

Basic of Robotics

Er. Arpit Joon
Doctorate Student
Poznan University of Technology



CONTENTS

01 What is Robotics?

02 Types of Robots

03 Kinematic Model

04 Previous Projects

An abstract graphic on the left side of the slide, composed of many overlapping triangles in various shades of blue, ranging from light sky blue to dark navy blue, creating a low-poly, crystalline effect.

What is Robotics?

Defination of a Robot: Machines controlled by computers that are used to perform jobs automatically. [1]

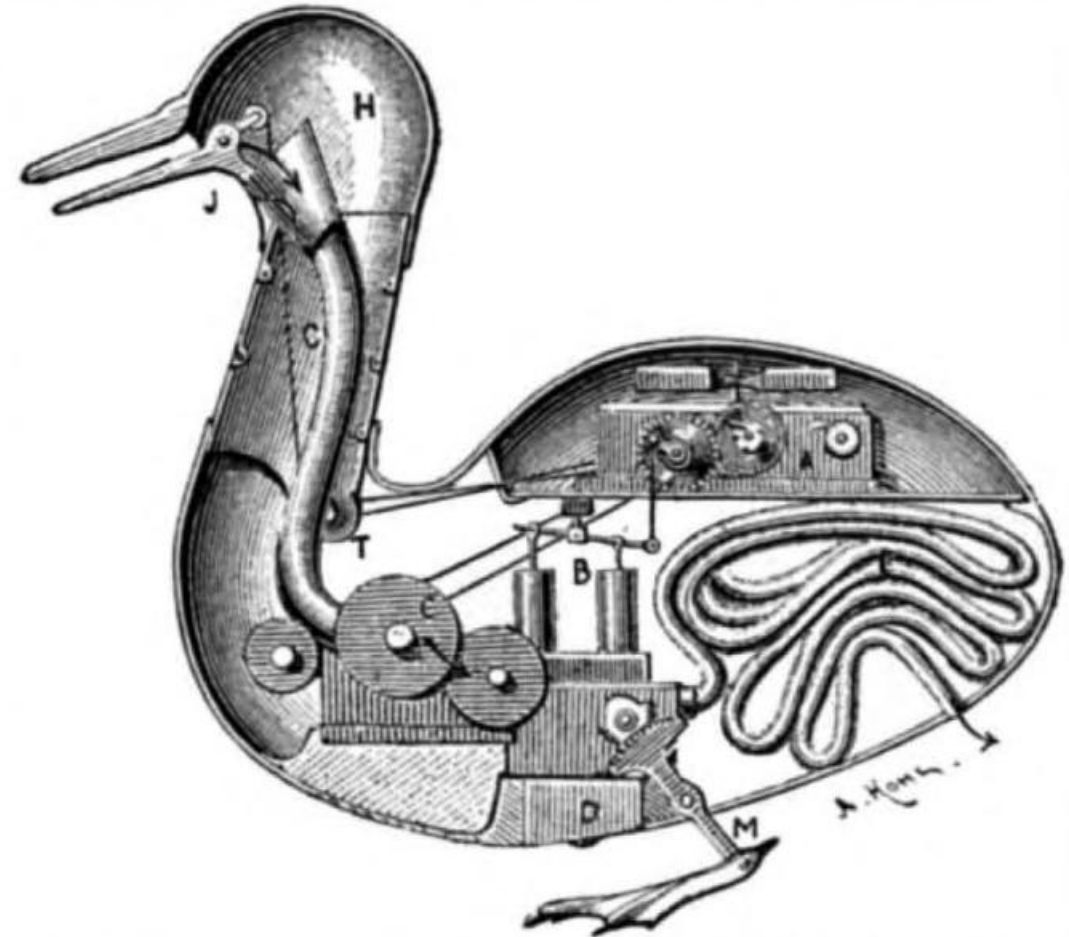
Defination of Robotics: the science of making and using robots. [1]

[1].Cambridge Dictionary. Accessed on 22-10-2023.
<https://dictionary.cambridge.org/dictionary/english/robotics>.

History of Robots

Year:1739

“Digesting Duck” was created by Jacques de Vaucanson in 1739 [2].



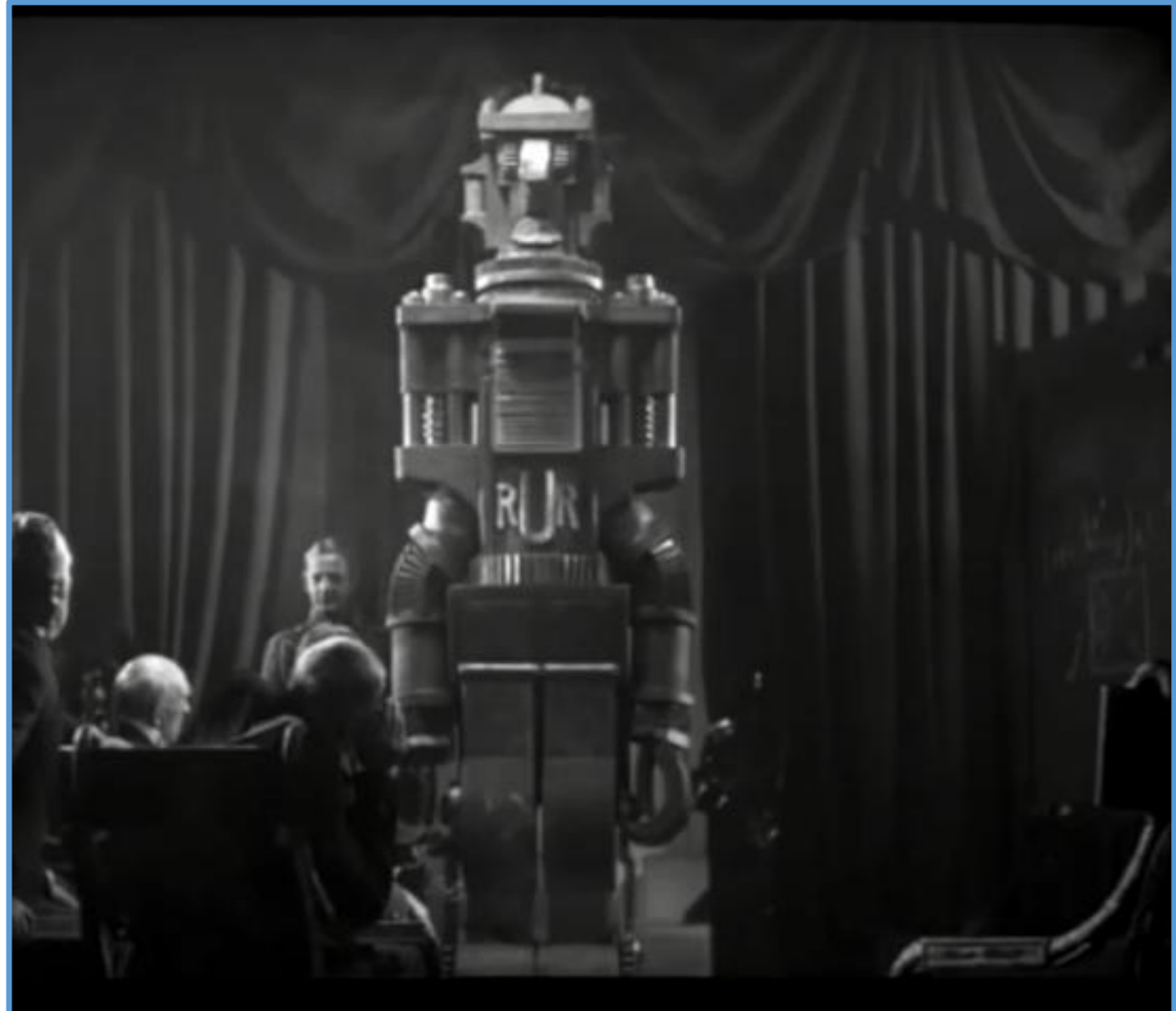
INTERIOR OF VAUCANSON'S AUTOMATIC DUCK.

A, clockwork; *B*, pump; *C*, mill for grinding grain; *F*, intestinal tube;
J, bill; *H*, head; *M*, feet.

History of Robots

Year: 1921

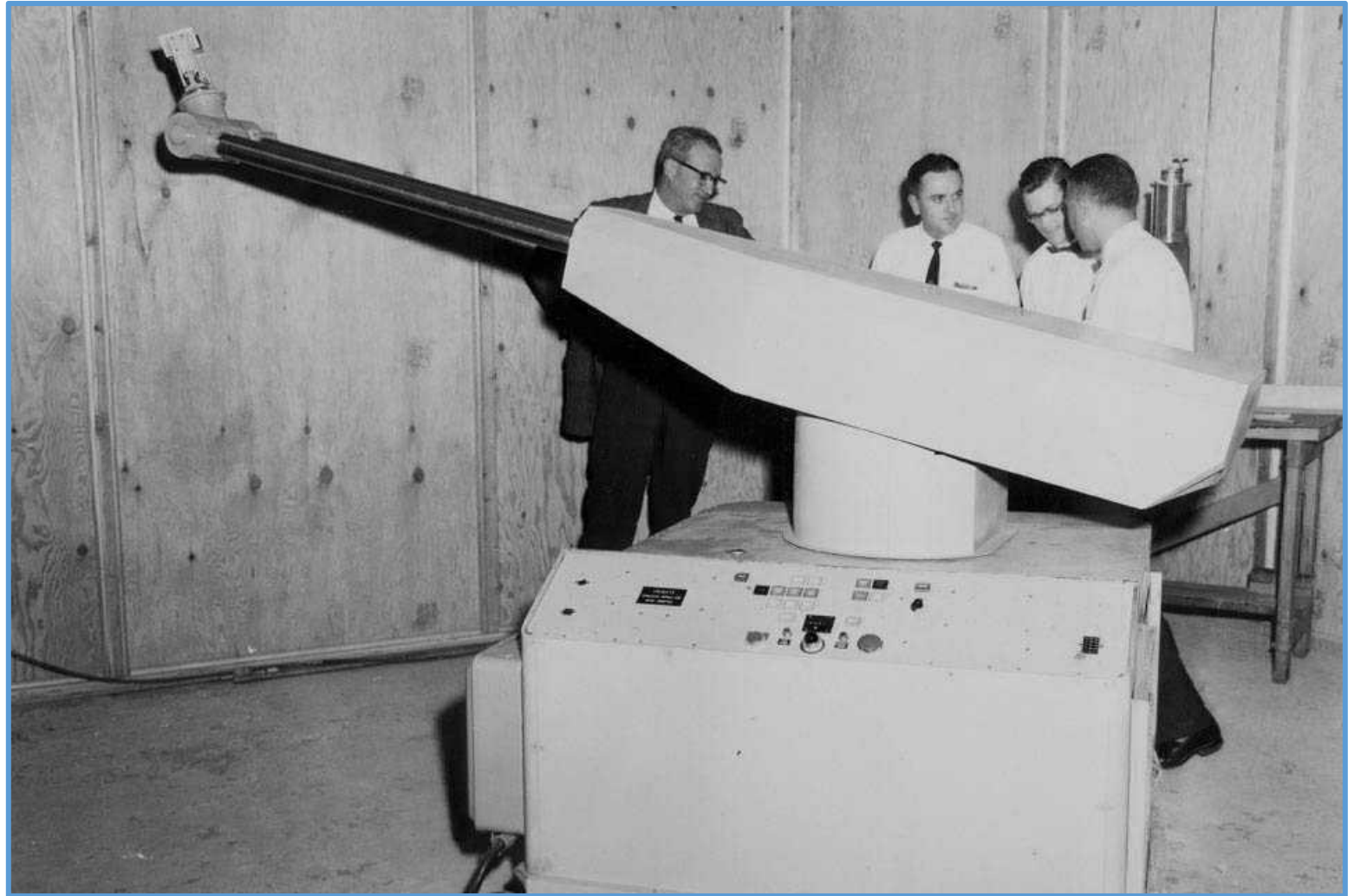
In his play “R.U.R.: Rossum’s Universal Robots,” the Czech writer Karel Čapek tells the tale of a factory in which thousands of synthetic humanoids have been created [2].



