31st AAS/AIAA Space Flight Mechanics Meeting

# Physics-Informed Neural Networks Applied to a Series of Constrained Space Guidance Problems

Andrea D'Ambrosio[1],
Enrico Schiassi[2], Fabio Curti[1], Roberto Furfaro[2]

[1]School of Aerospace Engineering, Sapienza University of Rome

[2]Department of Systems and Industrial Engineering, The University of Arizona

Space Systems Engineering Laboratory

SAPIENZA UNIVERSITÀ DI ROMA

# Contents

# Introduction

- Optimal Control Problems (OCPs) represent an essential field within space engineering.
  - It is important to design minimum time, fuel/energy optimal trajectories for space missions.
- The goal of this work is to present a new methodology to solve Constrained Optimal Control Problems (COCPs) for space guidance by means of the novel Physics-Informed Neural Network (PINN) framework named Extreme Theory of Functional Connections (X-TFC).
  - Indirect method exploiting the Pontryagin Minimum Principle (PMP) is used to retrieve the optimal control.
  - Problems considered: Feldbaum problem (typical OCP), minimum time – energy optimal Halo - Halo transfer, 1D fuel optimal lunar landing.

# Constrained Optimal Control Problems (COCPs)

### Constrained Optimal Control Problem (COCP)

$$\mathcal{J} = \boldsymbol{\Phi}(\boldsymbol{x}(t_0), t_0, \boldsymbol{x}(t_f), t_f) + \int_{t_0}^{t_f} \mathscr{L}(\boldsymbol{x}(t), \boldsymbol{u}(t), t)\, dt$$
$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}(t), \boldsymbol{u}(t), t)$$
$$\boldsymbol{\Phi}(\boldsymbol{x}(t_0), t_0) = \boldsymbol{\Phi}_0$$
$$\boldsymbol{\Phi}(\boldsymbol{x}(t_f), t_f) = \boldsymbol{\Phi}_f$$
$$u_i \in [d_i^-, d_i^+]$$

### New unconstrained control variable with saturation function

$$d_i = \phi_i(w_i)$$
$$\phi_i(w_i) = d_i^+ - \frac{d_i^+ - d_i^-}{1 + \exp(s w_i)} \quad \text{with} \quad s = \frac{c}{d_i^+ - d_i^-}$$

### Unconstrained Optimal Control Problem (COCP) with regularization term

$$\bar{\mathcal{J}} = \mathcal{J} + \epsilon \int_{t_0}^{t_f} ||w_i||^2\, dt$$
$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}(t), \boldsymbol{u}(t), t)$$
$$\boldsymbol{\Phi}(\boldsymbol{x}(t_0), t_0) = \boldsymbol{\Phi}_0$$
$$\boldsymbol{\Phi}(\boldsymbol{x}(t_f), t_f) = \boldsymbol{\Phi}_f$$

### Equality constraints in the Hamiltonian + first-order necessary conditions

$$H = \mathscr{L} + \boldsymbol{\lambda}^{\mathsf{T}} \boldsymbol{f} + \epsilon ||w_i||^2 + \sum_{i=1}^{q} \nu_i (d_i(u) - \phi_i(w_i))$$
$$\frac{\partial H}{\partial \boldsymbol{u}} = 0$$
$$\frac{\partial H}{\partial w_i} = 2\epsilon w_i - \nu_i \phi_i^{'}(w_i) = 0$$
$$\dot{\boldsymbol{x}} = \frac{\partial H}{\partial \boldsymbol{\lambda}}$$
$$\dot{\boldsymbol{\lambda}} = -\frac{\partial H}{\partial \boldsymbol{x}}$$
$$\frac{\partial H}{\partial \nu_i} = d_i(u) - \phi_i(w_i) = 0$$

### Transversality conditions

$$\boldsymbol{\lambda}(t_0) = -\frac{\partial \mathcal{J}}{\partial \boldsymbol{x}_0}$$
$$H(t_0) = \frac{\partial \mathcal{J}}{\partial t_0}$$
$$\boldsymbol{\lambda}(t_f) = \frac{\partial \mathcal{J}}{\partial \boldsymbol{x}_f}$$
$$H(t_f) = -\frac{\partial \mathcal{J}}{\partial t_f}$$

# Physics-Informed Neural Networks

- **(Data) + Neural Networks + Physics Laws = Physics-Informed Neural Networks (PINN)**
- PINNs are a newly developed framework for solving parametric DEs
  - The physics laws (modeled via parametric DEs), and eventually data, drive the training of the network

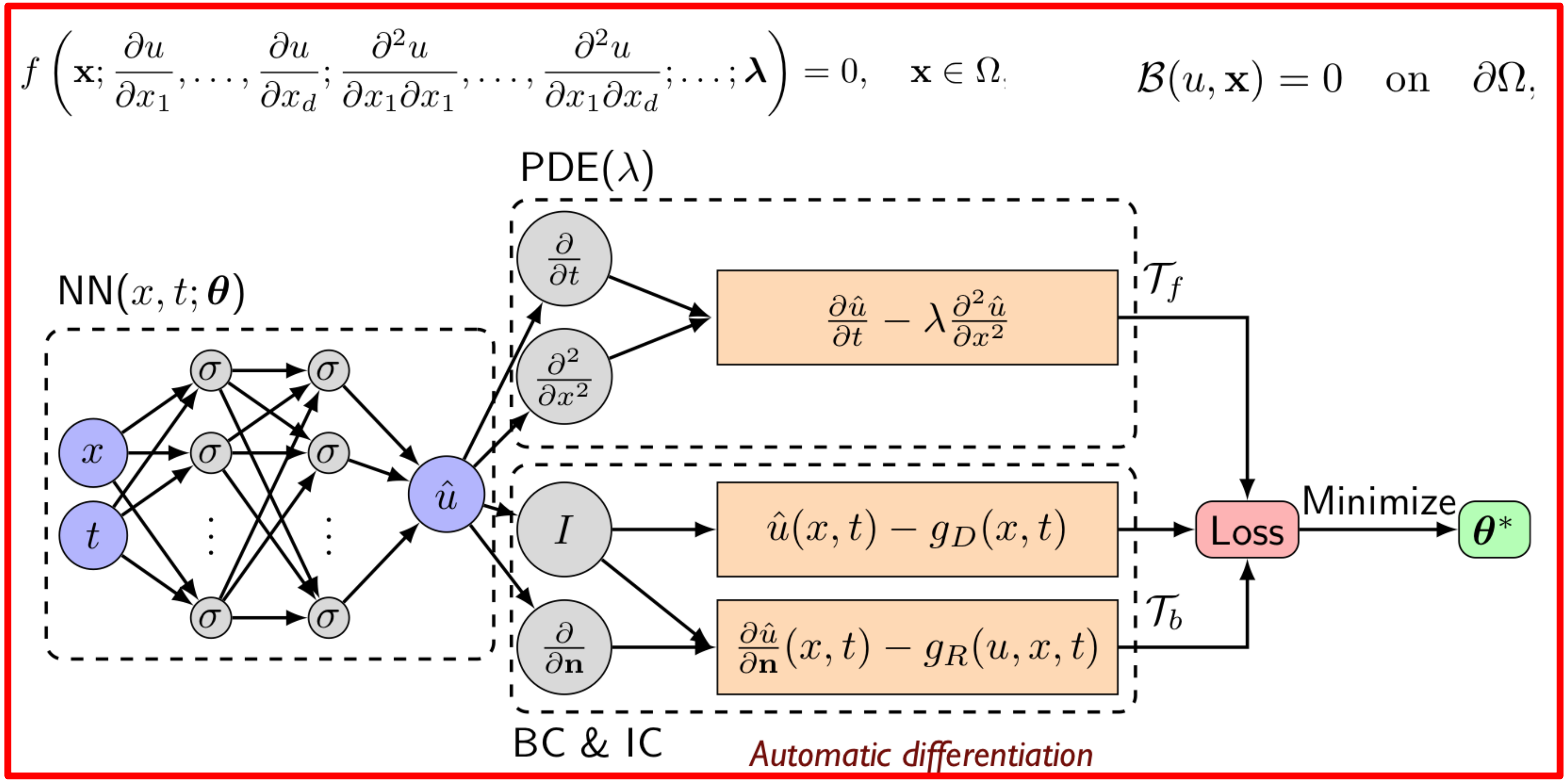



Image taken from: Lu, L., Meng, X., Mao, Z. and Karniadakis, G.E., 2019. DeepXDE: A deep learning library for solving differential equations. *arXiv preprint arXiv:1907.04502*.

