Topic 3: Recursive design of experiments for sysid

Aria Alinejad, Johannes Djupesland, Torstein Nordgård-Hansen, Cameron Penne

Summary

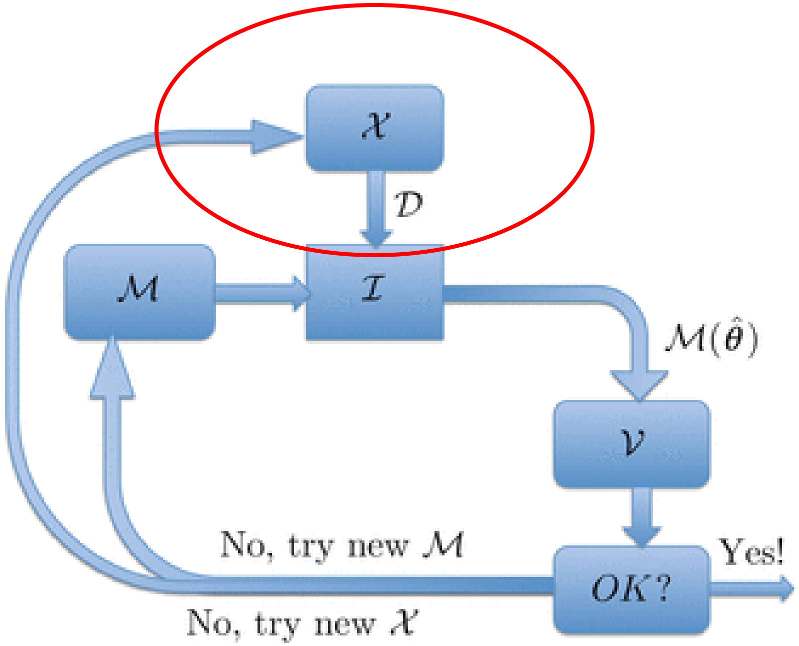
1. Introduction
   - Aria
2. General Experiment Design
   - Aria
3. General System Identification Experiments
   - Cameron
4. Example
   - Johannes

Introduction

- Black box problem
- Problem:
  - We know the input and outputs of the system
  - We do not know the structure or parameters of the model
- Goal:
  - Learn a model that reproduces the behavior of the system
    - Model structure
    - Model parameters

System identification

- $\mathcal{X}$: The experimental conditions under which the data is generated
- $\mathcal{D}$: The data
- $\mathcal{M}$: The model structure and its parameters θ
- $\mathcal{I}$: The identification method by which a parameter value $\hat{\theta}$ in the model structure $\mathcal{M}(\theta)$ is determined based on the data $\mathcal{D}$
- $\mathcal{V}$: The validation process that scrutinizes the identified model



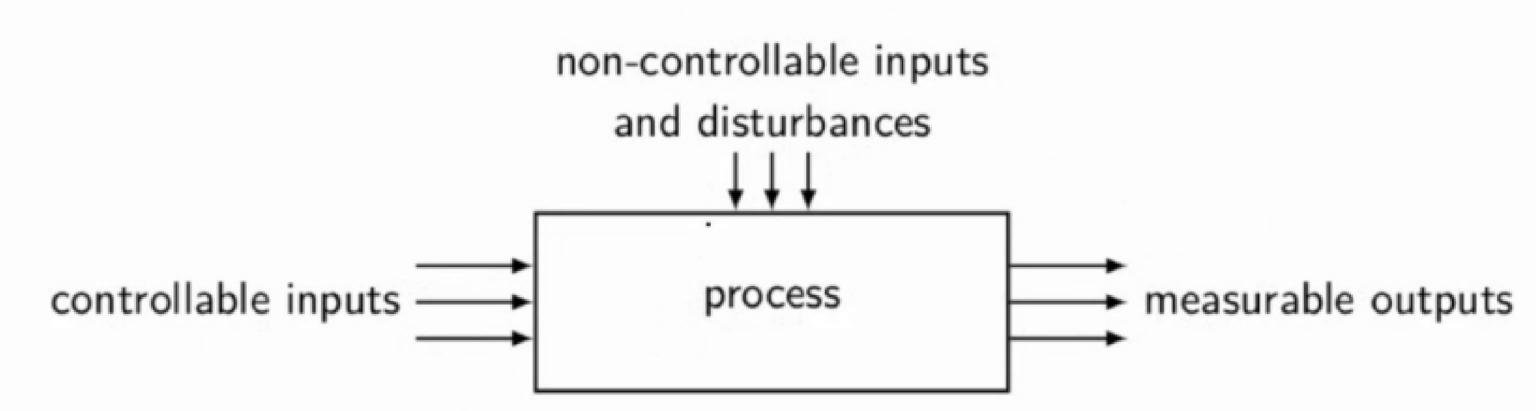
Link: [System Identification: An Overview | SpringerLink]

