Genet Program Evolvable Mach (2013) 14:155–190
DOI 10.1007/s10710-012-9175-4

# Inference of hidden variables in systems of differential equations with genetic programming

**Theodore W. Cornforth · Hod Lipson**

Received: 25 August 2012 / Revised: 31 October 2012 / Published online: 13 November 2012
© Springer Science+Business Media New York 2012

**Abstract** The data-driven modeling of dynamical systems is an important scientific activity, and many studies have applied genetic programming (GP) to the task of automatically constructing such models in the form of systems of ordinary differential equations (ODEs). These previous studies assumed that data measurements were available for all variables in the system, whereas in real-world settings, it is typically the case that one or more variables are unmeasured or "hidden." Here, we investigate the prospect of automatically constructing ODE models of dynamical systems from time series data with GP in the presence of hidden variables. Several examples with both synthetic and physical systems demonstrate the unique challenges of this problem and the circumstances under which it is possible to reverse-engineer both the form and parameters of ODE models with hidden variables.

**Keywords** Genetic programming · Ordinary differential equations · Hidden variables · Modeling · Symbolic identification

## 1 Introduction

Models that organize, explain and predict empirical observations play a central role in science. The form taken by a model is problem-specific and must be designed by a human expert, usually at the expense of considerable time and effort. In contrast,

T. W. Cornforth (✉)
Biological Statistics and Computational Biology, Cornell University, 1198 Comstock Hall, Ithaca, NY 14853-2601, USA
e-mail: twc63@cornell.edu

H. Lipson
Mechanical and Aerospace Engineering, Cornell University, 239 Upson Hall, Ithaca, NY 14853-7501, USA
e-mail: hl274@cornell.edu

the rate at which scientists are able to automatically generate empirical data is expected to steadily increase for the foreseeable future, as is the complexity of problems routinely studied in all fields [26, 75, 76]. As a result, there is a widening gap between our ability to generate data and our ability to distill data into useful scientific models. New computational methods will be necessary to close this gap by assisting humans in the creation of scientific models [30, 44, 82].

Genetic programming (GP) and related machine learning techniques based on evolutionary computation have been proposed as a means of automatically creating models with minimal human effort and background knowledge [13, 46]. Traditional modeling techniques such as nonlinear regression and Hidden Markov models typically require the model to have a generic structure substantially specified in advance by a domain expert. Other techniques such as neural networks and autoregressive models are predictive of unseen data but can be difficult to interpret. In contrast, GP places minimal constraints on model structure, requires little or no domain knowledge, and is capable of automatically inferring parsimonious explanatory models such as those that would ordinarily be hand-crafted from first principles by a human expert [68]. Such models are particularly valuable as they explicitly represent scientific knowledge and hypotheses as interpretable analytical expressions [71].

GP has been successfully applied to many symbolic modeling problems, including those involving data with measurements that vary over time [42, 54–56, 62, 68]. The models of these dynamical systems are in many cases most naturally represented as a system of ordinary differential equations (ODEs). In fact, several previous studies have used GP and closely related techniques to infer ODEs based on experimental data and to garner insight into the dynamical systems under study [6, 7, 11, 15, 18, 46]. Many authors have also combined GP for structural optimization with local search algorithms for improved estimation of parameters in the context of ODE model inference [4, 5, 23, 33, 34, 39, 40, 60, 66, 86].

It is often the case that dynamical systems encountered in the real-world can be modeled with systems of differential equations in which one or more of the dynamic variables are unmeasured or "hidden." As an example, consider the ideal rocket equations, which describe the basic Newtonian mechanics of rocket motion. When written explicitly as a system of ordinary differential equations, these rocket equations take the form:

$$\begin{cases} \frac{dX}{dt} = V \\ \frac{dV}{dt} = \frac{mV_e}{M} \\ \frac{dM}{dt} = -m \end{cases} \tag{1}$$

Here, $X$ is the position of the rocket at time $t$, $V$ is the velocity of the rocket at time $t$, $M$ is the mass of the rocket at time $t$, $m$ is the (constant) rate at which mass is lost by the rocket due to fuel expenditure, and $V_e$ is the (constant) velocity of the mass lost by the rocket. The position, and by extension, velocity of the rocket are easy to measure by an outside observer, whereas the mass of the rocket cannot be measured directly. However, it is essential to include the concept of mass for accurately and parsimoniously describing the mechanics of rocket motion, and so the mass is a

hidden variable. Hidden variables are often a component of models such as the rocket equations that provide an explanation of the modeled phenomenon. This is fundamentally different from a model such as an autoregressive polynomial that would be merely descriptive of the observed data [22].

Hidden variables frequently arise in real-world settings in all fields of science, usually when it is difficult or impossible to measure one or more of the system's dynamic variables. A classic example from biology is the set of ion channel gating variables in the Hodgkin-Huxley ODE system describing action potentials [37]. Additional examples include the latent prey subpopulations in predator–prey systems with rapid evolution [87], the concentration of several of the species in a large chemical reaction network [25], and the components of a dynamical model of DC motors [52].

A practical system that uses GP to automatically construct ODE models of real-world dynamical systems must take into account the presence of hidden variables. However, to date no previous study has addressed this issue, possibly because hidden variables present several challenges to automated modeling that are not present when all variables in the system are measured. Chief among these challenges is the inherent difficulty of identifying unique ODE models containing hidden variables that match given experimental observations. In the present work we explore the possibility of obtaining meaningful models despite this and other difficulties. Using examples from both synthetic and physical dynamical systems, we show that under many circumstances, it is possible to automatically construct both the form and parameters of ODE models with hidden variables using GP.

## 2 Related work

The inference of ODE models both with and without hidden variables has a long history. Here we briefly review this work and contrast it with the GP-based approach used in this paper. Although no single review article encompasses all work in this area, comprehensive reviews covering much of the area include Sjöberg et al. [73], Voss et al. [80], Aguirre and Letellier [3], and Hong et al. [38].

Much of the previous work in the inference of ODE models has been undertaken in the field of linear and nonlinear system identification [9, 10, 50]. Similarly to traditional regression, most system identification approaches to ODE inference involve the optimization of parameter values to match observed data while assuming the structure of the model is fixed in advance. These structures are typically polynomials or Taylor series [27, 31]. More recent system identification approaches often do not assume a fixed structure and instead use structure selection techniques, although these are still restricted to combinations of generic basis functions or do not involve ODEs [2, 38, 84].

Although work in the area of system identification frequently assumes that all dynamic variables are measured, even if only in noise-corrupted form, a significant body of work has also addressed dynamical systems with hidden variables. These techniques, often categorized as belonging to the area of nonlinear dynamics and chaos, are generally based on the observation that a complete phase space

representation of a system with hidden variables can be reconstructed using only one or a few observed variables [57, 77]. The reconstructed phase space is qualitatively similar to that of the true unknown system and previous studies have taken advantage of that fact to construct approximate models of systems with hidden variables using ODEs [21] as well as other model formulations [12, 28]. The trajectories of hidden variables obtained with phase space reconstruction techniques are not equivalent to those that would be obtained by measuring them directly in an experiment. As result, we do not use these trajectories as targets for GP in this work, where the goal is to reverse-engineer the exact analytical model of a system. However, it is possible that information obtained from the phase space reconstruction of a system could be incorporated into our approach here. For example, the search for a hidden variable expression could be seeded with the results of symbolic regression performed with the trajectories obtained from phase space reconstruction as targets.

Another significant body of work on ODE inference can be broadly categorized as knowledge-based, in contrast to the type of free-form modeling possible with GP. While both approaches are essentially a search through the space of possible model structures and parameter values, knowledge-based approaches use human-supplied domain knowledge to constrain the search space [1, 19, 29, 47, 48, 78, 83]. An example of this approach is inductive process modeling, in which differential equation models for continuous time data are constructed using predefined algebraic expressions and other constraints that represent expert domain knowledge [22, 49]. This and similar knowledge-based approaches have been successfully used to construct accurate and interpretable ODE models, even when hidden variables are present. However, a serious potential drawback of all knowledge-based approaches is that substantial domain knowledge must be supplied in the form of predefined model structures. For example, in a population dynamics modeling task, inductive process modeling as described in Bridewell et al. [22] exhaustively enumerates all possible combinations of one of a few different algebraic structures that each correspond to known models of population growth and decline.

Knowledge-based modeling is a valid approach that has achieved notable success in a variety of problem domains, yet it requires significant human involvement in the modeling process to specify domain knowledge and supply constraints. In light of the explosion of data in all scientific fields, our interest is in automated free-form modeling approaches that require no human-supplied domain knowledge aside from the specification of a set of algebraic primitives such as $+$, $-$, sin, cos, etc. Both the structure and parameters of the resulting model are constructed de novo from the primitives. Free-form approaches, including the GP-based approach used in the present work, rely on computationally-intensive stochastic searches through model space, yet recent studies have successfully induced complex ODE models and other natural laws without reliance on domain knowledge [18, 68].

GP and other free-form approaches have not been applied to the modeling of systems with hidden variables in any prior work. The motivation of the work presented here is to take the first steps toward combining GP's freedom from supplied domain knowledge with the capability to model ODEs with hidden

variables. This combination could extend the power of machine learning to assist scientists in the task of translating raw data into meaningful analytical models.

## 3 Methods

### 3.1 Problem statement

In the general case with no hidden variables, we consider dynamical systems empirically described by $n$ time series $y_1(t)$, $y_2(t)$, …, $y_n(t)$, where each time series consists of $m$ observations. The observed data $Y$ from such a system can be written as a $n \times m$ matrix:

$$Y = \begin{pmatrix} y_1(t_0), y_1(t_1), \ldots, y_1(t_{m-1}) \\ y_2(t_0), y_2(t_1), \ldots, y_2(t_{m-1}) \\ \vdots \\ y_n(t_0), y_n(t_1), \ldots, y_n(t_{m-1}) \end{pmatrix} \tag{2}$$

where $y_i(t_j)$, $(i = 1, 2, \ldots, n)$, $(j = 0, 1, \ldots, m-1)$ is the observed value of variable $y_i$ at time $t_j$. The dynamical system can be modeled as a system of ODEs $X$ with the general form:

$$X = \begin{cases} \frac{dX_1}{dt} = f_1(t, X_1, X_2, \ldots, X_n) \\ \frac{dX_2}{dt} = f_2(t, X_1, X_2, \ldots, X_n) \\ \vdots \\ \frac{dX_n}{dt} = f_n(t, X_1, X_2, \ldots, X_n) \end{cases} \tag{3}$$

where $X_i$ is the $i$th state variable and $f_i$ is an arbitrary function of the $n$ state variables and time $t$. Together with initial values for each of the state variables $\hat{y}_1(0), \hat{y}_2(0), \ldots, \hat{y}_n(0)$, system of ODEs $X$ can be solved with numerical integration techniques to yield the predicted time series data. Written as a $n \times m$ matrix as with $Y$ above, the predicted data $\hat{Y}$ is:

$$\hat{Y} = \begin{pmatrix} \hat{y}_1(t_0), \hat{y}_1(t_1), \ldots, \hat{y}_1(t_{m-1}) \\ \hat{y}_2(t_0), \hat{y}_2(t_1), \ldots, \hat{y}_2(t_{m-1}) \\ \vdots \\ \hat{y}_n(t_0), \hat{y}_n(t_1), \ldots, \hat{y}_n(t_{m-1}) \end{pmatrix} \tag{4}$$

To measure the accuracy of system of ODEs $X$ as a model of the observed data $Y$, we use the scaled mean absolute error:

$$\text{error (all variables measured)} = \frac{1}{mn} \sum_{i=1}^{n} \left( \frac{\sum_{j=0}^{m-1} \left| Y_{ij} - \hat{Y}_{ij} \right|}{\sigma_i} \right) \tag{5}$$

where $\sigma_i$ is the standard deviation of the $i$th observed time series. This scaling factor is used so that error values can be meaningfully compared for any studied dynamical

system. In the present work we are concerned with the "inverse problem" of identifying the form and parameters of system of ODEs $X$ so as to minimize this prediction error.

When inferring an ODE model with hidden variables, one or more of the time series $y_i(t)$ are not measured and are unavailable to guide the model inference process. In this case, the inference process is instead guided by the prediction error calculated solely on the basis of the measured variables. Out of the $n$ total variables in the system, a subset of size $o$ variables are measured or observed and $(n - o)$ variables are unmeasured or hidden. To calculate the error of a system of ODEs $X$ in this case, we use the scaled mean absolute error as above except that only measured variables are involved in the error calculation:

$$\text{error} = \frac{1}{mo} \sum_{i=1}^{o} \left( \frac{\sum_{j=0}^{m-1} \left| Z_{ij} - \hat{Z}_{ij} \right|}{\sigma_i} \right) \tag{6}$$

Here, $\sigma_i$ is the standard deviation of the $i$th observed time series and $Z$ is a matrix equal to $Y$ except that rows corresponding to hidden variables are omitted. Similarly, $\hat{Z}$ is a matrix equal to $\hat{Y}$ except that rows corresponding to hidden variables are omitted. Note that numerical integration of candidate ODE model $X$ with one or more hidden variables always produces $n$ predicted time series; the predicted time series corresponding to hidden variables are simply not used to calculate the error of $X$.

