Towards Intelligent Control via Genetic Programming

Francesco Marchetti, Edmondo Minisci, Annalisa Riccardi

5th workshop on Optimisation in Space Engineering (OSE)
Ljubljana, 21-22 November 2019
Table of Contents:

• Introduction
• Genetic Programming
• Intelligent Control
• Test Case
  • Results
• Conclusions and future work
Introduction

Background and Aim of the work

Aim:
• to investigate the use of Genetic Programming for the design of (intelligent) control systems

Rationale:
• Challenges presented by Hypersonic Vehicles, Space Access Vehicles and Space Exploration require a new approach to control design to improve performances and robustness beyond the limits of current standard approaches.

An Intelligent Control (IC) application using GP on a modified version of the Goddard Rocket test case is proposed.

Genetic Programming

- Originally introduced by Koza*.
- Pertain to the class of Evolutionary Algorithms.
- Allow the creation of computer programs following the laws of natural evolution.

GP applied to controller design:

- Both topology and parameters of the controller can be defined and optimized by GP**.
- When nonlinearities are introduced in the system, the definition of a new kind of controller is necessary.
- Unlike other ML techniques, GP produces mathematical equations which could be analysed by the user with classical control techniques.

What is intelligent control?

“An intelligent control system is designed so that it can autonomously achieve a high level goal, while its components, control goals, plant models and control laws are not completely defined, either because they were not known at the design time or because they changed unexpectedly.”*

Chosen approach to Intelligent control through GP:

- Thrust (control parameter) composed by the reference (obtained from open-loop optimal control) and a function of the relative errors created through GP.
- $f_{GP}$ is created when a change in the environment or in the system occurs.

Main issues:

- GP is computationally expensive.
- GP evaluation initial conditions must be defined a priori --> define evaluation time a priori!

Test Case: Goddard Rocket 2 Controls

The chosen test case is a modified version of the Goddard Rocket as a first step for a future application on a more complex model of a space access vehicle. The optimal trajectory was obtained with a Pseudospectral Collocation method*.

\[
\begin{align*}
\dot{r} &= v_r \\
\dot{\theta} &= \frac{v_t}{r} \\
\dot{v}_r &= \frac{T_r}{m} - \frac{D_r}{m} - g + \frac{v_r^2}{r} \\
\dot{v}_t &= \frac{T_t}{m} - \frac{D_t}{m} + \frac{v_t v_r}{r} \\
\dot{m} &= -\frac{\sqrt{T_r^2+T_t^2}}{g_0 I_{sp}}
\end{align*}
\]

Where:

- \( C_d = 0.6 \)
- \( S = 4.0 \, m^2 \)
- \( I_{sp} = 300 \, s \)
- \( m_0 = 100000 \, kg \)
- \( m_p = 0.99 \times m_p \)

Mission goals:

- reach an altitude of 400 km
- using less propellant as possible

\[
D_r = \frac{1}{2} \rho v_r \sqrt{v_r^2 + v_t^2 C_d S}
\]

\[
D_t = \frac{1}{2} \rho v_t \sqrt{v_r^2 + v_t^2 C_d S}
\]

\[
\rho = \rho_0 e^{-\beta r}
\]

Test Case: Goddard Rocket 2 Controls

Control scheme

\[ T_r = T_{r_{\text{ref}}} + f_{GP_r}(e_r, e_{v_r}) \]
\[ T_t = T_{t_{\text{ref}}} + f_{GP_t}(e_{\theta}, e_{v_t}) \]
3 scenarios are presented:

1. Variation of Cd at a random time and by a random magnitude.

2. Wind gust acting in a random altitude range with a constant random intensity in horizontal direction.

\[ v_r = v_r - v_{wind} \sin \theta \]
\[ v_t = v_t - v_{wind} \cos \theta \]

3. Real density model unknown at design time. Optimal trajectory obtained with simplified model and the real model is estimated during flight along with the controller law.

\[ \rho = \rho_0 e^{-\beta r} \quad \Rightarrow \quad \rho = \rho_{ISA}(r) \]

All simulations were run on a PC with 8GB of RAM and an Intel®Core™ i7-6700 CPU @3.40 GHz x 8 processors.
The scripts were coded in Python relying on the open source library DEAP*.

Results: Scenario 1 - Cd variation
Results: Scenario 2 - Wind Gusts
Results: Scenario 3 - Unknown density model
Results: Scenario 3 - Unknown density model
Results: Statistics

<table>
<thead>
<tr>
<th></th>
<th>Cd Variation</th>
<th>Wind Gust</th>
<th>Density Model</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>No learning</td>
<td>Learning</td>
<td>No learning</td>
</tr>
<tr>
<td>GP Evaluations</td>
<td>100</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>Fixed evaluation time</td>
<td>100 s</td>
<td>100 s</td>
<td>100 s</td>
</tr>
<tr>
<td>time interval</td>
<td>100 s</td>
<td>100 s</td>
<td>100 s</td>
</tr>
<tr>
<td>Min evaluation time</td>
<td>3.63 s</td>
<td>8.11 s</td>
<td>8.76s</td>
</tr>
<tr>
<td>Max evaluation time</td>
<td>06h08m50s</td>
<td>01h27m37s</td>
<td>38m41s</td>
</tr>
<tr>
<td>Time Constraint</td>
<td>66%</td>
<td>76%</td>
<td>44%</td>
</tr>
<tr>
<td>Success Rate</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Range Constraint</td>
<td>61%</td>
<td>64%</td>
<td>100%</td>
</tr>
<tr>
<td>Success Rate</td>
<td>51%</td>
<td>60%</td>
<td>44%</td>
</tr>
<tr>
<td>Total Success Rate</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Conclusions and Future Work

**Conclusions:**
- GP can be a powerful tool to design a control law, with very good generalization capabilities.
- The biggest issues for a practical use of GP for IC is its computational cost.

**Future work:**
- Implement more sophisticated methods to integrate models with "measured" data.
- Couple GP with an ANN, so to avoid the re-evaluation of the entire control law and perform the optimization of its components when needed.
Summary:

- Introduction
- Genetic Programming
- Intelligent Control
- Test Case
  - Results
- Conclusions and future work
References


THANKS

Francesco Marchetti, Edmondo Minisci, Annalisa Riccardi
{francesco.marchetti, edmondo.minisci, annalisa.riccardi}@strath.ac.uk
www.icelab.uk