Decision Making for Dynamic Systems under Uncertainty: Predictions, Recomputations, and a Mobile Tool

AHPCRC Technical Review Meeting – ARL APG – November 28, 2017

Martine Ceberio, Miguel Argaez, Horacio Florez, Leobardo Valera, Jesus Padilla, Phillip Hassoun

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

WHAT IS THE PROBLEM THAT WE ADDRESS?

Given the model of a dynamic system, use it to make **decisions**.

WHAT IS THE PROBLEM THAT WE ADDRESS?

Given the model of a dynamic system, use it to make **decisions**.

- What type of decisions?

WHAT IS THE PROBLEM THAT WE ADDRESS?

Given the model of a dynamic system, use it to make **decisions**.

- What type of decisions?
- What are the challenges? Why is it hard?

WHAT IS THE PROBLEM THAT WE ADDRESS?

Given the model of a dynamic system, use it to make **decisions**.

- What type of decisions?
- What are the challenges? Why is it hard?

Types of decisions of interest:

WHAT IS THE PROBLEM THAT WE ADDRESS?

Given the model of a dynamic system, use it to make **decisions**.

- What type of decisions?
- What are the challenges? Why is it hard?

Types of decisions of interest:

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

WHAT IS THE PROBLEM THAT WE ADDRESS?

Given the model of a dynamic system, use it to make **decisions**.

- What type of decisions?
- What are the challenges? Why is it hard?

Types of decisions of interest:

- **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 for instance

WHAT IS THE PROBLEM THAT WE ADDRESS?

Given the model of a dynamic system, use it to make **decisions**.

- What type of decisions?
- What are the challenges? Why is it hard?

Types of decisions of interest:

- **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 for instance
- **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

- Solving a dynamical system potentially leads to a **large system of equations** – possibly nonlinear

CHALLENGES

- Solving a dynamical system potentially leads to a **large system of equations** – possibly nonlinear
  - We can solve these large problems

CHALLENGES

- Solving a dynamical system potentially leads to a **large system of equations** – possibly nonlinear
  - We can solve these large problems
  - But it takes time
  - What can be done?

CHALLENGES

- Solving a dynamical system potentially leads to a **large system of equations** – possibly nonlinear
  - We can solve these large problems
  - But it takes time
  - What can be done?
- Let's add to that the possibility of **uncertainty** in the model, data, etc.

CHALLENGES

- Solving a dynamical system potentially leads to a **large system of equations** – possibly nonlinear
  - We can solve these large problems
  - But it takes time
  - What can be done?
- Let's add to that the possibility of **uncertainty** in the model, data, etc.
- And some interest in **reliability** / guaranteed results... as can be
  - It is not just about getting data to make decisions.

CHALLENGES

- Solving a dynamical system potentially leads to a **large system of equations** – possibly nonlinear
  - We can solve these large problems
  - But it takes time
  - What can be done?
- Let’s add to that the possibility of **uncertainty** in the model, data, etc.
- And some interest in **reliability** / guaranteed results... as can be
  - It is not just about getting data to make decisions.
  - We’d like to be able to rely on such data.

HOW WE ADDRESSED THESE CHALLENGES

- **Size.**
  - We worked on **reducing the size** of the discretized model

HOW WE ADDRESSED THESE CHALLENGES

- **Size.**
  - We worked on **reducing the size** of the discretized model
  - via: Model-Order Reduction (ROM)

HOW WE ADDRESSED THESE CHALLENGES

- **Size.**
  - We worked on **reducing the size** of the discretized model
  - via: Model-Order Reduction (ROM): *with wavelets, intervals*

HOW WE ADDRESSED THESE CHALLENGES

- **Size.**
  - We worked on **reducing the size** of the discretized model
  - via: Model-Order Reduction (ROM): *with wavelets, intervals*
- **Uncertainty.**
  - On Full-Order Models (FOMs): to generate ROMs in presence of uncertainty in the model or just to limit the number of FOM executions for snapshots

HOW WE ADDRESSED THESE CHALLENGES

- **Size.**
  - We worked on **reducing the size** of the discretized model
  - via: Model-Order Reduction (ROM): *with wavelets, intervals*
- **Uncertainty.**
  - On Full-Order Models (FOMs): to generate ROMs in presence of uncertainty in the model or just to limit the number of FOM executions for snapshots
  - On ROMs: to address uncertainty in input parameters, model constants

HOW WE ADDRESSED THESE CHALLENGES

- **Size.**
  - We worked on **reducing the size** of the discretized model
  - via: Model-Order Reduction (ROM): *with wavelets, intervals*
- **Uncertainty.**
  - On Full-Order Models (FOMs): to generate ROMs in presence of uncertainty in the model or just to limit the number of FOM executions for snapshots
  - On ROMs: to address uncertainty in input parameters, model constants
  - We used Interval Constraint Solving Techniques when ROMs were small enough

HOW WE ADDRESSED THESE CHALLENGES

- **Size.**
  - We worked on **reducing the size** of the discretized model
  - via: Model-Order Reduction (ROM): *with wavelets, intervals*
- **Uncertainty.**
  - On Full-Order Models (FOMs): to generate ROMs in presence of uncertainty in the model or just to limit the number of FOM executions for snapshots
  - On ROMs: to address uncertainty in input parameters, model constants
  - We used Interval Constraint Solving Techniques when ROMs were small enough
  - We used Stochastic Approaches when ROMs were still large for ICST

HOW WE ADDRESSED THESE CHALLENGES

- **Size.**
  - We worked on **reducing the size** of the discretized model
  - via: Model-Order Reduction (ROM): *with wavelets, intervals*
- **Uncertainty.**
  - On Full-Order Models (FOMs): to generate ROMs in presence of uncertainty in the model or just to limit the number of FOM executions for snapshots
  - On ROMs: to address uncertainty in input parameters, model constants
  - We used Interval Constraint Solving Techniques when ROMs were small enough
  - We used Stochastic Approaches when ROMs were still large for ICST
- **Reliability.**
  - Interval computations allow to carry guaranteed computations

FOCUS OF THIS PRESENTATION

- How to make predictions on observed (not controled) dynamic phenomena?

FOCUS OF THIS PRESENTATION

- How to make predictions on observed (not controled) dynamic phenomena?
- How to conduct parameters' recomputations for unfolding dynamic phenomena?

FOCUS OF THIS PRESENTATION

- How to make predictions on observed (not controled) dynamic phenomena?
- How to conduct parameters' recomputations for unfolding dynamic phenomena?
- Is this usable in practice? Or do we need to harness a lot of computational power to make such decisions?

