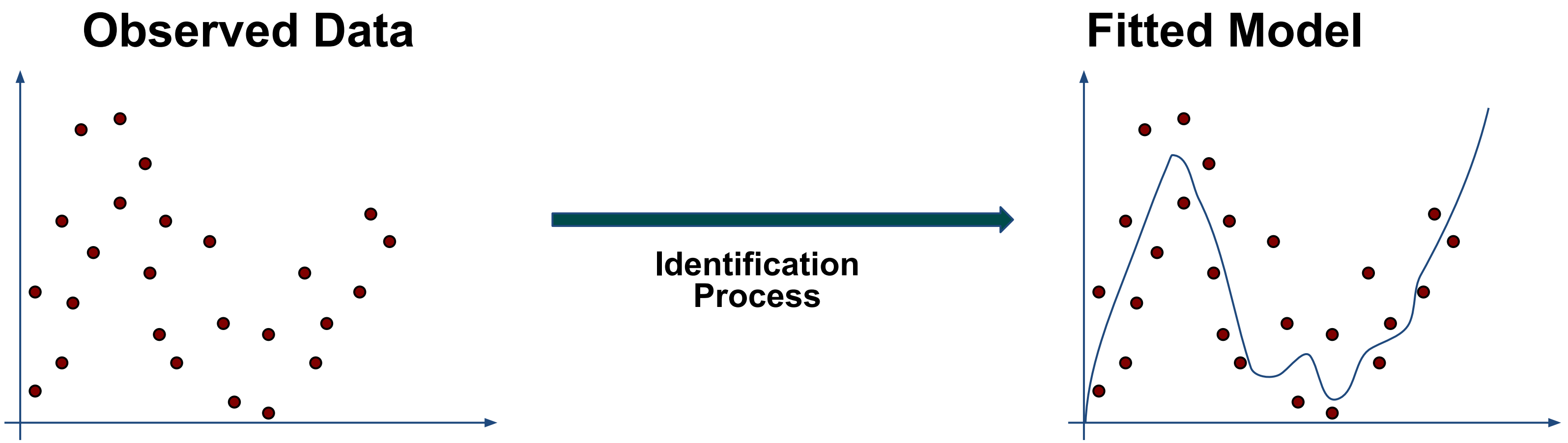
Physical verification of Neural Network Models

Pesaresi Seminar - 2023

Josafat Ribeiro Leal Filho

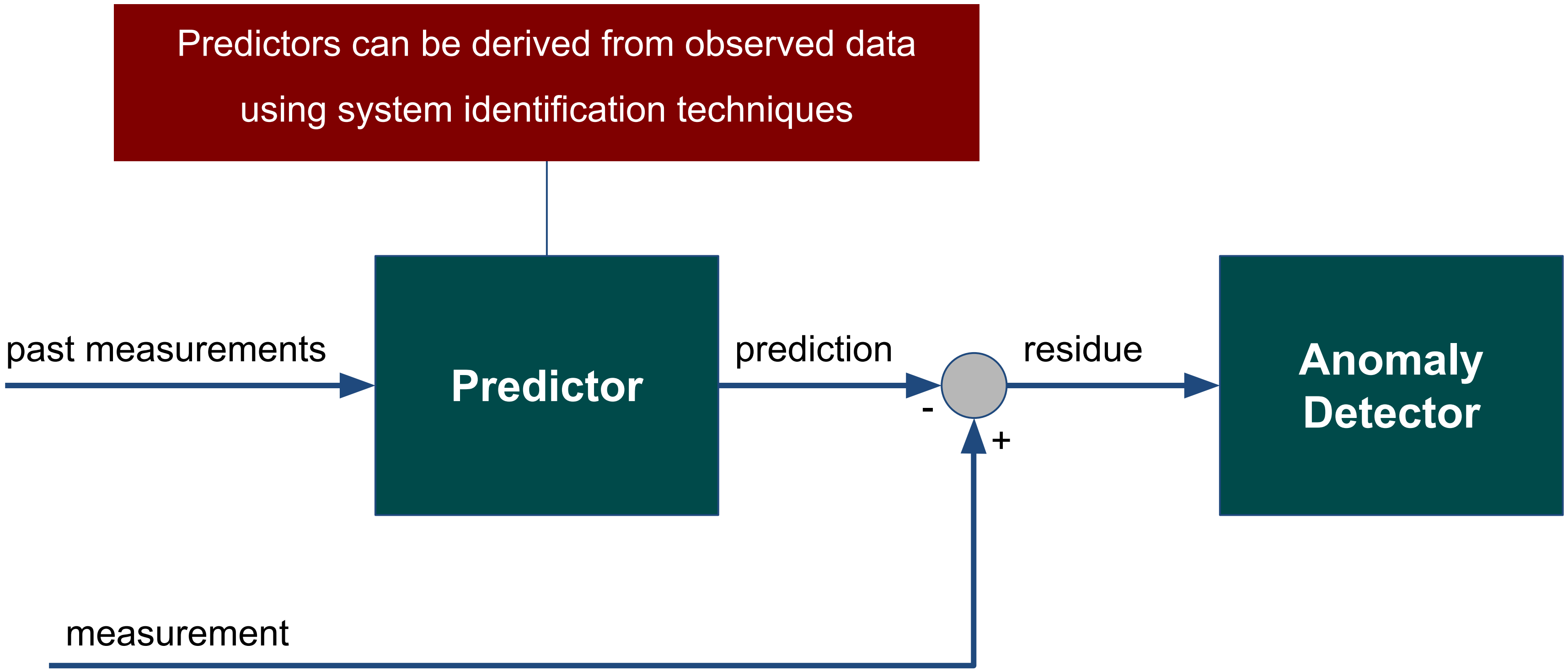
Supervisors:
Antônio Augusto Fröhlich
UFSC
Stefano Chessa
UNIPI

