Maschinenbau- und Betriebsinformatik
mbi

# Forschungsplattform *FLUGSIMULATION*

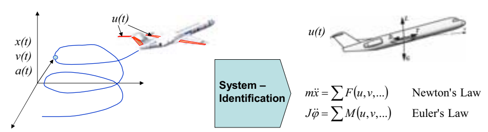
## IDENTIFICATION OF NONLINEAR DIFFERENTIAL EQUATIONS

### OBJECTIVE

**Identification of the differential equations of motion of complex dynamic systems** (like aircrafts) based on their

- trajectories and
- control parameters.

using an **evolutionary approach** and **changing the identification criterion depending on** the **available information** content of the training data.

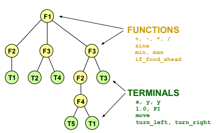



### CHALLENGE

The aircraft as a mechanical system is being described by means of **ordinary differential equations** (ODEs).

- For **simple systems**, direct modeling by **inductive methods** (deriving mathematical models based on the principles and laws of physics and engineering) is possible.
  When systems become more complex and underlying mechanisms are not clear any more, or information about a system can only be produced by means of (noisy) measurements, other approaches have to be used:
- We are especially interested in **finding physical meaningful models of mechanical systems based on noisy data**. Therefore we are seeking second order ODEs which relate the dynamic **behavior** (e.g. the trajectories of an aircraft under certain control settings) **to its causes** (accelerations/forces like *lift* and *drag*).
- Since we do not know the model structure for a certain problem a priori, we **can not simply optimize some model parameters** of a pre-defined model.
- Therefore we need to design algorithms that **also generate the model structure**, in a way that the resulting model explains the observed data in the best possible and yet **physical meaningful way**.

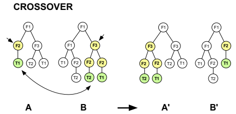
### GENETIC PROGRAMMING



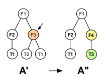

Genetic programming allows to automatically generate programs or other kind of structures, most commonly stored as trees. These trees are made up of predefined functions (operators) and terminals (e.g. variables, constants).

Starting from an initial population of structures (called individuals) new individuals are created by mating promising individuals (crossover). Individuals with better performance (fitness, objective value) are selected for mating with higher probability than others.
Offsprings created from them are likely to inherit the "good properties" from the parents.

CROSSOVER




MUTATION




Mutation is applied in addition to ensure a sufficient degree of search space exploration (avoiding local convergence)

### GENETIC PROGRAMMING FOR SYSTEM IDENTIFICATION

Depending on settings and selection criteria as well as the function and terminal sets, model types range from black-box symbolic regressions up to strongly-typed, dimensionally aware expressions. Complexity of the models is controllable by using appropriate information criteria or simply by restricting the individuals' tree sizes.

**Weaknesses of Current Approaches**

- Available GP methodologies are **not sophisticated enough to handle complex problems efficiently**. Babovic et al. (2000) showed that for complex systems a simple output error criterion does not allow the GP algorithm to find a good solution.
- Another major **problem is the generation of constants** (parameter identification) by GP, which works not very efficiently. Most of the researchers **apply additional optimization methods** (e.g. genetic algorithms, simulated annealing, and local search strategies) to optimize model parameters, once the GP system generated a structure. An approach that is **not useful if one is interested in physical meaningful** combinations of basic **constants**, e.g. the mass, the gravitational constant, densities, etc.

### MODELS USED IN SYSTEM IDENTIFICATION

**White-Box Models:**
All necessary information about a system is given. From underlying principles the model can be derived (mechanistic modeling approach).

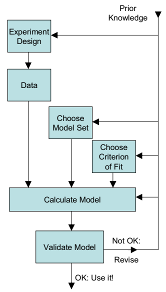
**Black-Box Models:**
Models without any information to the physical background (no a priori information).
The model parameters are basically used to fit the model behavior to the measured system data - often impossible to associate them with physical quantities of the system. Often used in system identification*).

