A Remote-controlled Robot-car in the TPS Tunnel

23 May, 2019, Melbourne, Australia

Tsung-Yu Lee
Accelerator Operation Group
NSRRC, Taiwan

OUTLINE

PARAMETERS

Energy (GeV)	3
Current (mA)	500
Circumference (m)	518.4
Straight section (m)	18 x 7 + 6 x 12
Bare lattice emittance (nm-rad)	1.6
Betatron tune (ν_x/ν_y)	DMB 26.14 / 13.24 Bare lattice (24.18/14.24)
Natural chromaticity ξ_x/ξ_y	-75/-26
Periodicity	6
Bending radius (m)	8.40338
Natural energy spread	8.86×10^{-4}
Momentum compaction α_1/α_2	$2.4\text{x}10^{-4}/2.1\text{x}10^{-3}$
Revolution frequency (kHz)	578.3
RF frequency (MHz)	499.654
Harmonic number	864
Synchrotron tune	0.00609
Bunch length (mm)	2.86

TAIWAN PHOTON SOURCE

MILESTONE

1. Oct.2013 Accelerator installation
2. 31.Dec. 2014 Stored beam and first light in storage ring
3. Dec. 2015 Achieved 520 mA stored beam
4. May 2016 300 mA top-up operation
5. Sep. 2016 Opened to general users
6. Nov. 2017 Started 400 mA operation
7. Jan. 2019 Approval for 500 mA operation

...

Oct. 2019 Start 500 mA operation


What is it like inside the tunnel
during normal operation?

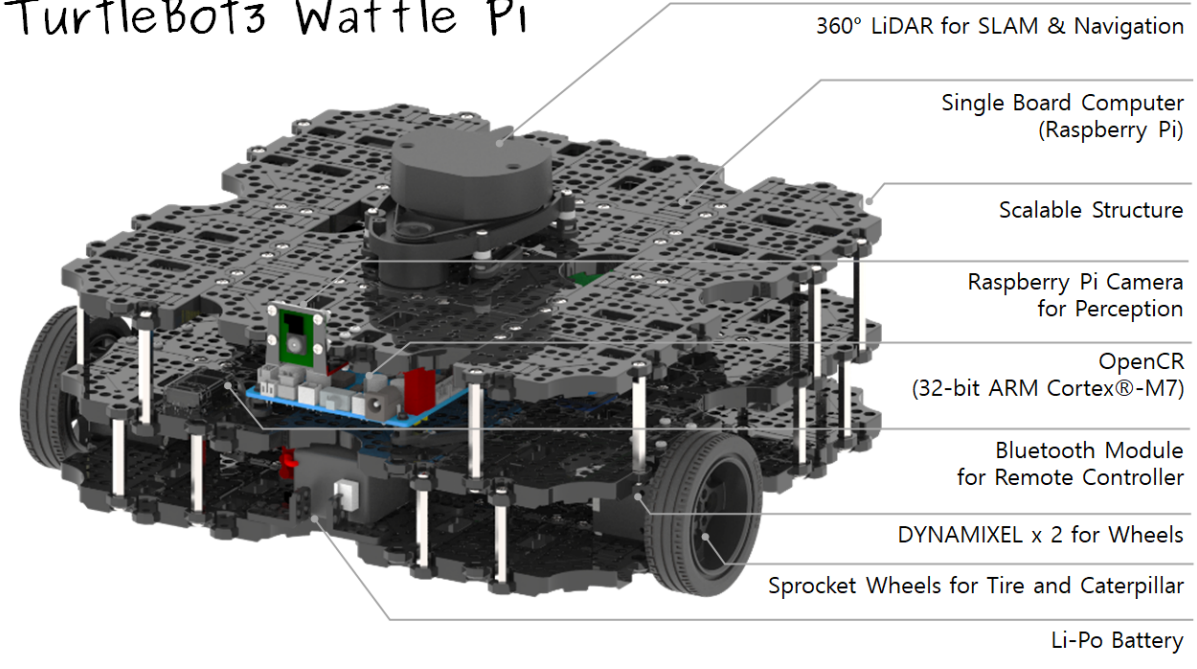
OUTLINE

SPECIFICATIONS

Maximum translational velocity	0.26 m/s
Maximum rotational velocity	1.82 rad/s (104.27 deg/s)
Maximum payload	30kg
Size (L x W x H)	281mm x 306mm x 141mm
Weight (+ SBC + Battery + Sensors)	1.8kg
Threshold of climbing	10 mm or lower
SBC (Single Board Computers)	Raspberry Pi 3 Model B and B+
Actuator	Dynamixel XM430-W210
LDS(Laser Distance Sensor)	360 Laser Distance Sensor LDS-01
Camera	Raspberry Pi Camera Module v2.1
IMU	Gyroscope 3 Axis
	Accelerometer 3 Axis
	Magnetometer 3 Axis
Power connectors	3.3V / 800mA
	5V / 4A
	12V / 1A
Battery	Lithium polymer 11.1V 1800mAh / 19.98Wh 5C

ROBOTIS

TurtleBot3 Waffle Pi



http://emanual.robotis.com/docs/en/platform/turtlebot3/specifications/#specifications

FEATURES

Robot Operating System

WORLD'S MOST POPULAR ROS PLATFORM

TurtleBot is the world's most popular open source robot
for education and research.

AFFORDABLE COST

TurtleBot is the most affordable platform
for educations and prototype research & developments.

SMALL SIZE

Imagine the TurtleBot in your backpack and bring it anywhere.

EXTENSIBILITY

Extend ideas beyond imagination
with various SBC, sensor, motor and flexible structure.

MODULAR ACTUATOR

Easy to assemble, maintain, replace and reconfigure.

OPEN SOURCE SOFTWARE

Variety of open source software for the user.
You can modify downloaded source code and share it with your friends.

OPEN SOURCE HARDWARE

Schematics, PCB Gerber, BOM and 3D CAD data
are fully opened to the user.

