

INTELIGENCIA ARTIFICIAL

http://journal.iberamia.org/

Genetic Algorithms for Satellite Launcher Attitude Controller Design

Paulo R. Silva^{1,A}, Ivanildo S. Abreu^{1,B}, Paulo A. Forte^{1,C}, Henrique M. C. do Amaral^{1,D}

Abstract For proper attitude control of space-crafts conventional optimal Linear Quadratic (LQ) controllers are designed via trial-and-error selection of the weighting matrices. This time consuming method is inefficient and usually results in a high order complex controller. Therefore, this work proposes a genetic algorithm (GA) for the search problem of the attitude controller gains of a satellite launcher. The GA's fitness function considers some control features as eigenstructure, control goals and constraints. According to simulation results, the search problem of controller parameters with evolutionary algorithms was faster than usual approaches and the designed controller reached all the specifications with satisfactory time responses. These results could improve engineering tasks by speeding up the design process and reducing costs.

Keywords: Genetic Algorithm, Optimal control, Attitude control

1 Introduction

Rockets and space-crafts engineering have led substantially improvements in communications, navigation, space and earth observation, bringing progress to society. In order to travel safely through space, these non-linear vehicles need good navigation and guidance modules with embedded digital controllers to control attitude angles and velocities [10, 14, 13, 7].

Modern optimal control strategies design controllers for linear time invariant systems to satisfy desired specifications by minimizing a quadratic performance index [4]. This index directly affects the control outcome and includes states and control vectors that must be weighted by user defined matrices. Selection of these matrices is not straightforward as it requires familiarity with the subject and massive simulations for refinement [16, 3].

Usage of Evolutionary approaches to tune optimal controllers has been stated in the literature in a range of areas producing exceptional results and reducing the time spent in the design process. In [15], the authors successfully tuned a Linear Quadratic Regulator (LQR) and Proportional-Integral-Derivative (PID) controllers with a Genetic Algorithm (GA) for the aircraft pitch control problem and demonstrated that the LQR is better than PID. [8] concluded that a GA-tuned LQR controller for the magnetically actuated attitude control of CubeSats is better than a simple LQR and a Proportional-Derivative, resulting in smaller steady state error and faster time response. [11] obtained optimal control gains via Genetic Algorithms, the GA-based controller is superior since the conventional tuning techniques

¹ Computer Engineering Department, at State University of Maranhao, Brazil.

^Apaulosilva14@aluno.uema.br

 $^{^{}B}$ ivanildoabreu@professor.uema.br

 $^{^{}C} {\tt pauloforte@aluno.uema.br}$

 $^{^{}D}$ henriqueamaral@professor.uema.br

is not effective due to unseen non-linearities of the tracker robot. [5] proposed a neural-genetic controller for the attitude control problem of a nonlinear satellite in chaotic motion due to large external motions without any previous knowledge of the system dynamics.

This work proposes a Genetic Algorithm approach for gain computing to the attitude control system of a satellite launcher, given that [2] proposed an analytical method for computing the controller gains of a satellite launcher by tuning the weighting matrices empirically.

This paper is organized as follows. Section 2 presents the satellite launcher longitudinal model and the Linear Quadratic method. Section 3 demonstrates the methodology used in this work. The Genetic Algorithm simulation results are presented in Section 4. Finally, conclusions are given.

2 Background

2.1 Model Description

The equations of motion are derived for a simplified model considering the forces acting on the body. As can be seen in Figure 1, the simplified space-craft is subject to aerodynamic forces, \vec{F}_{aero} , gravitational force, \vec{W} , thrust force, \vec{T} , due to the propellant burn and wind velocity, V_{wind} . Also in this model, the angle of attack, α , is the angle between the body reference line, u, and the oncoming wind velocity vector, V_{wind} . The attitude pitch angle is controlled by β_z , that represents the angular deflection of the boosters.

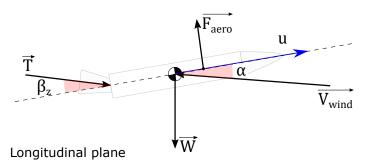


Figure 1: Simplified model of satellite launcher (longitudinal plane).

