Discovering Unmodeled Components in Astrodynamics with Symbolic Regression

1st Matteo Manzi
Aerospace Centre of Excellence
University of Strathclyde
Glasgow, UK
matteo.manzi@strath.ac.uk

2nd Massimiliano Vasile
Aerospace Centre of Excellence
University of Strathclyde
Glasgow, UK
massimiliano.vasile@strath.ac.uk

***Abstract*—The paper explores the use of symbolic regression to discover missing parts of the dynamics of space objects from tracking data. The starting assumption is that the differential equations governing the motion of an observable object are incomplete and do not allow a correct prediction of the future state of that object. Symbolic regression, making use of Genetic Programming (GP), coupled with a sensitivity analysis-based parameter estimation, is proposed to reconstruct the missing parts of the dynamic equations from sparse measurements of position and velocity. Furthermore, the paper explores the effect of uncertainty in tracking measurements on the ability of GP to recover the correct structure of the dynamic equations. The paper presents a simple, yet representative, example of incomplete orbital dynamics to test the use of symbolic regression.**

***Index Terms*—Symbolic Regression, Genetic Programming, Astrodynamics**

I. Introduction

Learning how to effectively build dynamical system models from raw data is a challenge with a rich history, whose most famous example is probably the work conducted by Kepler[1]. Because of the vast quantities of available tracking data (both on celestial objects and man-made satellites), there is an increasing interest in the automated identification and refinement of dynamical systems (e.g., [1], [2]) and, in particular, in astrodynamics. Numerous neural-network based approaches, for the modelling of time-series data, exist (e.g., [3], [4]); these approaches however lack interpretability, being highly parameter-based. The introduction of empirical acceleration, in orbit determination, displays a similar drawback: it does not allow one to infer dependencies between physical quantities [5]. Because of these reasons, symbolic regression appears a viable solution to discover unmodeled components of dynamical systems from raw observations. Alternative approaches, based on sparse regression [6], have stricter requirements, in terms of *a priori* assumptions. In fact, symbolic regression can be coupled with Genetic Programming, in order to efficiently perform targeted searches in large domains. This work introduces a general formulation whereby model uncertainty (the unmodeled part of the dynamics) in orbital mechanics needs to be quantified from measurements and a GP-based symbolic regression method to deal with such an uncertainty. The paper explores the applicability of symbolic regression, based on a hybrid GP with local refinement, to a simple, but representative, problem in orbital mechanics where the dynamics is effected by an unknown dissipative effect that causes a decay of the orbit. This work is part of the development of an AI system for Space Situational Awareness and Space Environment Management [7], called CASSANDRA (Computer Agent for Space Situational Awareness aNd Debris Remediation Automation), which can be found here: https://github.com/strath-ace-labs/CASSANDRA.

The paper is structured as follows: after a first general introduction to the general formulation of the problem, the paper presents our proposed GP algorithm for symbolic regression and how it handles uncertainty in the observations. Then the case study is introduced with some preliminary results that illustrate the potentiality of symbolic regression and the major difficulty when only sparse and uncertain observations are available.

II. Space Dynamics

Consider the two functions $f : S \times P \times [t_0 : t_0 + T] \longrightarrow \mathbb{R}^n$ and $\upsilon : S \times B \times [t_0 : t_0 + T] \longrightarrow \mathbb{R}^n$ with $S \subseteq \mathbb{R}^n$ and the initial value problem:

$$\begin{cases} \dot{s} = f(s, p, t) + \upsilon(s, b, t) \\ s(t_0) = s_0 \end{cases} \tag{1}$$

where s is the state vector. The term $\upsilon(s, b, t)$ represents some unknown function of the state that is capturing all unmodeled components, $p \in P \subseteq \mathbb{R}^{m_p}$ a set of uncertain model parameters, $b \in B \subseteq \mathbb{R}^{m_b}$ some unknown parameter vector of the unmodeled components, and t the time coordinate. In this paper, we will study only the case in which the unmodeled components are not explicitly a function of time and the missing component is added to the known component.

A. Problem Statement

The idea is that, given a set of N_o state observations s_o at times t_i, with $i = 1, ..., N_o$, one can obtain an approximation of $\upsilon(s, b, t)$ by introducing different combinations of functions of s and values b into the dynamics (Eq. (1)), propagating the dynamics till time t_i and comparing the propagated state

[1] This is particularly clear in his 1609 book, whose full title can be translated as: “New astronomy, treated by means of commentaries on the motion of Mars, from observations of Tycho Brahe”.