In some experiments below, noise was added to the observed time series. For each predicted data point, a random variate is generated from a Gaussian distribution with mean 0 and standard deviation equal to 0.01 times the standard deviation of the observed time series. This noise was also added to the initial values of the observed time series that are used as the starting points for numerical integration of candidate ODE models, thereby making evaluation of models considerably more difficult than in a standard symbolic regression task [18, 46]. Although sophisticated techniques such as unscented Kalman filtering exist for filtering out this kind of noise [41], our intention was to test the robustness of model inference to reasonable amounts of measurement noise. When several GP trials were performed, noise was added independently for each trial.

### 3.2 GP algorithm for ODE model inference

We use a variation on Koza's tree-based GP to identify both the form and parameters of ODEs [46]. Individuals are encoded as a forest of $n$ binary trees, where $n$ is the number of equations and dynamic variables in the system of ODEs. For example, the forest of two trees shown in Fig. 1 corresponds to the system of ODEs:

$$\begin{cases} \frac{dX_1}{dt} = X_1^2 + \sin(X_2) \\ \frac{dX_2}{dt} = \frac{1.75}{(X_1 - X_2)} \end{cases} \tag{7}$$

Here, we consider autonomous ODEs and each tree is composed of terminals from the set $\{a, X_1, X_2, \ldots, X_n\}$ where $X_i$ is the $i$th state variable in a system of ODEs with

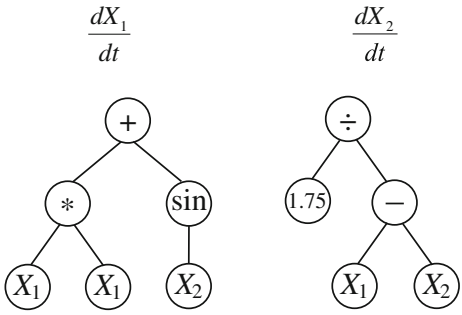



**Fig. 1** The system of ODEs given in Eq. (7) encoded as a forest of two binary trees

$n$ variables and $a$ represents ephemeral random constants in the range $(-9.0, 9.0)$. For all examples except Example 5 below, the set of possible arithmetic operations used in the binary trees was $\{+, -, *, /\}$, and for Example 5 this set was augmented to include trigonometric functions: $\{+, -, *, /, \sin, \cos\}$. Along with the trees representing individual ODEs, each individual is composed of a set of initial values $\hat{y}_1(0), \hat{y}_2(0), \ldots, \hat{y}_n(0)$, one for each of the $n$ equations in the system. For observed variables, these values are simply fixed to the known initial values obtained from the observed time series data, whereas for hidden variables, the values are optimized as if they were trees containing a single ephemeral random constant node. The maximum allowed depth of the trees was five. The population of 96 trees was initialized by randomly adding nodes to the leaves of an initially empty tree until a pre-chosen depth was reached. This depth was chosen randomly from the range (1, 5), with each possible depth equally likely.

As in standard GP, crossover and mutation were used to generate novel individuals. Each generation, two parents from the population of 96 were selected at random. With 90 % probability, they generated two offspring by subtree crossover with internal nodes selected as crossover points 90 % of the time and terminals selected 10 % of the time [46]. With 5 % probability, the parents generated two offspring by each undergoing mutation in which a node in the tree was randomly selected and replaced by a new randomly generated subtree. Finally, with 5 % probability, the two parents generated two offspring without modification. No recombination or mutation to a node of a different type was allowed for the trees in the forest that represented initial values. Reproduction continued into this manner until 96 children were generated, at which point the errors of the offspring trees were evaluated and survival selection took place as described below.

Recent work has shown that using genotypic age as an explicit objective in addition to fitness (or error) can be a powerful yet simple way to reduce bloat and maintain diversity in an evolving GP population [69, 70]. Here, we define age as the number of generations the individual has been present in the population, which is to be minimized. An individual descended from another individual, whether by mutation or recombination, inherited that parent's age. In the case of recombination, the age of the older parent was inherited. To provide a steady flow of low-age genotypes, one new individual with an age of 0 was added to the population each

generation. This new individual was randomly constructed in the same way as all 96 members of the initial population.

In contrast to standard GP, we use genotypic age to perform survival selection with the Age-Fitness Pareto algorithm [69, 70]. With this algorithm, the minimization of genotypic age is used as a second objective in addition to the minimization of error, making this a multiobjective optimization problem. Survival selection with elitism is implemented in this multiobjective setting with a Pareto tournament of size 2, in which individuals from the combined population of parents and offspring are randomly selected to participate. If either of the pair has both higher error and higher age than the other, it is discarded. This continues until the desired population size of 96 is reached, at which point reproduction takes place. In the event that a large number of tournaments failed to discard any individuals, individuals were randomly discarded until the desired population size was reached, although this never occurred in practice.

Many authors have noted that standard GP can have difficulty evolving floating point constants effectively. The solutions proposed generally involve embedding local search algorithms or other optimization techniques within a standard GP algorithm [39]. We use the simple strategy of hill-climbing as a local search technique [64]. Following Schmidt and Lipson [68], each generation, one hill-climbing step is performed in which each constant in each individual is perturbed with a small amount of 0 mean Gaussian noise. The standard deviation of the noise was set to 0.1 times the absolute value of the constant. The error of each individual is then reevaluated and if error increases, the old parameter values are restored.

As in all machine learning, the search for symbolic models with GP involves an inherent tradeoff between model complexity and accuracy [16]. The most accurate models, or those with lowest error on the training data set, tend to be unnecessarily complex and overfit the data in the sense that they perform poorly on unseen test data [53, 66]. For this reason, tracking summary statistics such as the training error of the best model over the course of the evolutionary search can be misleading. This is especially the case in the present study as we are interested primarily in recovering the exact structure of synthetic systems and in inferring parsimonious, explanatory models of physical systems. In light of this goal and instead of recording only the model with lowest error on the training or test data, throughout an evolutionary run we track a Pareto front of models that represent the best-so-far tradeoffs between model complexity and training error. After the completion of a trial, the models in this Pareto hall of fame are taken as the results of model evolution. Model complexity was measured simply as the sum of the number of terminals and operations used in all equations of the system of ODEs. For example, the model in Eq. (7) and depicted in Fig. 1 has a complexity of 11, with 6 terminals and operations in the first equation of the system and 5 terminals and operations in the second equation of the system.

### 3.3 Fitness evaluation

Instead of maximizing the fitness of evolved models, the goal of evolution in our GP algorithm is to minimize the error as defined in Sect. 3.1 above. For the error

calculation, we find the predicted time series by numerically integrating the system of ODEs encoded by an individual using the common fourth-order Runge–Kutta method with fixed step size [59]. In most of the examples below, we compared three different variations on this basic means of evaluating candidate models. In the first method, each candidate model was integrated for 128 time steps and all 128 resulting predicted data points were compared against the 128 corresponding observed data points using Eq. (6). This method is referred to as "128 All" below. In the second method, each candidate model was integrated for only 8 time steps and the 8 resulting predicted data points were compared against the 8 corresponding observed data points. This method is referred to as "8 All" below.

Although the evaluations of candidate models using only 8 data points are likely to be much less accurate than those using all 128 data points, each of the individual evaluations using 8 points required only 0.0625 times as much computational effort as when all 128 points were calculated. As computational effort in this application of GP is dominated by the evaluation of candidate models, 16 times as many individuals can be evaluated for a given amount of computational effort with the 8-point method. A priori it was not clear which method would perform better given this evaluation accuracy versus number of evaluations tradeoff. For purposes of fair comparison between methods, the total computational effort as measured by number of integration steps performed was held constant within each example shown below.

In the third method of evaluating candidate models, each candidate model was again integrated for 128 time steps, but now only 8 of the resulting predicted time series data points were compared against the corresponding observed time series data points. This method is referred to as "8 Coev." below. The selection of these 8 points from out of the 128 total generated data points was performed using the Coevolution of Fitness Predictors algorithm detailed in Schmidt and Lipson [67]. That study found that only a small number (approximately 8) of the data points from a complete training data set are necessary to accurately evaluate individual fitnesses on a symbolic regression task. Although those points can be selected randomly, coevolving the location of points based on the accuracy of the resulting fitness estimates was found to accelerate evolution the most. Although this main result does not apply here because the number of evaluated data points is defined by the number of integration steps taken, the authors also found that even when the advantage of decreased computational effort is not taken into account, coevolving the location of evaluation points accelerated evolution over other methods. We hypothesized that using only 8 out of the 128 obtained data points but selecting them with coevolution might confer a similar advantage for our purposes here. Note that there is no advantage in terms of computational effort of this coevolutionary method over simply using all 128 points to evaluate a candidate model, as in both cases an individual candidate model must be integrated for 128 time steps.

Depending on the method, either 128 integration points or 8 integration points were used as training data to calculate the error used to guide evolution of the models. However, in the results below, error is instead reported on test data not used to guide evolution in order to better evaluate how well the evolved models could make predictions. Regardless of the method used to train the models, this test error was obtained by integrating the evolved model for 256 time steps and comparing the

last 128 points to the corresponding points in the target model using the error calculation in Eq. (6).

In the examples of Sects. 4.1, 4.2, 4.3, and 4.6, the number of point evaluations and integration steps were as described above. However, due to the nature of the empirical data, in Sect. 4.4, 256 instead of 128 integration steps were used during training and 320 instead of 256 integration steps were used for computing test error. The 8 Coev. and 8 All methods still employed only 8 points. In Sect. 4.5, 128 integration steps were used during training, but only 192 integration steps were used for computing test error. Again, the 8 Coev. and 8 All methods employed only 8 points. The differences between Sects. 4.1–4.3 and 4.4–4.5 are reflected in the figures illustrating the data sets. For consistency we refer to the three evaluation methods as "128 All," "8 Coev.," and "8 All" for all examples below.

## 4 Results

We studied the performance of our algorithm on several systems with hidden variables. The six main systems studied are shown in Fig. 2, which summarizes the target system, the behavior of the system, and some results obtained with our algorithm.

### 4.1 Initial experiments

For our initial experiments, we used two simple synthetic nonlinear dynamical systems with known ODE models. The first was given by:

$$\begin{cases} \frac{dX_1}{dt} = X_1 X_2 \\ \frac{dX_2}{dt} = -0.5 X_1 \end{cases} \tag{8}$$

with initial values $y_1(0) = 1.0$, $y_2(0) = -1.0$. Data was generated from this target model, both with and without noise (Fig. 3). We designated $X_2$ as a hidden variable and the data available to our GP algorithm consisted only of the time series data for $X_1$. We then attempted to reverse-engineer the exact model given in Eq. (8) using only this time series data for $X_1$. This was repeated for 50 independent trials using each of the three fitness evaluation methods described in Sect. 3.3. Figure 4 summarizes the results of these runs and shows test error of the best model found as a function of computational effort for each of the three methods. For both noise-free data (Fig. 4a) and data to which noise was added (Fig. 4b), evaluating a candidate model by integrating for 128 time steps and then using all 128 resulting points (128 All) resulted in models that on average generalized better to unseen test data. Less successful were the methods of integrating for 128 steps but using only 8 coevolved points (8 Coev.) and of integrating for only 8 steps (8 All).

