Semester project on signals in robot systems

BEng in Robot Systems

3. Semester

Authors (Group 3)

Name	Username	Birthday
Noah V. Vryens	novry24	01/04/2002
Rasmus K. Nielsen	rasni19	25/12/1997
Emma B. Rasmussen	erasm24	15/05/2001
Jannick H. Irvold	jairv24	23/10/2002
Eymundur Ó. Pálsson	eypal24	31/08/2002

Supervisor

Thorbjørn M. Iversen

thmi@mmmi.sdu.dk

December 1, 2025

Abstract Placeholder

This report documents the design, implementation, and evaluation of an autonomous robotic system capable of identifying a target and throwing a projectile to hit it using a Universal Robots UR5 manipulator. The system addresses the challenges of dynamic object manipulation by integrating robust machine vision, ballistic trajectory planning, and real-time kinematic control.

The vision subsystem, developed in C++ using the OpenCV and Pylon libraries, employs a Basler camera to capture the workspace. It utilizes intrinsic camera calibration and homography transformation matrices to accurately map 2D pixel coordinates to the robot's 3D Cartesian frame. Target detection is achieved through image rectification and the Hough Circle Transform, allowing for the precise localization of circular targets on a planar surface.

For trajectory generation, a physics-based ballistic model calculates the required release velocity and launch angles (yaw and pitch) to reach the target coordinates. These parameters are processed by a MATLAB computational backend, which leverages the Robotics System Toolbox to model the UR5 kinematics. The system solves the inverse kinematics for the release configuration using a Generalized Inverse Kinematics (GIK) solver with strict joint and position constraints. To ensure the end-effector achieves the exact velocity vector required for the throw, a weighted Jacobian-based velocity controller generates the motion profile, incorporating a computed lead-up phase to account for joint acceleration limits and a follow-through phase to maintain orientation.

Robot control is executed via the ur_rtde interface, which establishes a real-time control loop (125 Hz) with the UR5 controller. The system utilizes speedJ commands to execute the generated joint-space velocities, ensuring smooth and dynamic motion. Synchronization with a custom-actuated gripper is handled through a TCP/IP socket connection, triggering the release at the precise calculated moment. This report details the software architecture, mathematical modelling of the kinematics and dynamics, and the experimental results regarding the system's throwing accuracy and repeatability.

Contents

1 Introduction **1**
- 1.1 Project Goals . . . 1
- 1.2 Problem Definition . . . 1
- 1.3 Constraints & Limitations . . . 1
- 1.4 System Overview . . . 1

2 Machine Vision **2**
- 2.1 Calibration . . . 2
- 2.2 Homography . . . 2
- 2.3 Finding circular objects . . . 2
- 2.4 Implementation with OpenCV . . . 2

3 Trajectory Planning **3**
- 3.1 Physics . . . 3
- 3.2 Kinematics . . . 3
  - 3.2.1 MATLAB Communication . . . 3

4 Robot Control **4**
- 4.1 Gripper . . . 4
- 4.2 UR5 . . . 4

5 Data Collection **5**

6 Evaluation **6**

7 Discussion **7**

8 Conclusion **8**

9 References **9**

A Distribution of tasks **10**

B Code **11**

1 Introduction

1.1 Project Goals

1.2 Problem Definition

1.3 Constraints & Limitations

1.4 System Overview

2 Machine Vision

2.1 Calibration

2.2 Homography

2.3 Finding circular objects

2.4 Implementation with OpenCV

3 Trajectory Planning

3.1 Physics

3.2 Kinematics

3.2.1 MATLAB Communication

4 Robot Control

4.1 Gripper

4.2 UR5

5 Data Collection

6 Evaluation

7 Discussion

8 Conclusion

9 References

A Distribution of tasks

B Code