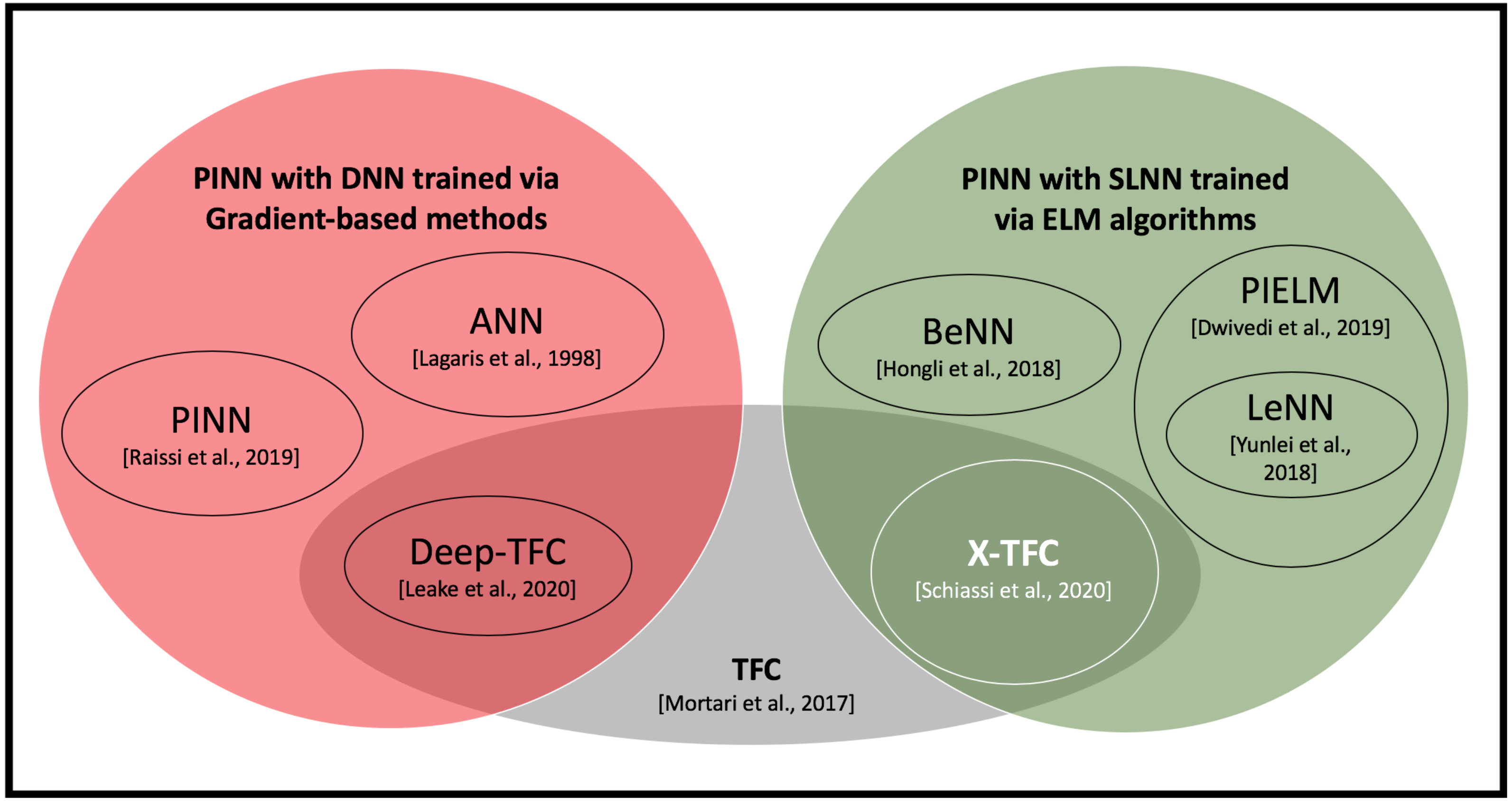
# PINN and TFC

- The **Theory of Functional Connections (TFC)** [Mortari, 2017] is a recently developed framework for functional interpolation
  - The functions are approximated via a **constrained expression**
    - Sum of a **free-chosen** function and a functional that **analytically** satisfies the constraints
  - **TFC can be applied to solve DEs**
    - The free-chosen function is an expansion of Chebyshev polynomials
    - The constraints are the Initial/Boundary Conditions (IC or BC)
- The **Physics-Informed Neural Network (PINN) Methods** are a novel approach, coming from the Machine Learning community
  - The DE latent solutions are approximated via a (Deep) Neural Network (NN), and the DEs drive the NN training (i.e., it acts as regulator)

| | **Pros** | **Cons** |
|---|---|---|
| **TFC (w/Cheb. Pol.)** | • ICs/BCs always analytically satisfied<br>• Accurate Solutions<br>• Low Computational Time | • Non-linear problems can be very sensitive to the initial guesses<br>• It suffers of the curse of dimensionality when solving ODE/PDE problems |
| **PINN** | • Expanding the latent solution via NN allows to apply this method to solve high order PDEs (e.g. no cur) | • Many training points required for high accuracy (ICs/BCs not analytically satisfied)<br>• Computational expensive when gradient based methods are used to train the NN |

# Extreme Theory of Functional Connections (X-TFC)

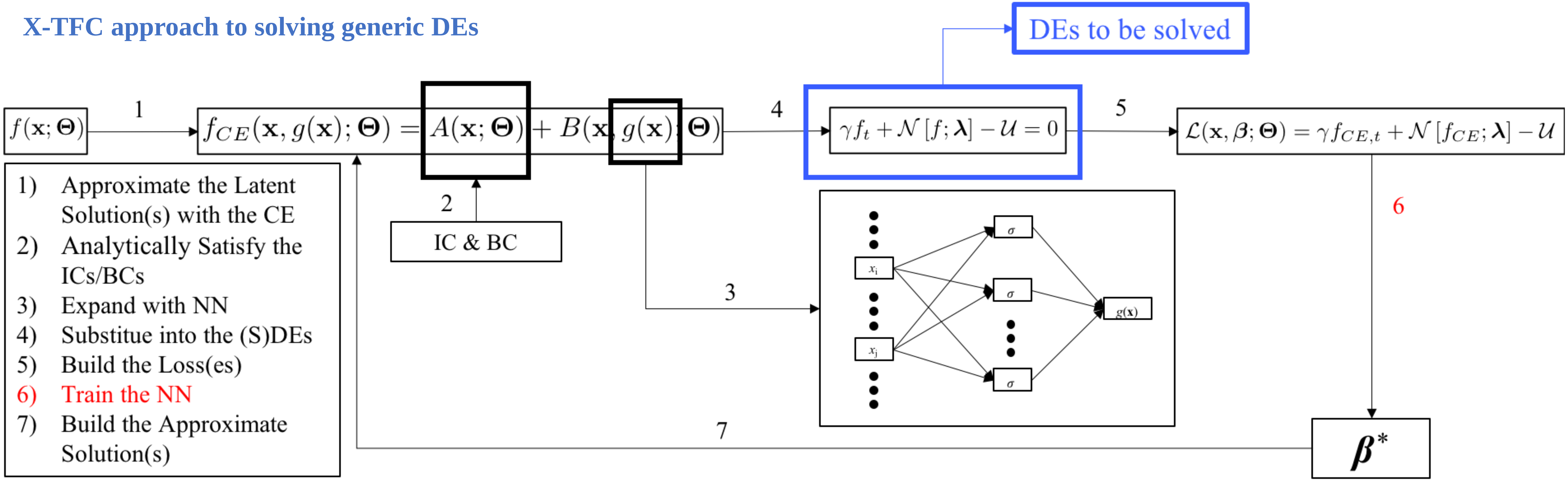
- The Physics-Informed **Extreme Theory of Functional Connections (X-TFC)** is a synergy of the TFC and the standard PINN methods that helps to overcome their limitations for solving DEs.