STRONG SENSOR LINEUPS

8MP Camera, Enhanced 360° LiDAR,
9-Axis Inertial Measurement Unit and precise encoder for your robot.

http://emanual.robotis.com/docs/en/platform/turtlebot3/features/#features

OUTLINE

01 INTRODUCTION

02 SPECIFICATIONS & FEATURES

03 **BASIC OPERATIONS**

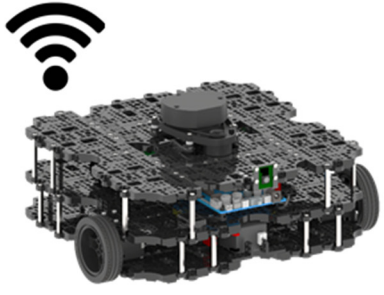
04 APPLICATION

05 PROSPECT

06 CONCLUSION

BASIC OPERATIONS

TurtleBot



Router

NSRRC-ROSD

Remote PC

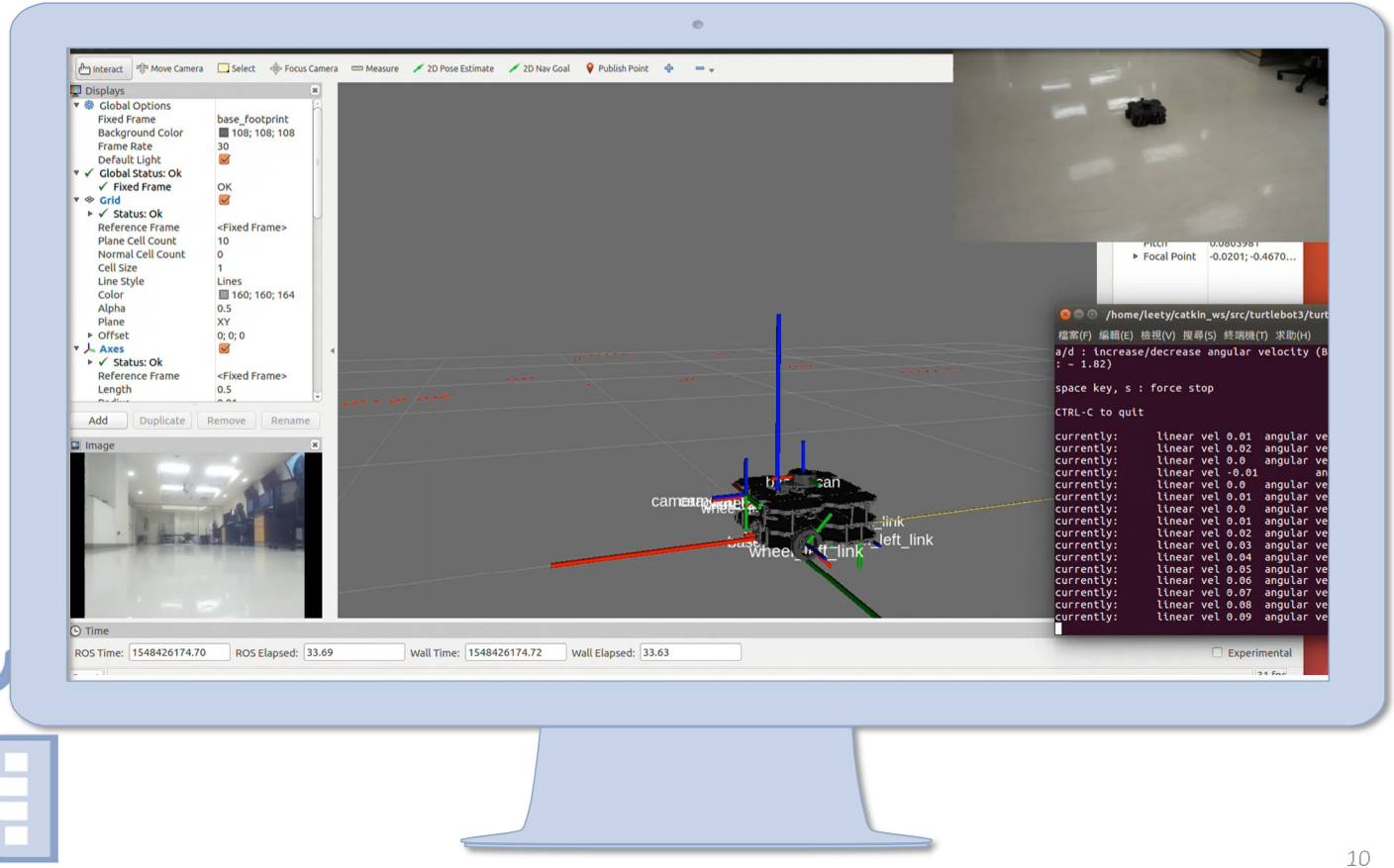
ROS_MASTER_URI = http://IP_OF_REMOTE_PC:11311
ROS_HOSTNAME = IP_OF_TURTLEBOT