Dynamic System Identification



- In recent years, Artificial Neural Networks (ANN) have gained significant attention as model candidates in system identification.
- **However, the effectiveness of ANNs in this domain depends on factors like training data quality and interpretability.**

Example : Predictor Based Monitoring and Anomaly Detection

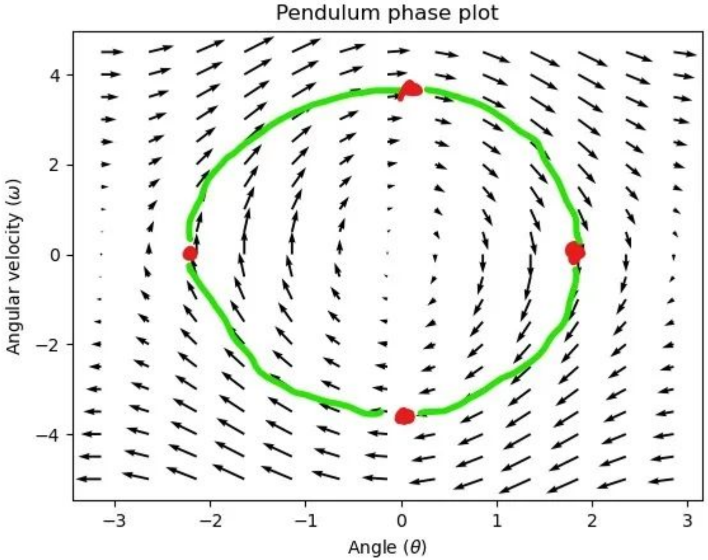


Governing equations

The governing equations in dynamical systems can possess physical and mathematical structures in the form of symmetries, symplecticity ("respects" the structure of the state-space) and energy conservation.

Governing equation
of a Ideal Pendulum

$$\frac{d^2\theta}{dt^2} + \frac{g}{\ell}\sin\theta = 0$$



Model-Based x Data Driven, Hybrid Approaches

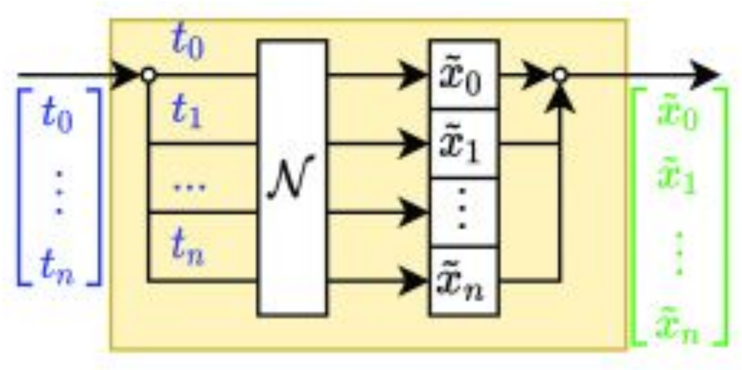
Model-Based	Data-Driven
- Model structure is derived from governing equations and parameters are determined based on observed data - Parameters are usually interpretable	- Model structure is selected based on data analysis and heuristics - Parameters are usually not tied to any interpretable quantity

- Hybrid approaches combine first principles based governing equations models with neural networks (Integration of domain knowledge).

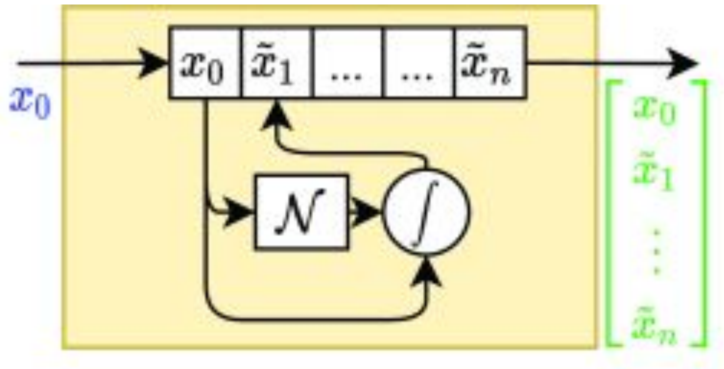
“All models are wrong, but some are useful”. George E. P. Box

