NuBot 2014 new Platform

RoboCup MSL Workshop

2014.11.10, Eindhoven

國防科學技術大學
National University of Defense Technology

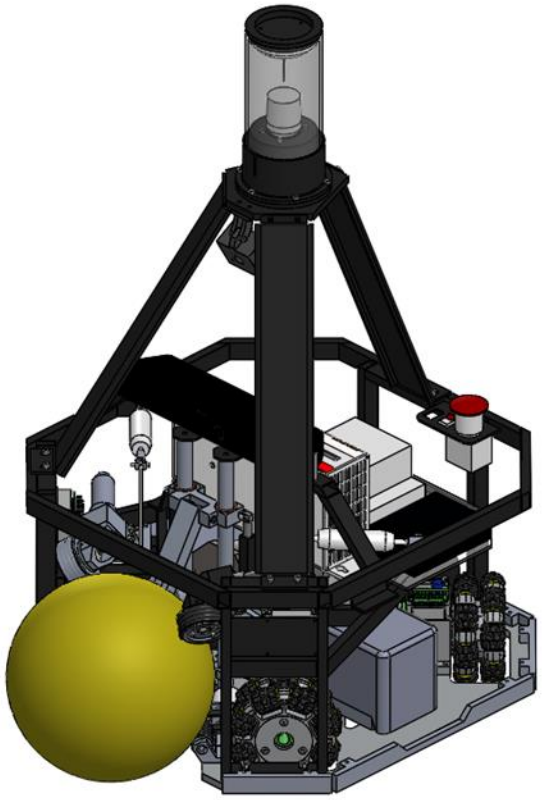
Outline

- New mechanical design
- The vision system
- The electrical system
- ROS-based softwares

Outline

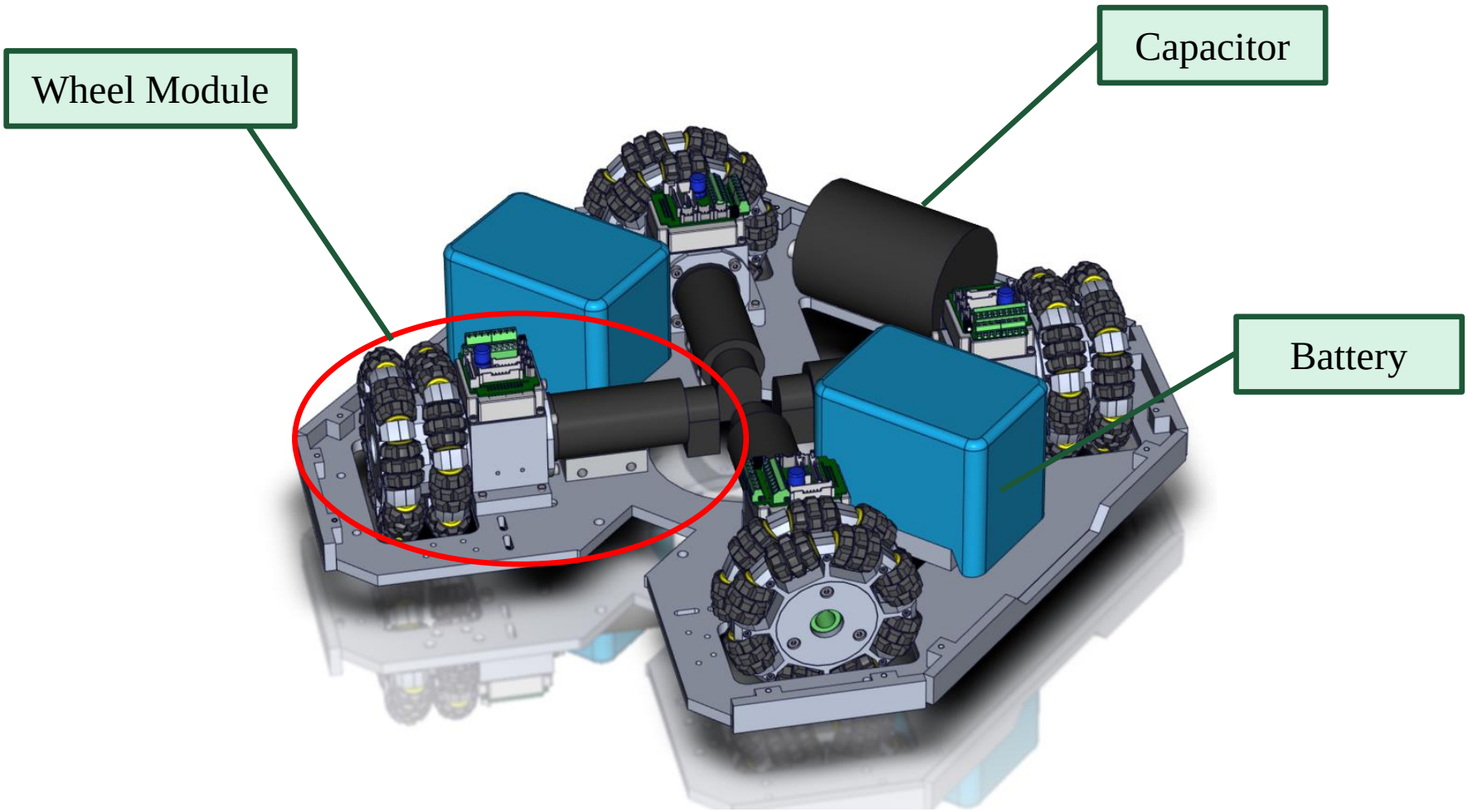
- New mechanical design
- The vision system
- The electrical system
- ROS-based softwares

New mechanical design

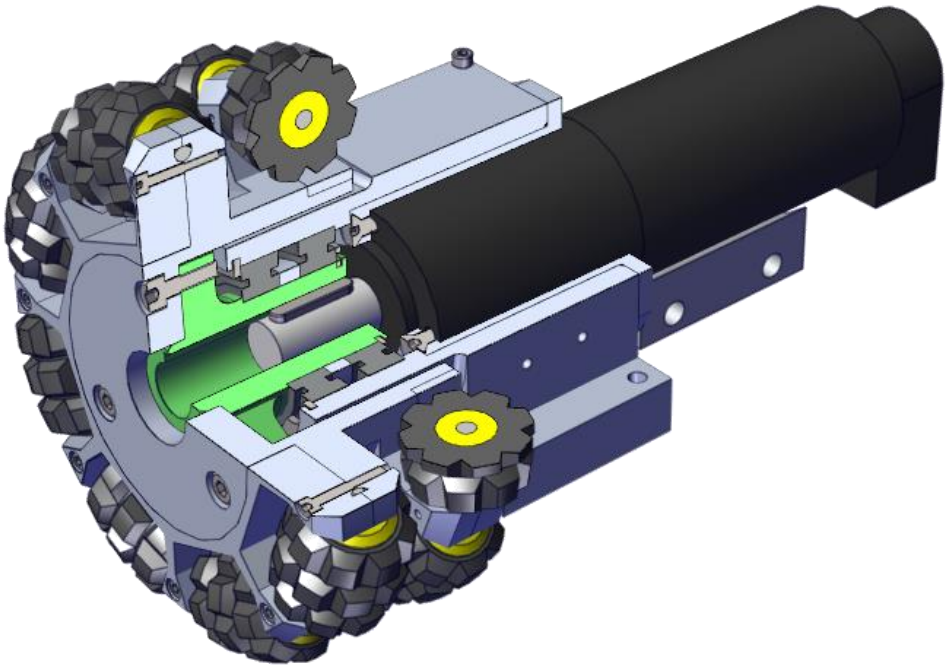


- Goals:
  - Light weight;
  - Robust to collisions;
  - Easy to maintain;
  - Maximum capabilities;