978-1-7281-6929-3/20/$31.00 ©2020 IEEE

$s(t_i)$ to the observed state $s_o(t_i)$. If N_o exact and distinct measurements are available, one needs to solve the following set of constraints:

$$s(t_i) - s_o(t_i) = 0; \quad i = 1, ..., N_o \tag{2}$$

On the other hand, when noisy measurements are available, parameter distributions have to be estimated to infer the nominal state of the spacecraft associated to the observation epochs.

III. Symbolic Regression

Symbolic regression is the process of accurately matching a given data set, making use of symbolic expressions; because of the rise of Genetic Programming and Machine Learning, together with the growth of available data sets, attempts have been made to automate the regression process [8]. In the case of dynamical systems, and in particular in astrodynamics, one is interested in building a symbolic representation of the differential equation associated to a number of state observations. The advantage of this approach, compared to other regression techniques, like linear regression (and polynomial fitting in general: [5], [6], [9], Section 7.3 of [10]), is that not constraining the basis of the expression to a give set (even though assumptions have to be made when defining the library of primitive operators) allows to obtain a more insightful expression, whose relation with the dynamical process is explicit. While, with symbolic regression, it is still possible to make use of the knowledge of the sparsity of the expression, there is no need for an *a priori* specification of the terms of the equation [11]. At the same time, the sparse nature of the differential equations governing the motion of the spacecraft does not appear to require a multi-gene genetic programming approach [12].

When performing symbolic regression, candidate solutions are usually represented as trees, where the terminal nodes (also called "leaves") correspond to constants and variables, and where all the other nodes encode mathematical functions (primitives) [13]. In this work vectors, representing, in a given coordinate system, the position and the velocity of the spacecraft, have been used as terminators of the tree: this physics-inspired approach allows to build symbolic relations in multi-dimensional spaces.

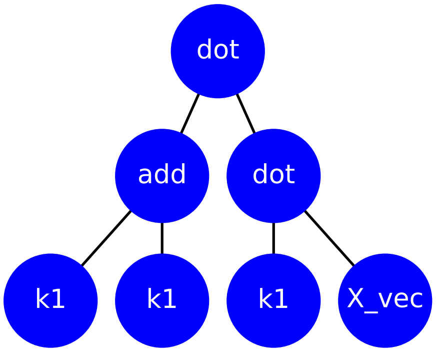


Fig. 1: Tree representation of: $(k_1 + k_1)(k_1 X_{vec})$

A. Genetic Programming

While symbolic regression can be performed using deterministic techniques (e.g., [14], [15]), this work focuses on the implementation of Genetic Programming. In order to do so, the DEAP (Distributed Evolutionary Algorithms in Python) library [16] has been selected as a starting point. This has been done because of the need to adapt existing evolutionary algorithms, in order to identify governing equations: DEAP allows for the customization of algorithms.

1) Primitives and terminals: The building blocks of each individual have been defined using Strongly Typed Genetic Programming (STGP): every primitive and terminal is assigned a specific type[2]; moreover, the output of the tree is forced to be a vector, whose size is given by the degrees of freedom of the problem under investigation. The terminals used are:

$$X_{vec}, V_{vec}, (k_1, \dots k_n)$$

symbolically representing the position and the velocity of the spacecraft in some reference system, together with a symbolic representation of a set of constant parameters: the introduction of the latter is justified in Section III-B. Of course, the choice of the reference system influences the complexity of the formulation of the equations of motion: it is however possible, as shown in [17], to develop methods discovering suitable coordinates and models at the same time. The primitives considered are:

- the sum between scalars;
- multiplication, constrained to take a scalar and a vector, as inputs;
- the Euclidean norm operator, returning a scalar from a vector;
- the exponential operator, working on scalars only;
- the user-defined functions c_m, returning the m^{th} component of the input vector;

2) Initialization and Selection: Before performing evolutionary operations, the population is initialized: this has been done using the *gp.genHalfAndHalf* function, producing trees with a maximum height of two. The selection process is then performed, at the beginning of each generation, with a tournaments of size two: such a low value is beneficial, in terms of population diversity.

