KUKA youBot obstacle avoidance

using Hokuyo scanning range finder + modified Tangent Bug algorithm

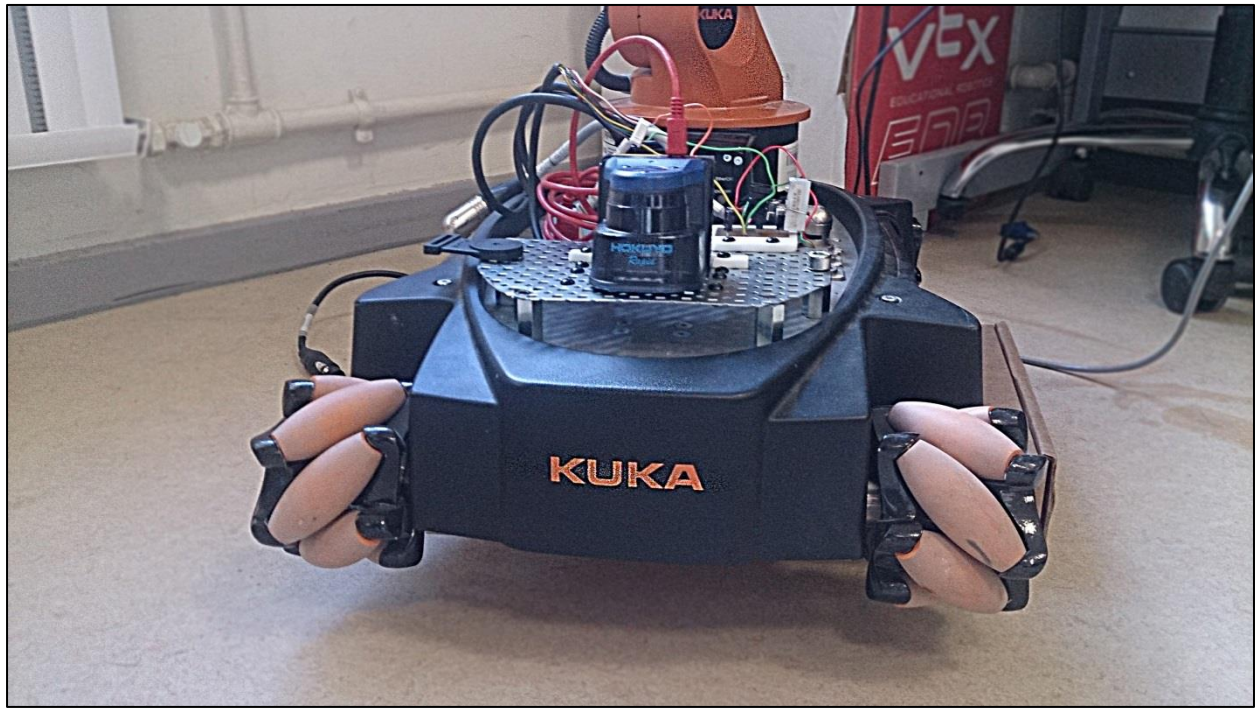
Vladislav Ostankovich
and
Aktore Barlybayev

Supervisor: Dr. Almas Shintemirov

NAZARBAYEV UNIVERSITY

Outline:

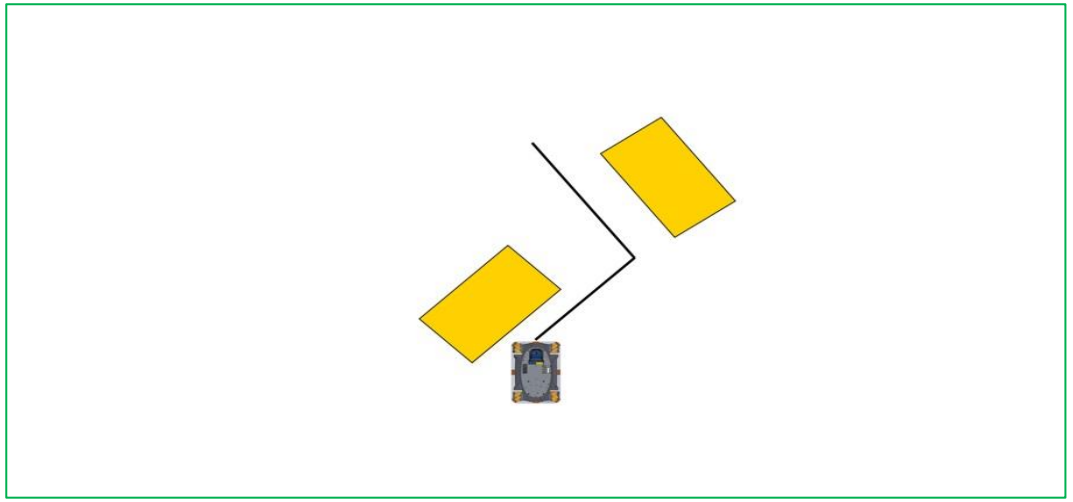
- Project task
- Joypad
- KUKA youBot
- Hokuyo
- Electric Circuit
- CAD design
- Algorithm implementation
- Testing