GENERAL ASSUMPTIONS

In order to predict behavior or recompute parameters:

- We need to know which dynamic system we are observing / modifying

GENERAL ASSUMPTIONS

In order to predict behavior or recompute parameters:

- We need to know which dynamic system we are observing / modifying
- We assume that we have access to a ROM of such problem

BASE PROBLEMS

- Original **FOM** problem. We have:

$$F : \mathbb{R}^n \times \mathbb{R}^m \longrightarrow \mathbb{R}^n, \quad F : \; (x, \lambda) \mapsto F(x, \lambda)$$

Knowing the value of λ, we often solve $F(x, \lambda) = F_\lambda(x) = 0$.

BASE PROBLEMS

- Original **FOM** problem. We have:

$$F:\mathbb{R}^n\times\mathbb{R}^m\longrightarrow\mathbb{R}^n,\quad F:\ (x,\lambda)\mapsto F(x,\lambda)$$

  Knowing the value of λ, we often solve $F(x,\lambda)=F_\lambda(x)=0$.

- **Reduced problem** (ROM). We now have:

$$F_\Phi:\mathbb{R}^p\times\mathbb{R}^m\longrightarrow\mathbb{R}^n,\quad F_\Phi:\ (\tilde{x},\lambda)\mapsto F(\Phi\cdot\tilde{x},\lambda),\quad \text{with: } p\ll n$$

  Knowing the value of λ, we could solve $F(\Phi\cdot\tilde{x},\lambda)=F_\lambda(\Phi\cdot\tilde{x})=0$, where $\tilde{x}\in\mathbb{R}^p$, and then we would recover x using $x=\Phi\cdot\tilde{x}$.

PREDICTING DYNAMIC SYSTEMS' BEHAVIOR

For **Predictions**, what problem are we solving?

PREDICTING DYNAMIC SYSTEMS' BEHAVIOR

For **Predictions**, what problem are we solving?

- We do not know the values of λ (and possibly of boundary conditions either).
- We have access to observed values of the original system's variables (FOM's variables)

$$\mathrm{Obs} = \{x_i, i \in \{1, \dots, n\}\}$$

PREDICTING DYNAMIC SYSTEMS' BEHAVIOR

For **Predictions**, what problem are we solving?

- We do not know the values of λ (and possibly of boundary conditions either).
- We have access to observed values of the original system's variables (FOM's variables)

$$Obs = \{x_i, i \in \{1, \dots, n\}\}$$

- So we have the following problem:

$$F(\lambda, x) = 0 \text{ is now: } F_{Obs}(\lambda, x \setminus Obs) = 0$$

Not such a different-looking problem from what we had before: somehow $G(X) = 0$ for some G.

PREDICTING DYNAMIC SYSTEMS' BEHAVIOR

For **Predictions**, what problem are we solving?

- We do not know the values of λ (and possibly of boundary conditions either).
- We have access to observed values of the original system's variables (FOM's variables)

$$Obs = \{x_i, i \in \{1, \ldots, n\}\}$$

- So we have the following problem:

$F(\lambda, x) = 0$ is now: $F_{Obs}(\lambda, x \setminus Obs) = 0$

Not such a different-looking problem from what we had before: somehow $G(X) = 0$ for some G.
However, we solve the Reduced version of F: $F(\Phi \cdot \tilde{x}, \lambda) = 0$ with $\tilde{x} \in \mathbb{R}^p$, $p \ll n$.
So we end up trying to solve the following **reduced system of (possibly nonlinear) equations**:

$$\begin{cases} F(\Phi\tilde{x}, \lambda) = 0 \\ \end{cases}$$

PREDICTING DYNAMIC SYSTEMS' BEHAVIOR

For **Predictions**, what problem are we solving?

- We do not know the values of λ (and possibly of boundary conditions either).
- We have access to observed values of the original system's variables (FOM's variables)

$$\mathrm{Obs} = \{x_i, i \in \{1, \ldots, n\}\}$$

- So we have the following problem:

$$F(\lambda, x) = 0 \text{ is now: } F_{\mathrm{Obs}}(\lambda, x \setminus \mathrm{Obs}) = 0$$

Not such a different-looking problem from what we had before: somehow $G(X) = 0$ for some G.
However, we solve the Reduced version of F: $F(\Phi \cdot \tilde{x}, \lambda) = 0$ with $\tilde{x} \in \mathbb{R}^p$, $p \ll n$.
So we end up trying to solve the following **reduced system of (possibly nonlinear) equations**:

$$\begin{cases} F(\Phi\tilde{x}, \lambda) = 0 \quad \textit{(same as before)} \\ \end{cases}$$

PREDICTING DYNAMIC SYSTEMS' BEHAVIOR

For **Predictions**, what problem are we solving?

- We do not know the values of λ (and possibly of boundary conditions either).
- We have access to observed values of the original system's variables (FOM's variables)

$$\mathrm{Obs} = \{x_i, i \in \{1, \dots, n\}\}$$

- So we have the following problem:

$$F(\lambda, x) = 0 \text{ is now: } F_{\mathrm{Obs}}(\lambda, x \setminus \mathrm{Obs}) = 0$$

Not such a different-looking problem from what we had before: somehow $G(X) = 0$ for some G.
However, we solve the Reduced version of F: $F(\Phi \cdot \tilde{x}, \lambda) = 0$ with $\tilde{x} \in \mathbb{R}^p$, $p \ll n$.
So we end up trying to solve the following **reduced system of (possibly nonlinear) equations**:

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

PREDICTING DYNAMIC SYSTEMS' BEHAVIOR

So here is the problem we are solving:

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

PREDICTING DYNAMIC SYSTEMS' BEHAVIOR

So here is the problem we are solving:

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

The catch is that observations usually include uncertainty. As a result, we have:

$$x_k = [\underline{x_k}, \overline{x_k}]$$

PREDICTING DYNAMIC SYSTEMS' BEHAVIOR

So here is the problem we are solving:

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

The catch is that observations usually include uncertainty. As a result, we have:

$$x_k = [\underline{x_k}, \overline{x_k}]$$

We solve such problems (with uncertainty and nonlinear) using interval computations and constraint solving techniques.

PREDICTING DYNAMIC SYSTEMS' BEHAVIOR

So here is the problem we are solving:

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

The catch is that observations usually include uncertainty. As a result, we have:

$$x_k = [\underline{x_k}, \overline{x_k}]$$

We solve such problems (with uncertainty and nonlinear) using interval computations and constraint solving techniques.

- *Interval computations*: because (1) they allow to explore the whole search space,

PREDICTING DYNAMIC SYSTEMS' BEHAVIOR

So here is the problem we are solving:

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

The catch is that observations usually include uncertainty. As a result, we have:

$$x_k = [\underline{x_k}, \overline{x_k}]$$

We solve such problems (with uncertainty and nonlinear) using interval computations and constraint solving techniques.