3) Mutation and Crossover: The genetic operations of mutation and crossover are then performed in order to explore the domain of possible tree structures, constrained to a maximum height of six: this limit allows to avoid overfitting. Both operations are associated to a given probability, in order to be performed only on a subset of the population and allow the selection process to reach an optimum; the implemented mating operator is described in Algorithm 1.

[2] See the following, last visited on 28-01-2020: https://deap.readthedocs.io/en/master/tutorials/advanced/gp.html

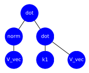
Algorithm 1 Random Mating Operator

```
Select two individuals $i_1$ and $i_2$
Randomly initialize $x$ in $[0, 1]$
if $x < 0.5$ then
    Perform the gp.cxOnePoint operation
else
    Perform the gp.cxOnePointLeafBiased operation
end if
```

The only mutation operation performed is the *gp.mutInsert*: inserts a new branch at a random position in the chosen individual. As also discussed later, describing Algorithm 2, this is the only mutation scheme we are interested in. In fact, an important hypothesis is that the *null individual* already leads to good results (i.e., the unmodeled components of the dynamics are small, compared to the modeled ones). We have here introduced such nomenclature, following the definition of the *null allele*: a mutated variant of a gene, not detectable solely from phenotypic observations[3]. This consideration allows to conjecture that the right combination of genetic operations can be able to build incrementally good approximations of the missing term: the initialization process can be thought of as a *gp.mutInsert* mutation, performed on the null individual.

B. Handling Real Model Parameters

Dealing with real-parameters when using Genetic Programming for symbolic regression remains a challenge. the DEAP library contains the *gp.mutEphemeral* function, allowing to mutate real-valued terminators, based on a pre-defined number generator: for the problem of interest, this does not allow to efficiently cover the domain. The same is true when using more specific mutation schemes, like the Gaussian and polynomial ones ([19], [20]). Of course, one could think of case-specific mutations: for example, a Gaussian mutation in which the standard deviation is expressed as a function of the set of fitnesses and associated parameters of those individuals, within a population, with the same structure, appears promising. Nevertheless, we here propose a white-box approach, displayed in Eq. (3), in which information can be extracted from the physical flow Φ.



$$\left[\;\bar{b} = [\bar{k}_1, \dots \bar{k}_n]\right] \longrightarrow v(s, b = \bar{b}, t) \xrightarrow{\Phi} s(t) \qquad (3)$$

Constant values are therefore treated symbolically in the evolutionary process, and unmodeled components of the dynamics can be estimated only when individuals are coupled with a set of values; the technique in which such values are determined is discussed in Section III-C1. Of course, this approach makes the evaluation of an individual more expensive, but the overall process is faster and, mostly, more accurate.

C. Individual Evaluation

1) Parameters Estimation: Since the algorithm symbolically represents also the parameters in the differential equation, in order to evaluate the fitness of an individual (i.e., a tree), the optimal set of values of the parameters has to be found. In order to do so, together with the state of the spacecraft, also the sensitivity matrix

$$\frac{\partial s}{\partial b}(s, p, b, t)$$

can be propagated, by means of the following initial value problem, obtained from Eq. 2.7 of [21]:

$$\begin{cases} \frac{\partial}{\partial t}\left(\frac{\partial s}{\partial b}\right) = \frac{\partial (f+v)}{\partial s}\frac{\partial s}{\partial b} + \frac{\partial v}{\partial b} \\ \frac{\partial s}{\partial b} = 0 \end{cases} \qquad (4)$$

The sensitivity matrix associated to the observation epochs can therefore be used to perform differential correction on the values of the unknown parameters b. This is done iteratively, as given in Algorithm 2, making use of the linear approximation

$$\Delta b = \Delta s / \left(\frac{\partial s}{\partial b}\right) \qquad (5)$$

where

$$\Delta s = s_o - s_{sym}$$

has the same size of b; this has been done considering only a subset of the available measurements.