Base plate



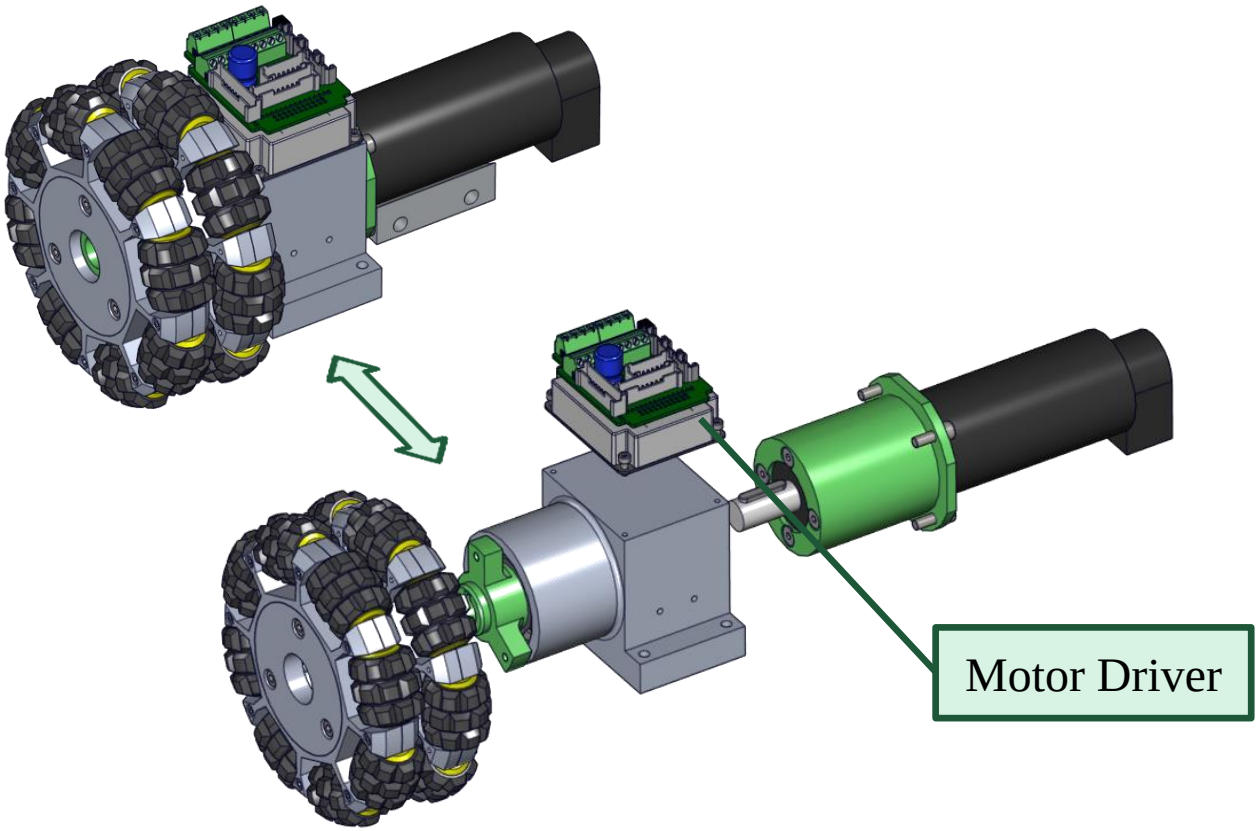
Wheel Module



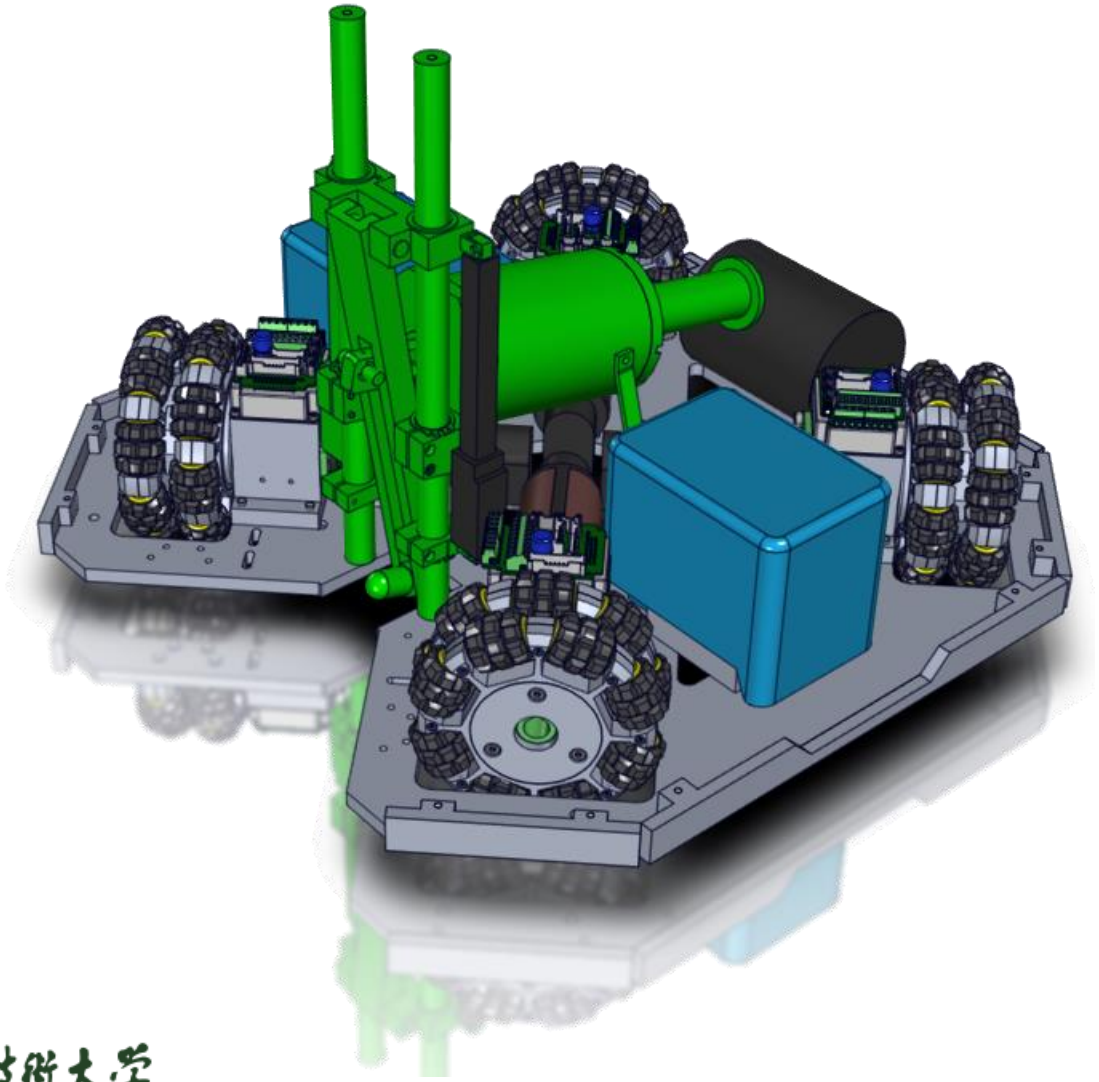
- Features:
  - Direct transmission;
  - Compact;
  - Easy maintain.

