Integration of Hybrid System Identification using Symbolic Regression into a Modeling Framework for Cyber-Physical Systems

TUHH
Hamburg University of Technology

29.09.2025

Harshitha Viswanath
Supervisor: Swantje Plambeck

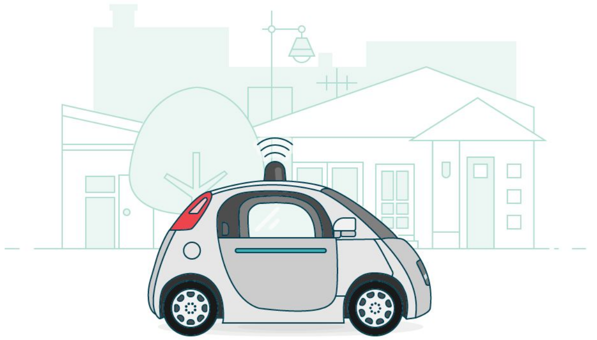
Agenda:

TUHH

Introduction – Cyber Physical System (CPS)

TUHH

- Integrated system that combines computation and communication with physical processes
- Example: Autonomous Driving
  - Cyber components
    - Onboard computer
    - Sensors
  - Physical components
    - Actuation, and power systems
  - Process sensor data and determine acceleration, braking, or steering
  - Send commands to physical components that execute maneuvering
  - **Safety critical**: Small delays/errors can trigger major failures


source: innovationcloud.com

Introduction – Modeling of CPS with Hybrid System

- **Hybrid Automaton**—widely used CPS abstraction
- System modeled as **discrete modes** (control logic) with **flow functions** (physical processes), transitioning via guard conditions
- Example: Smart Sensor
  - Two discrete modes of operation:
    - **Compression** (mode=1): governed by square root law

$$\mathbf{y = \sqrt{(u+1)}}$$

    - **Amplification** (mode=2): governed by quadratic law

$$\mathbf{y = 5u^2 + 3}$$

  - Guard conditions: Switch to mode 2, if mode = 1 and u >= 2; switch to mode 1, if mode = 2 and u <= 1

Introduction – Symbolic Regression (SR)

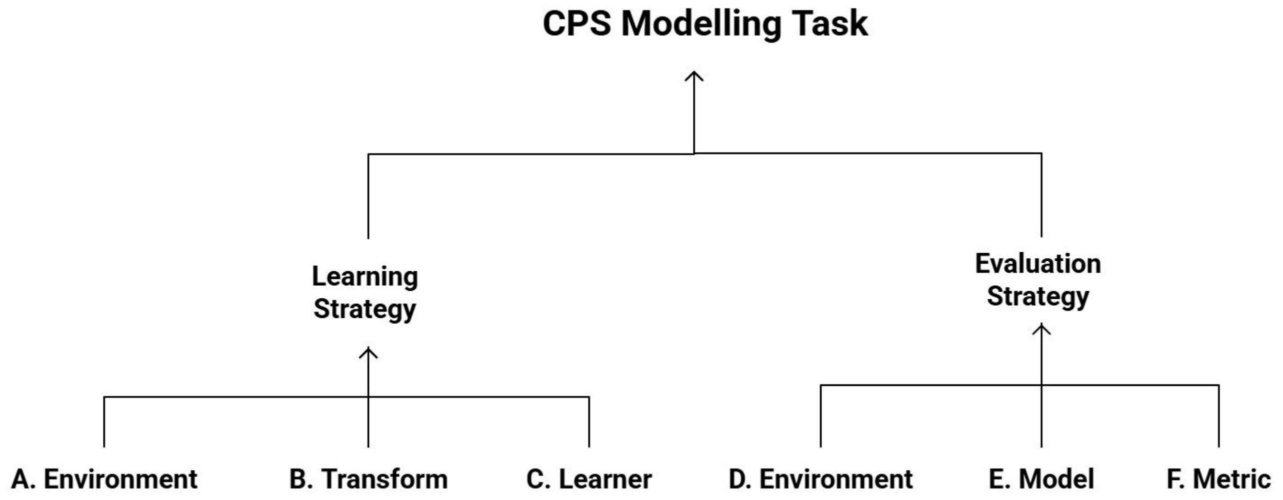
TUHH

- Nascent ML field infers **mathematical expressions** from data
- SR poses **data optimization task** spanned by analytical expressions
    - Optimization of **prediction error** and **complexity**
- **Interpretability** is the key, alongside accuracy
- **PySR**—an open-source Python library for human-interpretable symbolic models
- Approach: Uses genetic programming (on **evolutionary algorithm**)
- Employs **Pareto Front**: trade-off solutions where **accuracy cannot improve without increasing complexity** (or vice versa).