Algorithm 2 Fitness Evaluation

```
 1: Initialize: $b = 0$; $||\Delta s|| = 1$; $c = 0$
 2: while $c < 100$ and $||\Delta s|| > 0.001$ do
 3:     Propagate Sensitivity Matrix
 4:     if $||\frac{\partial s}{\partial b}|| < \epsilon$ then
 5:         break
 6:     end if
 7:     $\Delta s = s_o - s_{sym}$
 8:     $\Delta b = \Delta s / \left(5\frac{\partial s}{\partial b}\right)$
 9:     $b = b + \Delta b$; $c = c + 1$
10: end while
11: Evaluate fitness
```

Concerning Algorithm 2, some additional comments can be made:

- the initial guess $b = 0$ is related to the hypothesis on the behaviour of the null individual underlined at the end of Section III-A: because of the incremental behaviour displayed by the method, the hypothesis keeps holding during the evolution, as long as the known function in Eq. (16) is updated with the new best-known approximation of the dynamical system under investigation;
- the condition on the norm of the sensitivity matrix (the *numpy* machine ϵ has been used for this) terminates the process, if convergence to a local minimum is reached;
- in line 8, a corrective factor is added to Eq. (5): this correction speeds up the convergence of the process, avoiding overshooting [22];

[3] For a clear understanding of the use of biology jargon in symbolic regression see, e.g., [18], [13].

- as said in Section II-A, depending on the number of parameters to be determined and on the number of available observations, the linear system in line 8 could be solved in a least-square sense or exactly; in the latter case, a subset of the available conditions could be enough to determine the value of vector Δb.

2) Fitness Evaluation: In order to perform the previously mentioned genetic operations, a fitness has to be associated to each symbolic expression. This is done propagating the state associated to the individual, where the value of the parameter vector b is the one estimated with differential correction. The fitness is therefore defined as

$$fit = \frac{1}{N_0} \sum_{a=1}^{N_0} \sum_{c=1}^{n/2} (s_o^c(t_a) - s_{sym}^c(t_a))^2 \tag{6}$$

where N_0 is again the number of observations available, $t_1, \ldots, t_{N_0}$ are the epochs associated to such observations and c is the component of the state vector s. The objective is therefore the *minimization* of the fitness function.

D. Treatment of Stochastic Observations

The observations used to reconstruct the missing part of the dynamics are in general sparse and affected by a degree of uncertainty. This uncertainty can be of different nature and generally has both an aleatory and an epistemic component.

As shown in [5], in the case of sparse observations affected by uncertainty there can be more than one dynamical model that is compatible with the measured positions and velocities. Thus, one should expect that both the set of parameters b and the structure of the equation reconstructed by GP are affected by the uncertainty in the measurements. If the expected values of the state vector, coming from observations, are enforced as hard constraints, the result might not capture the actual missing components as the trajectory is forced to satisfy constraints that do not come from the natural dynamics but are dependent on the errors in the observations.

One option is to consider the most probable value for each observation and a cost function that maximises the likelihood of correct identification. The other option is to quantify the uncertainty in the observations and initial conditions as confidence intervals on the observed states. If the distribution of s_o is available and is precise, one can draw N_p samples and for each sample derive a complete symbolic solution with associated set of coefficients b. Thus, in the following we will assume that the distributions of the initial states and the observed states at given times are perfectly known. We will then draw N_p samples and we will analyse the symbolic structures generated by GP for each sample. Then, for similar families of structures we will look at the distribution of the values of the associated parameter vector b.

IV. Orbital Motion with Unmodeled Drag

Following [5], a test case is here used to show the results obtained with the proposed method. Such problem is an orbital motion with unknown drag component. The gravity component of the model is fully known but the observations show an additional component that is not modelled. The real dynamics is assumed to be governed by the following system of differential equations in polar coordinates:

$$\begin{array}{l} \dot{v_r} = -\frac{\mu}{r^2} + \frac{v_t^2}{r} - \frac{1}{2}\rho C_d v v_r \\ \dot{v_t} = -\frac{v_t v_r}{r} - \frac{1}{2}\rho C_d v v_t \\ \dot{r} = v_r \\ \dot{\theta} = \frac{v_t}{r} \end{array} \tag{7}$$

We assume a unitary area-to-mass ratio, and a constant density ρ such that the product of the density times the drag coefficient C_d is $\rho C_d = 10^{-7}\ kg/km^3$. Furthermore, we assume that the expected trajectory, given the known dynamic components, is a circular orbit with $v_r(t=0) = v_{r_0} = 0$ and $v_t(t=0) = v_{t_0}$. The orbital period, without drag, is $T = 2\pi\sqrt{r^3/\mu}$. The evolution of the state of the spacecraft is displayed, for each component, in the four plots of Figure 2: two observations per revolution are performed at $t \in [0, 3500, 6870, 9999]\ s$.

