Toward transparent and physically consistent machine learning models

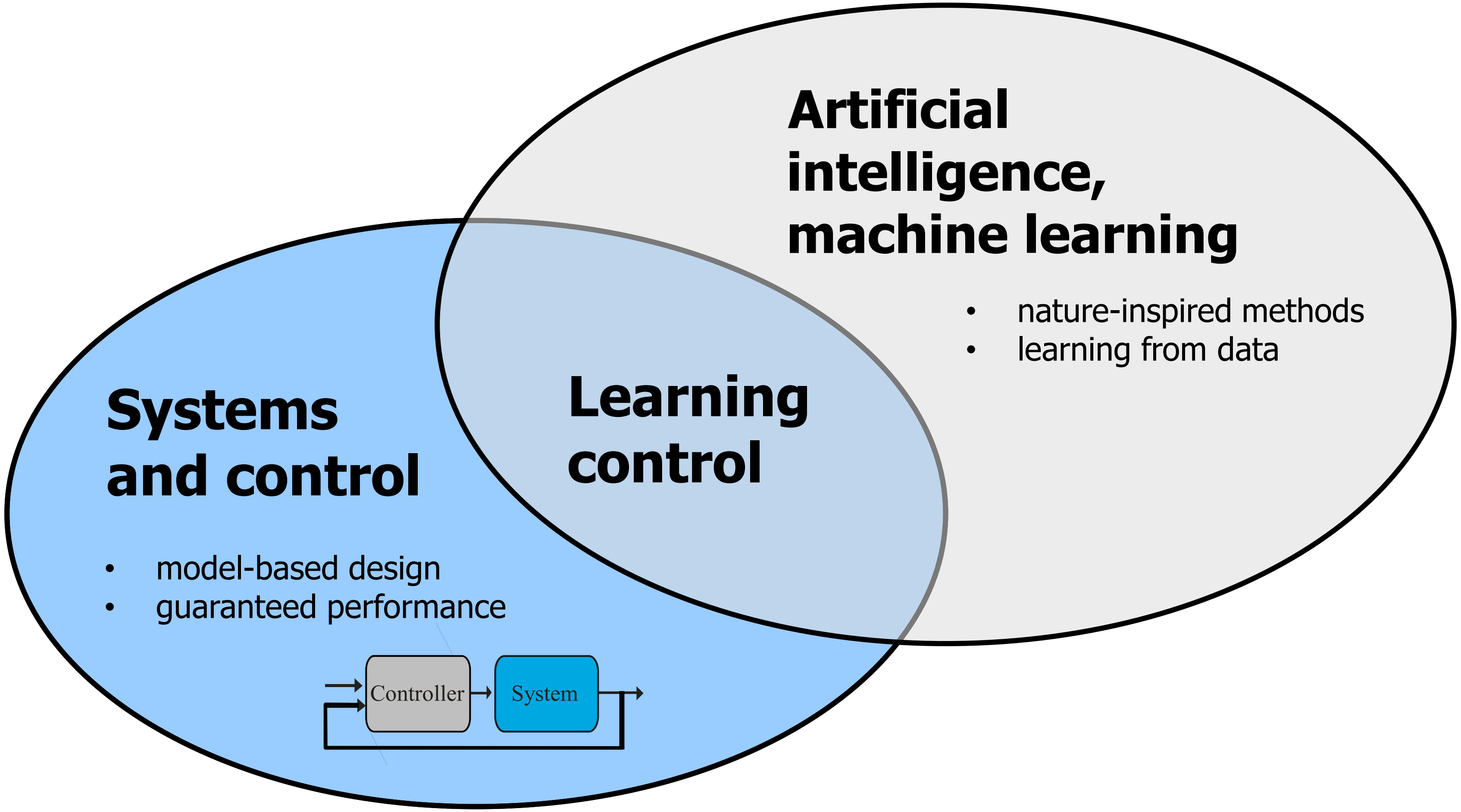
Robert Babuška

Czech Institute of Informatics, Robotics, and Cybernetics, CTU in Prague, Czech Republic

Department of Cognitive Robotics, TU Delft, The Netherlands

email: r.babuska@tudelft.nl

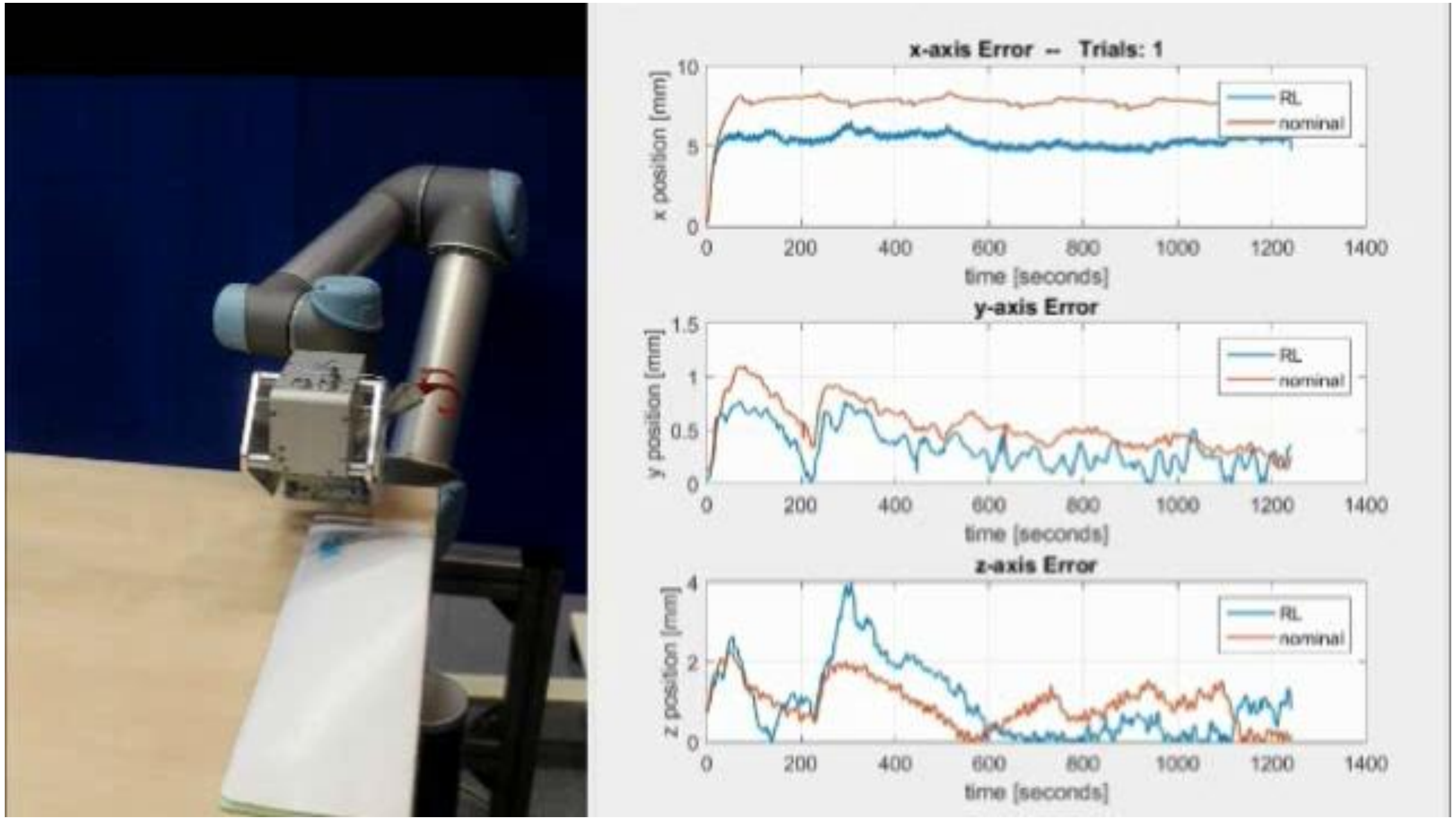
Background


Artificial intelligence, machine learning
• nature-inspired methods
• learning from data
Systems and control
• model-based design
• guaranteed performance
Learning control
Controller
System

