

A background image of an industrial robotic arm, likely a KUKA, with a black flexible cable and various mechanical components. The arm is positioned diagonally across the frame, with its end effector visible on the right side. The background is slightly blurred, showing other industrial equipment and a yellow safety barrier.

# CARTPOLE SIMULATION, CONTROL AND RENDERING

An Engineering Report and Guide

LS-CodeRocket  
December 16, 2023

## Contents

Project Objectives .....	3
System and Render Requirements .....	4
<b>CAD DESIGN</b> .....	4
<b>PRE-MADE COMPONENT SELECTION</b> .....	4
<b>SYSTEM MODELLING</b> .....	4
<b>CONTROLLER DESIGN</b> .....	5
<b>CONTROLLER TESTING</b> .....	5
<b>CONTROLLER ANALYSIS</b> .....	6
<b>RENDERING TYPES</b> .....	6
<b>ADDING REALTIME DATA</b> .....	6
Template .....	8
<b>Heading 3</b> .....	8
Template .....	9
<b>Heading 3</b> .....	9

# Project Objectives

The objective of this project is to create a virtual cartpole system that can be realistically modelled, simulated, controlled, and rendered in Blender. The goal is to make it as realistic as possible to practice all aspects of the average engineering project. In Blender, there will be at least two renders: a more realistic looking simulation for presentation and analysis, and then another render for an aesthetically pleasing (“satisfying”) look for posting online. The idea is to also make a product documentation and a report of engineering decisions for this as well.

Though most components made virtually will not reflect their real counterparts. Additionally, electrical systems will most likely not be modelled accurately because of lack of interest and approximate behaviors are close to observed phenomena. Additionally, this will be practice creating controllers of different types and how to generate a large amount of high-fidelity data for analysis.

# System and Render Requirements

## CAD DESIGN

The virtual design of the cartpole system must adhere somewhat to real life and obtainable components. The design of the cartpole system must be less than or equal to 1.5 meters in length (at minimum). All non-electronic/mechanical components must be manufacturable on a 30 cm x 30 cm 3D printer out of PLA, a purchasable bolt, nut or screw, or purchasable aluminum extrusion. However, most components must be made of PLA. The design must conform to the material; it must be structurally sound.

The materials must be specified in each CAD design. In order to practice CAD modelling, all parts must be modelled in a CAD software, and none can be obtained from an online resource. However, electrical components are very complicated and require a finer understanding of electrodynamics to 3D-model correctly (such as AC/DC motors). The intricate innards of these components are not required to be modelled to render well in Blender.

## PRE-MADE COMPONENT SELECTION

Additionally, if the chosen controller requires a specific electrical device to be purchased that is not made of any of these materials, the device does not have to be personally modelled (motor, camera, etc.), however, they must look aesthetically pleasing and fit the theme of the render (so it may be more beneficial to personally 3D-model anyway). Pre-made electrical components cannot be fantastical and must be based on science in some way for usage in one of the realistic renderings. For the more artistic render, the parts may/are not constrained by physical behaviors.

## SYSTEM MODELLING

The system must utilize mathematical models that are built on realistic measurements of the system for the more realistic render. If the system utilizes a motor, it cannot have infinite output. The system cannot have infinite friction nor no friction (no slipping or infinite slipping. These will constrain the render to make it more realistic which will serve the purposes described in the requirements. For the more artistic render, the system can utilize parameters

that aren't physically accurate or possible. In addition, all electronic components act on a clock cycle, which adds a time delay to the next input. They also have a maximum allowed control output. All simulations will be done in Python, C++, or MATLAB.

## CONTROLLER DESIGN

There will be several controllers that use a variety of methods: Proportional-Integral-Derivative (PID), Model Predictive Control (MPC), Gain Scheduling, Linear Quadratic Gaussian (LQG), Plant Inversion, Machine Learning (sensor based), Machine Learning (camera based) and Bang-Bang. This is to provide practice in using these systems and understanding how they work.