Model-Based x Data Driven, Hybrid Approaches

Direct-solution model

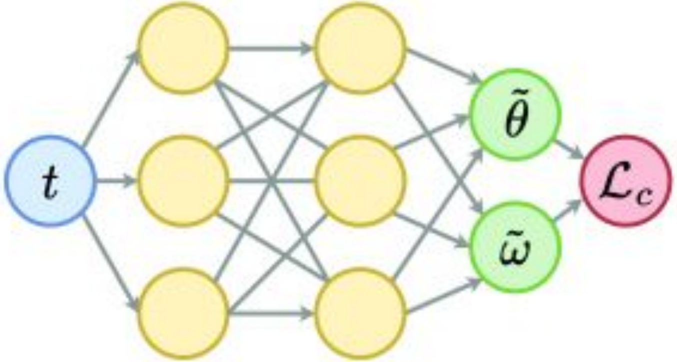


Time-stepper model

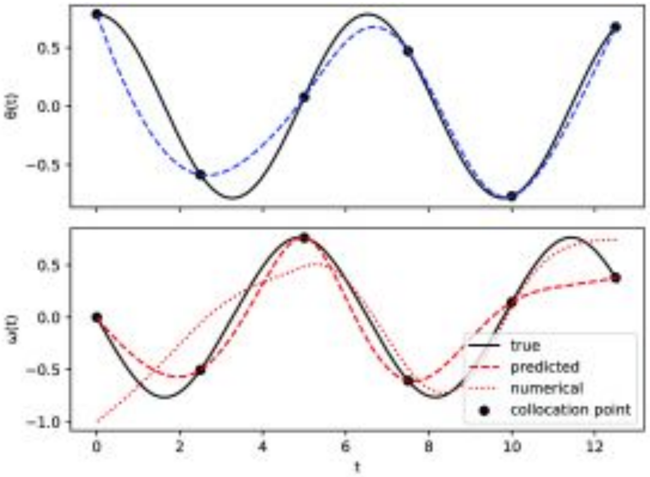


Introduction : Direct-Solution Models

Vanilla direct-solution



(a) Network structure.



(b) Predictions.

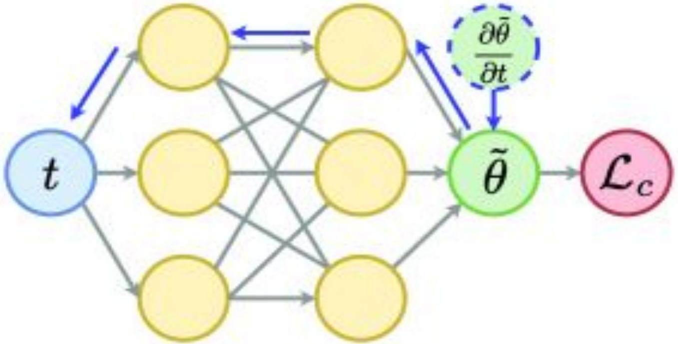
```
θ̃,ω̃ = 𝒩(t)
loss = ℒc((θ̃, ω̃), (θ, ω))
optimizer.step(loss)
```

(c) Training.

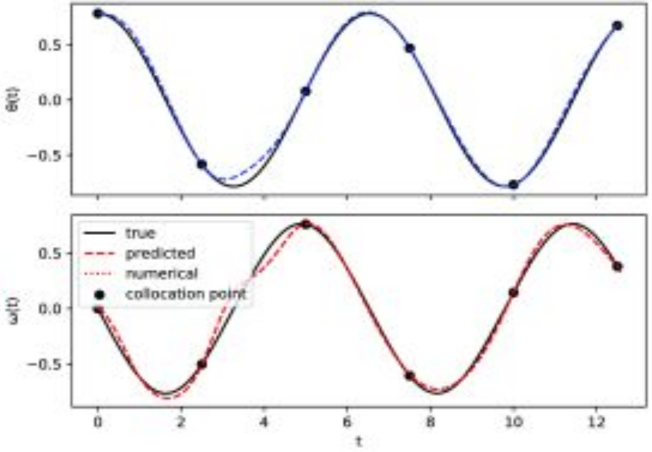
```
θ̃,ω̃ = 𝒩(t)
```

(d) Inference.

Automatic differentiation solution



(a) Network structure.



(b) Predictions

```
θ̃ = 𝒩(t)
ω̃ = gradient(θ̃, t)
loss = ℒc((θ̃, ω̃), (θ, ω))
optimizer.step(loss)
```

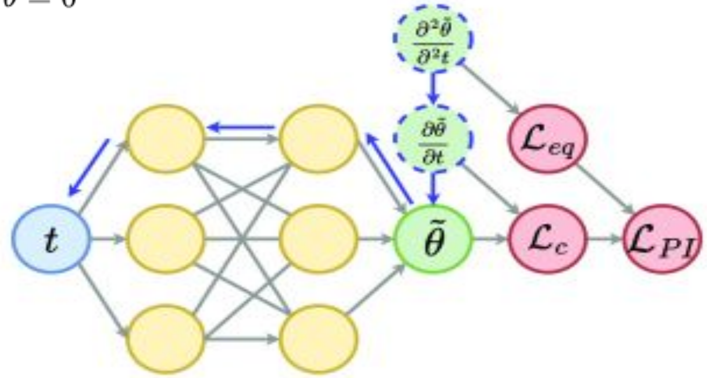
(c) Training.

```
θ̃ = 𝒩(t)
ω̃ = gradient(θ̃, t)
```

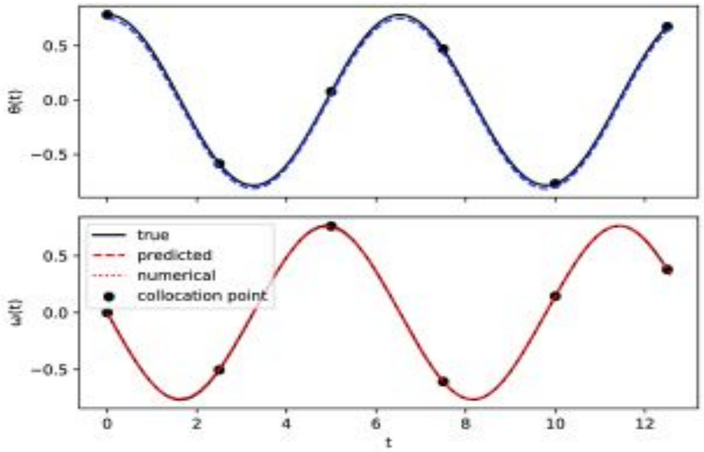
(d) Inference.