The simplified state-space model for the rigid-body dynamics of a launcher is given in [2] as

$$\begin{bmatrix} \dot{\theta} \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ \mu_{\alpha} & 0 \end{bmatrix} \begin{bmatrix} \theta \\ \dot{\theta} \end{bmatrix} + \begin{bmatrix} 0 \\ \mu_{\beta z} \end{bmatrix} \beta_{z} \tag{1}$$

where

 $\theta \rightarrow \text{pitch attitude angle};$

 $\beta_z \to \text{Booster deflection}$ angle due to actuator deflection;

 μ_{α} and $\mu_{\beta z} \to \text{angular acceleration coefficients.}$

2.2 The Linear Quadratic Method

As stated by [17], the Linear Quadratic (LQ) method for a system with state space representation given by

$$\dot{x} = Ax + Bu \tag{2}$$

focus on finding a state-feedback control input

$$u = -Kx \tag{3}$$

that minimizes a quadratic cost function,

$$J = \frac{1}{2} \int_{t_0}^{T} [x^T Q x + u^T R u] dt$$
 (4)

where Q is a positive semi-definite weighting matrix of the state vector, x, and R is a positive definite weighting matrix of the control input vector, u.

The optimal feedback gain matrix, K, is given by

$$K = R^{-1}B^TP (5)$$

where P is a symmetric matrix solution of the algebraic Ricatti equation (ARE),

$$A^{T}P + PA - PBR^{-1}B^{T}P + Q = 0 (6)$$

3 Methodology

The control structure, GA model and operators used in this work are discussed in the following subsections.

3.1 Control Structure

Figure 2 illustrates the control structure used in this work. In this structure the control input, β_z , is computed according to a PI controller that computes its output based on the attitude error $(\theta_{ref} - \theta)$ and an angular velocity feedback $(d\theta/dt)$. This structure ensures better tracking to reference commands, good robustness and temporal performance, [2].

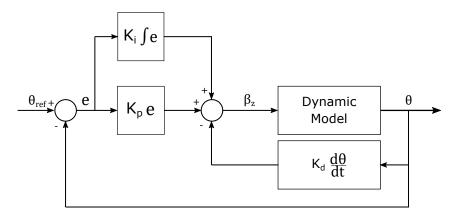


Figure 2: Control structure for the controller design.

Closed-loop model The closed loop state space model for the block diagram presented in Figure 2 is given as follows

$$\begin{bmatrix} \dot{x}_{2\times1} \\ \dot{\tau} \end{bmatrix} = \begin{bmatrix} A_{2\times2} & 0_{2\times1} \\ -1 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_{2\times1} \\ \tau \end{bmatrix} + \begin{bmatrix} B_{2\times1} \\ 0 \end{bmatrix} \beta_z + \begin{bmatrix} 0_{2\times1} \\ 1 \end{bmatrix} \theta_{ref} \tag{7}$$

where τ is the error integral

$$\tau = \int \theta_{ref}(t) - \theta(t)dt \tag{8}$$

and the control input, β_z , is given by

$$\beta_z = \begin{bmatrix} -Kp & -Kd & Ki \end{bmatrix} \cdot \begin{bmatrix} x_{2\times 1} \\ \tau \end{bmatrix} + K_p \theta_{ref}$$
 (9)

Once the system is in the form $\dot{x} = Ax + Bu$, the linear quadratic method can be used to find control gains in Equation 9. The concern now is how the control problem will be encapsulated in Genetic Algorithms and how the GA will converge to good weighting matrices Q and R that lead optimal control gains.

3.2 Genetic Algorithm Models and Operators

This part presents the genetic models and operators proposed for this work.

Chromosome Model Since $Q_{n\times n}$ and $R_{m\times m}$ are symmetric positive-definite matrices satisfying the linear quadratic specifications, the chromosome model can be given as a diagonal of $Q \in R$, [3]. The total genes is

$$g = n + m \tag{10}$$

The resulting chromosome is then

$$QR_z = [q_{11} \ q_{22} \ \dots \ q_{nn} \ r_{11} \ r_{12} \ \dots \ r_{nn}] \tag{11}$$

Population Model A population is a set of chromosomes. If a chromosome with g genes contains Q e R, then a population is represented by $QR_{n_{indiv}\times g}$, where n_{indiv} is the number of individuals in the population.