Current research interests

Reinforcement learning for nonlinear motion control in robotics

- Performance optimization in repetitive tasks

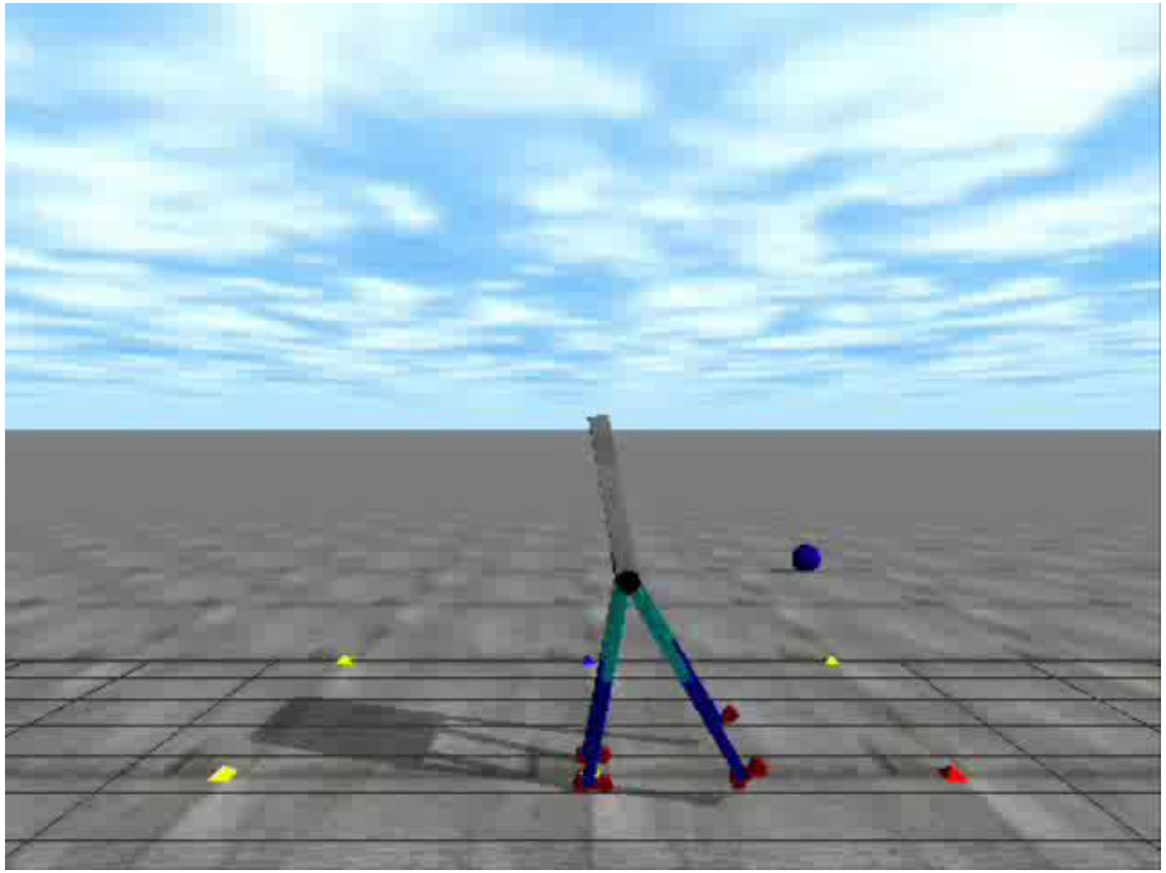


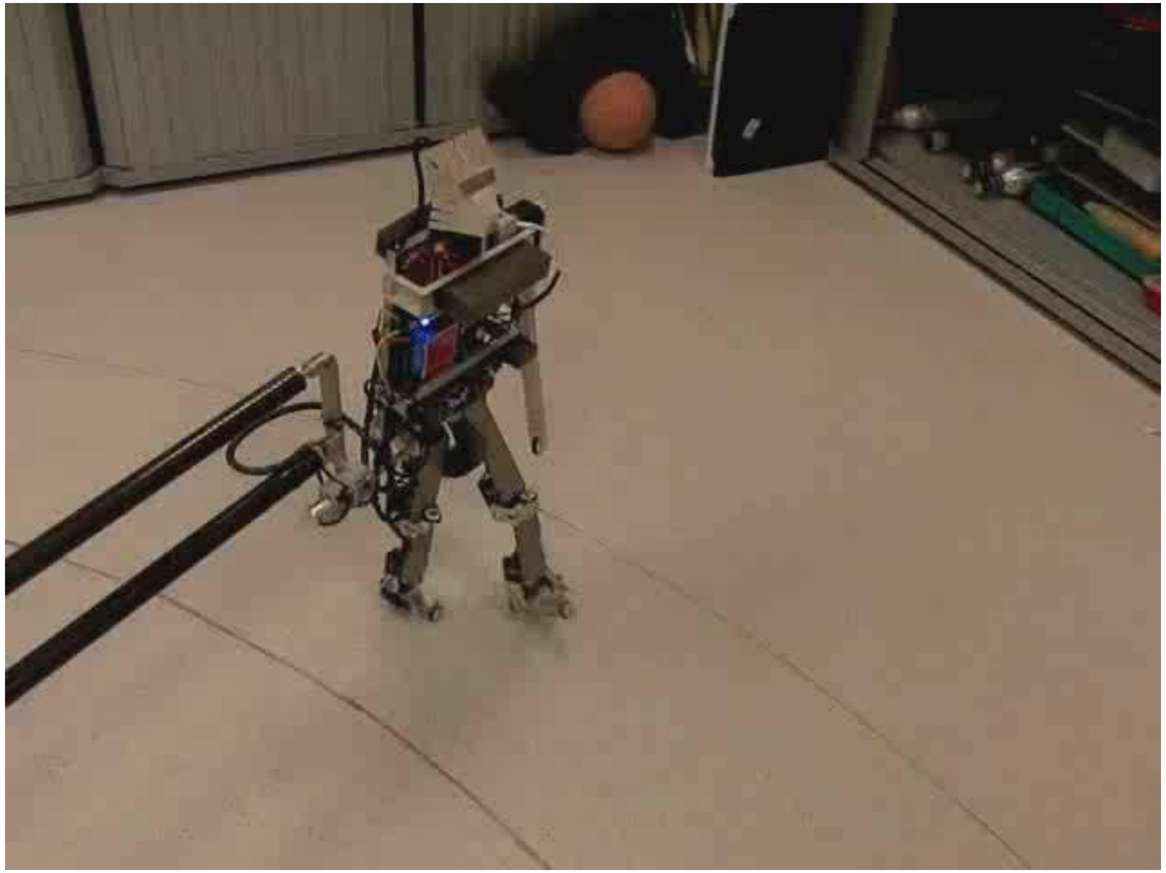
Research interests

Reinforcement learning for nonlinear motion control in robotics

- Performance optimization in repetitive tasks
- Bipedal walking

Model-based reinforcement learning

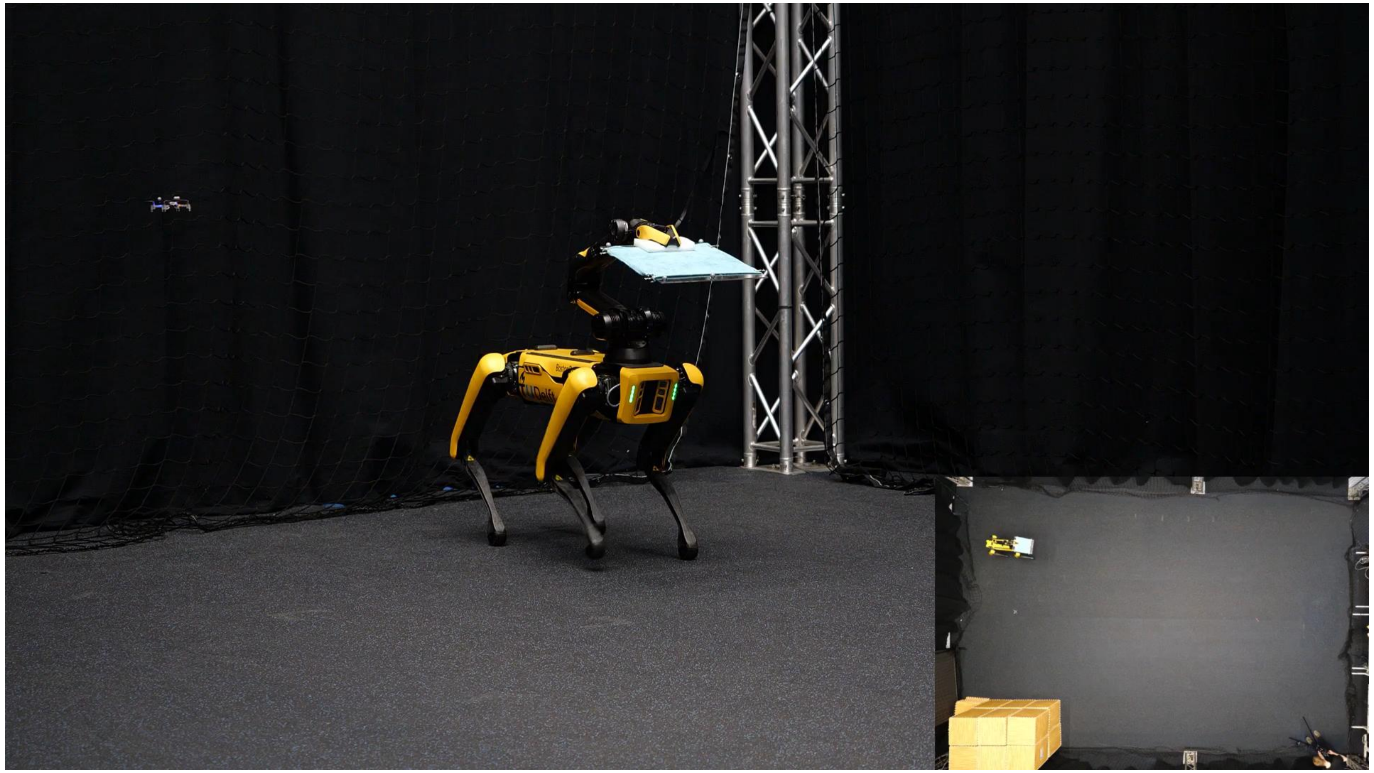




Research interests

Reinforcement learning for nonlinear motion control in robotics

- Performance optimization in repetitive tasks
- Bipedal walking
- Complex flight maneuvers



Research interests

Reinforcement learning for nonlinear motion control in robotics

- Performance optimization in repetitive tasks
- Bipedal walking
- Complex dynamic maneuvers

Machine learning for vision-based robot control in unstructured environments

- Grasping and manipulation of complex objects

Clearing a pile of tomato trusses

Results:

- Successfully clearing the pile in every attempt
- 93% needed one attempt
- 6% needed two attempts
- 1% required more than two attempts

Research interests

Reinforcement learning for nonlinear motion control in robotics

- Performance optimization in repetitive tasks
- Bipedal walking
- Complex dynamic maneuvers

ML for vision-based robot control in unstructured environments

- Grasping and manipulation of complex objects
- Tree pruning
- Fruit harvesting

Learning nonlinear dynamic models from data by using symbolic regression

Limitations of deep learning and neural networks

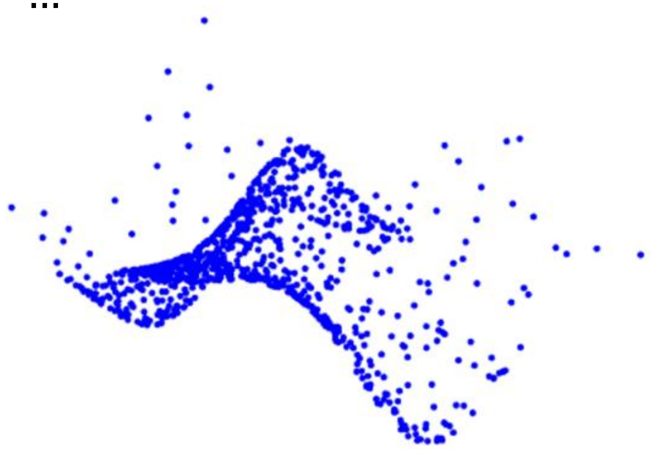
- Large amounts of training data
- Many hyperparameters
- Black-box models
- No straightforward way to include prior knowledge

Symbolic regression as an alternative machine learning approach

Symbolic regression

Data (from experiments or operation)

x1	x2	y
-3.14	-30.10	-23.34
-2.93	-10.14	-22.67
-2.72	2.31	-22.07
-2.30	13.22	-21.29
...		



Prior knowledge

f(0) = 0

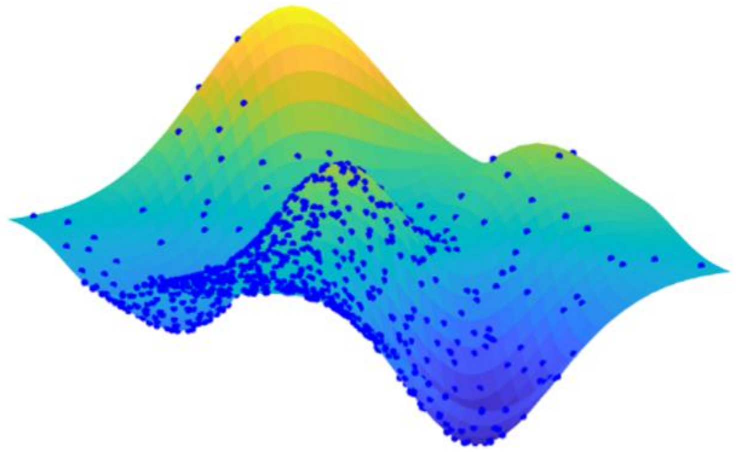
monotonicity

extremes, symmetry

etc.

Analytic model

y = -15.42 + 2.43 * (-1.49*x1 + 0.51*x2 + 0.07)
 + sqrt(x2 + power(2.17*x1 -2.93*x2, 2) + 1)
 ...