Physics-informed neural network

$$\frac{d^2\theta}{dt^2} + \frac{g}{\ell}\sin\theta = 0$$



(a) Network structure.



(b) Predictions.

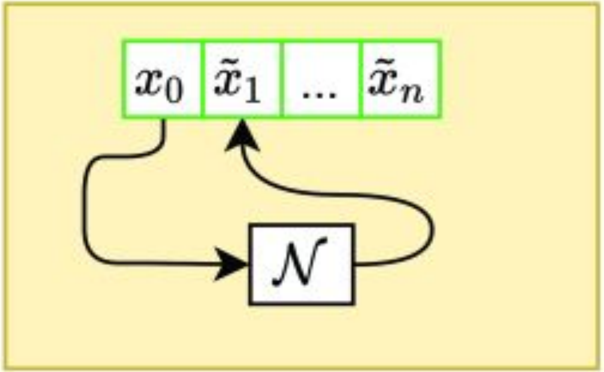
```
θ̃ = N(t)
ω̃ = gradient(θ̃, t)
∂ω̃ = gradient(ω̃, t)
loss = L_c((θ̃,ω̃),(θ,ω))+L_eq(θ̃,∂ω̃)
optimizer.step(loss)
```

(c) Training.

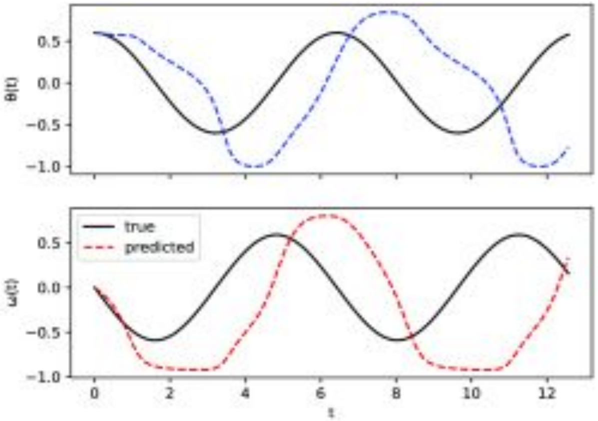
```
θ̃ = N(t)
ω̃ = gradient(θ̃, t)
```

(d) Inference.

Direct time-stepper



(a) Network structure.



(b) Predictions.

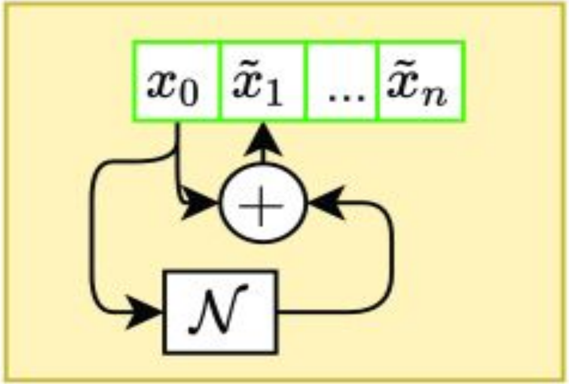
```
X̃[:,0,:] = X_{t_0}
for i in 0...m-1
  X̃[:,i+1,:] = 𝒩(X̃[:,i,:])
loss = ℒ_{ts}(X,X̃)
optimizer.step(loss)
```

(c) Training.

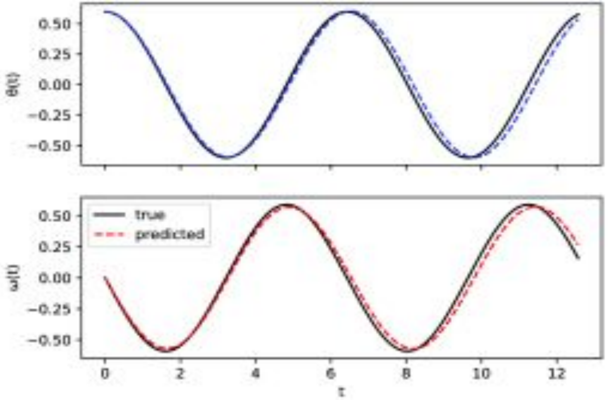
```
x̃[0,:] = x_{t_0}
for i in 0...m-1
  x̃[i+1,:] = 𝒩(x̃[i,:])
```

(d) Inference.

Residual time-stepper



(a) Network structure.



(b) Predictions.

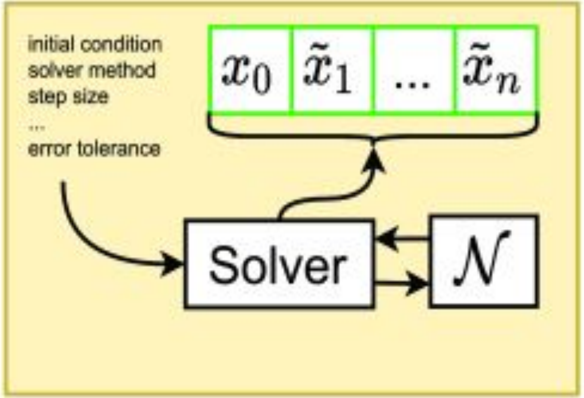
```
X̃[:,0,:] = X₀
for i in 0...m-1
  ΔX = 𝒩(X̃[:,i,:])
  X̃[:,i+1,:] = X̃[:,i,:] + ΔX
loss = ℒts(X,X̃)
optimizer.step(loss)
```

(c) Training.

