Nonlinear Control Structure Design using Grammatical Evolution and Lyapunov Equation based Optimization

Elias Reichensdörfer[1,2], Dirk Odenthal[3] and Dirk Wollherr[4]
[1]BMW Group, Knorrstr. 147, 80788, Munich, Germany
[2]Department of Electrical and Computer Engineering, Chair of Automatic Control Engineering, Technical University of Munich, Theresienstr. 90, 80333, Munich, Germany
[3]BMW M GmbH, Daimlerstr. 19, 85748 Garching near Munich, Germany
[4]Department of Electrical and Computer Engineering, Chair of Automatic Control Engineering, Technical University of Munich, Theresienstr. 90, 80333, Munich, Germany

Keywords: Nonlinear Control Structure Design, Lyapunov Equation, Grammatical Evolution.

Abstract: A new method for the automated synthesis of nonlinear control laws for nonlinear control systems using grammatical evolution is presented. The controller structure, its parameterization and a quadratic Lyapunov function are the result of a nonlinear, nonconvex optimization process. Evolutionary algorithms based on grammatical evolution are used to find candidates for the control law. These are evaluated using a fitness function incorporating eigenvalue specifications on the linearized closed loop system and bounds on the control input signals. The guaranteed domain of attraction subject to the closed loop performance and stability specifications is maximized by evaluating the solution of the Lyapunov equation on the nonlinear system. The method is tested on two different control systems that contain different types of nonlinearities. The results show that the proposed approach is capable of outperforming state of the art methods by providing stronger stability guarantees and/or better closed loop performance while making less restrictive assumptions.

1 INTRODUCTION

The Lyapunov method is one of the most powerful tools available to analyze stability properties of nonlinear dynamical systems (Lyapunov, 1992). The method provides a sufficient condition for stability of a nonlinear system: the existence of a Lyapunov function. Nevertheless, how to systematically find suitable Lyapunov function candidates is still unsolved. Although significant progress has already been made—Sum-of-Square programming (Prajna et al., 2002) partially solves the problem for polynomial systems—however it is still open in the general case. An extensive overview of Lyapunov function synthesis is presented by (Giesl and Hafstein, 2015). Even more challenging is to simultaneously find a controller structure, its parameterization and a suitable Control Lyapunov function (CLF) to prove stability. In (Sontag, 1983), an explicit formula, also known as "Sontag's formula", is presented to construct a stabilizing controller for nonlinear input affine systems assuming a predetermined CLF. While Sontag's formula provides a direct way to compute a controller for a nonlinear system, it still relies on the choice of an appropriate CLF which indirectly determines the structure of the underlying controller.

For design and analysis of linear control systems there exists a variety of methods like loop shaping or pole placement (Zhou et al., 1996). Also, for a stable linear system, a Lyapunov function can be computed by solving the Lyapunov equation. This equation can be solved efficiently by numerical methods (Bartels and Stewart, 1972). To take advantage of the variety of performance criteria from linear control theory, a common approach is to linearize the nonlinear system about an operating point. However, the domain of attraction (DoA) of such a controller design might be unacceptably small. Also, for some systems, nonlinear feedback might lead to better performance by exploiting the nonlinear behavior of the system. Most classical design approaches from control theory for both linear and nonlinear controllers rely on several structural assumptions.

Meta-heuristic search methods (MHSM), often inspired by nature, make very little assumptions on the underlying solution structure (Glover and Kochenberger, 2006). One major branch of MHSM are evolutionary algorithms (EA); a branch of EA are genetic

Reichensdörfer, E., Odenthal, D. and Wollherr, D.
Nonlinear Control Structure Design using Grammatical Evolution and Lyapunov Equation based Optimization.
DOI: 10.5220/0006839400550065
In *Proceedings of the 15th International Conference on Informatics in Control, Automation and Robotics (ICINCO 2018) - Volume 1*, pages 55-65
ISBN: 978-989-758-321-6

algorithms (GA) by (Holland, 1975). GA have been extended to synthesize arbitrary programs by (Koza, 1992) with the method of genetic programming (GP). A variation of GP is grammatical evolution (GE) by (Ryan et al., 1998). Both GP and GE can not only optimize parameters for a given structure, but are able to optimize the mathematical structure itself. Due to this generality, it is appealing to tackle the problem of nonlinear control synthesis with these methods.

Meta-heuristic search methods have proven useful in various practical control applications. In (Precup et al., 2015), input membership functions of Takagi-Sugeno-Kang fuzzy models are tuned for an anti-lock braking system using simulated annealing and particle swarm optimization (PSO). The work by (Saadat et al., 2017) applies different MHSM for training echo state neural networks for time series prediction. In (Vrkalovic et al., 2017), different MHSM are combined with linear matrix inequalities for Takagi-Sugeno fuzzy controller design, applied to a cart pole. Identification of an optimal time delay feedback for active vibration control using a GA is shown in (Mirafzal et al., 2016), (Hosen et al., 2015) use a GA and PSO to improve prediction capabilities of neural networks for time series prediction. Other applications include energy management of hybrid electric vehicles (Chen et al., 2014a), tuning of proportional-integral-derivative (PID) controllers for inductive power transfer systems (Chen et al., 2014b) and heating systems (Singh et al., 2015). More applications can be found in (Haupt and Haupt, 1998) and (Poli et al., 2008).

Evolutionary algorithms have already been used for both structural controller design and Lyapunov function synthesis. In (Shimooka and Fujimoto, 2000), GP is used to generate a controller for the nonlinear inverted pendulum, while (Cupertino et al., 2002) use structural chromosomes in their EA to generate controllers for a nonlinear DC drive with variable load. In (Koza et al., 1999), the structure of a controller for a linear three lag plant is synthesized with GP, while in (Koza et al., 2000) the same control system, extended by a time delay, is considered. Robust controllers have been evolved by GP for linear interval plants (Chen and Lu, 2011) and for nonlinear systems (Reichensdörfer et al., 2017) using GE. The authors of (Gholaminezhad et al., 2014) use pareto optimization to find the structure of a linear controller with GP. All of these approaches focus on the synthesis of control laws only. Stability is either investigated using linear methods in a subsequent design step or by heuristic arguments obtained with simulation. However, there also exists work on using EA for structural optimization of Lyapunov functions.

In (Banks, 2002), GP is used to evolve Lyapunov functions for nonlinear systems, while (Grosman and Lewin, 2005) included a penalty term in their GP algorithm to avoid overly complex solutions. Further, (McGough et al., 2010) use GE for Lyapunov function synthesis. These methods focus on finding a Lyapunov function for a system to prove stability, but do not incorporate the search for control laws. In (Tsuzuki et al., 2006), GP was used to find the structure of Control Lyapunov-Morse functions. However, the controllers were constructed using Sontag's formula, not by using the EA directly. Finally, (Verdier and Mazo Jr, 2017) used GP to construct a switching rule based on CLF for a switched state feedback controller. However, the modes of the controller were limited to linear state feedback and actuator limits were not taken into account. Also, simulations of the switched state feedback controller showed high frequency chattering, which is often undesirable in practice. In general, few research exists that has combined CLF and EA so far.

