Available online at www.sciencedirect.com

ScienceDirect

IFAC PapersOnLine 54-7 (2021) 379–384

Toolbox for Discovering Dynamic System Relations via TAG Guided Genetic Programming

Ştefan-Cristian Nechita * Roland Tóth *,** Dhruv Khandelwal * Maarten Schoukens *

* *Department of Electrical Engineering Eindhoven University of Technology, Eindhoven University of Technology, Eindhoven, The Netherlands ({s.c.nechita, r.toth, d.khandelwal, m.schoukens}@tue.nl).*
** *Systems and Control Laboratory, Institute for Computer Science and Control, Kende u. 13-17, H-1111 Budapest, Hungary.*

Abstract: Data-driven modeling of nonlinear dynamical systems often requires an expert user to take critical decisions a priori to the identification procedure. Recently, an automated strategy for data driven modeling of *single-input single-output* (SISO) nonlinear dynamical systems based on *genetic programming* (GP) and *tree adjoining grammars* (TAG) was introduced. The current paper extends these latest findings by proposing a *multi-input multi-output* (MIMO) TAG modeling framework for polynomial NARMAX models. Moreover, we introduce a TAG identification toolbox in Matlab that provides implementation of the proposed methodology to solve multi-input multi-output identification problems under NARMAX noise assumption. The capabilities of the toolbox and the modeling methodology are demonstrated in the identification of two SISO and one MIMO nonlinear dynamical benchmark models.

Keywords: Nonlinear system identification, Equation discovery, Tree Adjoining Grammar, Genetic Programming, Data-driven system modeling

1. INTRODUCTION

Control design for complex dynamic systems relies heavily on accurate system models. A way to obtain such models is through first principle modeling. While this method provides generic models with clear physical interpretation, it requires a considerable amount of time and user expertise. Another way to model the dynamical behaviour of a system is through data-driven system identification. Within this field there are numerous methods that require the user to take critical decisions (e.g. precisely selecting the model structure within *prediction error methods* (PEM)). In contrast, the machine learning strategies can automatically select or define model structures and features. Non-parametric machine learning methods for data-driven modeling such as *Gaussian Process* based Bayesian Estimators (Pillonetto et al., 2014), *support vector machines* (SVM) (Ming-guang Zhang et al., 2004) and *artificial neuron networks* (ANN), (Goodfellow et al., 2016), (Billings, 2013) describe large model spaces that can represent complex MIMO dynamics. However, the obtained models via these methods often lack interpretability and fail to provide generalization to unseen data or unseen operating regions of the system. On the other hand, the parametric machine learning methods, also known as symbolic regression, such as *tree adjoining grammar guided genetic programming* (TAG3P) (Khandelwal, 2020), and *equation discovery* (EQ) (Patelli and Ferariu, 2009) perform automated structure selection and yield time-domain solutions that directly represent the temporal modes of the system. In his doctoral thesis (Khandelwal, 2020), the author proposes a convenient way of defining the model set searching space through a novel TAG modeling framework and conveys the critical decision of selecting the right model structure into an automated evolutive procedure based on Genetic Programming. Moreover, this work shows how the proposed method can discover physical relation directly from data (Duffing oscilator). This latest development with respect to the modeling framework focused on the single-input single-output (SISO) polynomial NARMAX model set, but also included a considerable amount of variation (e.g. ability to embed sin(·), cos(·) or abs(·) nonlinear operators and TAG representation of Box-Jenkins models).

The current paper focuses on a novel grammar that extends the TAG modeling framework to multi-input multi-output (MIMO) polynomial NARMAX models. It is common for dynamic systems to have output channels with coupled dynamics. Our main contribution is to extending the framework such as multi-output candidate models are represented by only one compact syntactic tree. By this, the dynamic modes are created, evolved and parametrized with respect to all output signals at once, thus considering output couplings. Moreover, as our second contribution, a Matlab toolbox for identification with the TAG framework is provided. Using the toolbox, the user can easily select the structure searching space in terms of NARMAX (sub) model set(s). We have validated the modeling framework and Matlab implementation on two SISO and one MIMO

[⋆] This research was supported by the Ministry of Innovation and Technology NRDI Office within the framework of the Artificial Intelligence National Laboratory Program and by the Dutch Organization for Scientific Research (NWO, domain TTW, grant: 13852) which is partly funded by the Ministry of Economic Affairs of The Netherlands.

10.1016/j.ifacol.2021.08.389

Table 1. Sub model sets included in G_{NARMAX}.

Sub model	Grammar	Elementary trees
Input Poly.	G_{IP}	$\beta_1, \beta_4, \alpha_1$
LTI	G_{LTI}	$\beta_1, \beta_2, \beta_7, \alpha_1$
poly-NARX	G_{NARX}	$\beta_1, \beta_2, \beta_4, \beta_5, \beta_7, \alpha_1$
ext-NARX	G_{extNARX}	$\beta_1, \beta_2, \beta_4, \beta_5, \beta_7, \beta_8, \alpha_{1,2,3,4}$
exp-NARX	G_{expNARX}	$\beta_1, \beta_2, \beta_4, \beta_5, \beta_7, \beta_8, \alpha_{1,5,6}$

nonlinear benchmark models.

