# Interactive demo on the indoor localization, control and navigation of drones

Rian Beck
r0585849

Mathias Bos
r0582602

December 18, 2018


Faculty of Engineering Science
**Department of Mechanical Engineering**
Celestijnenlaan 300 box 2420 B-3001 Heverlee

# Outline

# Outline section 1

# Project goal

- Demo combining
  - Localization
  - Control
  - Autonomous Navigation


- Visually attractive
- Spectators with(out) background knowledge
- Benchmark: Compare to human

# Outline section 2

# Hardware selection - drone & localization system

- **Drone**: *Parrot Bebop 2*
  - Available, cheap
  - Small
  - Easy interface (ROS, wifi)

- **Localization system**: *HTC Vive*
  - Excellent accuracy (+-1cm)
  - Superior update rate (up to 100Hz)
  - 6D pose estimates
  - ROS compatible
  - Reasonable price

# Tracker Mounting - Schematic representation

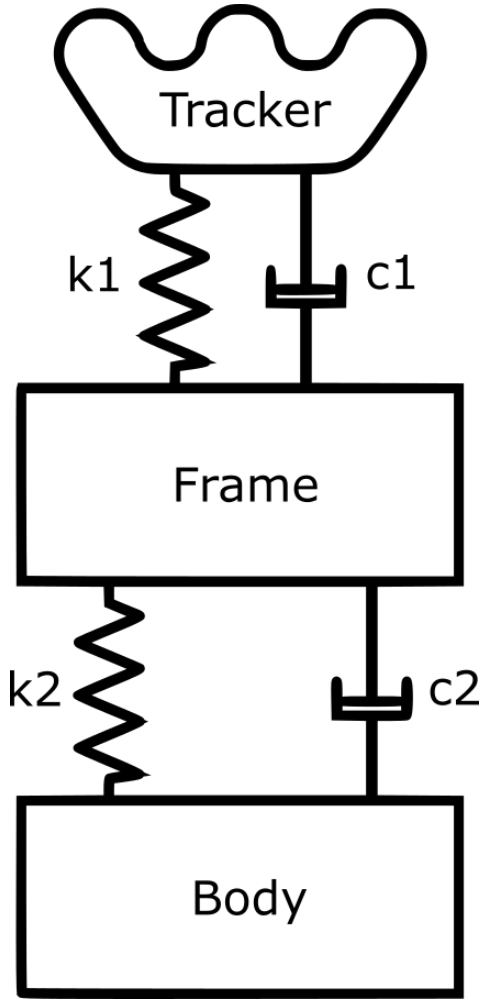



Figure 1: Mass-spring-damper representation of drone setup.