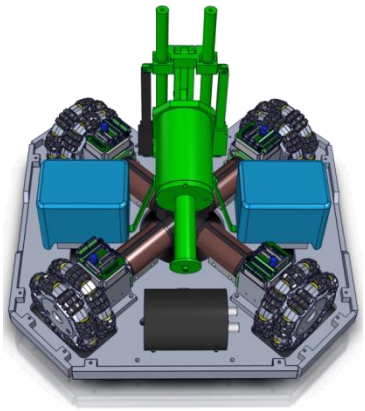
Wheel Module

Easy to assembly & disassembly

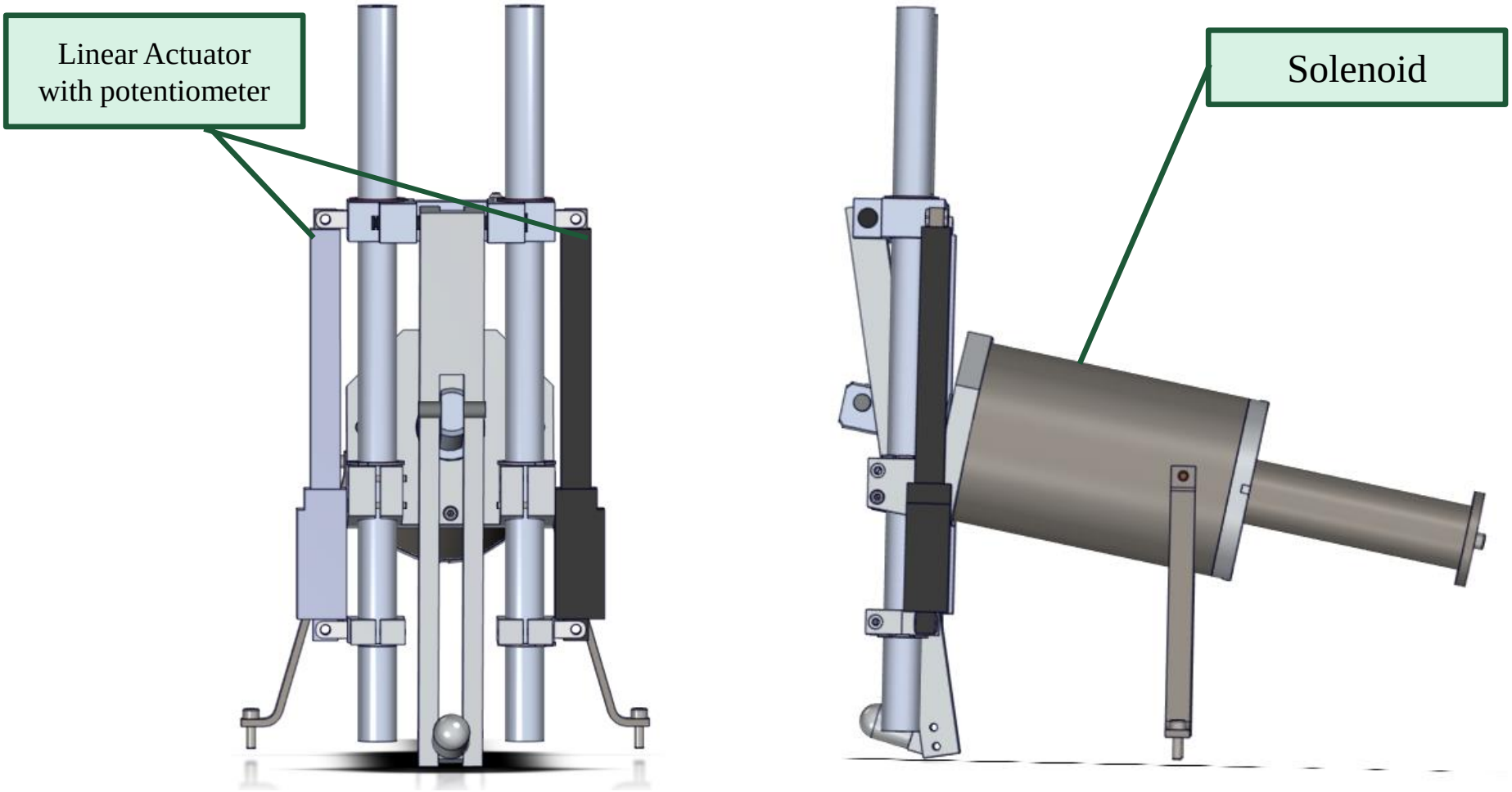


Kicking Device

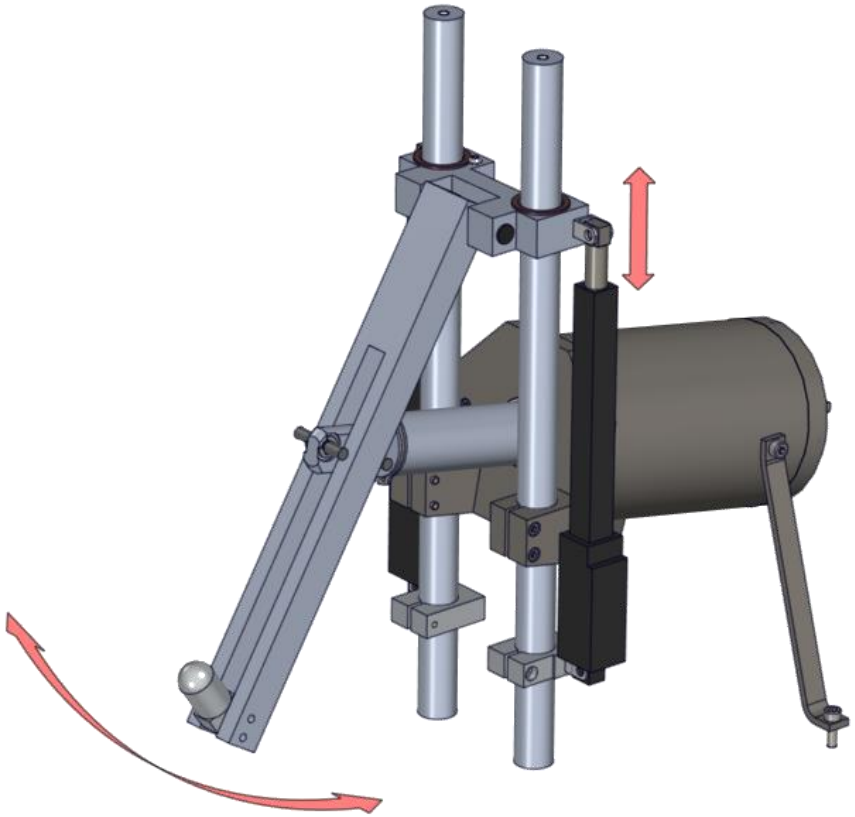




Kicking Device



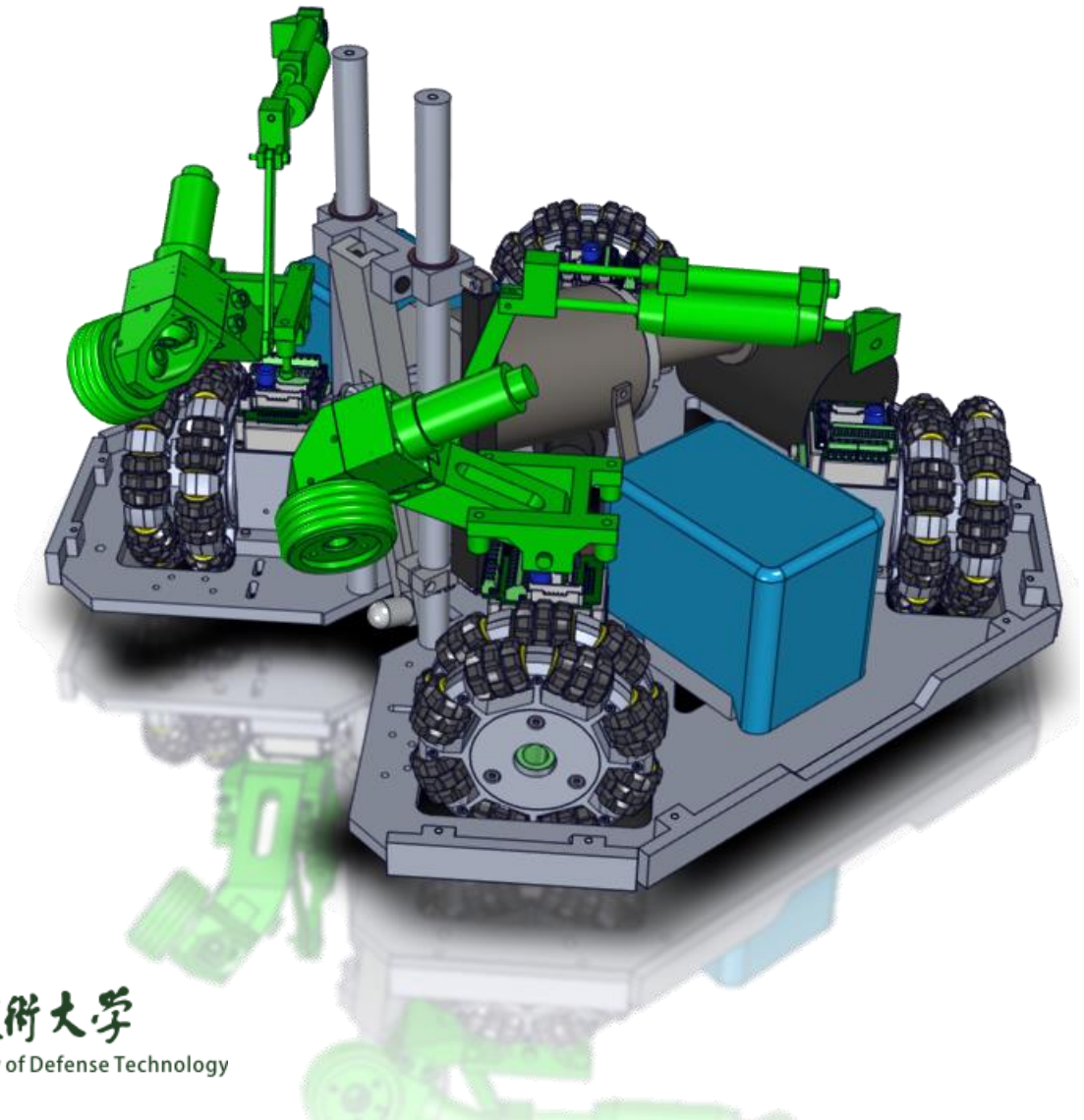
Shooting Device

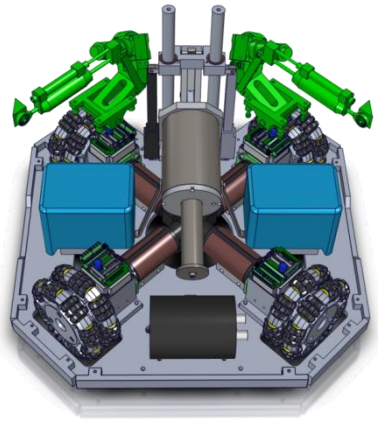


Features:

- Simplicity;
- Continuous height adjustment;
- Various shooting configurations.