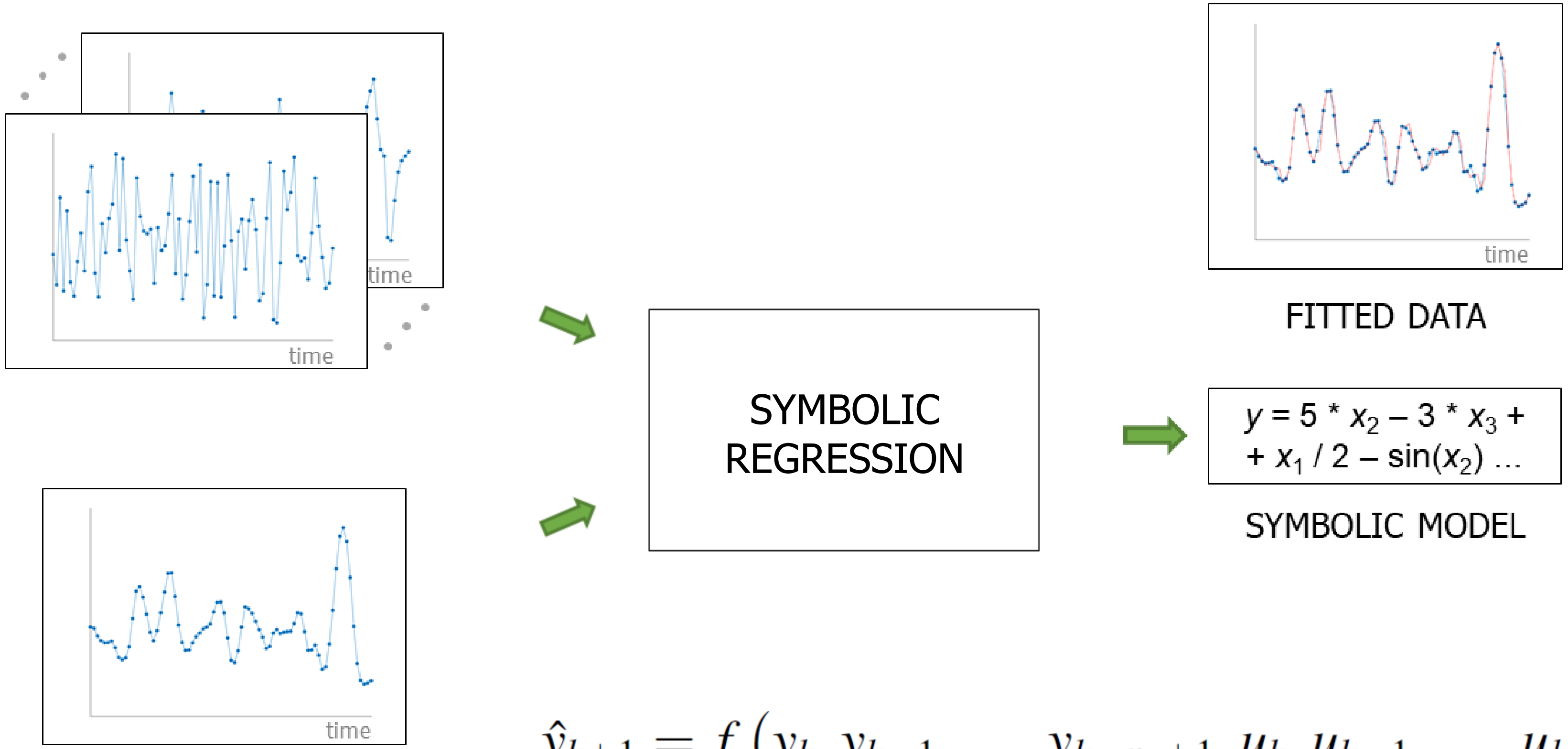
Model must be

accurate

simple

physically plausible

Modeling dynamic systems



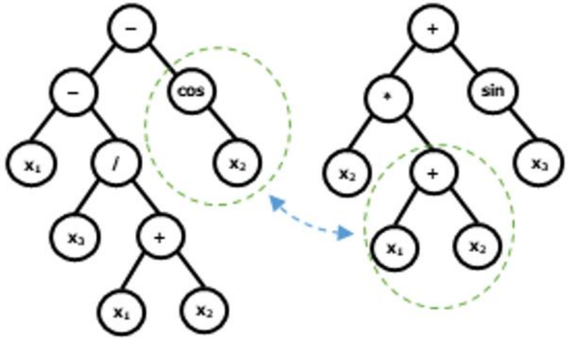
$$\hat{y}_{k+1} = f\left(y_k, y_{k-1}, \ldots, y_{k-n_y+1}, u_k, u_{k-1}, \ldots, u_{k-n_u+1}\right)$$

Predicted output — Past outputs — Past inputs

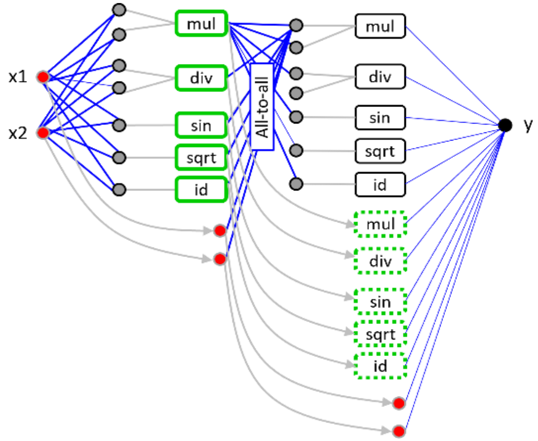
Nonlinear autoregressive with exogenous input model (NARX)

Symbolic regression methods

1. Genetic algorithms and evolutionary programming
   gradient-free optimization, genetic operators, global exploration



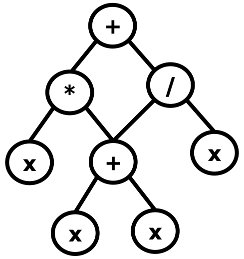
2. Feedforward multilayer neural networks
   gradient-based optimization, sparse topology, fine-tuning coefficients

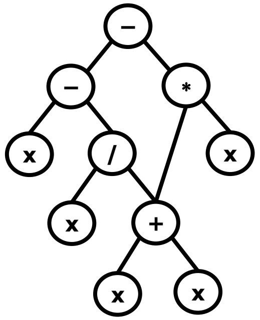


3. Transformers and foundation models
   train a transformer on a large number of data – formula pairs

Genetic algorithms for symbolic regression

$$y = \sum_{j=0}^{n_f} \alpha_j F_j(x_1, \dots, x_n)$$



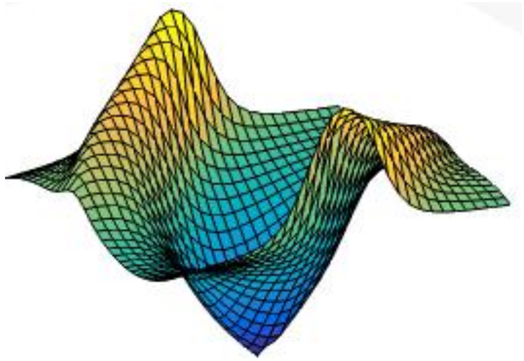


- Multiple Regression Genetic Programming [1]
- Evolutionary Feature Synthesis [2]
- Multi-Gene Genetic Programming [3]
- Single Node Genetic Programming [4, 5]

- [1] I. Arnaldo et al.: Multiple regression genetic programming (2014)
- [2] I. Arnaldo et al.: Building predictive models via feature synthesis (2015)
- [3] M. Hinchliffe et al.: Modelling chemical process systems using a multi-gene genetic programming algorithm (1996)
- [4] D. Jackson: Single node genetic programming on problems with side effects (2012)
- [5] J. Kubalík et al.: An improved Single Node Genetic Programming for symbolic regression (2015)

Example: value function approximator in value iteration

Symbolic V-function
from previous iteration



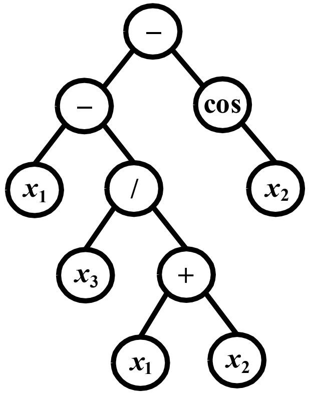
$$V_\ell(x) = 5 * x_2 - 3 * x_3 + \\ + \cos(x_1) - \sin(x_2) \dots$$

Target data

$$t_{i,\ell} = \max_j \big(r_{i,j} + \gamma V_{\ell-1}(x_{i,j})\big)$$

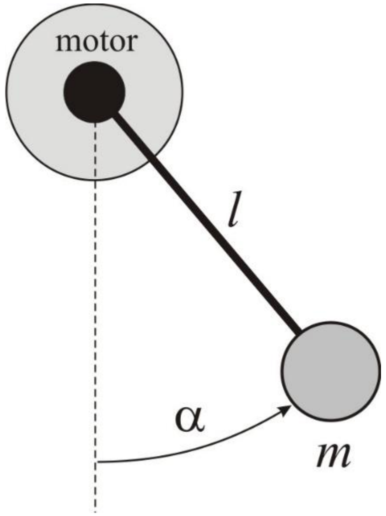
Symbolic regression

$$J_\ell^{\mathrm{SVI}} = \frac{1}{n_x} \sum_{i=1}^{n_x} \Big[\underbrace{t_{i,\ell}}_{\text{target}} - \underbrace{V_\ell(x_i)}_{\text{evolved}} \Big]^2$$



Pendulum swing-up

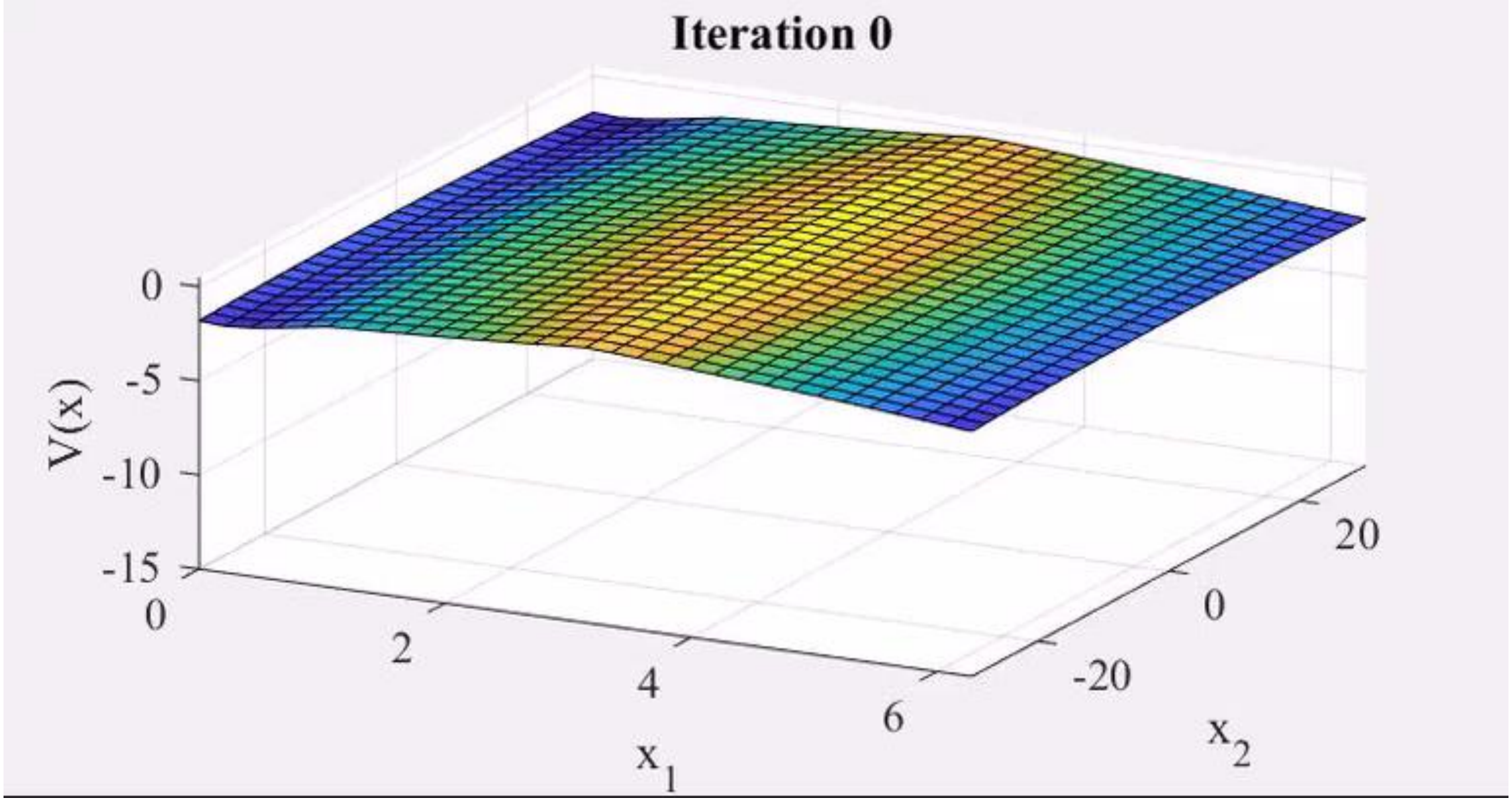
$$J\ddot{\alpha} \quad = \quad -mgl\sin(\alpha) - (b + \frac{K_t^2}{R})\dot{\alpha} + \frac{K_t}{R}u$$