Foundations – Flowcean

TUHH

- Automated **CPS model generation** via **data-driven learning**
- **Modular architecture** integrating multiple learning libraries and tools
- Conventional ML frameworks (e.g., PyTorch, Scikit-learn) lack CPS-specific learning pipelines

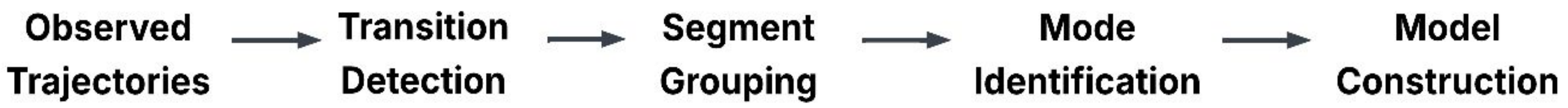


Foundations – Flowcean : Modeling Pipeline

- **Environment:** define dataset and compatible learning algorithm
- **Transform:** data preprocessing for effective learning (e.g., mean, median, mode)
- **Learning Strategy:** integrates environment, transform and learner to produce model
  - **Learner**—an algorithm that trains on data
  - **Model**—final system abstraction produced
- **Evaluation Strategy**: assess model performance based on the **Metric** (e.g., MSE, MAE)

Foundations – Hybrid System Identification using SR

TUHH



- **Central Idea**: detection of transitions by analyzing the **underlying system dynamics** directly from data
- Mechanism of system identification
  - **Identify governing equations** of continuous behaviors
  - **Separate data segments** that have separate behaviors
  - **Group segments** that share the same behaviors
- Outcome: **Human-readable equations**, enabling interpretations and deeper insights into the system behavior

Foundations – Hybrid System Identification using SR

Transition Detection

- Set **starting point**, **window size**, step size, **expression loss** (MSE)
- Move window along the data; **learn expression** and check the loss
- If **loss changes**, **mark the transition**; store segment and expression
- **Reset window** and continue until all the data points are processed

Segment Grouping and Mode Identification

- Set **first detected segment as first group**
- Combine each **new segment with existing group** and check for loss
- If **combined loss** is less, then add to the group; else, create a new group
- Continue until all segments are assigned

Motivation

Modeling Essentials:

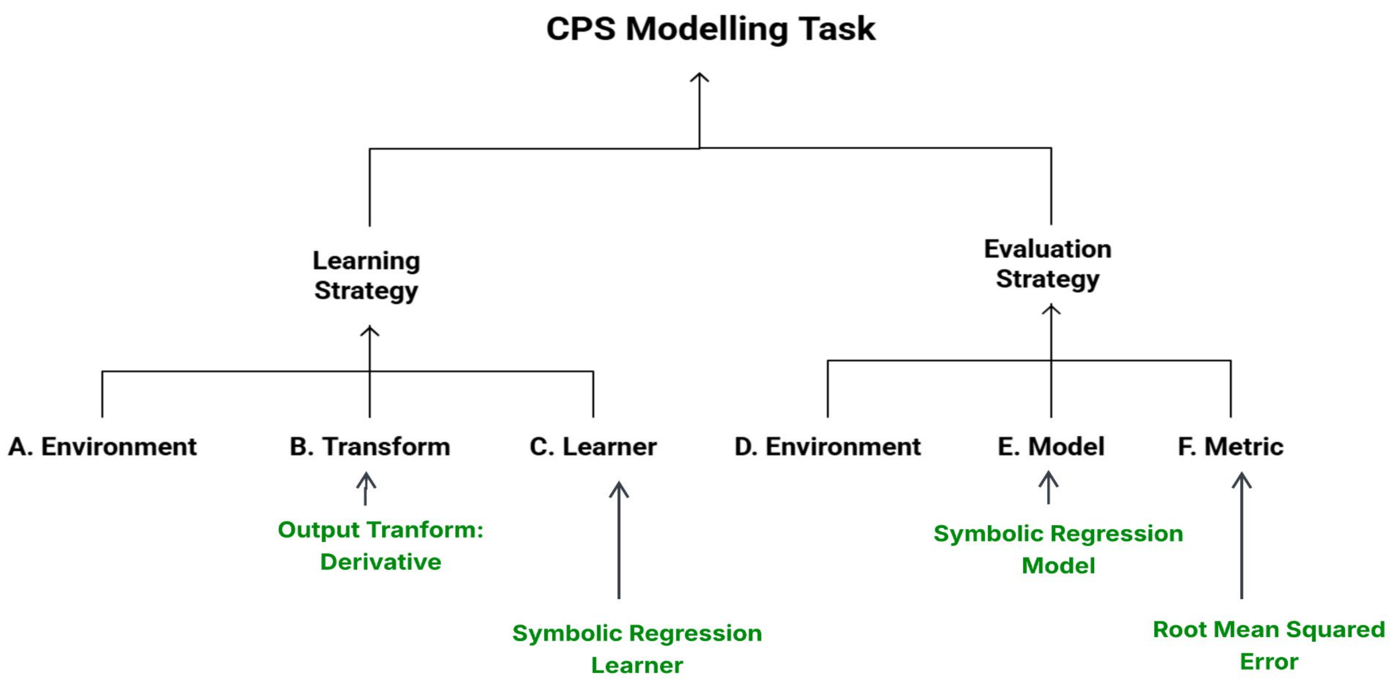
- ❑ **Data-driven modeling framework** dedicated for *CPS applications*
- ❑ **Hybrid automata** that naturally unifies the *continuous physical processes* and *discrete cyber systems* of the CPS
- ❑ **SR** for producing results in accurate, *human-interpretable* models