ROS_MASTER_URI = http://IP_OF_REMOTE_PC:11311
ROS_HOSTNAME = IP_OF_REMOTE_PC

* Example when ROS Master is running on the Remote PC

http://emanual.robotis.com/docs/en/platform/turtlebot3/pc_setup/#pc-setup

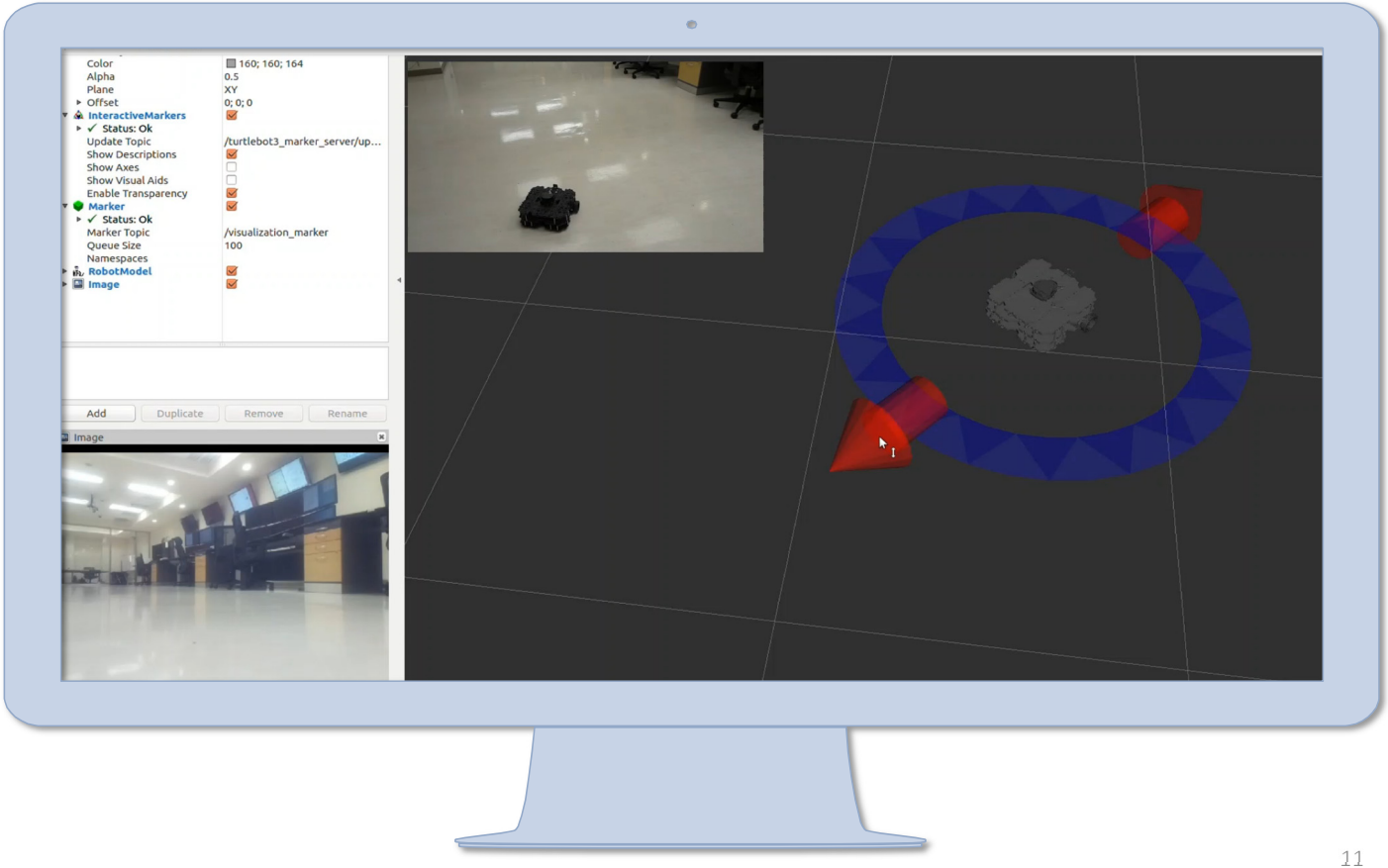
BASIC OPERATIONS



Keyboard

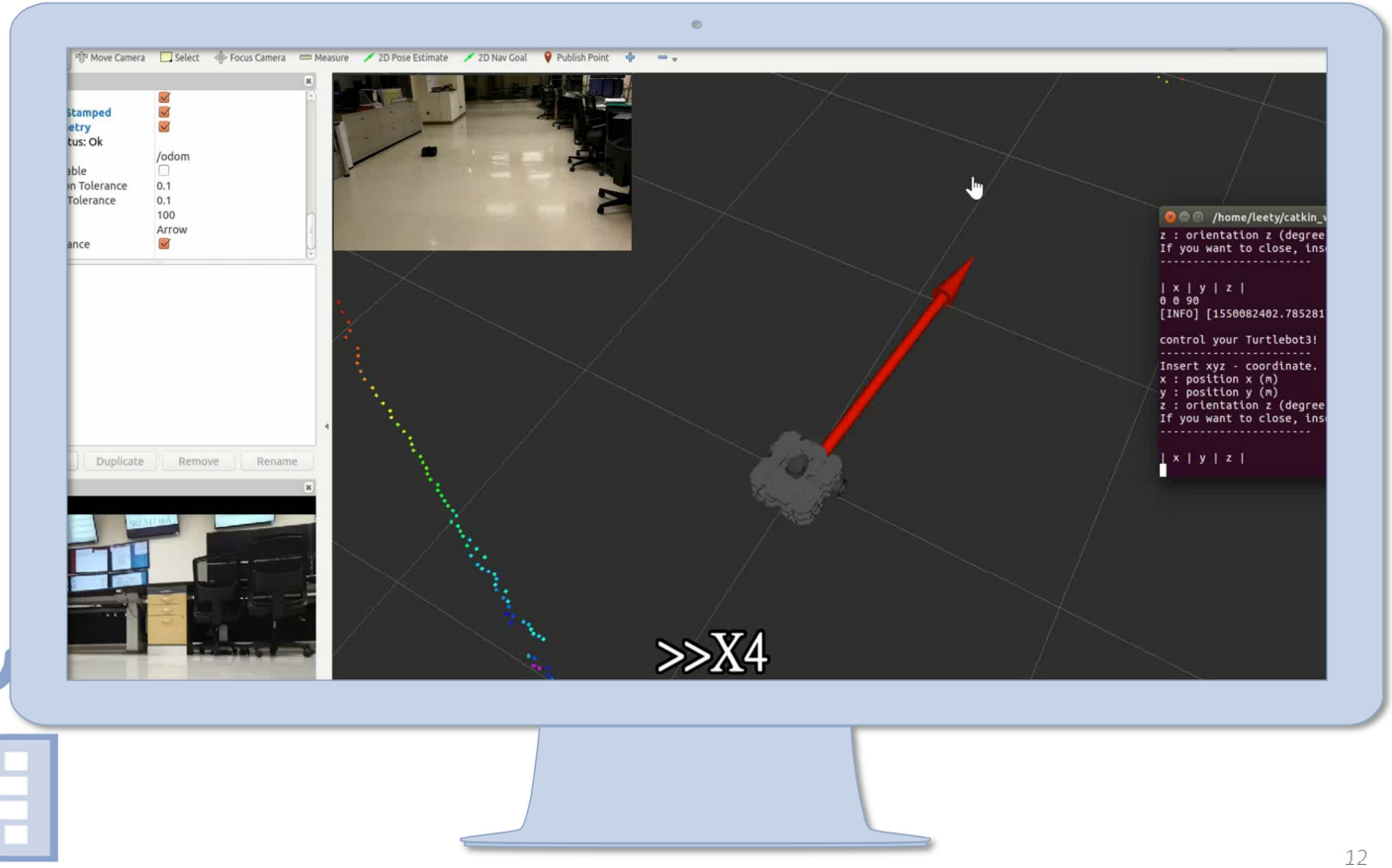
BASIC OPERATIONS

Interactive Markers