# Outline section 3

# Implementation overview

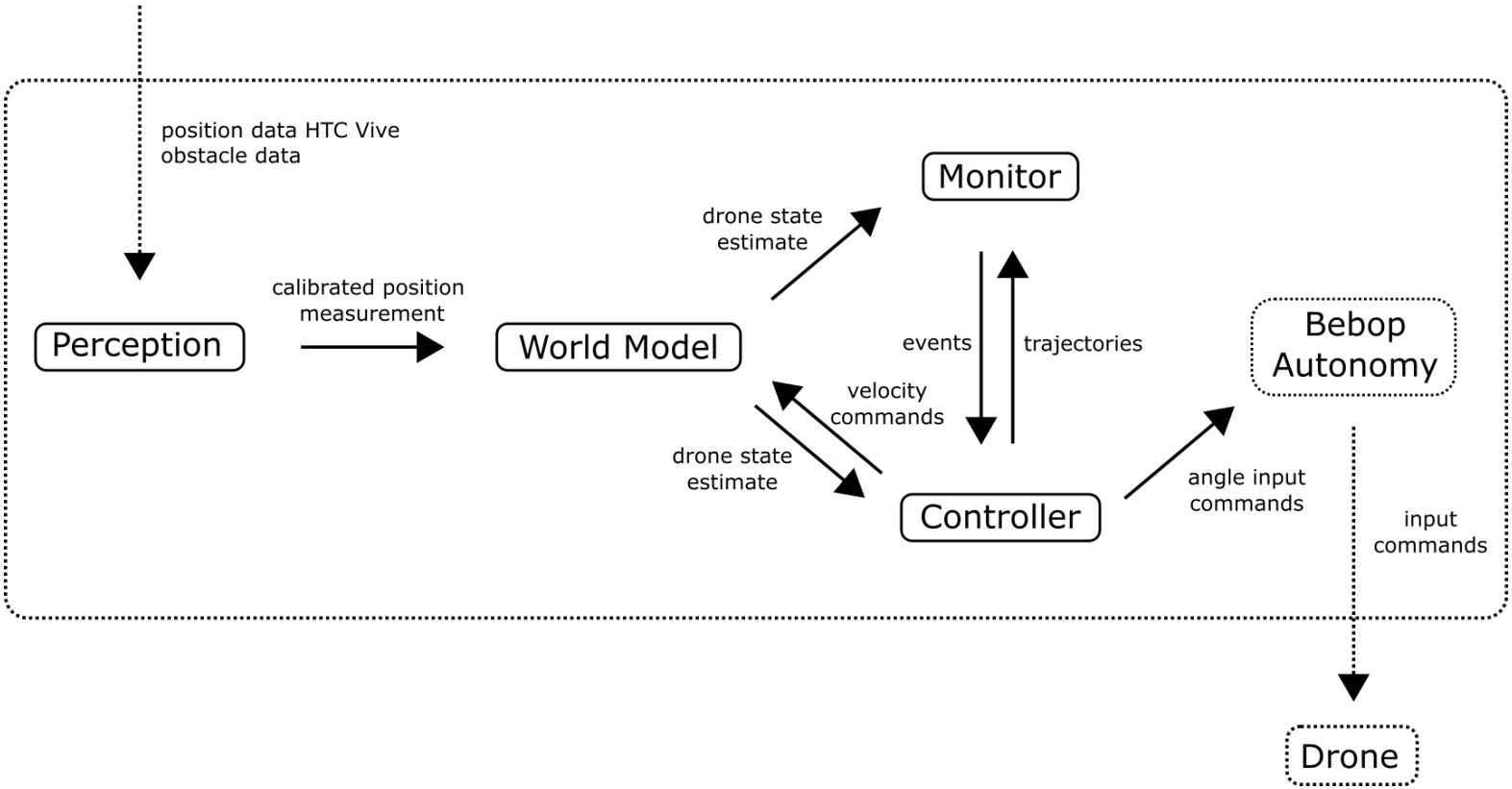



Figure 2: The nodes, inputs and outputs that make up the code.