Fitness Model The fitness function evaluates each individual in a population and ensures the GA will find a optimal solution. The function is given by [3] as:

$$K_{z} = LQR_{z}(A, B, Q_{z}, R_{z})$$

$$A_{z} = (A - BK_{z})$$

$$S_{z} = \frac{||V_{z}||^{2}||W_{z}||^{2}}{\langle V_{z}, W_{z} \rangle}$$

$$F_{S_{z}} = \sum_{z} S_{z}$$

$$R_{S_{z}} = rank(S_{z}, F_{S_{z}})$$
(12)

where $z = 1, ..., n_{indiv}$, A_z is the closed-loop matrix for the gain vector K_z . S_z is the sensibility, V_z and W_z are eigenvectors of A_z . F_{S_z} is the fitness and R_{S_z} represent each individual fitness. Additionally, each individual is graded according to user-defined control goals.

The fitness model in Equation 12 scores each individual based on its current location in the s-plan. If the closed-loop poles from A_z are located inside the user defined eigenstructure (red zone) of Figure 3 the fitness model will ensure a high score for this chromosome.

Elite Selection The elite selection ensures that the best individuals (highest fitness) of a given population will survive in the next generation. This operator avoids the fittest individuals being lost in crossover and mutation operations, [1].

The algorithm for the Elite operator is given by

Algorithm 1 Elite

```
mean = \sum_{j=1}^{m} f_i/m

for i = 1 \rightarrow m do

if f_i > mean then

Select individual i

end if

end for
```

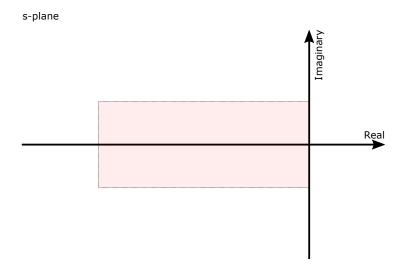


Figure 3: Fitness evaluation according to eigenstructure.

Roulette Selection This operator selects individuals based on the fitness. The operator can be given as a random experiment where

$$P[b_{j,t} \ selection] = \frac{f(b_{j,t})}{\sum_{k=1}^{m} f(b_{k,t})}$$

$$\tag{13}$$

The algorithm for the Roulette selection is given by [6] as

Algorithm 2 Roulette

```
p_i = f_i / \sum_{j=1}^n f_j
q_i = \sum_{j=1}^i p_j
while i < m do
x = random(0, 1);
if r < q_i then
Select individual i
end if
end while
```

Crossover Operator The crossover operator combines two individuals randomly in order to generate another two chromosomes.

The operator is given by [12] as

Algorithm 3 Crossover

```
\begin{array}{l} pos = random(1,...,g) \\ \textbf{for } i = 1 \rightarrow pos \ \textbf{do} \\ Child_1[i] = Parent_1[i]; \\ Child_2[i] = Parent_2[i]; \\ \textbf{end for} \\ \textbf{for } i = pos + 1 \rightarrow n \ \textbf{do} \\ Child_1[i] = Parent_2[i]; \\ Child_2[i] = Parent_1[i]; \\ \textbf{end for} \end{array}
```

Mutation Operator This operator is essential as it avoids premature convergence, [12]. This operator randomly changes a gene of a given individual based on the probability of mutation, p_m .

The algorithm is given by

```
Algorithm 4 Mutation

for i = 1 \rightarrow n do

if random(0, 1) < p_m then

Mutate gene
end if
end for
```

4 Simulation Results

This section aims to present performance results of the Genetic Algorithm and a time-domain analysis of the control gains found by the proposed method.

4.1 Simplified Model

The simplified model of the space-craft that will be used is given by [2] as:

$$\begin{bmatrix} \dot{\theta} \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 4.16 & 0 \end{bmatrix} \begin{bmatrix} \theta \\ \dot{\theta} \end{bmatrix} + \begin{bmatrix} 0 \\ 7.21 \end{bmatrix} \beta_z \tag{14}$$

1.5

4.2 GA Parameters and Results

Overshoot

For the proposed work, the parameters used for the GA simulation are presented in Table 1. The table also presents the control goals and constraints used in the controller design process, the sensitivity values that were include in order to enable the user to weight which parameter would be more valuable.