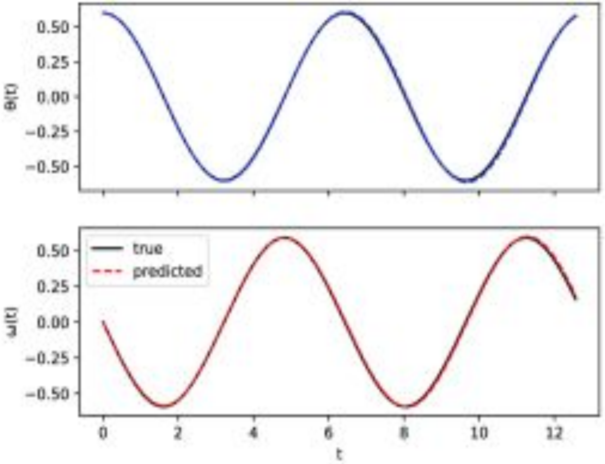
```
x̃[0,:] = x₀
for i in 0...m-1
  Δx = 𝒩(x̃[i,:])
  x̃[i+1,:] = x̃[i,:] + Δx
```

(d) Inference.

Neural Ordinary differential equations (NODEs)



(a) Network Structure.



(b) Predictions.

$\tilde{X}$ = odeint($\mathcal{N}$, X_0, t_{start}, t_{end}, "rk4")
loss = $\mathcal{L}_{ts}(X, \tilde{X})$
optimizer.step(loss)

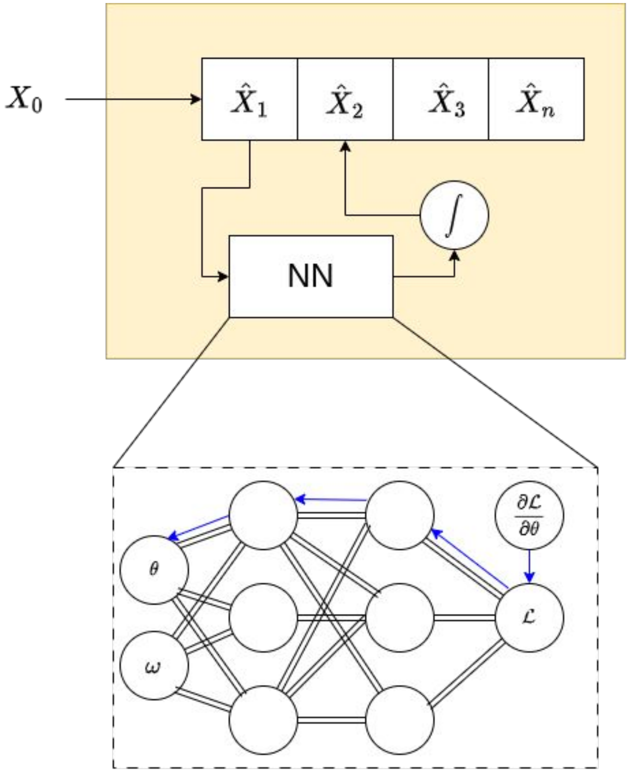
(c) Training.

$\tilde{x}$ = odeint($\mathcal{N}$, x_0, t_{start}, t_{end}, "rk4")

(d) Inference.

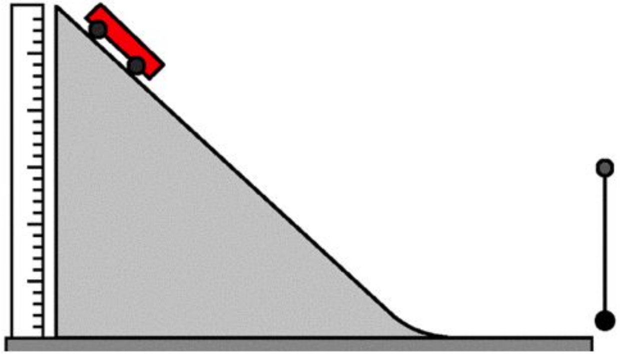
Introduction : Time-stepper models

Hamiltonian and Lagrangian Networks



$$L = T - V$$

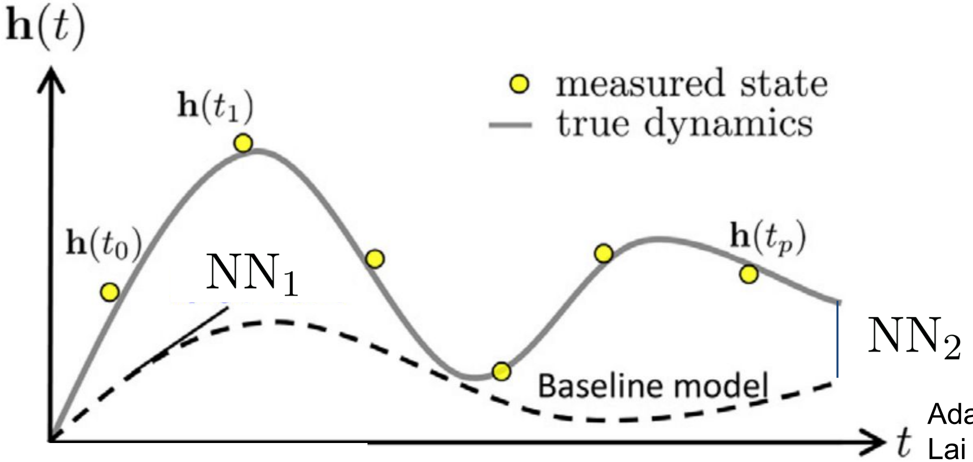
where T and V are the kinetic and potential energy of the system, respectively.