# Implementation - Perception

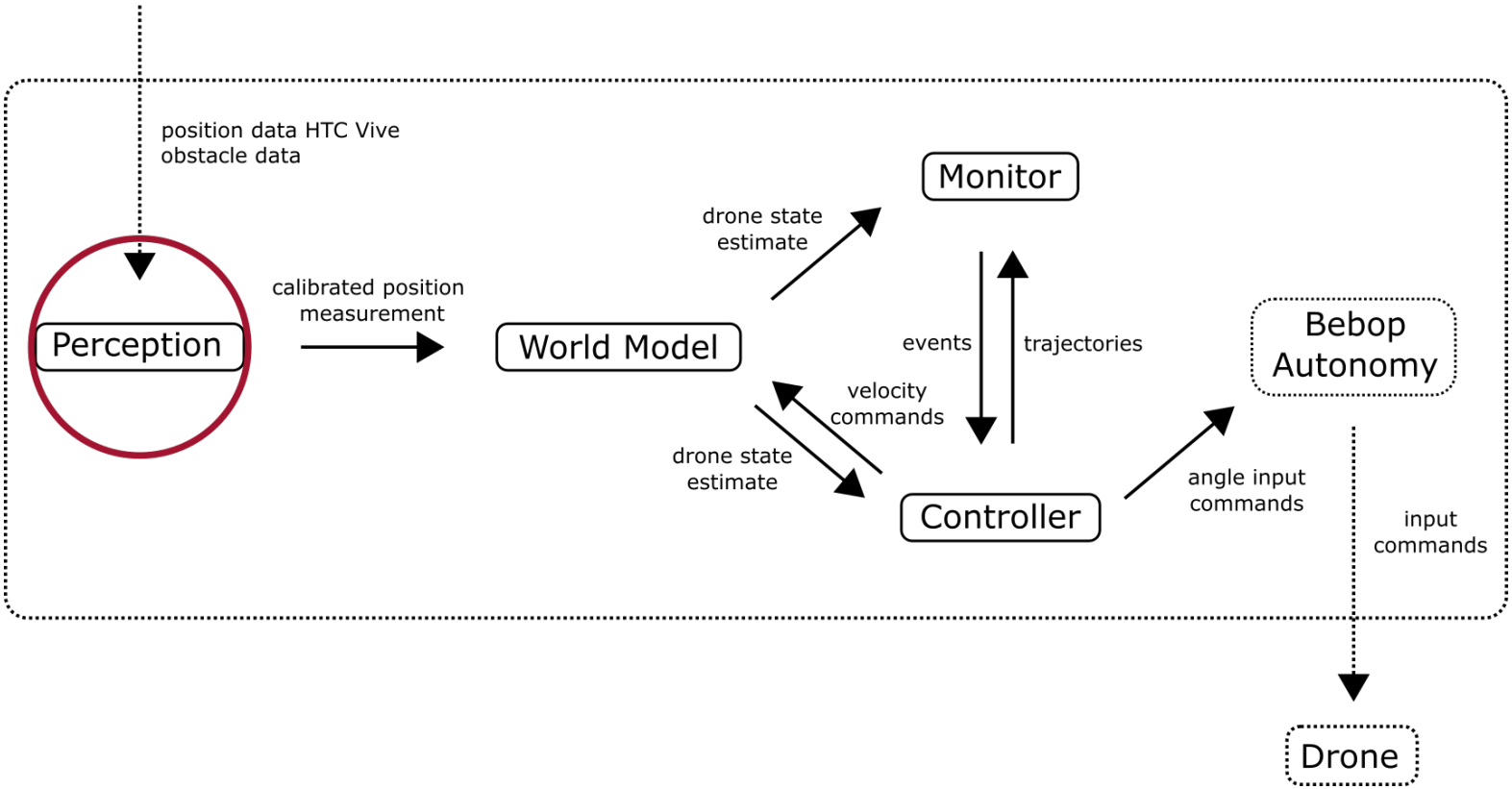



Figure 3: The nodes, inputs and outputs that make up the code.

# Vive Localization - Transformation tree

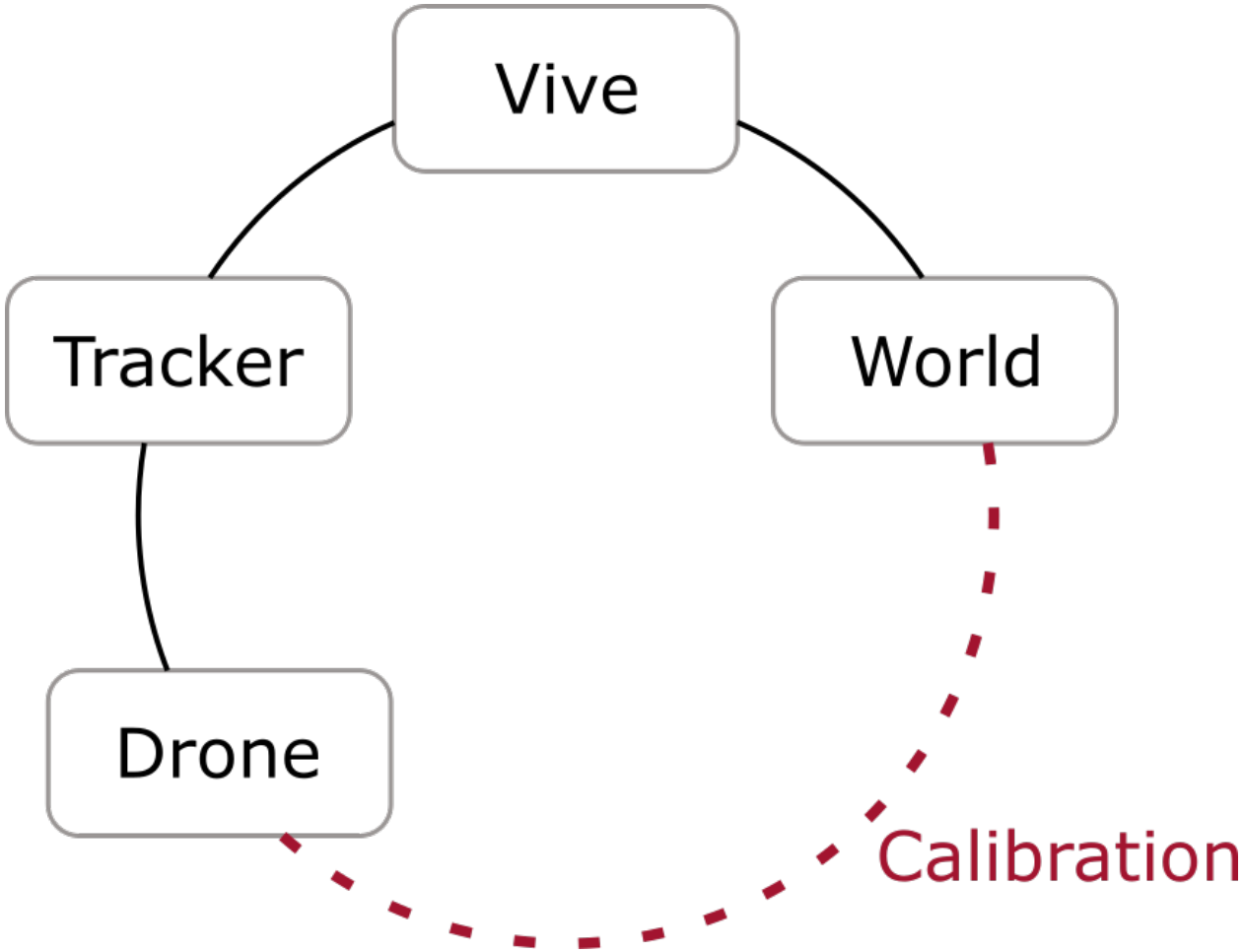



Figure 4: Transformation tree: Coordinate reference frames.