The paper is structured as follows: Section 2 details the novel TAG modeling framework. Section 3 describes the optimisation approach that drives the automated GP structure search procedure that is introduced in Section 4. Section 6 shows the identification results of several benchmark models. Section 5 describes the developed Matlab toolbox. In Section 7 we draw conclusions on our results and present several future research directions.

2. MODEL STRUCTURE VIA TAG

The symbolic regression identification problem consists of determining an appropriate dynamic structure and corresponding parameters of a data generating system. The solution space is described as $\mathcal{S} = \mathcal{W} \times \mathcal{P}$, where $\mathcal{W}$ is the a priori defined space of dynamic structures and $\mathcal{P} \in \mathbb{R}^n$ is its assigned parameter space, with n arbitrary large, but finite. Hence, naturally, a dual-optimization problem arises in the sense that finding the dynamic structure withing space $\mathcal{W}$ that minimizes the output error implies solving the parameters estimation problem within space $\mathcal{P}$. For the proposed identification approach, TAG is used to describe the structure space $\mathcal{W}$. This chapter presents briefly the TAG modeling framework followed by a novel grammar proposal for MIMO polynomial NARMAX models. For a complete definition see (Kallmeyer, 2009) and (Khandelwal, 2020). In short, a candidate model described by a TAG can be seen as an orientated graph encoded by its derived tree γ, that has a root node v_{r}, edges to its intermediate nodes v_{int} and leaves v_{l} all arranged in a purely (one to many) top to bottom fashion. The derived tree γ is constructed based on its derivation tree Γ_γ. The latter is formed by orientated (ordered) connections of elementary trees (β and α). The elementary trees are the "building blocks" of any TAG tree structure. In the case of system identification, they correspond to elementary algebraic operations for signals, applying time operators such as: time shift (e.g. β_7) and elementary nonlinear functions such as β_8 with $\alpha_1 \dots \alpha_6$. Alongside with label sets, the elementary tree forms a TAG G. The structure of the elementary trees, imposed by the choice and position of the intermediate nodes v_{int} (e.g. $expr_{0\dots3}$), defines the rules that a certain grammar imposes over the shape of the derived trees γ (i.e. it defines what a model set is that can be generated from the recursive application of elementary operations by connecting trees via TAG adjunction and substitution operators). Each such derived tree γ represents a function $\mathcal{F}_\gamma$ via an interpreter function $\mathcal{E}(\gamma)$ that transposes the tree structure into the mathematical function $\mathcal{F}$. In our context, the design of the elementary trees defines the TAG language $\mathfrak{L}(G)$ (all the trees γ that can be generated) thus, it directly defines the model set where $\mathcal{F}_\gamma = \mathcal{E}(\gamma)$ represents a model structure. Therefore, elementary trees can be designed such that a TAG can represent, via its language, an entire model set. TAGs are highly valuable as they allow to encode valid model representations and can seriously increase efficiency of GP based system identification as detailed in (Khandelwal, 2020).