In order to investigate the performance of the proposed technique, the estimated state of the spacecraft at time $t = 4T$ will be used. In the nominal case, this is equal to:

$$s(4T) = \begin{bmatrix} r \\ \theta \\ v_r \\ v_t \end{bmatrix} = \begin{bmatrix} 6855.7014 & km \\ 25.9035 & rad \\ -65.8 & m/s \\ 7.6 & m/s \end{bmatrix} \tag{8}$$

A. Exact Observations

The first results follow from considering noiseless, exact state measurements. The symbolic regression is able to reconstruct the exact form of the missing term (Figure 3) in the differential equation:

$$\dot{s} = f(s,p,t) - \frac{1}{2}\rho C_d v \begin{bmatrix} v_r \\ v_t \end{bmatrix} = f(s,p,t) + k_1 v \begin{bmatrix} v_r \\ v_t \end{bmatrix} \tag{9}$$

Associated to such structure, the parameter estimation process leads to

$$k_1 = -5.0124 \cdot 10^{-8}\ kg/km$$

with an error of approximately 0.2%.

It should be underlined how the hypothesis on the incremental accuracy of the model is supported by experimental results:

1) at first, the algorithm produces random structures, and the best individuals of the population are the ones in which a constant factor, set to 0, allows to set the whole expression to 0;
2) the procedure then expresses the missing term as $k[v_r, v_t]$, approximating the norm of the velocity vector as a constant. Indeed, since the initial orbit is circular, this is a good approximation;
3) finally, the algorithm converges to the exact structure of the unmodeled component of the dynamical system.

The fitness of the obtained solution is

$$fit = 8.713 \cdot 10^{-3}\quad km^2$$

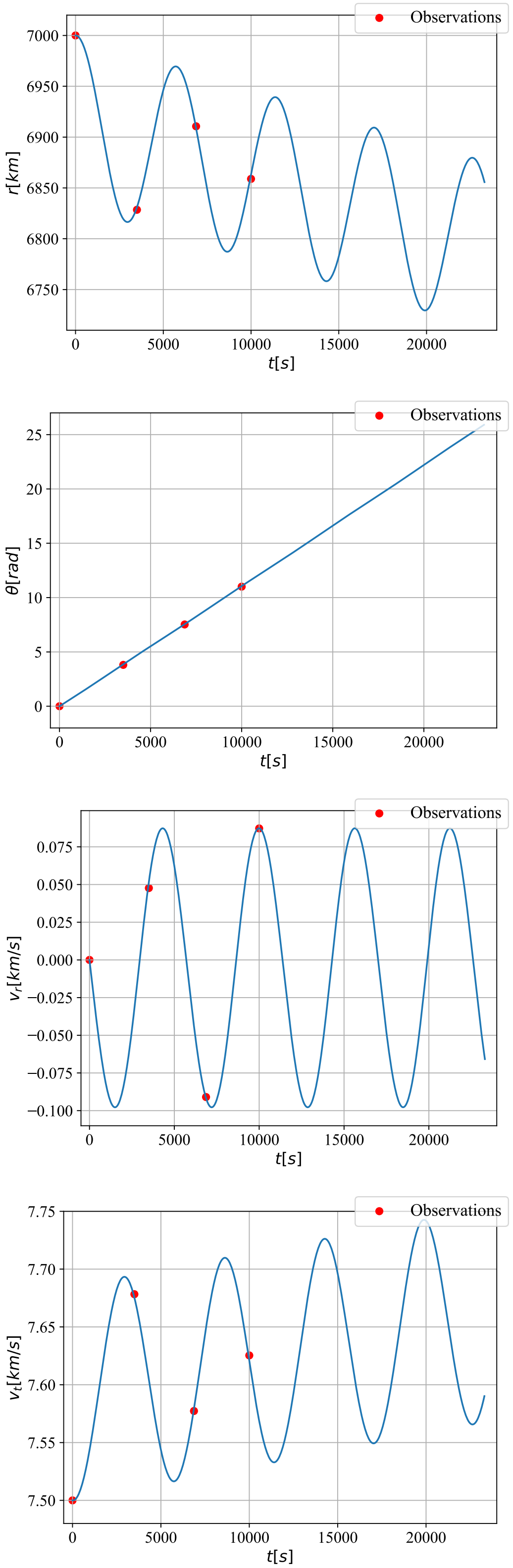


Fig. 2: State evolution of a spacecraft in Low Earth Orbit, for $t \in [0, 4T]$, and associated *exact* measurements.