BASIC OPERATIONS

Point Operation



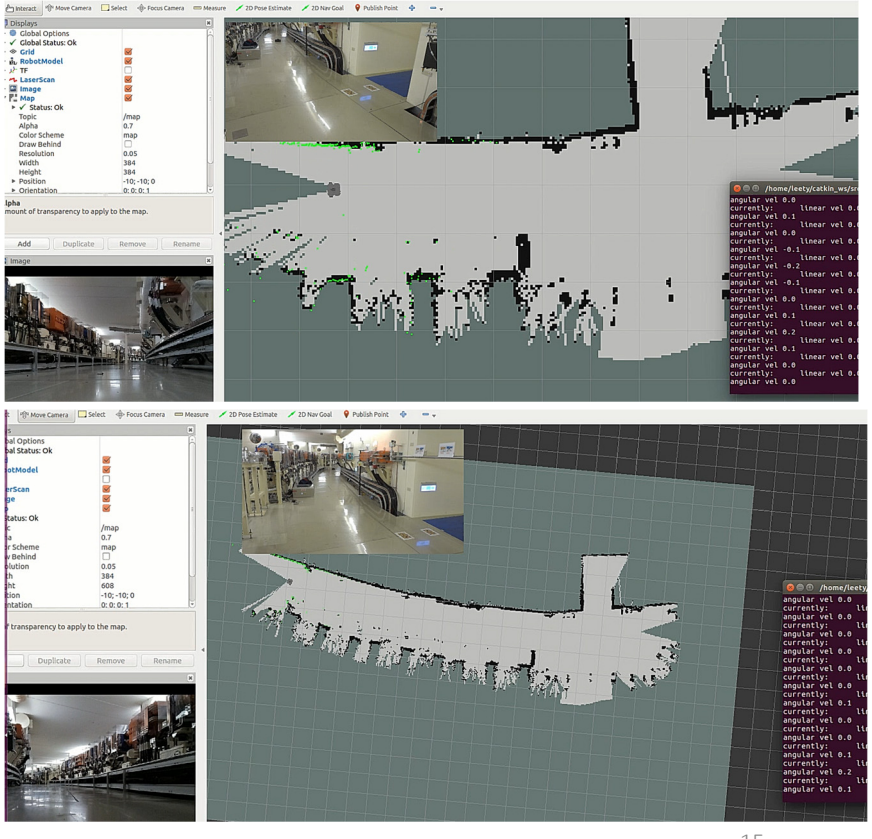

Where is the robot ?

POSITIONING

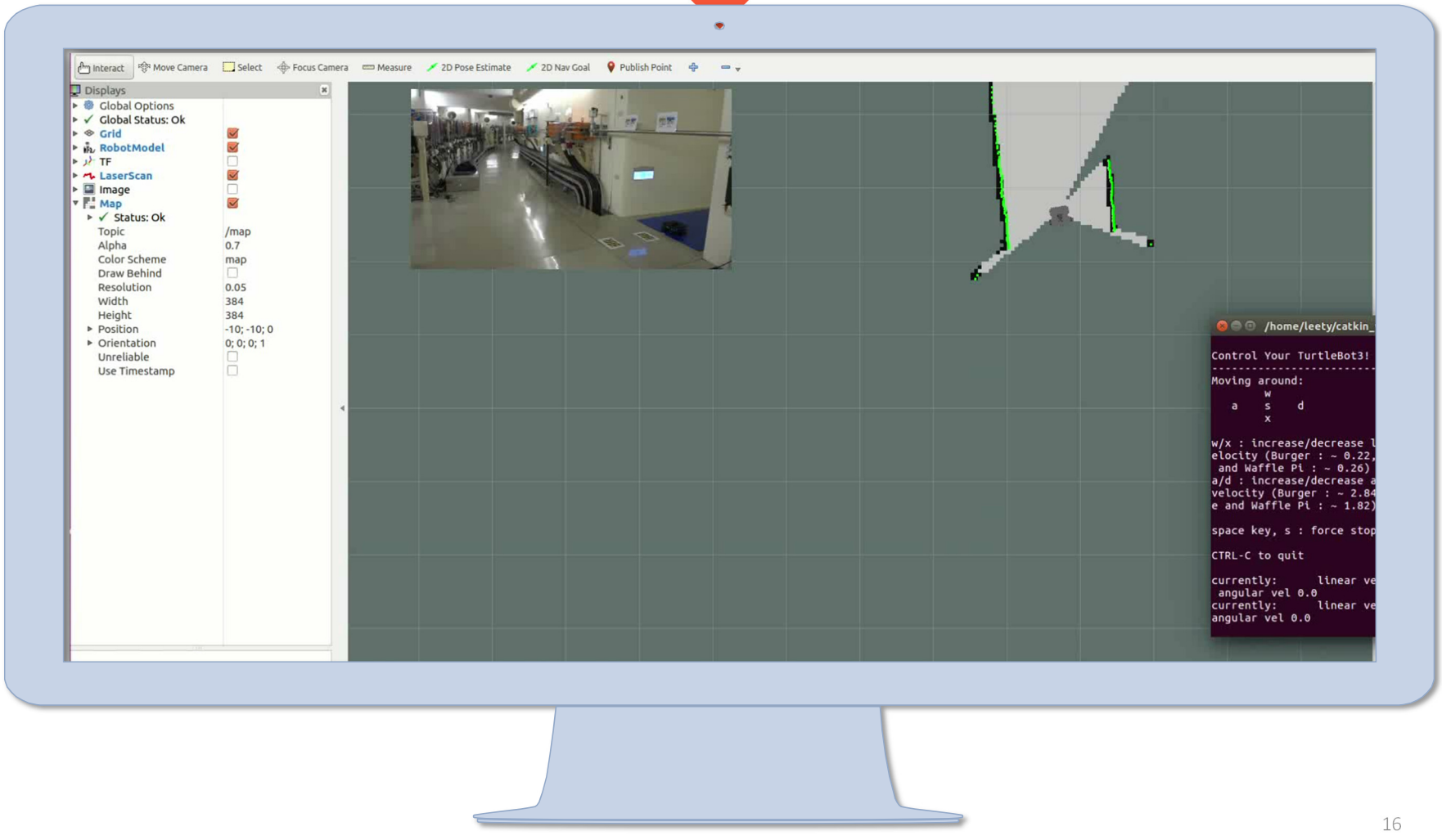
- GPS ✗
- Indoor Positioning Sensor ✗
  - Landmark(color,IR camera)
  - Indoor GPS
  - WiFi SLAM
  - Beacon
- Guide tape/wire ✗

Simultaneous **L**ocalizaton **A**nd **M**apping(**SLAM**)