In this paper, a new method is presented using GE to synthesize the structure of control laws for a nonlinear system and generate the CLF by solving the Lyapunov equation for the linearized system. The generated Lyapunov function is then evaluated on the *nonlinear system* as a performance measure. In the current state of the art, there is no method available to generate controller structures for nonlinear systems using CLF and EA that includes constraints on both the linearized dynamics and control effort. The main contribution is the introduction of a novel fitness function that simultaneously aims to maximize the guaranteed domain of attraction while taking into account constraints on the dynamics of the linearized closed loop system, as well as limitations of actuators. The motivation behind this is to identify controller structures that result in a predefined performance of the system when it is close to its equilibrium. At the same time, the domain of attraction is maximized in order to make the controller robust with respect to inhomogeneous initial conditions and perturbations. The GE algorithm chooses the structure of control laws such that the requirements on the linearized closed loop system are fulfilled and the domain of attraction of the systems associated quadratic Lyapunov function is maximized for the nonlinear system.

The remainder of this paper is organized as follows. In section 2, the concepts of Lyapunov stability and GE are summarized. Section 3 introduces a novel method for control structure design using GE. In section 4, we evaluate the proposed method on two nonlinear control systems. Finally, in section 5, a conclusion is drawn and an outlook for future work is given.

2 BACKGROUND

2.1 Lyapunov Stability and CLF

We consider autonomous dynamical systems of the form

$$\dot{x} = f(x) \tag{1}$$

with $f : \mathbb{R}^n \to \mathbb{R}^n$, $f \in C^\infty$ and $x \in X \subseteq \mathbb{R}^n$, where x denotes the state vector of the system. We are interested in the stability of equilibrium points of the system.

Definition 1. *A state x^* is called an equilibrium if $f(x^*) = 0 \; \forall t$.*

Remark 1. *We assume, without loss of generality, that $x^* = 0$ since any non zero equilibrium can be transformed into the origin by change of coordinates.*

The stability of a given equilibrium is defined as follows.

Definition 2. *An equilibrium x^* of the system (1) is called stable in the sense of Lyapunov if for every $\varepsilon > 0$ there exists some $\delta(\varepsilon, t_0)$ such that*

$$||x(t_0)|| < \delta \Rightarrow ||x(t)|| < \varepsilon, \forall t \geq t_0 . \tag{2}$$

Otherwise, the equilibrium is called unstable. If additionally for $||x(t_0)|| < \delta \Rightarrow \lim_{t\to\infty} ||x(t)|| = 0$, the equilibrium is called asymptotically stable.

First, the notion of positive and negative (semi)-definiteness of a function is required.

Definition 3. *Let $X_0 \subseteq \mathbb{R}^n$ be a connected set containing the origin. A function $V : \mathbb{R}^n \to \mathbb{R}$ is called positive definite on X_0, if*

$$V(0) = 0 \tag{3}$$

$$V(x) > 0, x \in X_0 \setminus \{0\} , \tag{4}$$

denoted by $V(x) \succ 0$ and positive semi-definite if inequality (4) is non-strict, denoted by $V(x) \succeq 0$. A function $V(x)$ is called negative (semi)-definite if $-V(x)$ is positive (semi)-definite, denoted by $V(x) \prec 0$ and $V(x) \preceq 0$ respectively.

Stability of (1) can be determined using the direct method of Lyapunov.

Definition 4. *A function $V : \mathbb{R}^n \to \mathbb{R}$ is a Lyapunov function for (1) if for $x \in X_0$*

$$V(x) \succ 0 \tag{5}$$

$$\dot{V}(x) = \frac{\partial V}{\partial x} f(x) \preceq 0 \tag{6}$$

and strict Lyapunov function if $\dot{V}(x) \prec 0$.

Existence of a (strict) Lyapunov function is sufficient to proof (asymptotic) stability of an equilibrium. If additionally $X_0 = \mathbb{R}^n$ and $\lim_{||x||\to\infty} V(x) \to \infty$, the stability property holds true globally.

The domain of attraction is given by the next definition.

Definition 5. *Domain of attraction*

$$\mathcal{D}(x^*) = \{x \in \mathbb{R}^n : \lim_{t\to\infty} \varphi(t, 0, x) = x^*\} , \tag{7}$$

where $\varphi(t, t_0, x_0)$ denotes the solution of (1) starting at t_0 under the initial condition x_0.

Now a dynamical system with static state feedback $u(x)$ of the form

$$\dot{x} = f(x, u(x)) \tag{8}$$

is considered. Here $u(x) \in \mathbb{R}^m$ is the control law to be defined. The idea of CLF is now to choose $u(x)$, such that the Lyapunov conditions from definition 4 hold true. Since this step is difficult in general, we want to automatize it using GE. Before we proceed with stating the optimization problem, an introduction to the GE algorithm is given.

2.2 Grammatical Evolution

Common evolutionary algorithms simulate natural selection mechanisms to evolve solutions. In this work, the focus is set on genetic algorithms (Holland, 1975). There, a fitness function determines the statistical survival probability of an individual within a population. Individuals are then selected based on their fitness and recombined by the crossover operator in order to produce offspring. The mutation operator finally adds additional diversity to the population by randomly modifying solutions. The flowchart of a typical GA is illustrated in figure 1. GA are well known for parameter tuning but have also been extended to perform structure optimization (Koza, 1992), an approach called genetic programming.

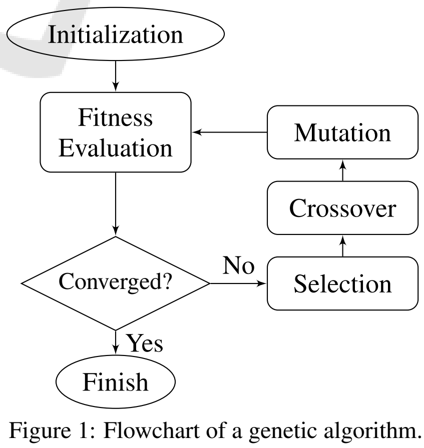

Figure 1: Flowchart of a genetic algorithm.

Grammatical evolution (Ryan et al., 1998), a variation of GP, is based on the idea to define a grammar in Backus-Naur form (BNF) (Backus, 1959; Naur et al., 1969) to parse an integer string into an executable program. This works by interpreting a string of integer numbers as a programming language. The GE algorithm compiles this language using a grammar in BNF. The generated program, which in many applications is just a mathematical formula, can then be evaluated. The result of this evaluation is rated by a user-defined metric, which serves as a fitness value for the corresponding individual. A grammar in BNF can be described by the following definition.

Definition 6. *A context free grammar is the 4-tuple* $\mathcal{G} = (\mathcal{N}, \Sigma, \mathcal{P}, \mathcal{S})$ *where* $\mathcal{N}$ *is the finite set of nonterminals,* Σ *the finite set of terminals with* $\mathcal{N} \cap \Sigma = \emptyset$, $\mathcal{S} \in \mathcal{N}$ *the start symbol. The vocabulary of* $\mathcal{G}$ *is* $\mathcal{V} = \mathcal{N} \cup \Sigma$, *which gives the finite set of production rules as* $\mathcal{P} \subset (\mathcal{V}^* \setminus \Sigma^*) \times \mathcal{V}$, *each of the form* $\mathcal{N} \to \mathcal{V}^*$.

Here, $*$ denotes the Kleene star operator. Example 1 shows, how this concept can be used to translate an integer vector into a mathematical expression.