Parameters Population size 120 Number of individuals produced by Elite 20 Number of individuals produced by Roulette 20 Number of individuals produced by Crossover 20 Number of individuals produced by Mutation 40 Number of new individuals 20 Mutation probability 5 Mutation factor 0.1 Goals and constraints Eigenstructure $-8 \pm 1.5j$ to $0 \pm 1.5j$ Settling time $t_s < 3$ sec. Rise time $t_r < 1 \text{ sec.}$ %OS < 40%Overshoot Sensitivities Eigenstructure 1 Settling time 1.2 Rise time 1.2

Table 1: Genetic Algorithm parameters.

Initial population Initial population was randomly created with 120 individuals. Mean fitness of this population was 1.625 and the best individual presented fitness of 3.9. As can be noted in Figure 4, initial population presented good diversity that is essential to avoid local maxima or minima.

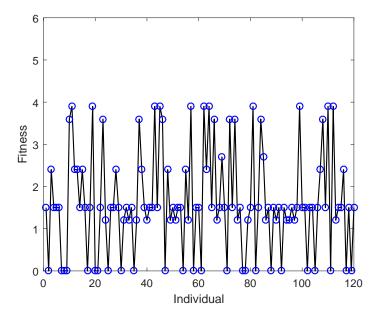


Figure 4: Initial population fitness.

Final population Final population mean fitness was 4. As can be noted in Figure 5, diversity was reduced suggesting that GA is close to the stop criterion. Last individuals of this population presented poor fitness as they were recently created.

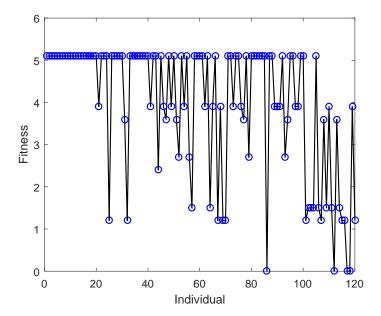


Figure 5: Final population fitness.

Fitness Evolution Figure 6 highlights the evolution of the mean fitness of populations. It can be noted that GA met the stop criteria with 50 iterations.

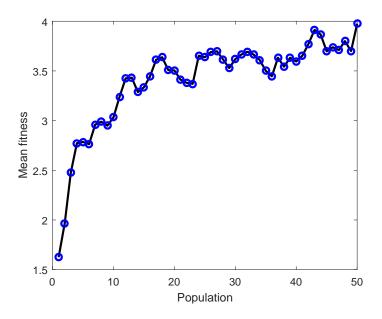


Figure 6: Fitness evolution.

Statistical analysis A Monte-Carlo simulation suggests that the algorithm converges with 30 populations as depicted by Figure 7.

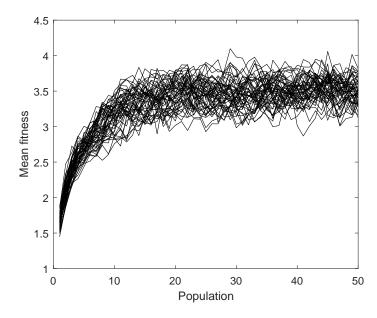


Figure 7: Monte Carlo simulation.

Additionally, for the proposed eigenstructure, goals and constraints, given in Table 1, the algorithm suggests that $0.1 < Q_{11} < 0.2$, $Q_{22} < 0.1$, $Q_{33} \approx 0.5$ and R < 0.1 often result in optimal gains, as can be seen in Figure 8.

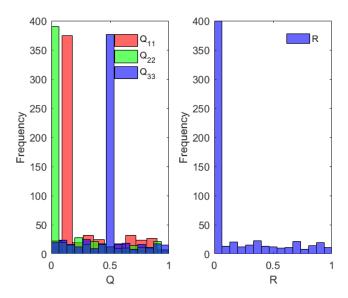


Figure 8: Histogram of Q and R.

4.3 Time domain analysis

The fittest chromosome for the 35-th population represents the following weighting matrices

$$Q = \begin{bmatrix} 0.1523 & 0 & 0 \\ 0 & 0.0892 & 0 \\ 0 & 0 & 0.5 \end{bmatrix} \text{ and } R = 0.089 \text{ that leads to } K = \begin{bmatrix} 3.5156 & 1.4023 & -2.3570 \end{bmatrix}$$

The closed-loop step response for these gains is represented in Figure 9. The time domains specifications for this closed-loop step response follows: $t_s = 3.655$ seconds, $t_r = 0.429$ seconds and %OS = 34%.

