Applying Artificial Intelligence techniques to the orbit propagation problem

Juan Félix San-Juan

Scientific Computing Group (GRUCACI)

UNIVERSIDAD DE LA RIOJA

1st International Round Table on Intelligent Control for Space Missions
November 24, 2017
ESTEC, Noordwijk (The Netherlands)

Outline

1. **Motivation (Industry/Space 4.0)**
2. **Orbit propagation problem**
3. **Hybrid propagation methodology**
4. **Forecasting technique: Neural networks**
5. **Hybrid SGP4 for Galileo-type orbits**
6. **Conclusions**

1. **Motivation (Industry/Space 4.0)**
2. Orbit propagation problem
3. Hybrid propagation methodology
4. Forecasting technique: Neural networks
5. Hybrid SGP4 for Galileo-type orbits
6. Conclusions

Space 4.0 is analogous to, and is intertwined with, Industry 4.0

Automation and robotics provide the muscle for Industry 4.0, AR/VR, cameras and other sensors provide the senses, and data and connectivity are its central nervous system. But the real brains behind this industrial revolution is AI (Artificial Intelligence)...

Joanne Moretti (https://www.jabil.com/insights/blog-main/artifical-intelligence-brains-behind-industry-40.html)

1. Motivation (Industry/Space 4.0)

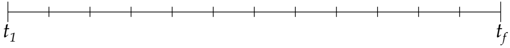
Space 4.0 is analogous to, and is intertwined with, Industry 4.0

Automation and robotics provide the muscle for Industry 4.0, AR/VR, cameras and other sensors provide the senses, and data and connectivity are its central nervous system. But the real brains behind this industrial revolution is AI (Artificial Intelligence)...

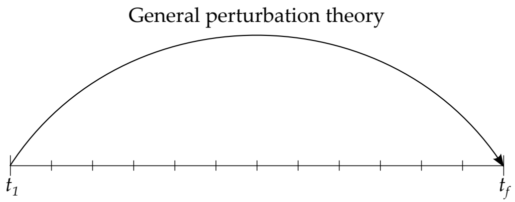
Joanne Moretti (https://www.jabil.com/insights/blog-main/artifical-intelligence-brains-behind-industry-40.html)

1. Motivation (Industry/Space 4.0)
2. **Orbit propagation problem**
3. Hybrid propagation methodology
4. Forecasting technique: Neural networks
5. Hybrid SGP4 for Galileo-type orbits
6. Conclusions

2. Orbit propagation problem

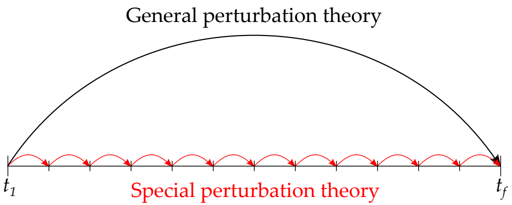


Classical resolution methods:



Classical resolution methods:

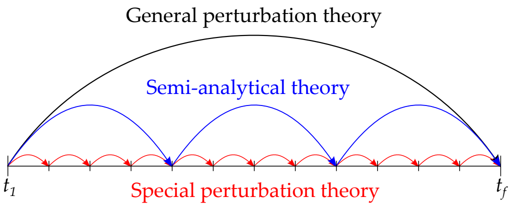
- **General perturbation theory**:
  - Series expansions + Analytical integration.
  - Low accuracy.



Classical resolution methods:

- **General perturbation theory**:
  - Series expansions + Analytical integration.
  - Low accuracy.
- **Special perturbation theory**:
  - Numerical integration.
  - Slow process.

2. Orbit propagation problem



Classical resolution methods:

- **General perturbation theory**:
  - Series expansions + Analytical integration.
  - Low accuracy.
- **Special perturbation theory**:
  - Numerical integration.
  - Slow process.
- **Semi-analytical theory**:
  - Elimination of short-period components + Numerical integration.
  - Intermediate accuracy and speed.

3. Hybrid propagation methodology

How can the classical theories be enhanced?:

- Improvement of the **physical models**.
- Higher orders in **analytical** and **semi-analytical** theories, making use of advanced perturbation models.
- **Increasing the computational efficiency of classical orbit propagators**: parallel computing (*multicore, GPUs*) or quantum computing.
- . . .