# Extreme Theory of Functional Connections (X-TFC)

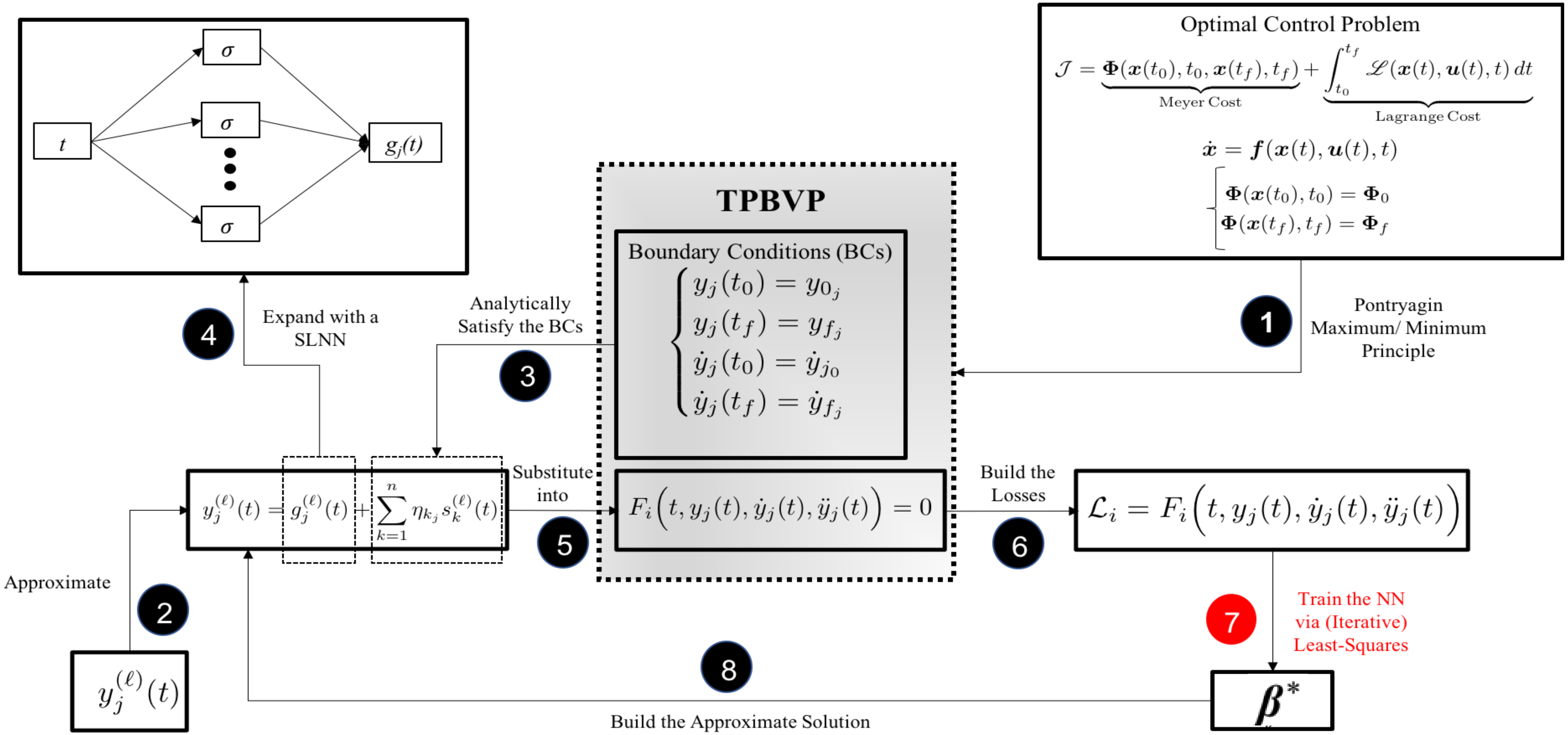
- X-TFC uses the TFC constrained expression where the free-chosen function $g$ is a Single Layer Feedforward NN (SLNN) trained via Extreme Learning Machine (ELM) Algorithm [Huang et al., 2006].

**X-TFC approach to solving generic DEs**



# Extreme Theory of Functional Connections (X-TFC)

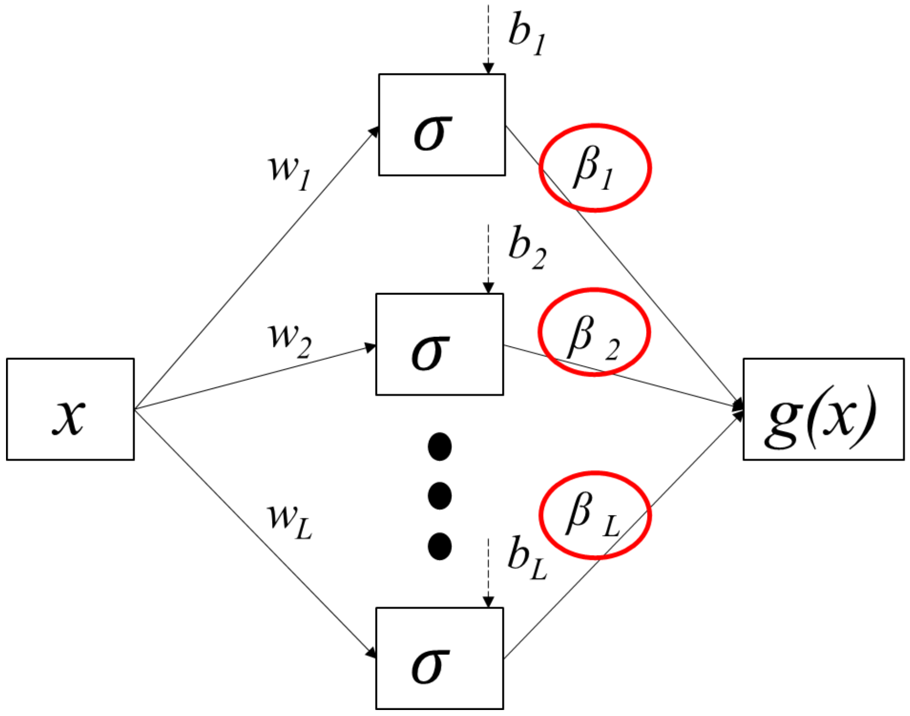
## X-TFC approach to solving generic UOCPs




- The Jacobian matrix required for an eventual iterative least-square procedure can be computed either analytically, or by the symbolic computation, or the automatic differentiation toolbox.