- *Interval computations*: because (1) they allow to explore the whole search space, (2) they provide guarantees on the computed solutions

PREDICTING DYNAMIC SYSTEMS' BEHAVIOR

So here is the problem we are solving:

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

The catch is that observations usually include uncertainty. As a result, we have:

$$x_k = [\underline{x_k}, \overline{x_k}]$$

We solve such problems (with uncertainty and nonlinear) using interval computations and constraint solving techniques.

- *Interval computations*: because (1) they allow to explore the whole search space, (2) they provide guarantees on the computed solutions
- *Constraint solving techniques*: because they always converge: so if they conclude that there is no solution, it is not to be blamed on a convergence issue

PREDICTING DYNAMIC SYSTEMS' BEHAVIOR

So here is the problem we are solving:

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

The catch is that observations usually include uncertainty. As a result, we have:

$$x_k = [\underline{x_k}, \overline{x_k}]$$

We solve such problems (with uncertainty and nonlinear) using interval computations and constraint solving techniques.

- *Interval computations*: because (1) they allow to explore the whole search space, (2) they provide guarantees on the computed solutions
- *Constraint solving techniques*: because they always converge: so if they conclude that there is no solution, it is not to be blamed on a convergence issue

In what follows, we illustrate our work on the Lotka-Volterra problem.

PREDICTIONS: LOTKA-VOLTERRA

Let's consider the following problem of predators and preys, defined by:

$$\frac{dv}{dt} = \theta_1 v(1-w) \quad \text{and} \quad \frac{dw}{dt} = \theta_2 w(v-1)$$

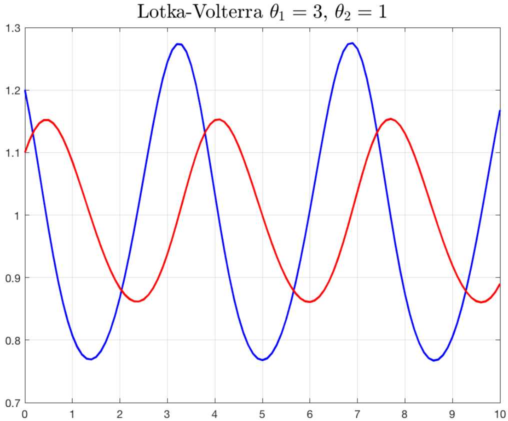
where v represents the number of preys and w the number of predators.

PREDICTIONS: LOTKA-VOLTERRA

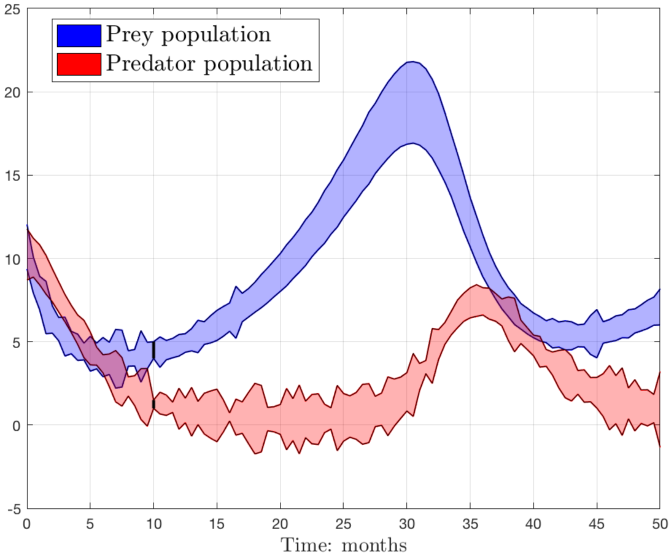
Let's consider the following problem of predators and preys, defined by:

$$\frac{dv}{dt} = \theta_1 v(1 - w) \quad \text{and} \quad \frac{dw}{dt} = \theta_2 w(v - 1)$$

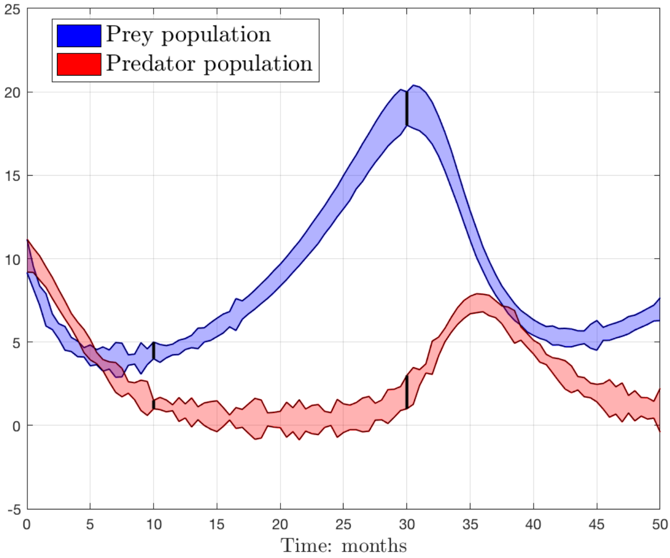
where v represents the number of preys and w the number of predators.