Example 1. *We choose an example grammar* $\mathcal{G}$ *like in definition 6 as* $\mathcal{N} = \{\langle e\rangle, \langle op\rangle, \langle v\rangle, \langle c\rangle\}$, $\Sigma = \{+, -, x, 1\}$, $\mathcal{S} = \langle e\rangle$ *and* $\mathcal{P} = \{\langle e\rangle \overset{r_1}{\to} \langle op\rangle(\langle e\rangle, \langle e\rangle), \langle e\rangle \overset{r_2}{\to} \langle v\rangle, \langle e\rangle \overset{r_3}{\to} \langle c\rangle, \langle op\rangle \overset{r_4}{\to} +, \langle op\rangle \overset{r_5}{\to} -, \langle v\rangle \overset{r_6}{\to} x, \langle c\rangle \overset{r_7}{\to} 1\}$.

Consider an example individual, arbitrarily chosen as $i_{GE} = \begin{bmatrix} 6 & 4 & 1 & 2 & 5 & 2 \end{bmatrix}^T$. *Dispatching the current rule* r_k *is done using the modulo operator as* $i_{GE,k}$ mod p *where* $i_{GE,k}$ *is called the k-th codon of* i_{GE} *and p is the number of available rules from the current non-terminal. This example starts with* $\mathcal{S} = \langle e\rangle$. *Since* $\langle e\rangle$ *can be expanded by the three rules* r_1, r_2, r_3, *in the first step,* $p = 3$. *Modulo dispatch selects the first rule since* $i_{GE,1}$ mod $p = 6$ mod $3 = 0$. *The translation process is then continued by unfolding all nonterminals from left to right as shown in table 1.*

Table 1: Parsing process on i_{GE} using $\mathcal{G}$.

$i_{\mathrm{GE},k}$	p	$i_{\mathrm{GE},k}$ mod p	Rule	Expression
-	-	-	$\mathcal{S}$	$\langle e\rangle$
6	3	0	r_1	$\langle op\rangle(\langle e\rangle, \langle e\rangle)$
4	2	0	r_4	$+(\langle e\rangle, \langle e\rangle)$
1	3	1	r_2	$+(\langle v\rangle, \langle e\rangle)$
2	1	0	r_6	$+(\mathrm{x}, \langle e\rangle)$
5	3	2	r_3	$+(\mathrm{x}, \langle c\rangle)$
2	1	0	r_7	$+(\mathrm{x}, 1)$

This procedure shows that the example individual represents the expression $x+1$. If the mapping is not finished yet but ran out of codons, the wrapping technique is used, starting from the beginning of i_{GE}. To avoid infinite loops, a maximum wrapping count is defined. After exceeding this count, all recursive rules are eliminated from $\mathcal{P}$ (in example 1 that would be r_1), such that the parsing is guaranteed to terminate in a finite number of steps.

3 METHODS

3.1 Linear Performance Measures

For applicability and interpretability it is desirable to define performance criteria from linear control theory for the synthesized controllers. Therefore, we use Jacobian linearization of the closed loop system about the equilibrium x^* with $\delta x = x - x^*$. An approximation of the system dynamics by linearization is then given by

$$\delta\dot{x} = \left[\frac{\partial f}{\partial x}\right]\Big|_{x=x^*} \delta x = A_c \delta x\,. \tag{9}$$

The first step is therefore to insert the GE-generated control law into (8) and perform the above linearization. Afterwards, methods from linear control theory can be applied as performance measures. In this work we use a common approach for pole placement presented in (Ackermann et al., 2002), where a suitable region Γ in the complex s-plane is defined,

$$\Gamma_T = \left\{s = \sigma + j\omega : \sigma \le -\frac{1}{T_{\max}}\right\}, \tag{10}$$

$$\Gamma_\omega = \left\{s = \sigma + j\omega : \sqrt{\sigma^2 + \omega^2} \le \omega_{\max}\right\}, \tag{11}$$

$$\Gamma_\zeta = \left\{s = \sigma + j\omega : |\omega| \le \frac{\sqrt{1-\zeta_{\min}^2}}{\zeta_{\min}} |\sigma|\right\}, \tag{12}$$

$$\Gamma = \Gamma_T \cap \Gamma_\omega \cap \Gamma_\zeta\,. \tag{13}$$

Here, $\zeta_{\min}$ specifies the minimum damping of the system, $T_{\max}$ the maximum time constant and $\omega_{\max}$ the maximum frequency of the linearized system from (9). The area spanned by this definition is a circular sector with “cropped top”, an example is shown in figure 2.

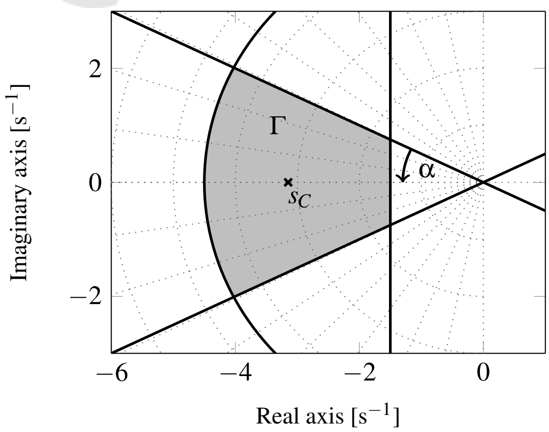

Figure 2: Example Γ-region (grey) for pole placement.

To assign a fitness value to a given controller, the centroid of this region can be used. We propose the following analytic formula for the centroid of the Γ-region from figure 2.

Proposition 1. *The centroid $s_C = \sigma_C + j\omega_C$ of the stability area Γ spanned in the complex s-plane by equation (13) as a function of the design parameters ζ_{min}, T_{max} and ω_{max} with $\zeta_{min} \in (0,1]$, $T_{max} > 0$ and $\omega_{max} > 1/T_{max}$ is given by*

$$\omega_C = 0\,,\, \sigma_C = \begin{cases} \dfrac{M_1+M_2}{A_1+A_2}\,, & \Delta > 0 \quad \text{(14a)} \\ M_1/A_1\,, & \Delta \leq 0 \quad \text{(14b)} \end{cases}$$

where

$$\Delta = \omega_{max}\zeta_{min} - \frac{1}{T_{max}}\,, \tag{15}$$

$$A_1 = \frac{1}{4}\omega_{max}^2\big(2\alpha - \sin(2\alpha)\big)\,, \tag{16}$$

$$M_1 = -\frac{1}{3}\omega_{max}^3 \sin^3(\alpha)\,, \tag{17}$$

$$A_2 = \frac{\sqrt{1-\zeta_{min}^2}(\omega_{max}^2\cos^2(\alpha)T_{max}^2 - 1)}{2\zeta_{min}T_{max}^2}\,, \tag{18}$$

$$M_2 = \frac{-\sqrt{1-\zeta_{min}^2}(\omega_{max}^3\cos^3(\alpha)T_{max}^3 - 1)}{3\zeta_{min}T_{max}^3} \tag{19}$$

and

$$\alpha = \begin{cases} \arctan\left(\dfrac{\sqrt{1-\zeta_{min}^2}}{\zeta_{min}}\right), & \Delta > 0 \quad \text{(20a)} \\ \arccos\left(\dfrac{1}{\omega_{max}T_{max}}\right), & \Delta \leq 0\,. \quad \text{(20b)} \end{cases}$$

Proof. Because of the symmetry of the Γ-region, ω_C has to be 0. It is sufficient to examine the curve above the real axis. For the real part of the Γ-centroid, two cases have to be considered.