⇩

Flowcean + Hybrid System Identification using SR

Implementation

TUHH



Implementation – Derivative Transform

- ➔ Computes the first-order differences between consecutive data points, measuring the change of the variable over time
- ➔ Mathematically expressed as

$$\square y_t = y_t - y_{t-1}$$

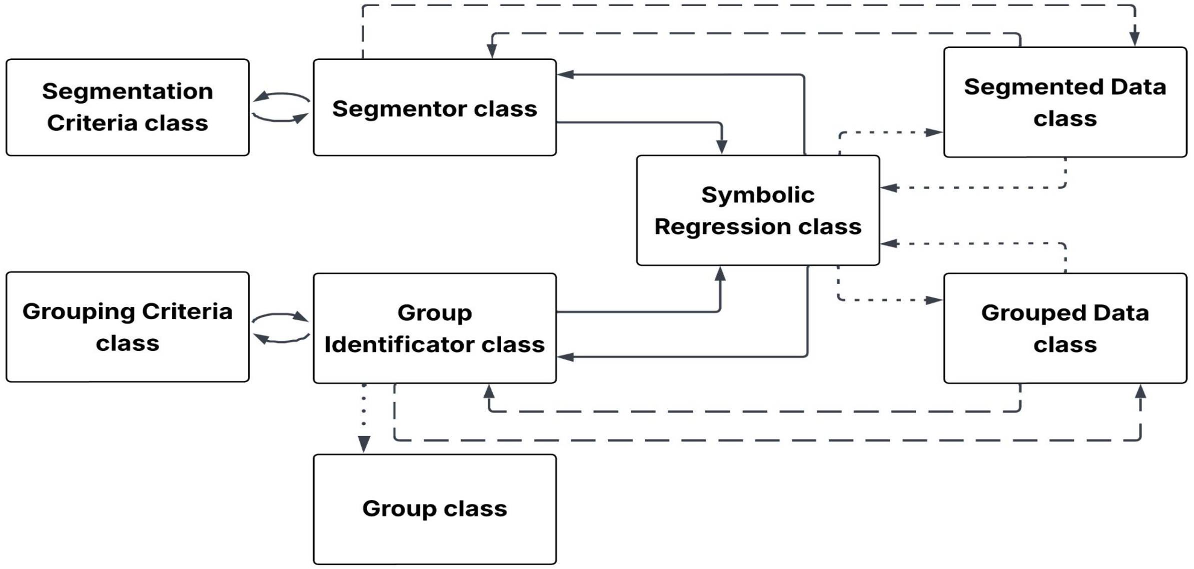
where,

y□ = current value at time t

y_{t-1} = previous value at time t-1

- ➔ Applicability:
  - ◆ Raw input data provides absolute measurements
  - ◆ The model often requires change in the variables recorded, to capture the continuous dynamics