# Implementation - World model

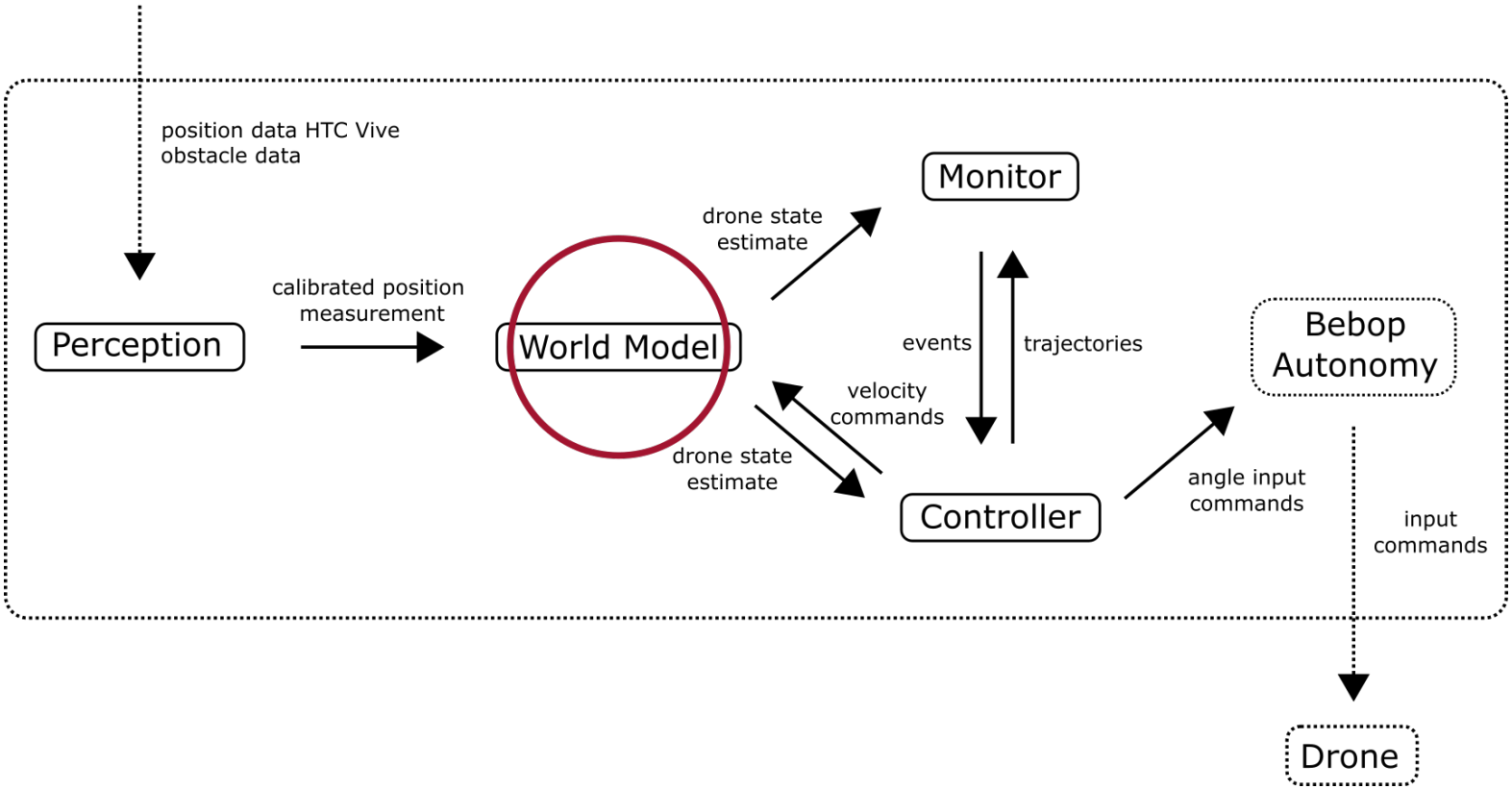
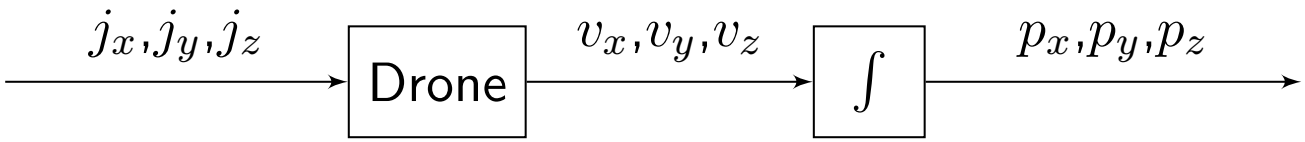



Figure 5: The nodes, inputs and outputs that make up the code.