To render in Cycles, the output cannot be more than 400 frames. This is because any longer and it would take a significant amount of time. Additionally, GIF renderers usually are capped at 50 frames-per-second (FPS). 24 FPS would provide around 16.6 seconds of simulation time. The simulation could be very high fidelity and could work at a much higher frequency than the render (i.e., 50 MHz), but it must lead to these 16.6 seconds of rendering. For a more artistic render, the simulation time may not be the entirety of the 16.6 seconds as to allow for creative freedoms.

For the best render, the settling time (2%) should be less than 2 seconds. The percent overshoot really can be anything, but the settling time must settle between 0 and 2 seconds. The earlier the settling time, the better, however the control outputs cannot exceed their maximum.

The final settling value should be the upward pendulum position at the center of the horizontal axis for most renders. If a render requires an angle constrained position such as 45 degrees, there will be no horizontal constraint. Along the horizontal axis, the system cannot exceed the limits described in the requirements.

## CONTROLLER TESTING

The controllers will be expected to compensate for disturbances that are added onto the system. Some disturbance like wind (a CFD could be computed over the assembly to approximate the behavior) or dampening could be added to the system. Additionally, some sensor error can be added. Some model uncertainty can be added into the controller to see how it compensates.

Additionally impulse forces will be added as disturbances from a large force input.

## CONTROLLER ANALYSIS

After each simulation, data about the simulation must be exported for rendering as well as analysis. Performance measures will be utilized to compute how good each controller is with respect to another. Another edited render will show the comparison between these.

## RENDERING TYPES

There will be five types of renders:

1. Realistic renders under nonzero initial conditions and no disturbances to just reach the final desired state. These will be shorter than the 400-frame limit and most-likely will be rendered in Cycles (but it isn't required). This will have steady camera angles and is lit on a single-color background.
2. The second renders will be realistic renders under nonzero initial conditions to resist disturbances, impulse, and white noise. This will have steady camera angles and is lit on a single-color background. These will be rendered in Eevee (if possible then Cycles).
3. The third renders will be realistic features with a changing camera angle for a better-looking shot. These will be rendered in Eevee (if possible then Cycles).
4. The fourth type of render will be a decomposition of the CAD design. I do not expect these to be longer than 16 seconds. This will be a "product presentation"-like. This will be rendered in Cycles to make it look the best.
5. The fifth type of render will be less realistic and more artistic. This will use minimalist features and little colors on a white background. This will also use a more fantastical design that is more minimalist. This will also be rendered in Cycles in order to make the render nicer to post online. The controller used will be the one with the best performance.

## ADDING REALTIME DATA

There will be several controllers that use a variety of methods: Proportional-Integral-Derivative (PID), Model Predictive Control (MPC), Gain Scheduling, Machine Learning (sensor based), Machine Learning (camera based) and Bang-Bang.



# Template

## Heading 3

To get started right away, just tap any placeholder text (such as this) and start typing to replace it with your own.

Want to insert a picture from your files or add a shape, text box, or table? You got it! On the Insert tab of the ribbon, just tap the option you need.

Find even more easy-to-use tools on the Insert tab, such as to add a hyperlink or insert a comment.

## HEADING 4

To get started right away, just tap any placeholder text (such as this) and start typing to replace it with your own.

Want to insert a picture from your files or add a shape, text box, or table? You got it! On the Insert tab of the ribbon, just tap the option you need.

Find even more easy-to-use tools on the Insert tab, such as to add a hyperlink or insert a comment.



# Template

## Heading 3

To get started right away, just tap any placeholder text (such as this) and start typing to replace it with your own.

Want to insert a picture from your files or add a shape, text box, or table? You got it! On the Insert tab of the ribbon, just tap the option you need.

Find even more easy-to-use tools on the Insert tab, such as to add a hyperlink or insert a comment.

## HEADING 4

To get started right away, just tap any placeholder text (such as this) and start typing to replace it with your own.

Want to insert a picture from your files or add a shape, text box, or table? You got it! On the Insert tab of the ribbon, just tap the option you need.

Find even more easy-to-use tools on the Insert tab, such as to add a hyperlink or insert a comment.