Examining the evolved ODE models revealed that in many of the 50 trials, the exact form of the target model was found:

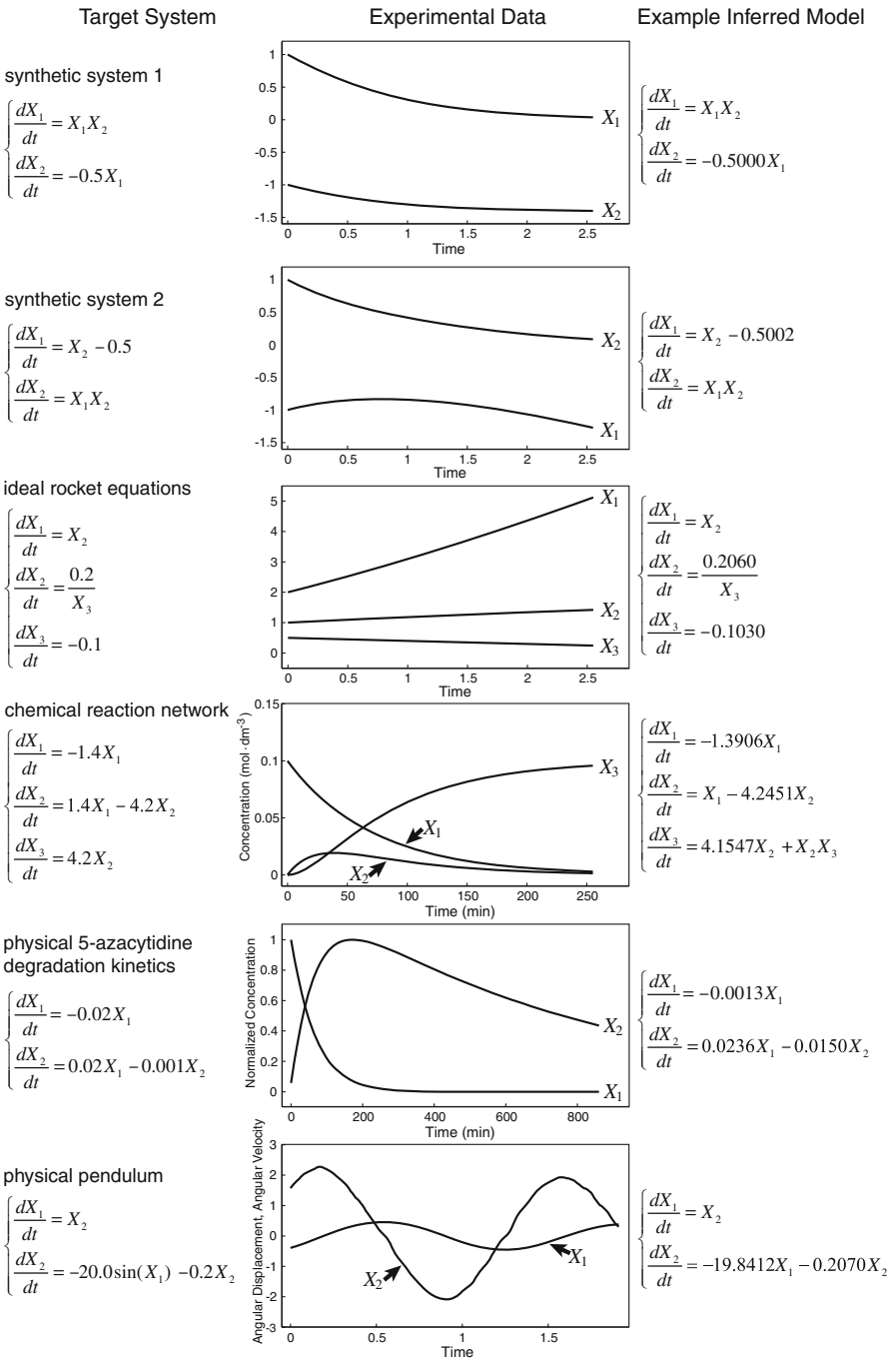



**Fig. 2** Summary of dynamical systems studied. For the two physical systems (5-azacytidine degradation kinetics and pendulum), the target model is the approximate theoretical model derivable from first principles

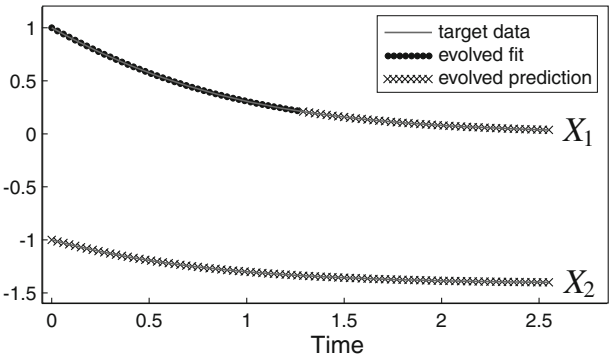



**Fig. 3** Behavior of the first target system and an evolved model from Sect. 4.1. Equation (8) was used to generate the target data shown in the *solid gray line*. The behavior of an evolved ODE model on the training data ("evolved fit") is shown in *solid circles* and the behavior of that evolved model to test data not seen during training ("evolved prediction") is shown with x's. The behavior of the hidden variable ($X_2$) in the evolved model is entirely plotted with x's as none of the time series data from $X_2$ was available for training

$$\begin{cases} \frac{dX_1}{dt} = X_1 X_2 \\ \frac{dX_2}{dt} = \alpha X_1 \end{cases} \tag{9}$$

with initial values $\hat{y}_1(0) = 1.0, \hat{y}_2(0) = \beta$, where $\alpha$ and $\beta$ are floating point constants. Table 1 shows the frequency with which this form was identified and the variability of the two constants across trials in which this form was identified. Figure 3 compares the response of a typical evolved model having the target form with the response of the target model, showing a very close match on both the observed time series for $X_1$ and the hidden time series for $X_2$.

Looking more closely at the trials in which the exact form of the target model was not identified, we frequently found that equally accurate and parsimonious models with a different structure were identified. For example, in 38 out of 50 trials without noise and using the 128 All evaluation method, the exact target model form was identified (Table 1). In all 12 out of the remaining 12 trials, accurate models with this alternative structure were identified:

$$\begin{cases} \frac{dX_1}{dt} = \alpha X_1 X_2 \\ \frac{dX_2}{dt} = X_1 \end{cases} \tag{10}$$

with initial values $\hat{y}_1(0) = 1.0, \hat{y}_2(0) = \beta$, where $\alpha$ and $\beta$ are floating point constants. In all cases, $\alpha$ was approximately $-0.5$ and $\beta$ was approximately 2.0. Both these models and the target model in Eq. (8) have a complexity of 6, and in most cases the training error for models with this structure was approximately the same as the training error for models with the target structure. In other words, models with the structure shown in Eq. (10) appear to represent a local optimum in the space of possible models that was very attractive to the evolutionary search, thereby making our goal of recovering the exact target structure difficult.

One approach to recovering the exact target structure despite the presence of local optima was motivated by the observation that the local optimum represented by

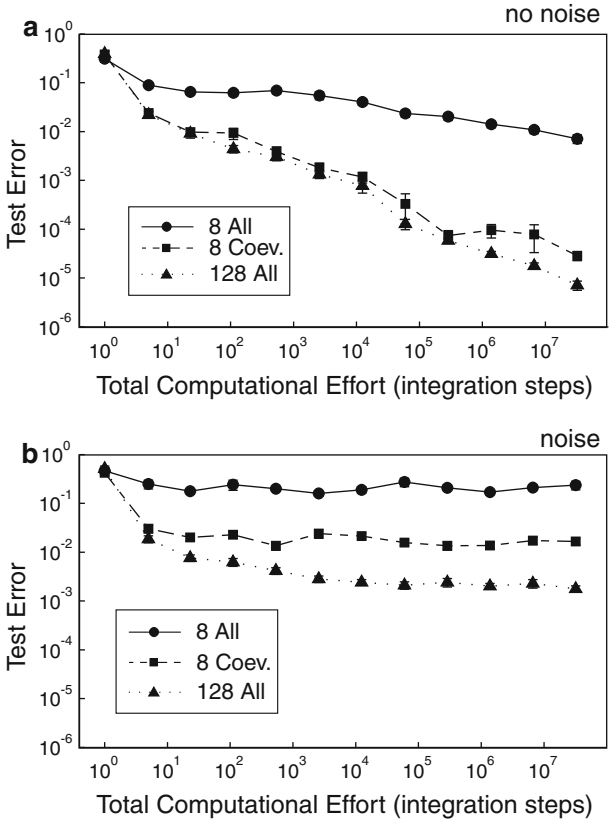



**Fig. 4** Summary of evolutionary runs for the first example in Sect. 4.1. The error of the best model on test data not seen during training was calculated each generation. In selected generations, the mean of this test error was plotted for each of the three evaluation methods with *symbols* as shown in the legend. *Error bars* represent ± the standard error of the mean. In most cases the *error bars* are smaller than the height of the markers and are not visible. **a** shows results for noise-free data and **b** shows results for data to which noise was added

**Table 1** Evolved models for the first system in Sect. 4.1

| Evaluation method, noise | Num. successful trials out of 50 | $\alpha$ | $\beta$ |
|---|---|---|---|
| 128 All, no noise | 38 | −0.5000 ± 0.0003 | −1.0000 ± 0.0000 |
| 128 All, noise | 21 | −0.5160 ± 0.0226 | −0.9928 ± 0.0092 |
| 8 Coev., no noise | 38 | −0.5000 ± 0.0002 | −1.0000 ± 0.0000 |
| 8 Coev., noise | 13 | −0.5239 ± 0.0294 | −0.9904 ± 0.0113 |
| 8 All, no noise | 8 | −0.5006 ± 0.0004 | −1.0000 ± 0.0000 |
| 8 All, noise | 0 | NA | NA |

A successful trial indicates that the form given in Eq. (9) was evolved in the trial. $\alpha$ and $\beta$ refer to the floating point constants in (9). The variability of the constants across trials in which the target form in (9) was evolved are given as mean ± standard deviation. No data for a given measurement is indicated by "NA."

models with the structure in Eq. (10) appeared to be associated with initial values $\hat{y}_1(0) = 1.0, \hat{y}_2(0) = 2.0$, whereas evolved models matching the true target structure tended to have initial values very close to the true values of $\hat{y}_1(0) = 1.0, \hat{y}_2(0) = -1.0$. As the initial value of the hidden variable $(\hat{y}_2(0))$ is treated as a free parameter to be optimized, it is possible to simply constrain this initial value to a range such as $(-1.1, -0.9)$, which could prevent the evolutionary search from becoming trapped in the local optimum associated with initial values of approximately 2.0. A priori, there would be no way to know that the initial values should be constrained in this way. However, assuming that the empirical time series data can be normalized to approximately order unity and at the expense of additional computational effort, several trials can be used to "scan" through the space of possible initial values.

We performed such scans with 50 trials each in which the initial values were constrained to a range of 0.2 and in which the center of the ranges was successively set to each of $(-6.0, -5.8, \ldots, 3.6, 3.8)$. When the range of initial values was set to $(-1.1, -0.9)$, models of the form shown in Eq. (8) were found with parameters closely matching those of the target. An example of one such model is:

$$\begin{cases} \frac{dX_1}{dt} = X_1 X_2 \\ \frac{dX_2}{dt} = -0.5000 X_1 \end{cases} \tag{11}$$

with initial values $\hat{y}_1(0) = 1.0, \hat{y}_2(0) = -1.0000$. This initial value scanning approach successfully recovered the target model when the 128 All and 8 Coev. methods were used, from both noise-free data and from data with added noise. Although the alternative model in Eq. (10) was also found in some trials of the scans, this approach is one potential means of reliably recovering the exact target model structure despite the presence of highly attractive local optima in model space.

In our initial experiments, we also used this target system:

$$\begin{cases} \frac{dX_1}{dt} = X_2 - 0.5 \\ \frac{dX_2}{dt} = X_1 X_2 \end{cases} \tag{12}$$

with initial values $y_1(0) = -1.0, y_2(0) = 1.0$. Again we designated $X_2$ as a hidden variable and data was generated both with and without noise (Fig. 5). This system is

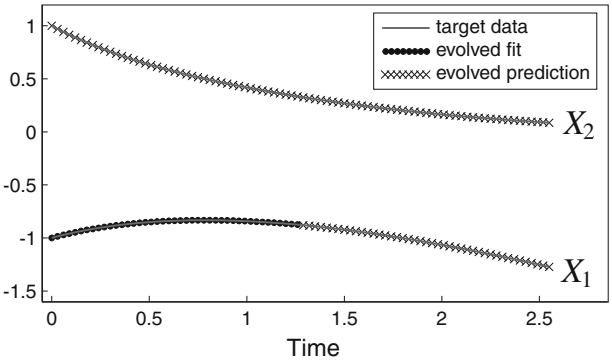



**Fig. 5** Behavior of the second target system and an evolved model from Sect. 4.1. Equation (12) was used to generate the target data shown in the *solid gray line*

very similar to Eq. (8) except that now the hidden variable is described by a nonlinear expression whereas in Eq. (8), the hidden variable was described by a linear expression. Figure 6 summarizes the results of 50 independent trials using only the time series data from $X_1$ in Eq. (12) as the basis for model identification. The results show that as before, evaluating a candidate model by integrating for 128 time steps and then using all 128 resulting points (128 All) was the most successful evaluation method.

The target form of this second system in general is:

$$\begin{cases} \frac{dX_1}{dt} = X_2 - \alpha \\ \frac{dX_2}{dt} = X_1 X_2 \end{cases} \tag{13}$$

with initial values $\hat{y}_1(0) = -1.0, \hat{y}_2(0) = \beta$, where $\alpha$ and$\beta$ are floating point constants. Table 2 shows the frequency with which this form was identified and the variability of the two constants across trials in which this form was identified. Figure 5 compares the response of a typical evolved model matching the target form with that of the target model.