Case 1. *If $\Delta \leq 0$, the Γ-region reduces to a circular segment. Then σ_C is given by M_1/A_1, the known standard formula for the centroid of a circular segment.*

Case 2. *If $\Delta > 0$, σ_C can be derived by calculating the centroids of the circular segment and the remaining quadrilateral. The centroid and the upper half of the area of the circular segment are known as*

$$\sigma_{C,1} = -\frac{4\omega_{max}\sin^3(\alpha)}{3(2\alpha - \sin(2\alpha))}\,, \tag{21}$$

$$A_1 = \frac{1}{4}\omega_{max}^2(2\alpha - \sin(2\alpha))\,. \tag{22}$$

The area and momentum of the upper half of the quadrilateral can be computed using integration

$$A_2 = -\frac{\sqrt{1-\zeta_{min}^2}}{\zeta_{min}}\int_{-\omega_{max}\cos(\alpha)}^{-1/T_{max}} \sigma d\sigma\,, \tag{23}$$

$$M_2 = -\frac{\sqrt{1-\zeta_{min}^2}}{\zeta_{min}}\int_{-\omega_{max}\cos(\alpha)}^{-1/T_{max}} \sigma^2 d\sigma\,, \tag{24}$$

which then, with the relation $\sigma_{C,i} = M_i/A_i$, yields the above result for the composite centroid of the Γ-region. □

Now we can define a fitness measure for the GE algorithm as follows.

Definition 7. *Given the eigenvalues λ_i from the system matrix A_c of a linearized system (9) and a Γ-region defined by the triple $(T_{max}, \zeta_{min}, \omega_{max})$ the first part of the total fitness value is given by*

$$\phi_{linear} = 1 + \frac{1}{n}\sum_{i=1}^{n} \begin{cases} ||\lambda_i - s_C||_2^2\,, & \lambda_i \notin \Gamma \quad \text{(25a)} \\ 0\,, & \lambda_i \in \Gamma\,. \quad \text{(25b)} \end{cases}$$

Remark 2. *The value of ϕ_{linear} describes the mean deviation of the eigenvalues of A_c not contained in Γ from the geometrical mean (the centroid) of Γ. Note that this only makes sense because Γ is convex and connected. Otherwise the centroid is not guaranteed to lie within the specified region.*

Remark 3. *In principle, different Γ-regions to the proposed could be used. The advantage of this region however, is that it is widely used, compare (Ackermann et al., 2002) or (Chilali and Gahinet, 1996), can be parameterized by three meaningful values and is both convex and connected, see remark 2.*

Next, we want the controller not only to place the poles of the linearized closed loop system (9) in the Γ-region (13), but also to obtain a guaranteed domain of attraction, which is as large as possible. We continue by stating this requirement as an optimization problem and then propose a formulation compatible to a GE algorithm, in order to make it computationally tractable.

3.2 Domain of Attraction Estimation

To obtain an estimate for the stability region of the synthesized controller, we use the fact that a subset of the DoA (7), $\Omega_c \subseteq \mathcal{D}(0)$ can be computed using

$$\Omega_c = \{x \in \mathbb{R}^n : V(x) \leq c\}\,, \tag{26}$$

with $c \in \mathbb{R}^+$, as long as $V(x)$ fulfills the Lyapunov conditions from definition 4 on Ω_c. The task of finding a controller that maximizes the domain of attraction and satisfies the constraints defined in section 3.1

can then be stated as the constrained, bilevel optimization problem

$$\underset{u\in U(\mathcal{G})}{\text{maximize}} \quad L(\Omega_c^\star) \tag{27a}$$

$$\text{subject to} \quad \Omega_c^\star = \{x\in\mathbb{R}^n : V(x)\le c^\star\}\,, \tag{27b}$$

$$c^\star = \underset{x\in\mathbb{R}^n_{\neq 0}}{\text{minimize}} \quad V(x) \tag{27c}$$

$$\text{subject to} \quad \dot V(x) = 0\,, \tag{27d}$$

$$\lambda_i(A_c)\in\Gamma \;\; \forall i\in\{1,\dots,n\}\,, \tag{27e}$$

$$A_c^T P + P A_c + Q = 0\,, \tag{27f}$$

$$P = P^T \succ 0\,, \tag{27g}$$

$$V(x) = x^T P x\,, \tag{27h}$$

$$|u_j(x)| \le u_{j,\max} \forall j\in\{1,\dots,m\}\,. \tag{27i}$$

Here, $L(\Omega_c^\star)$ denotes the Lebesgue-measure of $\Omega_c^\star$, which can be interpreted as the maximum volume of the estimated domain of attraction (26) for a given $u(x)$. Furthermore, $U(\mathcal{G})$ is the function space of control laws $u(x)$, that can be generated by a user-defined grammar $\mathcal{G}$, specified in definition 6. Following, $u_{\max}$ is the actuator limit, $\lambda_i(A_c)$ the i-th eigenvalue of A_c (9) and Γ the region for pole placement (13). Equations (27c) and (27d) describe the standard optimization problem of finding the largest level curve of $V(x)$ where $\dot V(x) \preceq 0$, for example discussed in (Saleme et al., 2011). In the proposed setup, this is only one constraint of the outer optimization problem (27a), (27b) of finding a suitable control law $u(x)$. Constraint (27e) ensures that the poles of the linearized system (9) are placed in the Γ-region (13), while (27i) limits control effort.

In order to produce correct results, it is necessary that $V(x)\succ 0$, which can be difficult to test numerically for arbitrary $V(x)$. Therefore we use the solution P of the Lyapunov equation, given $Q\succ 0$, of the linearized closed loop system (9) to construct a symbolic representation of $V(x)$ (27f), (27g), (27h). In this work we choose Q as the identity matrix. Now we perform a Monte Carlo sampling on $V(x)$ and $\dot V(x)$ within a region of interest of the state space $X_c\subset\mathbb{R}^n$. The time derivative $\dot V(x)$ is calculated for the nonlinear system symbolically by automatic differentiation. The DoA is estimated using algorithm 1, which is a modified version of the algorithm presented in (Najafi et al., 2016): We store the samples $V(x_i)$ in a list L (line 5), which is used to estimate the DoA (lines 8-11). This algorithm is used to solve the inner optimization problem (27c), (27d).

The proportion k/N is then a normalized approximation of the size of the domain of attraction on the sample set by Monte Carlo integration and gives the final fitness measure to be minimized by the GE algorithm, which includes (27a)-(27h),

Algorithm 1: Domain of attraction estimation.

1: **procedure** DoASize($V(x), \dot V(x), X_c, N$)
2: Initialization: $c\leftarrow\infty, k\leftarrow 0, L\leftarrow\emptyset$
3: **for** $i\leftarrow 1$ **to** N **do**
4: Pick random state $x_i\in X_c$
5: $L\leftarrow(L, V(x_i))$ ▷ Store $V(x_i)$ in a list L
6: **if** $V(x_i) < c$ **and** $\dot V(x_i)\ge 0$ **then**
7: $c\leftarrow V(x_i)$
8: **for** $i\leftarrow 1$ **to** N **do** ▷ Estimate the DoA size
9: **if** $L_i < c$ **then** ▷ L_i is the i-th element of L
10: $k\leftarrow k+1$
11: **return** k/N

$$\phi = \begin{cases} \phi_{\text{linear}}\,, & \phi_{\text{linear}} > 1 \quad (28a)\\ 1-k/N\,, & \phi_{\text{linear}} = 1\;. \quad (28b)\end{cases}$$

Additionally, we apply a smooth, symmetric saturation with $|u_j(x)|\le u_{j,\max}(x)$ as shown in (Graichen et al., 2010) to account for (27i). The function

$$u_{j,\text{sat}} = u_{j,\max} - \frac{2u_{j,\max}}{1+\exp(2u_j/u_{j,\max})} \tag{29}$$

approximates a “hard” saturation, keeping u_{sat} and therefore the CLF differentiable.