Hybrids approach

Real data application

- $\mathbf{h}(t)$ comprises the dynamic system representation.
- The evolution equation: $\dot{\mathbf{h}}(t) = A\mathbf{h}(t) + \mathbf{w}(t)$
- Neural networks approximate the system evolution.
- Discrete-time state evolution: $\mathbf{h}(t_{k+1}) = \mathbf{h}(t_k) + \mathrm{NN}_1(\mathbf{h}(t_k)) + \mathrm{NN}_2(\mathbf{h}(t_k))$

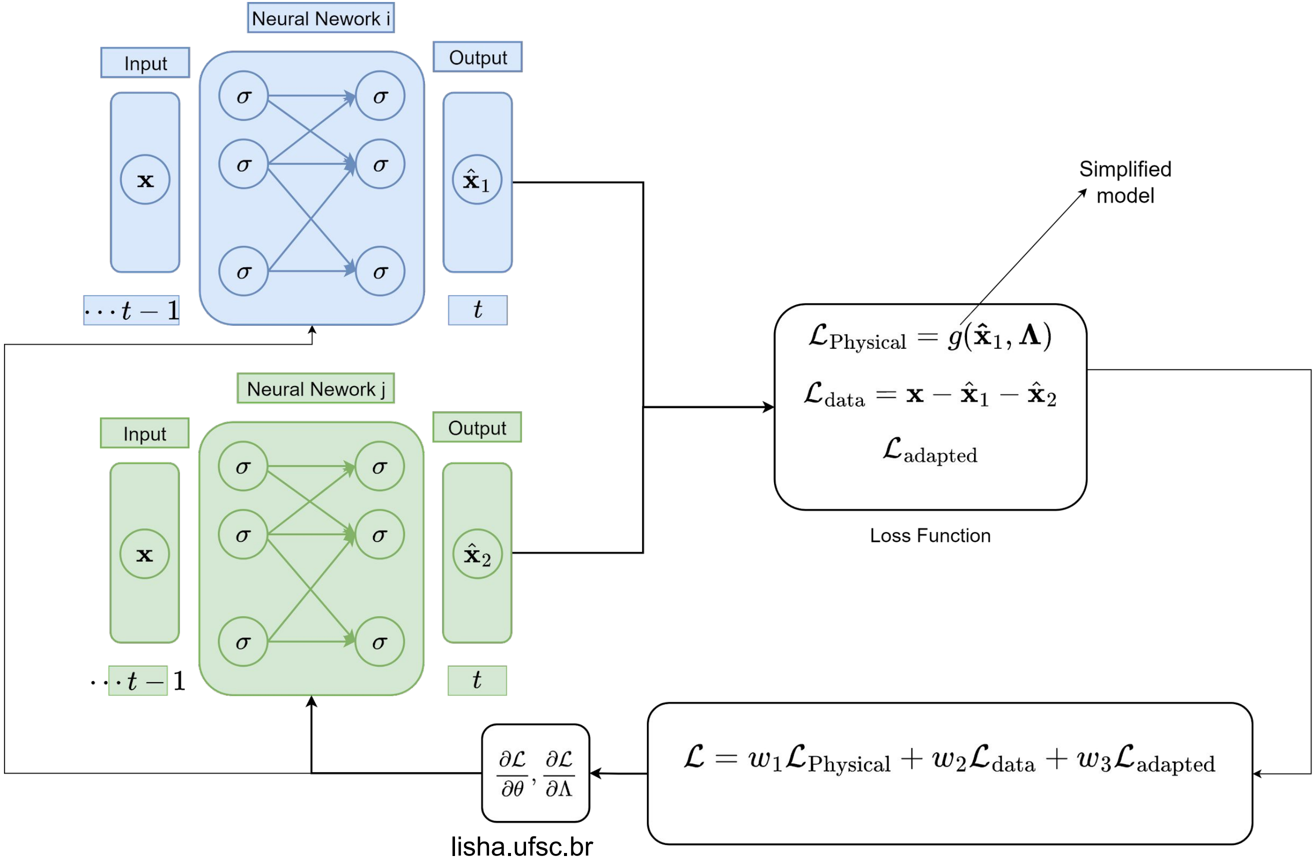


Adapted from Lai et al. 2021

Prior knowledge about the dynamic system

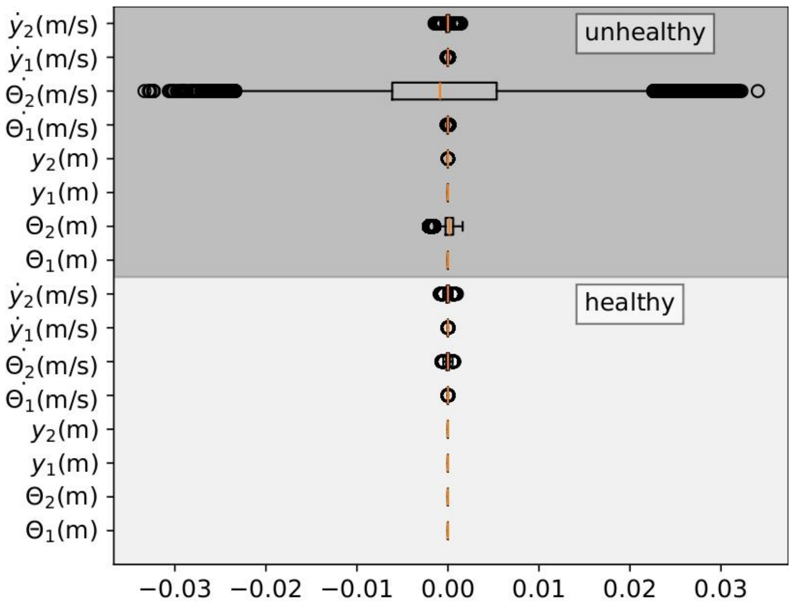
Hybrids approach

Real data application



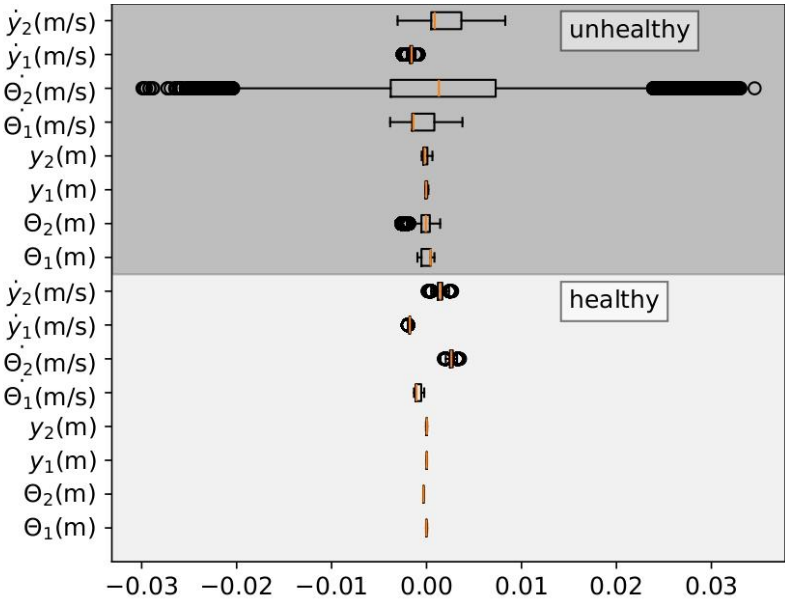
Real data application

Physics-informed Neural Network



MSE = 4.87007e-05

Vanilla Neural Network



MSE = 6.51007e-03

How hybrid approaches are validated?

Analytical recovered expression