# Model Identification

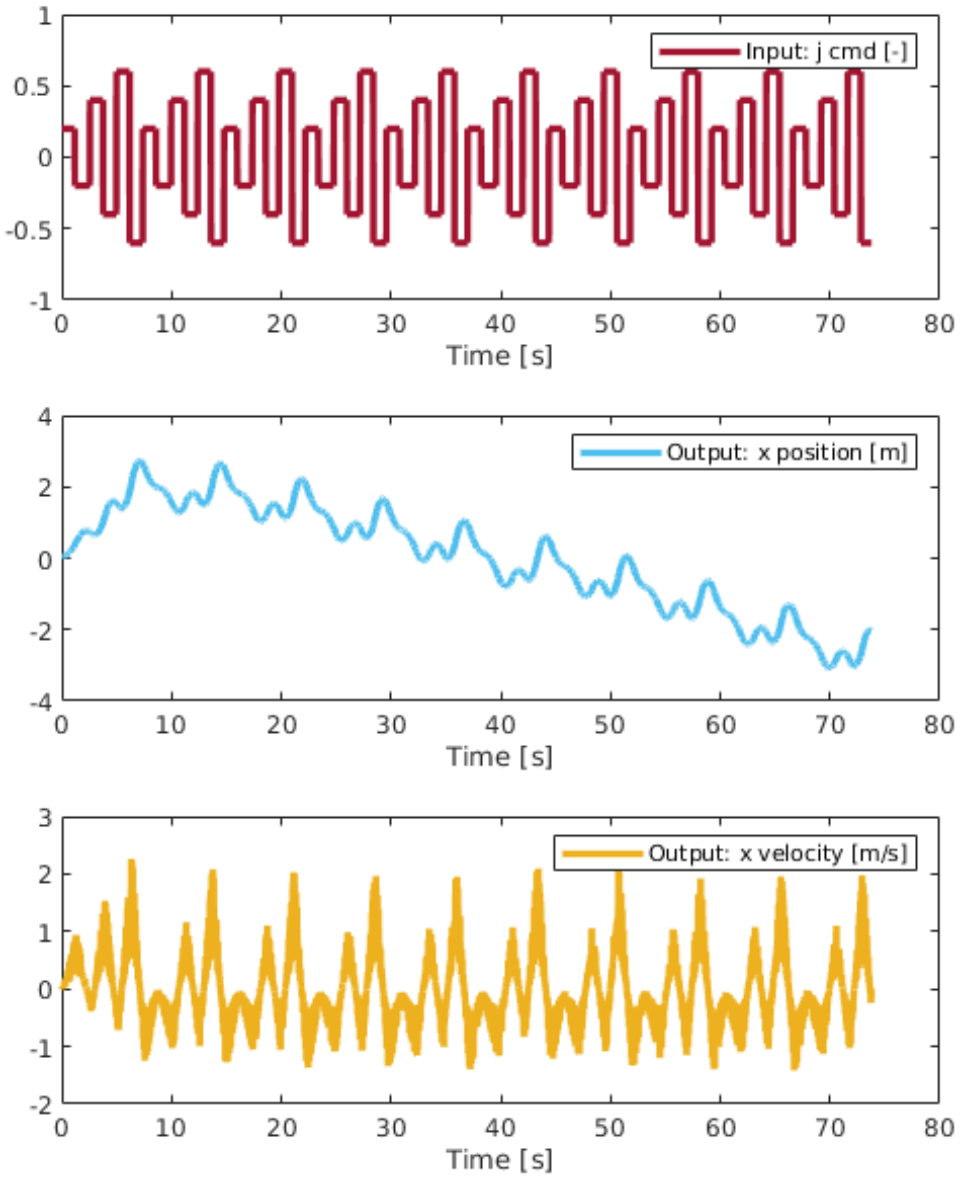



$$H_{XJ}(s) = \frac{X_{x,y}(s)}{J_{x,y}(s)} = \frac{b_0}{s^3 + a_2 s^2 + a_1 s + a_0}$$