As with the previous test case, alternative model structures were found that were difficult for the evolutionary search to reject on the basis of accuracy or parsimony. One frequently encountered alternative structure was:

$$\begin{cases} \frac{dX_1}{dt} = X_1 \\ \frac{dX_2}{dt} = X_1(X_2 + \alpha) \end{cases} \tag{14}$$

with initial values $\hat{y}_1(0) = -1.0, \hat{y}_2(0) = \beta$, where $\alpha$ and$\beta$ are floating point constants. In all cases, $\alpha$ and$\beta$ were approximately 0.5. Models with this structure have complexity equal to the target model in Eq. (12), and depending on the exact values of the constants present, often gave the same or lower training errors as models with the target structure. To overcome this attractive local optimum in the space of possible models, we again used the initial value scanning approach described above. When the range of initial values was set to (0.9, 1.1), models of the form shown in Eq. (12) were found with parameters closely matching those of the target. An example of one such model is:

$$\begin{cases} \frac{dX_1}{dt} = X_2 - 0.5002 \\ \frac{dX_2}{dt} = X_1 X_2 \end{cases} \tag{15}$$

with initial values $\hat{y}_1(0) = -1.0$, $\hat{y}_2(0) = 1.0003$. This initial value scanning approach worked successfully for the 128 All and 8 Coev. methods both with and without noise.

### 4.2 Ideal rocket equations

We next considered an instance of the ideal rocket equations given in Eq. (1):

$$\begin{cases} \frac{dX_1}{dt} = X_2 \\ \frac{dX_2}{dt} = \frac{0.2}{X_3} \\ \frac{dX_3}{dt} = -0.1 \end{cases} \tag{16}$$

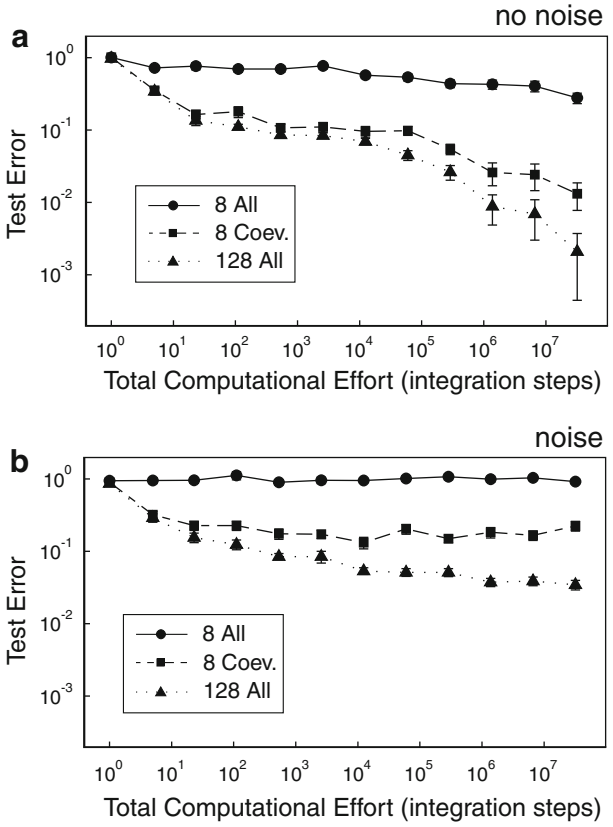



**Fig. 6** Summary of evolutionary runs for the second example in Sect. 4.1. **a** shows results for noise-free data and **b** shows results for data with noise

**Table 2** Evolved models for the second system in Sect. 4.1

| Evaluation method, noise | Num. successful trials out of 50 | $\alpha$ | $\beta$ |
|---|---|---|---|
| 128 All, no noise | 25 | $0.5000 \pm 0.0004$ | $1.0000 \pm 0.0005$ |
| 128 All, noise | 11 | $0.5036 \pm 0.0101$ | $1.0017 \pm 0.0043$ |
| 8 Coev., no noise | 24 | $0.4999 \pm 0.0006$ | $0.9998 \pm 0.0008$ |
| 8 Coev., noise | 11 | $0.5016 \pm 0.0029$ | $1.0031 \pm 0.0057$ |
| 8 All, no noise | 1 | $0.5001 \pm 0.0000$ | $1.0001 \pm 0.0000$ |
| 8 All, noise | 0 | NA | NA |

A successful trial indicates that the form given in Eq. (13) was evolved in the trial

with initial values $y_1(0) = 2.0$, $y_2(0) = 1.0$, $y_3(0) = 0.5$. $X_3$ corresponds to the mass variable in Eq. (1) and so it was taken as a hidden variable, whereas position ($X_1$) and velocity ($X_2$) were both observed variables. Target data from this system was simulated both with and without measurement noise as shown in Fig. 7. Figure 8

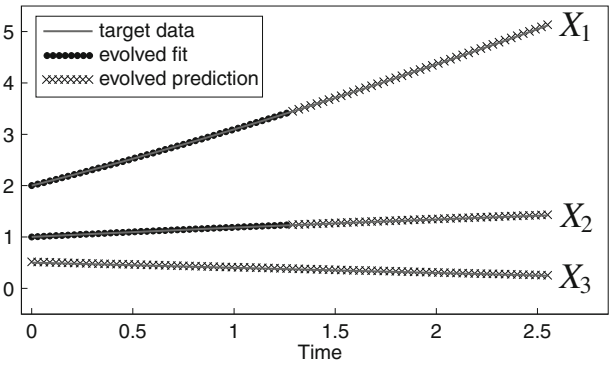



**Fig. 7** Behavior of the target system and an evolved model from Sect. 4.2. Equation (16) was used to generate the target data shown in the *solid gray line*

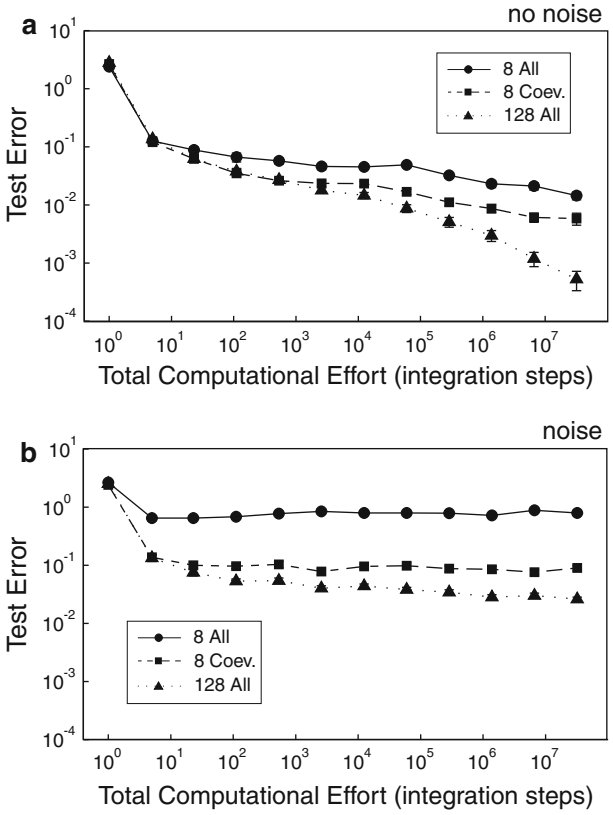



**Fig. 8** Summary of evolutionary runs for Sect. 4.2. **a** shows results for noise-free data and **b** shows results for data with noise

summarizes the results of 50 independent trials using the time series data from $X_1$ and $X_2$ in Eq. (16) as the basis for model identification.

The target form of the ODE models for this example in general is:

$$\begin{cases} \frac{dX_1}{dt} = X_1 \\ \frac{dX_2}{dt} = \frac{\alpha}{X_3} \\ \frac{dX_3}{dt} = \beta \end{cases} \tag{17}$$

with initial values $\hat{y}_1(0) = 2.0$, $\hat{y}_2(0) = 1.0$, $\hat{y}_3(0) = \chi$, where $\alpha$, $\beta$, $\chi$ are floating point constants. Table 3 shows the frequency with which this form was identified and the variability of the floating point constants across trials in which this form was identified. Unlike with the examples of Sect. 4.1 above, the true values of the floating point constants were difficult for the GP algorithm to identify. This was the case even when the correct form of the model was identified.

Although this difficulty in identifying parameter values appears to be due to a different type of local optimum in model space than that encountered in the examples of Sect. 4.1, the initial value scanning approach used there is again appropriate for attempting to recover the target model. When the range of initial values for the hidden variable was set to (0.4, 0.6), models of the form shown in Eq. (16) were found with parameters closely matching those of the target. This initial value scanning approach worked successfully for the 128 All and 8 Coev. methods without noise and the 128 All method with noise added. Figure 7 compares the response of the target model to the response of this typical evolved model obtained with initial value scanning:

$$\begin{cases} \frac{dX_1}{dt} = X_2 \\ \frac{dX_2}{dt} = \frac{0.2060}{X_3} \\ \frac{dX_3}{dt} = -0.1030 \end{cases} \tag{18}$$

with initial values $\hat{y}_1(0) = 2.0, \hat{y}_2(0) = 1.0, \hat{y}_0 = 0.5151$.

**Table 3** Evolved models for Sect. 4.2

| Evaluation method, noise | Num. successful trials out of 50 | $\alpha$ | $\beta$ | $\chi$ |
|---|---|---|---|---|
| 128 All, no noise | 42 | 0.7136 ± 2.5092 | −0.3565 ± 1.2542 | 0.0011 ± 0.0014 |
| 128 All, noise | 0 | NA | NA | NA |
| 8 Coev., no noise | 40 | −0.3243 ± 2.9437 | 0.1881 ± 1.4694 | 0.0013 ± 0.0012 |
| 8 Coev., noise | 5 | 1.1016 ± 2.3599 | −0.6617 ± 1.1341 | 0.0299 ± 0.0287 |
| 8 All, no noise | 8 | 0.5073 ± 3.1272 | −0.2541 ± 1.5642 | 0.0005 ± 0.0005 |
| 8 All, noise | 0 | NA | NA | NA |

A successful trial indicates that the form given in Eq. (17) was evolved in the trial

### 4.3 Chemical reaction network

Several previous studies that evolved ODE models have used a particular chemical reaction problem as a benchmark [4, 5, 15, 23]. This problem involves the reaction between formaldehyde and carbamide, which produces methylol urea and methylene urea according to:

$$\begin{aligned} \mathrm{HCHO} + (\mathrm{NH_2})_2\mathrm{CO} &\rightarrow \mathrm{H_2N \cdot CO \cdot NH \cdot CH_2OH} \\ \mathrm{H_2N \cdot CO \cdot NH \cdot CH_2OH} + (\mathrm{NH_2})_2\mathrm{CO} &\rightarrow (\mathrm{NH_2CONH})_2\mathrm{CH_2} \end{aligned} \tag{19}$$

The concentrations of formaldehyde ($X_1$), methylol urea ($X_2$), and methylene urea ($X_3$) change over time in a way that satisfies the ODE:

$$\begin{cases} \frac{dX_1}{dt} = -1.4X_1 \\ \frac{dX_2}{dt} = 1.4X_1 - 4.2X_2 \\ \frac{dX_3}{dt} = 4.2X_2 \end{cases} \tag{20}$$

with initial values $y_1(0) = 0.1$, $y_2(0) = 0.0$, $y_3(0) = 0.0$. The reaction rate parameters and initial values are typically observed experimental values given in Cao et al. [23]. Target data from this system was simulated both with and without measurement noise (Fig. 9).

Here we consider the problem of modeling the complete reaction network when data for one or more of the three concentration time series is not available. Due to the nature of the dependencies in the reaction network, the presence of hidden variables can drastically change the modeling problem. For example, if only the concentration of formaldehyde ($X_1$) is measured, there is no way to infer expressions for the rate of change of the concentration of methylol urea ($X_2$) or methylene urea ($X_3$). This is because the rate of change of the concentration of formaldehyde is effectively independent of the concentrations of all species other than formaldehyde itself, as can be seen in Eq. (20). There are six possible combinations of measurements that can be taken (aside from measuring all three concentrations): $\{X_1\}$, $\{X_2\}$, $\{X_3\}$, $\{X_1, X_2\}$, $\{X_1, X_3\}$, $\{X_2, X_3\}$. We will consider each of these cases

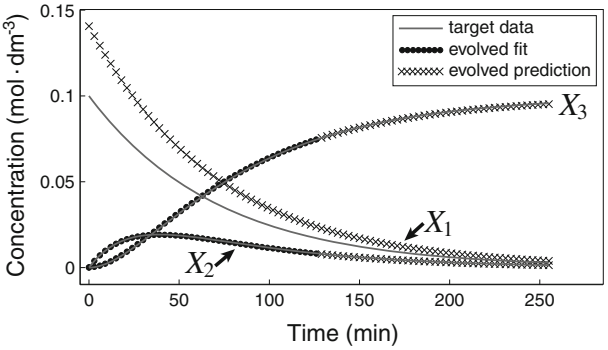



**Fig. 9** Behavior of the target system and an evolved model from Sect. 4.3. Equation (20) was used to generate the target data shown in the *solid gray line*

in turn and use only the variable names for brevity. Due to the large number of cases, for these experiments only the 128 All evaluation method was used, as previous results suggested it was the most effective method.