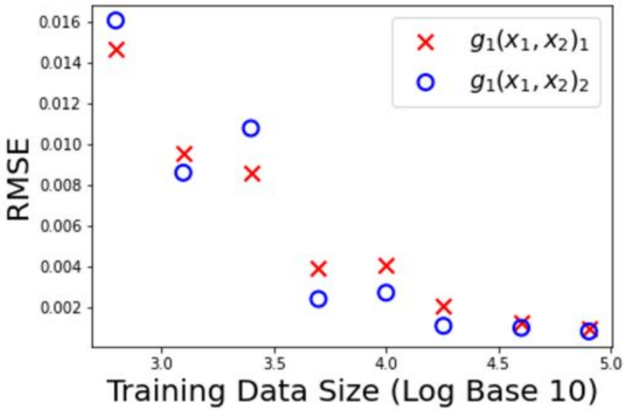
		True discrepancy model	Recovered discrepancy model
scheme 2	$\ddot{x}_1$	$-6x_1 + 3x_2 - 0.15\dot{x}_1 - 2x_1^3$	$-6.002x_1 + 2.998x_2 - 0.149\dot{x}_1 - 1.994x_1^3$
	$\ddot{x}_2$	$3x_1 - 6x_2 + 3x_3 - 0.15\dot{x}_2$	$3.008x_1 - 6.000x_2 + 2.999x_3 - 0.151\dot{x}_2$
	$\ddot{x}_3$	$3x_2 - 6x_3 + 3x_4 - 0.15\dot{x}_3$	$3.006x_2 - 5.975x_3 + 2.969x_4 - 0.151\dot{x}_3$
	$\ddot{x}_4$	$3x_3 - 3x_4 - 0.15\dot{x}_4$	$2.993x_3 - 2.982x_4 - 0.149\dot{x}_4$
scheme 3	$\ddot{x}_1$	$-2x_1^3$	$-1.996x_1^3$
	$\ddot{x}_2$	0	0
	$\ddot{x}_3$	0	0
	$\ddot{x}_4$	0	0

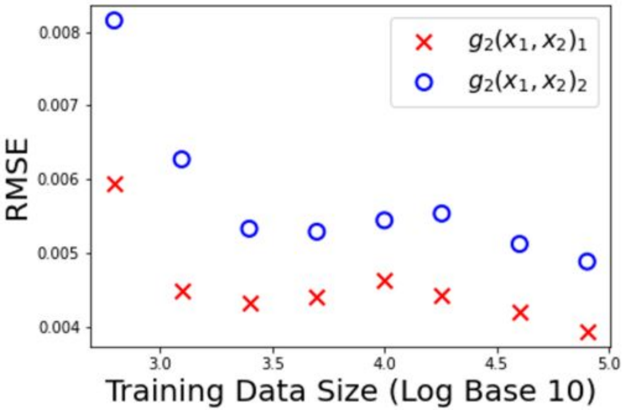
LAI, Zhilu et al. Structural identification with physics-informed neural ordinary differential equations. Journal of Sound and Vibration, v. 508, p. 116196, 2021.