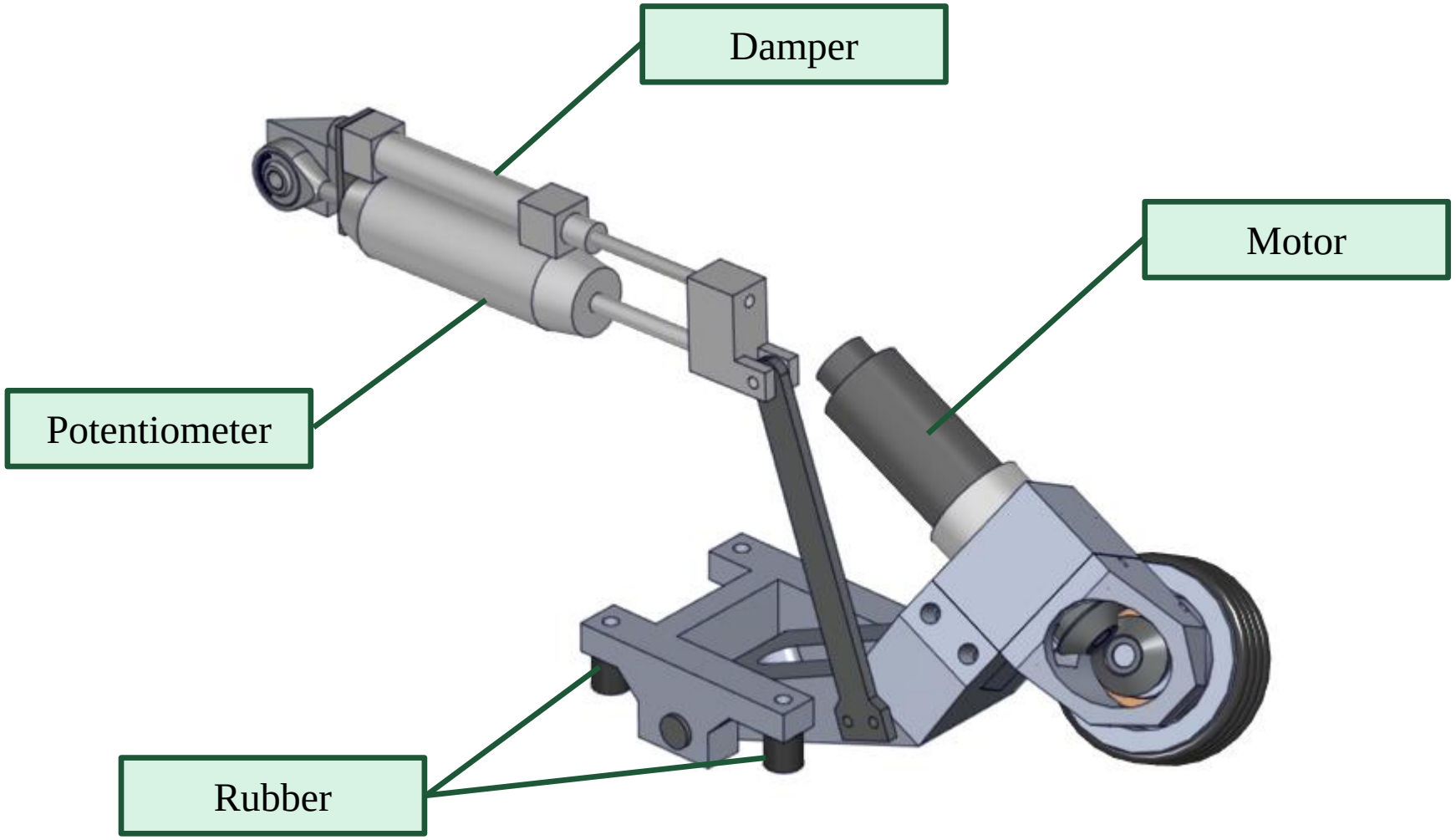
Ball Handling Device



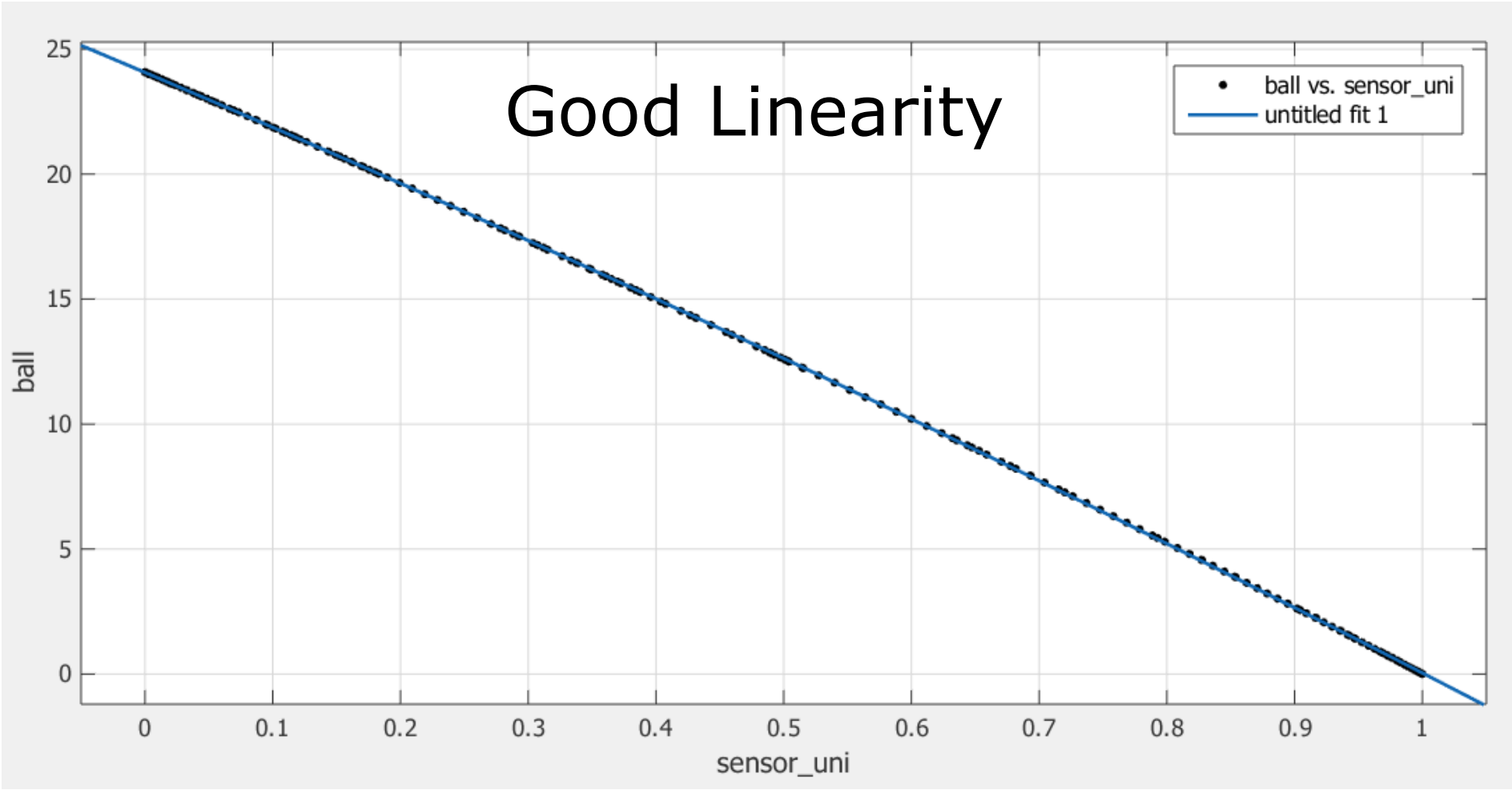


國防科學技術大學
National University of Defense Technology

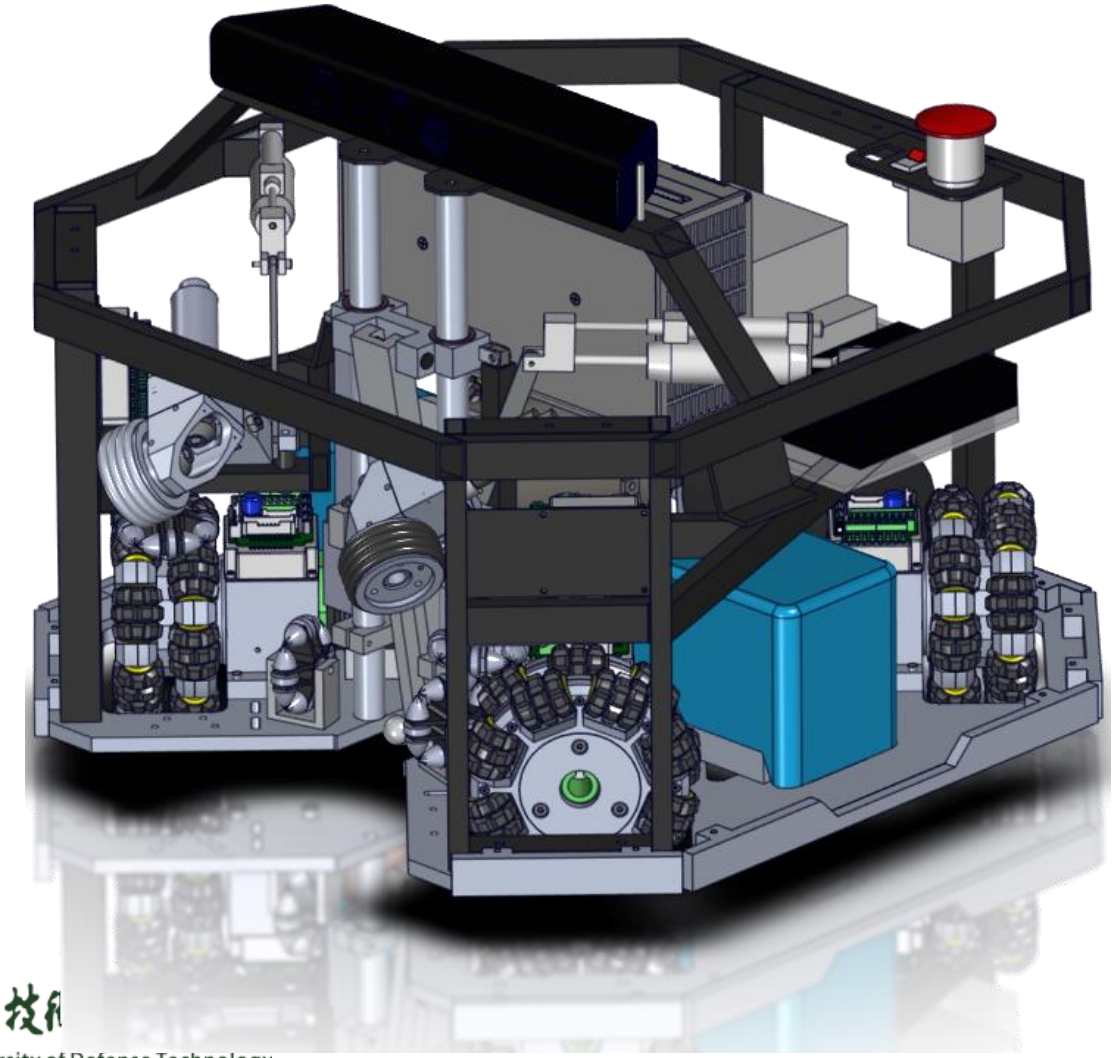
Ball Handling Device

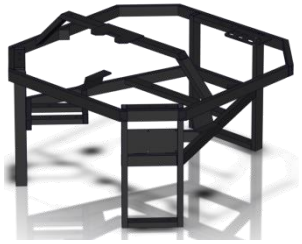


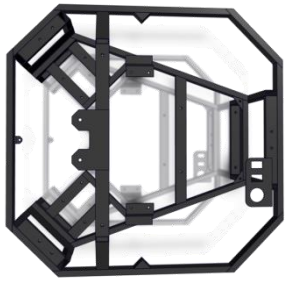
Ball Distance & Sensor Readings



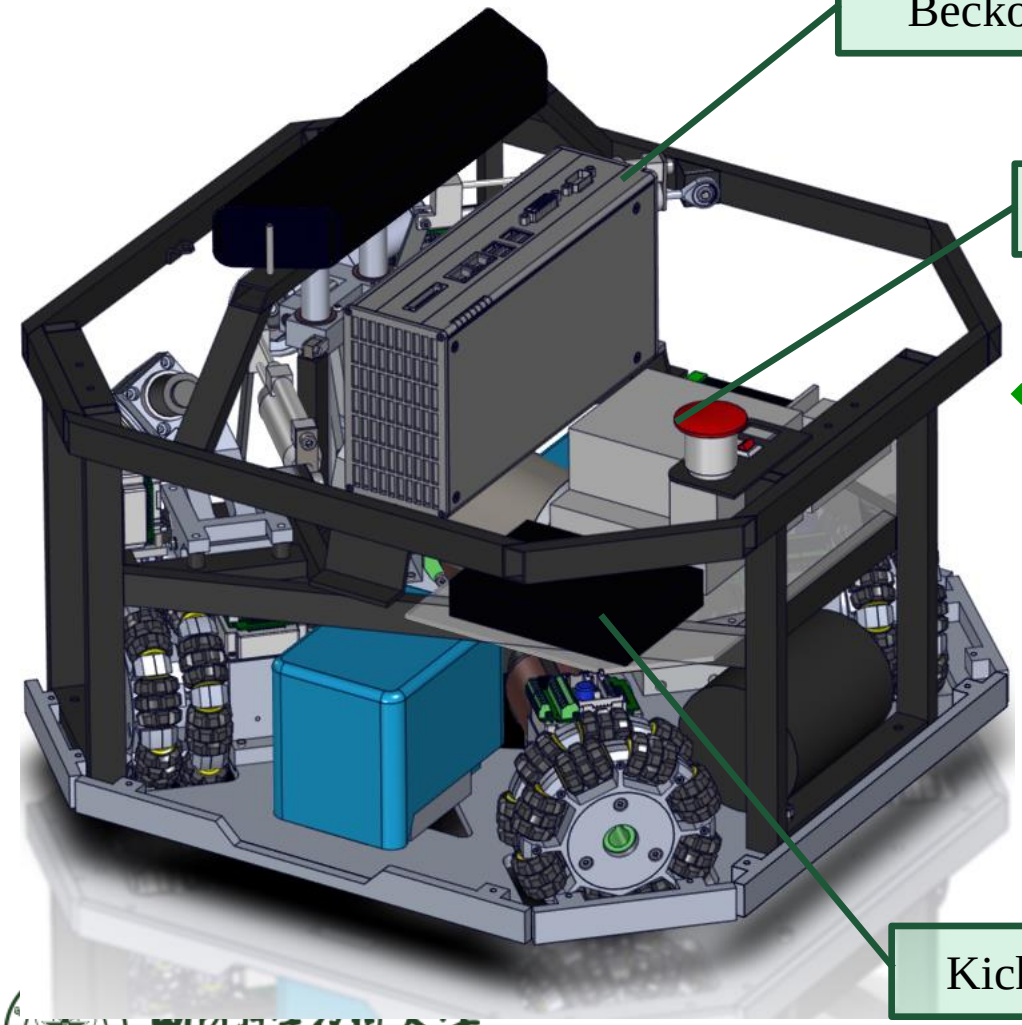
Welded Main Frame







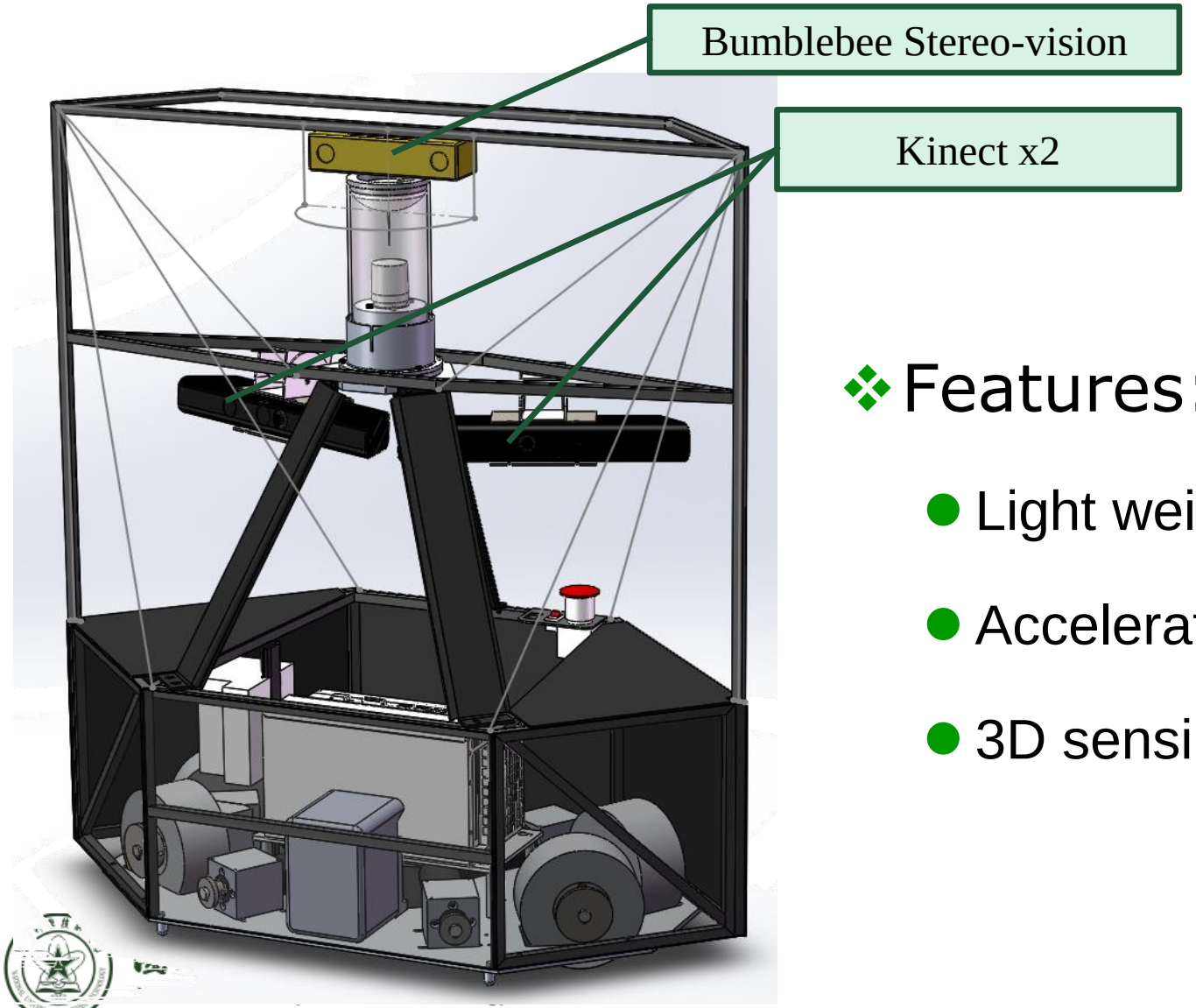