When only $\{X_1\}$ is measured, the expression for the rate of change of $X_1$ is constrained in the ODE model for the system, whereas no information exists as the basis for finding an expression for the rate of change of $X_2$ or $X_3$. Figure 10 summarizes the results of 50 independent trials using the time series data $\{X_1\}$ as the basis for model identification. The target form of the ODE models for this problem in general is:

$$\begin{cases} \frac{dX_1}{dt} = \alpha X_1 \\ \frac{dX_2}{dt} = \sim \\ \frac{dX_3}{dt} = \sim \end{cases} \tag{21}$$

with initial values $\hat{y}_1(0) = 0.1$, $\hat{y}_2(0) = \sim$, $\hat{y}_3(0) = \sim$, where $\alpha$ is a floating point constant and $\sim$ is arbitrary. Table 4 shows the frequency with which this form was identified and the variability of the floating point constant across trials in which this form was identified.

When the measurements consist of $\{X_2\}$ or $\{X_1, X_2\}$, the expressions for the rate of change of $X_1$ and $X_2$ are constrained in the ODE model for the system, whereas no information exists as the basis for finding an expression for the rate of change of $X_3$. Figure 10 summarizes the results of 50 independent trials using the time series data $\{X_2\}$ or $\{X_1, X_2\}$ as the basis for model identification. The target form of the ODE models for this problem in general is:

$$\begin{cases} \frac{dX_1}{dt} = \alpha X_1 \\ \frac{dX_2}{dt} = \beta X_1 - \chi X_2 \\ \frac{dX_3}{dt} = \sim \end{cases} \tag{22}$$

with initial values $\hat{y}_1(0) = \epsilon$, $\hat{y}_2(0) = 0.0$, $\hat{y}_3(0) = \sim$, where $\alpha$, $\beta$, $\chi$, $\varepsilon$ are floating point constants and $\sim$ is arbitrary. (In the case when $X_1$ and $X_2$ were measured, $\varepsilon$ was set to the observed initial value.) Table 4 shows the frequency with which this form was identified and the variability of the four constants across trials in which this form was identified.

When the measurements consist of $\{X_3\}$, $\{X_1, X_3\}$, or $\{X_2, X_3\}$, the expressions for the rate of change of all three variables are constrained in the ODE model for the system. Figure 11 summarizes the results of 50 independent trials using the time series data $\{X_3\}$, $\{X_1, X_3\}$, or $\{X_2, X_3\}$ as the basis for model identification. The target form of the ODE models for this problem in general is:

$$\begin{cases} \frac{dX_1}{dt} = \alpha X_1 \\ \frac{dX_2}{dt} = \beta X_1 - \chi X_2 \\ \frac{dX_3}{dt} = \delta X_2 \end{cases} \tag{23}$$

with initial values $\hat{y}_1(0) = \epsilon$, $\hat{y}_2(0) = \phi$, $\hat{y}_3(0) = 0.0$, where $\alpha$, $\beta$, $\delta$, $\chi$, $\varepsilon$, $\varphi$ are floating point constants. (In the case when $X_1$ and $X_3$ were measured, $\varepsilon$ was set to the observed initial value. When $X_2$ and $X_3$ were measured, $\varphi$ was set to the observed

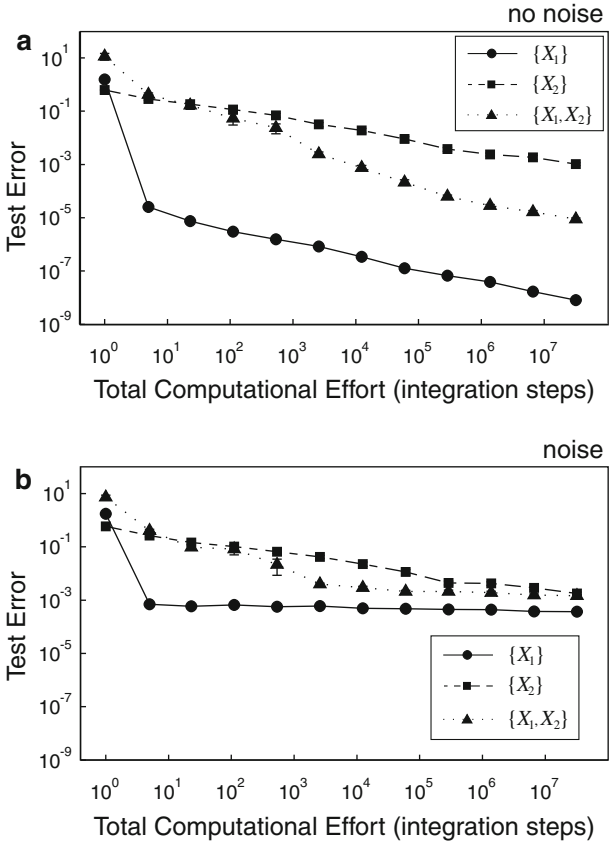



**Fig. 10** Summary of evolutionary runs for Sect. 4.3 with measured variables $(X_1\}$, $\{X_2\}$, $\{X_1, X_2\}$. All runs use the 128 All evaluation method. **a** shows results for noise-free data and **b** shows results for data with noise

initial value.) Table 4 shows the frequency with which this form was identified and the variability of the six constants across trials in which this form was identified.

The results show that identifying the target form and parameters of the model is difficult in all cases except when $\{X_1\}$ or $\{X_1, X_2\}$ are measured. As much of this difficulty could be due to becoming trapped in local optima in model structure and parameter space, we again used the initial value scanning approach, which was successful in many cases. For example, with $\{X_2\}$ measured, the following partial model was recovered from noisy data using the 128 All method:

$$\begin{cases} \frac{dX_1}{dt} = -1.4020X_1 \\ \frac{dX_2}{dt} = X_1 - 4.1926X_2 \\ \frac{dX_3}{dt} = \sim \end{cases} \tag{24}$$

with initial values $\hat{y}_1(0) = 0.1398$, $\hat{y}_2(0) = 0.0$, $\hat{y}_3(0) = \sim$, which closely matches Eq. (22). As another example, with $\{X_2, X_3\}$ measured, the following model was recovered from noisy data using the 128 All method:

**Table 4** Evolved models for Sect. 4.3

| {Measured variables}, noise | Num. successful trials out of 50 | $\alpha$ | $\beta$ | $\chi$ | $\delta$ | $\varepsilon$ | $\varphi$ |
|---|---|---|---|---|---|---|---|
| $\{X_1\}$, no noise | 50 | $-1.4000 \pm 0.0000$ | NA | NA | NA | NA | NA |
| $\{X_1\}$, noise | 50 | $-1.4000 \pm 0.0048$ | NA | NA | NA | NA | NA |
| $\{X_1, X_2\}$, no noise | 50 | $-1.4000 \pm 0.0000$ | $1.4000 \pm 0.0000$ | $4.1999 \pm 0.0001$ | NA | NA | NA |
| $\{X_1, X_2\}$, noise | 49 | $-1.4006 \pm 0.0048$ | $1.3992 \pm 0.0056$ | $4.1974 \pm 0.0161$ | NA | NA | NA |
| $\{X_2\}$, no noise | 25 | $-3.1863 \pm 1.3691$ | $1.0880 \pm 0.6903$ | $2.4107 \pm 1.3729$ | NA | $0.1715 \pm 0.1203$ | NA |
| $\{X_2\}$, noise | 18 | $-2.6312 \pm 1.3873$ | $1.0301 \pm 0.9361$ | $2.6135 \pm 1.4852$ | NA | $0.5836 \pm 1.0006$ | NA |
| $\{X_2, X_3\}$, no noise | 3 | $-1.3882 \pm 0.0129$ | $0.8377 \pm 0.2811$ | $4.2209 \pm 0.0389$ | $4.1974 \pm 0.0506$ | $0.1840 \pm 0.0757$ | NA |
| $\{X_2, X_3\}$, noise | 6 | $-2.7514 \pm 1.5170$ | $1.2043 \pm 1.8326$ | $2.8550 \pm 1.5655$ | $4.1916 \pm 0.0543$ | $0.1220 \pm 0.0437$ | NA |
| $\{X_3\}$, no noise | 1 | $-2.4779 \pm 0.0000$ | $1.0000 \pm 0.0000$ | $2.0000 \pm 0.0000$ | $0.0503 \pm 0.0000$ | $8.7270 \pm 0.0000$ | $0.2894 \pm 0.0000$ |
| $\{X_3\}$, noise | 0 | NA | NA | NA | NA | NA | NA |
| $\{X_1, X_3\}$, no noise | 1 | $-1.4000 \pm 0.0000$ | $-4.5143 \pm 0.0000$ | $4.5143 \pm 0.0000$ | $-1.4119 \pm 0.0000$ | NA | $0.0048 \pm 0.0000$ |
| $\{X_1, X_3\}$, noise | 0 | NA | NA | NA | NA | NA | NA |

For trials with $\{X_1\}$ measured, a successful trial indicates that the form given in Eq. (21) was evolved. For trials with $\{X_2\}$ or $\{X_1, X_2\}$ measured, a successful trial indicates that the form given in Eq. (22) was evolved. For the remaining trials, a successful trial indicates that the form given in Eq. (23) was evolved

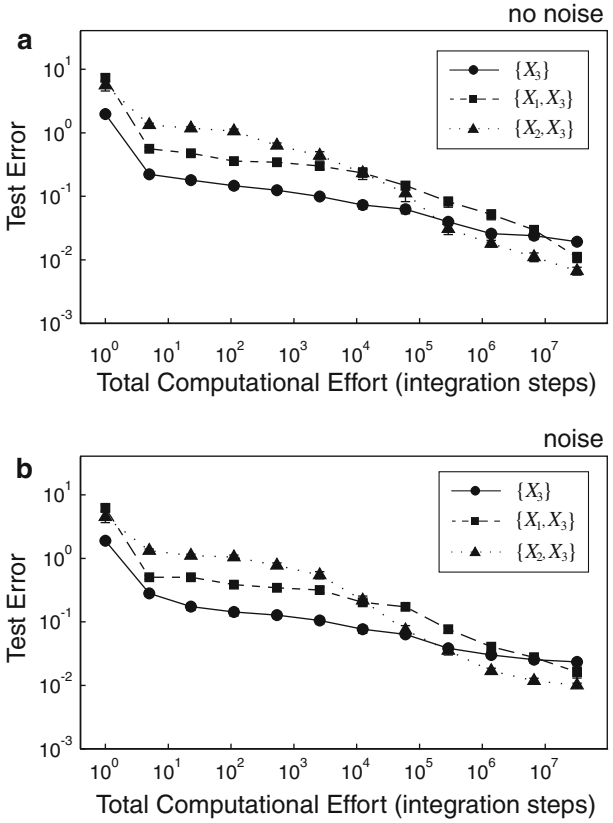



**Fig. 11** Summary of evolutionary runs for Sect. 4.3 with measured variables $\{X_3\}$, $\{X_1, X_3\}$, $\{X_2, X_3\}$. All runs use the 128 All evaluation method. **a** shows results for noise-free data and **b** shows results for data with noise

$$\begin{cases} \frac{dX_1}{dt} = -1.3906X_1 \\ \frac{dX_2}{dt} = X_1 - 4.2451X_2 \\ \frac{dX_3}{dt} = 4.1547X_2 + X_2X_3 \end{cases} \tag{25}$$

with initial values $\hat{y}_1(0) = 0.1407$, $\hat{y}_2(0) = 0.0000$, $\hat{y}_3(0) = 0.0002$, which closely matches Eq. (20) aside from the $X_2X_3$ term in the third equation. The behavior of Eq. (25) is plotted in Fig. 9 and compared with the behavior of the target model given in Eq. (20).

### 4.4 Physical 5-azacytidine degradation kinetics

5-azacytidine is an anti-cancer drug that rapidly degrades in the body when administered by intravenous injection [14, 45]. The initial product formed as a result of this degradation is *N*-(formylamidino)-*N*′-*β*-D-ribofuranosylurea (RGU-CHO). Due to the therapeutic benefits of 5-azacytidine, it is of interest to characterize the

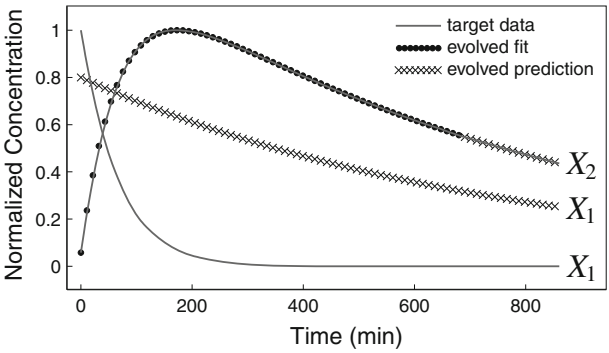



**Fig. 12** Behavior of the target system and an evolved model from Sect. 4.4. The target data shown in the *solid gray line* was obtained from physical experiments performed by Argemí and Saurina [8]

kinetics of this degradation reaction and many studies have done so [8, 14, 45]. Here we use data from the physical 5-azacytidine degradation reaction obtained by Argemí and Saurina [8]. These experimental measurements, shown in Fig. 12, were the input to our GP algorithm as we attempted to infer a model for the data.