History of Robots

Year: 1959

First robotic arm is installed on a factory floor [2].



An abstract graphic on the left side of the slide, consisting of a series of overlapping triangles in various shades of blue, ranging from light sky blue to dark navy blue, creating a low-poly, crystalline effect.

Robots In the Agriculture Industry

Seeding and Spraying Robots

Drones for seeding and spraying are still a new thing, but they are commercially available [3].



[3]. Agricultural Robots: A revolutionary tool for farmers worldwide. <https://howtorobot.com/expert-insight/agricultural-robots>. Accessed on 22-10-2023.

Fruit and Vegetable Harvesting Robots

Fruit and vegetable harvesting robots are still in the prototype stages. They show great promise [3].



[3]. Agricultural Robots: A revolutionary tool for farmers worldwide. <https://howtorobot.com/expert-insight/agricultural-robots>. Accessed on 22-10-2023.

Aerial Imaging Robots

Using aerial drones to inspect crops from the air is well established. Aerial imaging produces valuable insights into crop health and soil conditions [3].



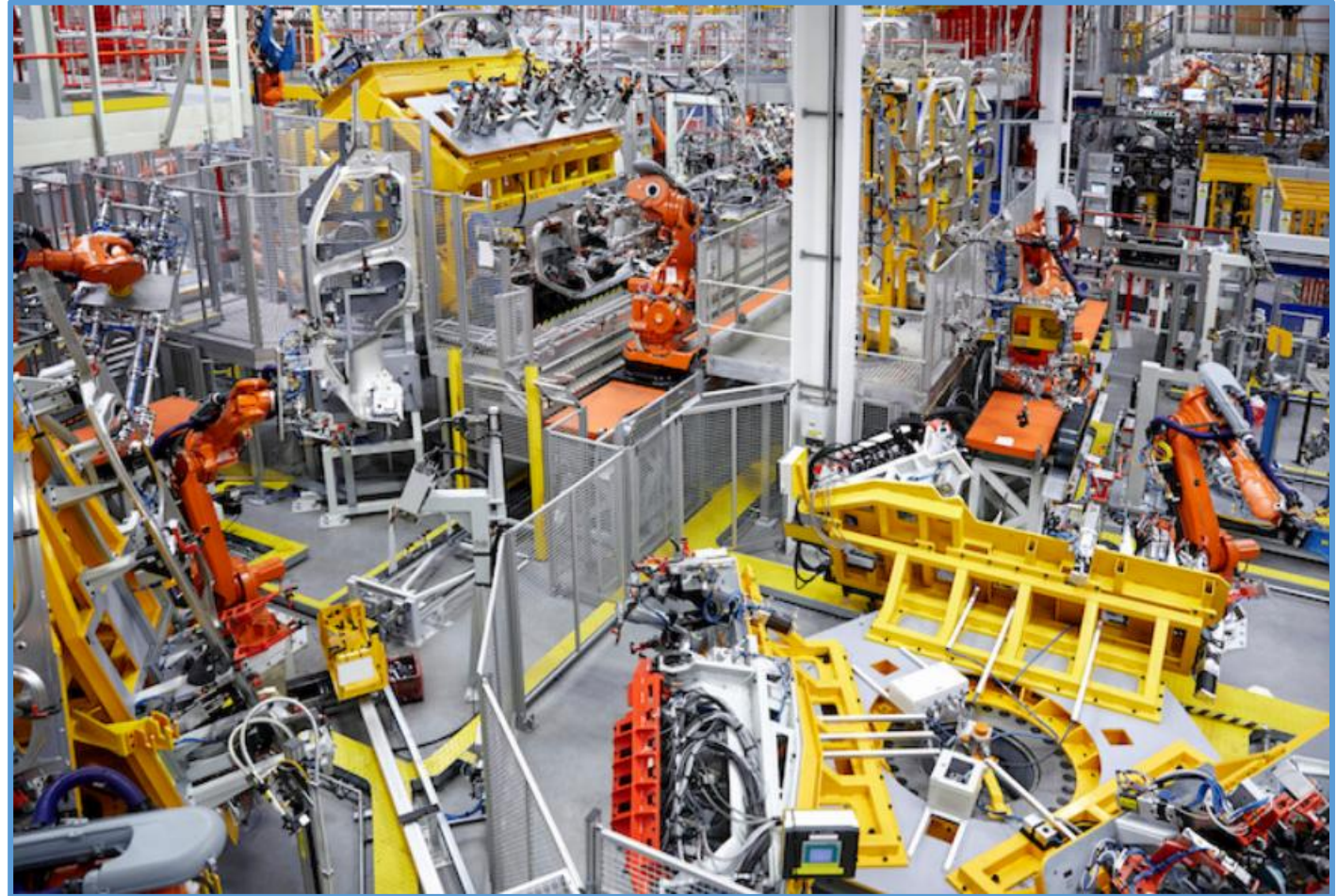
[3]. Agricultural Robots: A revolutionary tool for farmers worldwide. <https://howtorobot.com/expert-insight/agricultural-robots>. Accessed 22 Oct. 2023.

An abstract graphic on the left side of the slide, consisting of a series of overlapping triangles in various shades of blue, ranging from light sky blue to dark navy blue, creating a low-poly, crystalline effect.

Robots In Industries

Robots In Manufacturing

From factories to farms, there are a growing number of places and situations where industrial robots can be adapted to perform [4].



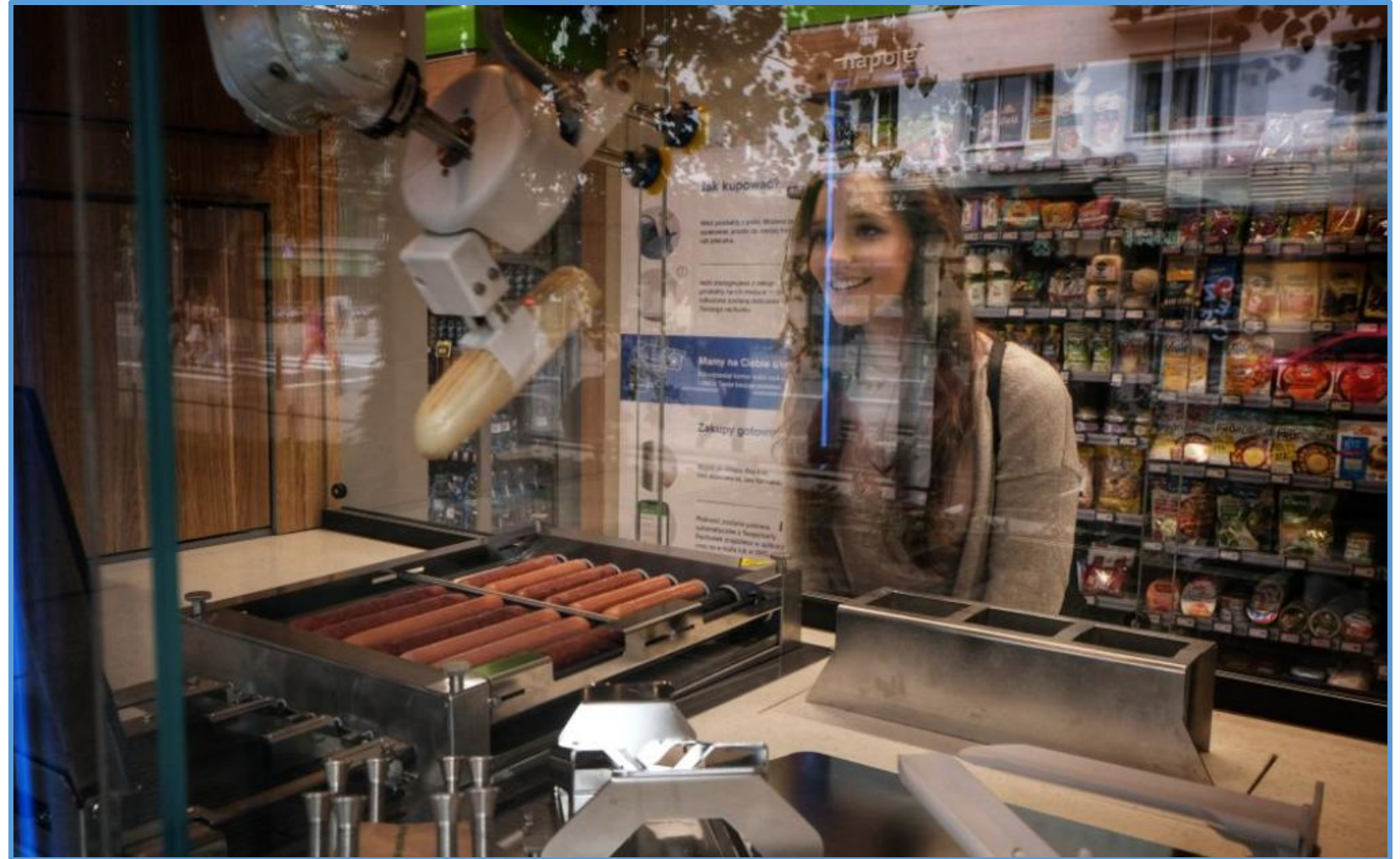
[4]. The good, the bad, and the ugly of Industrial Robots In Manufacturing. Industrial Robots: Manufacturing Examples. <https://www.autodesk.com/design-make/articles/industrial-robotics>. Accessed on 22-10-2023.