# Extreme Learning Machine (ELM)

- ELM is a training algorithm for Single Layer Feedforward Neural Network (SLNN) that randomly selects input weights and bias, and computes the output weights ($\boldsymbol{\beta}$) via least-square.
  - Input weights ($\boldsymbol{w}_i$) and bias ($b_i$) are not tuned during the training.
- The convergence of the ELM algorithm is proved by Huang et al (2006).
  - The convergence is guaranteed for any input weights and bias randomly chosen from any continuous probability distribution.




*SLNN example*

# Feldbaum Problem

- The Feldbaum problem is a typical optimal control problem.

**Original COCP**

$$\min \quad \mathcal{J} = \frac{1}{2}\int_0^1 (f^2 + u^2)\, dt$$

subject to

$$\dot{f} = \frac{df}{dt} = -f + u$$
$$0 \leq t \leq 1$$
$$f(0) = 1$$
$$u \in [u_{min}, u_{max}]$$

**Modified UOCP**

$$\min \quad \mathcal{J} = \frac{1}{2}\int_0^1 (f^2 + u^2)\, dt + \epsilon \int_0^1 w^2\, dt$$
$$H(t) = \frac{1}{2}\left(f^2 + u^2\right) + \lambda\left(-f + u\right) + \epsilon w^2 + \nu(u - \phi(w))$$
$$\frac{\partial H}{\partial u} = u + \lambda + \nu = 0$$
$$\frac{\partial H}{\partial w} = 2\epsilon w - \nu\phi'(w) = 0$$
$$\dot{f} = \frac{\partial H}{\partial \lambda} = -f + u$$
$$\dot{\lambda} = -\frac{\partial H}{\partial f} = \lambda - f$$
$$u = \phi(w)$$

- The following transversality condition has to be applied: $\lambda(1) = \lambda_f = 0$.

# Feldbaum Problem

- The CEs and their derivatives are:

$$
\begin{aligned}
f &= (\boldsymbol{\sigma} - \Omega_1 \boldsymbol{\sigma}_0)^{\mathrm{T}} \boldsymbol{\beta}_f + \Omega_1 f_0 \\
\lambda &= (\boldsymbol{\sigma} - \Omega_1 \boldsymbol{\sigma}_f)^{\mathrm{T}} \boldsymbol{\beta}_\lambda + \Omega_1 \lambda_f \\
w &= \boldsymbol{\sigma}^{\mathrm{T}} \boldsymbol{\beta}_w \\
u &= \boldsymbol{\sigma}^{\mathrm{T}} \boldsymbol{\beta}_u \\
\nu &= \boldsymbol{\sigma}^{\mathrm{T}} \boldsymbol{\beta}_\nu
\end{aligned}
$$

$$
\begin{aligned}
\dot{f} &= b^2 \left[ (\boldsymbol{\sigma}' - \Omega_1' \boldsymbol{\sigma}_0)^{\mathrm{T}} \boldsymbol{\beta}_f + \Omega_1' f_0 \right] \\
\dot{\lambda} &= b^2 \left[ (\boldsymbol{\sigma}' - \Omega_1' \boldsymbol{\sigma}_f)^{\mathrm{T}} \boldsymbol{\beta}_\lambda + \Omega_1' \lambda_f \right]
\end{aligned}
$$

Mapping coefficient from t in $[t_0; t_f]$ to z in [-1;1]

$$b^2 = c = \frac{z_f - z_0}{t_f - t_0}$$

- $\Omega$ are the **switching functions** and their analytical expressions are computed by imposing the boundary conditions.
- The unknowns of the problem are: $\boldsymbol{\beta} = \{ \boldsymbol{\beta}_f \quad \boldsymbol{\beta}_\lambda \quad \boldsymbol{\beta}_w \quad \boldsymbol{\beta}_u \quad \boldsymbol{\beta}_\nu \}^{\mathrm{T}}$
- The losses of the associated Two-Point Boundary Value Problem (TPBVP) are:

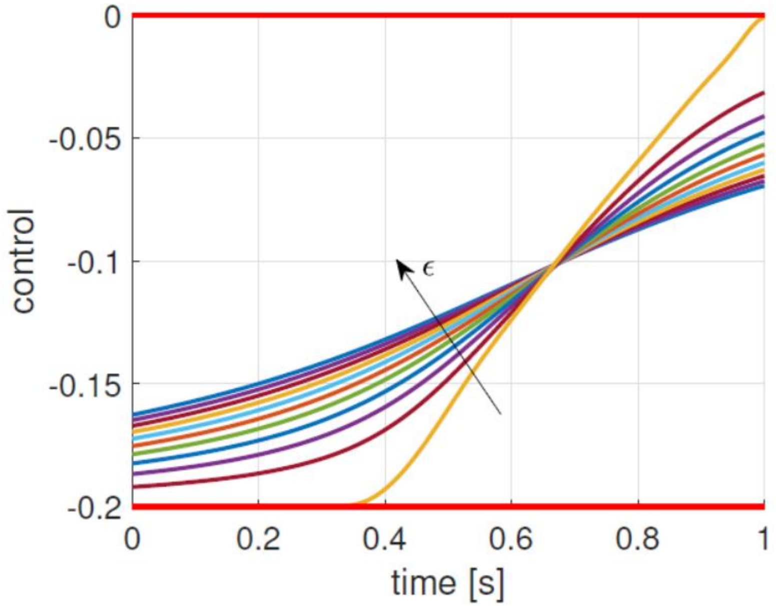
$$
\begin{aligned}
\mathcal{L}_f &= \dot{f} + f - u \\
\mathcal{L}_\lambda &= \dot{\lambda} - \lambda + f \\
\mathcal{L}_u &= u + \lambda + \nu \\
\mathcal{L}_w &= 2\epsilon w - \nu \phi'(w) \\
\mathcal{L}_d &= u - \phi(w)
\end{aligned}
$$

# Feldbaum Problem

- Parameters employed for the Feldbaum problem:

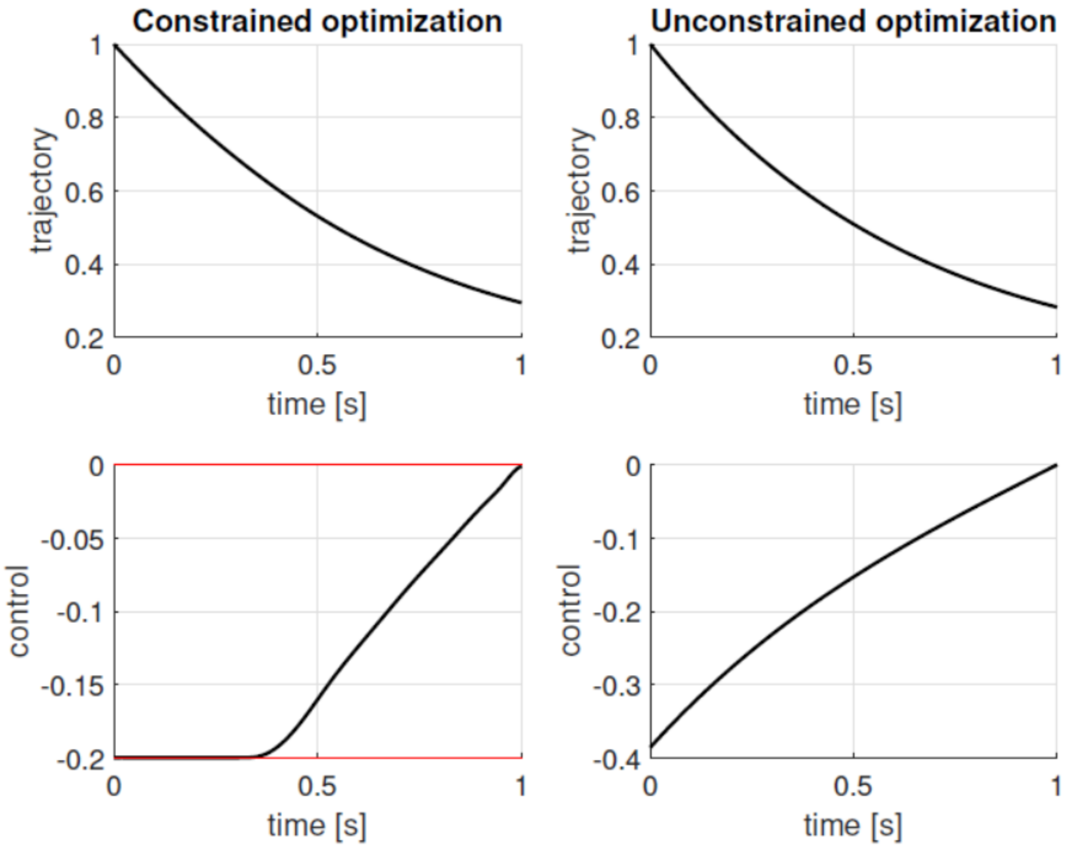
| $n$ | $L$ | $\sigma$ | $[u_{min}, u_{max}]$ | $c$ |
|---|---|---|---|---|
| 100 | 25 | Gaussian | [-0.2,0] | 4 |

- For all the chosen problems, the **initial guess** of the unknowns are chosen **randomly** within the interval (0,1).
- In order to make the UOCP close enough to the original COCP, a continuation procedure has been applied to decrease the value of $\epsilon$ from 1 to $10^{-5}$, while maintaining a good accuracy.
- Decreasing the value of $\epsilon$ induces higher values of the fictitious control *w*, which approaches the asymptotic limits.

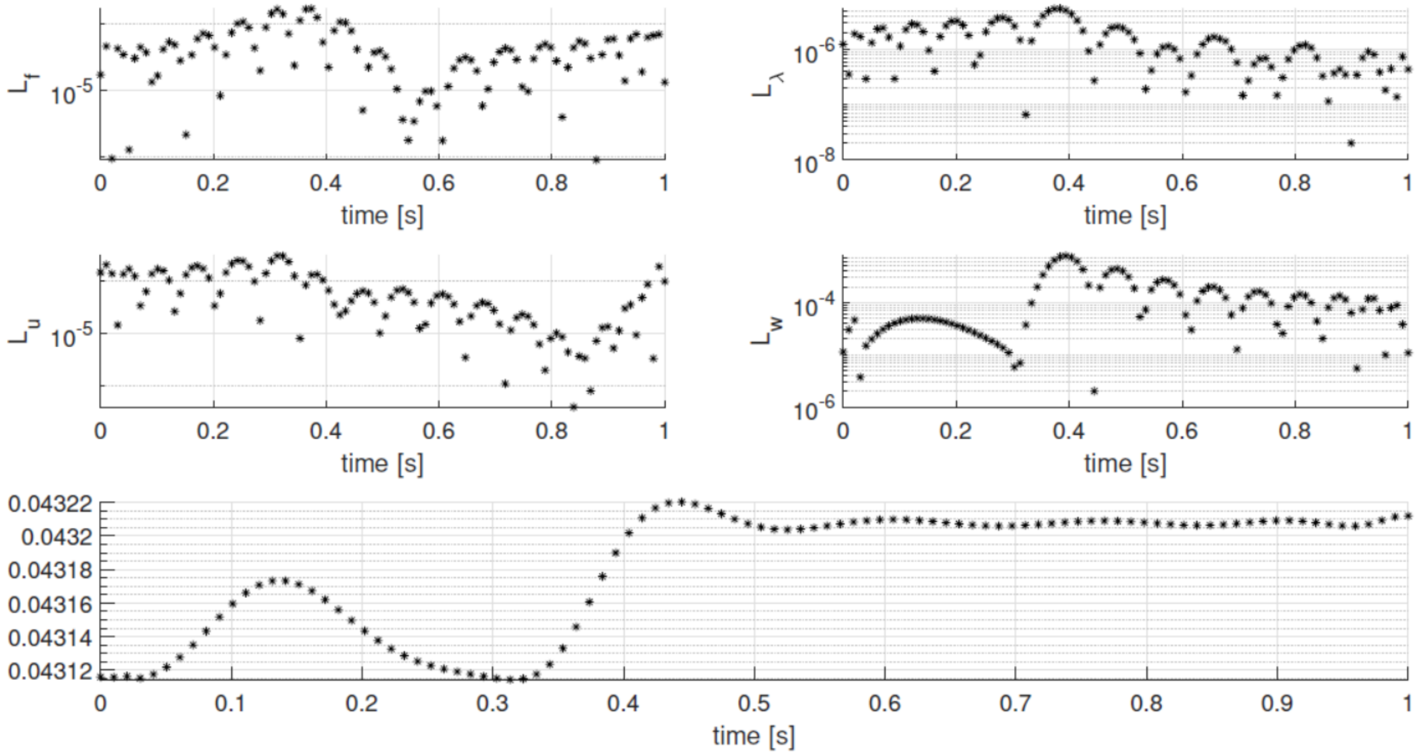


# Feldbaum Problem

- Results for the constrained Feldbaum problem with $\epsilon = 10^{-5}$




Trajectory and control plus the comparison with the unconstrained Feldbaum problem version




Performances of the constrained Feldbaum problem

# Minimum time – energy optimal Halo - Halo transfer

- The Circular Restricted Three Body Problem (CR3BP) framework is employed to study **constrained minimum time – energy optimal Halo-Halo transfers** in the Earth-Moon system (dimensionless units are considered).
- Due to the complexity of the problem, a **2-loop optimization procedure** is employed. An outer loop based on the Particle Swarm Optimization (PSO) is used to minimize the time, whereas the fixed time - energy optimal problem is solved rapidly via indirect method and X-TFC in the inner loop.

$$\min \mathcal{J} = \Gamma t_f + \frac{1}{2}\int_{t_0}^{t_f} \mathbf{u}^T\mathbf{u}\,dt + \epsilon\int_{t_0}^{t_f} \mathbf{w}^T\mathbf{w}\,dt \qquad \text{subject to:} \quad \begin{cases} \dot{\mathbf{r}} = \mathbf{v} \\ \dot{\mathbf{v}} = \nabla U(\mathbf{r}) + \mathbf{M}\mathbf{v} + \mathbf{u} \\ t_0 \le t \le t_f \\ \mathbf{r}(t_0) = \mathbf{r}_0 \\ \mathbf{v}(t_0) = \mathbf{v}_0 \\ \mathbf{r}(t_f) = \mathbf{r}_f \\ \mathbf{v}(t_f) = \mathbf{v}_f \\ u_{\min} \le u_i \le u_{\max} \end{cases}$$

- The Hamiltonian of the problem is:

$$H = \frac{1}{2}\mathbf{u}^T\mathbf{u} + \epsilon\mathbf{w}^T\mathbf{w} + \boldsymbol{\lambda}_r^T\mathbf{v} + \boldsymbol{\lambda}_v^T(\nabla U(\mathbf{r}) + \mathbf{M}\mathbf{v} + \mathbf{u}) + \boldsymbol{\nu}^T(\mathbf{u} - \phi(\mathbf{w}))$$