**Grey-Box Models:**
They are a blend of the two models above:
- The model structure may be deduced from available information about the system.
- There are additional adjustment-parameters which could be tuned to compensate the lack of knowledge and improve the fit to the data.

In many cases, values of certain parameters can be associated with some (unknown) physical parameters. That helps to understand and describe the considered system.

*) Examples: FIR (finite impulse response), ARMAX (AutoRegressive Moving Average with eXogenous inputs), Box-Jenkins, NARMAX (Nonlinear ARMAX), neural networks and wavelets, ...

### MODEL GENERATION IN SYSTEM IDENTIFICATION




**Manual Model Generation:**
- For each of the described steps, appropriate tools (numerical methods) could be applied manually and supervised by the engineer or researcher.
- Results are analyzed and decisions are made by human expertise.
- Model structure is modified by the engineer.

**Semi-Automatic Generation of Models:**
- Tools are used which assist the engineer in the model structuring task.
- There are strategies available for model simplification or model improvement (e.g. grow methods in neural network training).

**Automatic Generation by Means of Evolutionary Computation:**
This covers the whole identification loop (excl. the measurement process) by using algorithms based on the principles of biological evolution generating
- model structures and
- variables, constants, parameters, and mathematical expressions at the same time.

### IDENTIFICATION CRITERIA

**Output-Error-Approach:**

$$\left\| y(u) - \hat{y}(u) \right\|$$

where:
$y$ is a vector of measurement data,
$u$ the control input, and
$\hat{y}$ the vector containing the model output.

System of two first order ODEs:
$$\dot{y} = v$$
$$\dot{v} = a = f(y, v, u) \qquad \text{with } \dot{(\cdot)} = \frac{d(\ )}{dt} \text{ and } \frac{\Delta(\ )}{\Delta t} = \text{num.diff.}$$

**Identification Criteria:**

$$\left\| y(u) - \iint \hat{a}(u) \mathrm{d}t^2 \right\|$$

**Resulting trajectory:**
Disadvantage: algorithmic complexity (high computational cost for solving the ODE numerically with good precision and stability).

$$\left\| \frac{\Delta y(u)}{\Delta t} - \int \hat{a}(u) \mathrm{d}t \right\|$$

**Resulting velocity:**
One ODE integration only. Disadvantage: numerical differentiation of $y$ is just an approximation of $v$, the information content will be less and signal-to-noise ratio smaller.

$$\left\| \frac{\Delta^2 y(u)}{\Delta t^2} - \hat{a}(u) \right\|$$

**Direct approximation of acceleration:**
No ODE solving is required. The "true" $a$ is approximated by numerically calculating the second derivative of $y$, the signal-to-noise ratio decreases further

### CHOOSING THE MOST APPROPRIATE IDENTIFICATION CRITERION

Because of its reduced computational complexity for optimization, information criterion "acceleration" is preferred to criteria "velocity" and "displacement".
When the data's information content is exploited and further training would lead approximation already noise (over-fitting) the criterion has to change. Therefore the algorithm will switch to a criterion with better signal-to-noise ratio.

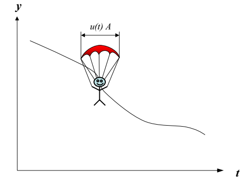
**Basic Algorithm**

```
1. Select criterion/data with least computational costs with
   respect to optimization (algorithmic complexity)

2. Use it for model building as long as there is useful
   information content

3. If stopping criteria are fulfilled: stop identification
   procedure

4. If not: Select other criterion/data with more information
   content and lowest possible algorithmic complexity

5. Continue with step 2
```