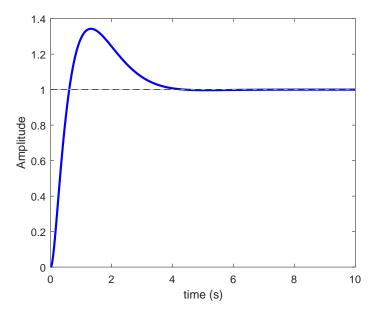


Figure 9: Step response for the controlled plant with matrices produced by the GA.

These specifications are due to the location of the closed loop poles inside the desired eigenstructure, as can be noted in Figure 10.

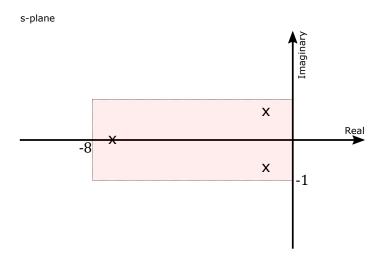


Figure 10: Poles in the desired eigenstructure.

Algebraic Method Comparison

The corresponding weighting matrices obtained via the methodology proposed in [9] are:

$$Q = \begin{bmatrix} 0.0048 & 0 & 0 \\ 0 & 0.0179 & 0 \\ 0 & 0 & -0.0049 \end{bmatrix} \text{ and } R = 0.001 \text{ that leads to } K = \begin{bmatrix} 0.6958 & 2.1909 & 0.3477 \end{bmatrix}$$

As can be seen in Figure 11 the closed loop step response produced by the algebraic approach is not acceptable as it does not follow the reference line (dashed black) in a finite time. Also, the weighting matrix Q presents a negative term that is not in accordance with the theory presented in Chapter 2. Furthermore, algebraic or analytical methods often requires expertise on the dynamic behaviour of a system. Usage of evolutionary approaches in this case is very welcomed by engineers since they often bring better results in a short time.

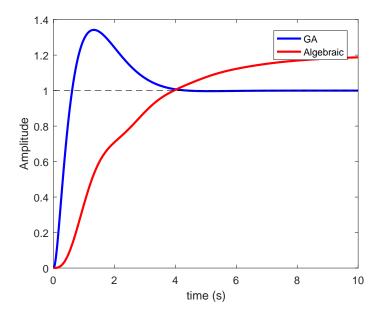


Figure 11: Step response comparison.

Comments on GA approach The algorithm converges in a finite number of iterations. Although, the time domain analysis shows that the settling time goal is not met, this is due to sensitivity values in Table 1. For this simulation, as the overshoot sensitivity is greater, the algorithm prioritize this parameter. Thus, in any other simulation with settling time sensitivity greater than overshoot the algorithm would find a satisfying solution.

In order to avoid the excitation of undesired dynamics mentioned in [2], the rise time goal must be made greater, however this effect was neglected here as this is not the focus of the research.

The effect of each parameter was evaluated. Mutation probability and mutation factor are the key parameters as they directly affect the speed of convergence. Very high or very low values of these parameters make the GA diverge and not to find a solution. Eigenstructure size can also smash the convergence, in this case the parameter was made smaller over time.

5 Conclusion

In this paper, a genetic algorithm for the control gains search problem of the satellite launcher attitude controller gains was proposed since currently techniques require prior experience about the problem and often result in inefficient controller.

Overall results show that the proposed method reaches the design specifications with 30 iterations with a population of 120 elements. The study also suggested the values of the weighting matrices (0.1 < Q_{11} < 0.2, Q_{22} < 0.1, $Q_{33} \approx 0.5$ and R < 0.1) to reach the design specifications. Indeed, usage of evolutionary techniques speeds up the search process and reduce design costs. Consequently, it is believed that the proposed approach can be used instead analytical methods.

For future work the authors will propose new fitness model approaches, usage of other evolutionary algorithms - neural networks or fuzzy logic - in the search problem and refine the control problem to a more realistic one.

Acknowledgements

The authors would like to thank Professor Alain Giacobini, Instituto Tecnológico de Aeronática, for his valuable comments on this work and Fundação de Amparo à Pesquisa e ao Desenvolvimento Científico e Tecnológico do Maranhão for the great opportunity to obtain a Master Degree.