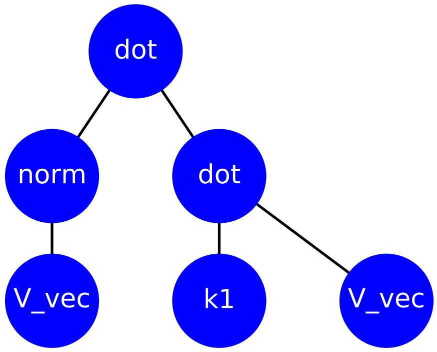


Fig. 3: Best individual structure

and the state of the spacecraft, at $t = 4T$, is:

$$s(4T) = \begin{bmatrix} r \\ \theta \\ v_r \\ v_t \end{bmatrix} = \begin{bmatrix} 6855.3582 & km \\ 25.9043 & rad \\ -65.9 & m/s \\ 7.59 & m/s \end{bmatrix} \tag{10}$$

In general, one would expect different governing equations to match a finite set of observations (even when these are exact); this issue, introduced in Section III-D and related to overfitting, is avoided taking into account the sparsity of the differential equation which, in the context of symbolic regression, limits the size of its tree representation.

B. Noisy Observations

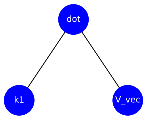
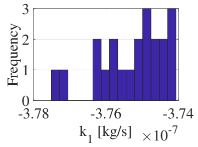
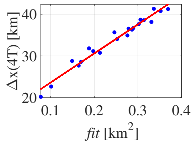
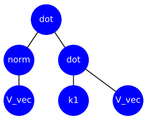
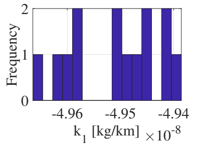
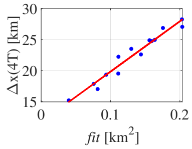
With the same spirit of [5] and [13], noise, modelled as zero-mean Gaussian distributions, has been added on top of ideal observations to investigate the robustness of the technique with respect to different tracking techniques [23] and understand its effect on the parameter estimation routine; because of this, three different noise levels have been considered and, for each of them, 35 simulations have been performed. The results given in Table I, Table II and Table III show:

- the estimated structures of the solution, identifying the unmodeled component of Eq. (1);
- for each family of solutions, the distribution of the associated parameter;
- a graph to investigate the relation between the value of the fitness, associated to the available measurements, and of $\Delta x(t = 4T)$. The correlation between these two quantities is a performance indicator of the symbolic regression extrapolation.

1) Noise level 1: In the first case, the standard deviation associated to the measurements is given by:

$$\begin{bmatrix} \sigma_r \\ \sigma_\theta \\ \sigma_{v_r} \\ \sigma_{v_t} \end{bmatrix} = \begin{bmatrix} 100 & m \\ 0 & rad \\ 1 & m/s \\ 1 & m/s \end{bmatrix} \tag{11}$$

TABLE I: Solutions - Noise level 1

Structure	Parameter	Extrapolation
		
		

While 14 simulations converge to the right structure of Eq. (9), 21 estimate the motion to be governed by:

$$\dot{s} = f(s,p,t) + k_1 \begin{bmatrix} v_r \\ v_t \end{bmatrix} \qquad (12)$$

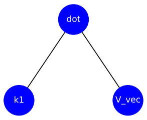
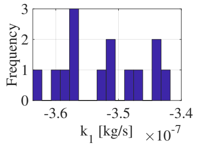
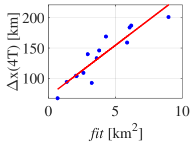
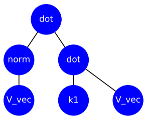
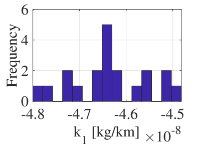
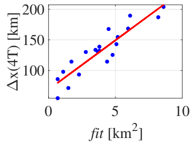
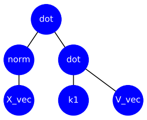
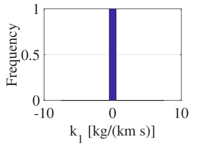
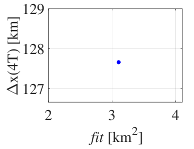
Both families of solutions nevertheless display comparable fitness values with an high correlation with the final position offset; as previously stated, this is due to the fact that the initial orbit is circular. It is not clear whether or not a longer simulation (i.e., more generations) would have allowed these solutions to evolve into the correct ones.