Remark 4. *While most methods cited in section 1 for evaluating the quality of a Lyapunov function within an EA framework for Lyapunov function synthesis use some kind of sampling, most only count the proportion of the samples that satisfy conditions in defnition 4. However, this measure is not very robust like the example $V(x_1,x_2) = x_1^2$ demonstrates. Since with random samples, it is practically impossible to hit a value on the zero line $x_2 = 0$, the algorithm will not detect positive semi-definiteness of $V(x_1,x_2)$. This however is crucial for stability analysis since $V(x_1,x_2)\succeq 0$ implies that no conclusion about stability can be drawn.*

Remark 5. *Explicitly sampling on the axis $x_2 = 0$ using grid based techniques does not address the problem either as the example $V(x_1,x_2) = (\sin(\beta)x_1 + \cos(\beta)x_2)^2$ with some $\beta\in(-\pi/2,\pi/2)\setminus\{0\}$ shows. We therefore expect a more robust measure using the proposed method, since $V(x)$, generated as a solution to (27f), is automatically positive definite. If (27f) is not solvable for some generated controller, it is assigned the worst possible fitness, $\phi=\infty$.*

In the following section, we use a GE algorithm to solve the optimization problem (27a)-(27i) for two nonlinear benchmark systems in order to show the efficiency of the proposed method.

4 EXPERIMENTS

4.1 Metrics and Setup

Table 2 shows the parameters of the GE algorithm.

Table 2: Grammatical evolution setup.

Parameter	Value
Population size	100
Number of generations	5000
Mutation rate	0.05
Initial genome length	$\sim \mathcal{U}\{10,20\}$
Initial genome values	$\sim \mathcal{U}\{0,20\}$
Max. wrapping count	5
Max. tree depth	15
Selection	Rank selection
Crossover	Cut and splice
Mutation	Uniform multi-point
Elitism rate	0.1
Selection pressure	1.5
Monte Carlo sample size	5000

The grammar $\mathcal{G} = (\mathcal{N}, \Sigma, \mathcal{P}, \mathcal{S})$ used for the optimization was chosen as

$$\mathcal{S} = \langle e \rangle\,, \tag{30}$$
$$\mathcal{N} = \{\langle e \rangle, \langle op \rangle, \langle f \rangle, \langle v \rangle, \langle c \rangle\}\,, \tag{31}$$
$$\Sigma = \{+, -, \times, /, \sin, \cos, \arctan, x_1, \ldots, x_n, C\}\,, \tag{32}$$

with the following mapping, denoted in BNF syntax, by the production rules $\mathcal{P}$,

$$\langle e \rangle ::= \langle op \rangle(\langle e \rangle, \langle e \rangle) \mid \langle f \rangle(\langle e \rangle) \mid \langle v \rangle \mid \langle c \rangle \tag{33a}$$
$$\langle op \rangle ::= + \mid - \mid \times \mid / \tag{33b}$$
$$\langle f \rangle ::= \sin \mid \cos \mid \arctan \tag{33c}$$
$$\langle v \rangle ::= x_1 \mid \ldots \mid x_n \tag{33d}$$
$$\langle c \rangle ::= C\,. \tag{33e}$$

Here, C represents 10000 randomly generated constants, drawn from a continuous uniform distribution $\sim \mathcal{U}(-10, 10)$. This provides a variety of different constants for the GE algorithm, such that parameters of the control law can be fine-tuned.

4.2 Nonlinear Oscillator Example

We validate the proposed method on two different test systems. The first example is adopted from (Doyle et al., 1996), describing a two dimensional oscillator with highly nonlinear dynamics,

$$\dot{x}_1 = x_2 \tag{34}$$
$$\dot{x}_2 = \exp(x_2)\big(\cos(x_1)\sin(x_1) - 2x_1 + u\big)\,, \tag{35}$$

with $x_1 \in [-4, 4]$, $x_2 \in [-4, 4]$. Additionally we introduce the constraint that $u \in [-5, 5]$. The parameters for the Γ-region were defined as $T_{\max} = 1$ s, $\zeta_{\min} = 0.7$ and $\omega_{\max} = 5\ \mathrm{s}^{-1}$. After several iterations, the GE algorithm produces the following control law of non obvious structure

$$u_{\mathrm{GE}} = -3x_2^3 + 2x_1x_2^2 + (x_1 - 2)x_2\,, \tag{36}$$

with the associated Lyapunov function

$$V(x) = 1.5x_1^2 + 0.5x_2^2 + x_1x_2\,. \tag{37}$$

The eigenvalues of the closed loop system are both located at $(-1, 0)$, satisfying the specified requirements on the linearization of the system. We now use Jacobian linearization (JL) of the uncontrolled, saturated system and pole placement to generate a linear controller with the same eigenvalues, given by

$$u_{\mathrm{JL}} = -2x_2\,. \tag{38}$$

Notice that both controllers (36) and (38) have the same linearized closed loop eigenvalues and therefore the same quadratic Lyapunov function (37) according to equation (27f). While their performance is therefore locally equivalent, it differs when getting farther from the equilibrium. Figure 3 shows the estimated domain of attraction for the GE controller (solid, blue) and the linear controller (dashed, blue). The orange lines indicate the level sets, where $\dot{V}(x) = 0$ for the GE controller (solid, orange) and the linear controller (dashed, orange). The GE controller shows a guaranteed DoA larger by a factor of 4.57 compared to the linear controller. This estimate was computed using algorithm 1 with a sample size of $N = 10^6$. Notice, that for the optimization itself, a significantly smaller sample size, namely $N = 5000$, as given in table 2, was sufficient to achieve this result.

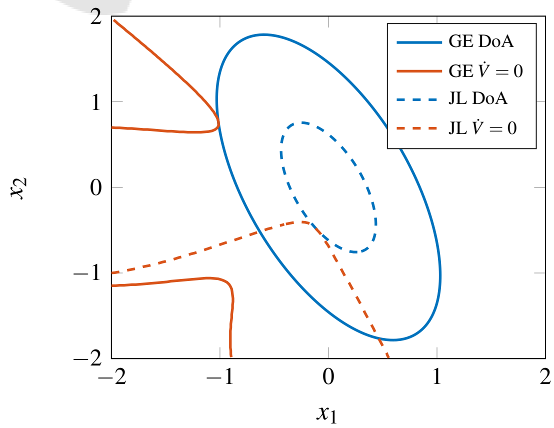

Figure 3: Comparison of domains of attraction.

One single optimization run on this example took approximately 30 seconds on a 2.8 GHz CPU, with the algorithm implemented in C++, using the options from table 2. Out of 100 different, randomly initialized optimization runs, each run produced a controller that satisfied the linear requirements. Also, the associated Lyapunov function guaranteed stability in an area of the state space approximately of the size shown in figure 3 within $\pm 5\%$ deviation.