References

- [1] Ulrich Bodenhofer. Genetic algorithms: Theory and applications. 2004.
- [2] D Carmona and WC Leite Filho. Analytical method for computing controller gains of a satellite launcher. *IFAC Proceedings Volumes*, 37(6):1019–1023, 2004. doi:10.1016/S1474-6670(17) 32313-3.
- [3] J. V. da Fonseca Neto, I. S. Abreu, and F. N. da Silva. Neural-genetic synthesis for state-space controllers based on linear quadratic regulator design for eigenstructure assignment. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 40(2):266–285, April 2010. doi:10.1109/TSMCB.2009.2013722.
- [4] S. Das, I. Pan, K. Halder, S. Das, and A. Gupta. Optimum weight selection based lqr formulation for the design of fractional order piλdμ controllers to handle a class of fractional order systems. In 2013 International Conference on Computer Communication and Informatics, pages 1–6, Jan 2013. doi:10.1109/ICCCI.2013.6466137.
- [5] D. C. Dracopoulos and A. J. Jones. Adaptive neuro-genetic control of chaos applied to the attitude control problem. *Neural Computing & Applications*, 6(2):102–115, Jun 1997. doi: 10.1007/BF01414007.

- [6] David E. Goldberg. Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1st edition, 1989.
- [7] Yu Han, James D Biggs, and Naigang Cui. Adaptive fault-tolerant control of spacecraft attitude dynamics with actuator failures. *Journal of Guidance, Control, and Dynamics*, 38(10):2033–2042, 2015. doi:10.2514/1.G000921.
- [8] Sarthak Kukreti, Alex Walker, Phil Putman, and Kelly Cohen. Genetic algorithm based lqr for attitude control of a magnetically actuated cubesat. In AIAA Infotech @ Aerospace, page 0886. 2015. doi:10.2514/6.2015-0886.
- [9] E. V. Kumar, J. Jerome, and K. Srikanth. Algebraic approach for selecting the weighting matrices of linear quadratic regulator. In 2014 International Conference on Green Computing Communication and Electrical Engineering (ICGCCEE), pages 1-6, March 2014. doi:10.1109/ICGCCEE.2014. 6922382.
- [10] F Landis Markley and John L Crassidis. Fundamentals of spacecraft attitude determination and control, volume 33. Springer, 2014. doi:10.1007/978-1-4939-0802-8.
- [11] M.H. Sangdani, A.R. Tavakolpour-Saleh, and A. Lotfavar. Genetic algorithm-based optimal computed torque control of a vision-based tracker robot: Simulation and experiment. *Engineering Applications of Artificial Intelligence*, 67:24 38, 2018. doi:10.1016/j.engappai.2017.09.014.
- [12] Kumara Sastry, David E. Goldberg, and Graham Kendall. *Genetic Algorithms*, pages 93–117. Springer US, Boston, MA, 2014. doi:10.1007/978-1-4614-6940-7_4.
- [13] Pyare Mohan Tiwari, S. Janardhanan, and Mashuq un Nabi. Rigid spacecraft attitude control using adaptive integral second order sliding mode. *Aerospace Science and Technology*, 42:50 57, 2015. URL: http://www.sciencedirect.com/science/article/pii/S1270963815000024, doi:https://doi.org/10.1016/j.ast.2014.11.017.
- [14] Pyare Mohan Tiwari, S. Janardhanan, and Mashuq un Nabi. Attitude control using higher order sliding mode. Aerospace Science and Technology, 54:108 113, 2016. URL: http://www.sciencedirect.com/science/article/pii/S1270963816301420, doi:https://doi.org/10.1016/j.ast.2016.04.012.
- [15] Vishal and J. Ohri. Ga tuned lqr and pid controller for aircraft pitch control. In 2014 IEEE 6th India International Conference on Power Electronics (IICPE), pages 1–6, Dec 2014. doi:10.1109/IICPE.2014.7115839.
- [16] C. Wongsathan and C. Sirima. Application of ga to design lqr controller for an inverted pendulum system. In 2008 IEEE International Conference on Robotics and Biomimetics, pages 951–954, Feb 2009. doi:10.1109/R0BIO.2009.4913127.
- [17] D. Xue, Y. Chen, and D. Atherton. *Linear Feedback Control*. Society for Industrial and Applied Mathematics, 2007. doi:10.1137/1.9780898718621.