

# UR3e OCE Stabilization System

Ben Anderson

October 24, 2022

## Contents

<b>1 Planning (Delete later)</b>	<b>1</b>
<b>2 Introduction</b>	<b>1</b>
<b>3 Stanley and the URScript API</b>	<b>3</b>
3.1 Teach Pendant . . . . .	3
3.2 Client interfaces . . . . .	3
3.3 URScript API . . . . .	3
<b>4 Python + the control loop</b>	<b>3</b>
4.1 Startup guide . . . . .	3
4.2 TransducerHoming.py . . . . .	3

## 1 Planning (Delete later)

Introduction section will summarize the body of the text,

I need a section detailing the interfaces of the UR3e robot and why I need to use the workaround that I have

I need a section detailing how my code works to optimize the strength of the input signal

I need a section detailing how all of these bits of code interact with each other (will be the hard bit)

Creative decision: I could either lead with the section detailing how all the components interact with each other, or I could lead with sections detailing individual components and close with a section summarizing their overlap. The tradeoff is the summary section will be missing a lot of context, but this could potentially be provided by linking forwards repeatedly. Mayhaps this is the move. I just need to learn how to link to a place further along in a document (section 2).

Okay that looks promising to me. Lets go with that.

## 2 Introduction

The focus of my work for the better part of the last year has been studying how to use a UR3e Cobot for the purposes of focusing and stabilizing a sensor at its focal length away from a target surface.

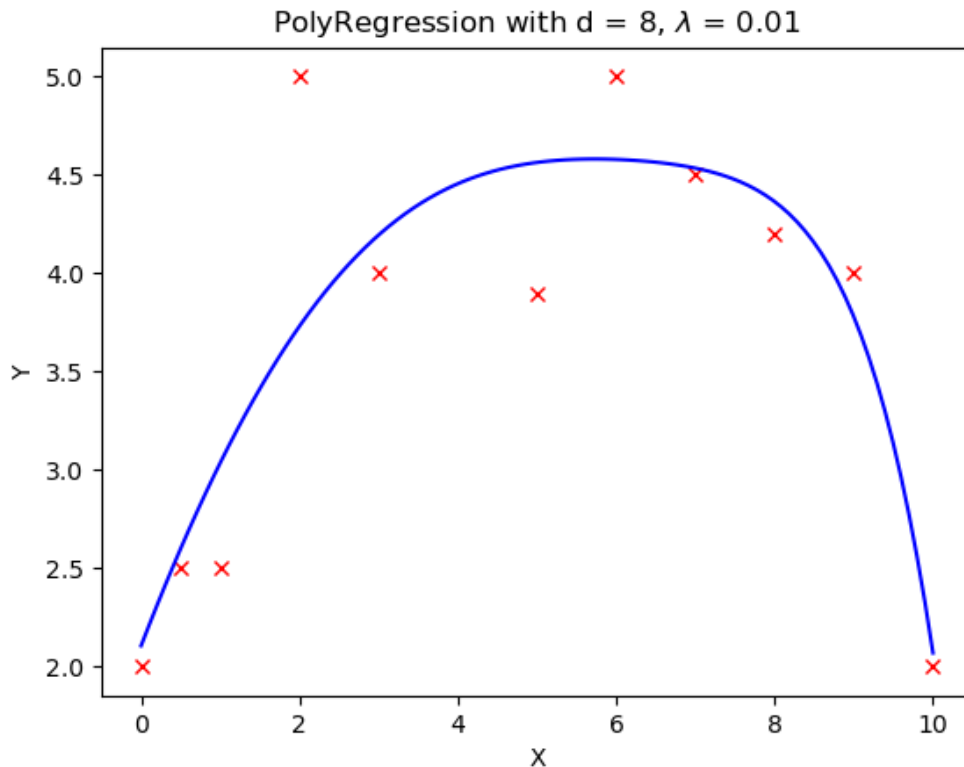
This has involved a thorough examination of the API and programming tools that accompany the Universal Robots ecosystem, as well as creating systems for synchronizing the robotic control system with signal processing and sensor systems.

In doing so a number of systems are being used in ways for which they were not designed. The consequence of this is that my codebase resembles a kludgy pile of half-fixes and temporary-solutions that, technically, have achieved the objective, at the expense of stability and ease of use. In recent months more efforts have been made to improve the readability of the codebase and allow for easier changes later on, but the work is ongoing and the project cannot wait for me to complete this work.

The hope is with this documentation, others in the OCE project can gain enough of an understanding of how the system works to use it in my absence as well as modify or extend the platform I've built.

Due to the kludgy nature of the platform, it may not be suitable to read this guide linearly. The introduction section should hopefully give you an overview of how the components of this section communicate and interface with each other, and a more thorough description of the workings of each module are in the following chapters. If you are planning to operate the robot I would recommend reviewing the section on operating the robot through the control pendant (subsection 3.1) and the section on running the code (subsection 4.1).

$$\frac{1}{n} \sum_{i=1}^n (\mathbb{E}[\hat{f}_m(x_i)] - f(x_i))^2 = \frac{1}{n} \sum_{j=1}^{n/m} \sum_{i=(j-1)m+1}^{jm} (\bar{f}^{(j)} - f(x_i))^2$$



## 3 Stanley and the URScript API

### 3.1 Teach Pendant

### 3.2 Client interfaces

### 3.3 URScript API

## 4 Python + the control loop

### 4.1 Startup guide

### 4.2 TransducerHoming.py