

**IE415: Control of Autonomous Systems**

**Tuning PID Controllers Using TD3 Reinforcement Learning for Nonlinear Systems**

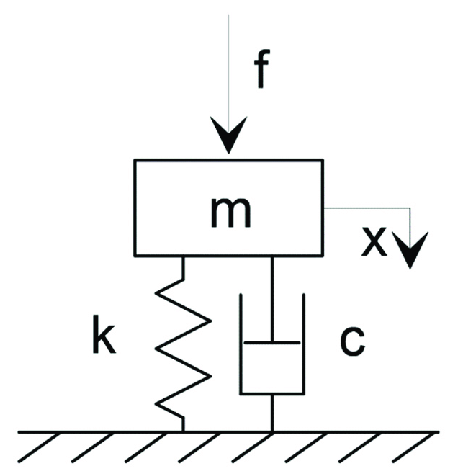
ID: 202201216

Name: Jeet Patel

#### 

## I. System Modeling (Equations and Dynamics)

The system used for this project is a **mass-spring-damper system**, which models real-world dynamic systems such as car suspensions. The dynamics are governed by the second-order differential equation:



**mx”+cx’+kx = F**

Where:

* m: Mass (kg).
* c: Damping coefficient (N·s/m).
* k: Spring constant (N/m).
* F: External force applied to the system.

In this project:

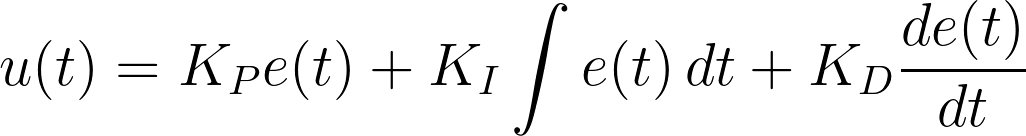
* The **state variables** are the position (x) and velocity (x’).
* The **control objective** is to stabilize the system with minimal overshoot and fast settling time by tuning the PID parameters (KP, KI, KD​).

#### 

## II. Control Strategies

**PID Control**:

* Proportional, Integral, and Derivative (PID) controllers are widely used for their simplicity and robustness.
* The controller computes the control input based on:



Where e(t) is the error between the desired setpoint and the actual position.

* However, tuning PID controllers manually for nonlinear systems is challenging, as they struggle with dynamic changes and system nonlinearities.

**Reinforcement Learning with TD3**:

* **TD3 (Twin Delayed Deep Deterministic Policy Gradient)** is an advanced RL algorithm designed to overcome limitations in earlier methods like DDPG.
* The **RL agent** learns to dynamically optimize PID parameters (KP, KI, KD​) by interacting with the system.
* Key TD3 Features:
  + **Clip Double Q-Learning**: Reduces overestimation bias by using two Q-networks.
  + **Target Policy Smoothing**: Adds noise to actions to improve stability.
  + **Delayed Policy Updates**: Ensures more stable learning by updating the policy less frequently.

In this project:

* The RL agent interacts with the mass-spring-damper system through a **Simulink model**.
* The agent receives feedback from the **Observation Block** (PID parameters) and adjusts the control force based on the **Reward Block**, which penalizes overshoot and weak control while rewarding stability.

#### 

## III. Results Comparison

* **Conventional PID Tuning**:
  + The system shows stability but with significant overshoot and slower settling times.
  + This highlights the limitations of static tuning methods in handling nonlinearities.
* **RL-Tuned PID (TD3)**:
  + The agent, trained over 500 episodes, demonstrates significant improvement:
    - Reduced overshoot.
    - Faster settling time.
    - Smooth transitions with minimal oscillations.
  + Consistent performance is observed under both constant and step input scenarios.

**Visual Examples**:

* In comparison simulations:

| **Grid** | **Conventional PID causes oscillations and slower responses** | **RL-Tuned PID achieves stability faster and avoids oscillations entirely** |
| --- | --- | --- |
| **Example** |  |  |
| **Simscape**  **Model** |  |  |

## 

## IV. References

1. **"Tuning of PID Controllers Using Reinforcement Learning for Nonlinear Systems Control” -** Gheorghe Bujgoi and Dorin Sendrescu
2. **Simscape Model From Matlab’s Community File Exchange -** [LINK](https://in.mathworks.com/matlabcentral/fileexchange/98689-modeling-and-simulation-of-spring-mass-damper-system-smd)
3. **MATLAB and Simulink documentation**
4. **Reinforcement Learning Onramp**