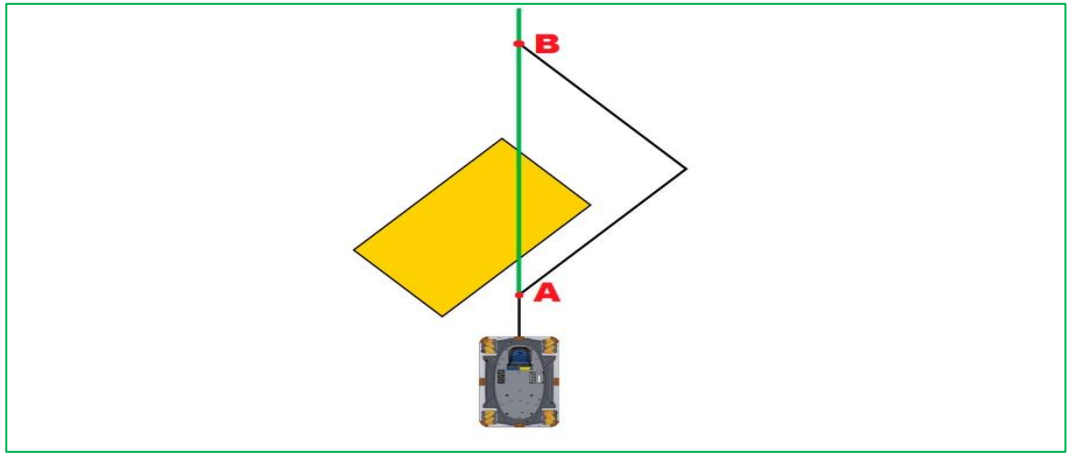


NAZARBAYEV UNIVERSITY

Kuka youBot and obstacle avoidance

- Controlled robot
    - JoyStick control
    - Keyboard control
- Autonoumous system
    - Avoid rectangular shapes
    - Different angles and positions





Applications

Remote control with obstacle avoidance algorithms is used in:

- Military robotics
- Space robotics
- Industrial robotics
- Human interactive robots

KUKA youBot Omni-Directional Mobile Platform

- Dimensions:
  - length: 580 mm
  - width: 380 mm
  - height: 140 mm
  - max speed: 0.8 m/s
- On-board PC
- Mini ITX PC-Board with embedded CPU, 2 GB RAM, 32 GB SSD Flash, USB
- Ports : 6 x USB 2.0, 1 x VGA, 2 x LAN
- Power supply

JoyStick – Defender Cobra M5

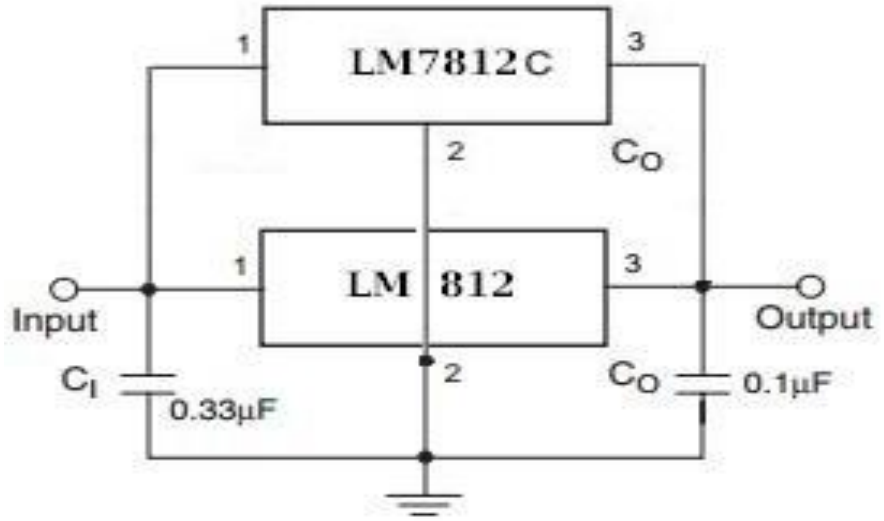
- Physical specifications:
  - Weight: 1.65 KG
  - Cord length: 1.2 m
- Windows XP/Vista/7, Linux Ubuntu
- 23 programmable buttons including 2 trigger ClusterFire™ (7 physical buttons + 2 pull)
- USB 2.0
- Power supply: 5V from USB-connection

Hokuyo UBG-04LX-F01 scanning range finder

- Supply voltage: 12V
- Measurement distance: 4m
- Field of view: 240°
- Pitch angle: 0.36° (682 steps total)
- Scan time: 28 ms/scan
- Interface RS-232C / USB

DC/DC converter for Hokuyo Lidar

- Voltage regulators
- 2 Capacitors
- Connection wires
- Heat sinks
- Thermal paste
- Board (Altium design)