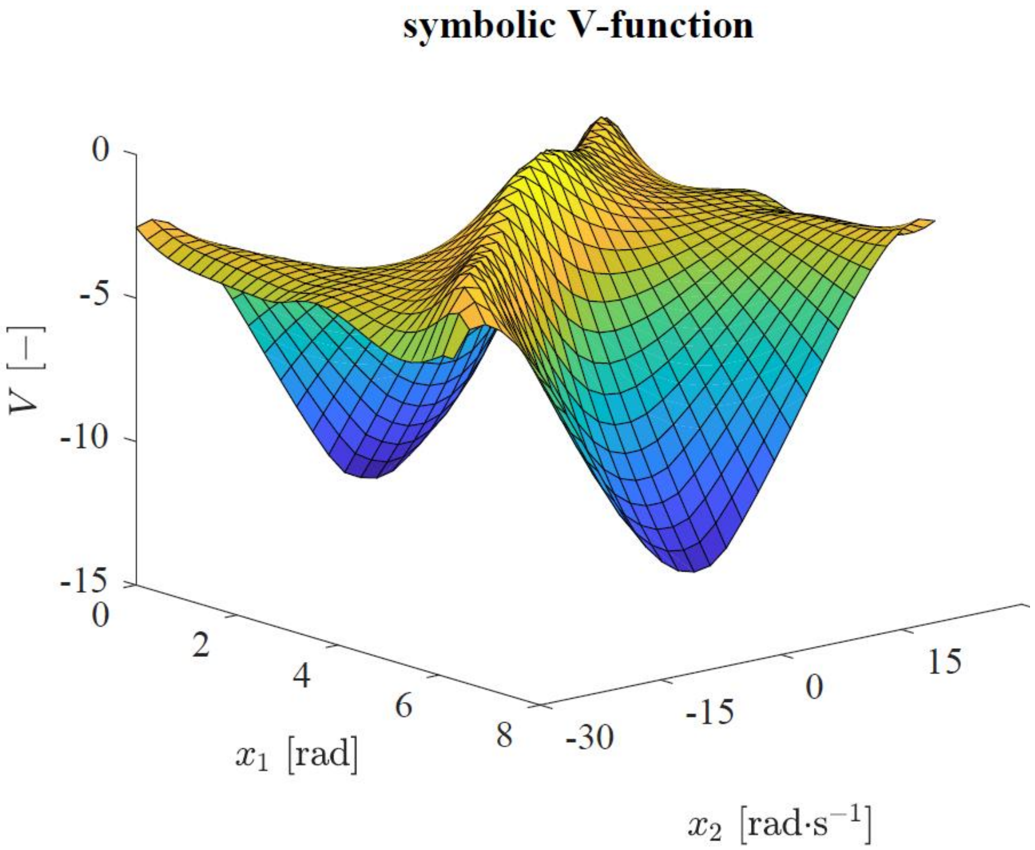
Control goal: bring mass to the upper equilibrium under the control action limited to:

$$|u_k| \leq 2\,[V], \quad \forall k$$

Value iteration



Value function: analytic expression

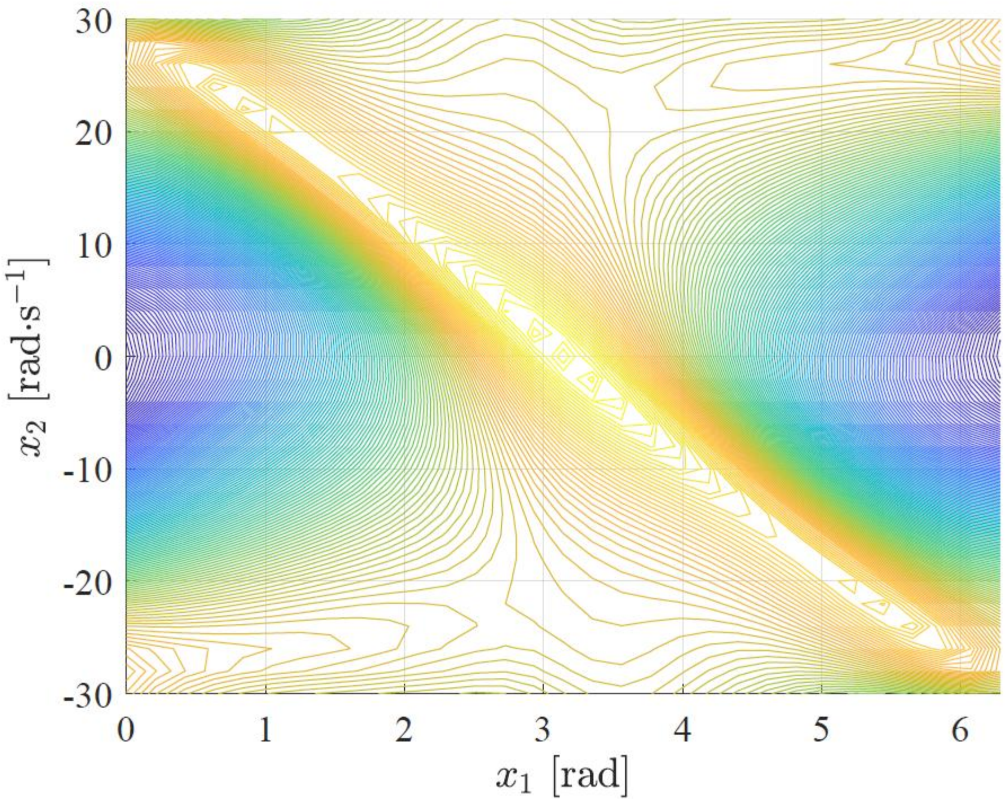


$$V(x) = 1.7\times10^{-5}(10x_2 - 12x_1 + 47)(4.3\times10^{-2}x_2 - 3.5x_1 + 11)^3 - 7.1\times10^{-4}x_2 - 4.6x_1 - 8.2\times10^{-6}(4.3\times10^{-2}x_2 - 3.5x_1 + 11)^3(0.2x_1 + 0.3x_2 - 0.5)^3 - 9.8\times10^{-3}(0.4x_1 + 0.1x_2 - 1.1)^6 + 11(0.1x_1 - 1.5)^3 + 11((0.6x_1 + 6.3\times10^{-2}x_2 - 1.7)^2 + 1)^{0.5} + 8.7\times10^{-6}((10x_2 - 12x_1 + 47)^2(4.3\times10^{-2}x_2 - 3.5x_1 + 11)^6 + 1)^{0.5} + 0.3((1.1x_1 + 0.4x_2 - 3.3)^2 + 1)^{0.5} + (3.9\times10^{-3}(4.3\times10^{-2}x_2 - 3.5x_1 + 11)^2(0.2x_1 + 0.3x_2 - 0.5)^2 + 1)^{0.5} + 6.5\times10^{-5}((1.2x_1 + 14x_2 - 10)^2(9.1\times10^{-2}x_2 - 2.9x_1 + 0.5((9.1\times10^{-2}x_2 - 2.9x_1 + 8.3)^2 + 1)^{0.5} + 7.8)^2 + 1)^{0.5} - 5.5\times10^{-2}(4.3\times10^{-2}x_2 - 3.5x_1 + 11)(0.2x_1 + 0.3x_2 - 0.5) - 1.7((3.6x_1 + 0.4x_2 - 11)^2 + 1)^{0.5} - 2((x_1 - 3.1)^2 + 1)^{0.5} - 1.3\times10^{-4}(1.2x_1 + 14x_2 - 10)(9.1\times10^{-2}x_2 - 2.9x_1 + 0.5((9.1\times10^{-2}x_2 - 2.9x_1 + 8.3)^2 + 1)^{0.5} + 7.8) + 23\,.$$