One Variable at a time (OVAT)

- Traditonal approach
- Assumes cause and effect relation between each variable and output
  - **Generally not true**
- DoE better
  - Idea is to: **Maximize information** from **minimal experimental effort**

Design of Experiments (DoE)

- There are also non-controllable inputs and disturbances, that can not be used in DoE



Examples (not recursive)

- Fractional factorial
- Full factorial
- Optimization designs
- Mixture designs
- Optimal designs

Source: Friends lecture notes from multivariate master course

General Experiment Design

- Offline
  - Use data already generated
- Online
  - Use data as it is generated in real-time

Offline Experiment Design

1. Data collection
2. Pre-processing
3. Model structure selection
4. Model parameter estimation
5. Model validation
   - If not satisfactory, return to step 3

Online Experiment Design

1. Model initialization
2. Data collection (continuous)
3. Model parameters update (recursive algorithm)
4. Model validation (error)
   - Run steps 2-4 until convergence

General System Identification Experiments

- Finite-history algorithms
  - Subset of data record
- Infinite-history algorithms
  - Entire data record

Finite-History Algorithms

- Finite-history algorithms only consider a limited number of past data points to update the model
    - Buffer or window that is moved forward as new data points are received
- Computationally efficient
- May miss long-term dependencies in the data, which can affect accuracy
- More stored data

Infinite-History Algorithms

- Infinite-history algorithms utilize all past data points to refine the model continuously
  - New model parameters are function of old parameters and the most recent data points
- Higher accuracy, especially for long-term trends and dependencies
- Increased computational complexity
- Less stored data
  - Use most recent data in an update step

Infinite-History Algorithms

$$\theta'(t) = \theta'(t-1) + K(t)\,(y(t) - y'(t))$$

- $\theta'(t)$ - current parameter estimate
- $\theta'(t-1)$ - previous parameter estimate
- $y(t)$ - output
- $y'(t)$ - predicted output
- $K(t)$ - update gain
  - Relates prediction error with the parameter update