3. Hybrid propagation methodology

1. Motivation
2. Orbit propagation problem
3. **Hybrid propagation methodology**
4. Forecasting technique: Neural networks
5. Hybrid SGP4 for Galileo-type orbits
6. Conclusions

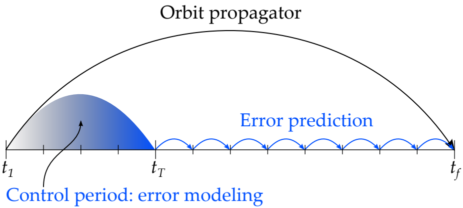
3. Hybrid propagation methodology

Hybrid propagation methodology:

1. **Classical theory** (initial approximation).

+

2. **Forecasting technique** (error estimation):
   - Statistical time series model.
   - Machine learning method.



Procedure:

1. Initial approximation at t_f:

$$\mathbf{x}^{\mathcal{I}}_{t_f} = \mathcal{I}(t_f, \mathbf{x}_{t_1}).$$

2. Time series of the error during the control period ($i : 1, \ldots, T$):

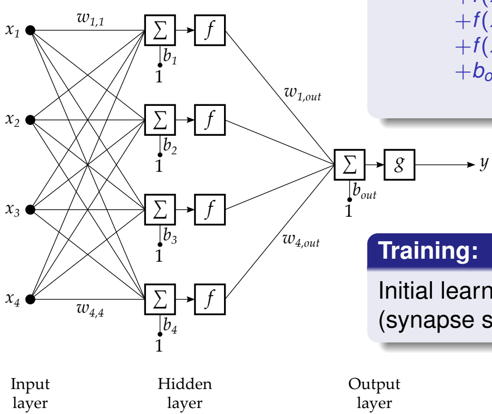
$$\varepsilon_{t_i} = \mathbf{x}_{t_i} - \mathbf{x}^{\mathcal{I}}_{t_i}.$$

3. Error modeling during the control period.
4. Error prediction at t_f: $\hat{\varepsilon}_{t_f}$.
5. Estimated value at t_f:

$$\hat{\mathbf{x}}_{t_f} = \mathbf{x}^{\mathcal{I}}_{t_f} + \hat{\varepsilon}_{t_f}.$$

4. Forecasting technique: Neural networks

1. Motivation
2. Orbit propagation problem
3. Hybrid propagation methodology
4. **Forecasting technique: Neural networks**
5. Hybrid SGP4 for Galileo-type orbits
6. Conclusions



Multi-layer feed-forward neural network:

$$
\begin{aligned}
y = \; & g(f(x_1\omega_{1,1} + x_2\omega_{2,1} + x_3\omega_{3,1} + x_4\omega_{4,1} + b_1)\,\omega_{1,out} \\
& + f(x_1\omega_{1,2} + x_2\omega_{2,2} + x_3\omega_{3,2} + x_4\omega_{4,2} + b_2)\,\omega_{2,out} \\
& + f(x_1\omega_{1,3} + x_2\omega_{2,3} + x_3\omega_{3,3} + x_4\omega_{4,3} + b_3)\,\omega_{3,out} \\
& + f(x_1\omega_{1,4} + x_2\omega_{2,4} + x_3\omega_{3,4} + x_4\omega_{4,4} + b_4)\,\omega_{4,out} \\
& + b_{out})
\end{aligned}
$$

Training:

Initial learning process aimed at fitting weights $\boldsymbol{\omega_{i,j}}$ (synapse strength) and bias $\boldsymbol{b_k}$ (trigger threshold).

5. Hybrid SGP4 for Galileo-type orbits

1. Motivation
2. Orbit propagation problem
3. Hybrid propagation methodology
4. Forecasting technique: Neural networks
5. **Hybrid SGP4 for Galileo-type orbits**
6. Conclusions

5. Hybrid SGP4 for Galileo-type orbits

Hybrid propagation:

1. **SGP4**.
   - Analytical integration.
   - Zonal harmonics J_2, J_3, J_4.
   - Geopotential resonance for 12- and 24-hour orbits.
   - Third body effect.
   - Atmospheric drag.