89 parameters

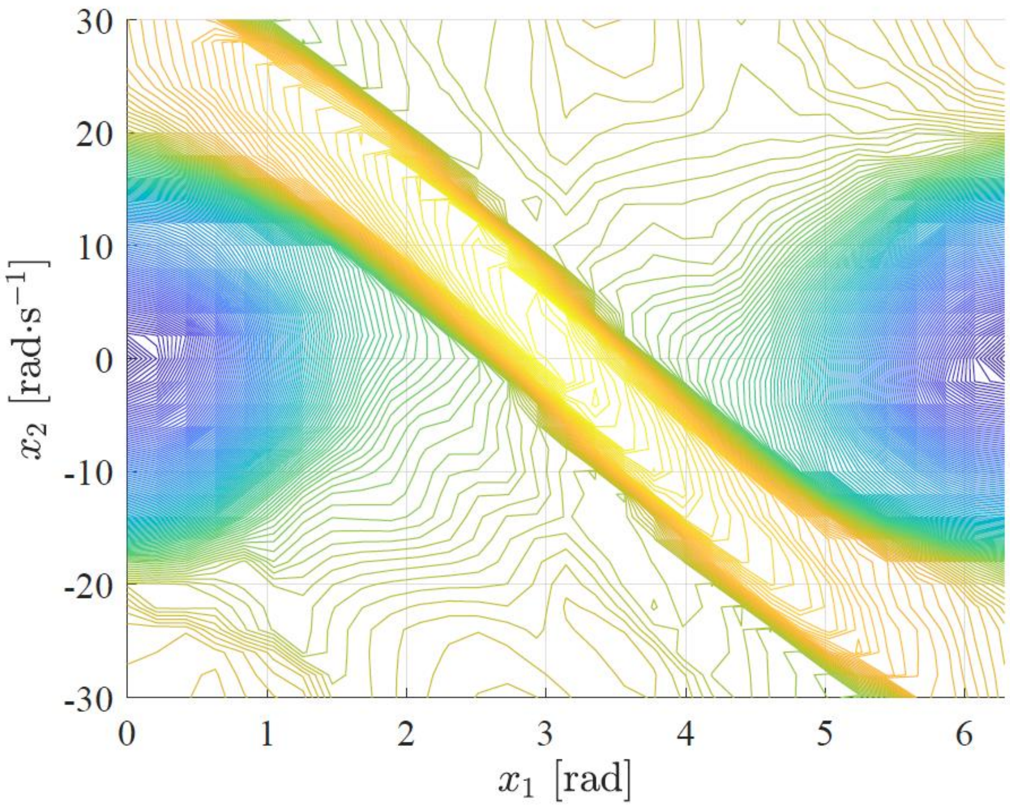
Comparison with a neural network

Symbolic V-function



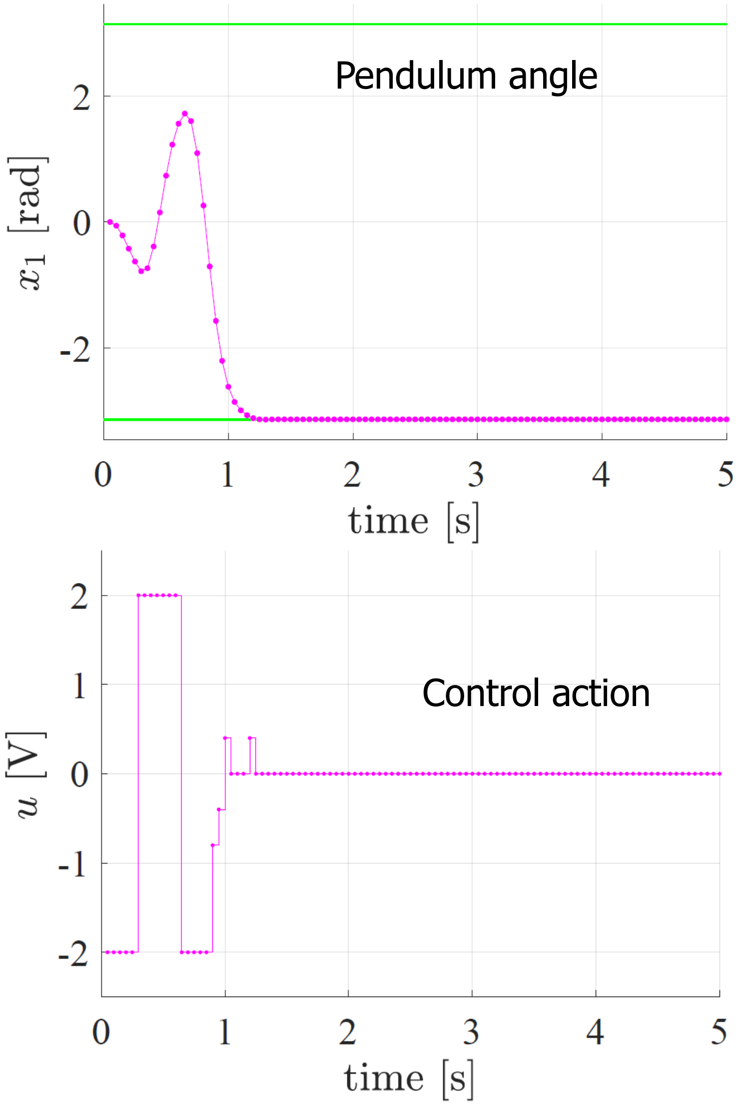
89 parameters

Neural network V-function



201 parameters

Swing-up experiment on the real system



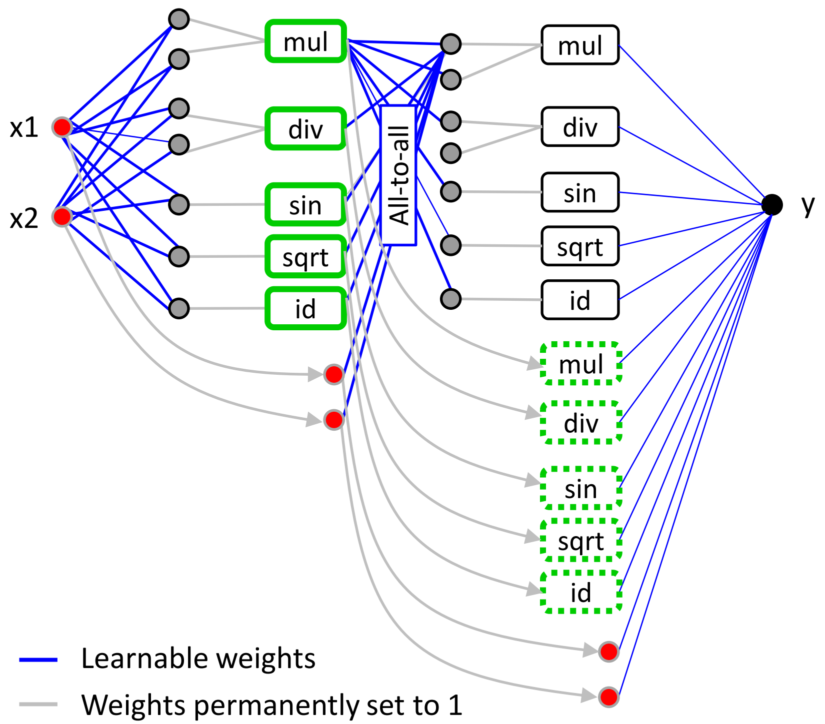
Pro’s and con’s of genetic programming

+ Straightforward approach, few hyperparameters
+ Global exploration of search space
+ Can build compact models for a broad range of systems, incl. hybrid systems
+ Prior knowledge in terms of expected mathematical functions

- Tuning real-valued parameters
- Computationally demanding

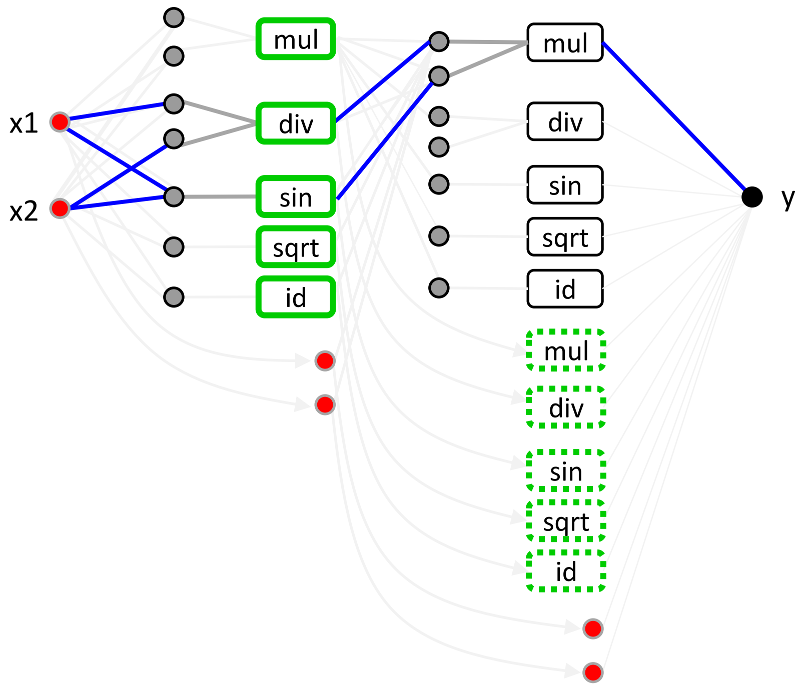
Neural networks for symbolic regression

Initial network



Backpropagation
&
regularization

Final model: $y = \sin(0.3 * \ x_1 - 1.2 * x_2) * \frac{x_1}{2*x_2}$



Kubalik et al. Toward Physically Plausible Data-Driven Models: A Novel Neural Network Approach to Symbolic Regression, IEEE 2023

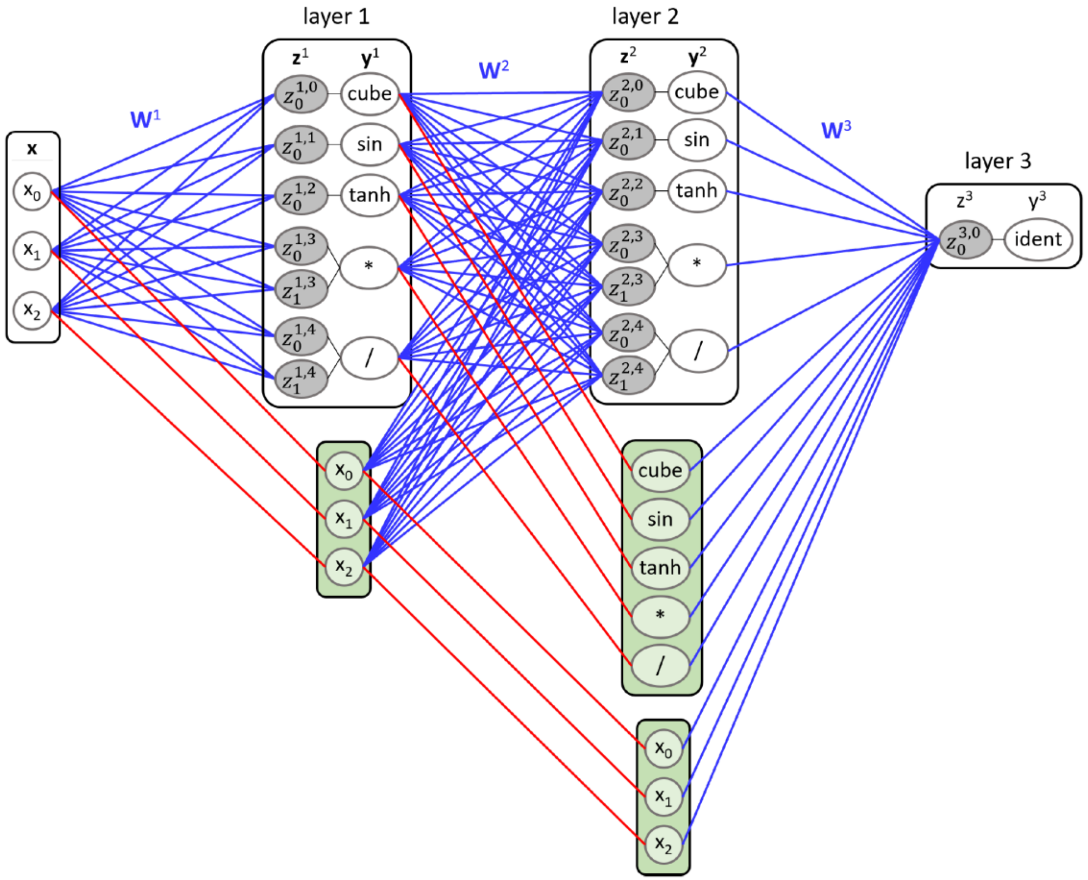
Pro’s and con’s of neural networks

+ Prior knowledge in terms of expected mathematical functions
+ Tuning real-valued parameters
+ Regularization for accuracy – complexity tradeoff

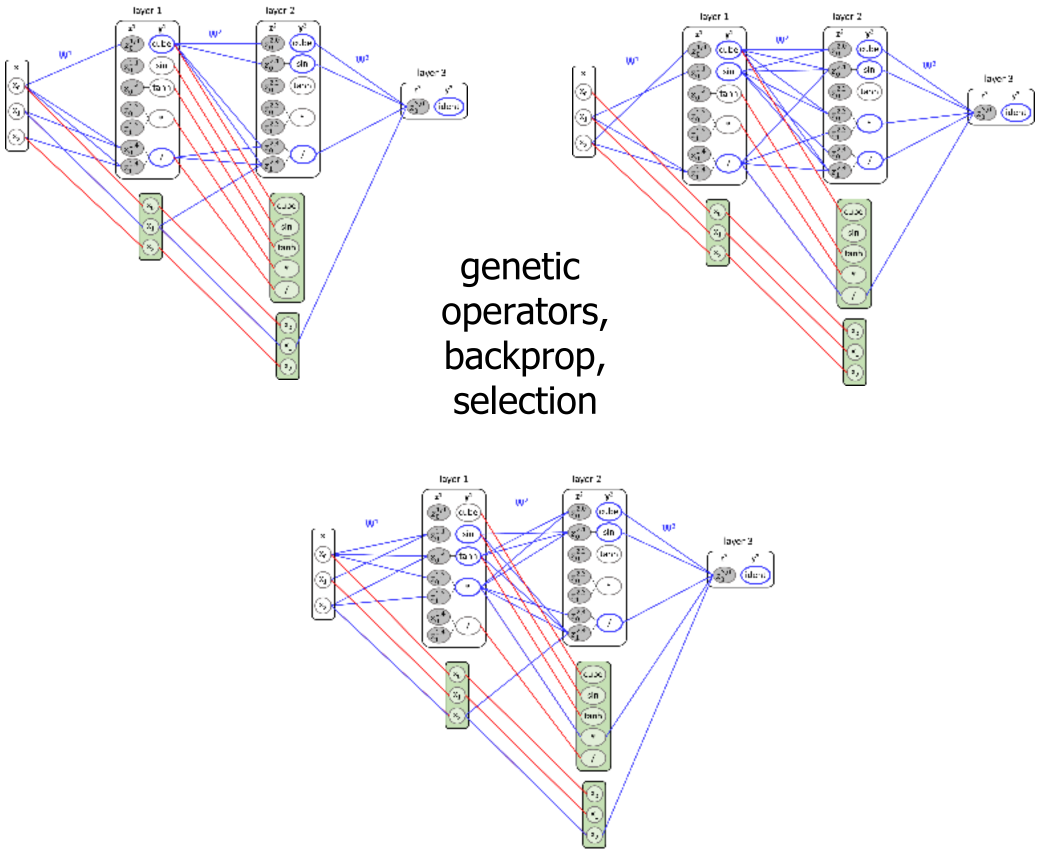
- Gradient-based method – local convergence, sensitive to learning rate
- More hyperparameters than genetic programming methods
- Needs differentiable functions