- Position: estimating the robot's position
  - Dead reckoning
    - Using the encoder values of both wheel axes
    - Calculate moving distance and moving angle, and then estimate position
    - Floor slip, mechanical, cumulative errors
    - Position compensation with inertial sensor such as IMU
    - Particle filter, Kalman filter
- Sensing: measuring obstacles
  - LiDar
- Map: building map with road and obstacles information
  - Gmapping, Cartographer, Hector Mapping

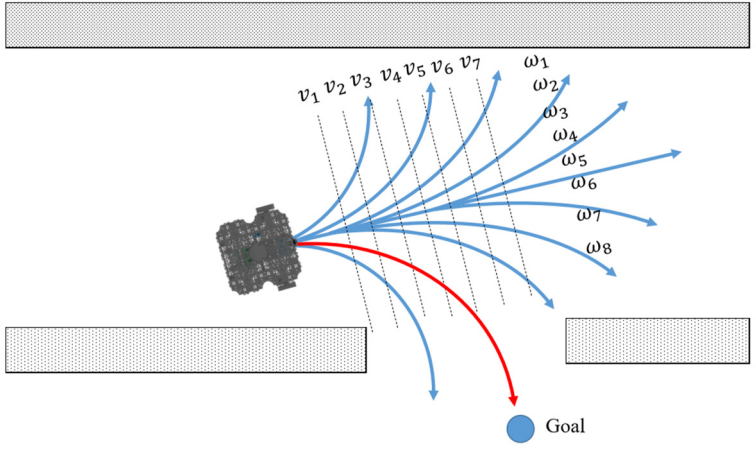


*https://github.com/ROBOTIS-GIT/ros_seminar/blob/master/11_SLAM_and_Navigation.pdf


Interact
Move Camera
Select
Focus Camera
Measure
2D Pose Estimate
2D Nav Goal
Publish Point
Displays
Global Options
Global Status: Ok
Grid
RobotModel
TF
LaserScan
Image
Map
Status: Ok
Topic /map
Alpha 0.7
Color Scheme map
Draw Behind
Resolution 0.05
Width 384
Height 384
Position -10; -10; 0
Orientation 0; 0; 0; 1
Unreliable
Use Timestamp
/home/leety/catkin_
Control Your TurtleBot3!
Moving around:
w
a s d
x
w/x : increase/decrease l
elocity (Burger : ~ 0.22,
and Waffle Pi : ~ 0.26)
a/d : increase/decrease a
velocity (Burger : ~ 2.84
e and Waffle Pi : ~ 1.82)
space key, s : force stop
CTRL-C to quit
currently: linear ve
angular vel 0.0
currently: linear ve
angular vel 0.0

Dynamic Window Approach

- The search space is restricted to safe circular trajectories that can be reached within a short time interval and are free from collisions.
- The optimization goal is to select a heading and velocity that brings the robot to the goal with the maximum clearance from any obstacle.



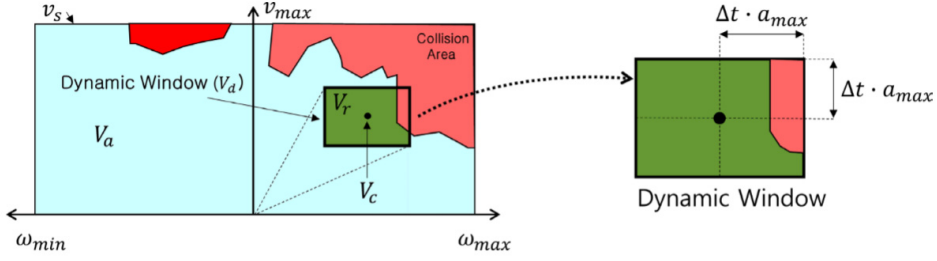
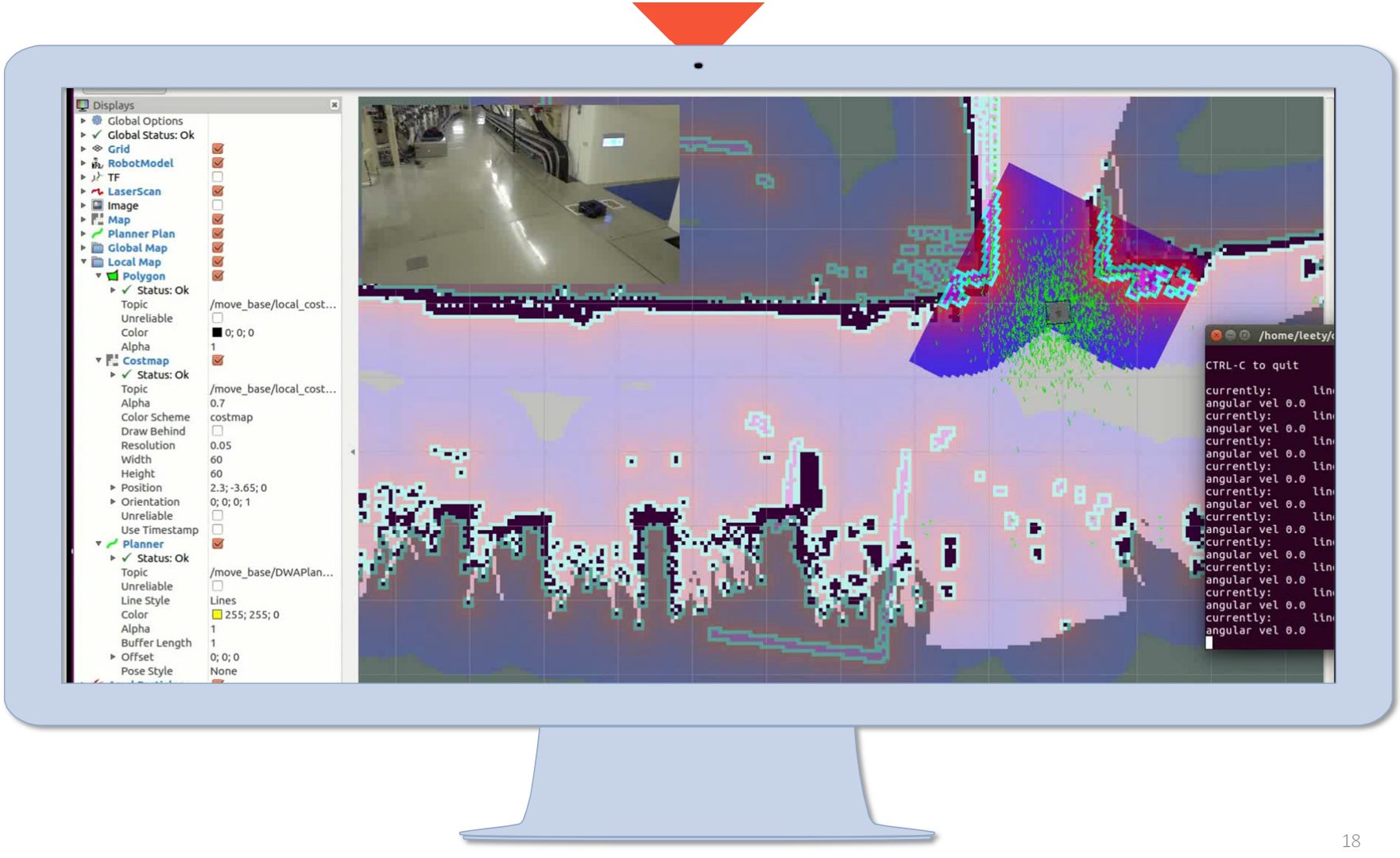