+

2. **Neural network correction**.

ARIADNA (ACT)

5. Hybrid SGP4 for Galileo-type orbits

Hybrid propagation:

1. **SGP4.**
   - Analytical integration.
   - Zonal harmonics J_2, J_3, J_4.
   - Geopotential resonance for 12- and 24-hour orbits.
   - Third body effect.
   - Atmospheric drag.

+

2. **Neural network correction.**

Accurate ephemerides: AIDA.

- Numerical integration.
- 50×50 gravitational field.
- Third body effect.
- Solar radiation pressure.
- Atmospheric drag.

ARIADNA (ACT)

5. Hybrid SGP4 for Galileo-type orbits

1. Get the data.

- TLEs of Galileo constellation (GALILEO-PFM, GALILEO-FM2, GALILEO-FM3, GALILEO-FM4, GALILEO 7 (263), GALILEO 8 (264),...) propagated by SGP4 and AIDA.
- Set of variables (Delaunay, polar-nodal and equinoctial variables).

$$\mathcal{V} = (l, g, h, L, G, H, r, \theta, \nu, R, \Theta, N, a, h, k, p, q, \lambda, I)_{\{A,S\}}$$

2. Data preprocessing.

- Study of the order of influence of each variables or combination of them.
- Error analysis (distance error).

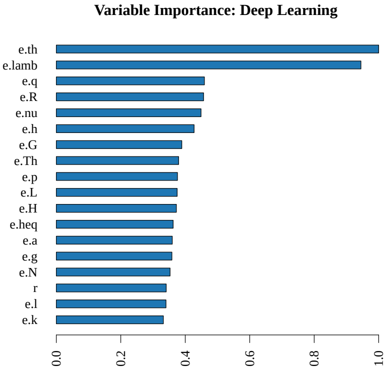
1. Brute-force analysis.

- Model one variable, 60 days of propagation.
  $\text{error}_l = \text{error}[(l_A, g_A, h_A, L_A, G_A, H_A), (l_A, g_S, h_S, L_S, G_S, H_S)]$

Delaunay	Distant	Equinoctial	Distant	Polar-nodal	Distant
None	69.0055	None	69.0055	None	69.0055
l	7430.32	a	69.005	r	69.005
g	7477.82	h	68.8001	$\boldsymbol{\theta}$	3.83258
h	70.5239	k	63.5887	h	70.5239
L	243.062	p	69.0772	R	70.5239
G	218.204	q	67.7076	Θ	69.0053
H	218.204	$\lambda = M + \omega + \Omega$	67.7076	N	69.0053
$\boldsymbol{l, g}$	6.42224	λ, a	7.31672	r, θ	3.13726
l, g, h	4.30029	λ, h	7.08983	θ, h	3.329
l, g, h, L	180.55	$\boldsymbol{\lambda, k}$	3.24524	θ, h, N	3.07805
l, g, h, G	201.371	λ, h, k	2.97125	r, θ, h	1.68063
l, g, h, H	4.22253	λ, q	6.28884	r, θ, h, R	1.68063
l, g, L, G	3.17302	λ, p, q	6.32335	r, θ, h, Θ	1.69206
$\boldsymbol{l, g, h, L, G}$	1.69206	$\boldsymbol{\lambda, h, k, p, q}$	0.457133	$\boldsymbol{r, \theta, h, N}$	0.124164

2. Using deep learning techniques (Gedeon's method)

$$(\varepsilon^{l}, \varepsilon^{g}, \varepsilon^{h}, \varepsilon^{L}, \varepsilon^{G}, \varepsilon^{H}, \varepsilon^{r}, \varepsilon^{\theta}, \varepsilon^{\nu}, \varepsilon^{R}, \varepsilon^{\Theta}, \varepsilon^{N}, \varepsilon^{a}, \varepsilon^{h}, \varepsilon^{k}, \varepsilon^{p}, \varepsilon^{q}, \varepsilon^{\lambda}, \mathrm{DistanceError})$$



Parsimonious models:

- Simpler models with similar accuracy.
- These **parsimonious models** are more robust, easier to maintain, and besides, they mitigate the effects of the curse of dimensionality.