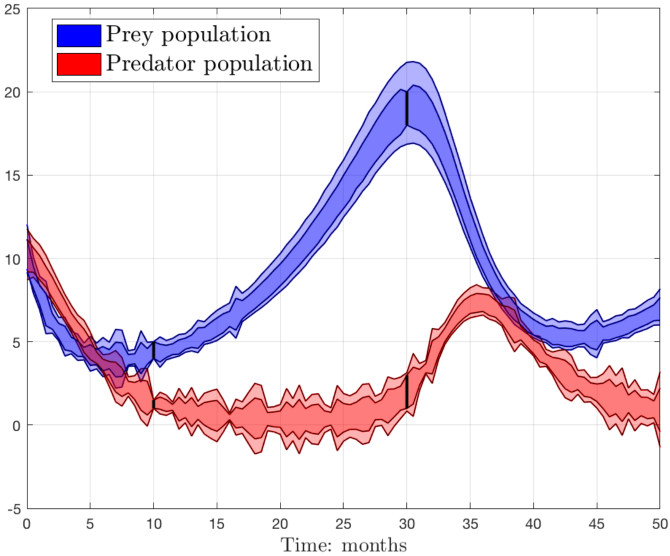
PREDICTIONS: LOTKA-VOLTERRA



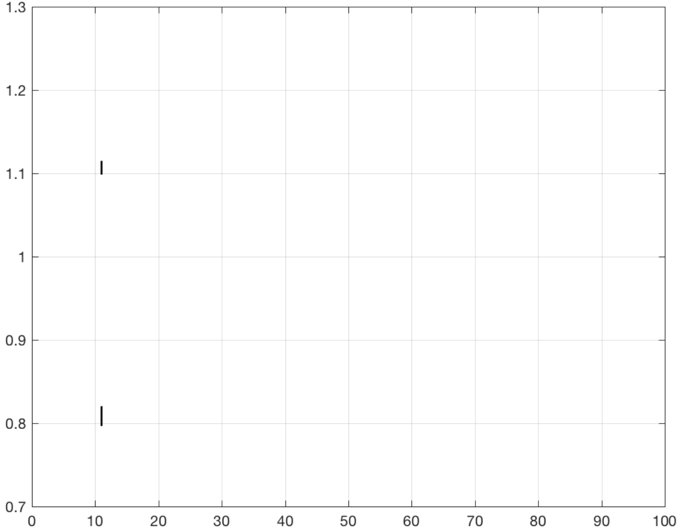
PREDICTIONS: LOTKA-VOLTERRA



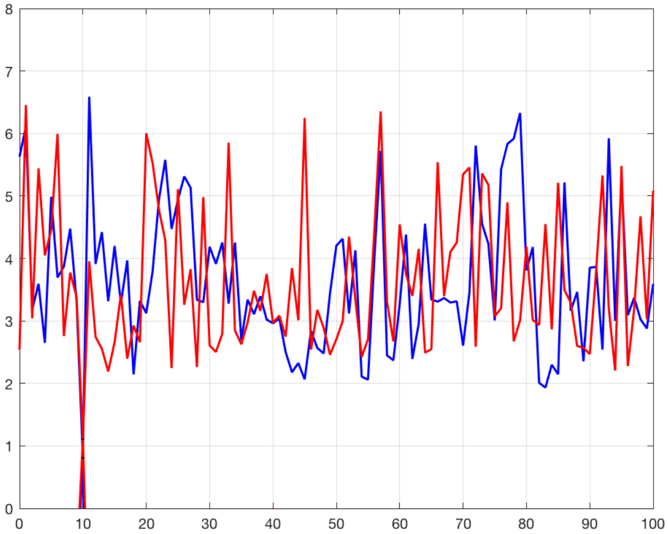
PREDICTIONS: LOTKA-VOLTERRA



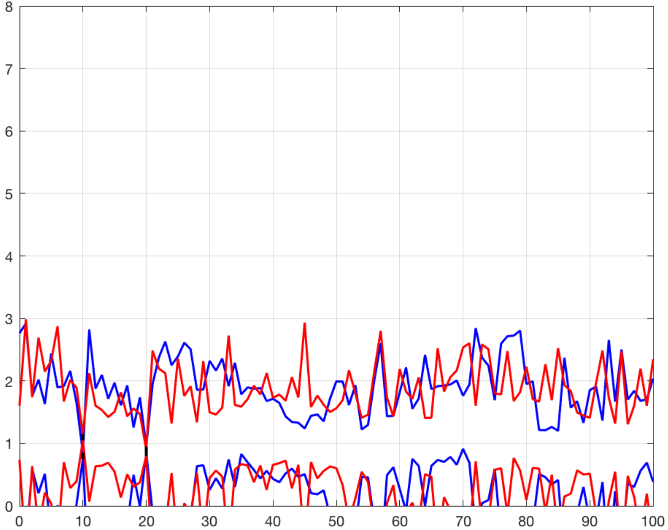
PREDICTIONS: LOTKA-VOLTERRA



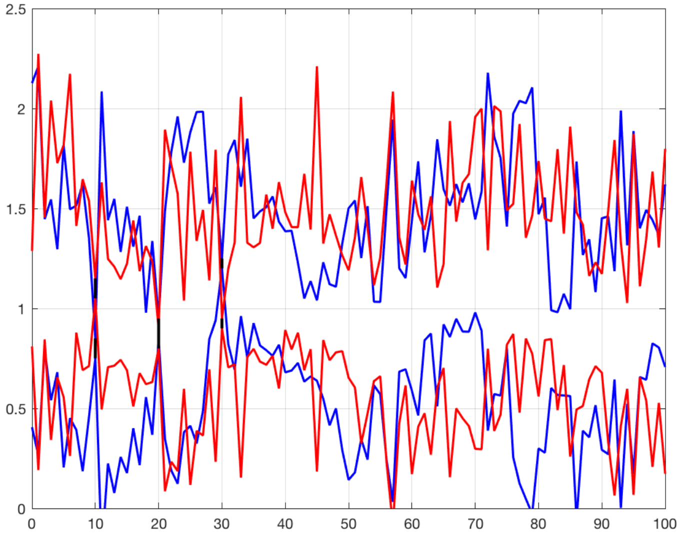
PREDICTIONS: LOTKA-VOLTERRA



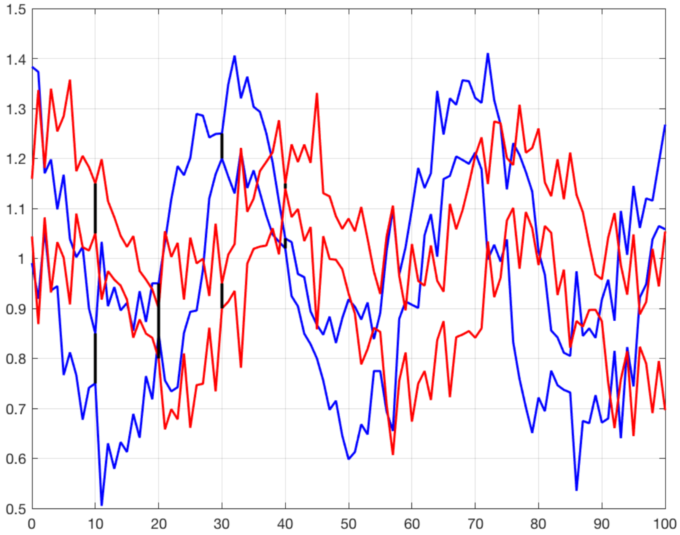
PREDICTIONS: LOTKA-VOLTERRA



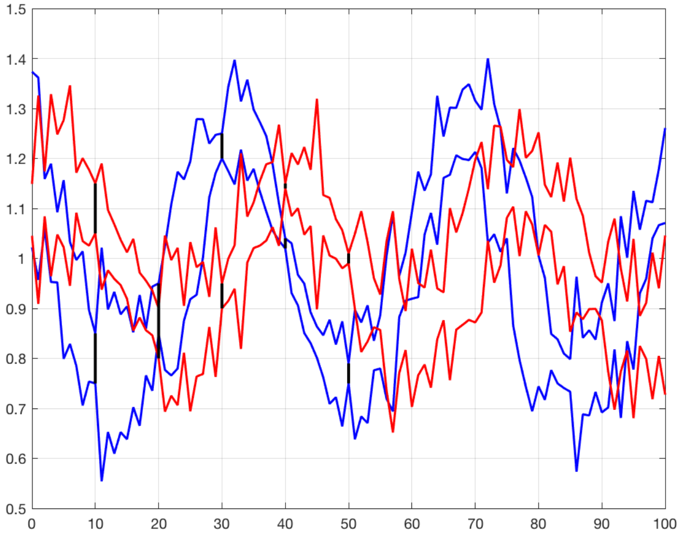
PREDICTIONS: LOTKA-VOLTERRA



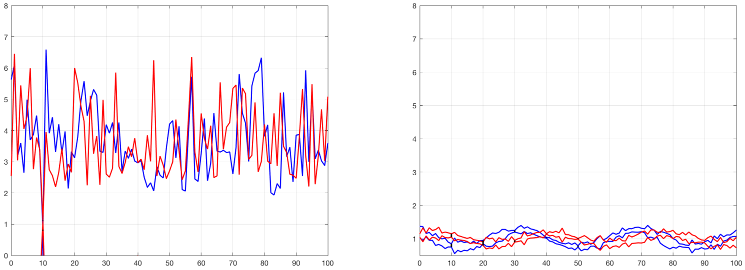
PREDICTIONS: LOTKA-VOLTERRA



PREDICTIONS: LOTKA-VOLTERRA



PREDICTIONS: LOTKA-VOLTERRA



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:

PREDICTING DYNAMIC SYSTEMS' BEHAVIOR

Some conclusions:

- We are able to make predictions

PREDICTING DYNAMIC SYSTEMS' BEHAVIOR

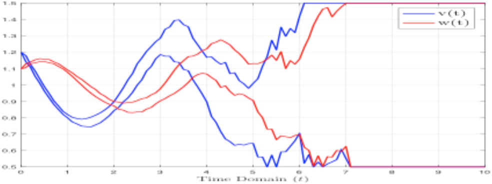
Some conclusions:

- We are able to make predictions
- We observed that predictions on ROM yield less uncertainty than predictions on FOM

PREDICTING DYNAMIC SYSTEMS' BEHAVIOR

Some conclusions:

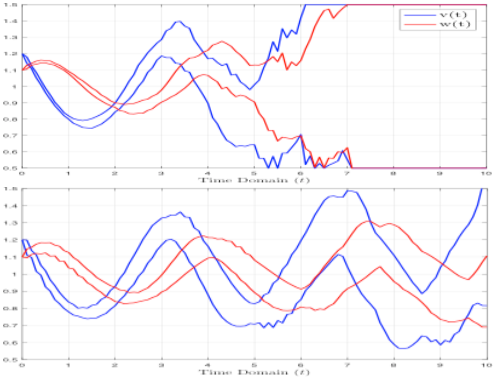
- We are able to make predictions
- We observed that predictions on ROM yield less uncertainty than predictions on FOM



PREDICTING DYNAMIC SYSTEMS' BEHAVIOR

Some conclusions:

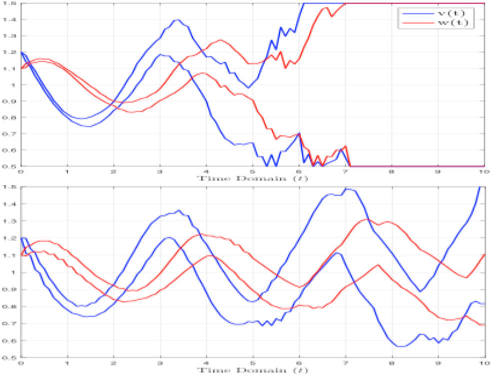
- We are able to make predictions