2) Noise level 2: In the second case, measurement errors have been modelled using:

$$\begin{bmatrix} \sigma_r \\ \sigma_\theta \\ \sigma_{v_r} \\ \sigma_{v_t} \end{bmatrix} = \begin{bmatrix} 1 & km \\ 0 & rad \\ 10 & m/s \\ 10 & m/s \end{bmatrix} \qquad (13)$$

The two main solution families are the same as above; the parameters are more widely distributed and the correlation between the fitness and the final state is lower, because of higher noise. It is finally interesting to note how one of the simulations converged to the null individual.

TABLE II: Solutions - Noise level 2

Structure	Parameter	Extrapolation
		
		
		

3) Noise level 3: For the final case, the standard deviation associated to the measurements is given by:

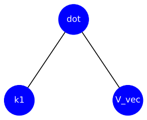
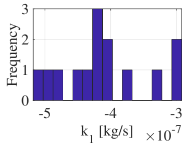
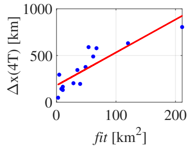
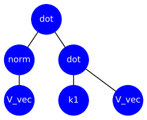
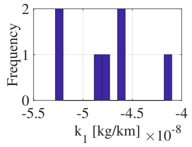
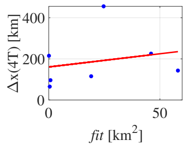
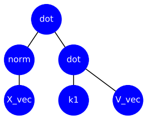
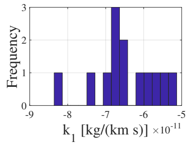
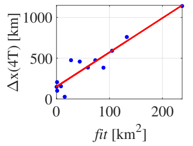
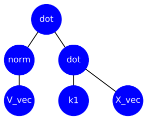
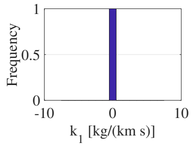
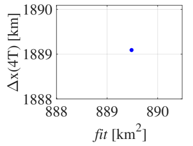
$$\begin{bmatrix} \sigma_r \\ \sigma_\theta \\ \sigma_{v_r} \\ \sigma_{v_t} \end{bmatrix} = \begin{bmatrix} 10 & km \\ 0 & rad \\ 100 & m/s \\ 100 & m/s \end{bmatrix} \qquad (14)$$

With such a level of noise, a new non-trivial family of solutions arises:

$$\dot{s} = f(s,p,t) + k_1 r \begin{bmatrix} v_r \\ v_t \end{bmatrix} \qquad (15)$$

Solutions associated to the correct differential equation outperform all the others, in extrapolating the spacecraft position, even if their fitness is not necessarily lower.

TABLE III: Solutions - Noise level 3

Structure	Parameter	Extrapolation
		
		
		
		

V. Final Remarks

The paper presented a simple, yet representative, case in which missing components of the dynamics of a space objects are discovered using symbolic regression. It has been shown how a combination of symbolic regression with local sensitivity analysis is effective at capturing both the structure of the equations and the values of the model parameters. One of the key findings is that, when observations are affected by uncertainty, different families of dynamic equations can be identified and for each of them the model parameters follow a probability distribution that is a function of the probability distribution of the observations.

Future works will consider more complex forms of uncertainty in the observations and more complex dynamic structures of the perturbation, whose modelling will be enabled by including more primitives. It will incorporate also e multiplicative unknown term in the dynamics, leading to the following more general formulation:

$$\begin{cases} \dot{s} = \upsilon_2(s,b,t) f(s,p,t) + \upsilon_1(s,b,t) \\ s(t_0) = s_0 \end{cases} \tag{16}$$

Finally, the development of a technique to discover coordinate systems in which the dynamics can be sparsely represented, using symbolic regression, appears desirable, as mentioned in Section III-A1.

Acknowledgment

This research is supported by the EU H2020 MSCA ITN Stardust-R, grant agreement 813644.

References

[1] J. Bongard and H. Lipson, “Automated reverse engineering of nonlinear dynamical systems,” *Proceedings of the National Academy of Sciences*, vol. 104, no. 24, pp. 9943–9948, 2007.