An abstract graphic on the left side of the slide, consisting of a series of overlapping triangles in various shades of blue, ranging from light sky blue to dark navy blue, creating a low-poly, crystalline effect.

Robots In Food Industries

Poland introduces hot dog-making robot

The brainchild of convenience store chain Żabka and Kraków tech firm VeloxAlpha SA, the one-armed sausage server called Robbie can complete its task in three simple steps [5].



[5]. Robbie, Zabka's 24/7 Hot Dog-Making Robot, Debuts in Warsaw Convenience Store https://www.franchising.com/articles/robbie_zabkas_247_hot_dogmaking_robot_debuts_in_warsaw_convenience_store.html. Accessed on 22-10-2023.

Poland introduces hot dog-making robot

First the robotic arm pulls a bun out of a closed drawer compartment and puts it on a hot grill to heat and toast it. Then it takes out the bun and slips it under a tube that squirts in your condiment of choice [5].



[5]. Robbie, Zabka's 24/7 Hot Dog-Making Robot, Debuts in Warsaw Convenience Store https://www.franchising.com/articles/robbie_zabkas_247_hot_dogmaking_robot_debuts_in_warsaw_convenience_store.html. Accessed on 22-10-2023.

Poland introduces hot dog-making robot

Next the robot places the bun on a little metal bed and carefully slides the hot dog into it. The arm uses suctions to pick up a paper sleeve for the hot dog and slips it onto the bun.



[5]. Robbie, Zabka's 24/7 Hot Dog-Making Robot, Debuts in Warsaw Convenience Store https://www.franchising.com/articles/robbie_zabkas_247_hot_dogmaking_robot_debuts_in_warsaw_convenience_store.html. Accessed on 22-10-2023.

Poland introduces hot dog-making robot

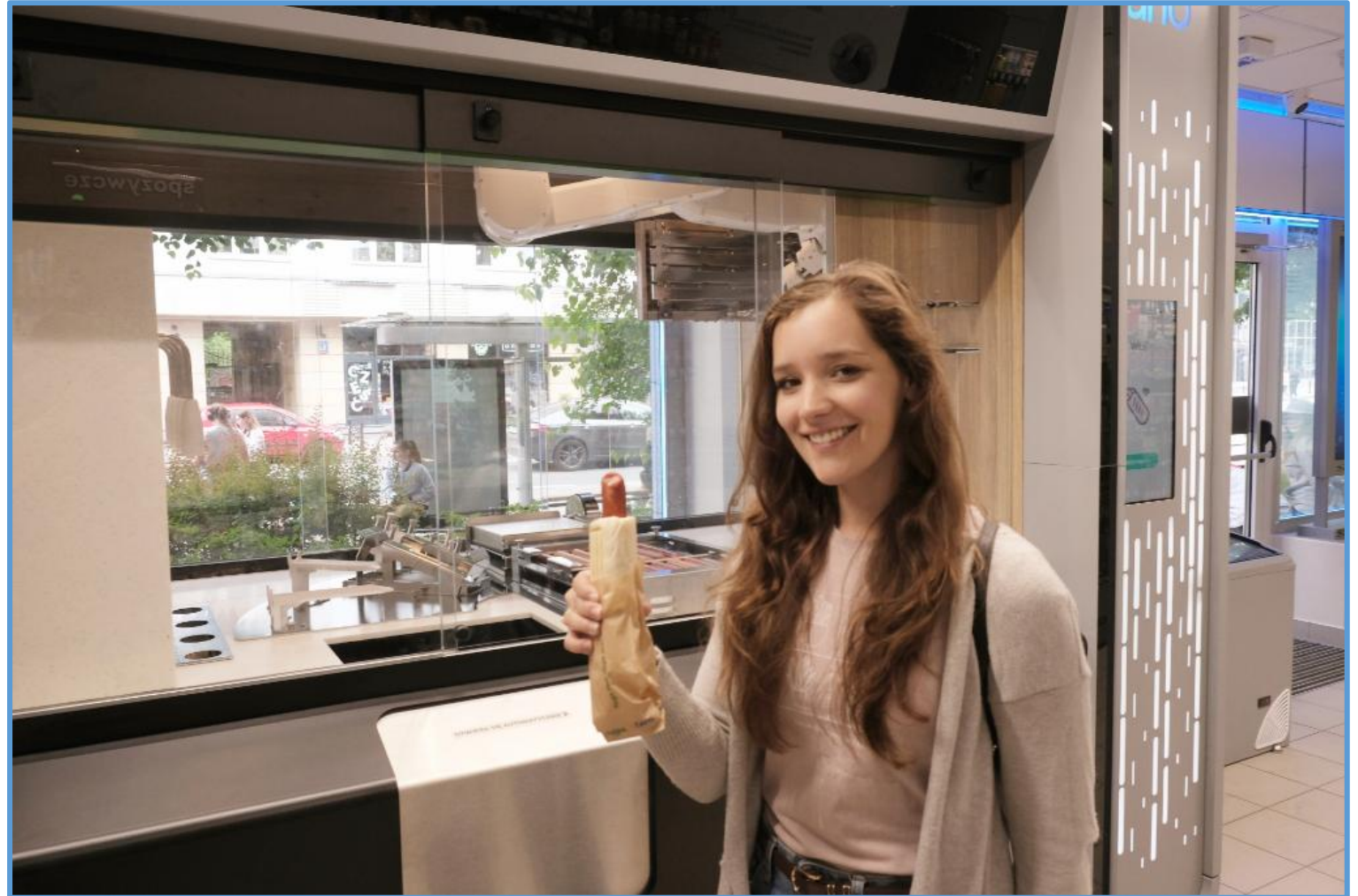
Lastly, the robot picks the hot dog up and places it into a stand up tray which lowers and the door slides open when the whole process is finished so you can safely take it out.



[5]. Robbie, Zabka's 24/7 Hot Dog-Making Robot, Debuts in Warsaw Convenience Store https://www.franchising.com/articles/robbie_zabkas_247_hot_dogmaking_robot_debuts_in_warsaw_convenience_store.html. Accessed on 22-10-2023. 17

Poland introduces hot dog-making robot

Happy customer: Monica said the experience was quick and efficient but regretted that the company has yet to introduce a vegan version [5].



[5]. Robbie, Zabka's 24/7 Hot Dog-Making Robot, Debuts in Warsaw Convenience Store https://www.franchising.com/articles/robbie_zabkas_247_hot_dogmaking_robot_debuts_in_warsaw_convenience_store.html. Accessed on 22-10-2023.

An abstract graphic on the left side of the slide, consisting of a series of overlapping triangles in various shades of blue, ranging from light sky blue to dark navy blue, creating a low-poly, crystalline effect.

What is a Mobile Robot?

A mobile robot is a machine controlled by software that use sensors and other technology to identify its surroundings and move around its environment [6].

[6]. Tech Target network. <https://www.techtarget.com/iotagenda/definition/mobile-robot-mobile-robotics>. Accessed on 22-10-2023.



Types of Mobile Robot

Car-like robot



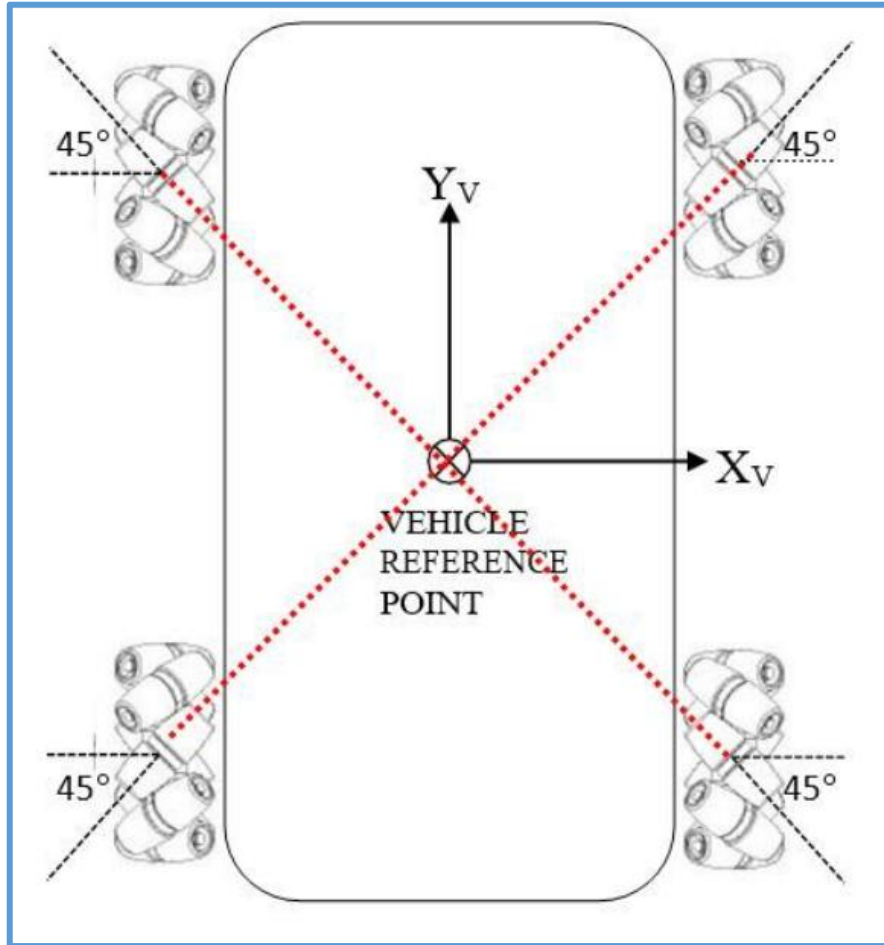
[7]



[7]

[7]. https://www.google.com/search?q=car+like+robot&sca_esv=575810318&tbm=isch&sxsrf=AM9HkK1Rh0-qfjWqqFrCohNdG_Zbph8q3g:1698073474637&source=lnms&sa=X&ved=2ahUKEwi9gIjSuIyCAxWNSmwGHelpBa8Q_AUoAXoECAEQAw&biw=1536&bih=695&dpr=1.25#imgsrc=MRESkn6Q-6WntM

Omni-Directional Mobile Robot with Mecanum Wheels



[8]



[8]

[8]. Force Vector Diagrams for an Omni-directional Mobile Robot. <https://automaticaddison.com/force-vector-diagrams-for-an-omni-directional-mobile-robot/>. Accessed on 22-10-2023.