- We observed that predictions on ROM yield less uncertainty than predictions on FOM



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

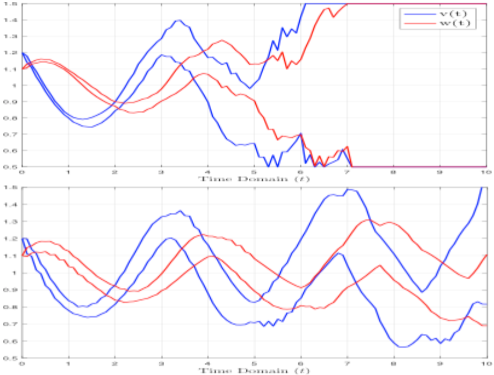


The runtime is 74,596ms for FOM and 4,616ms for ROM.

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



The runtime is 74,596ms for FOM and 4,616ms for ROM.
This poses the question of the quality of the ROM: *should we consider local bases?*

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

RECOMPUTING DYNAMIC SYSTEMS' PARAMETERS

- Another assumption: we **can modify the values** of the dynamic system's parameters.

RECOMPUTING DYNAMIC SYSTEMS' PARAMETERS

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

RECOMPUTING DYNAMIC SYSTEMS' PARAMETERS

- Another assumption: 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

RECOMPUTING DYNAMIC SYSTEMS' PARAMETERS

- Another assumption: 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

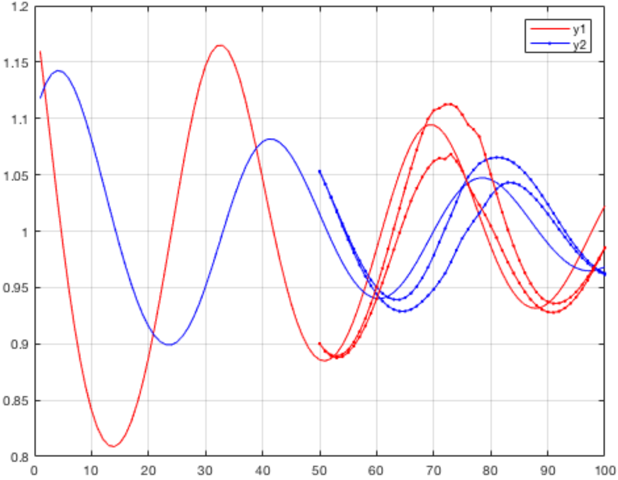
RECOMPUTING DYNAMIC SYSTEMS' PARAMETERS

- Another assumption: 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

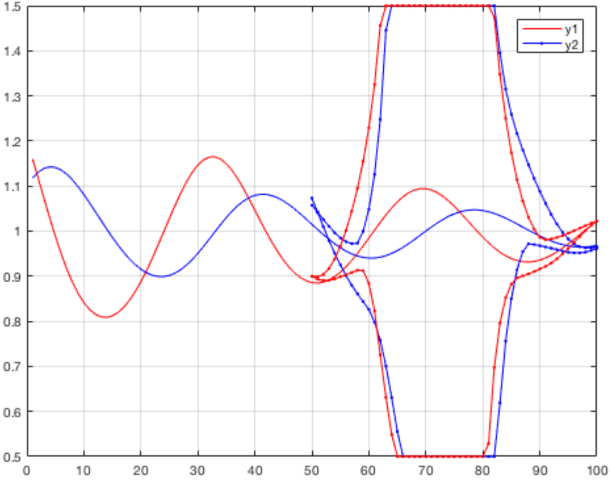
RECOMPUTING DYNAMIC SYSTEMS' PARAMETERS

- Another assumption: 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 in collaboration with Luis Bravo, ARL APG: what type of fuel mix? what geometry of the nozzle, etc.