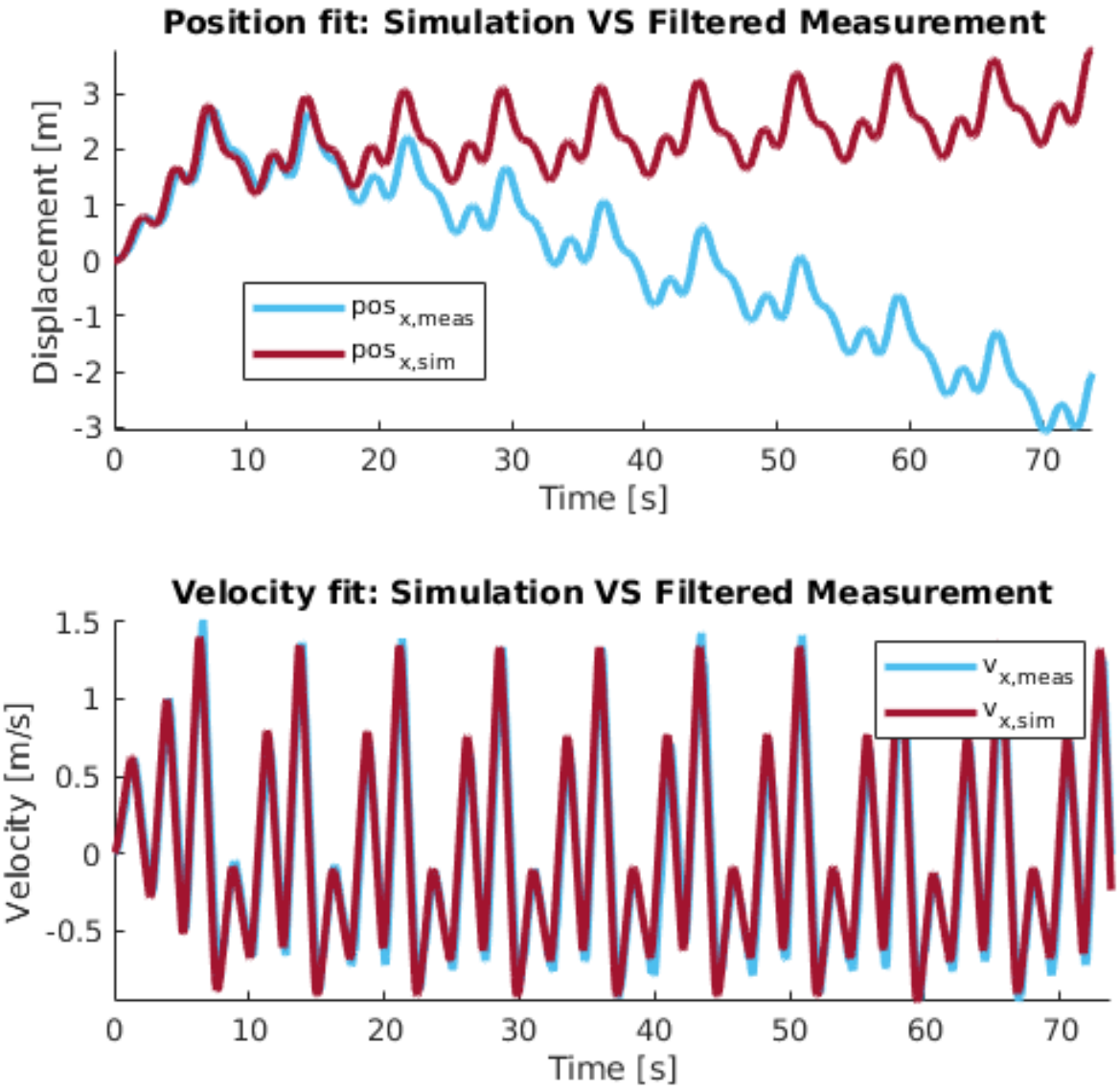
$$H_{VJ}(s) = \frac{V_{x,y}(s)}{J_{x,y}(s)} = \frac{b_0}{s^2 + a_1 s + a_0}$$

$\rightarrow$ Least-squares discrete time domain parameter identification

# Identification experiment - X, Y



# Identification results - X, Y



# Asynchronous Kalman Filter - Introduction

- State estimator needed
- Combine model & measurement
- Combine measurements
- Controller vs measurement rate