Neuro-Evolutionary Approach to Symbolic Regression

Master topology



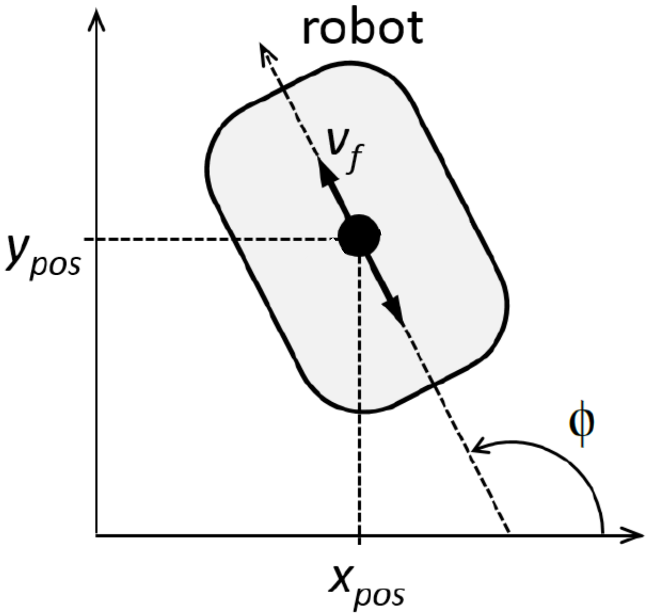
Population of subtopologies



Mechanistic models vs. models learned from data

- Mechanistic models correctly represent the physics but are inaccurate as prediction models.
- Data-driven models are accurate but typically do not respect physical constraints.

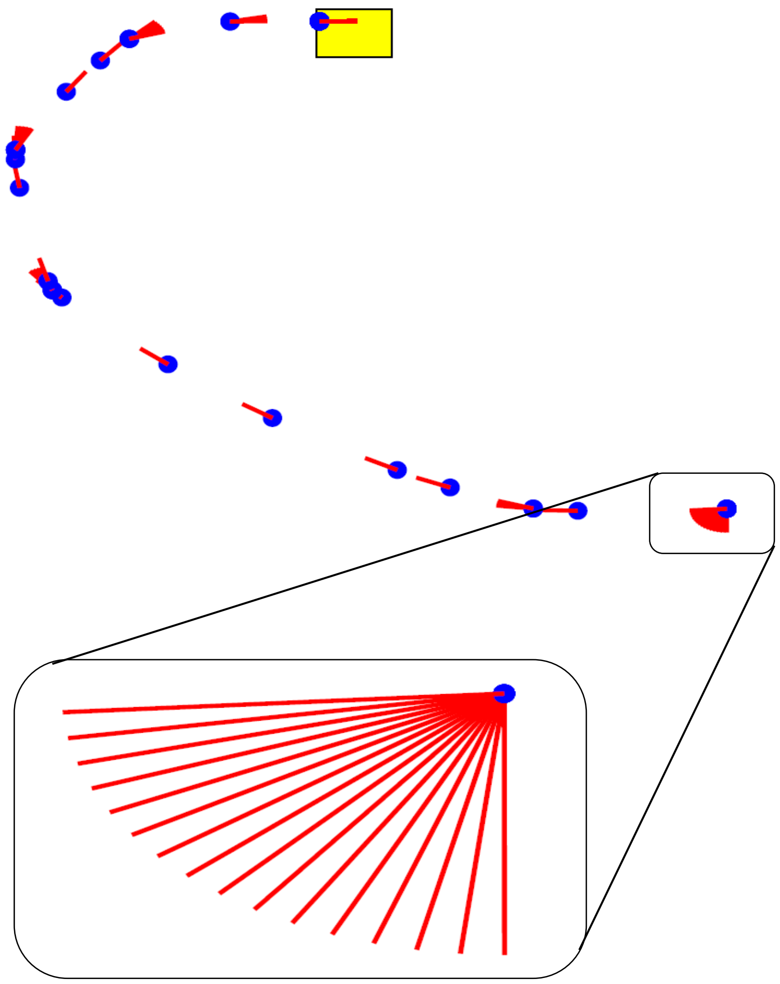
Example: mobile robot

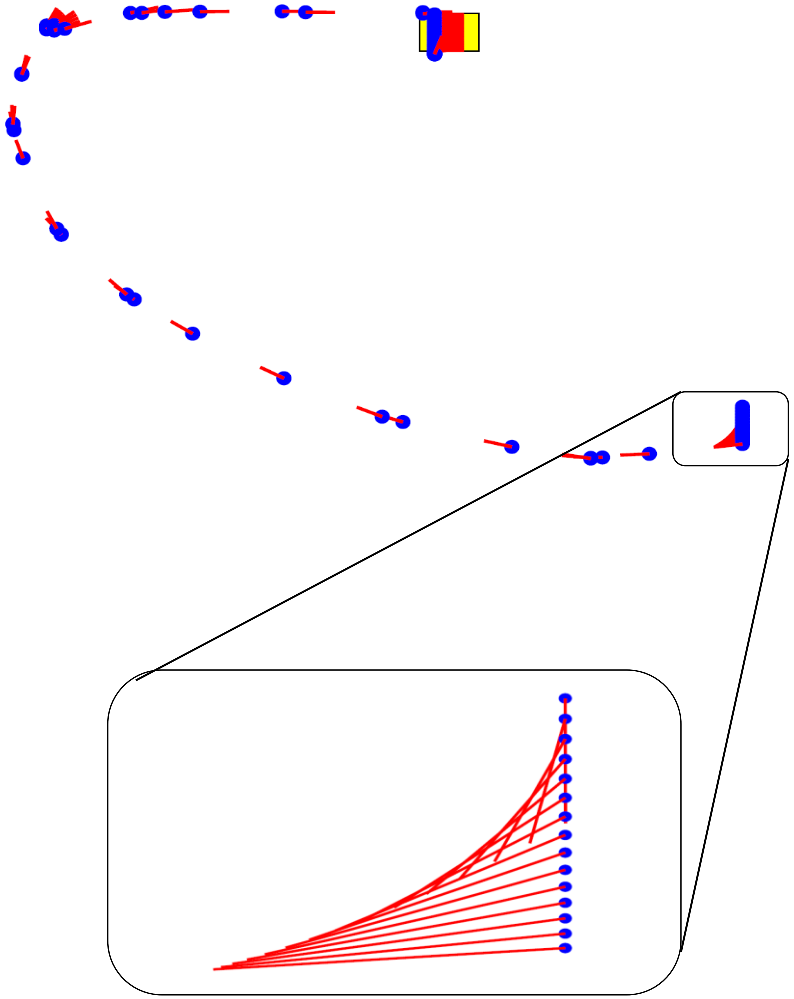


Mechanistic model:

$$\dot{x}_{pos} = v_f \cos(\phi)$$
$$\dot{y}_{pos} = v_f \sin(\phi)$$
$$\dot{\phi} = v_a$$

Motion planning: mechanistic model vs data-driven model





Include prior knowledge in learning

Generate synthetic data samples representing physical constraints, use multi-objective optimization.

Examples of constraints:

- Equilibrium under zero input

$$x_0 = f(x_0, 0)$$

- Non-holonomic constraint (robot cannot move sideways)

$$y_{pos} = f_y([x_{pos}, y_{pos}, \phi]^T, [v_f, 0])$$

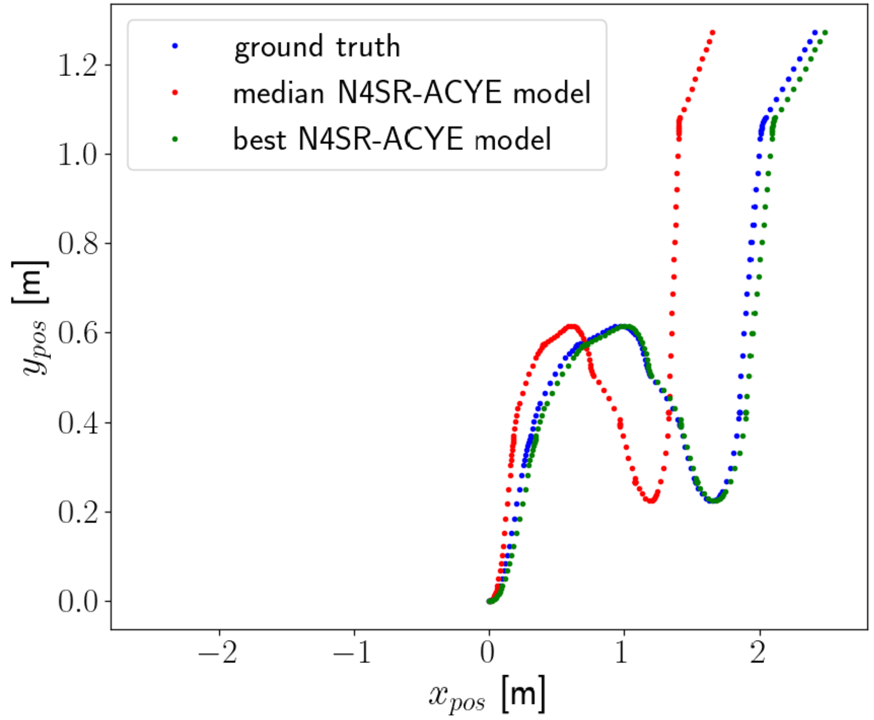
Mobile robot: neural network SR approach

Model structure assumed:

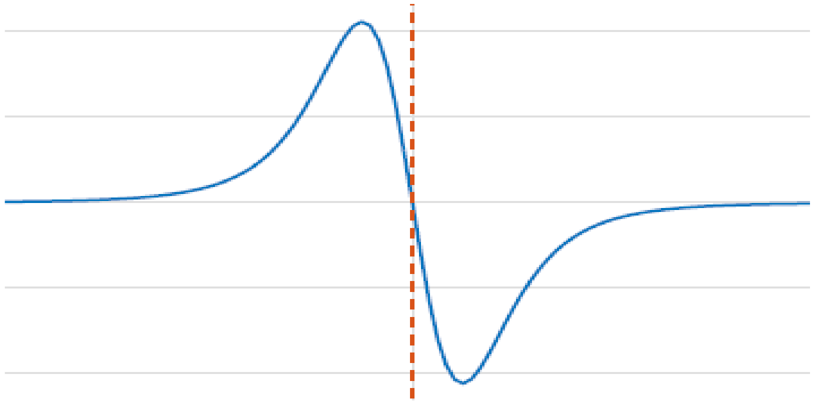
$$\hat{x}_{pos,k+1} = f^{x_{pos}}(\hat{x}_{pos,k}, y_{pos,k}, \phi_k, v_{f,k}, v_{a,k})$$