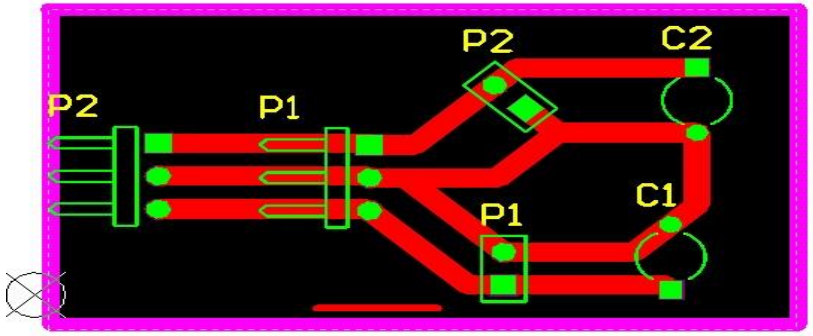


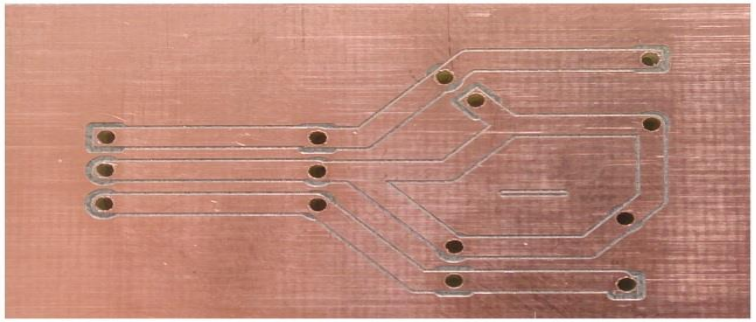
http://www.reuk.co.uk/24V-12V-DC-DC-Converter.htm

NAZARBAYEV UNIVERSITY

DC/DC converter board design/print

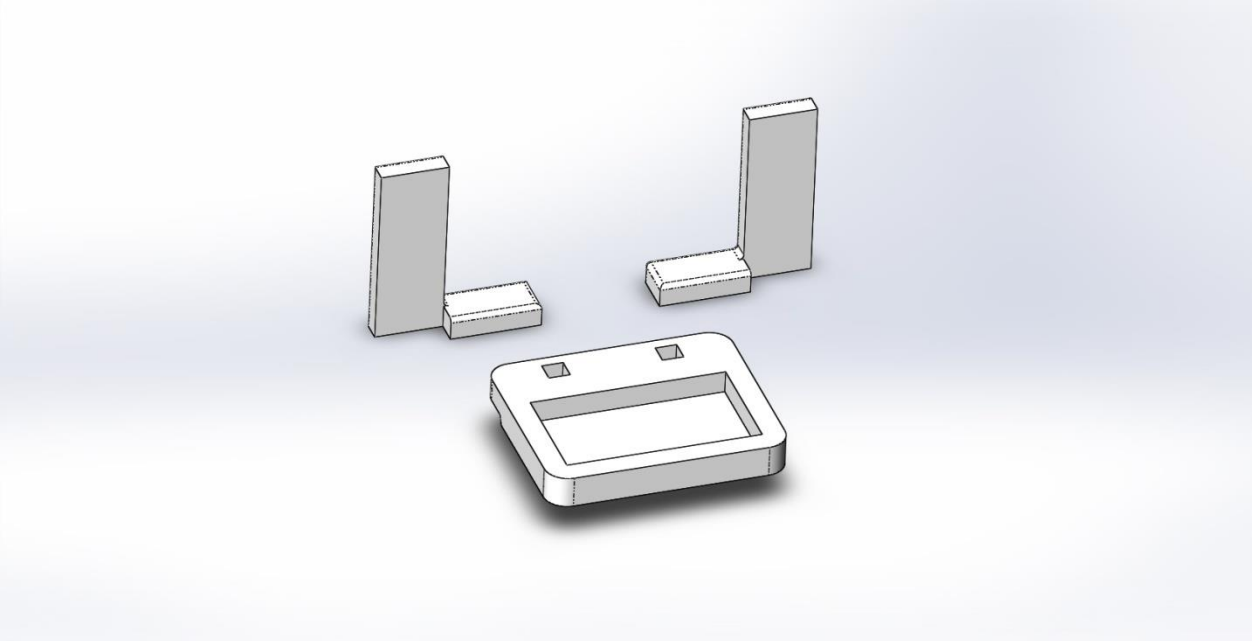
- Altium design
- Board print
- Soldering

