Optimal Control via CPSO: From Theory to Implementation

Giovanni Giuseppe Iacuzzo

November 17, 2025

Abstract

This report investigates the optimal control of a second-order dynamical system using Continuous Particle Swarm Optimization (CPSO). The study covers theoretical formulation, numerical integration techniques, implementation in MATLAB, and analysis of results.

Contents

1 Introduction

The aim of this work is to recover a target control signal that drives a second-order system to a desired trajectory. The study combines theoretical background, numerical simulations, and population-based optimization methods.

The report is structured as follows:

- *Theoretical background:* formulation of the control problem.
- *Forward problem:* numerical methods for ODE integration.
- *CPSO implementation:* parameterization and optimization procedure.
- *Results:* post-processing, visualization, and analysis.

2 Theoretical Background

We consider a second-order system governed by the ODE:

$$u''(t) + \alpha u'(t) + \beta u(t) = g(t), \quad u(0) = u_0, \quad u'(0) = v_0, \tag{1}$$

where $u(t)$ is the state, $g(t)$ the control input, and α, β are system parameters.

The optimal control $g(t)$ minimizes the cost functional:

$$J(g) = \int_0^T \Big(u(t) - u_{\text{target}}(t)\Big)^2 \, dt + \lambda \sum_{i=1}^{M} p_i^2, \tag{2}$$

where $u_{\text{target}}(t)$ is the desired trajectory, p_i the control parameters, and λ a regularization weight.

The system is discretized using a second-order finite difference scheme:

$$u_{n+1} = \frac{1}{A}\Big(g_n + c_1 u_n - c_2 u_{n-1} - c_3 u_{n-1} - \beta u_n\Big), \tag{3}$$

$$A = \frac{1}{\Delta t^2} + \frac{\alpha}{2\Delta t}, \quad c_1 = \frac{2}{\Delta t^2}, \quad c_2 = \frac{1}{\Delta t^2}, \quad c_3 = \frac{\alpha}{2\Delta t}. \tag{4}$$

Initial values:

$$u_0 = u(0), \tag{5}$$

$$u_1 = u_0 + \Delta t \, v_0 + \frac{\Delta t^2}{2}\Big(g(0) - \alpha v_0 - \beta u_0\Big). \tag{6}$$

Control signals are parameterized using:

- *Piecewise constant segments.*
- *Cubic spline interpolation* of control parameters p_i.

Parameters are bounded: $p_{\min} \le p_i \le p_{\max}$.

3 Implementation

Implemented in MATLAB with key functions:

```
forward_solver.m        % numerical ODE integration
build_control.m         % constructs g(t) from parameters
objective_cpso.m        % evaluates cost function
postprocess.m           % results analysis and visualization
```

Workflow:

1. Load system parameters and target trajectory.
2. Initialize CPSO parameters and run optimization.
3. Compute system response using optimal control.
4. Generate plots and statistics.