We first assumed that all the data was available, that is, that both the concentrations of 5-azacytidine ($X_1$) and RGU-CHO ($X_2$) were measured as in the original experiment [8]. Using the 128 All evaluation method, models of the following form were readily obtained:

$$\begin{cases} \frac{dX_1}{dt} = \alpha X_1 \\ \frac{dX_2}{dt} = \beta X_1 - \chi X_2 \end{cases} \tag{26}$$

with initial values $\hat{y}_1(0) = 1.0000$, $\hat{y}_2(0) = 0.0578$, where $\alpha$, $\beta$, $\chi$ are floating point constants. In all cases, $\alpha$ was approximately $-0.02$, $\beta$ was approximately 0.02, and $\chi$ was approximately 0.001. This explanatory model is consistent with the known reaction mechanism [14] and provided a baseline for comparison with subsequent experiments described next.

We then assumed that only the concentration of RGU-CHO ($X_2$) was measured and that the concentration of 5-azacytidine ($X_1$) was a hidden variable. Figure 13 summarizes the results of 50 independent trials using the empirical $X_2$ data as the basis for model identification. A model structure resembling that of (26) was frequently identified:

$$\begin{cases} \frac{dX_1}{dt} = \alpha X_1 \\ \frac{dX_2}{dt} = \beta X_1 - \chi X_2 + \delta \end{cases} \tag{27}$$

with initial values $\hat{y}_1(0) = \epsilon$, $\hat{y}_2(0) = 0.0578$, where $\alpha$, $\beta$, $\chi$, $\delta$, $\varepsilon$ are floating point constants. Table 5 shows the frequency with which the form in Eq. (27) was identified and the variability of the floating point constants across trials in which this form was identified. Regardless of whether the form in Eq. (27) was identified or how low their test error was, the evolved models showed high variability in the values of the floating point constants across several trials.

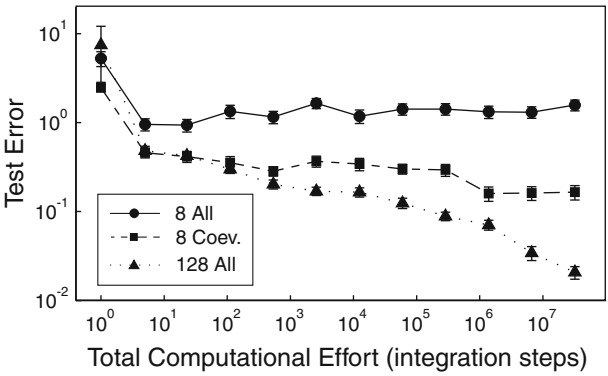



**Fig. 13** Summary of evolutionary runs for Sect. 4.4

We used the initial value scanning approach for attempting to identify the model more precisely and reliably with $X_1$ as a hidden variable. The scan in this case was not successful at identifying models matching the experimental data. Even when models of the form shown in Eq. (26) or Eq. (27) were found, the nature of the coupled system of ODEs made identifying the correct parameters difficult. For example, with the 128 All method, this model was found:

$$\begin{cases} \frac{dX_1}{dt} = -0.0013X_1 \\ \frac{dX_2}{dt} = 0.0236X_1 - 0.0150X_2 \end{cases} \tag{28}$$

with initial values $\hat{y}_1(0) = 0.8000$, $\hat{y}_2(0) = 0.0578$. Figure 12 compares the response of this evolved model to the empirical data, showing that the behavior of the hidden variable $X_1$ is not predicted accurately even though the correct model form was recovered and the behavior of the measured variable $X_2$ is predicted accurately.

### 4.5 Physical pendulum

In ODE form, a simple pendulum is modeled with two dynamic variables, one for the angular displacement of the bob ($\theta$) and one for its angular velocity ($\omega$). We will continue the practice of referring to the variables as $X_1$ and $X_2$, with $X_1$ representing angular displacement and $X_2$ representing angular velocity. Here we use data obtained from a physical pendulum obtained by Bongard and Lipson [18]. These experimental measurements, shown in Fig. 14, were the input to our GP algorithm as we attempted to infer a model for the data.

As with the 5-azacytidine physical system, we first assumed that all the data was available, that is, that both the angular displacement ($X_1$) and angular velocity ($X_2$) of the pendulum were measured as in the original experiment [18]. Using the 128 All evaluation method, models of the following form were readily obtained:

$$\begin{cases} \frac{dX_1}{dt} = X_2 \\ \frac{dX_2}{dt} = \alpha \sin(X_1) - \beta X_2 \end{cases} \tag{29}$$

**Table 5** Evolved models for Sect. 4.4

| Evaluation method | Num. successful trials out of 50 | $\alpha$ | $\beta$ | $\chi$ | $\delta$ | $\varepsilon$ |
|---|---|---|---|---|---|---|
| 128 All | 15 | $-0.0032 \pm 0.0032$ | $0.2120 \pm 0.5560$ | $0.0127 \pm 0.0032$ | $0.0039 \pm 0.0017$ | $0.8600 \pm 0.3445$ |
| 8 Coev. | 4 | $-0.0054 \pm 0.0059$ | $0.2683 \pm 0.4881$ | $0.2626 \pm 0.5063$ | $0.1074 \pm 0.2087$ | $0.8453 \pm 0.3095$ |
| 8 All | 0 | NA | NA | NA | NA | NA |

A successful trial indicates that the form given in Eq. (27) was evolved in the trial

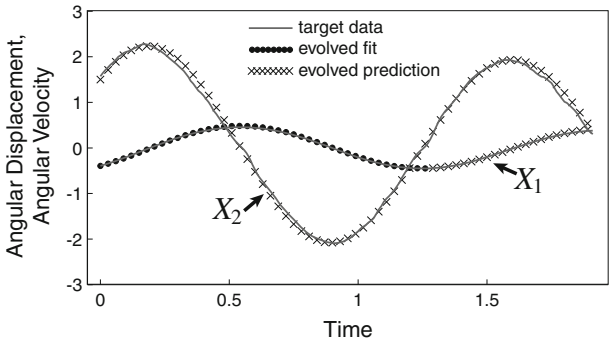



**Fig. 14** Behavior of the target system and an evolved model from Sect. 4.5. The target data shown in the *solid gray line* was obtained from physical experiments performed by Bongard and Lipson [18]

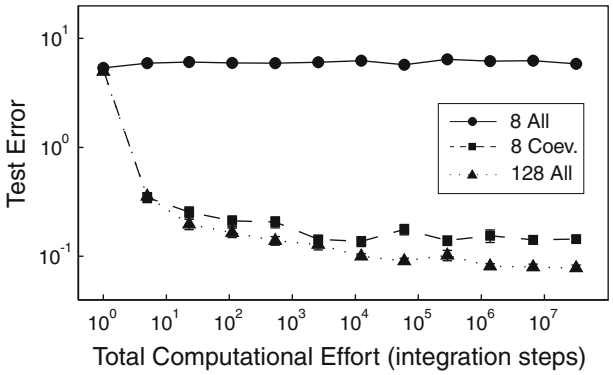



**Fig. 15** Summary of evolutionary runs for Sect. 4.5

with initial values $\hat{y}_1(0) = -0.3933$, $\hat{y}_2(0) = 1.5701$, where $\alpha$, $\beta$ are floating point constants. In all cases, $\alpha$ was approximately $-20.0$ and $\beta$ was approximately 0.2 as in Bongard and Lipson [18]. This explanatory model matches the well-known model derivable from first principles with Newtonian physics and provided a baseline for comparison with subsequent experiments described next.

We then assumed that only the angular displacement ($X_1$) was measured and that the angular velocity ($X_2$) was a hidden variable. Figure 15 summarizes the results of 50 independent trials using the empirical $X_1$ data as the basis for model identification. Despite many models achieving low test error, no models with the structure of Eq. (29) were identified, and this hidden variable modeling problem appeared to be the most difficult example considered so far.

However, the initial value scanning approach generated models of the following form with both the 128 All and 8 Coev. evaluation techniques:

$$\begin{cases} \frac{dX_1}{dt} = X_2 \\ \frac{dX_2}{dt} = \alpha X_1 - \beta X_2 \end{cases} \tag{30}$$

with initial values $\hat{y}_1(0) = -0.3933$, $\hat{y}_2(0) = \chi$, where $\alpha$, $\beta$, $\chi$ are floating point constants. In all cases, $\alpha$ was approximately $-20.0$ and $\beta$ was approximately 0.2. An example of one of these models is:

$$\begin{cases} \frac{dX_1}{dt} = X_2 \\ \frac{dX_2}{dt} = -19.8412X_1 - 0.2070X_2 \end{cases} \tag{31}$$

with initial values $\hat{y}_1(0) = -0.3933$, $\hat{y}_2(0) = 1.5000$. Figure 14 compares the behavior of this model to the empirical data, showing that the model accurately predicts both the measured and hidden variables despite being more parsimonious than the first principles model given in Eq. (29).

### 4.6 Random systems of ODEs

In order to study how the GP algorithm scales with increasing problem difficulty, we measured performance as a function of target system complexity for random systems of ODEs. These random systems were generated by building a random binary tree of operations and operands for each variable in the system. Each tree was then simplified using the MATLAB Symbolic Math Toolbox to obtain an accurate measure of complexity for the system. Random systems with two variables and with three variables were generated. An example of a random ODE system with two variables and a complexity of 8 is:

$$\begin{cases} \frac{dX_1}{dt} = X_1X_2 \\ \frac{dX_2}{dt} = X_1 - 4.4362X_2 \end{cases} \tag{32}$$

with initial values $y_1(0) = 3.5121$, $y_2(0) = -8.3313$. An example of a random ODE system with three variables and a complexity of 17 is:

$$\begin{cases} \frac{dX_1}{dt} = \frac{4.0801}{X_2 - 7.1723} + 9.4551 \\ \frac{dX_2}{dt} = X_1^2X_3^2 \\ \frac{dX_3}{dt} = 1.1222 + X_1 \end{cases} \tag{33}$$

with initial values $y_1(0) = -1.6493$, $y_2(0) = -0.8373$, $y_3(0) = 7.4122$.

Figure 16 summarizes the results for random systems with two variables and shows performance in terms of test error of the best model found at the end of each run as a function of model complexity. 50 independent trials with the 128 All fitness evaluation method were performed for each of 12 different model complexities ranging from 2 to 24. For each trial within each complexity class, a different random target was generated. In the experiments summarized in Fig. 16a, both variables in the target system were measured. Test error was uniformly low and increased smoothly as a function of target complexity, both without measurement noise and with noise added. In the experiments summarized in Fig. 16b, one of the variables in the system was hidden. The relatively flat test errors across different target complexities for both noisy and noise-free conditions indicates that the GP algorithm had difficulty identifying the more complex target models.

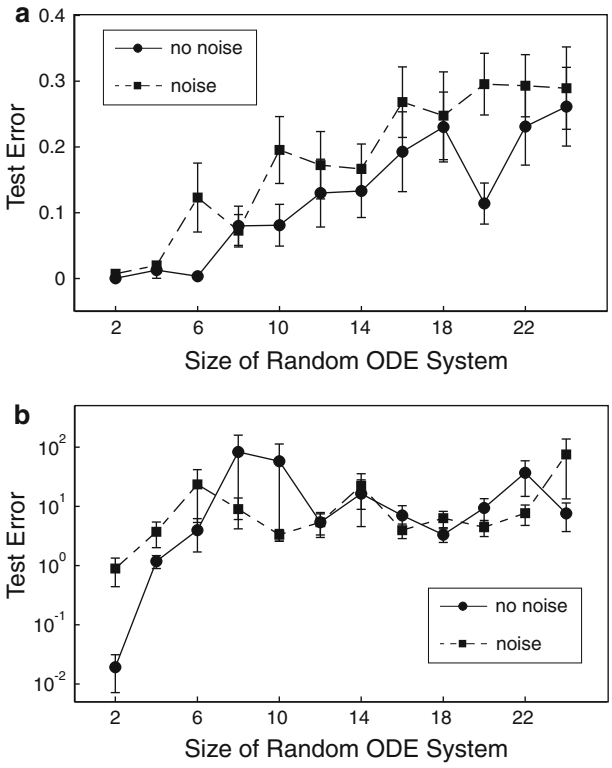



**Fig. 16** Summary of experiments with random systems of ODEs of two variables. The error of the best model on test data not seen during training was calculated at the end of each run. The mean of this test error was plotted for trials with and without added measurement noise using symbols as shown in the legend. *Error bars* represent ± the standard error of the mean. **a** shows results for trials in which both variables in the system were measured and **b** shows results for trials with one hidden variable

Figure 17 summarizes the results for random systems with three variables and 12 different model complexities ranging from 3 to 25. In the experiments summarized in Fig. 17a, all three variables were measured. In Fig. 17b, one variable was hidden and in Fig. 17c, two variables were hidden. Again, the results collectively indicate that the more complex target models can not be reliably modeled when hidden variables are present, with little or no effect of added measurement noise.