Verification of Machine Learning models

How hybrid approaches are validated?

Evolution of error

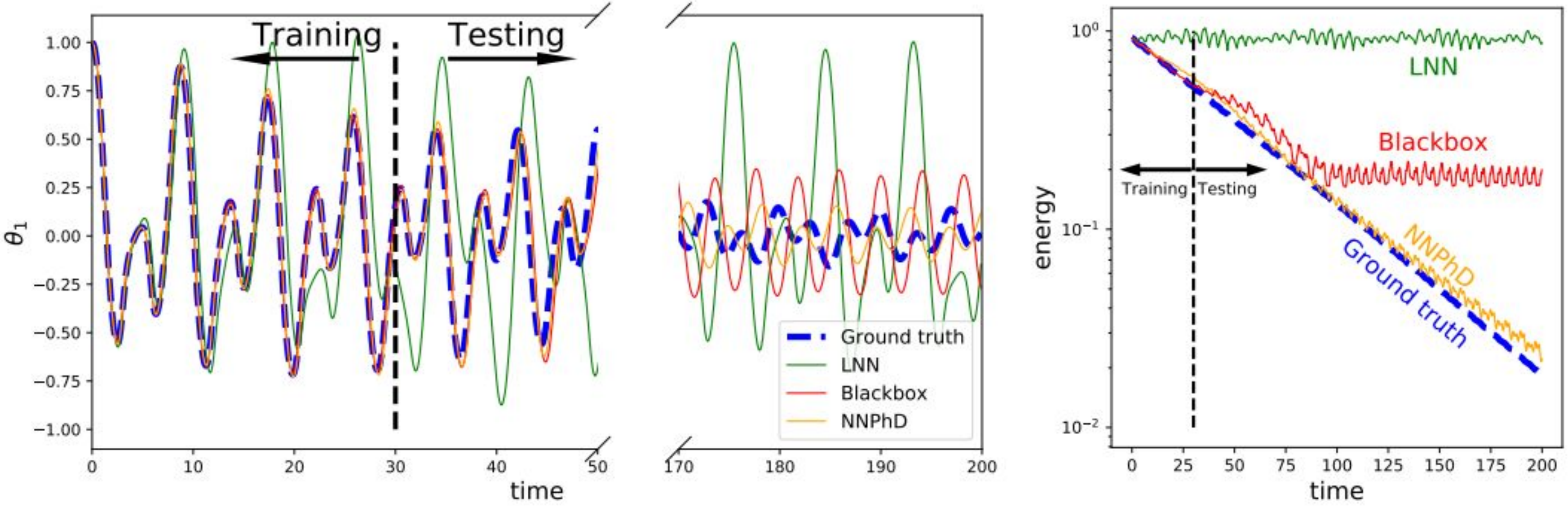




O'LEARY, Jared; PAULSON, Joel A.; MESBAH, Ali. Stochastic physics-informed neural ordinary differential equations. Journal of Computational Physics, v. 468, p. 111466, 2022.

How hybrid approaches are validated?

Energy



LIU, Ziming et al. Machine-learning nonconservative dynamics for new-physics detection. **Physical Review E**, v. 104, n. 5, p. 055302, 2021.

Open Questions

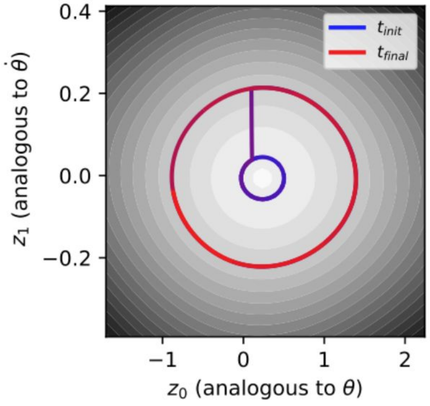
- What are the **key physical properties** that are crucial for evaluating the representation of a physical system by a neural network?

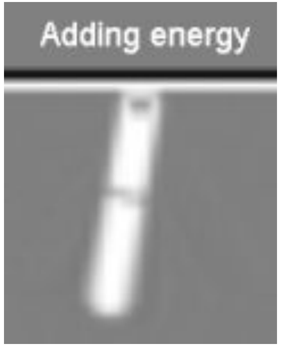
- How can error analysis help in identifying discrepancies between predicted and observed physical properties, thus indicating where the neural network might encounter challenges in accurately capturing system dynamics?

- In what way can partial knowledge of a system serve as a useful guide for evaluating the efficacy of neural network solutions?

Aplications

- Examining the subtle impact of system modifications during monitoring.
- Identifying sources of error influence, including noise, model representativeness, solution uncertainty, and parameter dimensionality.
- Techniques for detecting and addressing abnormal behaviors in online learning environments.





Thank you!

Physical verification of Neural Network Models

josafat@lisha.ufsc.br