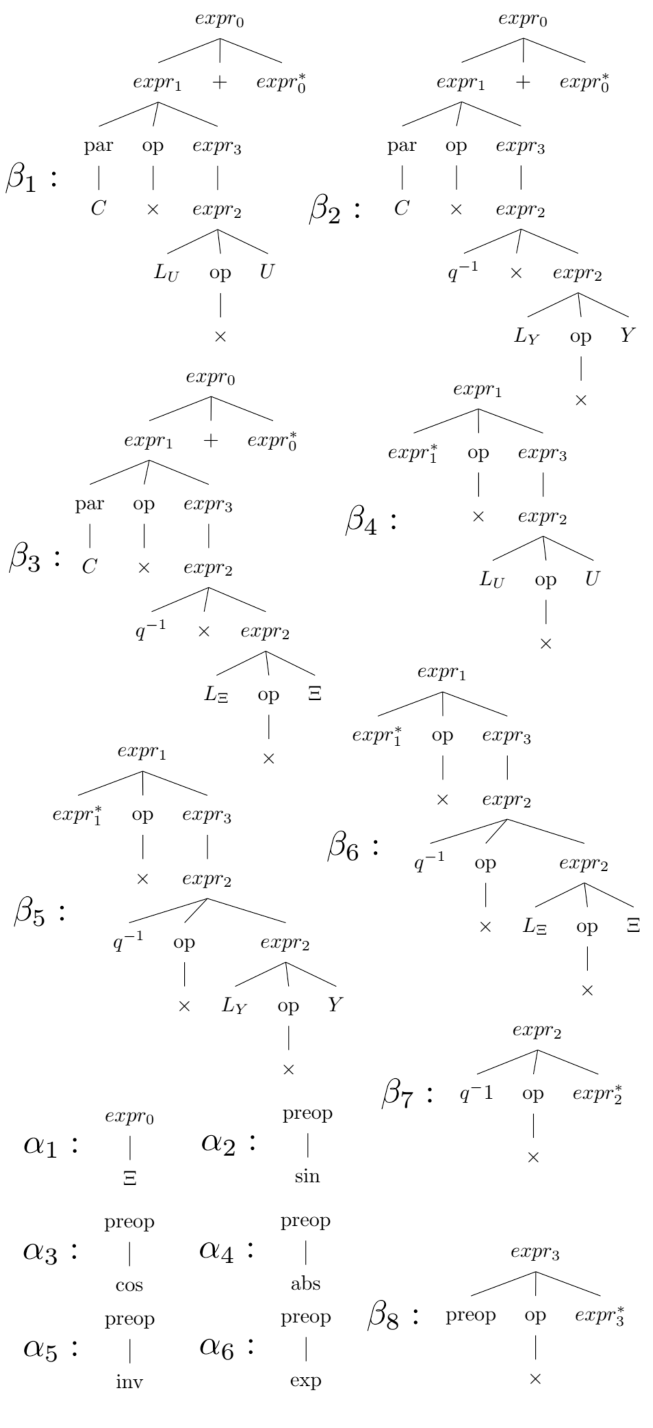

Fig. 1. Elementary trees $I \cup A$ of the extented G_{NARMAX}

2.1 TAG p-NARMAX modeling framework

Within this paper we focus on discrete-time MIMO polynomial NARMAX model set. Such a noise structure often provides enough flexibility to represent many dynamic systems in practice. Further, we consider systems of the form:

$$Y(k) = \mathcal{F}(\{u_i(k-j)\}_{j=1}^{n_u}, \{y_i(k-m)\}_{m=1}^{n_y}, \\ \{\xi_i(k-l)\}_{l=1}^{n_s}, i \in \mathrm{r}_{\{u,y,\xi\}}) \tag{1}$$

where $U(k)$, $Y(k)$ and $\Xi(k)$ are multi-channel input, output and process noise signals respectively with dimensions $\mathrm{r}_{\{u,y,\xi\}} \times k$, $\mathrm{r}_{\{u,y,\xi\}} \in \mathbb{N}$ and n_u, n_y and n_ξ are finite discrete time-delays with $n_u, n_\xi \in \mathbb{N} \cup \{0\}$, $n_y \in \mathbb{N}$ and $k \in \{1 \ldots \mathrm{N}\}$ finite number of time samples. If the case (1) is restricted to polynomial relations, a suitable way to represent (1) for TAG modeling framework, is as follows:

$$\begin{aligned} Y(k) = \sum_{i=1}^{p} C_i \prod_{j=0}^{q_u} \prod_{s_u}^{b_{i,j}} L_{U,i,j} U(k-j) \times \\ \prod_{m=1}^{q_y} \prod_{s_y}^{a_{i,m}} L_{Y,i,m} Y(k-m) \prod_{l=1}^{q_\xi} \prod_{s_\xi}^{d_{i,l}} L_{\Xi,i,l} \Xi(k-l) + \Xi(k) \end{aligned} \tag{2}$$

where $L_{\{U,Y,\Xi\}}$ is a so called *linking array* defined as:

$$\begin{aligned} & L_X \in \mathbb{R}^{1\times \mathrm{r}}, \mathrm{r} = \dim(X), L = [\mathrm{l}_i]_{i=1}^{\mathrm{r}}, \mathrm{l}_i \in \{0,1\} \\ & L_X \neq 0_{1\times \mathrm{r}} \end{aligned} \tag{3}$$

and $p \in \mathbb{N}$. The operation: $\prod_{s=1}^{g_i} L_{X,i,s} X(k-i)$ is defined as a right hand side matrix multiplication with $\prod_{s=1}^{0} L_{X,i,s} X(k-i) = 1$, where $X(k-i)$ is the value of signal X at time moment $k-i$, s is a selector operator counter, $L_{X,i,s}$ is a random linking array generated by (3) and g_i is the amount of right hand side multiplication of $X(k-i)$ with itself (e.g. $X(k-i)^{g_j}$). The form (2) can represent polynomial terms of all elements of the involved signals and their time-shifted representatives $u_i(k-j)$, $y_i(k-m)$ and $\xi_i(k-l)$. A given function $\mathcal{F}(\cdot)$ within the model set (2) can be represented by a derived tree γ.

Proposition 1. TAG for MIMO p-NARMAX models
Let $G_{\mathrm{NARMAX}} = \langle N, T, S, I, A \rangle$ be a TAG with

- $N = \{expr_0, expr_1, expr_2, \mathrm{op}, \mathrm{par}\}$,
- $T = \{U, Y, \Xi, +, C, \times, q^{-1}, L_Y, L_U, L_\Xi\}$, where L_Y, L_U and L_Ξ are "linking arrays", U, Y, Ξ are the input, output and output noise signals and C the is parameters vector.
- $S = \{expr_0\}$,
- $I = \{\alpha_1\}$,
- $A = \{\beta_1, \beta_2, \beta_3, \beta_4, \beta_5, \beta_6, \beta_7\}$, where the elementary trees β_i and α_1 are depicted in Figure 1.

The model set $M(G_{\mathrm{NARMAX}})$ represents the set of all polynomial models defined by Equation (2) with $p, n_y, n_\xi \in \mathbb{N}$ and $n_u \in \mathbb{N} \cup \{0\}$.

Proposition 1 represents our main contribution over the TAG based modeling framework. As described in (Khandelwal, 2020), the TAG that represents the polynomial NARMAX model set can be enhanced or extended by considering sin(·), cos(·), abs(·), inv(·) and exp(·) functions over the polynomial variables enlisted above. This modeling extension is enabled in the TAG modeling framework by considering the β_8 auxiliary tree and $\alpha_{2\ldots6}$ initial trees depicted in the lower part of Figure 1. Similarly other functions can be added. Moreover, sub model sets included in G_{NARMAX} can be considered by selecting specific constituent elementary trees. Further extensions to the existed noise structure can be directly achieved as discussed in (Khandelwal, 2020) by extending the elementary trees with further elements over the noise structure. A list of useful model sets is shown in Table 1. An example of model set with TAG specification is given in: `TAG3P_Call_Example.m`.

