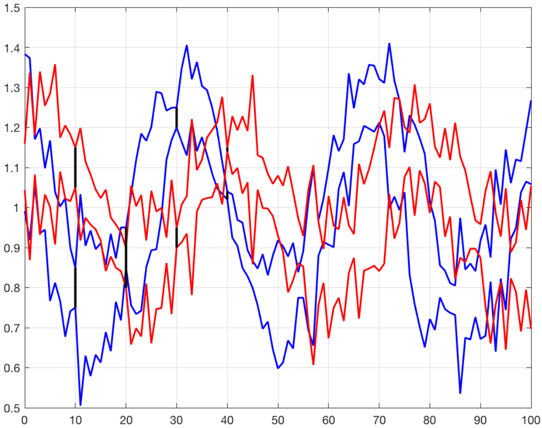
PREDICTIONS: SOME RESULTS



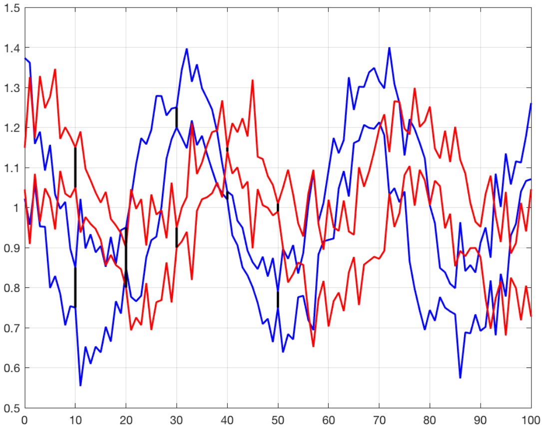
PREDICTIONS: SOME RESULTS



PREDICTIONS: SOME RESULTS

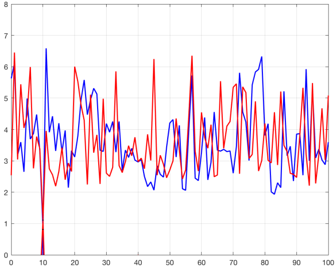


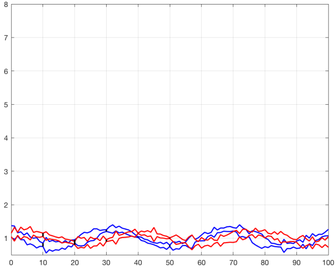
PREDICTIONS: SOME RESULTS



PREDICTIONS: SOME RESULTS

A look at the improvement from start to finish on the same scale:





Left: one observation set and $\theta_1 = \theta_2 = [0, 6]$.
Right: five observation sets, $\theta_1 = [0.1875, 6]$ and $\theta_2 = [0, 4.6875]$

PREDICTING DYNAMIC SYSTEMS' BEHAVIOR

Some conclusions:

- We are able to make predictions
- We observed that predictions on ROM yield less uncertainty than predictions on FOM
- But we still need:

PREDICTING DYNAMIC SYSTEMS' BEHAVIOR

Some conclusions:

- We are able to make predictions
- We observed that predictions on ROM yield less uncertainty than predictions on FOM
- But we still need:
  - to handle outliers: *at best no solution, at worst erroneous ones*
  - to handle time horizon uncertainty

A SIMILAR PROBLEM: CONTROL / RECOMPUTATION

- What if: we **can modify the values** of the dynamic system's parameters.

A SIMILAR PROBLEM: CONTROL / RECOMPUTATION

- What if: we **can modify the values** of the dynamic system's parameters.
- **Why/when** would we do that?

A SIMILAR PROBLEM: CONTROL / RECOMPUTATION

- What if: we **can modify the values** of the dynamic system's parameters.
- **Why/when** would we do that?
  - to fix an unfolding phenomenon after an unexpected event

A SIMILAR PROBLEM: CONTROL / RECOMPUTATION

- What if: we **can modify the values** of the dynamic system's parameters.
- **Why/when** would we do that?
  - to fix an unfolding phenomenon after an unexpected event
  - to ensure a given behavior

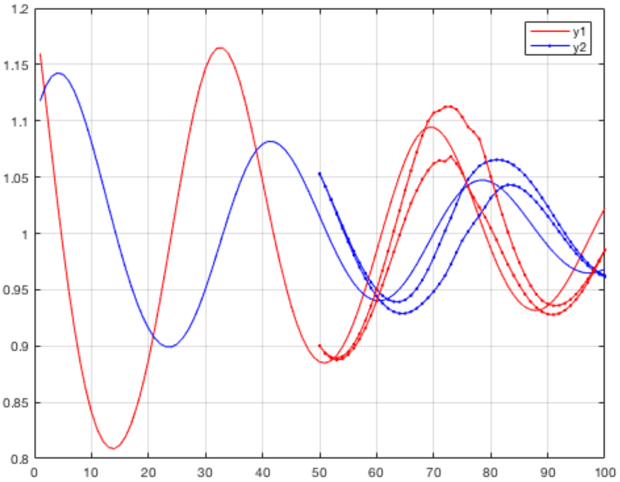
A SIMILAR PROBLEM: CONTROL / RECOMPUTATION

- What if: we **can modify the values** of the dynamic system's parameters.
- **Why/when** would we do that?
  - to fix an unfolding phenomenon after an unexpected event
  - to ensure a given behavior
  - to prevent a given behavior