Implementation – SR Learner



➔ **Training algorithm**: culmination of **transition detection**, **segment grouping**, and **mode identification** into a **single learner** module

Implementation – SR Learner

TUHH

- ➔ Elimination of verbose YAML configs
- ➔ **Input parameters** to the learner classified into two categories
  - ◆ **User-defined arguments**
    - ● Dynamic, CPS-specific (e.g., sliding window frame length)
    - ● Provided at initialization of SR learner
  - ◆ **Default arguments**
    - ● Hyper-parameter passed to PySR() (e.g., population number)
    - ● Fixed within SR learner module
- ➔ Precise parameter tuning
  - ◆ **Configurable parameters**: can override default arguments
  - ◆ Use case: parameter-sensitive segmentation and grouping

Implementation – SR Model

- ➔ **Static grouping: same data for training and evaluation**
    - ◆ Maps group start and end indices (from SR learner) on data
    - ◆ Assigns a group ID to each data point
- ➔ Hybrid system representation
    - ◆ Data points with **same group ID form discrete modes**
    - ◆ Each mode linked to its **symbolic equation**, representing continuous dynamics
- ➔ Prediction and evaluation
    - ◆ Symbolic equations evaluated to generate numerical values
    - ◆ **Output**: dataframe with **predicted numerical values**
    - ◆ Enables **metric-based evaluation** (e.g., MSE, MAE)

Implementation – Metric

- ➔ Existing Flowcean metrics
  - ◆ Mean Squared Error (MSE)
  - ◆ Mean Absolute Error (MAE)
  - ◆ R² Score, Max Error
- ➔ **New addition: Root Mean Squared Error (RMSE)**

$$\sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}$$

where, y_i = actual values; y_i^ = predicted values; N = total data points

- ➔ Sensitive to large deviations (unlike MAE)
- ➔ Higher stability (no division by near-zero value)
- ➔ Same units as prediction values → easier to interpret than MSE

Optuna – Hyper-parameter Tuning

- ➔ SR accuracy and efficiency depend on hyper-parameter values
- ➔ Manual tuning is **time-consuming** and **non-reproducible**
- ➔ **Optuna tuning: data-driven** search strategy
  - ◆ Objective: **minimize loss** (MSE) across entire dataset
  - ◆ Explores **multiple hyper-parameter combinations**
  - ◆ **Reduces ambiguity** in selection of hyper-parameter values
- ➔ Implementation
  - ◆ Optimization over **20 trials** per case study
  - ◆ Optimized set of hyper-parameters: Niterations, Parsimony, Binary and unary operators, Number of populations, and Model selection