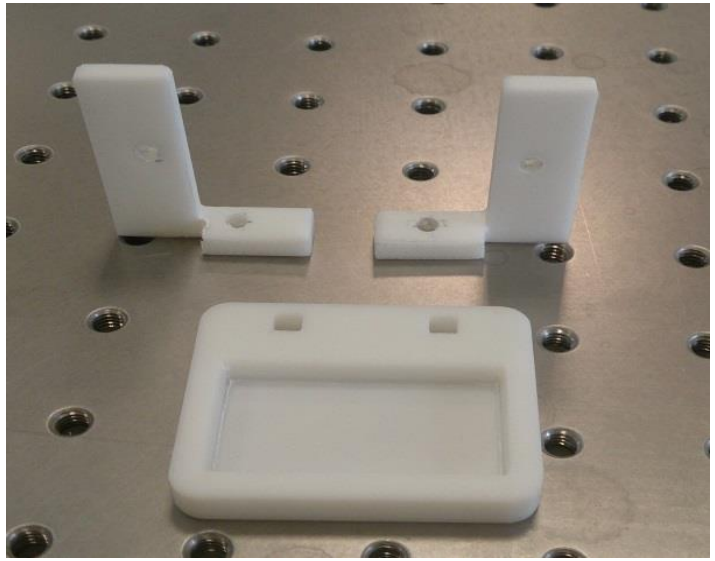




NAZARBAYEV UNIVERSITY

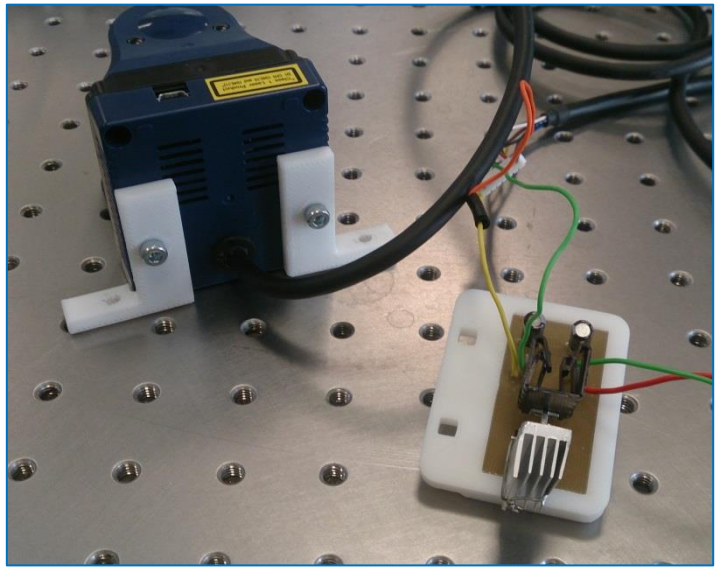
3D design – Hokuyo and circuit holder





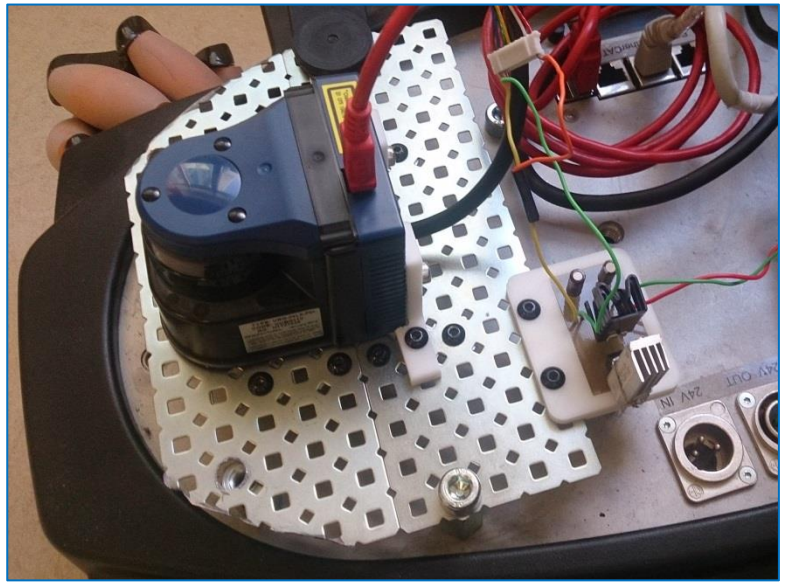
NAZARBAYEV UNIVERSITY

3D design – Hokuyo and circuit holder



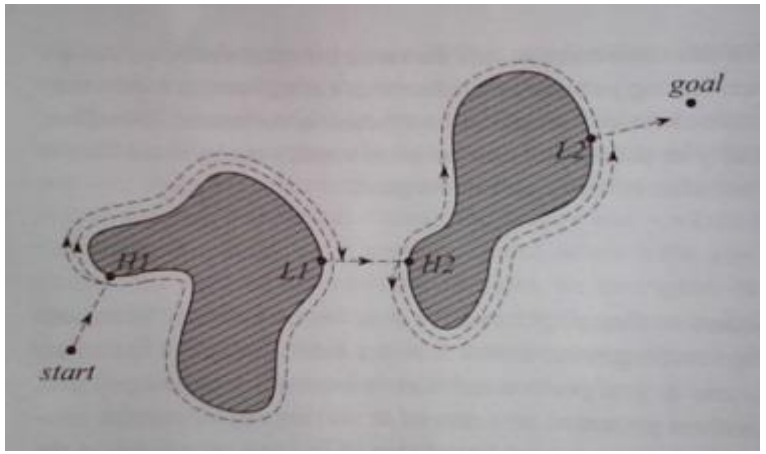
Hokuyo Rapid holders and DC/DC buck converter circuit holder

On board

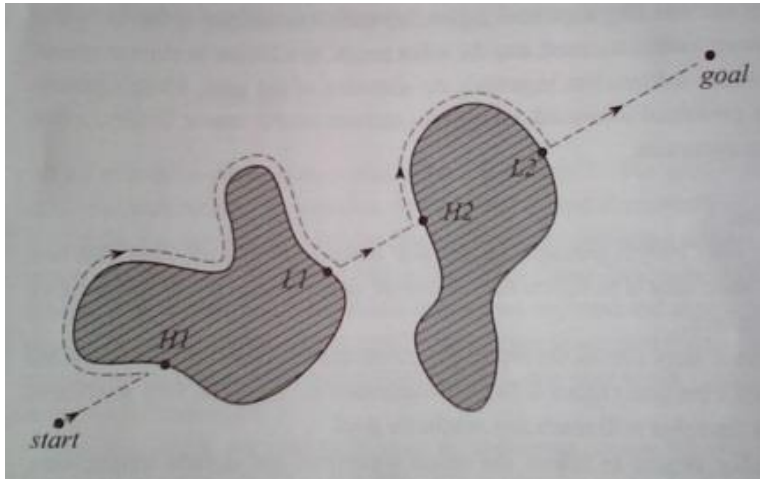


Obstacle avoidance algorithms

- Bug algorithms
  - Bug 1
  - Bug 2
  - Tangent Bug – most optimal / bug
- Vector field algorithm
- Bubble band technique
- Dynamic window approaches
- The Schlegel approach
- Nearness diagram
- Gradient method