Omni-Directional Mobile Robot with Mecanum Wheels



[9]

[9]. Robot shop. <https://www.robotshop.com/products/4wd-omni-directional-arduino-compatible-mobile-robot>. Accessed on 22-10-2023.



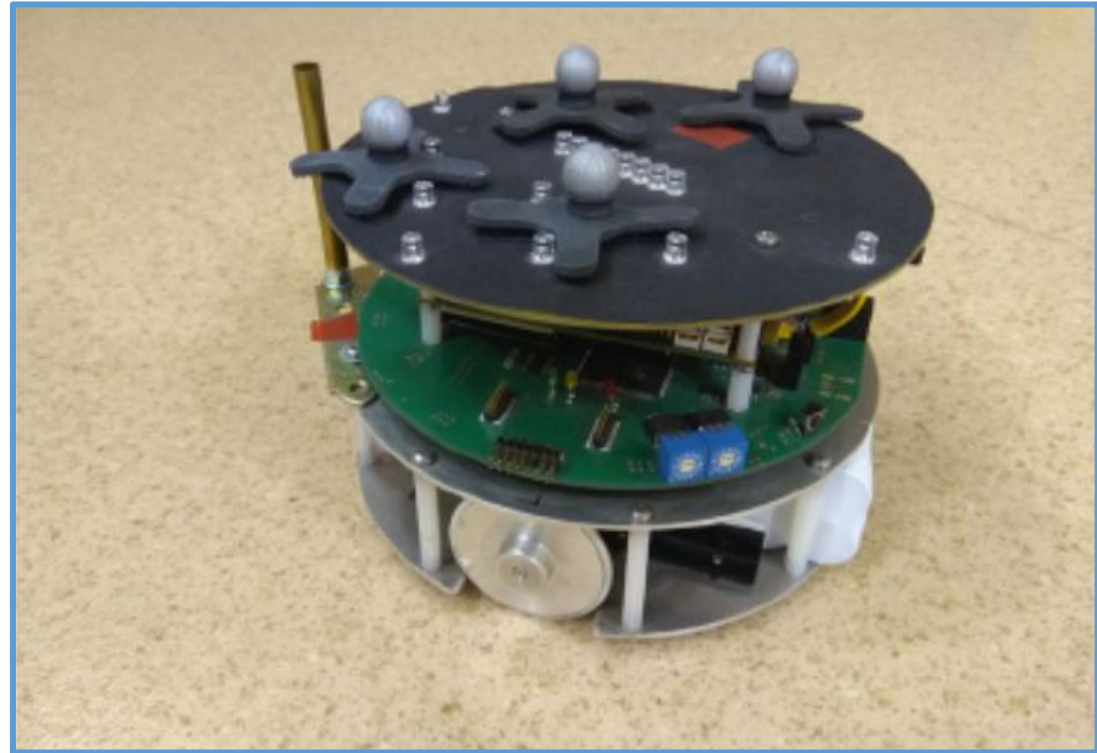
[10]

[10]. superdroidrobots. <https://www.superdroidrobots.com/store/usage/programmable-robots/product=1486>.

Unicycle-Like Mobile Robot / Two Wheeled Mobile Robot



[11]



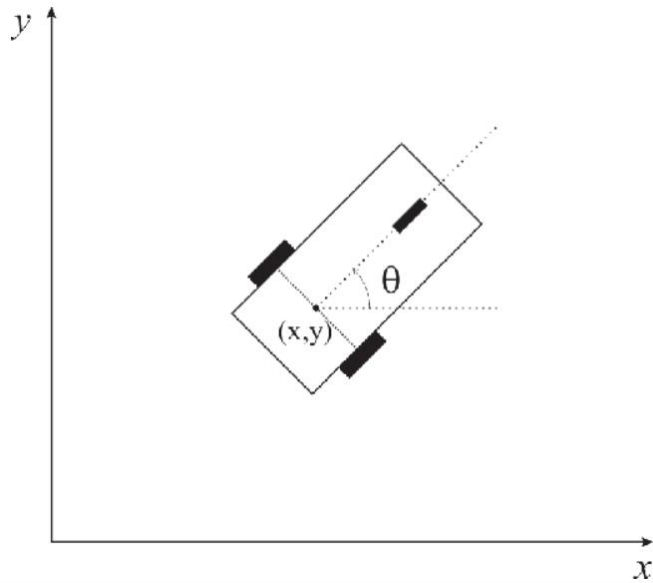
[12]

[11].Sun, Chung-Hsun et al. "Design of T-S fuzzy controller for two-wheeled mobile robot." Proceedings 2011 International Conference on System Science and Engineering (2011): 223-228.

[12]. Kowalczyk, Wojciech. (2019). Rapid Navigation Function Control for Two-Wheeled Mobile Robots. Journal of Intelligent & Robotic Systems. 93. 10.1007/s10846-018-0879-4.

Kinematic Model of Unicycle-Like Mobile Robot

The kinematic model of the mobile robot is written as:



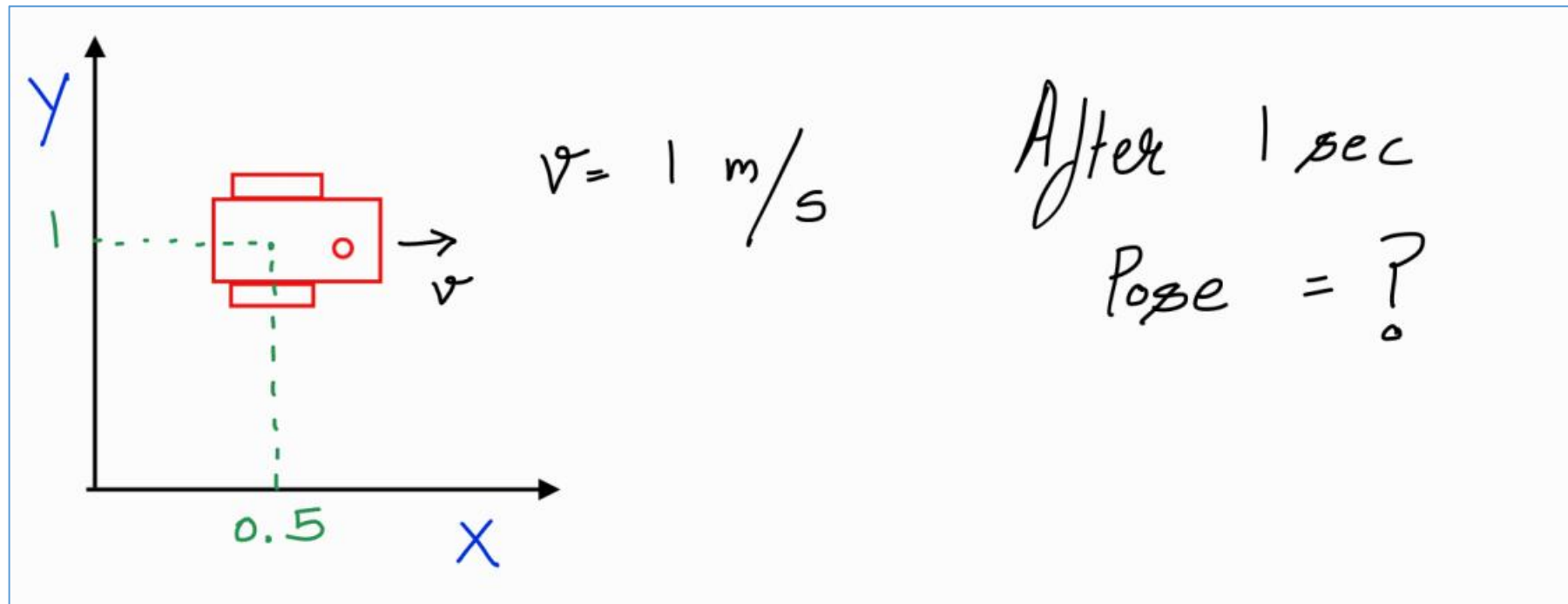
$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos \theta & 0 \\ \sin \theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix},$$

The robot pose is represented by vector $[x \ y \ \theta]^\top$. x, y, θ are the variables representing the pose of the robot in the global reference frame. The control vector is $[v \ \omega]^\top$ where v and ω are linear and angular velocity controls of the robot respectively.