DECISIONS UNDER UNCERTAINTY ON A MOBILE APP

- **Objective:** Show that we can handle uncertainty in ways we showed before on a computationally-limited device.

DECISIONS UNDER UNCERTAINTY ON A MOBILE APP

- **Objective:** Show that we can handle uncertainty in ways we showed before on a computationally-limited device. That was our **capstone** project.

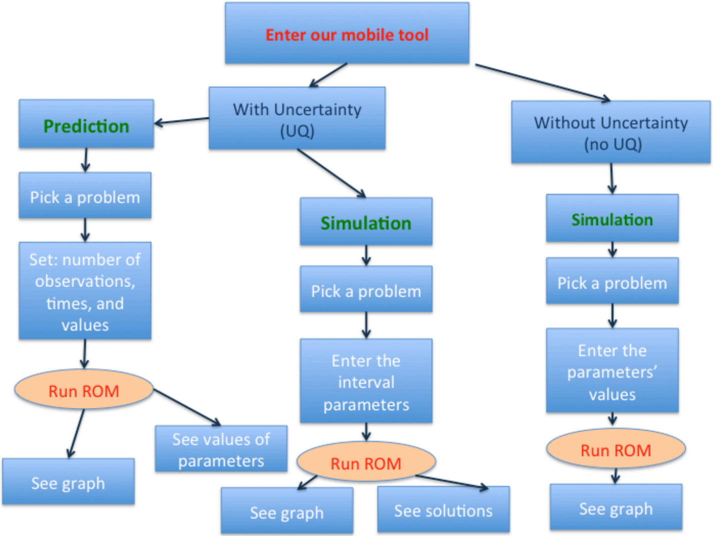
DECISIONS UNDER UNCERTAINTY ON A MOBILE APP

- **Objective:** Show that we can handle uncertainty in ways we showed before on a computationally-limited device. That was our **capstone** project.
- We implemented all techniques used before on an android device.

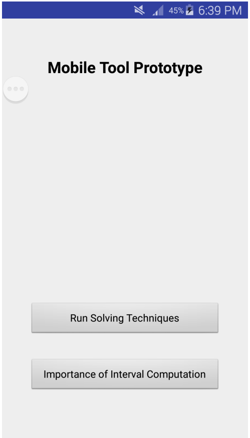
DECISIONS UNDER UNCERTAINTY ON A MOBILE APP

- **Objective:** Show that we can handle uncertainty in ways we showed before on a computationally-limited device. That was our **capstone** project.
- We implemented all techniques used before on an android device.
- Let's take a look at the intended use and functionalities of this app.
- *Note: this work was conducted with feedback from Simon Su, ARL APG*