"Intro to Autonomous Mobile Robots"

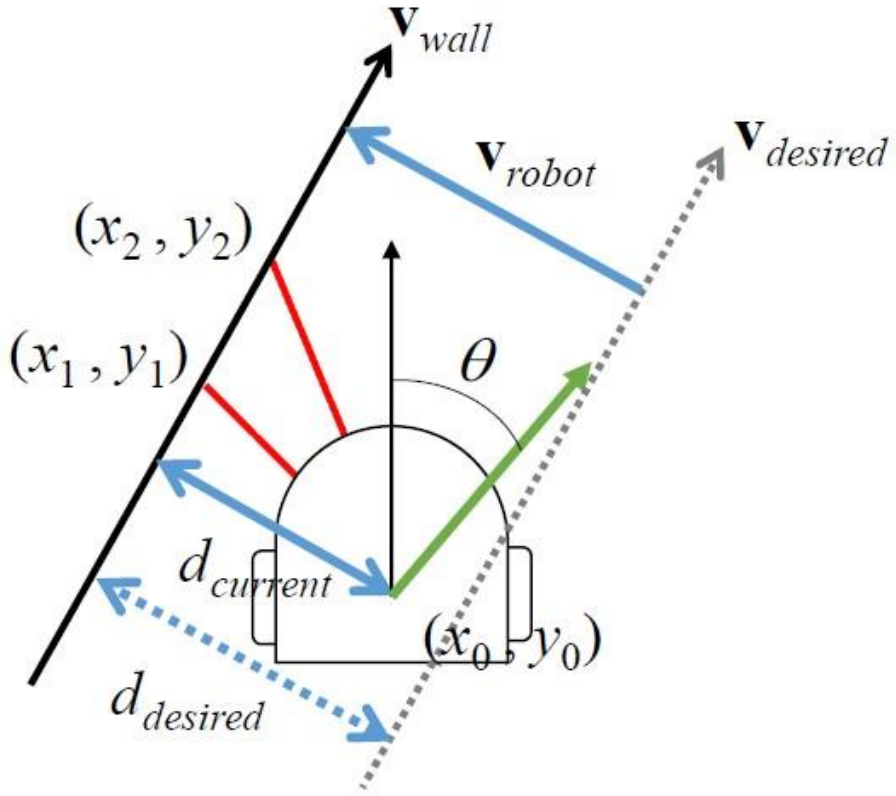


Tangent Bug obstacle avoidance algorithm

Three States:

- Avoiding obstacle (wall-following)
- Transitioning from obstacle avoidance to going to the goal
- Going to the goal

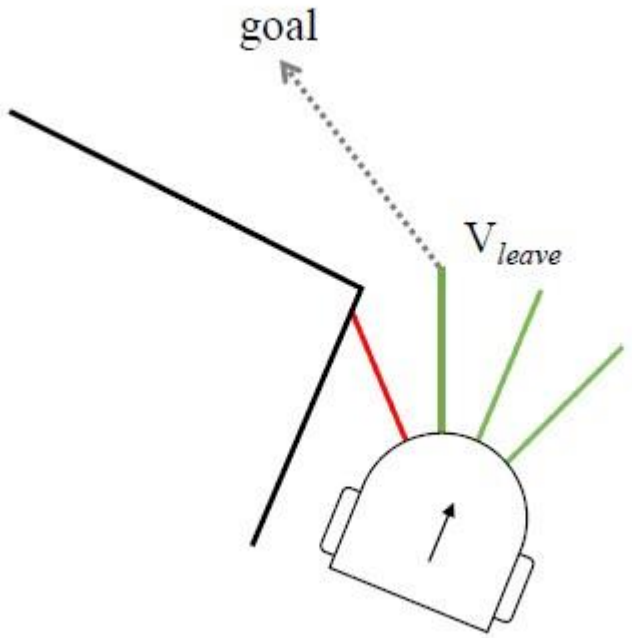
Tangent Bug obstacle avoidance algorithm: wall following (step 1)



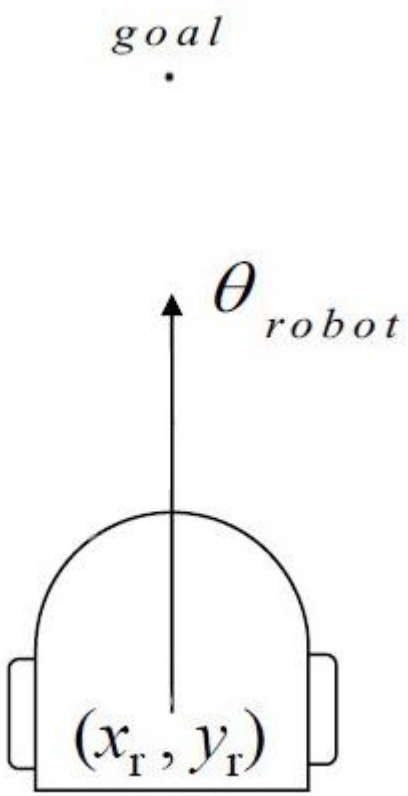
Two closest sensor values to the robot estimate the closest wall:

$$V_{wall} = \begin{pmatrix} x_2 - x_1 \\ y_2 - y_1 \end{pmatrix}$$

Tangent Bug obstacle avoidance algorithm: transition (step 2)