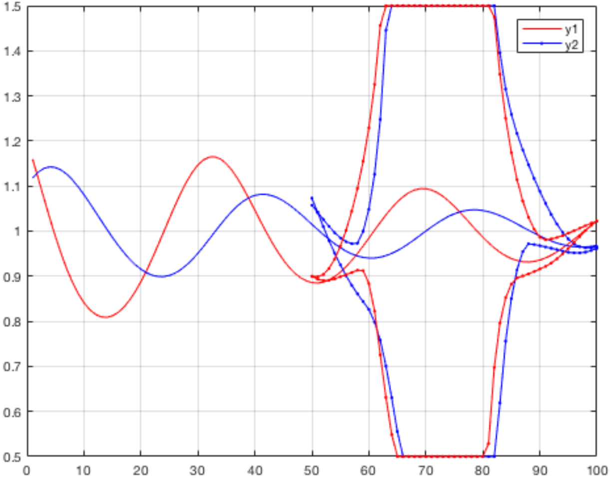
A SIMILAR PROBLEM: CONTROL / RECOMPUTATION

- What if: we **can modify the values** of the dynamic system's parameters.
- **Why/when** would we do that?
  - to fix an unfolding phenomenon after an unexpected event
  - to ensure a given behavior
  - to prevent a given behavior
- **What does it look like?**

RECOMPUTING DYNAMIC SYSTEMS' PARAMETERS



RECOMPUTING DYNAMIC SYSTEMS' PARAMETERS



RECOMPUTING DYNAMIC SYSTEMS' PARAMETERS

Still using the Lokta-Volterra problem:

$$\begin{cases} y_1' = \theta_1 y_1 (1 - y_2) = \theta_1 y_1 - \theta_1 y_1 y_2, \\ y_2' = \theta_2 y_2 (y_2 - 1) = \theta_2 y_2 y_1 - \theta_2 y_2, \end{cases}$$

We choose: $y_1(0) = 1.2$, $\theta_1 = 2.95$, $y_2(0) = 1.1$, and $\theta_2 = 1.0$.

RECOMPUTING DYNAMIC SYSTEMS' PARAMETERS

Still using the Lokta-Volterra problem:

$$\begin{cases} y_1' = \theta_1 y_1 (1 - y_2) = \theta_1 y_1 - \theta_1 y_1 y_2, \\ y_2' = \theta_2 y_2 (y_2 - 1) = \theta_2 y_2 y_1 - \theta_2 y_2, \end{cases}$$

We choose: $y_1(0) = 1.2$, $\theta_1 = 2.95$, $y_2(0) = 1.1$, and $\theta_2 = 1.0$.

But then we decide to alter the value of θ_1 to 1.5 at the 50^{th} time step, and we recompute θ_2 to enforce that y_1 and y_2 will converge to the same end point as in the initial problem at the 100^{th} time step.

RECOMPUTING DYNAMIC SYSTEMS' PARAMETERS

Still using the Lokta-Volterra problem:

$$\begin{cases} y_1' = \theta_1 y_1(1 - y_2) = \theta_1 y_1 - \theta_1 y_1 y_2, \\ y_2' = \theta_2 y_2(y_2 - 1) = \theta_2 y_2 y_1 - \theta_2 y_2, \end{cases}$$

We choose: $y_1(0) = 1.2$, $\theta_1 = 2.95$, $y_2(0) = 1.1$, and $\theta_2 = 1.0$.

But then we decide to alter the value of θ_1 to 1.5 at the 50^{th} time step, and we recompute θ_2 to enforce that y_1 and y_2 will converge to the same end point as in the initial problem at the 100^{th} time step.

We obtained no solution despite $\theta_2 = [0, 5]$.

RECOMPUTING DYNAMIC SYSTEMS' PARAMETERS

Some conclusions:

RECOMPUTING DYNAMIC SYSTEMS' PARAMETERS

Some conclusions:

- We are able to (re-)compute parameters

RECOMPUTING DYNAMIC SYSTEMS' PARAMETERS

Some conclusions:

- We are able to (re-)compute parameters
- But we still have to take recomputation time into account when doing it "on the fly"

RECOMPUTING DYNAMIC SYSTEMS' PARAMETERS

Some conclusions:

- We are able to (re-)compute parameters
- But we still have to take recomputation time into account when doing it “on the fly”
- **Future steps?** identify parameters that, even under uncertainty, guarantee a certain behavior. E.g., combustion problem with uncertain fuel mix

NEXT STEPS

- **Approaches: model time uncertainty**

NEXT STEPS

- **Approaches: model time uncertainty**
  - Time horizon
  - Observation times
  - Recomputation time
  - Control prediction time

NEXT STEPS

- **Approaches: model time uncertainty**
    - Time horizon
    - Observation times
    - Recomputation time
    - Control prediction time
- **Applications:**

NEXT STEPS

- **Approaches: model time uncertainty**
  - Time horizon
  - Observation times
  - Recomputation time
  - Control prediction time
- **Applications:**
  - Fuel mix uncertainty
  - Combustion nozzle geometry
  - Problems with discontinuities

THANK YOU FOR YOUR ATTENTION

Any questions?

Martine Ceberio, mceberio@utep.edu
Leobardo Valera, lvalera@gmail.com
Angel Garcia Contreras, afgarciacontreras@miners.utep.edu