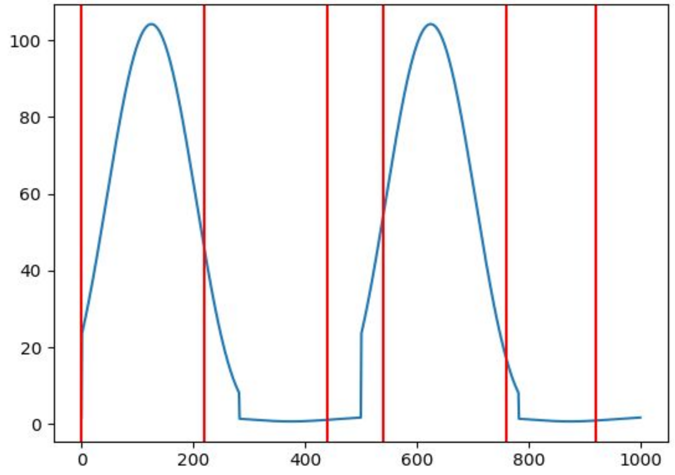
Case Study – Smart Sensor

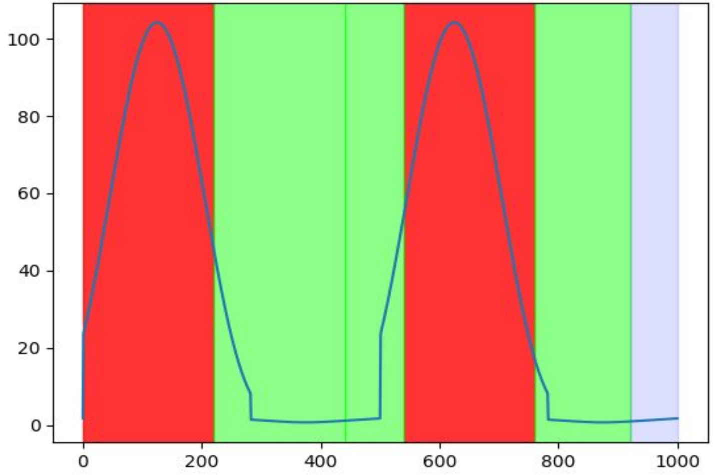
- ➔ System Overview:
  - ◆ Alternates between **two operational regimes**
    - ● Compression (q = 1) : $\mathbf{y = \sqrt{(u+1)}}$
    - ● Amplification (q = 2) : $\mathbf{y = 5u^2 + 3}$
- ➔ Mode switching when input *u(t)* crosses over guard conditions
- ➔ Guard conditions
  - ◆ If q = 1 and *u* >= 2, switch to mode 2
  - ◆ If q = 2 and *u* <= 1, switch to mode 1
- ➔ Dataset
  - ◆ Records time, input *u(t)*, output *y*
  - ◆ Captures detailed switching behavior

Case Study – Smart Sensor

TUHH

Parameter	Value	Importance (%)
niterations	120 (segmentation only)	9%
parsimony	5.158×10^{-5}	3%
binary_operators	[+, -, *]	62%
unary_operators	[sqrt, sin, cos]	3%
populations	48	4%
model_selection	best	5%





Case Study – Smart Sensor

Group 0: $y = u \cdot (u \cdot 4.781374 + 1.6382203)$

Group 1: $y = \dfrac{u^2}{0.19230054}$

Group 2: $y = \sqrt{u + 1}$

- ➔ Start width = 100; Step width = 60 (Sliding window)
- ➔ Evaluation metrics: MSE = 5.732 & RMSE = 2.394
- ➔ **Insights from grouping**
  - ◆ Group 0 & Group 2 → distinct mode behaviors
  - ◆ Group 1 → **mixed-mode dynamics**
    - ● Captures **overlap of both modes** due to sliding window width; identifies as a separate mode

Conclusion & Future Work

- ❖ **Seamless integration** of toolchain with emphasis on hybrid systems
- ❖ **Key enhancements**
  - ➢ **Derivative transform + SR learner + SR model + RMSE metric**
  - ➢ **Optuna** integration for efficient hyper-parameter tuning
  - ➢ **Validation from case studies**: robust hybrid system identification leveraging Flowcean's modular architecture
- ❖ **Future Work**
  - ➢ Enhanced hyper-parameter optimization
  - ➢ Improved interpretability using SymPy library
  - ➢ Scalability to high-dimensional data sets
  - ➢ Data-driven selection of segmentation window parameters

References

[1] Schmidt, M., Plambeck, S., Fey, G., Knitt, M., Rose, H., Wieck, J. C., Balduin, S., "Flowcean—Model Learning for Cyber-Physical Systems," Institute of Embedded Systems & Institute of Logistics Engineering, Hamburg University of Technology, Hamburg, Germany; Fraunhofer CML, Hamburg, Germany; OFFIS - Institute for Information Technology, Oldenburg, Germany.

[2] Plambeck, S., Schmidt, M., Fey, G., Subias, A., Travé-Massuyès, L., "Dynamics-Based Identification of Hybrid Systems using Symbolic Regression," Hamburg University of Technology, Germany; LAAS CNRS & INSA Université de Toulouse, France.

[3] Makke, N., Chawla, S., "Interpretable Scientific Discovery with Symbolic Regression: A Review," Qatar Computing Research Institute, HBKU, Doha, Qatar, May 3, 2023.

[4] Cranmer, M., "Interpretable Machine Learning for Science with PySR and SymbolicRegression.jl," Princeton University, Princeton, NJ, USA; Flatiron Institute, New York, NY, USA; arXiv:2305.01582v3 [astro-ph.IM], May 5, 2023.

Thank you very much.

Hamburg University of Technology (TUHH)
Am Schwarzenberg-Campus 1
21073 Hamburg
www.tuhh.de

tuhh.de

TUHH
Hamburg
University of
Technology