3. IDENTIFICATION PROBLEM

Given a flexible model structure we would like to obtain an estimate of the underlying data generating system by finding a structure form with adequate complexity to achieve a desired level of approximation. This minimization can be formally defined as a dual optimization problem. Consider a TAG G_{Model} and its equivalent model set $\mathcal{W}_{\mathrm{Model}}$ and a data generating system $\mathcal{F}_{\gamma_0}(\theta_0)$ described by a tree $\gamma_0 \in \mathfrak{L}(G_{\mathrm{Model}})$, its tree-based equivalent model structure w_{γ_0} with the real parameters θ_0 that yield the real output sequence $Y_0(w_{\gamma_0}|\theta_0, D_{\mathrm{N}}) = Y_0(k)$, where $D_{\mathrm{N}} = \{U(k), Y_0(k)\}_{k=1}^{\mathrm{N}}$ is a data set of length N with $U(k)$ input sequence and $Y_0(k)$ stochastic response. Let $\mathcal{F}_{\hat{\gamma}}(\hat{\theta}) = \mathcal{E}(\hat{\gamma})$ be a candidate model represented by tree $\hat{\gamma}$, its equivalent model structure w_γ and its assigned set of parameters $\hat{\theta}$. For the data set D_{N}, the model $\mathcal{F}_{\hat{\gamma}}(\hat{\theta})$ yields the one step ahead prediction response $\hat{Y}_{\mathrm{p}}(w_{\hat{\gamma}}|\hat{\theta}, D_{\mathrm{N}}) = \hat{Y}_{\mathrm{p}}(k)$ and simulation response $\hat{Y}_{\mathrm{s}}(w_{\hat{\gamma}}|\hat{\theta}, D_{\mathrm{s,N}}) = \hat{Y}_{\mathrm{s}}(k)$, where $D_{\mathrm{s,N}} = \{U(k), \hat{Y}_{\mathrm{s}}(k)\}_{k=1}^{\mathrm{N}}$. The two responses generate an error score $E = (E_{\mathrm{s}}, E_{\mathrm{p}}) \in \mathbb{R}^2$ where E_{s} is the root mean square simulation error ($\mathrm{RMS_s}$) produced by $\hat{Y}_{\mathrm{s}}(k)$ and E_{p} is the root mean square prediction error ($\mathrm{RMS_p}$) produced by $Y_{\mathrm{p}}(k)$. The main aim of the identification strategy is to minimize the error score E. Minimizing both errors is requested for reliable generalization property of the obtained models. (Khandelwal et al., 2019) Therefore the identification procedure searches for the solution of the following dual optimization problem:

$$\begin{aligned} & \min_{w_\gamma} \qquad J(w_\gamma, \hat{\theta}) = \min\left(E\left(w_\gamma, \hat{\theta}\right)\right) \\ & \text{s.t.} \\ & \hat{\theta} = \min_{\theta} \; J_{\mathrm{sub}}(\theta) = \quad \omega_{\mathrm{s}} E_{\mathrm{s},\tau}(\theta) + \omega_{\mathrm{p}} E_{\mathrm{p}}(\theta) \end{aligned} \tag{4}$$

$$E_{\mathrm{s}}(\theta) = \tfrac{1}{\mathrm{r}_y} \sum_{\mathrm{i}=1}^{\mathrm{r}_y} \sqrt{\tfrac{1}{\mathrm{N}} \mathrm{e}_{\mathrm{i,s}}^\top \mathrm{e}_{\mathrm{i,s}}}, \;\; E_p(\theta) = \tfrac{1}{\mathrm{r}_y} \sum_{\mathrm{i}=1}^{\mathrm{r}_y} \sqrt{\tfrac{1}{N} \mathrm{e}_{\mathrm{i,p}}^\top \mathrm{e}_{\mathrm{i,p}}} \tag{5}$$

where

$$\mathrm{e}_{\mathrm{i},\{\mathrm{s,p}\}} \; = [y_{0,\mathrm{i}}(k) - \hat{y}_{\mathrm{i},\{\mathrm{s,p}\}}(w_\gamma, k|\hat{\theta}, D_{\mathrm{N}})]_{k=1}^{\mathrm{N}}, \tag{6}$$

ω_{s} is the simulation error weight and ω_{p} is the prediction error weight. The weight values play a role in determining what parameter estimation procedure can be deployed to solve the sub-optimization problem. They will be further detailed later.

4. ESTIMATION VIA GENETIC PROGRAMMING

To solve the multi-objective dual optimization problem described above, we designed and implemented a *genetic programming* (GP) algorithm that evolves a population of tree structures through TAG designed crossover and mutation genetic operators, perform parameter estimation for each structure and sorts each generation based on two fitness criterion $\mathrm{RMS_s}$ and $\mathrm{RMS_p}$ using the multi-objective non-dominating sorting algorithm.