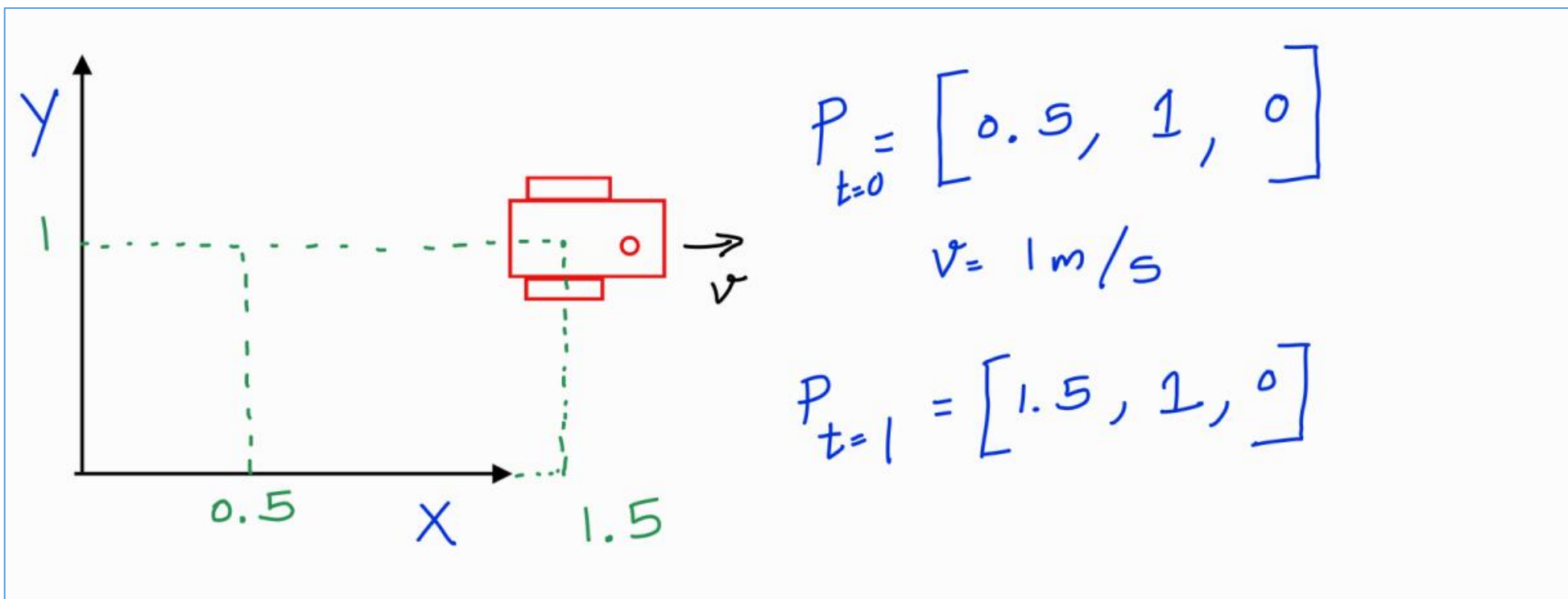
Linear velocity

Linear velocity, v , is defined as the rate of change of linear displacement, s , with respect to time, t , and for motion in a straight line:

$$v = \frac{s}{t}$$



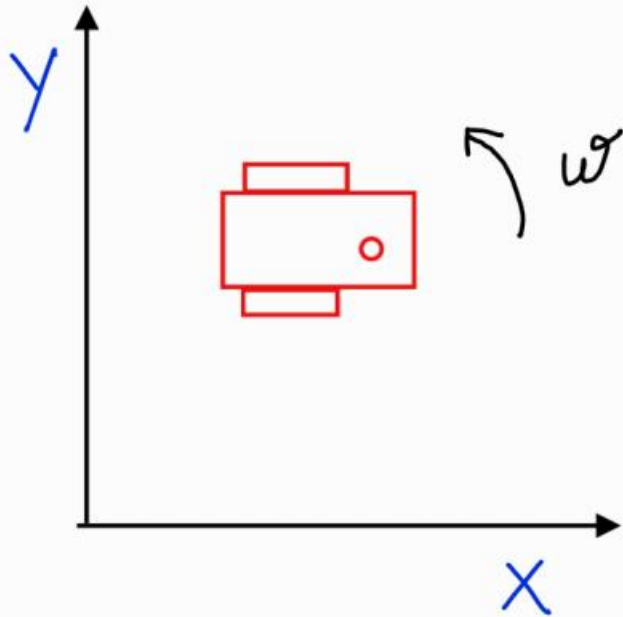
Linear velocity



Angular velocity

Angular velocity ω is defined as the rate of change of angular displacement, θ , with respect to time, t , :

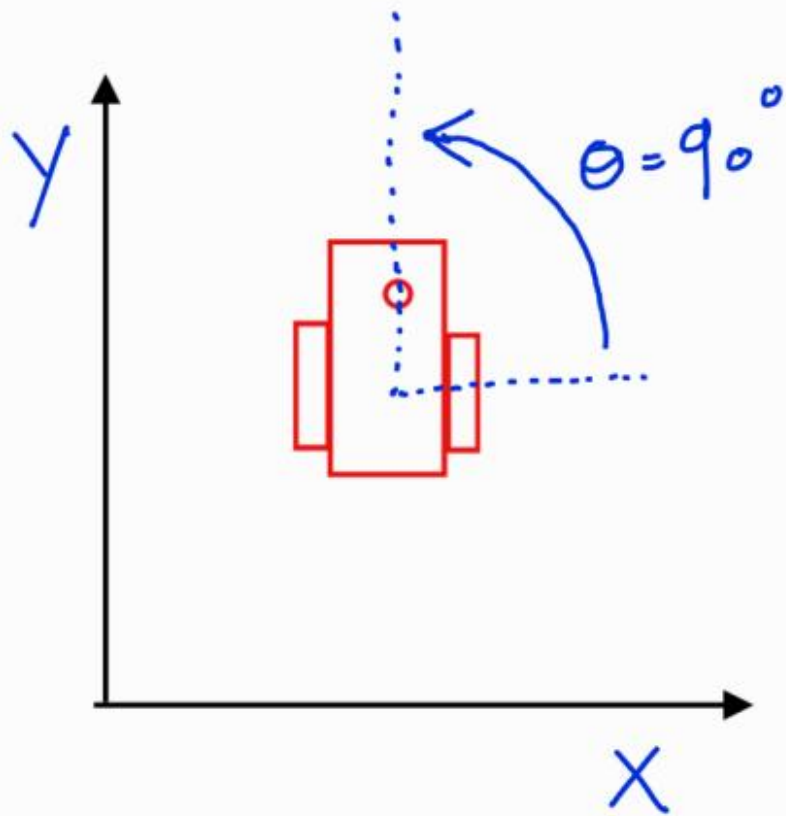
$$\omega = \frac{\theta}{t}$$



$$\varphi = 0$$

$$\omega = 1.57 \text{ rad/s}$$

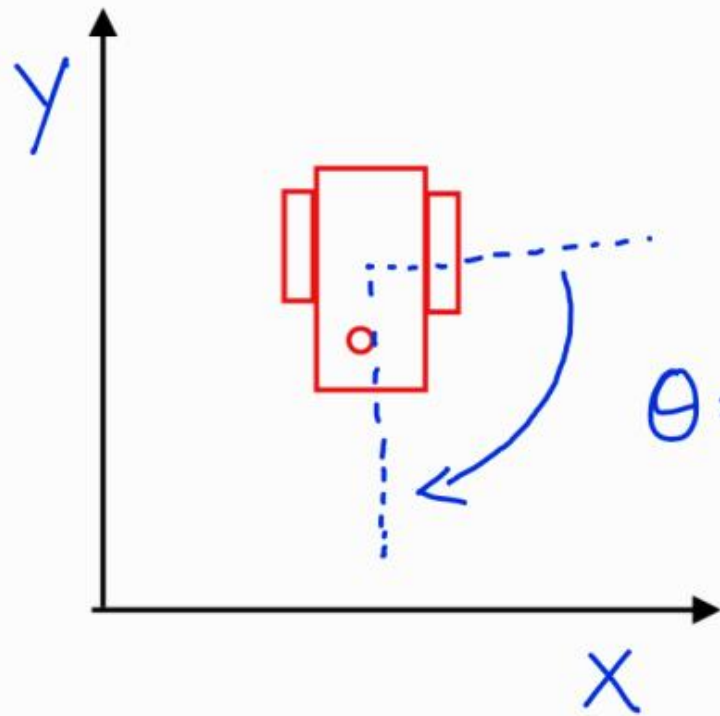
After 1 sec = ?



$$P_{t=0} = [0.5, 1, 0]$$

$$v = 0 \text{ m/s}, \omega = 1.57$$

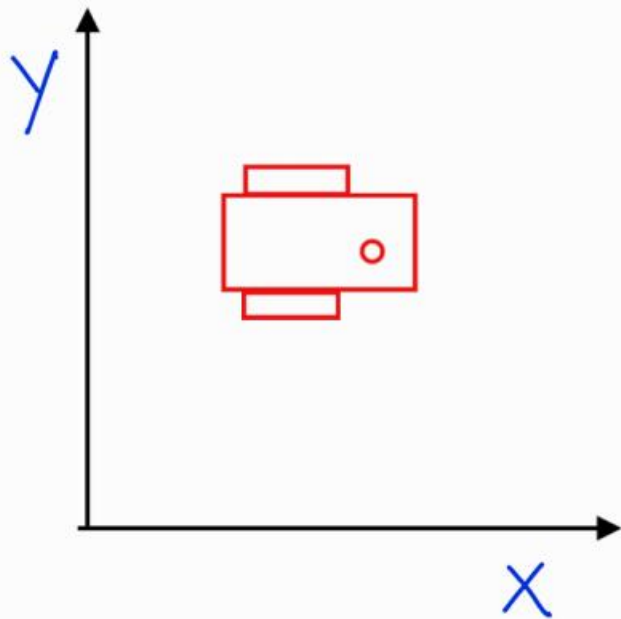
$$P_{t=1} = [0.5, 1, 1.57]$$



$$P_{t=0} = [0.5, 1, 0]$$

$$V = 0 \text{ m/s}, W = -1.57$$

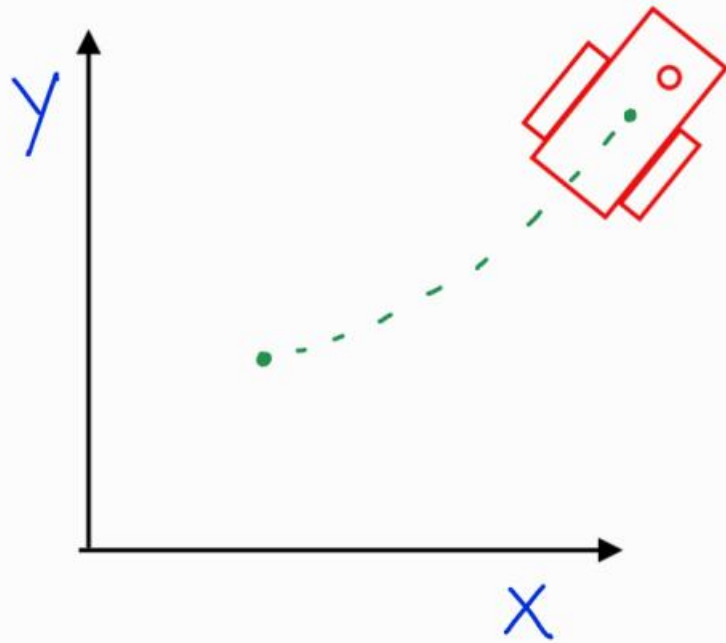
$$P_{t=1} = [0.5, 1, -1.57]$$



$$V = 0.1 \text{ m/s}$$

$$W = 0.1 \text{ m/s}$$

?