Fig: Robot's velocity search space and dynamixel window

v: Translational velocity (meter/sec)
ω: Rotational velocity (radian/sec)
V_s: Maximum velocity area
V_a: Permissible velocity area
V_c: Current velocity
V_r: Speed area in Dynamic Window
a_{max}: Maximum acceleration / deceleration rate
Objective function:
$G(v,\omega) = \sigma \cdot (\alpha \cdot heading(v, \omega) + \beta \cdot dist(v, \omega) + \gamma \cdot velocity(v, \omega))$

*Fox, D.; Burgard, W.; Thrun, S. (1997). "The dynamic window approach to collision avoidance". Robotics & Automation Magazine, IEEE. 4 (1): 23–33.


Displays
Global Options
Global Status: Ok
Grid
RobotModel
TF
LaserScan
Image
Map
Planner Plan
Global Map
Local Map
Polygon
Status: Ok
Topic
/move_base/local_cost…
Unreliable
Color
0; 0; 0
Alpha
1
Costmap
Status: Ok
Topic
/move_base/local_cost…
Alpha
0.7
Color Scheme
costmap
Draw Behind
Resolution
0.05
Width
60
Height
60
Position
2.3; -3.65; 0
Orientation
0; 0; 0; 1
Unreliable
Use Timestamp
Planner
Status: Ok
Topic
/move_base/DWAPlan…
Unreliable
Line Style
Lines
Color
255; 255; 0
Alpha
1
Buffer Length
1
Offset
0; 0; 0
Pose Style
None
/home/leety/
CTRL-C to quit
currently:
angular vel 0.0

OUTLINE

01 INTRODUCTION

02 SPECIFICATIONS & FEATURES

03 BASIC OPERATIONS

04 **APPLICATION**

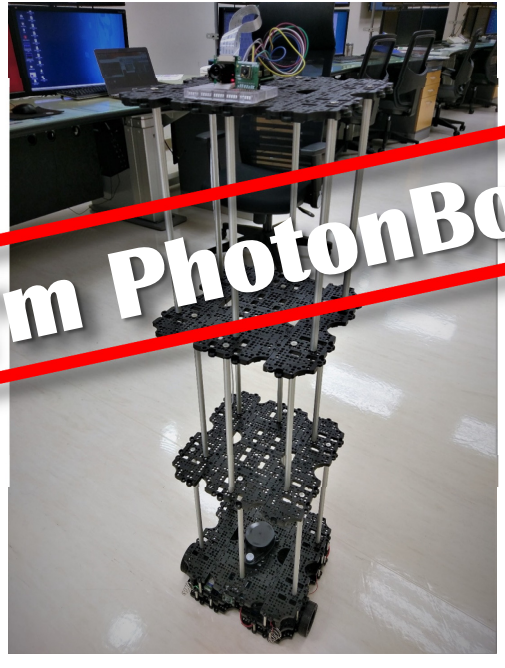
05 PROSPECT

06 CONCLUSION

BEFORE INSERTING THE ROBOT INTO THE TUNNEL…

Height

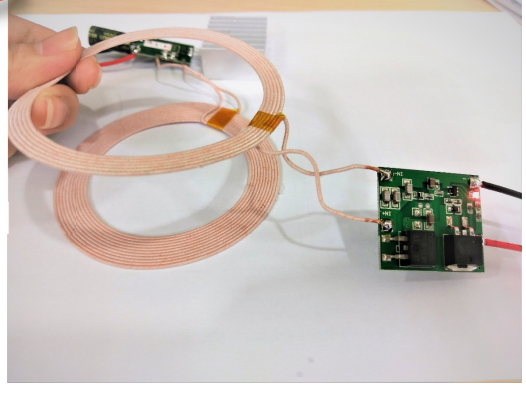
- The base plane for the **electron orbit** is **1350mm** from floor level.
- Three more layers, separated by a 300mm space, were added to gain the final robot height of **1100mm**.



Power

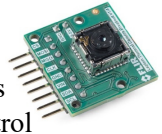
- The 1800mAh **Li-Po battery** was replaced with three batteries in parallel for a total of **16000mAh**.
- **wireless charging module**

	XKT801-29	Commercial
Output (volt)	12v	5v/9v
Power (watt)	40w	<10w(5w)
Charging distance	~5cm	<4cm (5mm)