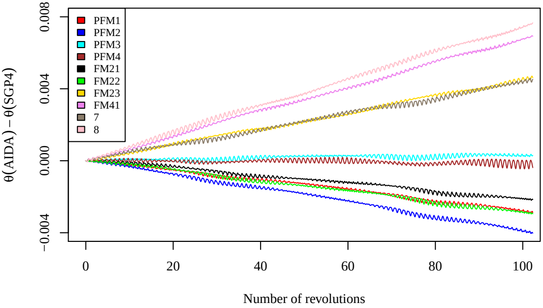
5. Hybrid SGP4 for Galileo-type orbits

Parsimonious models:

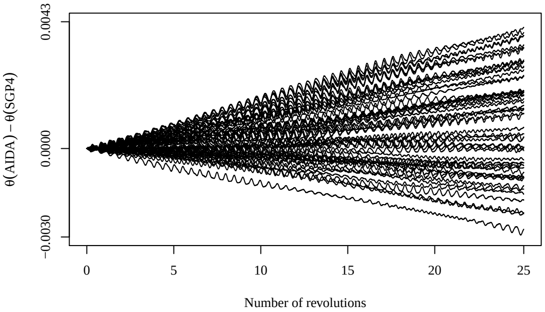
- Simpler models with similar accuracy.
- These **parsimonious models** are more robust, easier to maintain, and besides, they mitigate the effects of the curse of dimensionality.

Hybrid SGP4 for Galileo-type orbits based on modelling ε^{θ}.

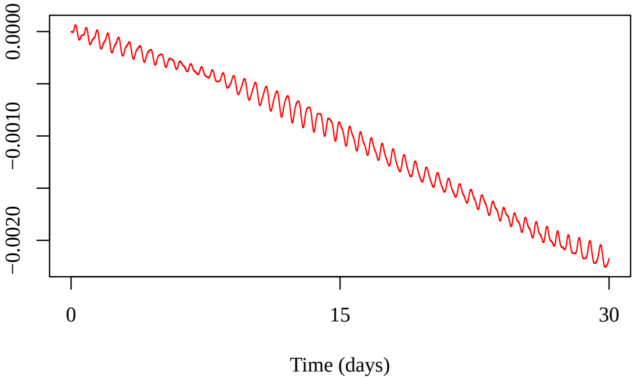
Behaviour of ε^{θ} for ten Galileo satellites.



Behaviour of ε^{θ} for 53 different TLEs of Galileo 8.



Behaviour of ε^θ for a TLE.



Forecasting strategy: Sliding window.

Inputs	Output
001 002 003 004 005 006 007 ... 498 499 500	501
002 003 004 005 006 007 ... 498 499 500 501	502
003 004 005 006 007 ... 498 499 500 501 502	503
004 005 006 007 ... 498 499 500 501 502 503	504

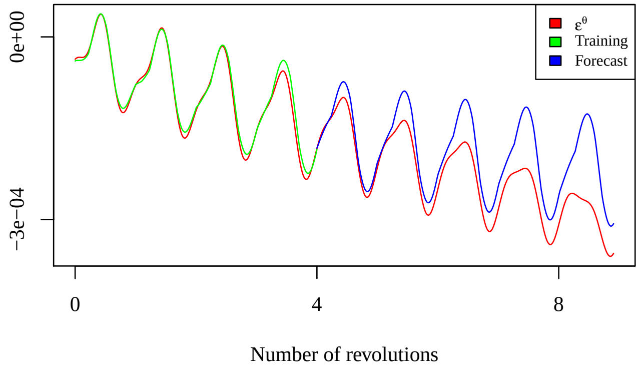
Process:

- **Preparation of vectors**: training data, validation data, test data.
- **Hyper-parameter optimization**.
- **Forecasting**.

5. Hybrid SGP4 for Galileo-type orbits

Inputs: 1720(1 rev.), **Training data**: 4 satellite revolutions, **Hidden layers**: 1.