# General working principle

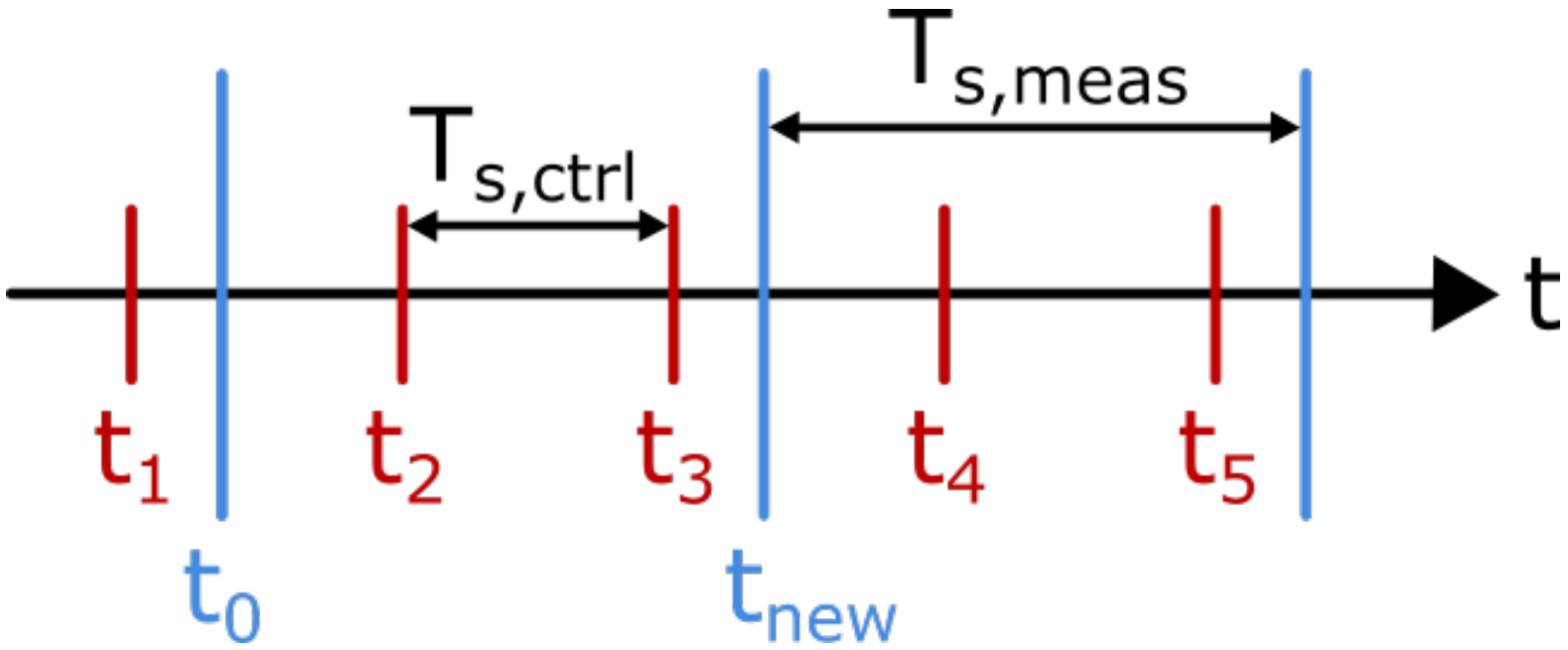



Figure 6: General time schedule of input and measurement time stamps.