## 5 Discussion

Here we present the first work on inferring systems of ODEs with hidden variables using GP. Our goal was to reverse-engineer the exact form and parameters of the ground truth model in the case of synthetic data, or a parsimonious explanatory model in the case of physical data. Many of the results were promising, but we encountered three main challenges to achieving this goal.

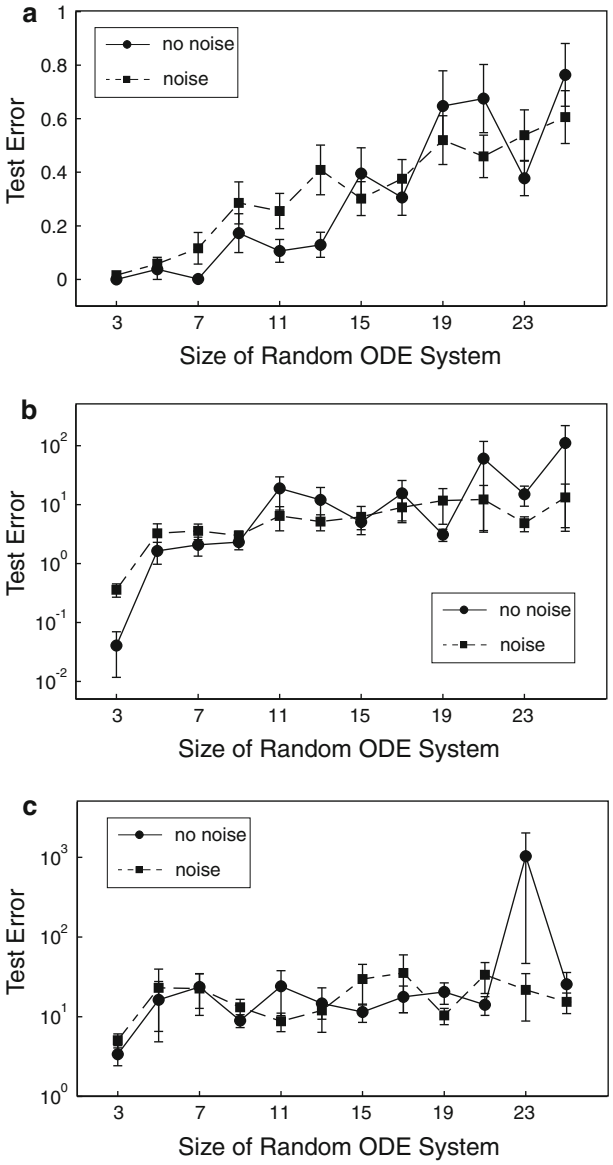



**Fig. 17** Summary of experiments with random systems of ODEs of three variables. **a** shows results for trials in which all variables in the system were measured, **b** shows results for trials with one hidden variable, and **c** shows results for trials with two hidden variables

As mentioned previously, the primary challenge is that ODE models with alternative forms and parameter values often appear as attractive local optima in the space of possible models. Disambiguating between these on the basis of error or parsimony criteria can be difficult in practice or even theoretically impossible, and it depends on the structure of model space and the nature of measurement noise that

are unique to each modeling problem. This is essentially a problem of model non-identifiability, and many studies have addressed this issue for ODEs in an optimization context where the model structure is fixed but parameters are free to vary [36, 61]. General statements can be made about the identifiability of systems of linear ODEs, although it is much more difficult to draw conclusions about the identifiability of arbitrary systems of nonlinear ODEs, especially when both the form and parameters of the model are free to vary [24, 35, 79, 81].

The simple examples considered here suggest that non-identifiability of both model structures and of parameter values play a role in shaping the fitness landscape. We proposed the initial value scanning approach as a means of systematically traversing one dimension of the landscape in an attempt to isolate regions in which the target model resides. In practice, this allowed us to recover the desired models much more often than would otherwise be possible. The primary downside of this approach, besides the computational effort involved, is that each trial in the scan produces a Pareto front of candidate models representing the best tradeoffs between model error and complexity found within that run. Independent trials of the scan often produced models of equivalent error and complexity, and distinguishing between these models would necessarily involve human effort and possibly domain knowledge. However, we argue that this type of post hoc human involvement in the modeling process is very different than the a priori involvement that characterizes knowledge-based modeling. The assessment of high-quality candidate models generated by GP would conceivably involve less human effort than the manual creation of a knowledge database that would then be provided to a knowledge-based modeling algorithm.

We tended to consider trials unsuccessful unless the evolved models precisely matched particular model forms and parameter values. However, in many cases trials that fail due to non-identifiability can still generate meaningful results. For example, in Sect. 4.2 we looked at the ideal rocket equations. The target model for this example was given in Eq. (16). The initial value scanning approach recovered the target model but also generated models such as:

$$\begin{cases} \frac{dX_1}{dt} = X_2 \\ \frac{dX_2}{dt} = \frac{2.0373}{X_3} \\ \frac{dX_3}{dt} = -1.0183 \end{cases} \tag{34}$$

with initial values $\hat{y}_1(0) = 2.0$, $\hat{y}_2(0) = 1.0$, $\hat{y}_3(0) = 5.0929$. Equation (34) has parameter values that differ greatly from the target in Eq. (16) and the predicted behavior of the hidden variable $X_3$ matches the behavior of $X_3$ in the target system very poorly. However, the parameter values in Eq. (34) are very close to 10 times the values in Eq. (16). If the target Eq. (16) uses units of meters for position ($X_1$), meters/second for velocity ($X_2$) and kg for mass ($X_3$), then Eq. (34) has nearly identical behavior in all variables if its units are simply reinterpreted as meters for $X_1$, meters/second for $X_2$, and kg $\cdot$ 10 for $X_3$. Although this is a relatively simple example, it shows that the essential scientific knowledge captured by a model can still be conveyed despite non-identifiability.

A second challenge to GP-based modeling with hidden variables is that the number and meaning of hidden variables would generally not be known in advance.

Here, we partially sidestepped this issue in our examples by assuming that the number of relevant variables in the system is known. This is not without justification as it would be possible to simply repeat any modeling trial with different numbers of hidden variables until the "correct" number of hidden variables was found. The correct number of hidden variables could be recognized as the number that resulted in the most accurate and parsimonious models. Alternatively, techniques from nonlinear dynamics theory such as correlation dimension analysis could be applied to estimate the relevant number of dynamic variables [1, 32, 43].

A third challenge unique to modeling with hidden variables is that the search space of possible models is much larger than other ODE modeling tasks. This is because the problem cannot be "partitioned" into simpler symbolic regression tasks [18]. For example, consider the problem of modeling a system of ODEs consisting of two variables $X_1$ and $X_2$. If both $X_1$ and $X_2$ are measured, $dX_1/dt$ and $dX_2/dt$ can be estimated and then a model for $dX_1/dt = f_1(t, X_1, X_2)$ can be found separately from a model for $dX_2/dt = f_2(t, X_1, X_2)$. The observed data for $X_1$ and $X_2$ is substituted as necessary on the right hand side of the equation to evaluate candidate models. This is not possible when hidden variables are present and the entire system must be integrated every time a candidate model is to be evaluated. As a result, the size of the search space scales linearly with the number of variables in the system when all variables are measured, but scales exponentially with the number of variables when one or more hidden variables are present. This could account for the difficulty encountered in inferring the more complex randomly generated systems in Sect. 4.6. It was our hope that some of this difficulty could be offset by simply performing fewer integration steps and more candidate model evaluations with the 8 All evaluation method, although the results indicate universally poorer performance of this method versus the 128 All and 8 Coev. methods.

Despite these challenges, we were able in many cases to reverse-engineer the form and parameters of target models. Success with the modest examples presented here could be extended in several ways. First, sophisticated parameter estimation and noise handling techniques could be incorporated directly into our GP algorithm [23, 60]. Although this could greatly increase the computational effort required to evaluate individual candidate models, the tradeoff might be desirable for many applications. Alternatively, additional parameter optimization could be applied as a post-processing, fine-tuning step for the evolved models. Techniques designed for multiresponse data might be particularly useful in this regard [63, 74]. Second, the traditional tree-based GP that we use here has been outperformed in many problem domains by linear GP [20], grammar-based GP [65, 85], and other evolutionary algorithms. Studying the performance of these alternative approaches when evolving ODE models in the presence of hidden variables is a promising area for future research. Finally, instead of drawing a sharp distinction between free-form and knowledge-based approaches to automated modeling as we did here, the two might be fruitfully combined. Such a hybrid approach could be one way of circumventing the issues with non-identifiability encountered in a pure free-form algorithm while improving the flexibility of a pure knowledge-based method.

Most of the dynamical systems considered in this work have qualitative properties that make them relatively amenable to modeling with the proposed

algorithm. However, dynamical systems encountered in the real world often exhibit behavior that could make modeling with a GP-based approach such as ours considerably more difficult. For example, physical systems and their corresponding ODE models are often studied in the neighborhood of singularities that could make it difficult to obtain the data measurements needed to evaluate the fitness of a candidate model [58]. Many systems exhibit a high sensitivity to one or more parameters in a region of interest, such as the laminar to turbulent flow transition boundary in a fluid [72]. Due to the large changes in observed behavior that result from small changes in parameter values, systems studied in the region of such bifurcation points could be particularly difficult to model with any approach that relies on a fitness gradient, including ours. As the algorithm described in the present work also relies on optimization of the initial values of ODEs in some cases, chaotic systems with high sensitivity to the choice of those initial values, such as those modeling weather, could further increase the complexity of the fitness landscape [51]. Finally, the degree of coupling between observed and hidden variables in a system may influence the ease with which the hidden variables can be modeled [17]. Complete decoupling would make modeling of hidden variables impossible, as shown in the chemical reaction network example of Sect. 4.3, whereas weak coupling could greatly increase the amount of observed data required to infer the algebraic forms describing any hidden variables. It will be important for future work in GP-based automated modeling of dynamical systems to address these complicating factors.

**Acknowledgments** Support was provided by the Tri-Institutional Training Program in Computational Biology and Medicine, National Science Foundation grant ECCS 0941561 on Cyber-enabled Discovery and Innovation (CDI), National Institutes of Health NIDA grant RC2 DA028981, and the U.S. Defense Threat Reduction Agency (DTRA) grant HDTRA 1-09-1-0013.

## References

1. F. Adachi, T. Washio, H. Motoda, Scientific discovery of dynamic models based on scale-type constraints. IPSJ Digit. Cour. **2**, 607–619 (2006)
2. L.A. Aguirre, U.S. Freitas, C. Letellier, J. Maquet, Structure-selection techniques applied to continuous-time nonlinear models. Physica D **158**, 1–18 (2001)
3. L.A. Aguirre, C. Letellier, Modeling nonlinear dynamics and chaos: a review. Math. Probl. Eng. (2009). doi:10.1155/2009/238960
4. D.P. Ahalpara, A. Sen, in *Proceedings of the EuroGP 2011*, ed. by S. Silva, J.A. Foster, M. Nicolau, P. Machado, M. Giacobini. A sniffer technique for an efficient deduction of model dynamical equations using genetic programming (Springer, Berlin, 2011), pp. 1–12
5. S. Ando, E. Sakamoto, H. Iba, Evolutionary modeling and inference of gene network. Inf. Sci. **145**(3–4), 237–259 (2002)
6. H. Andrew, in *Proceedings of the 2nd International Conference on Adaptive Computing in Engineering Design and Control*. System identification using genetic programming (University of Plymouth, Plymouth, 1996), pp. 57–62
7. P.J. Angeline, D.B. Fogel, in *Proceedings of the SPIE*. An evolutionary program for the identification of dynamical systems (1997), pp. 409–417
8. B. Argemí, J. Saurina, Study of the degradation of 5-azacytidine as a model of unstable drugs using a stopped-flow method and further data analysis with multivariate curve resolution. Talanta **74**, 176–182 (2007)
9. K.J. Åström, P. Eykhoff, System identification—a survey. Automatica **7**(2), 123–162 (1971)