Welded Main Frame



- Features:
  - Light weights;
  - Easy separation between base plate and upper body;

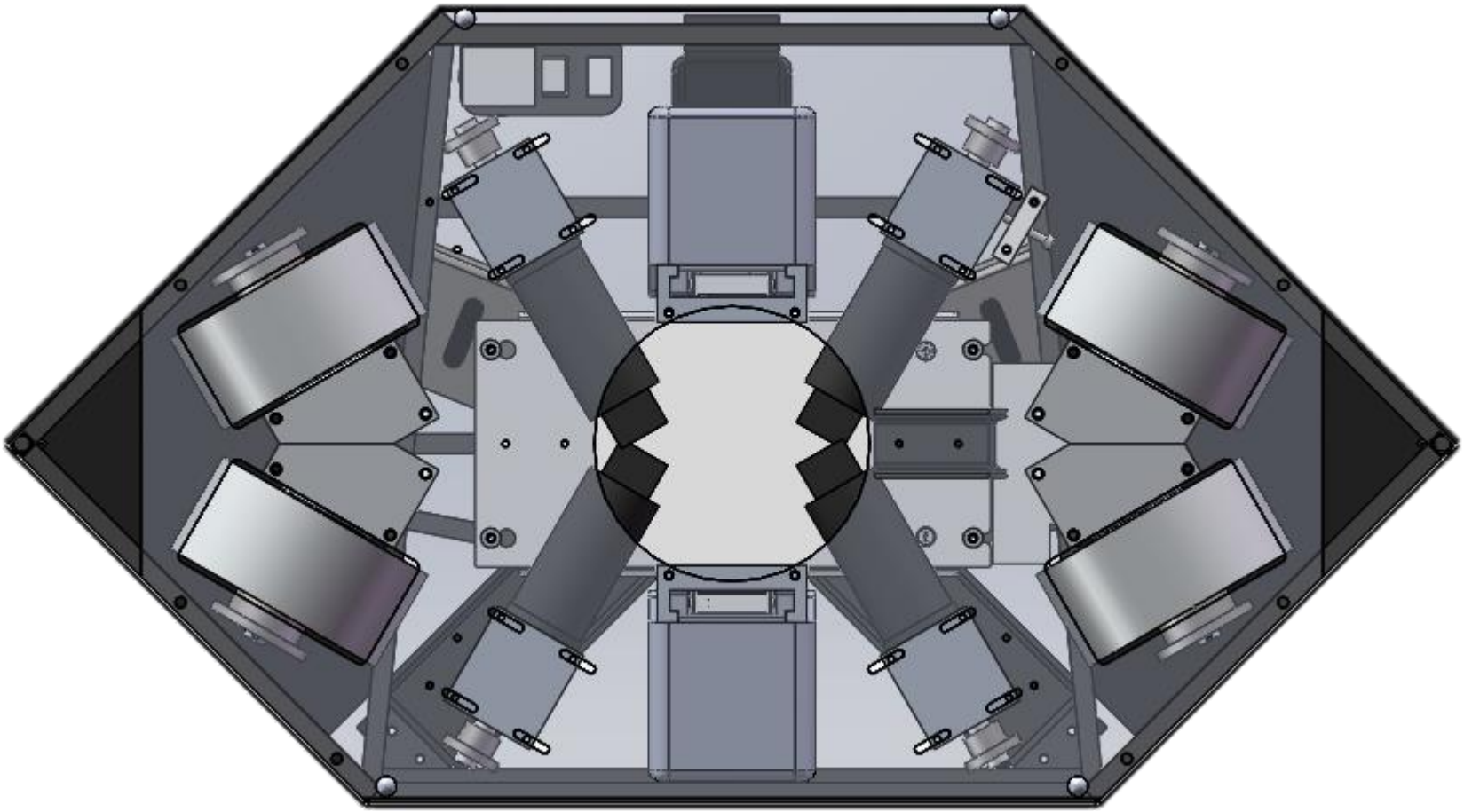
Vision Systems

Goalie



- Features:
  - Light weight;
  - Acceleration enhanced;
  - 3D sensing enhanced.

Goalie

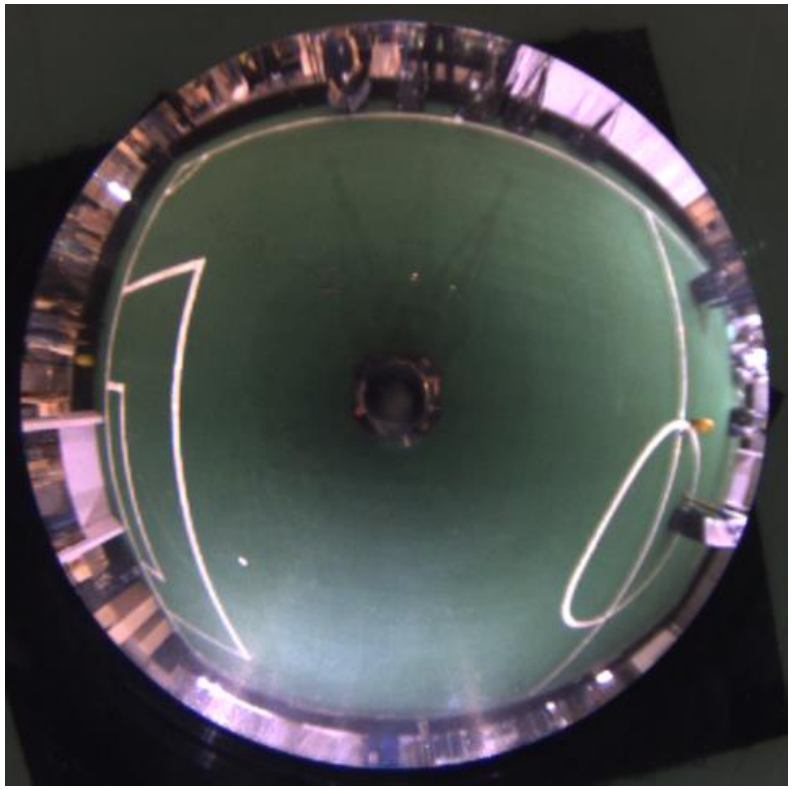


Inspired by TU/e

Outline

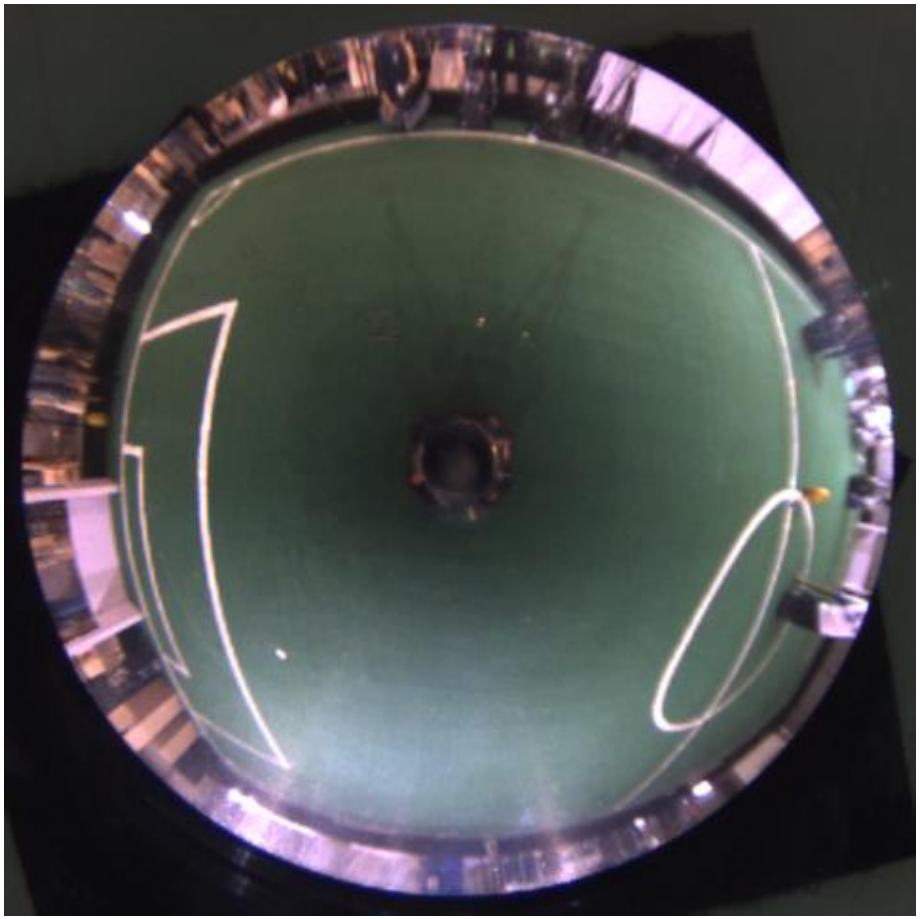
- New mechanical design
- The vision system
- The electrical system
- ROS-based softwares