# Implementation - Controller

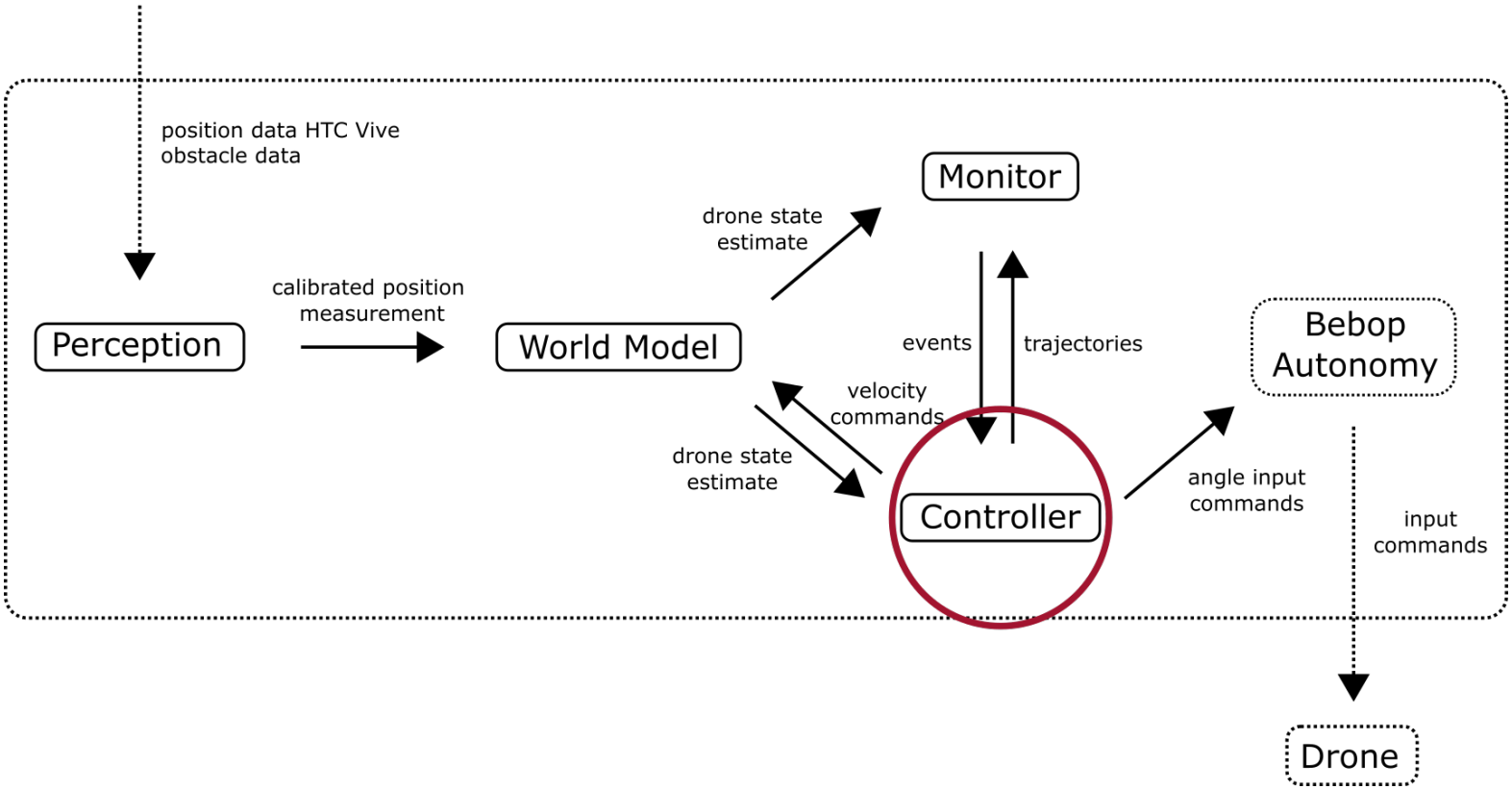



Figure 7: The nodes, inputs and outputs that make up the code.

# OMG-tools - Controller block diagram

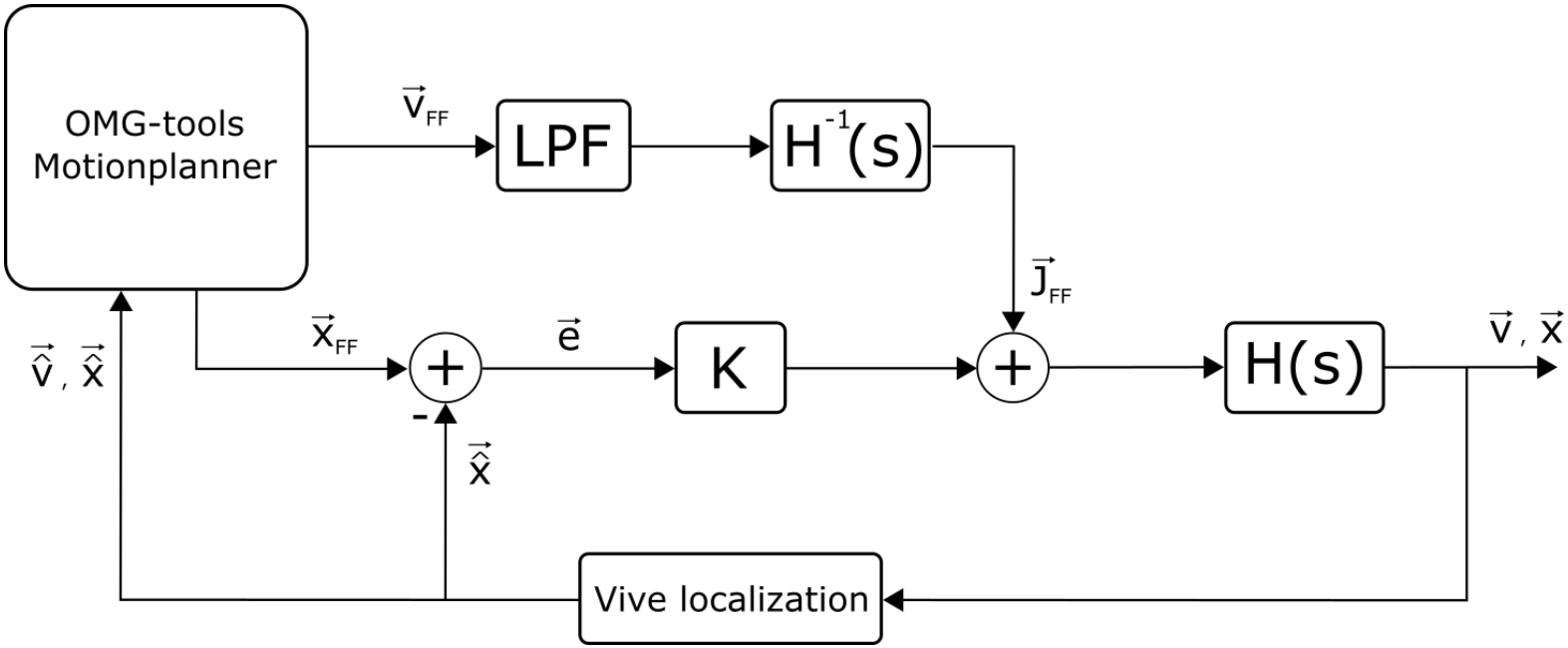



Figure 8: Control diagram combining feed-forward and feedback.

# Outline section 4

# Conclusion: Halftime results

- Vive: Read out position data, transform to a chosen (calibrated) reference frame
- Tracker mounts
- Models identified
- Kalman filter designed
- Control + OMG-tools
- **Drone can fly to goal in 2D + indicate goal reached**

# 2D Demo: Real and Virtual Twin

# Conclusion: Future work

- Improve tracking robustness
- Evade obstacles
- Extend to 3D flying
- Implement demo

# Thank you for your attention!