When the linear velocity and angular velocity will be given then the robot will form a curve or a circle.

PYTHON IDE

<https://code.visualstudio.com/>

The image shows the Visual Studio Code website on the left and a screenshot of the Visual Studio Code application interface on the right.

Visual Studio Code Website:

- Visual Studio Code Docs Updates Blog API Extensions FAQ Learn
- Search Docs Download
- Version 1.83 is now available! Read about the new features and fixes from September.
- Code editing. Redefined.
- Free. Built on open source. Runs everywhere.
- Download for Windows Stable Build
- Web, Insiders edition, or other platforms
- By using VS Code, you agree to its license and privacy statement.

Visual Studio Code Application Interface:

- File Edit Selection View Go Debug Terminal Help serviceWorker.js - create-react-app - Visual Studio Code - In...
- EXTENSIONS: MARKETPLACE
- @sort:installs
- Python 2019.6.24221 54.9M 4.5
Linting, Debugging (multi-threaded, ...
Microsoft Install
- GitLens — Git sup... 9.8.5 23.1M 5
Supercharge the Git capabilities built...
Eric Amodio Install
- C/C++ 0.24.0 23M 3.5
C/C++ IntelliSense, debugging, and ...
Microsoft Install
- ESLint 1.9.0 21.9M 4.5
Integrates ESLint JavaScript into VS ...
Dirk Baeumer Install
- Debugger for Ch... 4.11.6 20.6M 4
Debug your JavaScript code in the C...
Microsoft Install
- Language Supp... 0.47.0 18.7M 4.5
Java Linting, Intellisense, formatting, ...
Red Hat Install
- vscode-icons 8.8.0 17.2M 5
Icons for Visual Studio Code
VSCode Icons Team Install
- Vetur 0.21.1 17M 4.5
Vue tooling for VS Code
Pine Wu Install
- C# 1.21.0 15.6M 4
C# for Visual Studio Code (powered ...
Microsoft Install
- src > JS serviceWorker.js > register > window.addEventListener('load') callback
- 39
40
41 checkValidServiceWorker(swUrl, config);
42 // Add some additional logging to localhost, p
43 // service worker/PWA documentation.
44 navigator.serviceWorker.ready.then(() => {
45
46 product
47 productSub
48 removeSiteSpecificTrackingException
49 removeWebWideTrackingException
50 requestMediaKeySystemAccess
51 sendBeacon
52 serviceWorker (property) Navigator.serviceWorke...
53 storage
54 storeSiteSpecificTrackingException
55 storeWebWideTrackingException
56 } userAgent
57 vendor
58
59 function registerValidSW(swUrl, config) {
60 navigator.serviceWorker
61 .register(swUrl)
62 .then(registration => {
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
- TERMINAL ... 1: node
You can now view create-react-app in the browser.
Local: http://localhost:3000/
On Your Network: http://10.211.55.3:3000/
Note that the development build is not optimized.
- Ln 43, Col 19 Spaces: 2 UTF-8 LF JavaScript

How to write mathematical equations into codes?

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos \theta & 0 \\ \sin \theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix},$$

```
def update_pose(self, v, omega, dt):  
    # Update the robot's pose based on linear velocity (v) and angular velocity (omega)  
    self.x += v * np.cos(self.theta) * dt  
    self.y += v * np.sin(self.theta) * dt  
    self.theta += omega * dt  
  
    return self.x, self.y, self.theta
```

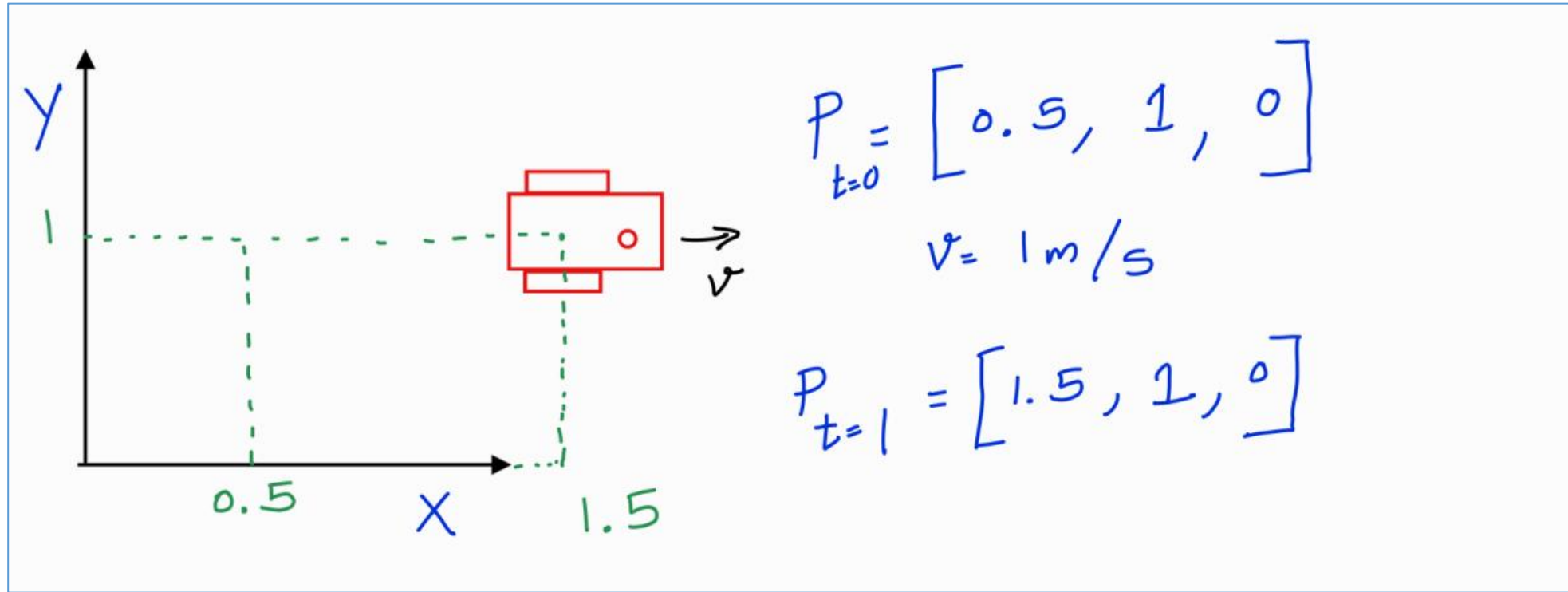
How to write mathematical equations into codes?

```
1  import numpy as np
2
3  class UnicycleRobot:
4      def __init__(self, x, y, theta):
5          self.x = x  # Initial X position
6          self.y = y  # Initial Y position
7          self.theta = theta  # Initial orientation (angle)
8
9      def update_pose(self, v, omega, dt):
10         # Update the robot's pose based on linear velocity (v) and angular velocity (omega)
11         self.x += v * np.cos(self.theta) * dt
12         self.y += v * np.sin(self.theta) * dt
13         self.theta += omega * dt
14
15         return self.x, self.y, self.theta
16
```

How to write mathematical equations into codes?

```
17  init_x = 0.5
18  init_y = 1
19  init_theta = 0
20
21  print("Initial Pose :",init_x,init_y,init_theta)
22
23  Robot_1 = UnicycleRobot(init_x, init_y, init_theta)
24
25  v = 0 # m/s
26  omega = 1.52 # rad/sec
27  dt = 1 # sec
28  x,y,theta = Robot_1.update_pose(v, omega, dt)
29
30  print("Pose :",x,y,theta)
```


How to write mathematical equations into codes?



```
Initial Pose : 0.5 1 0  
Pose : 0.5 1.0 1.52  
PS C:\Users\ARPIT JOON>
```

Lets Introduce Plots to the Codes...

```
1  import numpy as np
2  import matplotlib.pyplot as plt
3  import math
4
5  x_list = []
6  y_list = []
7  theta_list = []
8  time_list = []
9
10 class UnicycleRobot:
11     def __init__(self, x, y, theta):
12         self.x = x # Initial X position
13         self.y = y # Initial Y position
14         self.theta = theta # Initial orientation (angle)
15
16     def update_pose(self, v, omega, dt):
17         # Update the robot's pose based on linear velocity (v) and angular velocity (omega)
18         self.x += v * np.cos(self.theta) * dt
19         self.y += v * np.sin(self.theta) * dt
20         self.theta += omega * dt
21
22         if(self.theta > math.radians(360)):
23             self.theta = 0
24
25         return self.x, self.y, self.theta
26
```