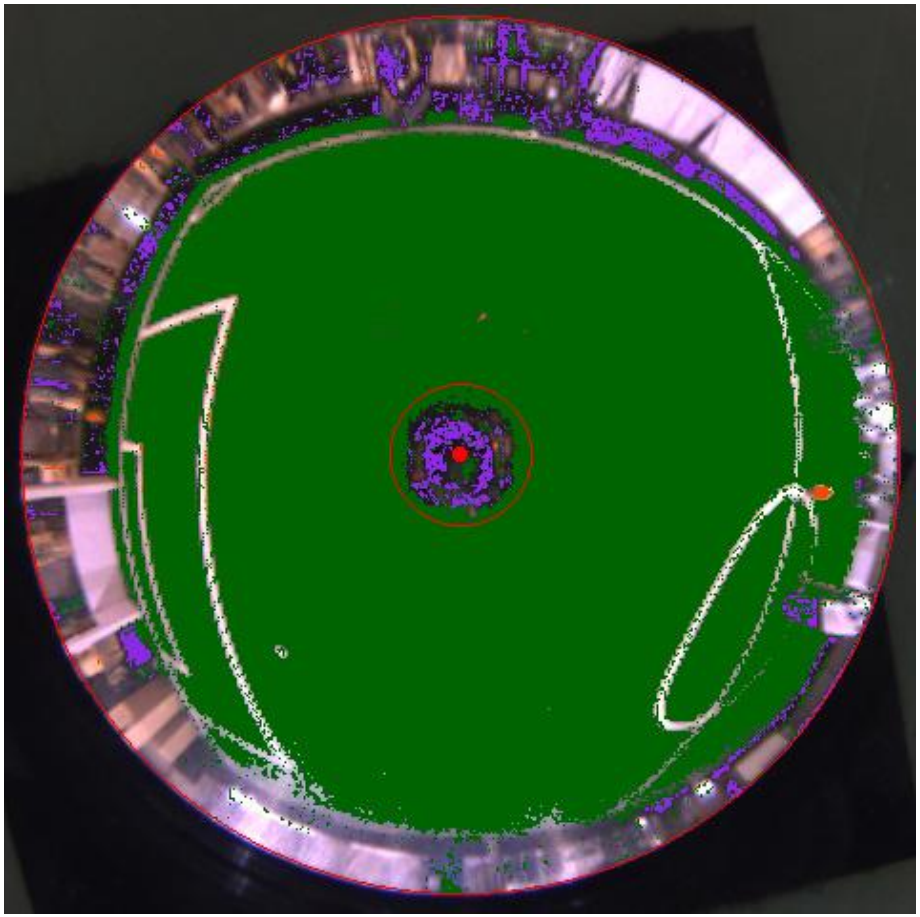
Omnidirectional vision for NuBot



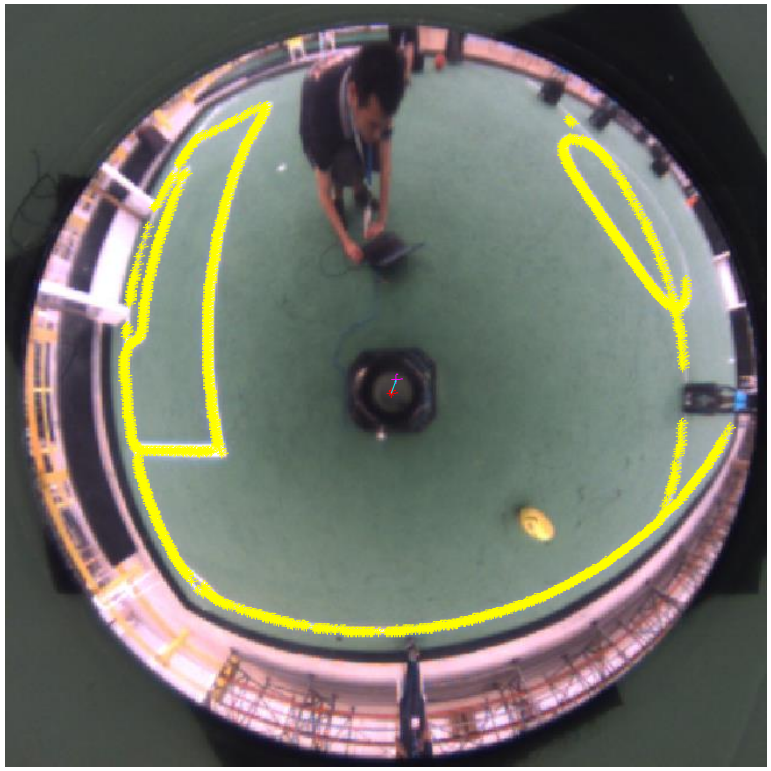
The mirror was developed by TU/e

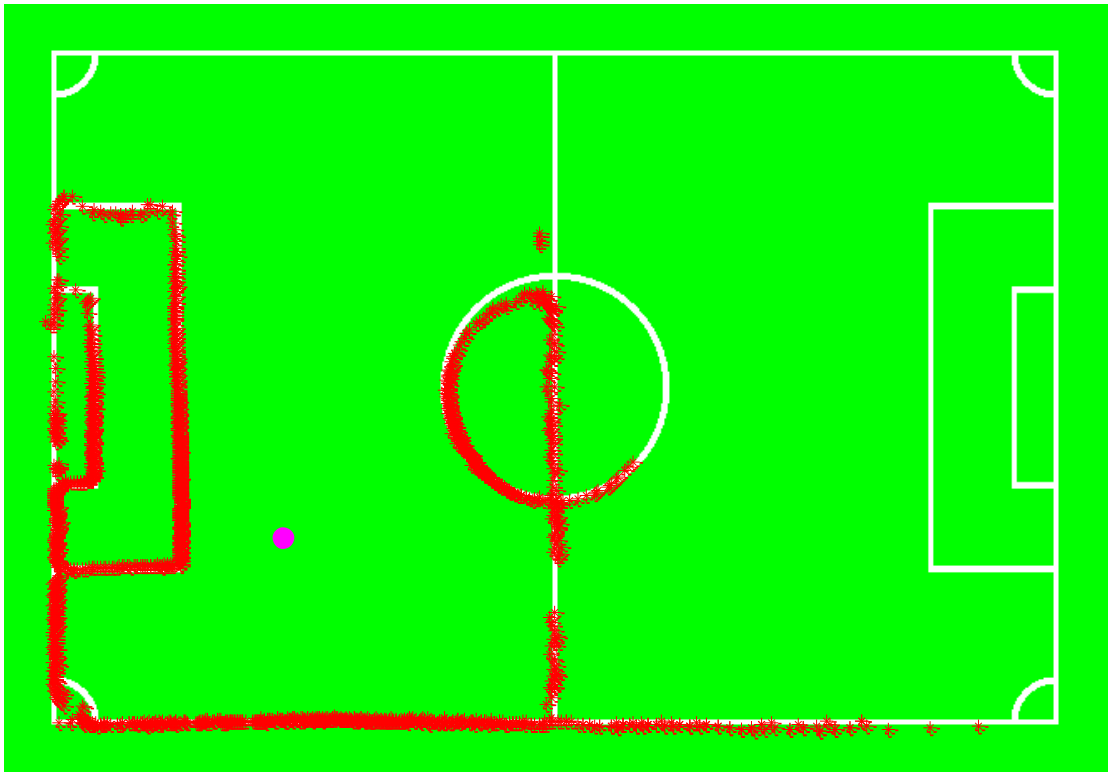
Omnidirectional vision for NuBot



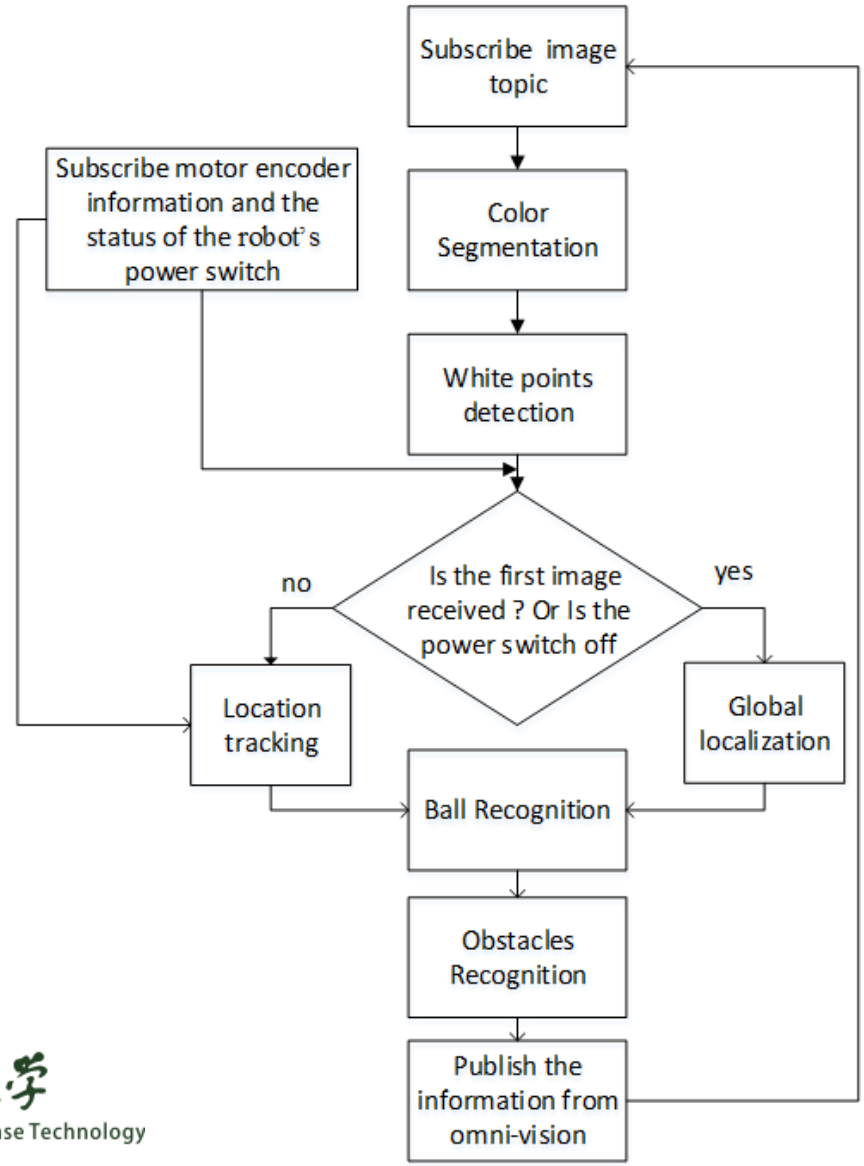


Omnidirectional vision for NuBot

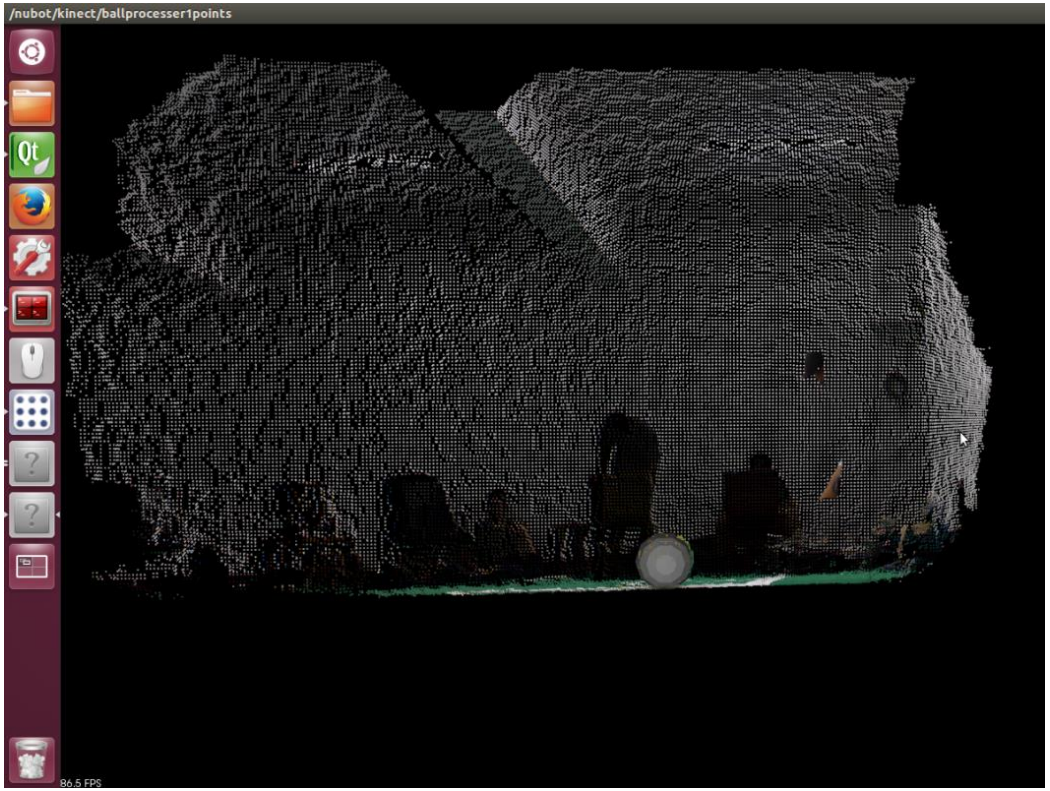




Flow Diagram of Omnivision Node



RGB-D Sensors for Goalie



Images acquiring and processing with OpenNI+PCL

Algorithm Flow Chart

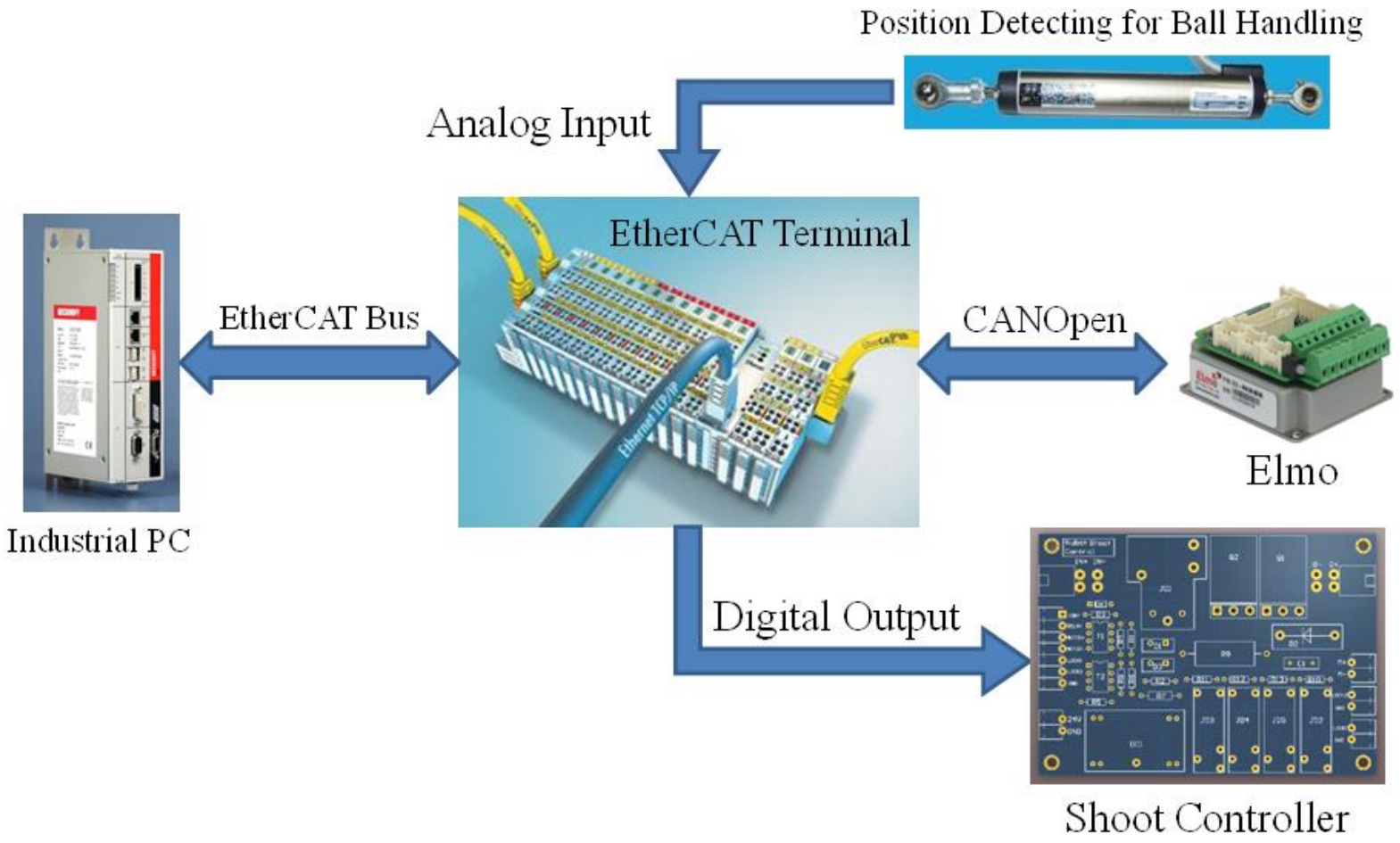
1. Find out ball candidates witch color-segmentation
2. Sphere fitting with RANSAC
3. Accumulate a history of ball positions(3d)
4. Parabola Fitting with Least-square-method