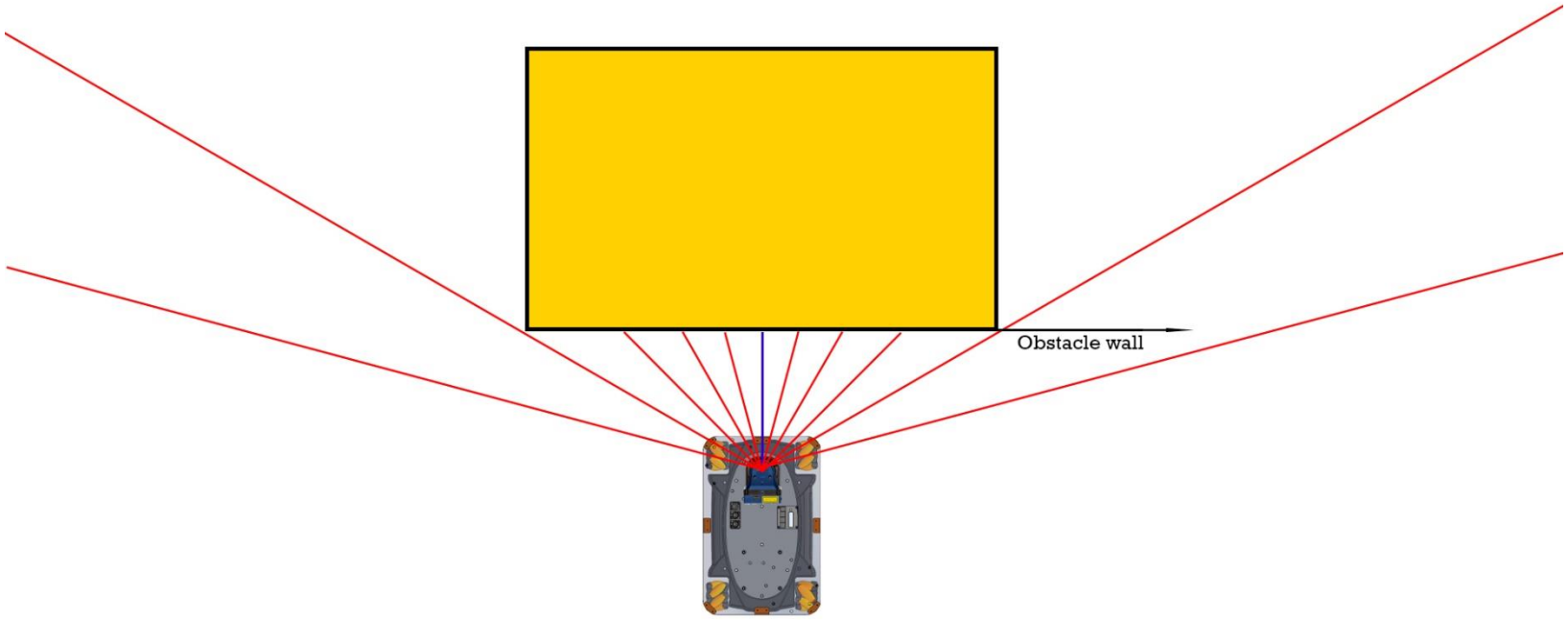
Tangent Bug obstacle avoidance algorithm: going to goal (step 3)



