Position Sensing and Imitation
Final Presentation

Konstantin Koslowski, Mathis Schmieder, Moksha Birk

TU Berlin
Department of Telecommunication Systems
Telecommunication Networks Group

July 29th, 2015

Telecommunication Networks Group

Introduction

Reminder: Goal Statement

- **Goal:** Mimic position and motion of a plate

- **Sensing:** 3D MEMS attitude sensor embedded in a plate
- **Communicating:** Implement industrial bus
- **Actuating:** Rotate a plate using motors

Reminder: Functional Overview

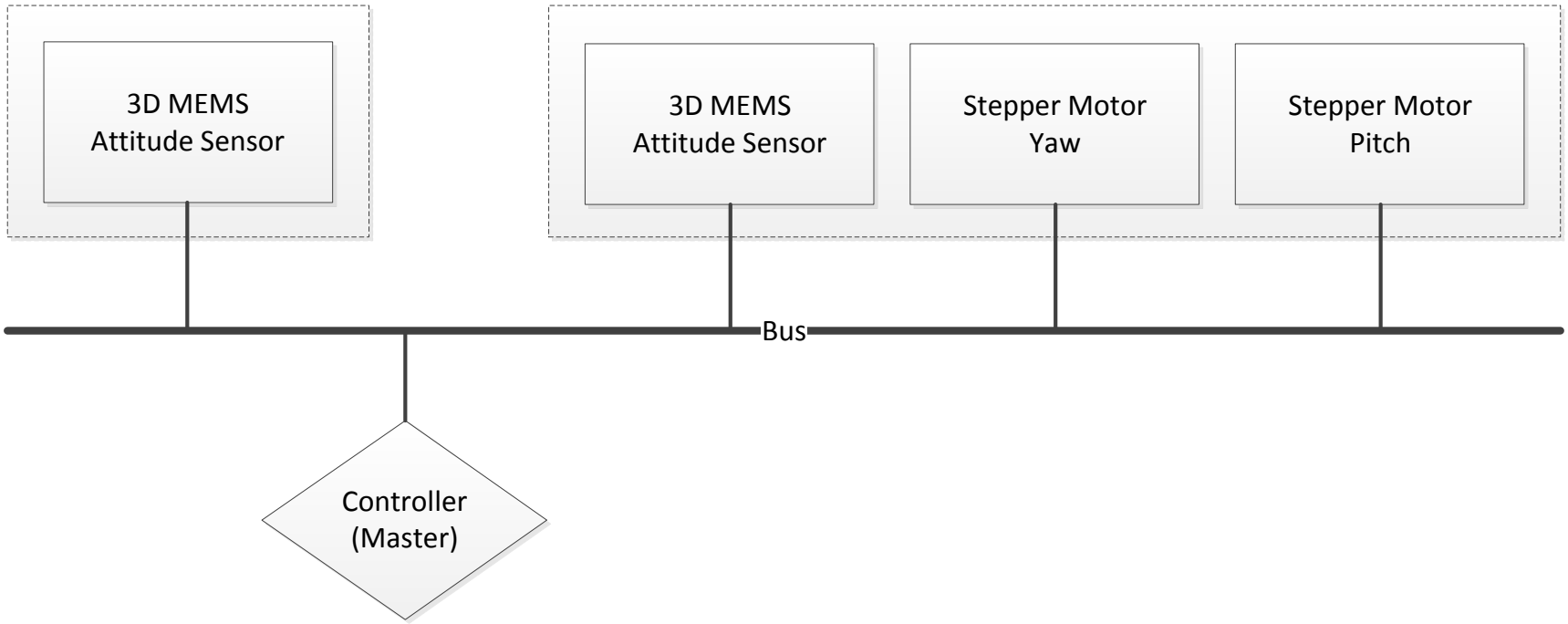

Figure : Diagram of the Functional Specification

Telecommunication
Networks Group

Reminder: Major Milestones

- **Sensing:** Read and process MEMS data
- **Actuation:** Control stepper motors
- **Mechanics:** Construct movable plate
- **Communication:** Implement industrial bus
- **Controller:** Bus master, main computational unit

Overview

Milestone: Sensing

Read and process MEMS data

Status:

- Reading data via I2C works
- Computing plate position from data works
- Additional filtering might be required

Milestone: Actuation

Control stepper motors

Status:

- Communication with stepper drivers via SPI works
- Control of stepper motors works
- Additional work on control daemon necessary

Milestone: Mechanics

Construct movable plate

Status:

- First version of plate construction printed
- Works for now
- Design on second, refined version in progress

Milestone: Communication

Implement industrial bus

Status:

- A lot of research was done
- EtherCAT selected as most interesting
- We couldn't get EtherCAT working
- CAN used as fallback

Milestone: Controller

Bus master, main computational unit

Status:

- Modular design to fit CAN and EtherCAT
- High-level controller class
  - Receives periodic sensor input events
  - Computes angle corrections for all drives
- CAN wrapped into classes to provide the events and send corrections
- Component interaction via Sockets
- Built on a BeagleBone Black

System Specifications

Timing

Timing goal: Move plate to desired position within 1 second

Fixed timings:

- Sensors
  - Sample every 10 ms
  - Report mean value every 100 ms
- Actuation takes up to 500 ms

Delay constraint: 500 ms to compute & communicate

Bus

Bus specification

- EtherCAT could not be implemented
- Using fallback option CAN
- ...

CAN

Bus Design

- Required cycle time: 100 ms
- All nodes are BeagleBone Blacks
- CAN controller: SN65HVD230
- Sensor values are periodically fed to the Controller from a buffer
- Possibly with a communication scheme like TTCAN

Messsage ID Descriptions

Node Name	ID / Priority	Master / Slave
Controller	1	Master
Source Sensor	2	Slave
Target Sensor	3	Slave
Stepper Driver	4	Slave

Table : Nodes in the network

Message Description

Bus Option 2

Description	Data Request	Length
Sensor Position	Allowed	6 Bytes
Motor Status	Allowed	4 Bytes
Rotation Command	Not Allowed	3 Bytes
Reset Command	Not Allowed	3 Bytes

Table : Possible messages in the network

Message Sequence Charts

- ?

- ?

Discussion

Thanks for your attention!

Questions? Ideas? Suggestions?