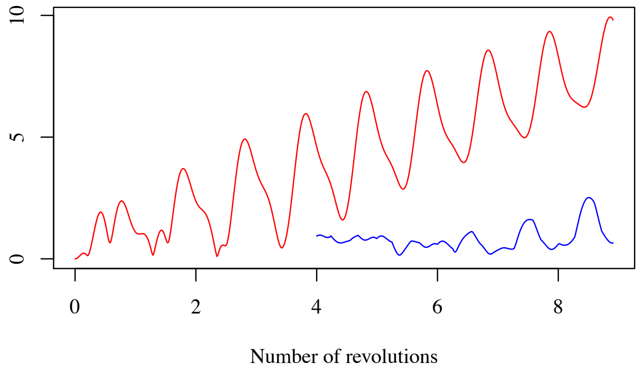
- **Hidden neurons**: 74.
- **Total number of weights & bias**: 127354.
- **Activation function**: Maxout.



5. Hybrid SGP4 for Galileo-type orbits

Inputs: 1720(1 rev.), **Training data**: 4 satellite revolutions, **Hidden layers**: 1.

- **Hidden neurons**: 74.
- **Total number of weights & bias**: 127354.
- **Activation function**: Maxout.

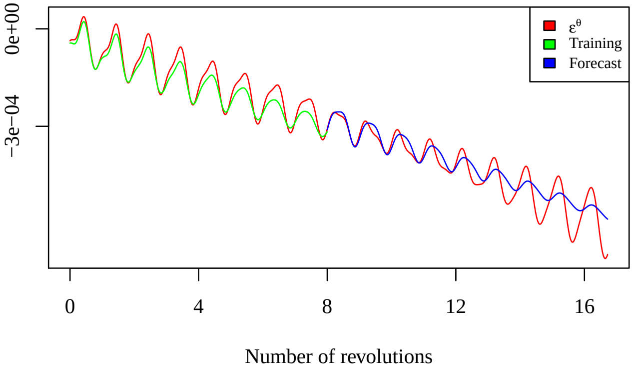


Distance error AIDA –SGP4 = 10 km and AIDA –HSGP4 = 2.5 km

5. Hybrid SGP4 for Galileo-type orbits

Inputs: 1720(1 rev.), **Training data**: 8 satellite revolutions, **Hidden layers**: 1.

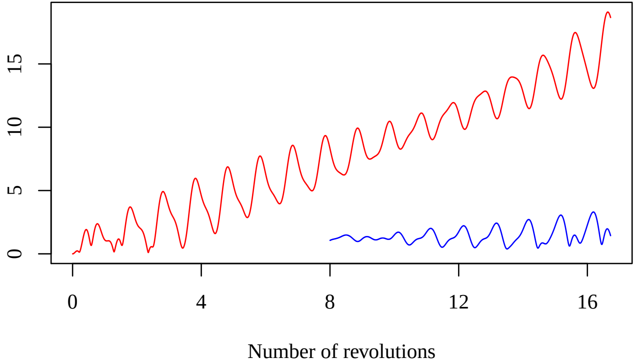
- **Hidden neurons**: 73.
- **Total number of weights & bias**: 125634.
- **Activation function**: Maxout.



5. Hybrid SGP4 for Galileo-type orbits

Inputs: 1720(1 rev.), **Training data**: 8 satellite revolutions, **Hidden layers**: 1.

- **Hidden neurons**: 73.
- **Total number of weights & bias**: 125634.
- **Activation function**: Maxout.



Distance error AIDA –SGP4 = 19.1 km and AIDA –HSGP4 = 3.3 km

5. Conclusions

1. Motivation
2. Orbit propagation problem
3. Hybrid propagation methodology
4. Forecasting technique: Neural networks
5. Hybrid SGP4 for Galileo-type orbits
6. **Conclusions**

5. Conclusions

1. New methodology: **Hybrid orbit propagators** (**HOPs**) are composed of a classical theory plus a forecasting technique.

2. The **forecasting technique** is developed from control data so as to complement the approximation generated by the classical theory by modeling and reproducing the missing dynamics.

3. **Neural networks** can be used as the forecasting technique.

    1. **Hyper-parameter optimization** is very important for finding accurate models.

    2. **Parsimonious models** only include the most relevant variables.

Thank you
for your attention