## COMPUTATIONAL EXAMPLE / EXPERIMENTAL RESULTS

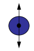
### MODEL

**Parachute Model**




**Mechanical Model**

$$F_D = \frac{\rho \cdot v^2}{2} c_w \cdot u \cdot A$$

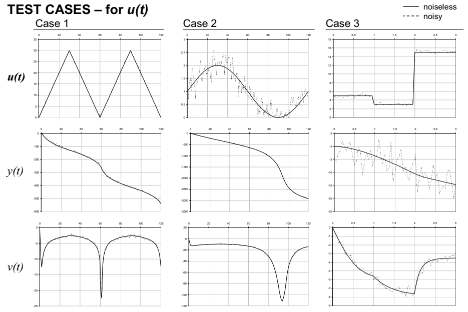


$$F_G = m \cdot g$$

**ODE system**

$$\dot{y} = v$$
$$\dot{v} = a = -g + \frac{1}{2m} c_w \cdot \rho \cdot u \cdot A \cdot v^2$$

**TEST CASES – for $u(t)$**




### ERROR CALCULATION AND FITNESS MEASURE

**Error for each test case:**

$$CaseError_k = \frac{1}{N_k} \sum_{i=1}^{N_k} e_{ki}^2$$

where $N_k$ is the number of intervals for case $k$, and $e_{ki}$ is the error for the $i^{th}$ interval of case $k$.

**Error for all test cases:**

$$Error = \sum_{k=1}^{N} c_k \cdot CaseError_k$$

where $N$ is the number of cases and $c_k$ is a weight to emphasize certain cases. In our cases $c_k$ is 1/3 for all three cases.

**Normalized Fitness:**

$$fitness = \frac{1}{1 + Error}$$

which is bounded between 0 and 1.
Lower Error causes a higher fitness measure.

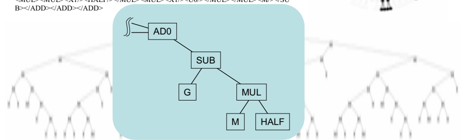
### EXPERIMENTAL SETUP

| | Experiment | Identification Criterion | Test Set + ODE Solver | Validation Set + ODE Solver |
|---|---|---|---|---|
| clean data | Experiment 1 | $\left\| \frac{\Delta v(u)}{\Delta t} - \hat{a}(u) \right\|$ | calculated acceleration from clean data; no ODE solving required | clean data ($y(u)$) RK-4 (20/interval) |
| | Experiment 2 | $\left\| y(u) - \iint \hat{a}(u) \mathrm{d}t^2 \right\|$ | trajectory (clean data); RK-4 (10/interval) | clean data ($y(u)$) RK-4 (20/interval) |
| | Experiment 3 | same as Experiment 1 for 40 generations, then same as Experiment 2 | same as for Experiment 1 for 40 generations, then same as for Experiment 2 | clean data ($y(u)$) RK-4 (20/interval) |
| noisy data | Experiment 4 | $\left\| \frac{\Delta v(u)}{\Delta t} - \hat{a}(u) \right\|$ | calculated acceleration from noisy data; no ODE solving required | clean data ($y(u)$) RK-4 (20/interval) |
| | Experiment 5 | $\left\| y(u) - \iint \hat{a}(u) \mathrm{d}t^2 \right\|$ | trajectory (noisy data), RK-4 (10/interval) | clean data ($y(u)$) RK-4 (20/interval) |
| | Experiment 6 | same as Experiment 4 for 40 generations, then same as Experiment 5 | same as for Experiment 4 for 40 generations, then same as for Experiment 5 | clean data ($y(u)$) RK-4 (20/interval) |