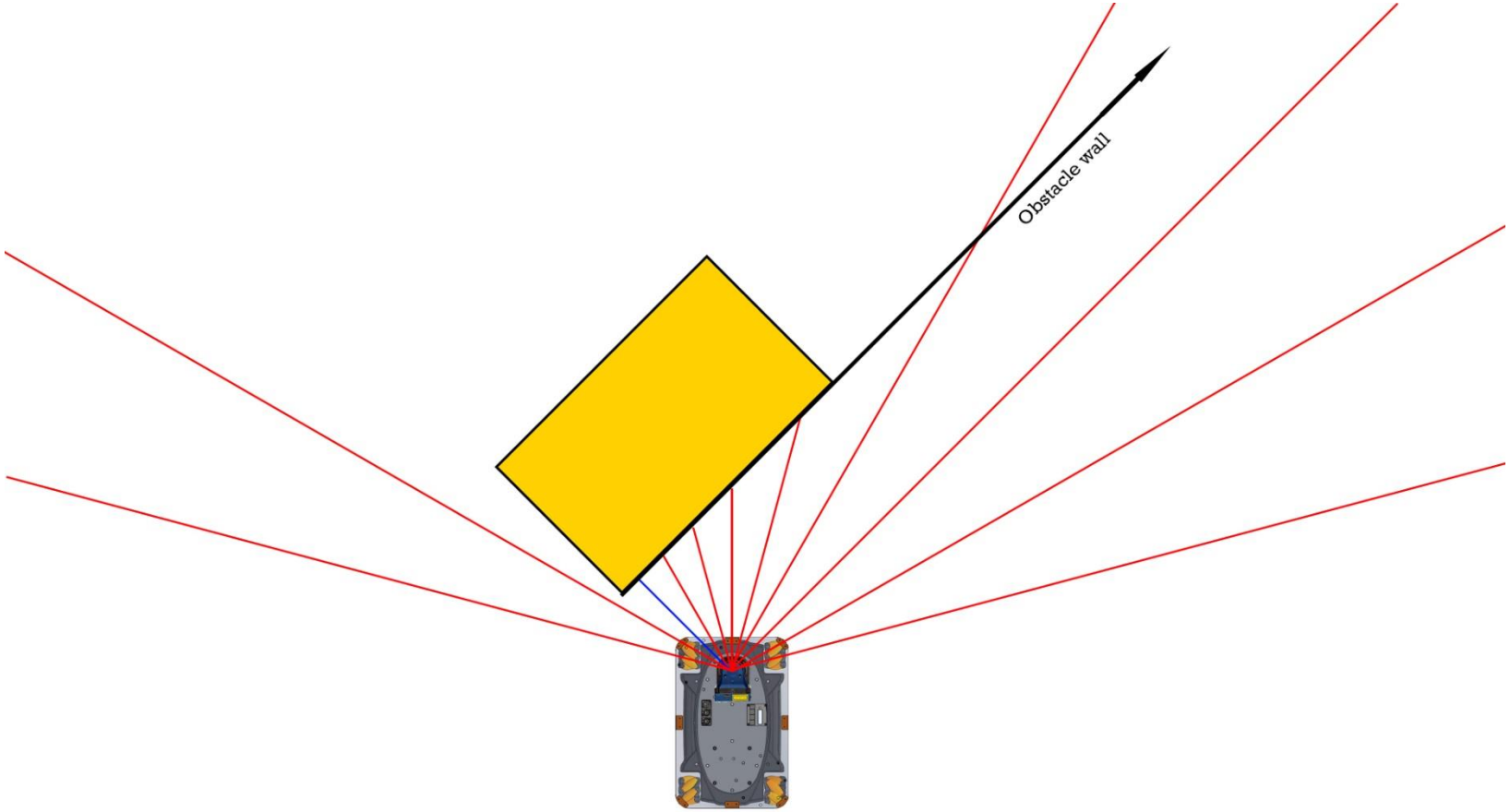
Modified TangentBug algorithm

- Joypad defines desired trajectory line
- ~~User defined goal point~~ → Robot defined goal point
- One closest sensor value estimates the obstacle wall
- Robot has to return to initial joypad-defined line

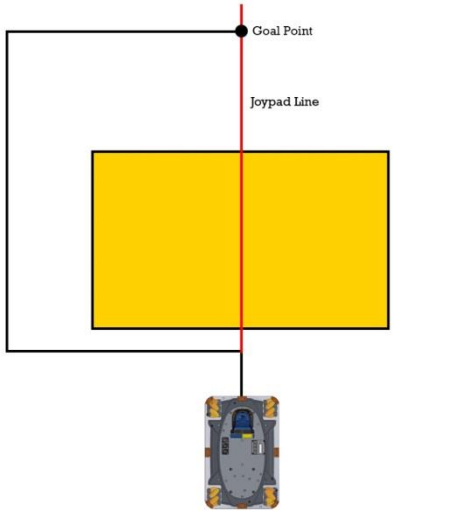
Modified TangentBug algorithm: obstacle wall definition

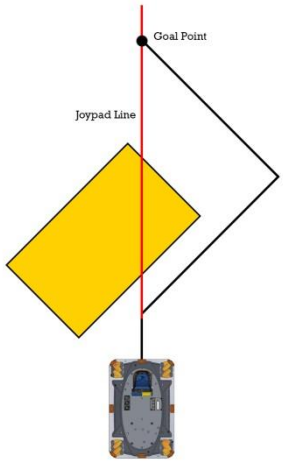


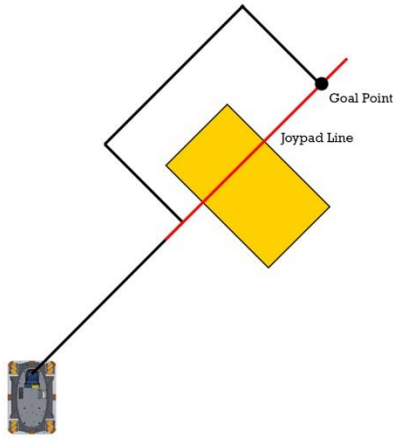
Modified TangentBug algorithm: obstacle wall definition