4.1 Main Algorithm

The main steps of the GP algorithm are presented in Algorithm 1. The GP is initialized by defining the genetic parameters: population size (Pop), number of generations (Gen), number of maximum auxiliary trees that can be used in each derivation tree (Complexity) and crossover parameter ($\mu \in [0 - 100\%]$). The genetic evolution starts from an initial population of randomly generated trees G(1). Inside the iterative loop, the crossover, mutation,

Algorithm 1 TAG GP main

```
Define Pop                                   ▷ Define Population Size
Define Complexity                            ▷ Define maximum complexity
Define Gen              ▷ Define the maximum number of generations
G(1) $\leftarrow$ RandomPopulation ▷ Generate a random population of trees
G(1) $\leftarrow$ Interpreter(G(1))          ▷ Construct the candidate model
G(1) $\leftarrow$ ParameterEstimation(G(1))
G(1) $\leftarrow$ Evaluate(G(1))             ▷ Compute $E_\mathrm{s}$ and $E_\mathrm{p}$ for G(1)
while $i \leq$ Gen do
    Q$_1$ $\leftarrow$ CrossoverOffsprings(G(i))        ▷ Card(Q$_1$) = Pop
    Q$_2$ $\leftarrow$ MutationOffsprings(G(i))         ▷ Card(Q$_1$) = Pop
    Q$_{1,2}$ $\leftarrow$ Interpreter(Q$_{1,2}$)       ▷ see CreateTreeFunction.m
    Q$_{1,2}$ $\leftarrow$ ParameterEstimation(Q$_{1,2}$)
    Q$_{1,2}$ $\leftarrow$ Evaluate( Q$_{1,2}$)         ▷ Compute $E_\mathrm{s}$ and $E_\mathrm{p}$ for Q$_{1,2}$
    R $\leftarrow$ G(i) $\cup$ Q$_1$ $\cup$ Q$_2$
    R $\leftarrow$ NSGA-II(R)                           ▷ Sorting R into Pareto fronts
    G(i + 1) $\leftarrow$ R(1 : Pop)  ▷ Select the first Pop candidates from the
                                                        first Pareto fronts of R
end while
Save G(Gen)                                  ▷ collect the Pareto solution
```

interpreter function, parameter estimation, evaluation and non-dominating sorting procedures are executed sequentially in order to propose, construct, evaluate and sort new dynamical structures. At the end, the solution is considered to be the first Pareto front of the last generation. Since within the Pareto solution the models do not dominate each other, in terms of the two considered fitness criterion, any of them can be selected as a final candidate model that minimizes problem (4). Next we will explain the main procedures in detail.

4.2 Crossover and Mutation genetic operators

In Crossover, two parents (individuals of population) have their genotype combined in order to form new individuals called offsprings. Through crossover, no new information is added to the population. By switching strings of genotype between individuals, over generations, the genes that yield smaller fitness values tend to become more frequent in the population. In this way, a local exploration of the search space is performed. Consequently, via crossover, a population is exploring a local minimum of the cost score surface. The crossover operator is defined within the description of TAG3P+ in (Hoai et al., 2003). In Mutation, an offspring is proposed by eliminating or adjoining elementary trees starting from a derivation tree $\Gamma \in \mathrm{G}(i)$. In our implementation, for each structure of $\mathrm{G}(i)$ an offspring is created by mutation. By random addition or deletion of elementary trees to or from the parent derivation tree, the mutation operator is the procedure through which the evolution process performs global exploration of the searching space. Both crossover and mutation functions are called within `TAG_GP_Step1.m`.

4.3 Parameter estimation procedures

Every model constructed through crossover, mutation and random generation requires optimization of its parameters to assess its accuracy in terms of (4). The parameter estimation can be performed with respect to both simulation and prediction error (non zero ω_s and ω_p weights) or only prediction error ($\omega_\mathrm{s} = 0$ and $\omega_\mathrm{p} = 1$) . Considering both $\mathrm{RMS_s}$ and $\mathrm{RMS_p}$ in parameter estimation transforms the sub optimization problem into a non-convex optimization problem, making it considerable difficult and time-consuming to solve. If only the prediction error is considered, any model defined by a function $\mathcal{F}_\gamma$ with $\gamma \in \mathfrak{L}(G_\mathrm{NARMAX})$ can be rewritten as (7)

$$\Psi = \Phi\Theta + E_\Theta \tag{7}$$

where, for p polynomial terms as described in (2), $\hat{\Psi} \in \mathbb{R}^{\mathrm{N}\times n_y}$ is the model output data set, $\Phi \in \mathbb{R}^{\mathrm{N}\times p}$ is the evolution of each polynomial term over D_N and $\Theta \in \mathbb{R}^{p\times n_y}$ is the matrix corresponding to the parameter vector Θ. The set of parameters that minimize the sub optimization problem (4) is computed as

$$\hat{\Theta} = \left(\Phi^\top\Phi\right)^{-1}\Phi^\top\Psi. \tag{8}$$

In the toolbox, the parameter estimation procedure is called inside the main loop in `TAG_GP_Step2.m`. Moreover, the toolbox user has the option to choose between three parameter estimation procedures: least squares (see `ParEst_LS.m`), swarm-optimization approach *covariance matrix adaptation - evolutionary strategies* CMA-ES by (Hansen and Ostermeier, 2001) (see `CMAES.m`) and unconstrained iterative method (see `ParEst_fminunc.m`).