4.3 Anti-lock Braking Example

Additionally, we test the proposed method on a nonlinear anti-lock braking system (ABS) with longitudinal vehicle dynamics using a quarter car model. This model serves as another benchmark problem to demonstrate the method. The system equations are

$$J\dot{\omega} = -rF_x - T_b \operatorname{sign}(\omega) \tag{39}$$

$$m\dot{v} = F_x - F_a\,. \tag{40}$$

For details see for example (Petersen et al., 2001). Here, ω is the angular velocity of the wheel, v the longitudinal velocity of the vehicle and T_b the brake torque. Further, F_x is the tire friction force, which can be computed by the Pacejka tire model (Pacejka, 2005) as shown in (41) and F_a the aerodynamic drag force (42), given by

$$F_x(\lambda,\mu) = \mu m g \sin\left(C_p \arctan\left(B_p \frac{\lambda}{\mu}\right)\right), \tag{41}$$

$$F_a(v) = \frac{1}{2} c_w \rho A_s v|v|\,, \tag{42}$$

with λ, the longitudinal wheel slip defined as

$$\lambda = \frac{\omega r - v}{\max(|\omega| r, |v|)}\,. \tag{43}$$

The friction coefficient between tire and road is $\mu \in (0,1]$. The other parameters are listed in table 3.

Table 3: Vehicle model parameters.

Name	Description	Value	Unit
m	Quarter car mass	450	kg
J	Wheel moment of inertia	3	$\mathrm{kg\,m^2}$
r	Wheel radius	0.33	m
ρ	Air density	1.1	$\mathrm{kg\,m^{-3}}$
A_s	Vehicle front surface	2.4	$\mathrm{m^2}$
c_w	Aerodynamic drag coef.	0.3	–
g	Gravitational constant	9.81	$\mathrm{m\,s^{-2}}$
B_p	Tire model slope coef.	10	–
C_p	Tire model shape coef.	1.3	–
T_{act}	Actuator time constant	0.025	s

Since the dynamics of the vehicle velocity evolve significantly slower than the angular wheel velocity, it is common to assume $\dot{v} \approx 0$. For the braking case, the slip dynamics can be derived, compare (Petersen et al., 2001). The state vector includes the tire slip error $x_1 = \lambda - \lambda_{\mathrm{set}}$, its integral $x_2 = \int_0^t \lambda - \lambda_{\mathrm{set}}\, d\tau$ and actuator dynamics x_3 as $x = \begin{bmatrix} x_1 & x_2 & x_3 \end{bmatrix}^T$. The state equations of the augmented error dynamics including actuator dynamics are given as

$$\dot{x_1} = -\frac{r}{Jv}\left(rF_x - u_{\max}(x_3 + u^*)\right) - \frac{F_x}{mv}\left(1-\lambda\right), \tag{44}$$

$$\dot{x_2} = x_1\,, \tag{45}$$

$$\dot{x_3} = \frac{1}{T_{\mathrm{act}}}\left(u - x_3\right), \tag{46}$$

with $\lambda = \lambda_{\mathrm{set}} + x_1$, $\dot{\lambda}_{\mathrm{set}} = 0$ and u^* required for shifting the equilibrium of the system to the origin given as

$$u^* = F_x(\lambda_{\mathrm{set}},\mu)\frac{m r^2 - J(\lambda_{\mathrm{set}} - 1)}{u_{\max}\, r m}\,. \tag{47}$$

The limit for the input is $|u| \le 3000\,\mathrm{Nm} = u_{\max}$. The region in the state space is defined as $x_1 \in [-1,1]$, $x_2 \in [-0.2, 0.2]$, $x_3 \in [-1,1]$, the parameters for the Γ-region as $T_{\max} = 0.1\,\mathrm{s}$, $\zeta_{\min} = 0.4$ and $\omega_{\max} = 30\,\mathrm{s}^{-1}$. For the optimization we set $\dot{\lambda}_{\mathrm{set}} = 0, \lambda_{\mathrm{set}} = 0.5, \mu = 0.2$ and $v = 50\,\mathrm{km/h}$. After several iterations, the GE algorithm produces the following control law of non obvious structure

$$u_{\mathrm{GE}} = -x_1 - 10.158x_2 + x_2 \arctan(\psi_1)\,, \tag{48}$$

$$\psi_1 = x_1 - x_2 + \psi_2\psi_3\,, \tag{49}$$

$$\psi_2 = \arctan(x_1^2 + 11.16x_2)\,, \tag{50}$$

$$\psi_3 = \psi_2 + 11.16x_2(x_2 - 11.16)\,. \tag{51}$$

The Lyapunov function associated to system (46) with controller (48) saturated by (29) is derived to

$$V(x) = x^T P x = x^T \begin{bmatrix} 0.0869 & 0.4663 & 0.0247 \\ 0.4663 & 9.9604 & 0.0012 \\ 0.0247 & 0.0012 & 0.0271 \end{bmatrix} x\,. \tag{52}$$

The eigenvalues of the closed loop system are located at $(-11.13, \pm 20.68)$ and $(-17.51, 0)$, matching the specified requirements. We now use Jacobian linearization of (48) to generate a linear PI controller,

$$u_{\mathrm{PI}} = -x_1 - 10.158x_2\,. \tag{53}$$

Notice that as in the previous example, both controllers (48) and (53) have the same linearized closed loop eigenvalues and therefore the same quadratic Lyapunov function. Figure 4 shows the DoA estimates of the two methods on system (46).

The GE controller shows a DoA that is approximately 3.55 times larger than that of the PI controller. This estimate was obtained using algorithm 1 with a sample size of $N = 10^6$. The nonlinearity in the

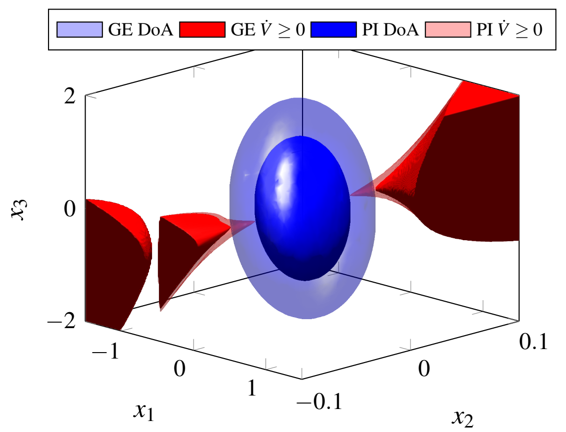

Figure 4: Comparison of domains of attraction.

control law (48) seems to be beneficial for increasing the guaranteed DoA in this case. Additionally, we evaluate the nonlinear controller on the original system (40). For the simulation, the controller (48) is discretized with a sample time of 10 ms and acts as a limiter for the brake force commanded by the driver. The integrator state x_2 is fed back with anti-windup. Furthermore, we add Gaussian white noise with a signal-to-noise ratio of 50 dB to both ω and v in order to simulate measurement noise. Communication delay is modeled with a time delay of 10 ms for both the actuator and the sensors. The simulation starts at $v = 130\,\mathrm{km/h}$ and $\lambda = 0$ on dry asphalt with $\mu = 1$. The driver starts braking at $t = 0$ with 40% braking force. At $t = 1\,\mathrm{s}$, the friction coefficient of the road drops to $\mu = 0.2$ (snow). Therefore, the setpoint is changed to $\lambda_{\mathrm{set}} = -3\%$. The controller gets activated at $t = 1.02\,\mathrm{s}$, 20 ms after the event at $t = 1\,\mathrm{s}$. Figure 5 shows the wheel slip λ as well as λ_{set}, v, ω and the control signal T_b. The controller (48) identified by the GE algorithm successfully stabilizes the nonlinear wheel dynamics. The wheel slip reaches a peak value of $\approx 40\%$. The controller prevents the wheels from decelerating further and therefore prevents locking of the wheels.