# Minimum time – energy optimal Halo - Halo transfer

- By using the CEs, the latent solutions are approximated.

$$\begin{aligned}
\mathbf{r} &= \left(\boldsymbol{\sigma} - \Omega_1\boldsymbol{\sigma}_0 - \Omega_2\boldsymbol{\sigma}_f - \Omega_3\boldsymbol{\sigma}_0' - \Omega_4\boldsymbol{\sigma}_f'\right)^{\mathrm{T}}\beta_r + \Omega_1\mathbf{r}_0 + \Omega_2\mathbf{r}_f + \frac{\Omega_3\mathbf{v}_0 + \Omega_4\mathbf{v}_f}{b^2} \\
\boldsymbol{\lambda}_r &= \boldsymbol{\sigma}^{\mathrm{T}}\beta_{\lambda_r} \\
\boldsymbol{\lambda}_v &= \boldsymbol{\sigma}^{\mathrm{T}}\beta_{\lambda_v} \\
\mathbf{w} &= \boldsymbol{\sigma}^{\mathrm{T}}\beta_w \\
\boldsymbol{\nu} &= \boldsymbol{\sigma}^{\mathrm{T}}\beta_\nu
\end{aligned}$$

- The first order necessary conditions and the corresponding losses are:

$$\begin{aligned}
\dot{\mathbf{r}} &= \frac{\partial H}{\partial \boldsymbol{\lambda}_r} = \mathbf{v} \\
\dot{\mathbf{v}} &= \frac{\partial H}{\partial \boldsymbol{\lambda}_v} = \nabla U(\mathbf{r}) + \mathbf{M}\mathbf{v} + \mathbf{u} \\
\dot{\boldsymbol{\lambda}}_r &= -\frac{\partial H}{\partial \mathbf{r}} = -\nabla\nabla U^T\boldsymbol{\lambda}_v \\
\dot{\boldsymbol{\lambda}}_v &= -\frac{\partial H}{\partial \mathbf{v}} = -\boldsymbol{\lambda}_r - \mathbf{M}^T\boldsymbol{\lambda}_v
\end{aligned}$$

$$\begin{aligned}
\frac{\partial H}{\partial \mathbf{u}} &= \mathbf{u} + \boldsymbol{\lambda}_v + \boldsymbol{\nu} = \mathbf{0} \\
\frac{\partial H}{\partial \mathbf{w}} &= 2\epsilon\mathbf{w} - \boldsymbol{\nu}\odot\phi'(\mathbf{w}) = \mathbf{0} \\
\frac{\partial H}{\partial \boldsymbol{\nu}} &= \mathbf{u} - \phi(\mathbf{w}) = \mathbf{0}
\end{aligned}$$

$\longrightarrow$

$$\begin{aligned}
\mathcal{L}_a &= \dot{\mathbf{v}} - \nabla U(\mathbf{r}) - \mathbf{M}\mathbf{v} - \phi(\mathbf{w}) \\
\mathcal{L}_{\lambda_r} &= \dot{\boldsymbol{\lambda}}_r + \nabla\nabla U^T\boldsymbol{\lambda}_v \\
\mathcal{L}_{\lambda_v} &= \dot{\boldsymbol{\lambda}}_v + \boldsymbol{\lambda}_r + \mathbf{M}^T\boldsymbol{\lambda}_v \\
\mathcal{L}_u &= \phi(\mathbf{w}) + \boldsymbol{\lambda}_v + \boldsymbol{\nu} \\
\mathcal{L}_w &= 2\epsilon\mathbf{w} - \boldsymbol{\nu}\odot\phi'(\mathbf{w})
\end{aligned}$$

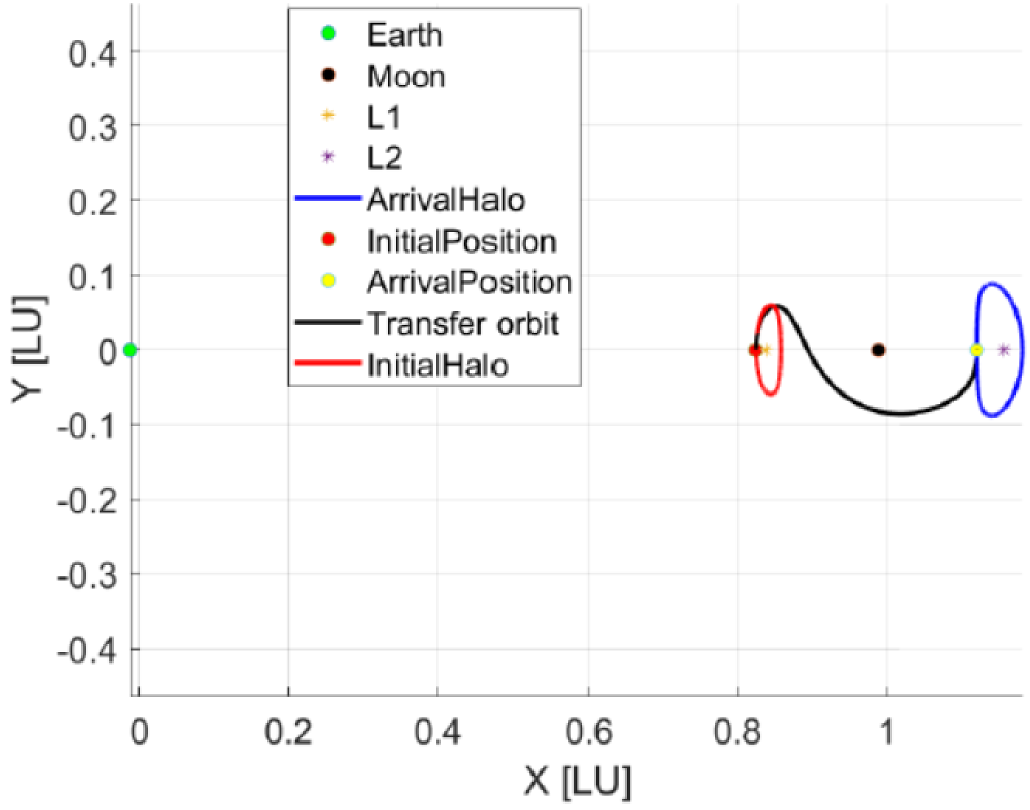
# Minimum time – energy optimal Halo - Halo transfer

- The goal is to pass from a L1 Halo to a L2 Halo orbit.
- For PSO, the cost function associated to the particles is the sum of the time of flight and the norm of the loss vector.
- The other parameters used for this example are:

| $n$ | $L$ | $\sigma$ | $[u_{i,min}, u_{i,max}]$ km/s$^2$ | c |
|---|---|---|---|---|
| 100 | 35 | Logistic | $[-5, 5] \cdot 10^{-7}$ | 10 |