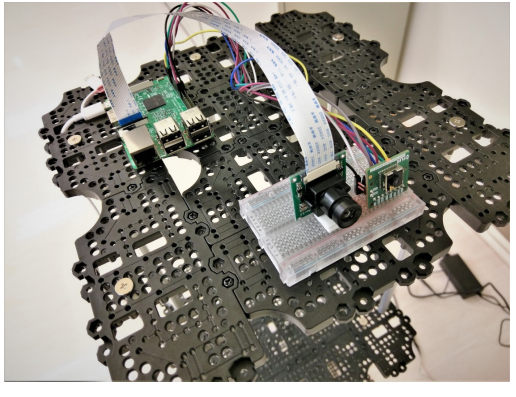


Sensors

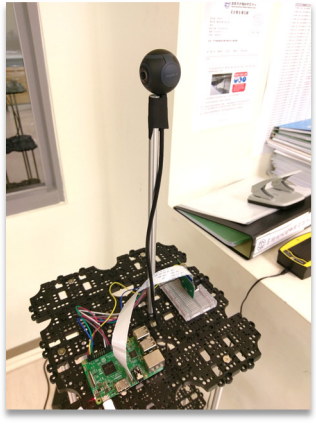
FLiR Dev Kit

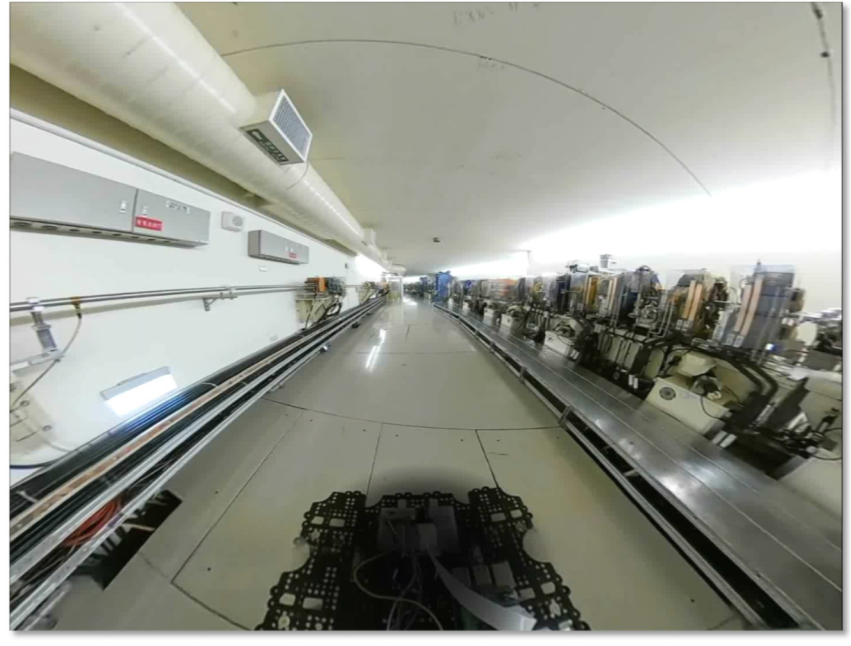


- LWIR sensor, wavelength 8 to 14 µm
- 51-deg HFOV, 63.5-deg diagonal
- 80 (h) × 60 (v) active pixels
- Thermal sensitivity <50 mK
- MIPI and SPI video interfaces
- Two-wire I2C-like serial-control interface
- Fast time to image (< 0.5 sec)
- Low operating power, nominally 150 mW (< 160 mW over full temperature range)