4.4 Multi-objective non-dominated sorting

The evolution of dynamical structure as presented above can be guided by a multi-objective criterion. In the presented algorithm, we have considered only simulation and prediction error $(E_\mathrm{s}, E_\mathrm{s})$, but other criterion like derivation tree complexity (see (Khandelwal, 2020)) can also be included. In the multi-objective genetic programming literature, most of the evolutionary strageties bases their findings on Pareto optimality criterion. Further, We introduce the Pareto dominance definition (Emmerich and Deutz, 2018).

Definition 2. Pareto dominance
Given two vectors in the objective space, $O^{(1)}, O^{(2)} \in \mathbb{R}^m$, then the point $O^{(1)}$ said to *Pareto dominate* the point $O^{(2)}$ $(O^{(1)} \prec_\mathrm{Pareto} O^{(2)})$, if and only if $\forall i \in \{1,\ldots,m\}$: $O_i^{(1)} \leq O_i^{(2)}$ and $\exists j \in \{1,\ldots,m\}$: $O_j^{(1)} < O_j^{(2)}$. In case that $O^{(1)} \prec_\mathrm{Pareto} O^{(2)}$ the first vector is not worse in each of the objectives and better in at least one objective than the second vector.

Based on the Pareto dominance $\prec_\mathrm{Pareto}$, one can group a set of candidates into fronts. Each candidate has a dominance level and it is based on the number of how many other candidates are Pareto dominated by it. A Pareto front, F_i, can be seen as a contour on which all the candidates have the same dominance level. The order of dominance sorts the Pareto fronts. The Pareto optimal solution is the front that has the highest dominance level, as known as the set of non-dominated solution. A way to construct the Pareto fronts for a given set of dynamical structures is the NSGA-II algorithm detailed in (Deb et al., 2002) (see `NSGAII.m`). The NSGA-II algorithm is called in `TAG_GP_Step4`. For the structure sorting procedure, the new models constructed through crossover and mutation in every generation are benchmarked against a test data set $D_\mathrm{N}^\mathrm{test}$.

5. TOOLBOX MATLAB IMPLEMENTATION

The toolbox is publicly available at: `gitlab.com/tu-e1/tag3p-matlab-toolbox`. The repository contains an explanatory demo video that shows how to setup and run the algorithm by following the script `TAG3P_Call_Example.m`. After setting up the structures `Data`, `Parameters` and `ModelSpace`, the identification algorithm can be called by function `TAG3P.m`. The `Data` structure contains all the input output data sets arranged by role ($D_\mathrm{N}^\mathrm{est}$, $D_\mathrm{N}^\mathrm{test}$,$D_\mathrm{N}^\mathrm{val}$). Each input or output data set is defined as a matrix

Table 2. $\mathrm{RMS_s}$ and $\mathrm{RMS_p}$ results of TAG3P Matlab Toolbox over the benchmark models in comparison with other system identification strategies from the literature.

Bouc-Wen hysteresis model	$\mathrm{RMS_s}$	$\mathrm{RMS_p}$
TAG3P - G_{NARX}	6.52e−5	7.37e−6
Full PLNSS (Esfahani et al., 2017)	1.20e−5	-
Decoupled PLNSS (Esfahani et al., 2017)	1.40e−5	-
LMN - NARX (Belz et al., 2017)	-	9.86e−6
LMN - NFIR (Belz et al., 2017)	1.63e−4	-
Coupled electric drive	$\mathrm{RMS_s}$	$\mathrm{RMS_p}$
TAG3P - G_{extNARX}	1.28e−1	3.27e−3
TAG3P (Khandelwal, 2020)	1.2e−1	3.73e−3
GA + DE (Ayala et al., 2014)	1.8e−1	4.0e−2
Continuous Stirring Tank Reactor	$\mathrm{RMS_s}$	BFR
TAG3P - G_{expNARX}	1.6749	92.80%
LPV-OBF (Tóth et al., 2010)	-	97.54%

$\in \mathbb{R}^{\mathrm{r}_{u,y} \times \mathrm{N}}$. For this implementation all the data sets should contain the same amount of time-samples N. The function returns `LastGeneration` and `Data` structures. The former contains the Pareto solution alongside the last generation of models and the latter contains the input-output data structure with additional information. The function `./TAG_main_files/TAG3P_ResultsPrint.m` can be used to compute various error metrics and create simulation and prediction profiles for a chosen model over a validation data set.

6. RESULTS

We tested the TAG3P identification algorithm against two SISO and one MIMO benchmark models. For each model we considered three distinct data sets categories: $D_{\mathrm{N}}^{\mathrm{est}}$ for parameter estimation, $D_{\mathrm{N}}^{\mathrm{test}}$ for multi-objective sorting and $D_{\mathrm{N}}^{\mathrm{val}}$ for computing validation $\mathrm{RMS_s}$ and $\mathrm{RMS_p}$ metrics described in Equations (5). These metrics are used to compare the results obtained through the proposed method with the ones presented in literature. For all benchmark systems, the comparison is shown in Table 2. For each benchmark model, out of the Pareto solution, we have selected the candidate model that yields the lowest average simulation error over the $D_{\mathrm{N}}^{\mathrm{test}}$ data sets. For the MIMO benchmark model described in (Tóth et al., 2010), the authors measured their identification method performance in Best Fit Rate (BFR). Thus, for the MIMO case, alongside the $\mathrm{RMS_s}$ value we have also computed a BFR metric.