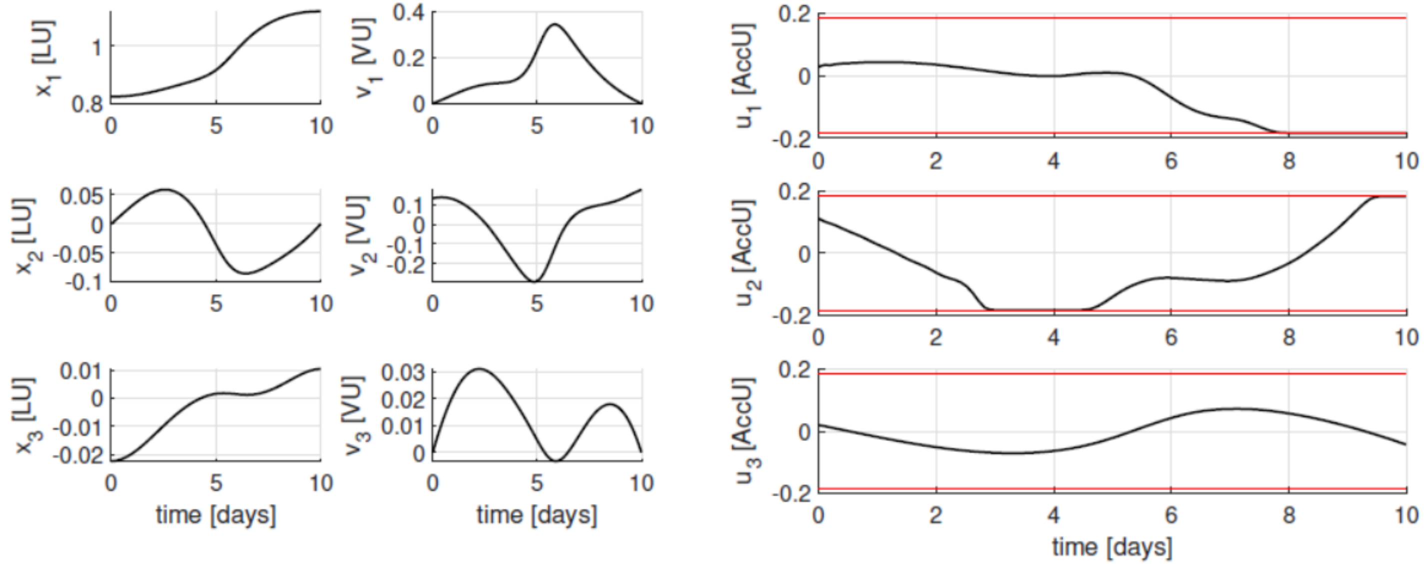
- A continuation procedure has been applied to decrease the value of $\epsilon$ from 1 to $10^{-6}$.
- The following boundary conditions are considered:

$$\begin{aligned}
\mathbf{r}_0 &= [0.823385182067467, 0, -0.022277556273235] \\
\mathbf{v}_0 &= [0, 0.134184170262437, 0] \\
\mathbf{r}_f &= [1.119601641146371, 0, 0.010405691913477] \\
\mathbf{v}_f &= [0, 0.178317957880263, 0]
\end{aligned}$$



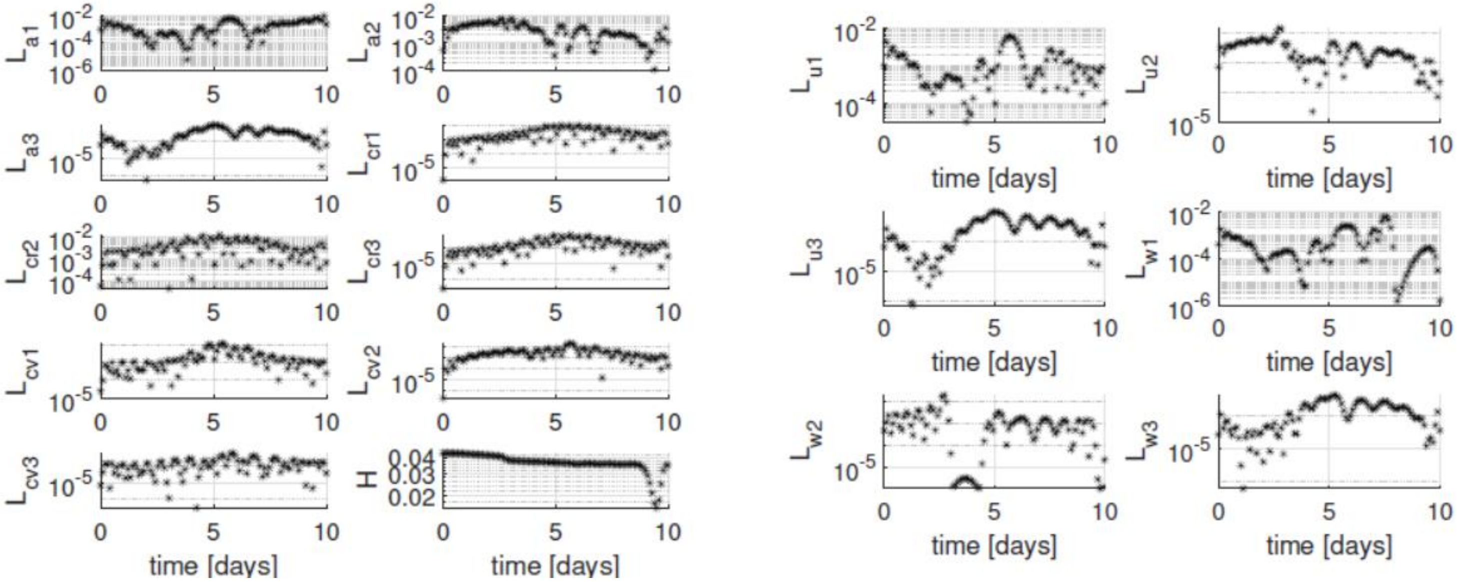

Trajectory with $\epsilon = 10^{-6}$

# Minimum time – energy optimal Halo - Halo transfer




States and control
with $\epsilon = 10^{-6}$

- Time of flight = 9.98 days.
- $\Delta V$ = 0.36 m/s.




Performances
with $\epsilon = 10^{-6}$

# 1D fuel optimal lunar landing

- The 1D fuel optimal landing has been chosen to test the ability of the proposed framework to deal with possible discontinuities in the control.
- The fuel optimal problem is intrinsically constrained between a maximum and a minimum value. The use of a saturation function could result in a convenient choice.

$$\min \mathcal{J} = \int_{t_0}^{t_f} u(t)\,dt + \int_{t_0}^{t_f} \epsilon\, w^2(t)\,dt \quad \text{subject to:} \quad \begin{cases} \dot{h} = v \\ \dot{v} = -g + u \\ t_0 \leq t \leq t_f \\ h(t_0) = h_0 \\ v(t_0) = v_0 \\ h(t_f) = h_f \\ v(t_f) = v_f \\ u_{\min} \leq u \leq u_{\max} \end{cases}$$

- The Hamiltonian of the problem is:

$$H = u + \lambda_h v + \lambda_v(-g + u) + \epsilon w^2 + \nu(u - \phi(w))$$

# 1D fuel optimal lunar landing

- By using the CEs, the latent solutions are approximated.

$$\begin{aligned} h &= \left(\boldsymbol{\sigma} - \Omega_1\boldsymbol{\sigma}_0 - \Omega_2\boldsymbol{\sigma}_f - \Omega_3\boldsymbol{\sigma}'_0 - \Omega_4\boldsymbol{\sigma}'_f\right)^{\mathrm{T}}\boldsymbol{\beta}_h + \Omega_1 h_0 + \Omega_2 h_f + \frac{\Omega_3 v_0}{b^2} + \frac{\Omega_4 v_f}{b^2} \\ \lambda_h &= \boldsymbol{\sigma}^{\mathrm{T}}\boldsymbol{\beta}_{\lambda_h} \\ \lambda_v &= \boldsymbol{\sigma}^{\mathrm{T}}\boldsymbol{\beta}_{\lambda_v} \\ w &= \boldsymbol{\sigma}^{\mathrm{T}}\boldsymbol{\beta}_w \\ \nu &= \boldsymbol{\sigma}^{\mathrm{T}}\boldsymbol{\beta}_\nu \end{aligned}$$