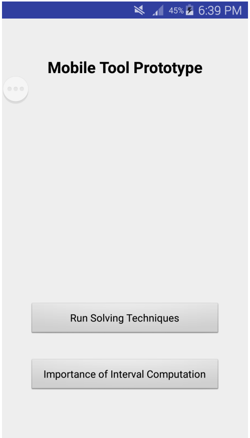
OUR MOBILE TOOL

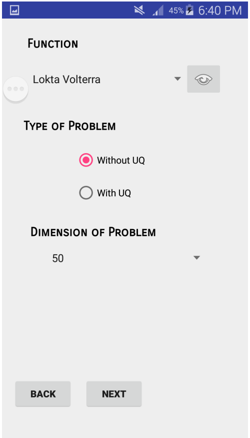


OUR MOBILE TOOL

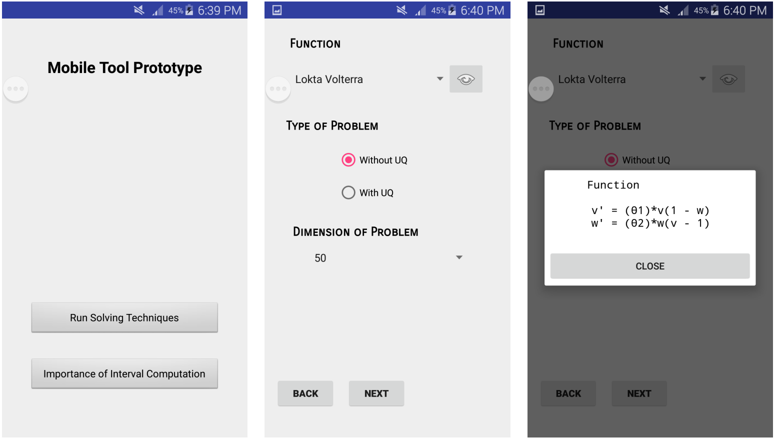


OUR MOBILE TOOL

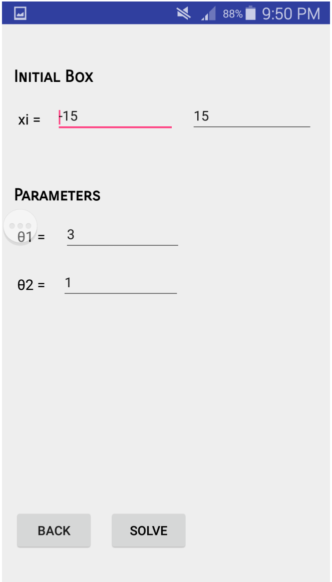




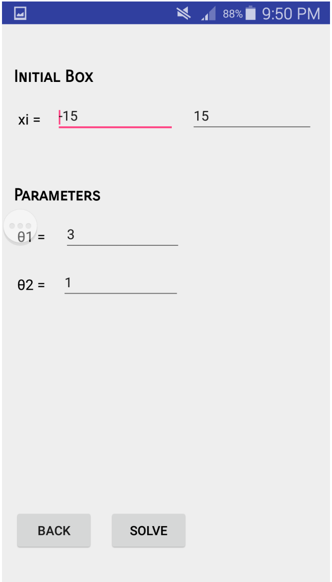
OUR MOBILE TOOL

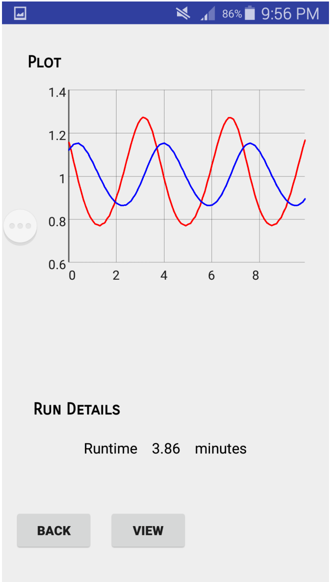


OUR MOBILE TOOL: SIMULATION W/O UQ

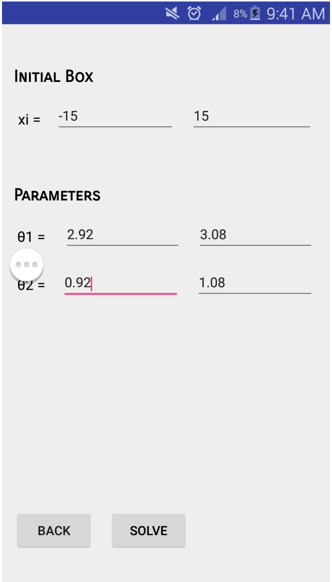


OUR MOBILE TOOL: SIMULATION W/O UQ

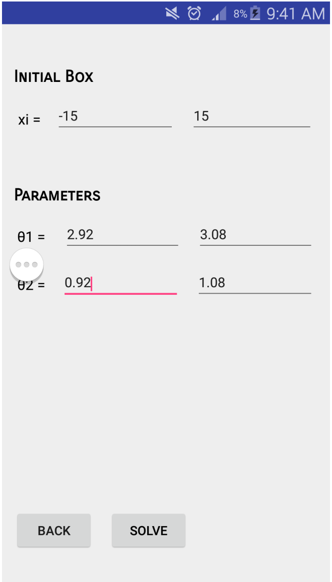


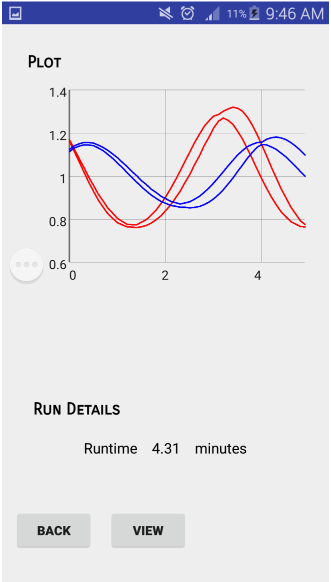


OUR MOBILE TOOL: SIMULATION WITH UQ

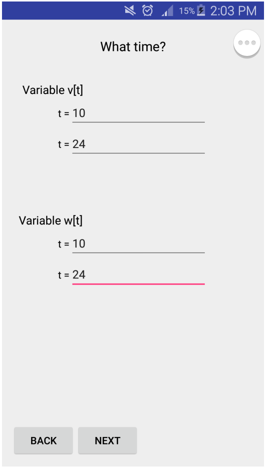


OUR MOBILE TOOL: SIMULATION WITH UQ

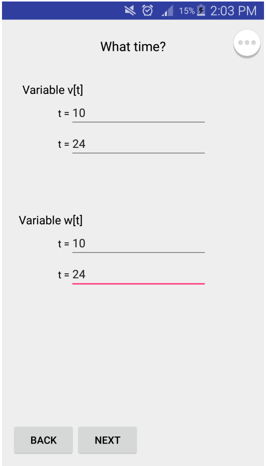




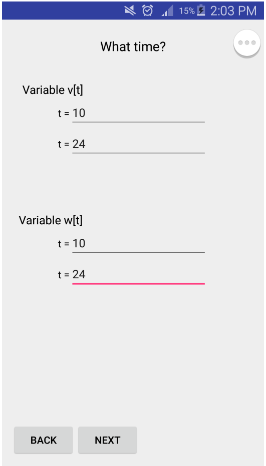
OUR MOBILE TOOL: PREDICTION (WITH UQ)

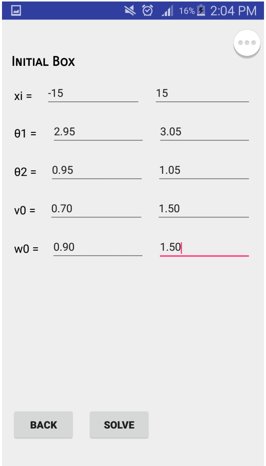


OUR MOBILE TOOL: PREDICTION (WITH UQ)

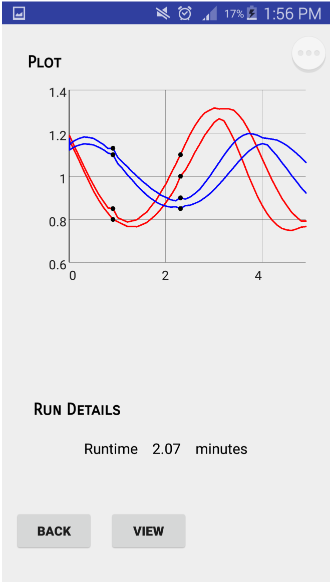


OUR MOBILE TOOL: PREDICTION (WITH UQ)





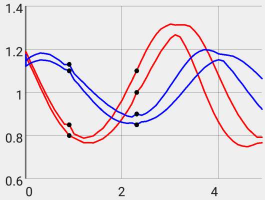
OUR MOBILE TOOL: PREDICTION (WITH UQ)



OUR MOBILE TOOL: PREDICTION (WITH UQ)

17% 1:56 PM

Plot



Run Details

Runtime 2.07 minutes

BACK VIEW

17% 1:56 PM

BACK

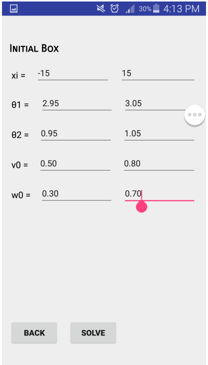
Hull All Solutions

Output Detail

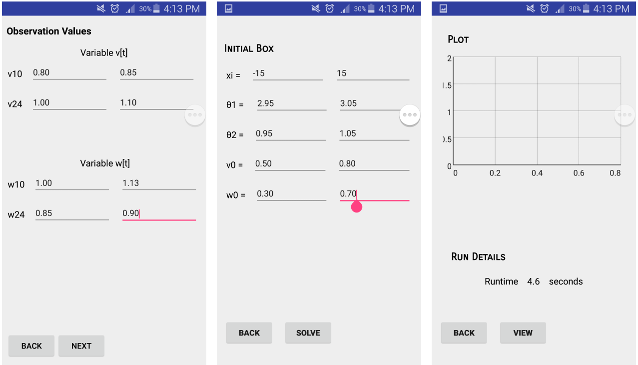
```
Solutions: 49
Function:
 v' = (θ1)*v(1 - w)
 w' = (θ2)*w(v - 1)
Initial Box
 xi = [-15.0,15.0]
 θ1 = [2.95,3.05]
 θ2 = [0.95,1.05] v0 = [0.7,1.5]
 w0 = [0.9,1.5]
Parameters

Box1
 v[1]=[1.16519418288,1.18857764561]
 v[2]=[1.11958081648,1.13946168551]
 v[3]=[1.07261493326,1.09261510602]
 v[4]=[1.02296678247,1.04530425043]
 v[5]=[0.97701077844,0.99959411163]
 v[6]=[0.93003025554,0.95792814917]
 v[7]=[0.89128378776,0.91740113855]
 v[8]=[0.85605902646,0.88430014653]
 v[9]=[0.82782711787,0.8522046992]
 v[10]=[0.8,0.85]
 v[11]=[0.78850809499,0.80756512129]
 v[12]=[0.77758528196,0.79677692846]
 v[13]=[0.7665912551,0.78737453682]
 v[14]=[0.76607292834,0.79233461908]
```