Table 3. TAG and genetic parameters used for the benchmark problem.

Benchmark model	TAG	Pop	Gen	Complexity
BoucWen oscillator	G_{NARX}	36	350	150
Coupled electric drive	G_{extNARX}	50	400	150
Stirred tank model	G_{expNARX}	60	350	120

6.1 SISO benchmark models

Bouc-Wen model The Bouc-Wen model represents hysteretic effects and it is widely used in mechanical engineering. This benchmark is based on synthetic data and represents a challenging system to identify. The system has a dynamic nonlinearity that is governed by a non measurable internal variable. To identify this model we used the TAG with parameters Table 3.

For the parameter estimation and testing data sets ($D_{\mathrm{N}}^{\mathrm{est}}$, $D_{\mathrm{N}}^{\mathrm{test}}$) we have generated 5 data sets of N = 4096 samples each using the algorithm indicated in (Noël and Schoukens, 2020). The validation data set $D_{\mathrm{N}}^{\mathrm{val}}$ was considered the sine sweep data set provided by the authors. For this data set, the identification results are reported in Table 2.

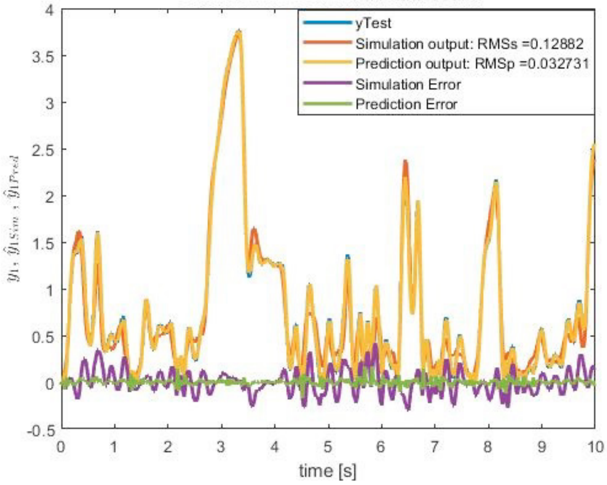

Fig. 2. Validation results on the coupled electric drive system terms of simulation and prediction responses of the estimated model on $D_{\mathrm{N}}^{\mathrm{val}}$.

Coupled electric drive model The coupled electric drives consists of two electric motors that drive a pulley using a flexible belt. The pulley is held by a spring, resulting in a lightly damped dynamic mode. The drive control for the pulley is designed only for tracking the speed reference signal. A pulse counter is used to measure the angular speed of the pulley. Thus, the sign of the velocity is unknown. The available data sets are short (N = 500), and together with the absolute value component of the velocity profile make this system interesting from an identification point of view. For this system we used the extended TAG and genetic programming parameters in Table 3.

As described in (Wigren and Schoukens, 2017), the estimation data set $D_{\mathrm{N}}^{\mathrm{est}}$ was constructed with u11 as input and z11 as output, while $D_{\mathrm{N}}^{\mathrm{test}}$ and $D_{\mathrm{N}}^{\mathrm{val}}$ data sets contained u12 as input and z12 as output.

The identification results, for the same validation data set, are fairly similar to the TAG3P implementation (in Mathematica) described in (Khandelwal, 2020). The result portrayed in Figure 2 and Table 2 shows that the new Matlab toolbox can match in error score the former Mathematica implementation.

6.2 MIMO benchmark model

Continuous Stirred Tank Reactor (CSTR) model The main contribution of this paper is the extension of the TAG modeling framework to MIMO complex models. For this we tested the TAG3P MIMO identification procedure on an ideal, simulated, CSTR that is fully described in (Tóth et al., 2010). In short, the CSTR resembles a chemical conversion of an inflow substance into a product. The chemical conversion is described by a highly nonlinear dynamic relation between input signals $U = [\mathrm{Q_1}, \mathrm{T_c}, \mathrm{C_1}]^\top$ (input flow, coolant temperature and concentration of the inflow) and output signals $Y = [\mathrm{T_2}, \mathrm{C_2}]^\top$ (temperature in the reactor and concentration in the reactor). Since the benchmark model is fully known, ten $D_{\mathrm{N}}^{\mathrm{est}}$, $D_{\mathrm{N}}^{\mathrm{test}}$ and one $D_{\mathrm{N}}^{\mathrm{val}}$ data sets were generated as described in (Tóth et al., 2010). Because of the known inverse and exponential terms within the model equations, to identify this model we used the extended TAG with parameters in Table

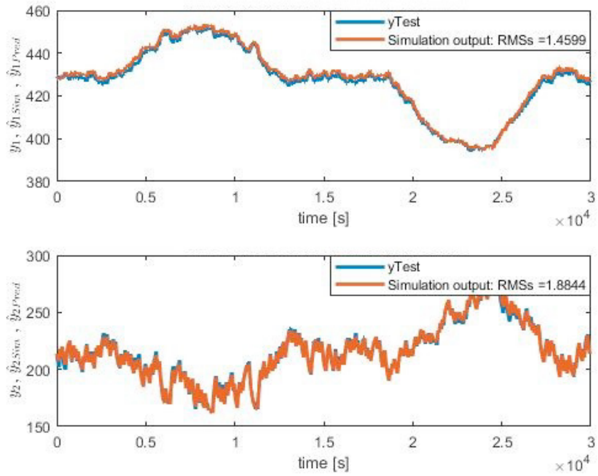

Fig. 3. Validation results on the CSTR in terms of simulation responses of the estimated model on $D_{\mathrm{N}}^{\mathrm{val}}$

3. For all identification use-cases we have observed that the estimation converges faster when the user selects a precise initial dynamics structure search space rather than most generic search space that includes all of the possible extensions. This is due to the fact that the searching space is increased significantly for each added extension thus, a bigger population size is required to cover the search space.