[2] M. Schmidt and H. Lipson, “Distilling Free-Form Natural Laws from Experimental Data,” *Science*, vol. 324, no. 5923, pp. 81–85, 2009.

[3] P. R. Vlachas, W. Byeon, Z. Y. Wan, T. P. Sapsis, and P. Koumoutsakos, “Data-driven forecasting of high-dimensional chaotic systems with long short-term memory networks,” *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 474, no. 2213, 2018.

[4] M. Raissi, P. Perdikaris, and G. E. Karniadakis, “Multistep neural networks for data-driven discovery of nonlinear dynamical systems,” *arXiv: Dynamical Systems*, 2018.

[5] M. Vasile, “Polynomial representation of model uncertainty in dynamical systems,” *Computational Methods in Applied Sciences*, pp. 419–432, 2019.

[6] S. Brunton, J. Proctor, and J. Kutz, “Discovering governing equations from data: Sparse identification of nonlinear dynamical systems,” *Proceedings of the National Academy of Sciences*, vol. 113, p. 3932–3937, 2015.

[7] T. Maclay and D. McKnight, “Space environment management: Framing the objective and setting priorities for controlling orbital debris risk,” *70th International Astronautical Congress (IAC), Washington D.C., United States*, 2019.

[8] S.-M. Udrescu and M. Tegmark, “AI Feynman: A physics-inspired method for symbolic regression,” *Science Advances*, vol. 6, no. 16, 2020.

[9] K. Kaheman, E. Kaiser, B. Strom, J. N. Kutz, and S. L. Brunton, “Learning discrepancy models from experimental data,” *ArXiv*, vol. abs/1909.08574, 2019.

[10] S. L. Brunton and J. N. Kutz, *Data-Driven Science and Engineering: Machine Learning, Dynamical Systems, and Control*. Cambridge University Press, 2019.

[11] S. Atkinson, W. Subber, L. Wang, G. Khan, P. F. Hawi, and R. Ghanem, “Data-driven discovery of free-form governing differential equations,” *ArXiv*, vol. abs/1910.05117, 2019.

[12] A. H. Gandomi and A. H. Alavi, “A new multi-gene genetic programming approach to nonlinear system modeling. Part I: materials and structural engineering problems,” *Neural Computing and Applications*, pp. 171–187, 2012.

[13] S. Gaucel, M. Keijzer, E. Lutton, and A. Tonda, “Learning Dynamical Systems Using Standard Symbolic Regression,” in *Genetic Programming*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2014, pp. 25–36.

[14] T. Mcconaghy, “FFX: Fast, Scalable, Deterministic Symbolic Regression Technology,” *Genetic Programming Theory and Practice IX*, pp. 235–260, 2011.

[15] H. Vaddireddy and O. San, “Equation Discovery Using Fast Function Extraction: a Deterministic Symbolic Regression Approach,” *Fluids*, 2019.

[16] F.-A. Fortin, F.-M. De Rainville, M.-A. Gardner, M. Parizeau, and C. Gagné, “DEAP: Evolutionary algorithms made easy,” *Journal of Machine Learning Research*, pp. 2171–2175, 2012.

[17] K. Champion, B. Lusch, J. N. Kutz, and S. L. Brunton, “Data-driven discovery of coordinates and governing equations,” *Proceedings of the National Academy of Sciences*, vol. 116, no. 45, pp. 22 445–22 451, 2019.

[18] H. Iba, “Evolutionary Approach to Machine Learning and Deep Neural Networks,” *Neuro-Evolution and Gene Regulatory Networks*, 2018.

[19] R. Cazacu, “Comparative study between the improved implementation of 3 classic mutation operators for genetic algorithms,” *Procedia Engineering*, pp. 634 – 640, 2017, 10th International Conference Interdisciplinarity in Engineering.

[20] K. Deb and D. Deb, “Analysing mutation schemes for real-parameter genetic algorithms,” *International Journal of Artificial Intelligence and Soft Computing*, pp. 1–28, 2014.

[21] A. Milani and G. Gronchi, *Theory of Orbit Determination*. Cambridge University Press, 2009.

[22] R. S. Varga, *Matrix Iterative Analysis*. Springer, 2009.

[23] B. Tapley, B. Schutz, and G. Born, “Chapter 3 - Observations,” in *Statistical Orbit Determination*. Burlington: Academic Press, 2004, pp. 93–158.