- The first order necessary conditions and the corresponding losses are:

$$\begin{aligned} \dot{h} &= \frac{\partial H}{\partial \lambda_h} = v \\ \dot{v} &= \frac{\partial H}{\partial \lambda_v} = -g + u \\ \dot{\lambda}_h &= -\frac{\partial H}{\partial h} = 0 \\ \dot{\lambda}_v &= -\frac{\partial H}{\partial v} = -\lambda_h \end{aligned}$$

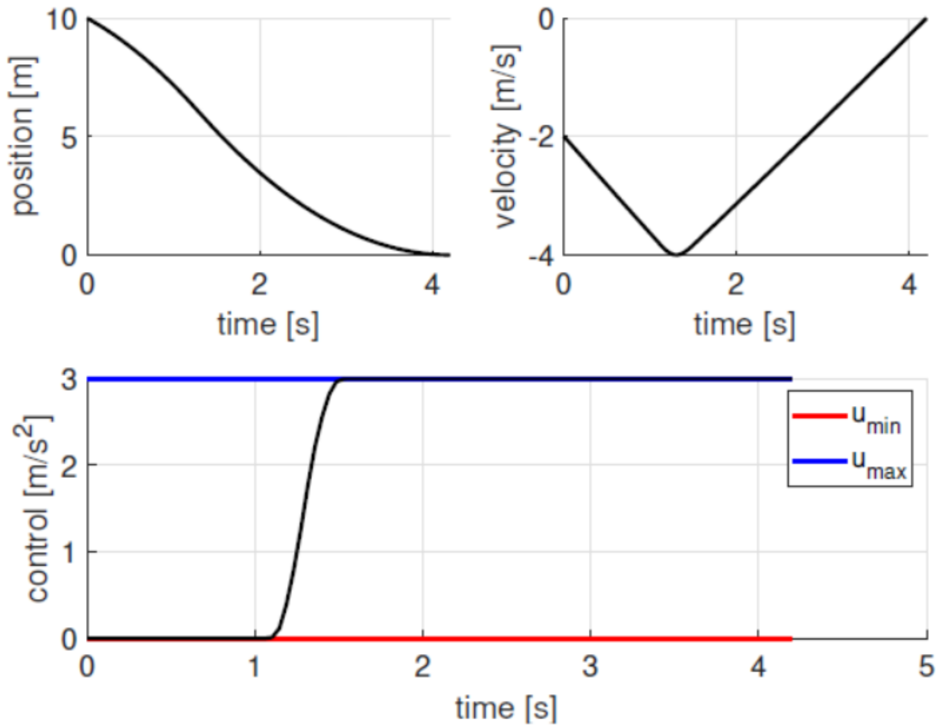
$$\begin{aligned} \frac{\partial H}{\partial u} &= 1 + \lambda_v + \nu = 0 \\ \frac{\partial H}{\partial w} &= 2\epsilon w - \nu\phi'(w) = 0 \\ \frac{\partial H}{\partial \nu} &= u - \phi(w) = 0 \end{aligned} \longrightarrow$$

$$\begin{aligned} \mathcal{L}_s &= \dot{v} + g - \phi(w) \\ \mathcal{L}_{\lambda_h} &= \dot{\lambda}_h \\ \mathcal{L}_{\lambda_v} &= \dot{\lambda}_v + \lambda_h \\ \mathcal{L}_u &= 1 + \lambda_v + \nu \\ \mathcal{L}_w &= 2\epsilon w - \nu\phi'(w) \end{aligned}$$

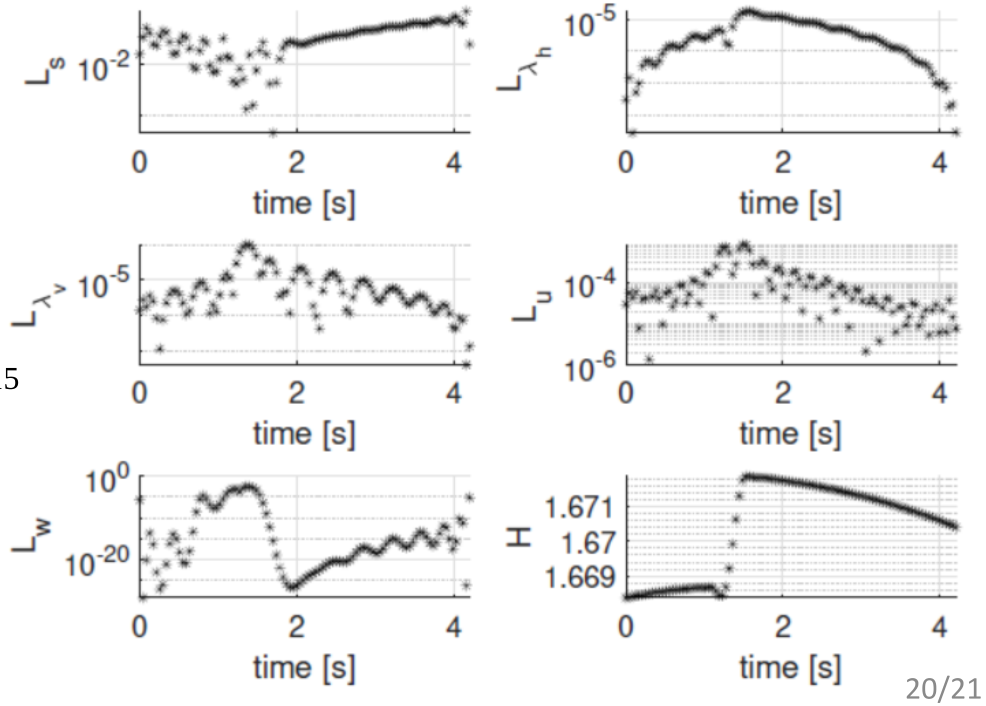
# 1D fuel optimal lunar landing

- The following conditions are imposed: $r_0 = 10$ m, $v_0 = -2$ m/s, $t_f = 4.2$ s, $g = 1.62$ m/s$^2$ .
- A continuation procedure has been applied to decrease both the value of $\epsilon$ from 100 to $10^{-15}$ and the coefficient $c$ of the saturation function from $c_0$ to $c_f$.
- The parameters employed for the simulation are:

| $n$ | $L$ | $\sigma$ | $[u_{min}, u_{max}]$ m/s$^2$ | $c_0$ | $c_f$ |
|---|---|---|---|---|---|
| 100 | 30 | Logistic | [0,3] | 100 | 10 |




States, control and performances with $\epsilon = 10^{-15}$



# Conclusions

- A new PINN-based algorithm is proposed to solve optimal control problems with control constraints using saturation functions.
- X-TFC has been successfully employed to solve the TPBVP arising from the application of the PMP to the Hamiltonian of the problem.
- COCPs for space guidance have been solved
  - Minimum time-energy optimal Halo-Halo transfer.
  - 1D fuel optimal lunar landing. The bang-bang and bang-off-bang type of solution could be well approximated by means of saturation functions.
- The proposed algorithm can be potentially suitable for real time applications if a compromise between the optimality of the results and the precision of the dynamics is carried out.

➢ The possibility to solve more complex fuel optimal problems will be taken into account considering a combination of saturation functions.

# THANK YOU FOR YOUR ATTENTION!

**Andrea D'Ambrosio**
andrea.dambrosio@uniroma1.it

Space Systems
Engineering Laboratory

SAPIENZA
UNIVERSITÀ DI ROMA