OUR MOBILE TOOL: PREDICTION (WITH UQ)

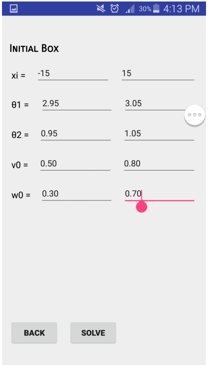
OUR MOBILE TOOL: PREDICTION (WITH UQ)

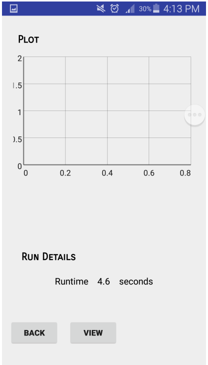


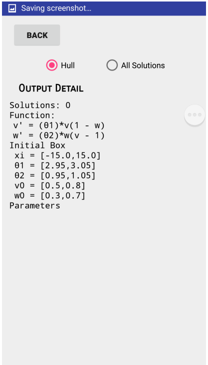
OUR MOBILE TOOL: PREDICTION (WITH UQ)



OUR MOBILE TOOL: PREDICTION (WITH UQ)







OUR MOBILE TOOL

Some conclusions:

OUR MOBILE TOOL

Some conclusions:

- We have a working app: no need for internet or powerful devices/machines

OUR MOBILE TOOL

Some conclusions:

- We have a working app: no need for internet or powerful devices/machines
- But we still have:
  - to work on prioritizing the size for predictions to enhance computation time
  - to include computations with local bases: we plan to leverage multiple cores
  - to include recomputations: similar computations to predictions, but will need to handle time uncertainty

OUR MOBILE TOOL

Some conclusions:

- We have a working app: no need for internet or powerful devices/machines
- But we still have:
  - to work on prioritizing the size for predictions to enhance computation time
  - to include computations with local bases: we plan to leverage multiple cores
  - to include recomputations: similar computations to predictions, but will need to handle time uncertainty
- We can also easily:
  - expand it to handle larger problems: with a connection to a webserver and a powerful machine
  - to include recomputations: similar computations to predictions, but will need to handle time uncertainty

CONCLUSION

What we learned and demonstrated:

- Handling uncertainty: an opportunity to do more, to draw more types of decisions
- Intervals: allow to provide guarantees on results

CONCLUSION

What we learned and demonstrated:

- Handling uncertainty: an opportunity to do more, to draw more types of decisions
- Intervals: allow to provide guarantees on results

In addition, we also worked on handling uncertainty in a large problem (even as ROM):

- a combustion problem brought to us by Luis Bravo (ARL APG)
- two articles currently in progress

CONCLUSION

What we learned and demonstrated:

- Handling uncertainty: an opportunity to do more, to draw more types of decisions
- Intervals: allow to provide guarantees on results

In addition, we also worked on handling uncertainty in a large problem (even as ROM):

- a combustion problem brought to us by Luis Bravo (ARL APG)
- two articles currently in progress

This work was done with feedback from and in collaboration with ARL researchers:

- Luis Bravo (Combustion problem) and Simon Su (Capstone project), ARL APG
- Rad Balu (Uncertainty), ARL ALC
- Craig Barker (ROM and Uncertainty), ARL, APG

CONCLUSION

What we learned and demonstrated:

- Handling uncertainty: an opportunity to do more, to draw more types of decisions
- Intervals: allow to provide guarantees on results

In addition, we also worked on handling uncertainty in a large problem (even as ROM):

- a combustion problem brought to us by Luis Bravo (ARL APG)
- two articles currently in progress

This work was done with feedback from and in collaboration with ARL researchers:

- Luis Bravo (Combustion problem) and Simon Su (Capstone project), ARL APG
- Rad Balu (Uncertainty), ARL ALC
- Craig Barker (ROM and Uncertainty), ARL, APG

The UTEP team also consists of: Miguel Argaez (Co-PI), HoracioFlorez (Post-doc at ARL ALC), Leobardo Valera (Ph.D. student at UTEP), Jesus Padilla and Phillip Hassoun (undergraduate students at UTEP).

PRODUCTS OF OUR WORK (SINCE 2014)

Conference papers: 16 + 2 in progress with Luis Bravo (ARL APG)

- among which 2 best student paper awards

Journal articles and chapters: 2 + 1 in progress

- Florez and Argaez, in Applied Mathematical Modelling (2017)
- Ceberio and Valera, in the Journal of Uncertain Systems (2016)

Invited presentations: 2

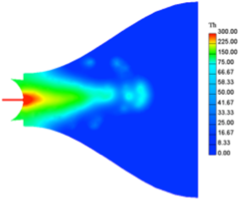
- Ceberio: Plenary talk at the International Conference SCAN'16 on Validated Computing
- Ceberio: Invited seminar at the University of Paris 6, Pierre and Marie Curie, in fall 2017

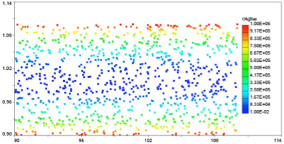
Mobile application for decisions with UQ

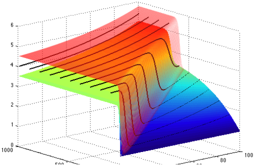
THANK YOU FOR YOUR ATTENTION

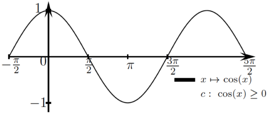
Any Questions?

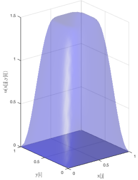
Below are illustrations of other parts of our work:

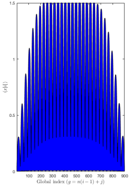


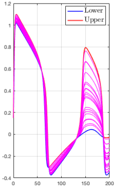


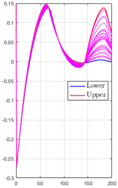












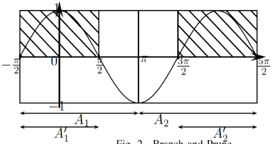


Fig. 2. Branch and Prune