Infinite-History Algorithms

$$K(t) = Q(t)\,\psi(t)$$

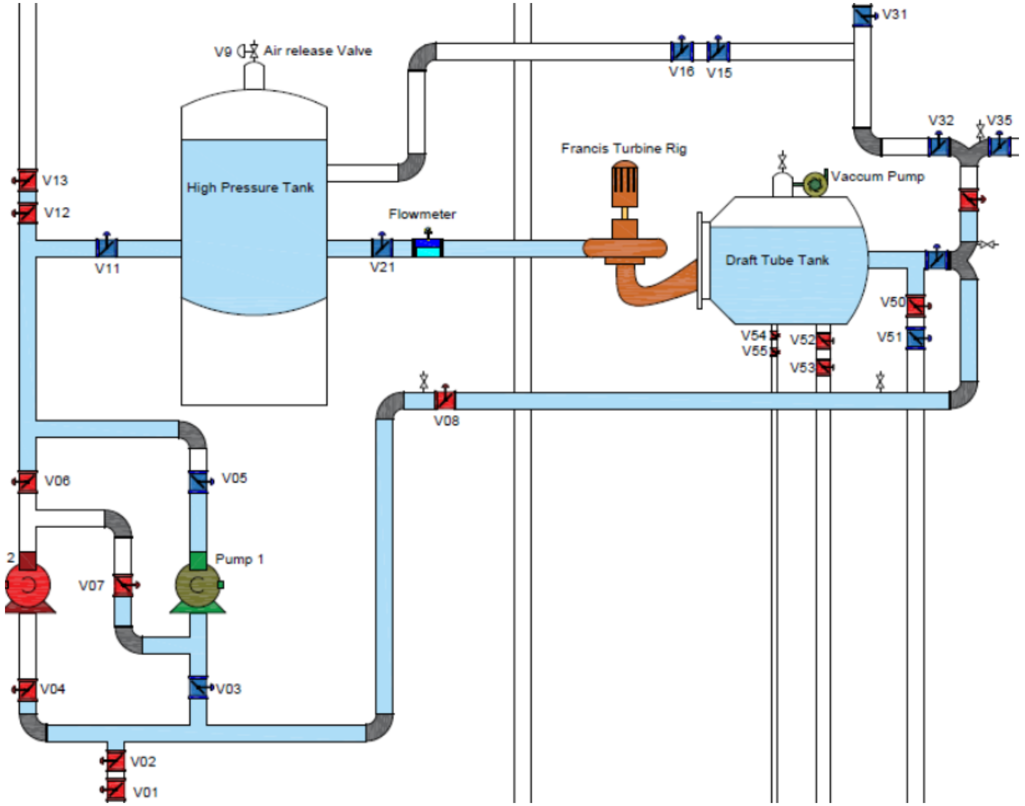
- $\psi(t)$ - regression vector, gradient of parameters
- $Q(t)$ - gain

Recursive Algorithms - Gain Q(t)

- Normalized or unnormalized gradient
    - Constant gain Q(t)
- Forgetting factor adaptation
- Kalman filter

Example

- Closed-loop Francis turbine test rig
- Objective:
  - Create a dynamic model for experienced head
- Rationale:
  - Create a control system for controlling head
- Actuator:
  - Pump
- Manipulated variables:
  - Generator speed
  - Guide vane angle



Example

- Establish initial experimental design
  - Full factorial design

Operational range

	Min.	Mid.	Max.
Pump speed			
Generator speed			
Guide vane angle			

Excitation signals

	White noise	Impulse	Ramp
Pump speed			
Generator speed			
Guide vane angle			

- Create a model and evaluate accuracy
  - Linear ARX models
  - State-space models
  - Transfer function models

Example

- Extract key insights:
  - Factor importance
  - Interactions
  - Emerging patterns
- Refine experimental design based of insights:
  - Narrow down factor ranges
  - Test higher resolution in critical areas
- Update model, reevaluate accuracy, and extract insights
- Repeat until model accuracy converge

Sources

- https://se.mathworks.com/help/ident/ug/algorithms-for-online-estimation.html
- https://www.youtube.com/watch?v=uLbjeQrQJ3Q
- https://se.mathworks.com/help/ident/ug/algorithms-for-online-estimation.html
- https://se.mathworks.com/help/ident/online-parameter-estimation.html?s_tid=CRUX_lftnav
- https://se.mathworks.com/help/ident/ug/what-is-online-estimation.html
- https://www.mathworks.com/help/ident/gs/about-system-identification.html
- https://link.springer.com/referenceworkentry/10.1007/978-1-4471-5058-9_100#Sec18279
- https://www.control.isy.liu.se/student/graduate/idkurs/Id4_4.pdf
- Design of experiments – Wikipedia
- Optimal experimental design – Wikipedia
- System identification - Wikipedia