### RESULTS – ODE IDENTIFICATION – GP RAW OUTPUT

**Best of Experiment 3 – fitness= 0.5538:**


<ADD><ADD><ADD><ADD><SUB><U0/><M/></SUB><ADD><HALF/><MUL><X1/><MUL><X1/><U0/></MUL></MUL></ADD></ADD><SUB><SUB><SUB><SUB><U0/><M/></SUB><MUL><M/><HALF/></MUL></SUB><G/></SUB><U0/></SUB></ADD><ADD><ADD><SUB><SUB><SUB><U0/><M/></SUB><MUL><M/><HALF/></MUL></SUB><U0/></SUB><ADD><HALF/><MUL><X1/><MUL><X1/><U0/></MUL></MUL></ADD></ADD><SUB><G/><MUL><M/><HALF/></MUL></SUB></ADD></ADD><ADD><SUB><SUB><SUB><MUL><MUL><X1/><C/></MUL><MUL><X1/><U0/></MUL></MUL><M/></SUB><SUB><M/><MUL><MUL><X1/><HALF/></MUL><MUL><X1/><U0/></MUL></MUL></SUB></SUB><M/></SUB><ADD><ADD><SUB><SUB><SUB><HALF/><M/></SUB><MUL><M/><HALF/></MUL></SUB><U0/></SUB><ADD><HALF/><MUL><X1/><MUL><X1/><U0/></MUL></MUL></ADD></ADD><SUB><MUL><MUL><X1/><HALF/></MUL><MUL><X1/><U0/></MUL></MUL><M/></SUB></ADD></ADD></ADD>


### RESULTS – ODE IDENTIFICATION – GP RAW OUTPUT

**Best of Experiment 3 – simplified*:**

$$\hat{a} = \frac{2}{m} - 10 + (4 + c_w) u v^2$$

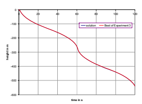
**Analytical solution:**

$$a = -g + \frac{1}{2m} c_w \rho u A v^2$$

Substituting the constants for $m$ and $c_w$:

$$\hat{a} = -9.975 + 0.05425 u v^2 \qquad a = -9.807 + 0.053125 u v^2$$

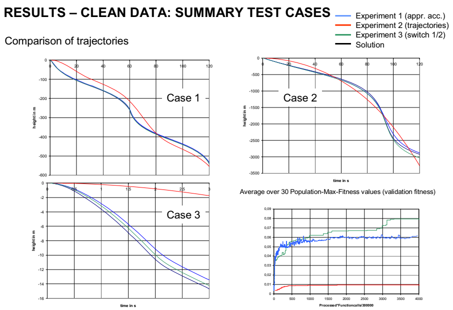
**Comparison:**
- very good numerical approximation
- $u v^2$ – term qualitatively correct
- physical constants not always used correctly



*) Using *Mathematica*

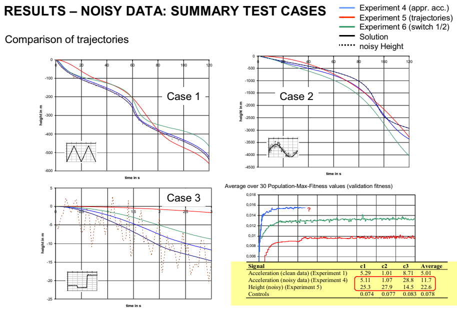
### RESULTS – CLEAN DATA: SUMMARY TEST CASES

Comparison of trajectories




### RESULTS – NOISY DATA: SUMMARY TEST CASES

Comparison of trajectories




| Signal | c1 | c2 | c3 | Average |
|---|---|---|---|---|
| Acceleration (clean data) (Experiment 1) | 5.29 | 1.01 | 8.71 | 5.01 |
| Acceleration (noisy data) (Experiment 4) | 5.11 | 1.07 | 28.8 | 11.7 |
| Height (noisy) (Experiment 5) | 25.3 | 27.9 | 14.5 | 22.6 |
| Controls | 0.074 | 0.077 | 0.083 | 0.078 |


From: *Algorithms for Identification of Nonlinear Differential Equations for Complex Dynamical Systems based on Experimental Data*, S.Vössner, T.Buchsbaum, IFORS 2005


# *SYSTEM IDENTIFICATION*

Das Land Steiermark
Innovation, Infrastruktur und Energie

TUG