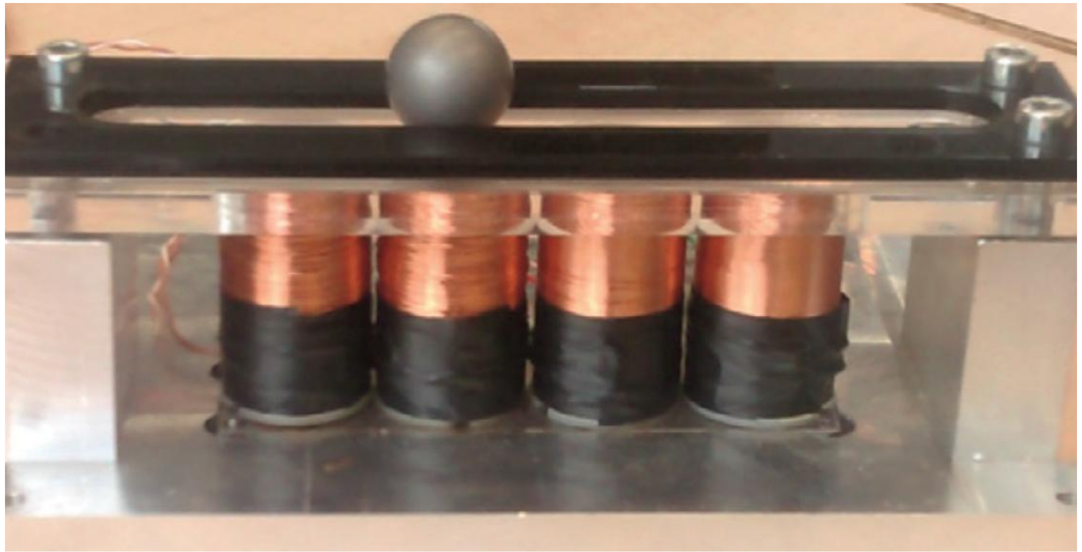
Best model:

$$x_{pos,k+1} = 0.1693\, v_{f,k} \sin(0.9838\, \phi_k + 1.5337) + 0.999986\, x_{pos,k}.$$



Magnetic manipulation



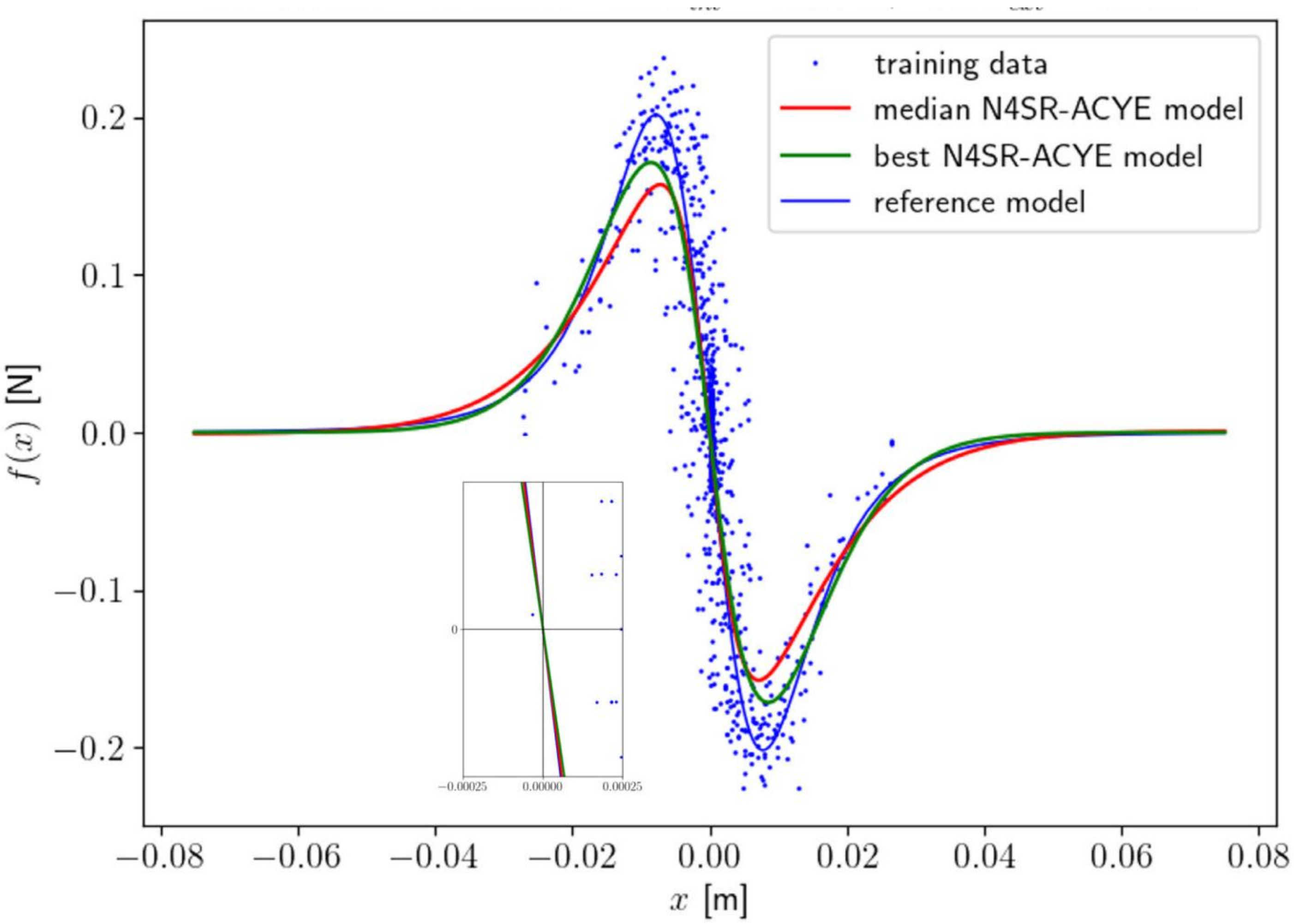


Empirical model:

$$F(x, I) = g(x)\, I = \frac{-\alpha\, x}{(x^2 + \beta)^3}\, I$$

Prior knowledge:
- g(x) = 0 for x = 0
- g(x) is monotonically decreasing around zero
- g(x) goes monotonically to zero for x large

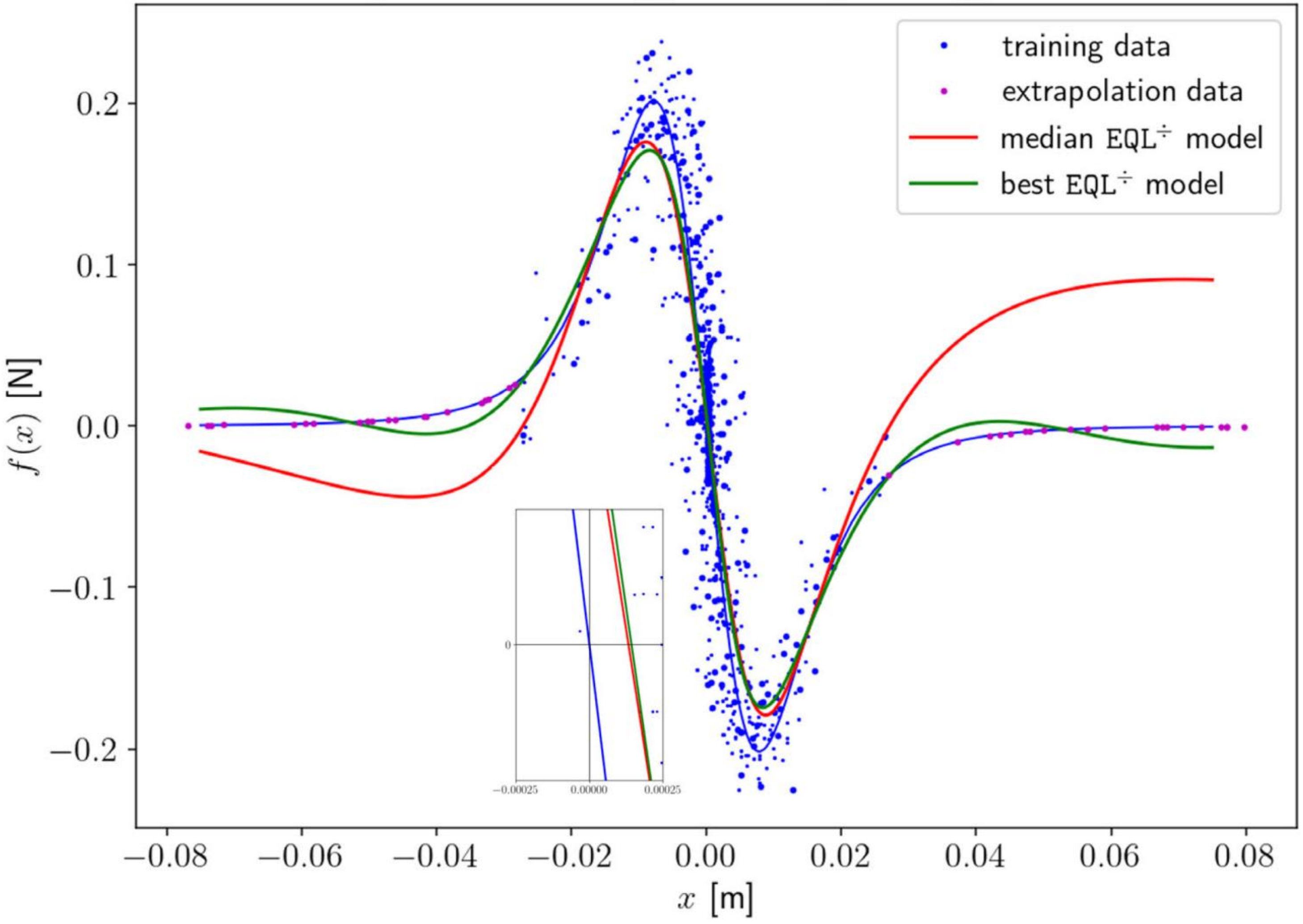
Magnetic manipulation: results with prior knowledge



Best model:

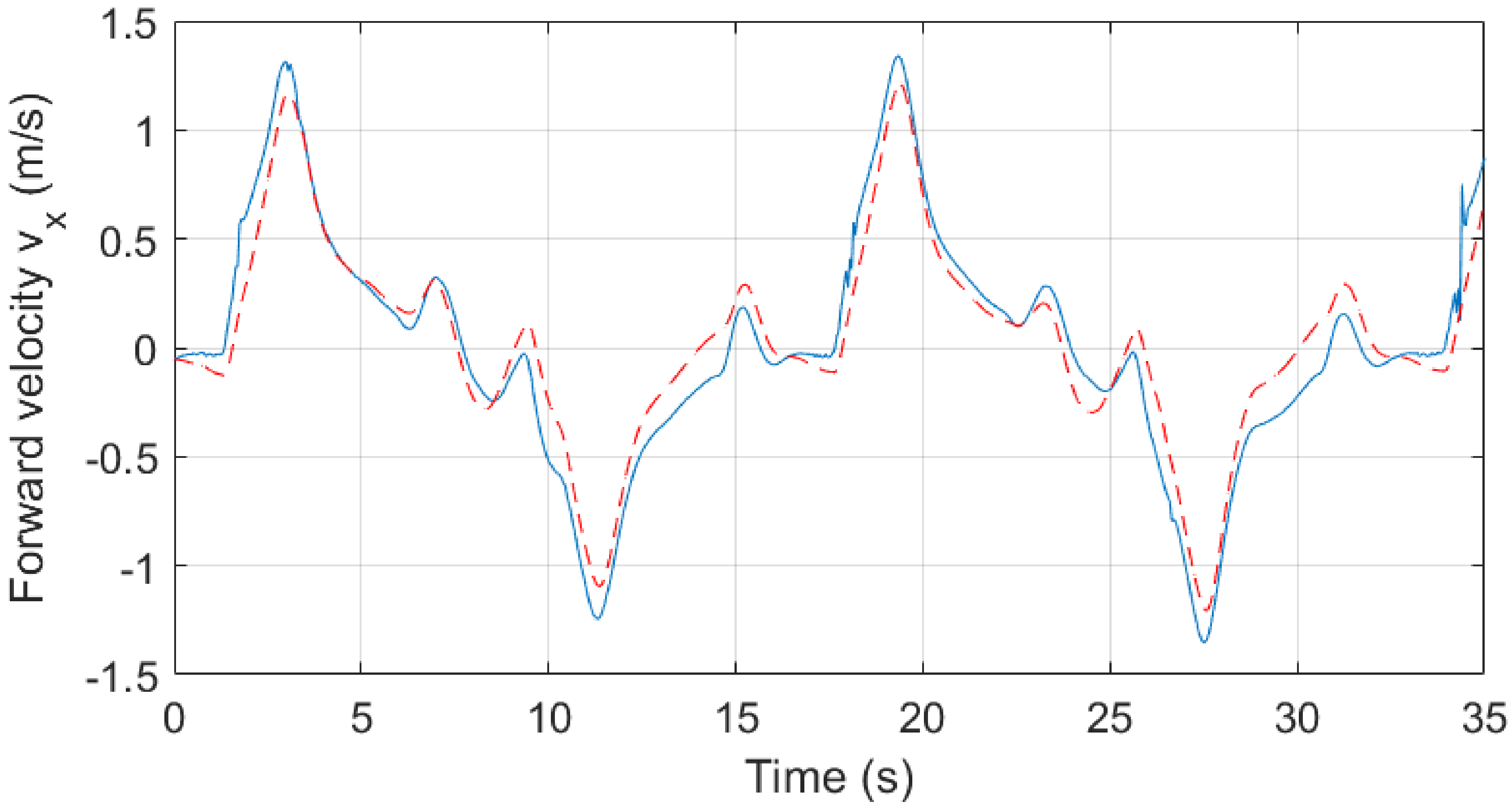
$$
\begin{aligned}
f(x) = &-0.429 \sin(1.971 \tanh(4.294\,x) \\
&+ 5.33 \tanh(8.494 \tanh(4.294\,x))) \\
&- 2.027 \tanh(4.701 \tanh(8.494 \tanh(4.294\,x))) \\
&+ 1.607 \tanh(8.494 \tanh(4.294\,x)) \\
&+ 0.873 \tanh(4.294\,x)
\end{aligned}
$$

Magnetic manipulation: results without prior knowledge

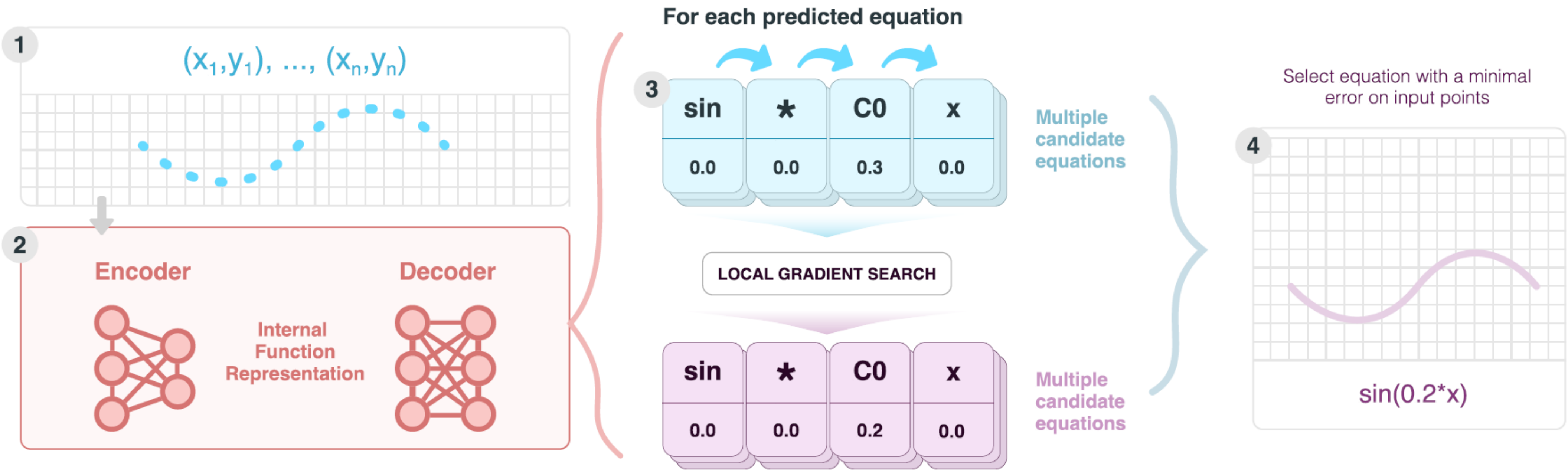


Quadcopter forward velocity model

Typical result: $v_x(k+1) = 0.985\, v_x(k) + 0.473\, \theta(k)$



Transformer-based symbolic regression

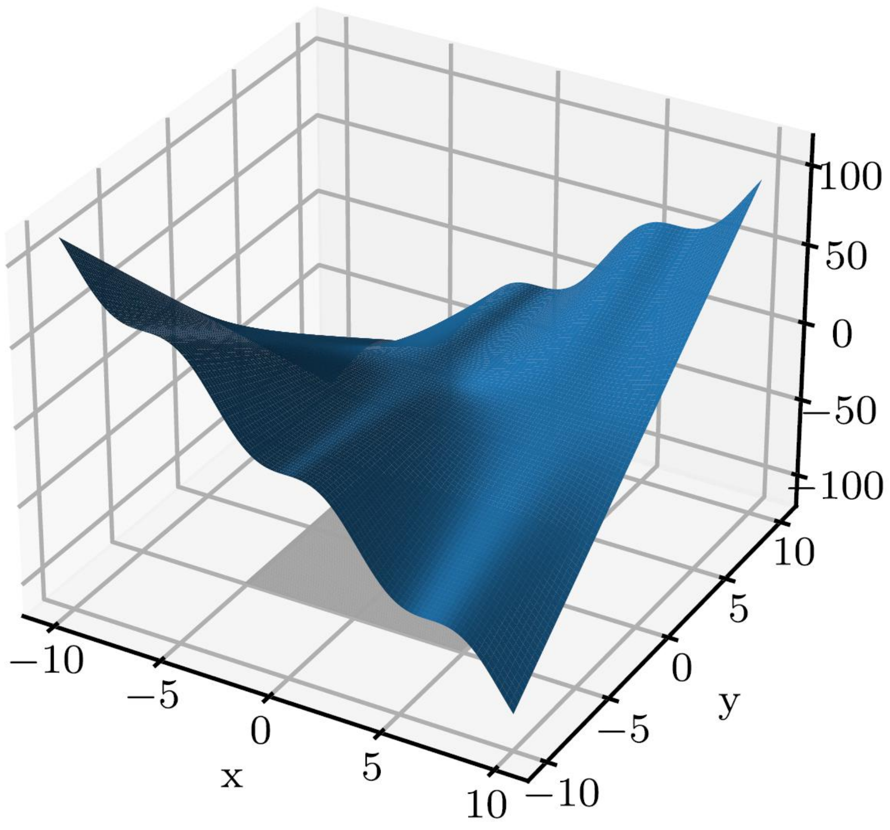


Trained on 130 million univariate functions and 100 million bivariate functions (33 hours on A100 GPUs).

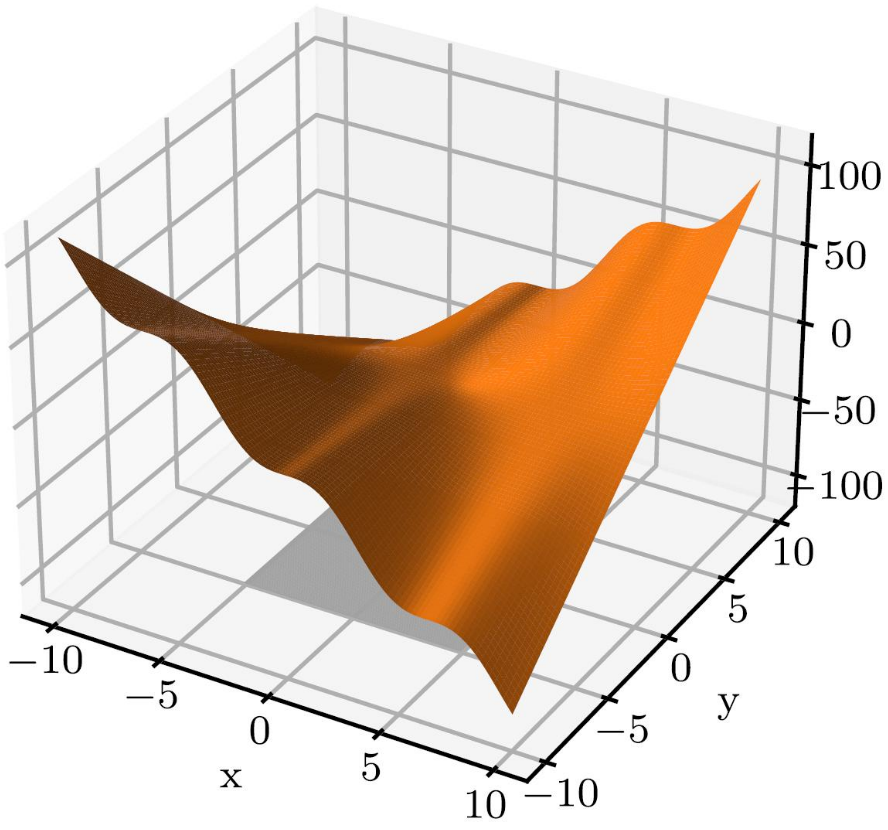
Vastl et al. SymFormer: End-to-end symbolic regression using transformer-based architecture. IEEE, 2024.

Examples of functions

(a) GT: $5x + x^2 - x\cos(x^2 + xy)$,



Pred: $5x + x^2 + x\cos(x^2 + xy + 3.14)$



Pro’s and con’s of transformer-based symbolic regression

+ Shifts computational burden to transformer training (hours to days)
+ Actual search for regression model is relatively fast (about a minute)
+ Extensions to learning differential equations (ODE former)

- Does not scale well to larger problems
- Many hyperparameters to tune

Conclusions

Symbolic regression is a promising ML method that can

- construct models from small datasets
- include prior knowledge: expected functions, physical constraints
- low-complexity, transparent models (compared to neural networks)

Challenges

- computationally expensive
- dynamic models as differential equations
- time-varying problems (online learning, adaptation)
- standard tools and codebases

Acknowledgements

Collaborators:

Jiri Kubalik, Erik Derner, Martin Vastl, Jonas Kulhanek, Hai Zhu, Luuk van den Bent, Tomas Coleman, Jens Kober, Bas van der Heijden, Yudha Pane, Erik Schuitema, Pieter Jonker, Martijn Wisse.

Research on symbolic regression received funding from

- European Union, reg. no. CZ.02.01.01/00/22_008/0004590, ROBOPROX,
- European Union, Horizon Europe, grant agreement No. 101070254 CORESENSE,
- European Regional Development Fund, grant no. CZ.02.1.01/0.0/0.0/15_003/0000470, Robotics for Industry 4.0.
- Grant Agency of the Czech Republic, grant no. 15-22731S, Symbolic Regression for Reinforcement Learning in Continuous Spaces.