5. Calculate the touchdown-point

Demo

Outline

- New mechanical design
- The vision system
- The electrical system
- ROS-based softwares

System overview



Onboard PC

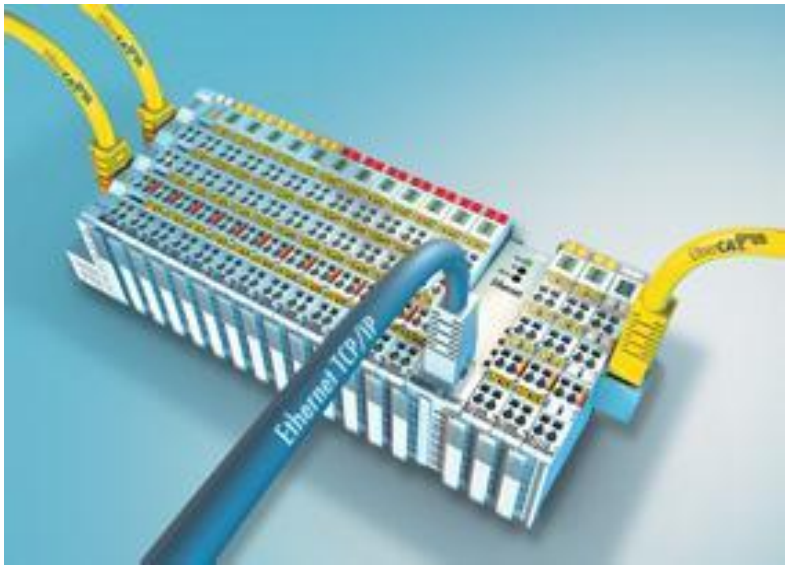
Beckoff Industrial PC:

- **Size** : 65 x 231 x 116 mm;
- **CPU** : Intel® Core™ i7, 4 Cores;
- **RAM** : 8G, DDR3;
- **HDD** : 60G, SSD;
- **Power:** DC 24V;

❖ Features:

- Compact;
- Reliable;
- High performance;
- Very Expensive!

EtherCAT Terminals



- **EK1100**
  The EtherCAT Coupler terminal which is essential.
- **EL2008**
  Digital Output terminal for shooting control.
- **EL6751**
  Gateway terminal to bridge EtherCAT with CANOpen.
- **EL3064-0010**
  Analog Input terminal for ball position detecting.