One optimization run took again approximately 30 seconds on average. Figure 6 shows an example error evolution of the best individual during one optimization run. It can be seen there that the linear requirements are fulfilled at generation 281, where $\phi \leq 1$ according to (28a) and (28b). At this point, the constraints (27e)-(27i) are satisfied, as the poles of the linearized closed loop system are located in the specified Γ-region. Therefore, the Lyapunov equation (27f)-(27g) is guaranteed to have a unique solution and a positive definite $V(x)$ can be generated. From thereon, the size of the guaranteed domain of attraction is increased over the subsequent generations.

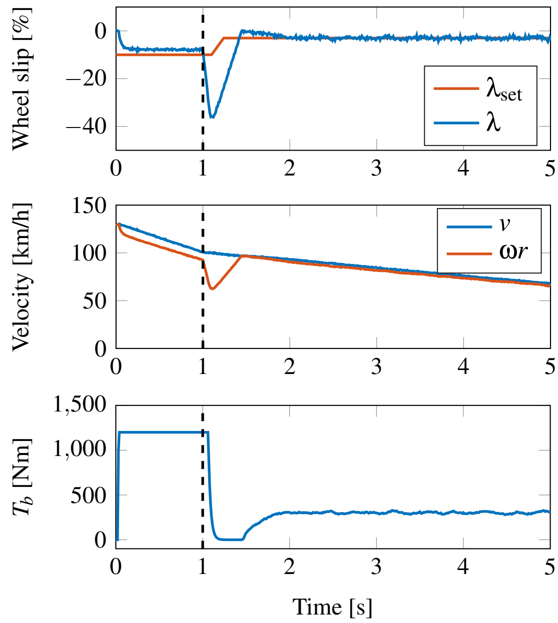

Figure 5: Simulation of the tire slip controller (48).

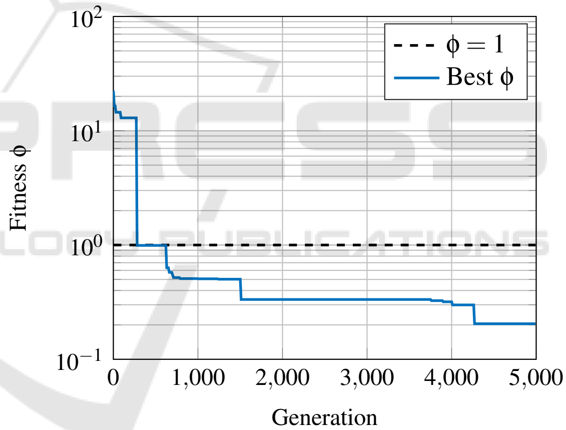

Figure 6: Example error evolution during the optimization.

Again, we evaluated the method for 100 different, randomly initialized optimization runs, using the parameters from table 2 for the GE algorithm. As for the nonlinear oscillator benchmark system from section 4.2, each run produced a controller that placed the poles of the linearized closed loop system in the desired Γ-region. This was despite the fact that all runs started with all, or at least some of the poles being located outside of the Γ-region. The distance of the poles to the centroid of the Γ-region was successfully used as a performance measure for pole placement. The average fitness achieved over all 100 optimization runs was 0.4419 with a variance of 0.0662.

5 CONCLUSION

A new method for controller synthesis based on grammatical evolution has been proposed. We derived an explicit formula for the centroid of the considered stability region used for pole placement and integrated it into a GE optimization algorithm. Additionally, the GE algorithm was extended to maximize the guaranteed domain of attraction by using Monte Carlo integration in order to solve a bilevel optimization problem. On the considered nonlinear oscillator system and the ABS control system, the method showed convincing performance for maximizing the guaranteed domain of attraction while satisfying the specified constraints. For the ABS system, the generated controller was validated by simulation. One limitation of the presented approach is that it is computationally expensive such that it cannot be applied as an online optimization method.

Future work can extend the proposed approach in different directions. One possibility is to include further linear performance measures into the optimization process. For example, the parameter space approach (Ackermann et al., 2002) provides measures to ensure robustness of the controller with respect to uncertain parameters for the linearized system. Also, different Γ-regions than the proposed one might be used for the optimization. Further, it would be interesting to evaluate more algorithms for estimating the domain of attraction, for example using a sampling algorithm with memory as proposed in (Najafi et al., 2016). Also, the MIMO (multi-input, multi-output) case could be investigated. Finally, one could remove the assumption of a quadratic Lyapunov function and let the GE algorithm search for the structure of both the Lyapunov function and the control laws at the same time. However, as mentioned in remarks 4 and 5, ensuring positive definiteness in this general setup might be challenging. Future work could therefore focus on creating algorithms that can reliably detect positive semi-definiteness of general, non-quadratic Lyapunov functions.

The presented approach is applicable for nonlinear control synthesis for a large range of nonlinear systems with different types of nonlinearities. Input constraints and performance specifications adopted from linear control theory can be incorporated into the control design process. In general, the system order is not restricted. The method presents a nonlinear control structure design tool which is applicable to a variety of practical applications. Results suggest that reasonable stability guarantees are achieved with the combination of structural control law optimization and domain of attraction maximization.

REFERENCES

Ackermann, J., Blue, P., Bünte, T., Güvenc, L., Kaesbauer, D., Kordt, M., Muhler, M., and Odenthal, D. (2002). *Robust control: the parameter space approach*. Springer Science & Business Media.

Backus, J. W. (1959). The syntax and semantics of the proposed international algebraic language of the Zurich ACM-GAMM conference. *Proceedings of the International Conference on Information Processing*.

Banks, C. (2002). Searching for Lyapunov functions using genetic programming. Virginia Polytech Institute and State University, Blacksburg, Virginia. Available online (last visit 24/05/2018): http://www.aerojockey.com/files/lyapunovgp.pdf.

Bartels, R. H. and Stewart, G. (1972). Solution of the matrix equation $AX + XB = C$: Algorithm 432. *Communications of the ACM*, 15(9):820–826.

Chen, P. and Lu, Y.-Z. (2011). Automatic design of robust optimal controller for interval plants using genetic programming and Kharitonov theorem. *International Journal of Computational Intelligence Systems*, 4(5):826–836.

Chen, Z., Mi, C. C., Xiong, R., Xu, J., and You, C. (2014a). Energy management of a power-split plug-in hybrid electric vehicle based on genetic algorithm and quadratic programming. *Journal of Power Sources*, 248:416–426.

Chen, Z., Yuan, X., Ji, B., Wang, P., and Tian, H. (2014b). Design of a fractional order PID controller for hydraulic turbine regulating system using chaotic non-dominated sorting genetic algorithm II. *Energy Conversion and Management*, 84:390–404.