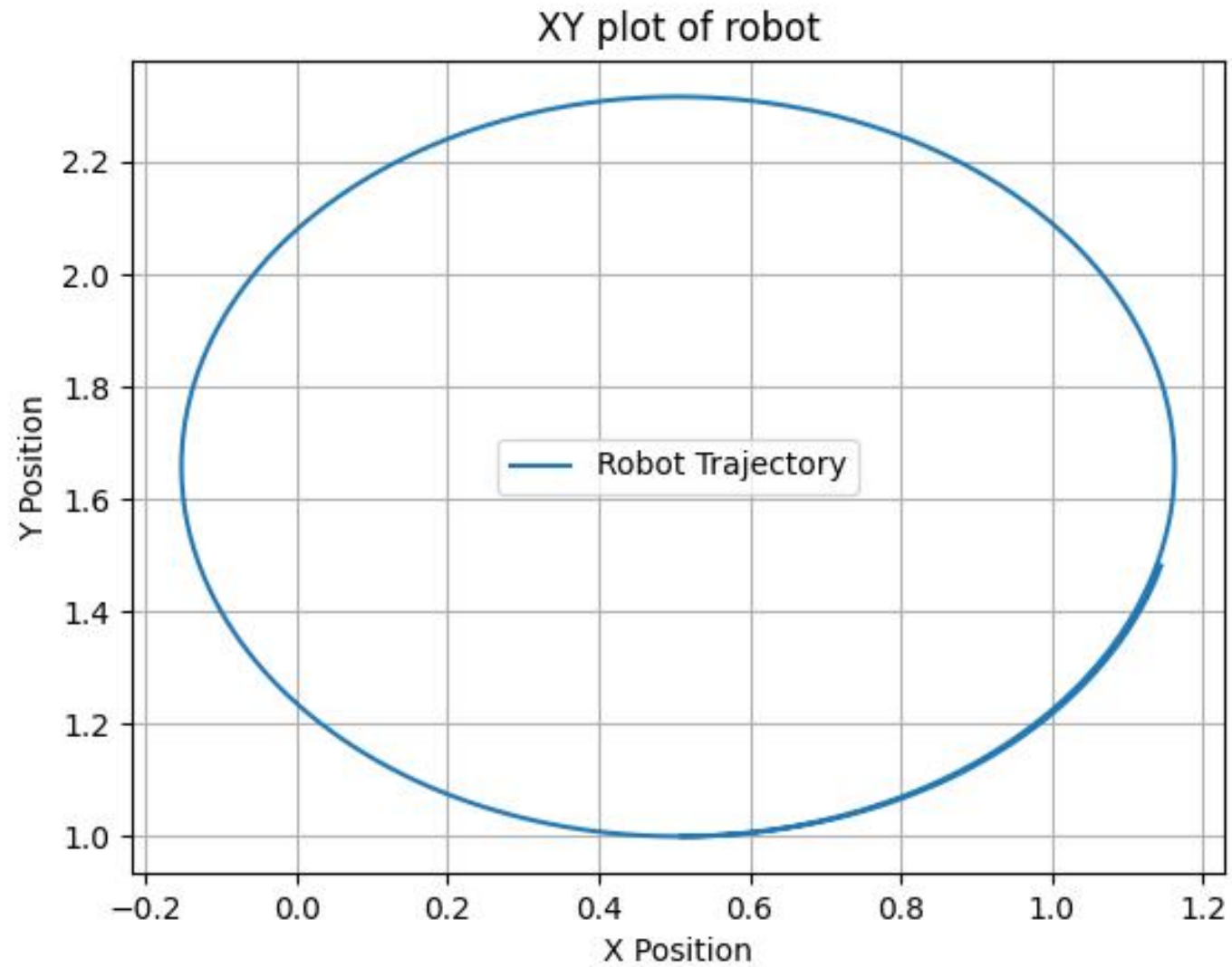
Lets Introduce Plots to the Codes...

```
26
27   init_x = 0.5
28   init_y = 1
29   init_theta = 0
30   print("Initial Pose :",init_x,init_y,init_theta)
31
32   Robot_1 = UnicycleRobot(init_x, init_y, init_theta)
33
34   v = 0.1 # m/s
35   omega = 0.152 # rad/sec
36   dt = 0.1 # sec
37   time = 0
38   for x in range(500):
39       x,y,theta = Robot_1.update_pose(v, omega, dt)
40       time +=dt
41
42       x_list.append(x)
43       y_list.append(y)
44       theta_list.append(theta)
45       time_list.append(time)
46
```

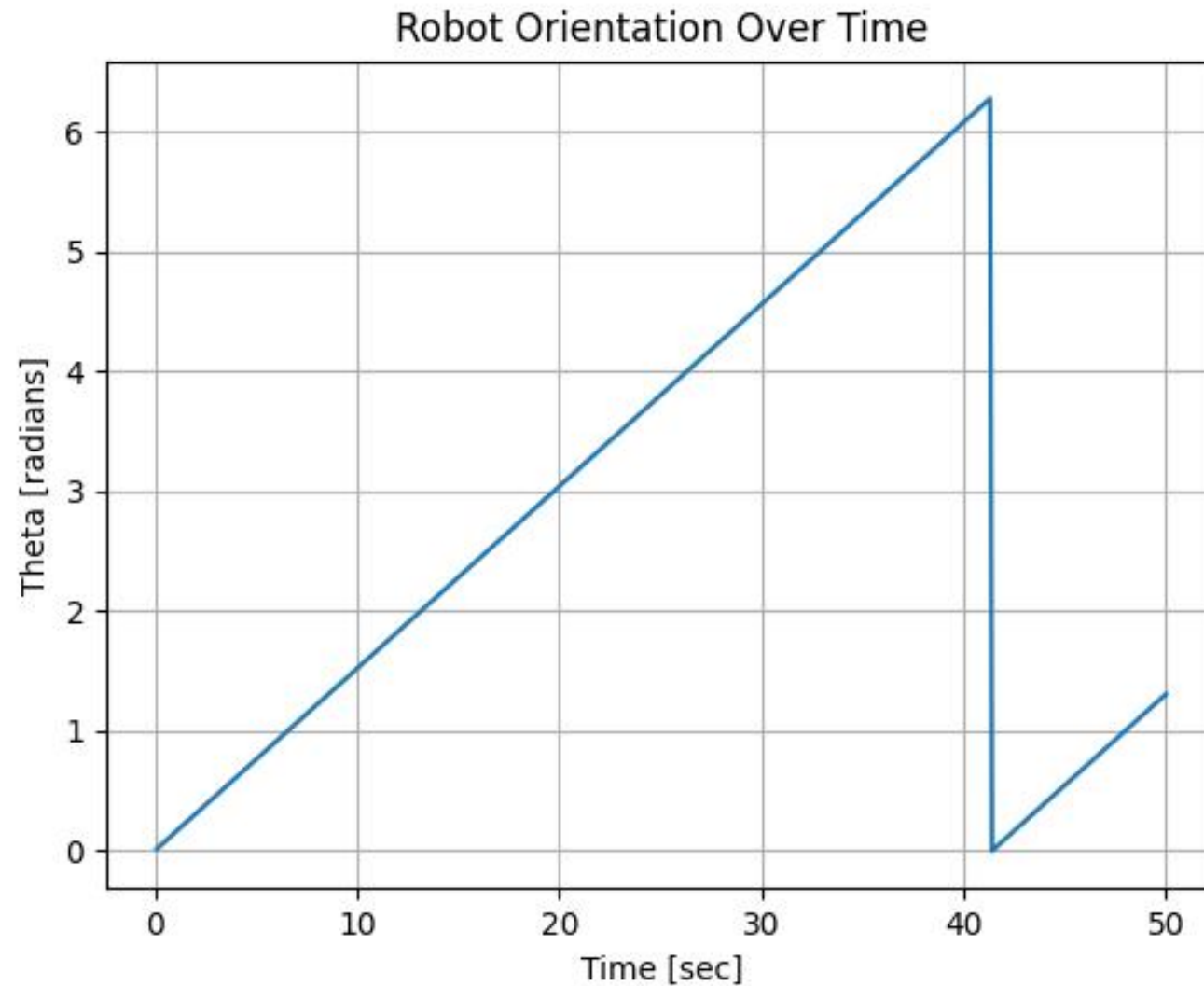
Lets Introduce Plots to the Codes...

```
46
47     plt.figure()
48     plt.plot(x_list, y_list, label="Robot Trajectory")
49     plt.xlabel("X Position")
50     plt.ylabel("Y Position")
51     plt.title("XY plot of robot")
52     plt.grid(True)
53     plt.legend()
54
55     plt.figure()
56     plt.plot(time_list, theta_list, label="Robot theta")
57     plt.xlabel("Time [sec]")
58     plt.ylabel("Theta [radians]")
59     plt.title("Robot Orientation Over Time")
60     plt.grid(True)
61
62     plt.show()
```


Lets Introduce Plots to the Codes...



Lets Introduce Plots to the Codes...



Download Codes from Github

github.com/Arpitjoon

The screenshot shows the GitHub profile page for Arpit Joon. The header includes the GitHub logo, the username 'Arpitjoon', and navigation links for Overview, Repositories (4), Projects, Packages, and Stars. The main content area features a circular profile picture of a man in a light blue blazer and blue shirt. Below the picture is the name 'Arpit Joon' and the bio 'Arpitjoon · he/him', with an 'Edit profile' button. To the right, the 'Popular repositories' section lists 'Basic_of_robotics' as a public repository in Python. Below that, the '12 contributions in the last year' section shows a calendar grid with green squares indicating contributions on specific days. The bottom right corner of the profile page shows the 'Contribution activity' section.

Arpitjoon

Overview Repositories 4 Projects Packages Stars

Popular repositories

Basic_of_robotics Public

Python

12 contributions in the last year

	Oct	Nov	Dec	Jan	Feb	Mar	Apr
Mon							
Wed							
Fri							

Learn how we count contributions

Contribution activity

Download Codes from Github

The screenshot shows the GitHub interface for a repository named 'Basic_of_robotics' owned by 'Arpitjoon'. The repository is public and has 1 branch (main) and 0 tags. The 'Code' tab is selected, showing a list of files: 'LICENSE', 'kinematics with plot.py', and 'kinematics.py'. The 'LICENSE' file was committed initially 15 hours ago. The other two files were added via upload, with 'kinematics.py' also added 15 hours ago. A green 'Code' button is visible in the top right, and a green 'Add a README' button is at the bottom right.