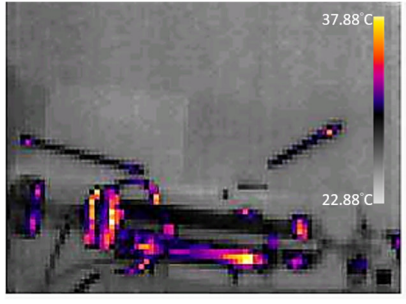
RESULTS
Virtual Reality



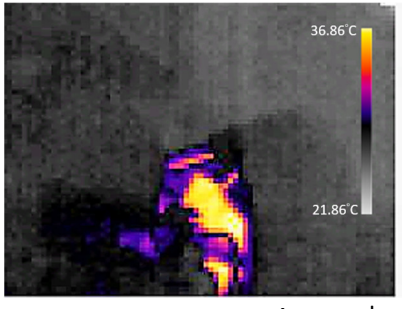


RESULTS

thermal images at 405mA stored beam

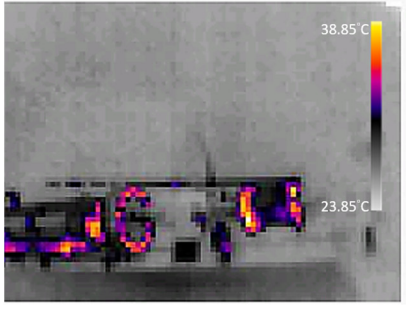


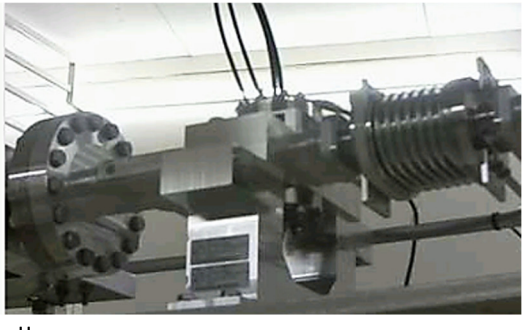
A strip line kicker



A quadrupole magnet

A flange



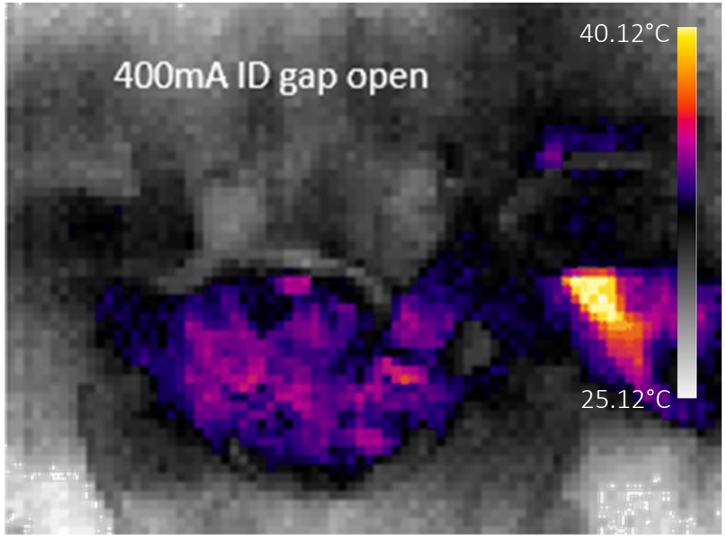


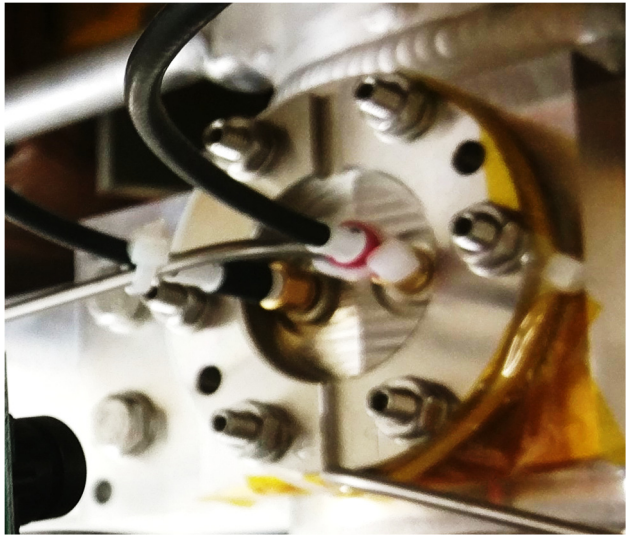
A flange and bellow

RESULTS

thermal images at 405mA stored beam

IU downstream BPM





OUTLINE

01 INTRODUCTION

02 SPECIFICATIONS & FEATURES

03 BASIC OPERATIONS

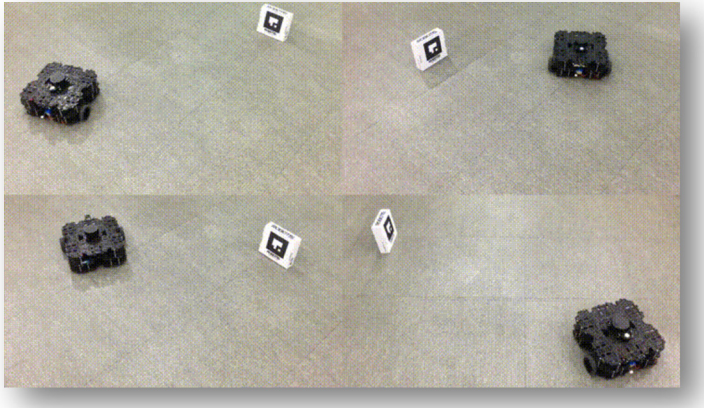
04 APPLICATION

05 PROSPECT

06 CONCLUSION

PROSPECT

- Robot arms
- Radiation detectors
- Imaging recognition
- Auto-parking
- Other sensors…



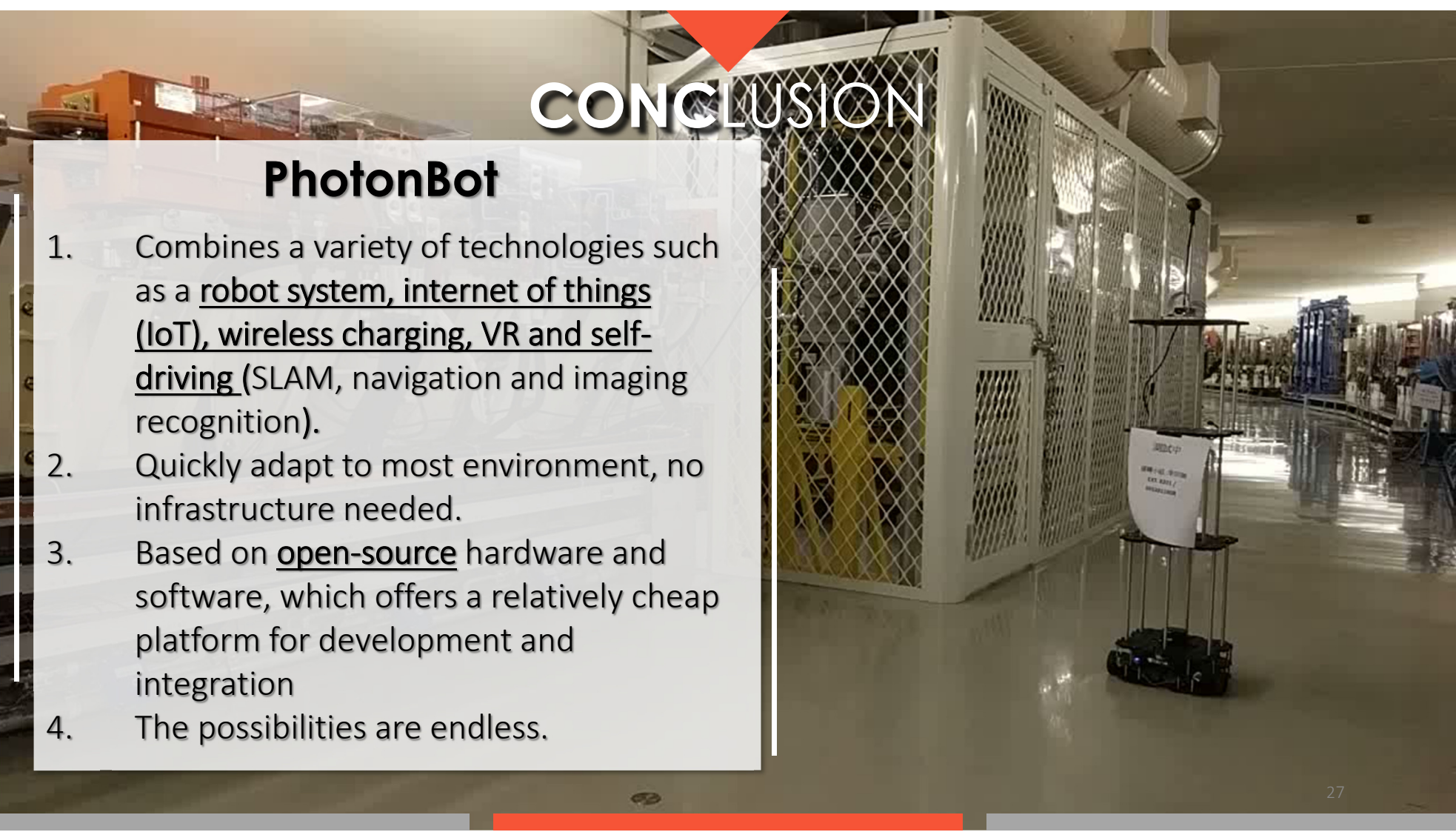
http://emanual.robotis.com/docs/en/platform/turtlebot3/applications/#applications

OUTLINE

CONCLUSION

PhotonBot

1. Combines a variety of technologies such as a robot system, internet of things (IoT), wireless charging, VR and self-driving (SLAM, navigation and imaging recognition).
2. Quickly adapt to most environment, no infrastructure needed.
3. Based on open-source hardware and software, which offers a relatively cheap platform for development and integration
4. The possibilities are endless.



THANK YOU

Thanks to Chang-Hor Kuo, Yin-Tao Chang, Yi-Chih Liu, Ting-Wei Hsu,
Bo-Ying Chen, Wei-Yu Lin, Ping-Yuan Huang

E-mail: lee.ty@nsrrc.org.tw