7. CONCLUSION AND FUTURE WORK

As presented in the Table 2 the new Matlab implementation of the TAG3P identification strategy could identify the three SISO benchmark models with a various degrees of fidelity. The results for Bouc-Wen oscillator and Coupled Electric Drive show $\mathrm{RMS_s}$ and $\mathrm{RMS_p}$ values on par with other literature solutions while for the Parallel Wiener-Hammerstein model, the obtained $\mathrm{RMS_s}$ is smaller than the Best Linear Approximation but considerable larger than the specialized Parallel W-H solution proposed in (Schoukens et al., 2015) by a factor of 3. In case of the MIMO CSTR system, the BFR metric shows that the proposed TAG modeling framework can obtain a valid model from data. In this paper we have shown that, the Matlab implementation of TAG3P can provide reliable candidates that represent complex SISO or MIMO nonlinear system without providing the significant amount of a priori dynamics structure specialized knowledge. Moreover, it shows a fine trade-off between performance of the identified model and the amount of critical decision the user has to take. Nevertheless the paper introduced and made available the first version of the Matlab Toolbox for TAG3P identification strategy.

In terms of the modeling framework and model space selection, the current TAG MIMO framework and Matlab implementation offer enough flexibility for proposing a genetic programming guided identification procedure for systems that can be described by polynomial nonlinear state space models. The aim of such framework is to enable the genetic evolution to automatically select the dynamic structure and the number of states.

REFERENCES

Ayala, H.V., da Cruz, L.F., Freire, R.Z., and dos Santos Coelho, L. (2014). Cascaded free search differential evolution applied to nonlinear system identification based on correlation functions and neural networks. In *IEEE Symp. on Comp. Intel. in Control and Automation*.

Belz, J., Münker, T., Heinz, T.O., Kampmann, G., and Nelles, O. (2017). Automatic modeling with local model networks for benchmark processes. *IFAC-PapersOnLine*, 50(1), 470–475. 20th IFAC World Congress.

Billings, S. (2013). *Nonlinear System Identification.*

Deb, K., Pratap, A., Agarwal, S., and Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE Trans. on Evol. Comp.*, 6.

Emmerich, M.T.M. and Deutz, A.H. (2018). A tutorial on multiobjective optimization: fundamentals and evolutionary methods. *Natural Computing*, 17.

Esfahani, A.F., Dreesen, P., Tiels, K., Noël, J.P., and Schoukens, J. (2017). Polynomial state-space model decoupling for the identification of hysteretic systems. *IFAC-PapersOnLine*, 50.

Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep Learning.* MIT Press.

Hansen, N. and Ostermeier, A. (2001). Completely derandomized self-adaptation in evolution strategies. *Evolutionary Computation*, 9.

Hoai, N.X., McKay, R.I., and Abbass, H.A. (2003). Tree adjoining grammars, language bias, and genetic programming. In *Proceedings of the 6th European Conference on Genetic Programming*, EuroGP'03, 335–344. Springer-Verlag, Berlin.

Kallmeyer, L. (2009). A declarative characterization of different types of multicomponent tree adjoining grammars. *Research on Language and Computation*, 7.

Khandelwal, D. (2020). *Automating Data-driven Modelling of Dynamical Systems - An Evolutionary Computation Approach.* Phd thesis, Eindhoven University of Technology.

Khandelwal, D., Schoukens, M., and Toth, R. (2019). Data-driven modelling of dynamical systems using tree adjoining grammar and genetic programming. In *2019 IEEE Congress on Evolutionary Computation (CEC)*, 2673–2680. IEEE.

Ming-guang Zhang, Wei-wu Yan, and Zhan-ting Yuan (2004). Study of nonlinear system identification based on support vector machine. In *International Conf. on Machine Learning and Cybernetics*. IEEE, Shanghai, China.

Noël, J.P. and Schoukens, M. (2020). Hysteretic Benchmark with a Dynamic Nonlinearity.

Patelli, A. and Ferariu, L. (2009). Nonlinear system identification by means of genetic programming. In *European Control Conf.*

Pillonetto, G., Dinuzzo, F., Chen, T., De Nicolao, G., and Ljung, L. (2014). Kernel methods in system identification, machine learning and function estimation: A survey. *Automatica*, 50.

Schoukens, M., Marconato, A., Pintelon, R., Vandersteen, G., and Rolain, Y. (2015). Parametric identification of parallel wiener–hammerstein systems. *Automatica*, 51.

Tóth, R., Van den Hof, P., Ludlage, J., and Heuberger, P. (2010). Identification of nonlinear process models in an lpv framework. In *IFAC Proceedings Volumes*, volume 43.

Wigren, T. and Schoukens, M. (2017). Coupled electric drives data set and reference models. Technical report.