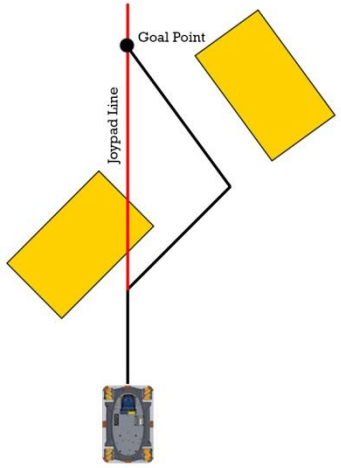


Modified TangentBug algorithm: trajectories

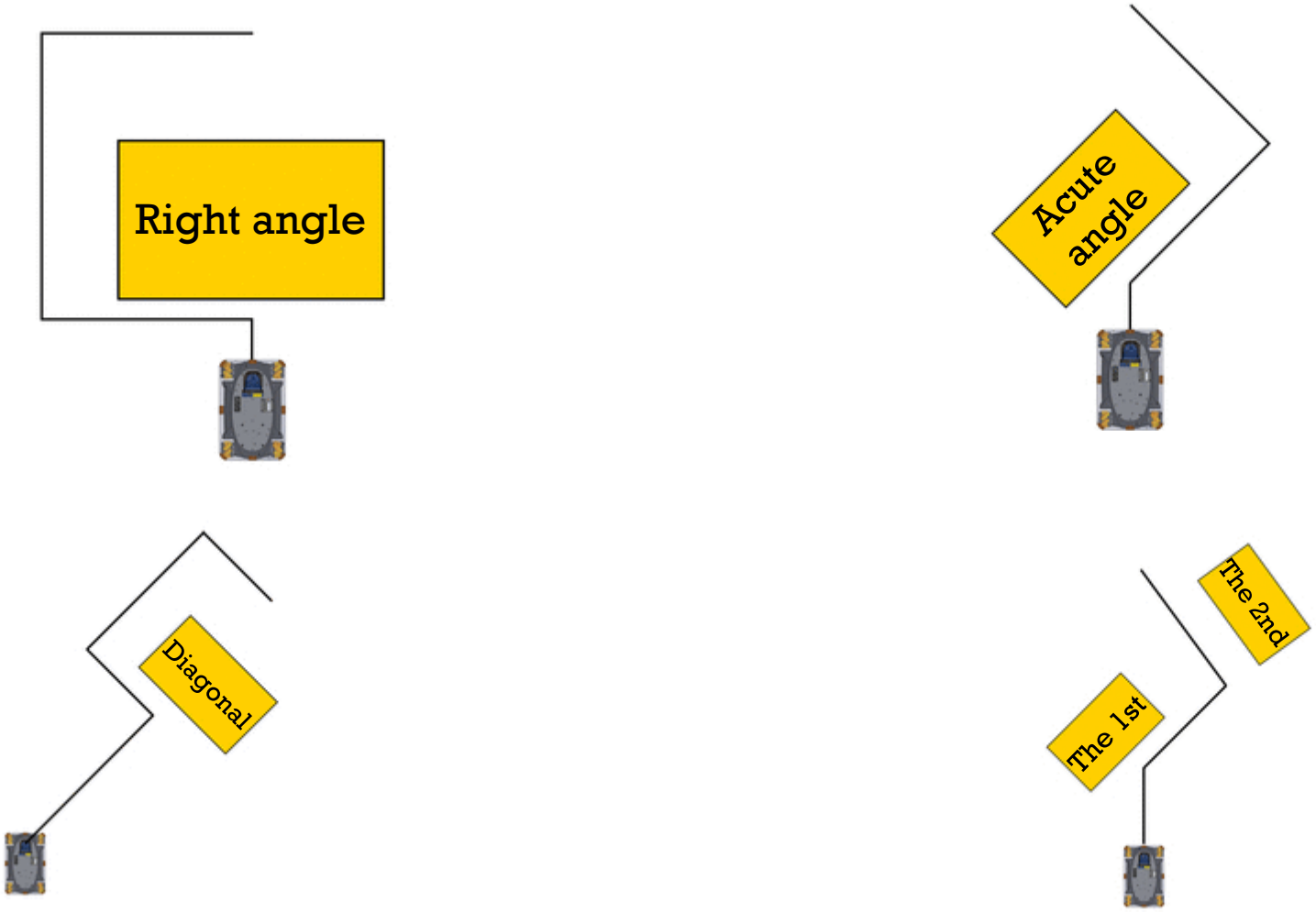








Modified TangentBug algorithm: trajectories



Pseudocode

1: **while** True **do**
2: **repeat**
3: Get scan
4: **until**
- The closest point less than 300mm is detected

5: Parameters calculation (angle, [x y] coordinates)
6: Estimation of tangent and three velocities (longitudinal, transversal and rotational)
7: Begin wall-following
8: **repeat**
9: **repeat**
10: Continuously update [x y] coordinates of displacement from initial point
11: **until**
- Sensor detects no wall **or**
- Robot reaches goal point

12: Estimation of perpendicular and three velocities
13: **until**
- Robot reaches goal point

14: **end while**

Project Testing: demonstration video

Further improvements

- Introducing obstacles with different shapes
- Implementing other algorithms for obstacle avoidance

Summary and results

- Control algorithm
- Tangent Bug obstacle avoiding algorithm
- Complementary tasks:
  - Design and print 3D models of holders
  - Design and print buck converter circuit

- Programming (C++)
- Design
- Embedded system

References:

- Choset, H.M. 2004. "Principles of Robot Motion: Theory, Algorithms, and Implementation", pp.17-38
- Kamon, I., Rimon, E., Rivlin, E. 1998. "TangentBug: A Range-Sensor-Based Navigation Algorithm", *The International Journal of Robotics Research*, vol.17, no.9, pp.934-953
- "KUKA youBot User Manual ", 2012
- Meyer, B., Shin, J. 2013. "Control and obstacle avoidance"
- Siegwart, R., Nourbakhsh, I.R., Scaramuzza, D. 2011. "Introduction to Autonomous Mobile Robots", 2nd ed.
- http://www.hokuyo-aut.jp/02sensor/07scanner/ubg_04lx_f01.html
- http://reuk.co.uk/24V-12V-DC-DC-Converter.htm
- http://www.youbot-store.com