Chilali, M. and Gahinet, P. (1996). H_∞ design with pole placement constraints: an LMI approach. *IEEE Transactions on automatic control*, 41(3):358–367.

Cupertino, F., Naso, D., Salvatore, L., and Turchiano, B. (2002). Design of cascaded controllers for DC drives using evolutionary algorithms. In *Evolutionary Computation, 2002. CEC'02. Proceedings of the 2002 Congress on*, volume 2, pages 1255–1260. IEEE.

Doyle, J., Primbs, J. A., Shapiro, B., and Nevistic, V. (1996). Nonlinear games: examples and counterexamples. In *Decision and Control, 1996., Proceedings of the 35th IEEE Conference on*, volume 4, pages 3915–3920. IEEE.

Gholaminezhad, I., Jamali, A., and Assimi, H. (2014). Automated synthesis of optimal controller using multi-objective genetic programming for two-mass-spring system. In *Robotics and Mechatronics (ICRoM), 2014 Second RSI/ISM International Conference on*, pages 041–046. IEEE.

Giesl, P. and Hafstein, S. (2015). Review on computational methods for Lyapunov functions. *Discrete and Continuous Dynamical Systems, Series B*, 20(8):2291–2337.

Glover, F. W. and Kochenberger, G. A. (2006). *Handbook of metaheuristics*, volume 57. Springer Science & Business Media.

Graichen, K., Kugi, A., Petit, N., and Chaplais, F. (2010). Handling constraints in optimal control with saturation functions and system extension. *Systems & Control Letters*, 59(11):671–679.

Grosman, B. and Lewin, D. R. (2005). Automatic generation of Lyapunov functions using genetic programming. *IFAC Proceedings Volumes*, 38(1):75–80.

Haupt, R. L. and Haupt, S. E. (1998). *Practical genetic algorithms*, volume 2. New York: Wiley.

Holland, J. H. (1975). *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence.* University of Michigan Press.

Hosen, M. A., Khosravi, A., Nahavandi, S., and Creighton, D. (2015). Improving the quality of prediction intervals through optimal aggregation. *IEEE Transactions on Industrial Electronics*, 62(7):4420–4429.

Koza, J. R. (1992). *Genetic programming: on the programming of computers by means of natural selection*, volume 1. MIT press.

Koza, J. R., Keane, M. A., Yu, J., Bennett, F. H., Mydlowec, W., and Stiffelman, O. (1999). Automatic synthesis of both the topology and parameters for a robust controller for a nonminimal phase plant and a three-lag plant by means of genetic programming. In *Decision and Control, 1999. Proceedings of the 38th IEEE Conference on*, volume 5, pages 5292–5300. IEEE.

Koza, J. R., Keane, M. A., Yu, J., Mydlowec, W., and Bennett, F. H. (2000). Automatic synthesis of both the topology and parameters for a controller for a three-lag plant with a five-second delay using genetic programming. In *Workshops on Real-World Applications of Evolutionary Computation*, pages 168–177. Springer.

Lyapunov, A. M. (1992). The general problem of the stability of motion. *International Journal of Control*, 55(3):531–534.

McGough, J. S., Christianson, A. W., and Hoover, R. C. (2010). Symbolic computation of Lyapunov functions using evolutionary algorithms. In *Proceedings of the 12th IASTED International Conference*, volume 15, pages 508–515.

Mirafzal, S. H., Khorasani, A. M., and Ghasemi, A. H. (2016). Optimizing time delay feedback for active vibration control of a cantilever beam using a genetic algorithm. *Journal of Vibration and Control*, 22(19):4047–4061.

Najafi, E., Babuška, R., and Lopes, G. A. (2016). A fast sampling method for estimating the domain of attraction. *Nonlinear Dynamics*, 86(2):823–834.

Naur, P., Backus, J. W., Bauer, F. L., Green, J., Katz, C., McCarthy, J., and Perlis, A. J. (1969). *Revised report on the algorithmic language Algol 60*. Springer.

Pacejka, H. (2005). *Tyre and vehicle dynamics*. Butterworth-Heinemann, Oxford, 2 edition.

Petersen, I., Johansen, T. A., Kalkkuhl, J., and Lüdemann, J. (2001). Wheel slip control in ABS brakes using gain scheduled constrained LQR. In *European Control Conference (ECC)*, pages 606–611. IEEE.

Poli, R., Langdon, W. B., and McPhee, N. F. (2008). *A field guide to genetic programming*. Published via http://lulu.com and freely available online (last visit 24/05/2018) http://www.gp-field-guide.org.uk. (With contributions by J. R. Koza).

Prajna, S., Papachristodoulou, A., and Parrilo, P. A. (2002). Introducing SOSTOOLS: A general purpose sum of squares programming solver. In *Decision and Control, 2002, Proceedings of the 41st IEEE Conference on*, volume 1, pages 741–746. IEEE.

Precup, R.-E., Sabau, M.-C., and Petriu, E. M. (2015). Nature-inspired optimal tuning of input membership functions of Takagi-Sugeno-Kang fuzzy models for anti-lock braking systems. *Applied Soft Computing*, 27:575–589.

Reichensdörfer, E., Odenthal, D., and Wollherr, D. (2017). Grammatical evolution of robust controller structures using Wilson scoring and criticality ranking. In *European Conference on Genetic Programming*, pages 194–209. Springer.

Ryan, C., Collins, J., and Neill, M. O. (1998). Grammatical evolution: evolving programs for an arbitrary language. In *European Conference on Genetic Programming*, pages 83–96. Springer.

Saadat, J., Moallem, P., and Koofigar, H. (2017). Training echo state neural network using harmony search algorithm. *International Journal of Artificial Intelligence*, 15(1):163–179.

Saleme, A., Tibken, B., Warthenpfuhl, S. A., and Selbach, C. (2011). Estimation of the domain of attraction for non-polynomial systems: A novel method. *IFAC Proceedings Volumes*, 44(1):10976–10981.

Shimooka, H. and Fujimoto, Y. (2000). Generating robust control equations with genetic programming for control of a rolling inverted pendulum. In *Proceedings of the 2nd Annual Conference on Genetic and Evolutionary Computation*, pages 491–495. Morgan Kaufmann Publishers Inc.

Singh, R., Kuchhal, P., Choudhury, S., and Gehlot, A. (2015). Implementation and evaluation of heating system using PID with genetic algorithm. *Indian Journal of Science and Technology*, 8(5):413.

Sontag, E. D. (1983). A Lyapunov-like characterization of asymptotic controllability. *SIAM Journal on Control and Optimization*, 21(3):462–471.

Tsuzuki, T., Kuwada, K., and Yamashita, Y. (2006). Searching for control Lyapunov-Morse functions using genetic programming for global asymptotic stabilization of nonlinear systems. In *Proceedings of the 45th IEEE Conference on Decision and Control*, pages 5114–5119. IEEE.

Verdier, C. and Mazo Jr, M. (2017). Formal controller synthesis via genetic programming. *IFAC-PapersOnLine*, 50(1):7205–7210.

Vrkalovic, S., Teban, T.-A., and Borlea, I.-D. (2017). Stable Takagi-Sugeno fuzzy control designed by optimization. *Int. J. Artif. Intell*, 15:17–29.

Zhou, K., Doyle, J. C., Glover, K., et al. (1996). *Robust and optimal control*, volume 40. Prentice hall New Jersey.