10. E. Baake, M. Baake, H.G. Bock, K.M. Briggs, Fitting ordinary differential equations to chaotic data. Phys. Rev. A **45**(8), 5524–5529 (1992)
11. V. Babovic, M. Keijzer, in *Proceedings of the 4th International Conference on Hydroinformatics*. Evolutionary algorithms approach to induction of differential equations (Iowa City, USA, 2000), pp. 251–258
12. R. Bakker, J.C. Schouten, C.L. Giles, F. Takens, C.M. van den Bleek, Learning chaotic attractors. Neural Comput. **12**(10), 2355–2383 (2000)
13. W. Banzhaf, P. Nordin, R.E. Keller, F.D. Francone, *Genetic Programming: An Introduction on the Automatic Evolution of Computer Programs and Its Applications* (Morgan Kaufmann, San Francisco, 1997)
14. J.A. Beisler, Isolation, characterization, and properties of a labile hydrolysis product of the antitumor nucleoside, 5-azacytidine. J. Med. Chem. **21**(2), 204–208 (1978)
15. H.S. Bernardino, H.J.C. Barbosa, Inferring systems of ordinary differential equations via grammar-based immune programming. LNCS **6825**, 198–211 (2011)
16. C.M. Bishop, *Pattern Recognition and Machine Learning* (Springer, New York, 2006)
17. S. Boccaletti, *The Synchronized Dynamics of Complex Systems* (Elsevier, Amsterdam, 2008)
18. J. Bongard, H. Lipson, Automated reverse engineering of nonlinear dynamical systems. PNAS **104**(24), 9943–9948 (2007)
19. E. Bradley, M. Easley, R. Stolle, Reasoning about nonlinear system identification. Artif. Intell. **133**, 139–188 (2001)
20. M. Brameier, W. Banzhaf, *Linear Genetic Programming* (Springer, New York, 2007)
21. J.L. Breeden, A. Hübler, Reconstructing equations of motion from experimental data with unobserved variables. Phys. Rev. A **42**(10), 5817–5826 (1990)
22. W. Bridewell, P. Langley, L. Todorovski, S. Dzeroski, Inductive process modeling. Mach. Learn. **71**, 1–32 (2008)
23. H. Cao, L. Kang, Y. Chen, J. Yu, Evolutionary modeling of systems of ordinary differential equations with genetic programming. Genet. Program Evolvable Mach. **1**(4), 309–337 (2000)
24. O.-T. Chis, J.R. Banga, E. Balsa-Canto, Structural identifiability of systems biology models: a critical comparison of methods. PLoS ONE **6**(11), e27755 (2011)
25. K.-H. Cho, S.-Y. Shin, H.-W. Kim, O. Wolkenhauer, B. McFerran, W. Kolch, Mathematical modeling of the influence of RKIP on the ERK signaling pathway. LNCS **2602**, 127–141 (2003)
26. D. Clery, D. Voss, All for one and one for all. Science **308**, 809 (2005)
27. J. Cremers, A. Hübler, Construction of differential equations from experimental data. Zeitschrift für Naturforschung A **42**(8), 797–802 (1987)
28. J.P. Crutchfield, B.S. McNamara, Equations of motion from a data series. Complex Syst. **1**(3), 417–452 (1987)
29. S. Džeroski, L. Todorovski, in *Proceedings of the 10th International Conference on Machine Learning*. Discovering dynamics (Kaufmann, Amherst, 1993), pp. 97–103
30. J. Evans, A. Rzhetsky, Machine science. Science **329**, 3994400 (2010)
31. G. Gouesbet, Reconstruction of the vector fields of continuous dynamical systems from numerical scalar time series. Phys. Rev. A **43**(10), 5321–5331 (1991)
32. P. Grassberger, I. Procaccia, Measuring the strangeness of strange attractors. Physica D **9**, 189–208 (1983)
33. G.J. Gray, D.J. Murray-Smith, Y. Li, K.C. Sharman, in *Proceedings of the 2nd International Conference on Genetic Algorithms in Engineering Systems: Innovations and Applications*. Nonlinear model structure identification using genetic programming (Inst of Electrical Engineers, London, 1997), pp. 308–313
34. G.J. Gray, D.J. Murray-Smith, Y. Li, K.C. Sharman, T. Weinbrenner, Nonlinear model structure identification using genetic programming. Control Eng. Pract. **6**, 1341–1352 (1998)
35. M.S. Grewal, K. Glover, Identifiability of linear and nonlinear dynamical systems. IEEE Trans. Automat. Control **21**(6), 833–837 (1976)
36. S. Hengl, C. Kreutz, J. Timmer, T. Maiwald, Data-based identifiability analysis of non-linear dynamical models. Bioinformatics **23**(19), 2612–2618 (2007)
37. A.L. Hodgkin, A.F. Huxley, A quantitative description of membrane current and its application to conduction and excitation in nerve. J. Phys. **117**, 500–544 (1952)
38. X. Hong, R.J. Mitchell, S. Chen, C.J. Harris, K. Li, G.W. Irwin, Model selection approaches for non-linear system identification: a review. Int. J. Syst. Sci. **39**(10), 925–946 (2008)

39. H. Iba, H. de Garis, T. Sato, in *Proceedings of the PPSN III*, ed. by Y. Davidor, H.-P. Schwefel, R. Männer. Genetic programming with local hill-climbing (Springer, London, 1994), pp. 302–311
40. H. Iba, Inference of differential equation models by genetic programming. Inf. Sci. **178**(23), 4453–4468 (2008)
41. S.J. Julier, J.K. Uhlmann, in *Proceedings of the International Symposium on Aerospace/Defense Sensing, Simulation, and Controls.* A new extension of the Kalman filter to nonlinear systems (1997), pp. 182–193
42. M.A. Kaboudan, Genetic programming prediction of stock prices. Comput. Econ. **16**(3), 207–236 (2000)
43. M.B. Kennel, R. Brown, H.D. Abarbanel, Determining embedding dimension for phase-space reconstruction using a geometrical construction. Phys. Rev. A **45**(6), 3403–3411 (1992)
44. R.D. King, K.E. Whelan, F.M. Jones, P.G.K. Reiser, C.H. Bryant, S.H. Muggleton, D.B. Kell, S.G. Oliver, Functional genomic hypothesis generation and experimentation by a robot scientist. Nature **427**, 247–252 (2004)
45. L.D. Kissinger, N.L. Stemm, Determination of the antileukemia agents cytarabine and azacitidine and their respective degradation products by high-performance liquid chromatography. J. Chromatogr. **353**, 309–318 (1986)
46. J. Koza, *Genetic Programming: On the Programming of Computers by Means of Natural Selection* (MIT Press, Cambridge, 1992)
47. P. Langley, Data-driven discovery of physical laws. Cognitive Sci. **5**, 31–54 (1981)
48. P. Langley, H.A. Simon, G.L. Bradshaw, J.M. Zytkow, *Scientific Discovery: Computational Explorations of the Creative Processes* (MIT Press, Cambridge, 1987)
49. P. Langley, J. Sanchez, L. Todorovski, S. Džeroski, in *Proceedings of the 19th International Conference on Machine Learning*. Inducing process models (Morgan Kaufmann, Sydney, 2002), pp. 347–354
50. L. Ljung, *System Identification—Theory For the User*, 2nd edn. (Prentice Hall, Upper Saddle River, NJ, 1999)
51. E.N. Lorenz, Deterministic non-periodic flow. J. Atmos. Sci. **20**(2), 130–141 (1963)
52. G. Mamani, J. Becedas, V.F. Batlle, H. Sira-Ramírez, Algebraic observer to estimate unmeasured state variables of DC motors. Eng. Lett. **16**(2), 248–255 (2008)
53. D.J. Montana, Strongly typed genetic programming. Evol. Comput. **3**(2), 199–230 (1995)
54. B.S. Mulloy, R.L. Riolo, R.S. Savit, in *Proceedings of the GECCO 1996*. Dynamics of genetic programming and chaotic time series prediction (MIT Press, Cambridge, MA, 1996), pp. 166–174
55. C. Neely, P. Weller, R. Dittmara, Is technical analysis in the foreign exchange market profitable? A genetic programming approach. J. Financ. Quant. Anal. **32**, 405–426 (1997)
56. H. Oakley, Two scientific applications of genetic programming: stack filters and non-linear equation fitting to chaotic data, in *Advances Genetic Programming*, ed. by K.E. Kinnear Jr (MIT Press, Cambridge, MA, 1994), pp. 369–389
57. N.H. Packard, J.P. Crutchfield, J.D. Farmer, R.S. Shaw, Geometry from a time series. Phys. Rev. Lett. **45**(9), 712–716 (1980)
58. S.N. Pnevmatikos (ed.), *Singularities & Dynamical Systems* (Elsevier, Amsterdam, 1985)
59. W.H. Press, B.P. Flannery, S.A. Teukolsky, W.T. Vetterling, *Numerical Recipes: The Art of Scientific Computing*, 3rd edn. (Cambridge University Press, Cambridge, 2007)
60. L. Qian, H. Wang, E.R. Dougherty, Inference of noisy nonlinear differential equation models for gene regulatory networks using genetic programming and Kalman filtering. IEEE Trans. Signal Process. **56**(7), 3327–3339 (2008)
61. A. Raue, V. Becker, U. Klingmüller, J. Timmer, Identifiability and observability analysis for experimental design in nonlinear dynamical models. Chaos **20**, 045105 (2010)
62. K. Rodriguez-Vázquez, P.J. Fleming, Evolution of mathematical models of chaotic systems based on multiobjective genetic programming. Knowl. Inf. Syst. **8**(2), 235–256 (2005)
63. K. Routray, G. Deo, Kinetic parameter estimation for a multiresponse nonlinear reaction model. AIChE J. **51**(6), 1733–1746 (2005)
64. S. Russell, P. Norvig, *Artificial Intelligence: A Modern Approach*, 3rd edn. (Prentice Hall, Upper Saddle River, NJ, 2009)
65. C. Ryan, J.J. Collins, M. O'Neill, in *Proceedings of the EuroGP 1998*, ed. by W. Banzhaf, R. Poli, M. Schoenauer, T.C. Fogarty. Grammatical evolution: evolving programs for an arbitrary language (Springer, London, 1998), pp. 83–96

66. E. Sakamoto, H. Iba, in *Proceedings of the CEC 2001*. Inferring a system of differential equations for a gene regulatory network by using genetic programming (2001), pp. 720–726
67. M.D. Schmidt, H. Lipson, Coevolution of fitness predictors. IEEE Trans. Evol. Comput. **12**(6), 736–749 (2008)
68. M.D. Schmidt, H. Lipson, Distilling free-form natural laws from experimental data. Science **324**(5923), 81–85 (2009)
69. M.D. Schmidt, H. Lipson, in *Proceedings of the GECCO 2010*. Age-fitness Pareto optimization (2010), pp. 543–544
70. M.D. Schmidt, H. Lipson, Age-fitness Pareto optimization. Genet. Program. Theory Pract. **8**, 129–146 (2010)
71. M. Schwabacher, P. Langley, in *Proceedings of the ICML 2001*, ed. by C.E. Brodley, A.P. Danyluk. Discovering communicable scientific knowledge from spatio-temporal data (Morgan Kaufmann Publishers Inc., San Francisco, CA, 2001), pp. 489–496
72. K.V. Sharp, R.J. Adrian, Transition from laminar to turbulent flow in liquid filled microtubes. Exp. Fluids **36**(5), 741–747 (2004)
73. J. Sjöberg, Q. Zhang, L. Ljung, A. Benveniste, B. Delyon, P.-Y. Glorennec, H. Hjalmarsson, A. Juditsky, Nonlinear black-box modeling in system identification: a unified overview. Automatica **31**(12), 1691–1724 (1995)
74. W.E. Stewart, M. Caracotsios, J.P. Sørensen, Parameter estimation from multiresponse data. AIChE J. **38**(5), 641–650 (1992)
75. S.H. Strogatz, Exploring complex networks. Nature **410**, 268–276 (2001)
76. A. Szalay, J. Gray, 2020 Computing: science in an exponential world. Nature **440**, 413–414 (2006)
77. F. Takens, Detecting strange attractors in turbulence. Lect. Notes Math. **898**, 366–381 (1981)
78. L. Todorovski, S. Džeroski, in *Proceedings of the ICML 1997*. Declarative bias in equation discovery (Kaufmann, Nashville, 1997), pp 376–384
79. S. Vajda, H. Rabitz, Identifiability and distinguishability of general reaction systems. J. Phys. Chem. **98**, 5265–5271 (1994)
80. H.U. Voss, J. Timmer, J. Kurths, Nonlinear dynamical system identification from uncertain and indirect measurements. Int. J. Bifurcat. Chaos **14**(6), 1905–1933 (2004)
81. É. Walter, L. Pronzato, *Identification of Parametric Models From Experimental Data* (Springer, New York, 1997)
82. D. Waltz, B.G. Buchanan, Automating science. Science **324**(5923), 43–44 (2009)
83. T. Washio, H. Motoda, Y. Niwa, in *Proceedings of the ICML 2000*. Enhancing the plausibility of law equation discovery (2000), pp. 1127–1134
84. H.-L. Wei, S.A. Billings, Model structure selection using an integrated forward orthogonal search algorithm assisted by squared correlation and mutual information. Int. J. Model. Identif. Control **3**(4), 341–356 (2008)
85. P.A. Whigham, in *Proceedings of the Workshop on Genetic Programming: From Theory to Real-World Applications*, ed. by J.P. Rosca. Grammatically-based genetic programming (Tahoe City, California, 1995), pp. 33–41
86. S. Winkler, M. Affenzeller, S. Wagner, New methods for the identification of nonlinear model structures based upon genetic programming techniques. J. Syst. Sci. **31**, 5–14 (2005)
87. T. Yoshida, L.E. Jones, S.P. Ellner, G.F. Fussmann, N.G. Hairston Jr, Rapid evolution drives ecological dynamics in a predator-prey system. Nature **424**, 303–306 (2003)