Elmo motor controller

CANOpen

DC brushless motor

Motor power

Encoder

DC 24V

EL6751 Configuration Using SOEM

- SOEM - Simple Open-source EtherCAT Master.
- The configuration of EL6751 using SOEM is tricky and few have succeed!
- Solution:
  - Generate Startup codes and configuration codes with the help of TwinCAT, which is supported by Beckhoff;
  - Configure EL6751 with the codes generated above using simple functions of SOEM;

◆The configuration codes generation using TwinCAT
(please find the details in the supplement slides)

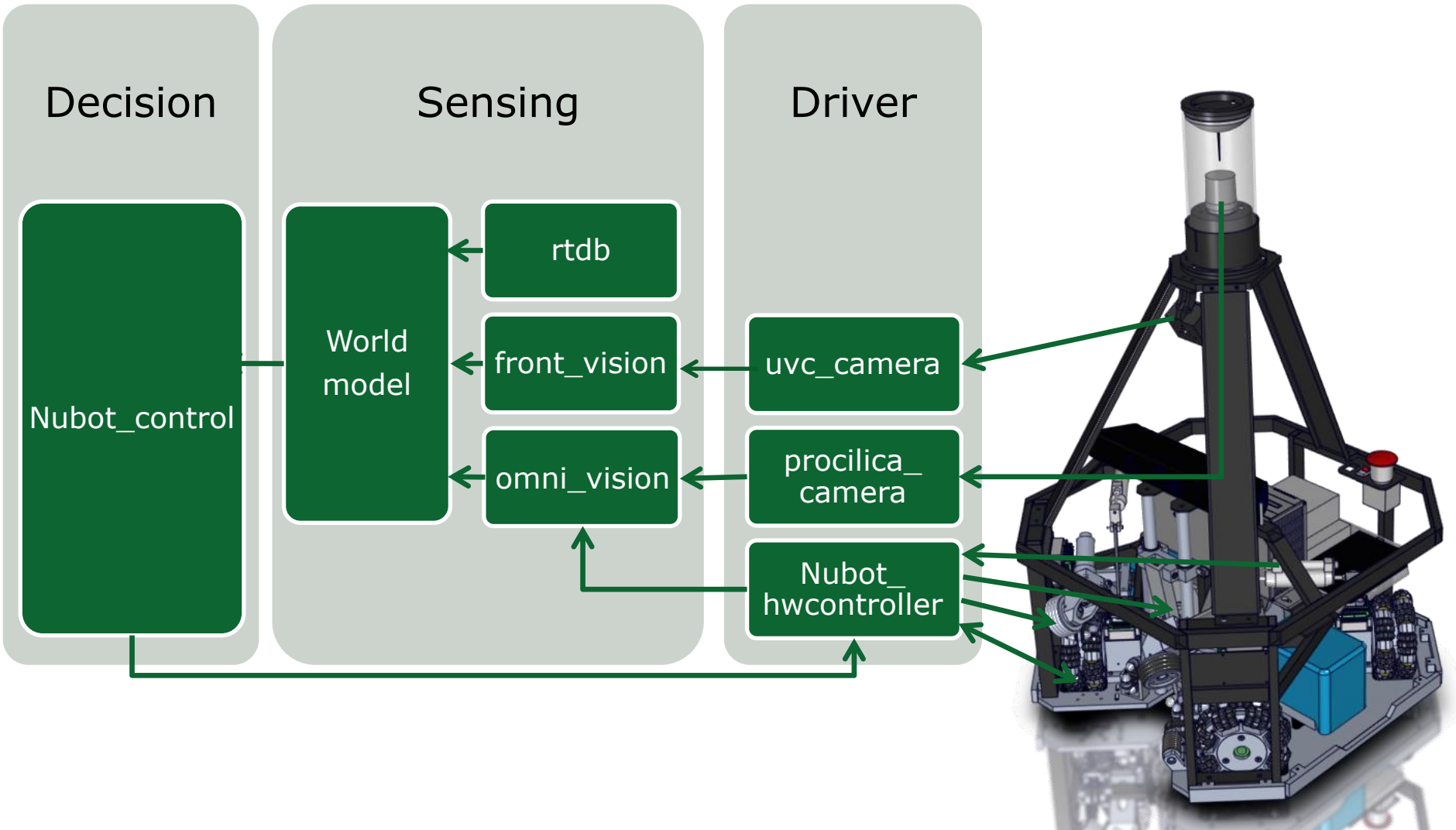
Outline

- New mechanical design
- The vision system
- The electrical system
- ROS-based softwares

Why ROS?

- Great flexibility and easy to use;
- The modularity and re-usability of the code can be improved;
- Lots of useful tools for debugging;
- Maybe a good chance for promoting the mixed team for RoboCup. ROS community is trying to provide better support for multi-robot system;
- Better data logging and sharing between teams?

System Structure of Nubot



Existing problems

- Not optimized for real-time application, users have to DIY process management and scheduling;
- Version problems. Some driver packages are obsoleted and not supported in newer ROS version;
- Solutions to fix questions depend on ros.answer, but seldom have replies;
- A kinds of small bugs;

RT-patched Linux kernel for Ubuntu

micROS linux r t
项目得分：35

3	2	5
成员	关注	缺陷

概述

This project aims at replacing the van illa Linux kernel in Ubuntu using RT-pr eempt patched kernel. The goal is to provide real time capabilities on the O S layer to real-time demanding ROS n odes in the future. Necessary perform ance tests and comparisons will also be carried out in this project.

使用RT-preempt补丁替换ubuntu的Lin ux内核，并进行相应测试。
目的是从操作系统层面改善内核的实 时性，从而提高ROS在单节点上的实 时性。

创建时间：2014-07-21 08:53

标签:
暂无标签！

项目工具集

项目交流

项目动态 | 缺陷 | 新闻 | 讨论区 | 版本库

rt patched Linux kernel for Ubuntu

An rt patched Linux kernel for Ubuntu was released, under the version 0.0.1.

Hardware/software environment:

- i386 CPU;
- Ubuntu 12.04 32-bit system;
- Working for both physical and virtual machines.

Installation:

Copy the attached Debian package to any directory in Ubuntu and run:

```
sudo dpkg -i linux-image-XXXX.deb
```

When completed, reboot the system and the Linux kernel is now replaced.

You can confirm the kernel version by running the following command:

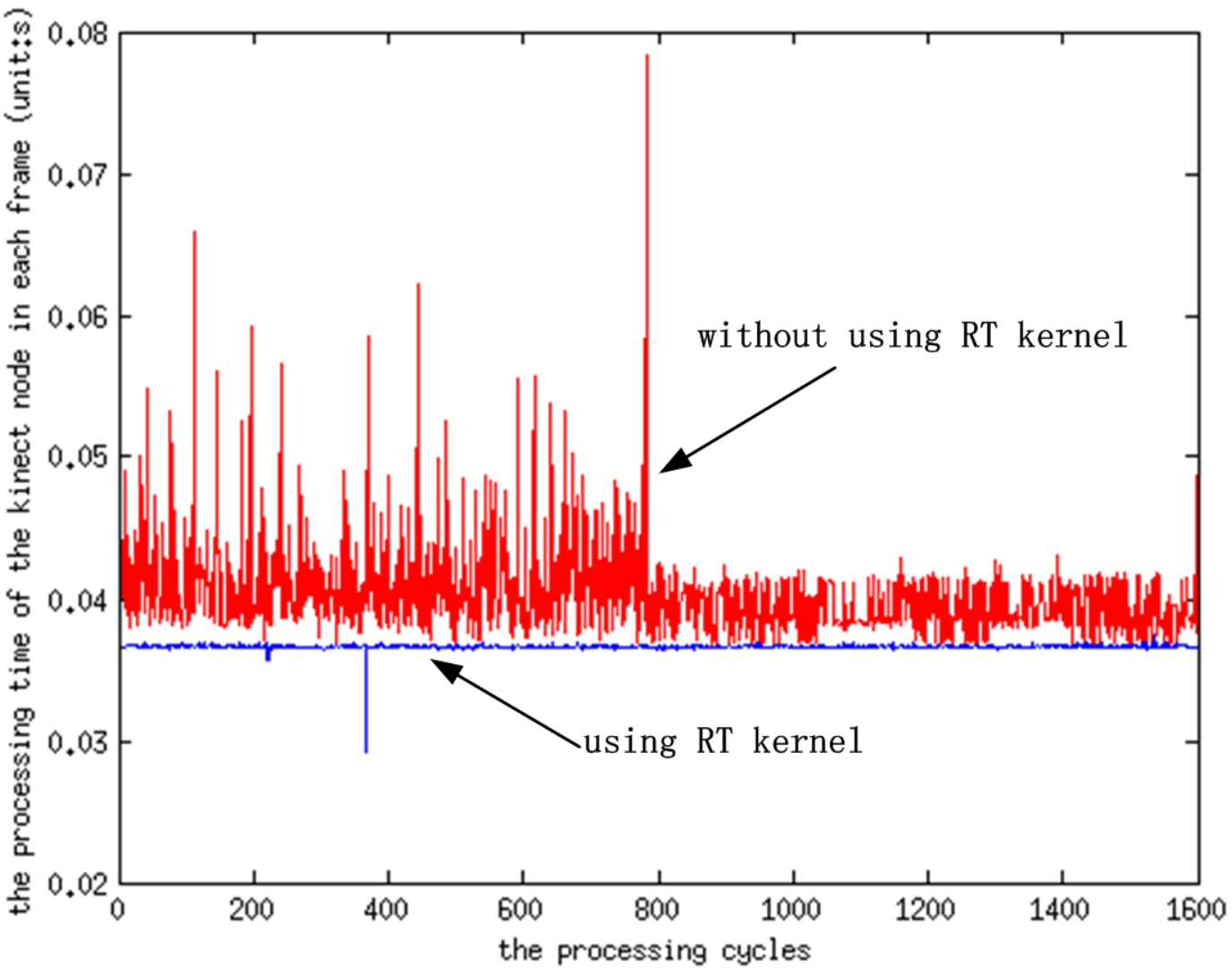
```
uname -a
```

The current rt patched kernel should be postfixed with -rt.

linux-image-3.8.13-r... (44.182 MB) VolodjaNiu, 2014-09-10 09:45
patch-3.8.13-rt16.pa... (152.396 KB) sukha, 2014-09-26 22:50
config-3.8.13-rt16_x86 (156.103 KB) sukha, 2014-09-26 22:50
linux-3.8.13.tar.bz2 (80.666 MB) sukha, 2014-09-26 23:06

- http://micros.nudt.edu.cn/ros/news/217

Experiment



Processing time in the kinect node with and without RT Kernel

micROS RT

micROS RT

an implementation of ROS 1.x on DDS

Download ZIP File | Download TAR Ball | View On GitHub

Please contact us through bding@msn.com. Any feedback would be greatly appreciated.

More information can be found in the user's manual of micROS RT (download) and our website http://micros.nudt.edu.cn/

Introduction

micROS RT (micROS Real-Time) is a modified ROS C++ kernel which adopts OMG's DDS (Data Distribution Service for Real-time Systems) as its underlying message transfer protocol. DDS is an Object Management Group's standard for pub/sub middleware (http://portals.omg.org/dds/). It supports high-performance, scalable and QoS-assuring message delivery. It has been applied into many industry-level systems. By replacing the original ROS message protocols (TCPROS & UDPROS) with DDS, a set of advanced features can be supported in the ROS message delivery process.

(1) **Built-in multicast support**. When there are *n* listeners in a topic (*n*>=2), significant performance advantage can be obtained.

(2) **Robustness in some adverse network environments**. For example, it has better reconnection behavior when dropping out of wireless (according to the report of Dirk Thomas link).

(3) **Real-time and other QoS assurance in message delivery**. For example, you can set the transport priority and latency budget of messages, specify expected message arriving deadline and the behavior when the deadline is not met, set time-based filter to the messages on a topic, and so on.

Thanks for your attention!

https://github.com/nubot-nudt/