Arpitjoon / Basic_of_robotics

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

Basic_of_robotics Public

main 1 branch 0 tags

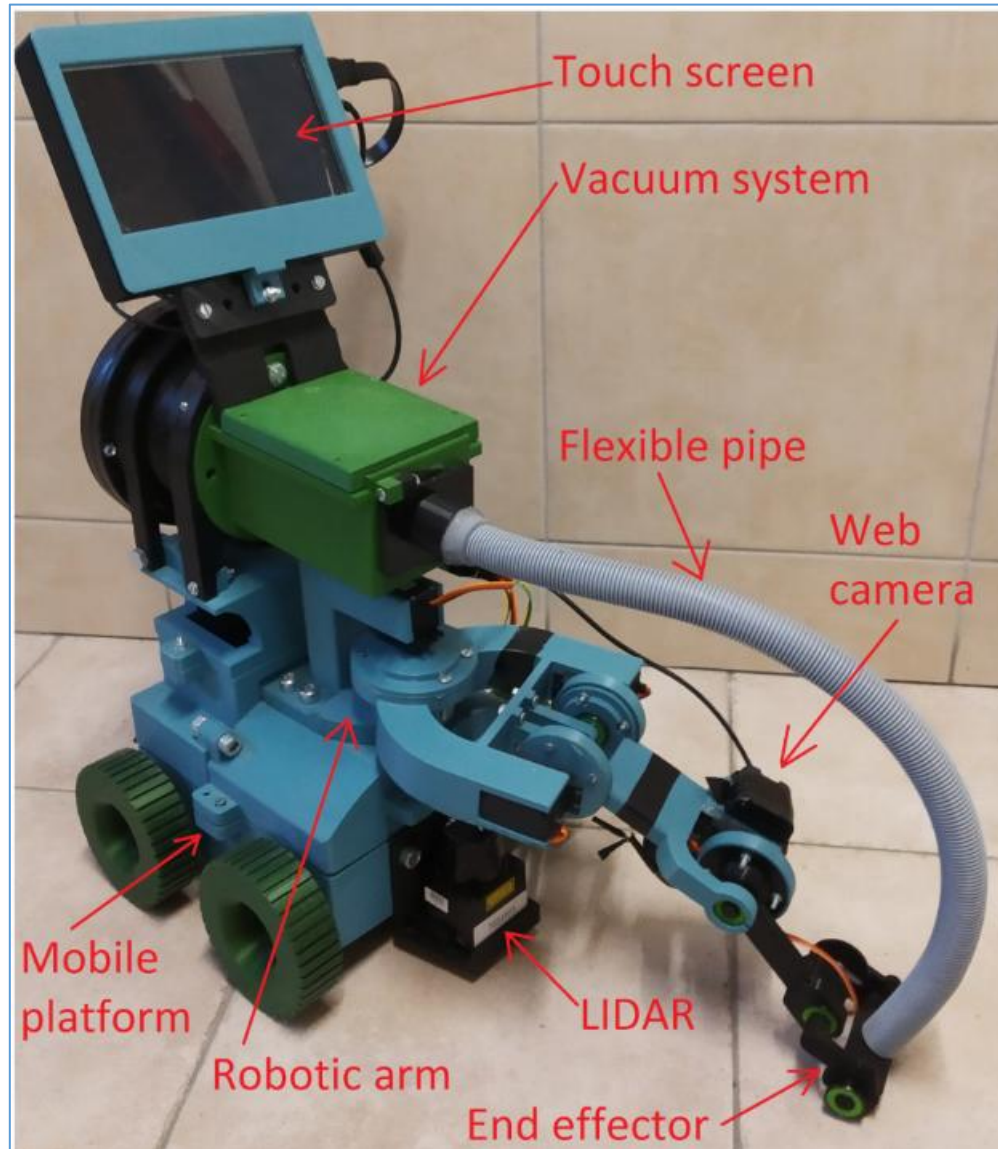
Go to file Add file <> Code

Arpitjoon Add files via upload 58b37ad now 3 commits

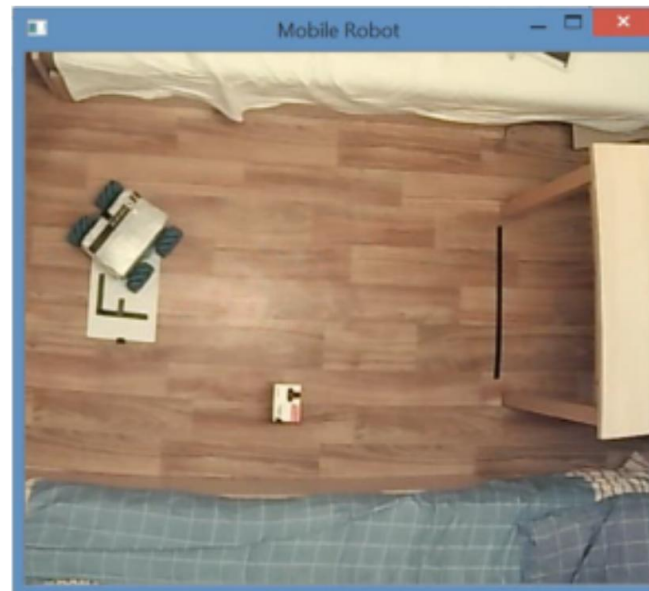
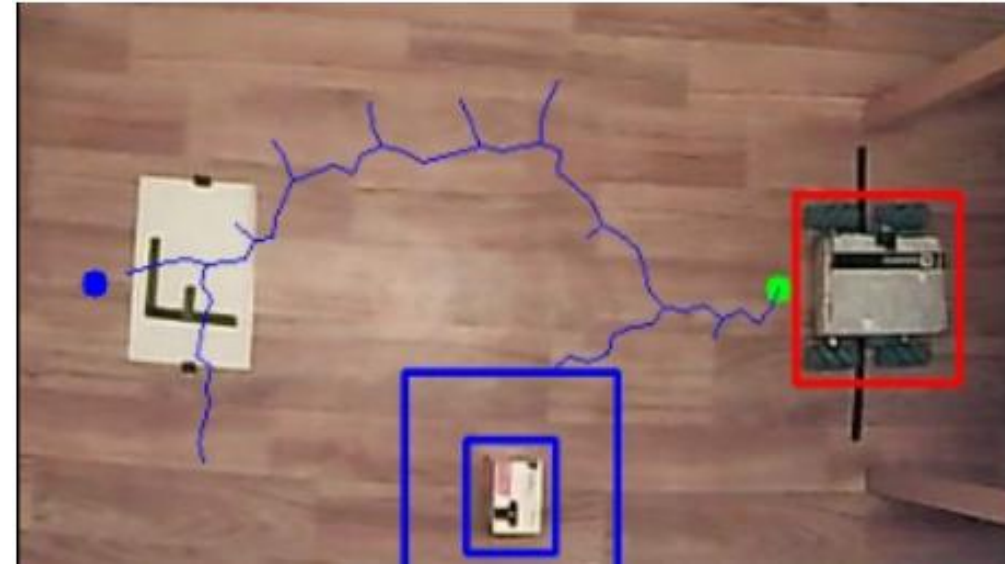
LICENSE	Initial commit	15 hours ago
kinematics with plot.py	Add files via upload	now
kinematics.py	Add files via upload	15 hours ago

Help people interested in this repository understand your project by adding a README. Add a README

My Previous Projects



My Previous Projects



My Previous Projects



My Previous Projects



My Previous Projects



My Previous Projects



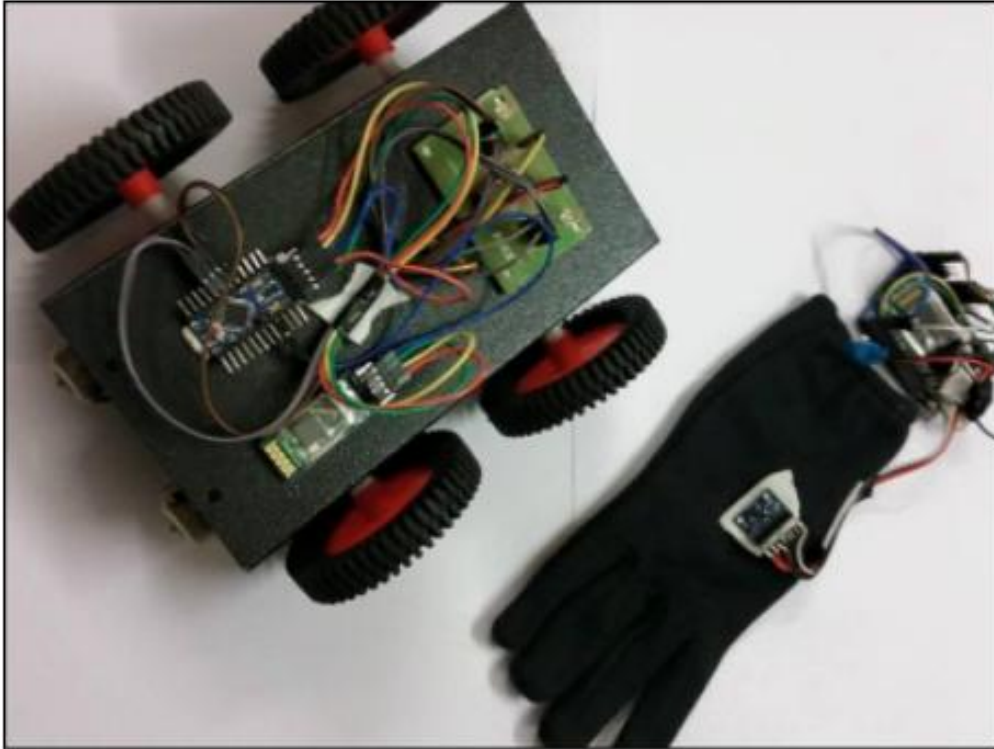
My Previous Projects



My Previous Projects



My Previous Projects



Research papers

**applied sciences**

[Submit to this Journal](#)
[Review for this Journal](#)
[Propose a Special Issue](#)

Article Menu

Academic Editor
**Alessandro Gasparetto**

[Subscribe SciFeed](#)

[Recommended Articles](#)

[Related Info Link](#)

Version is current.

Open Access

Article

Design of Autonomous Mobile Robot for Cleaning in the Environment with Obstacles

by  Arpit Joon *   and  Wojciech Kowalczyk *  

Institute of Automatic Control and Robotics, Poznan University of Technology, Piotrowo 3A, 60-965 Poznan, Poland

* Authors to whom correspondence should be addressed.

Appl. Sci. **2021**, *11*(17), 8076; <https://doi.org/10.3390/app11178076>

Received: 30 July 2021 / Revised: 27 August 2021 / Accepted: 27 August 2021 / Published: 31 August 2021

(This article belongs to the Special Issue **Advances in Robot Motion and Control—In Memory of Professor Krzysztof Kozlowski**)

[Download](#) [Browse Figures](#)[Versions Notes](#)

[Order Article Reprints](#)


Altmetric


Share


Help



Cite



Discuss in SciProfiles



Endorse


Comment

Research papers

IEEE Xplore® Browse ▾ My Settings ▾ Help ▾ Institutional Sign In 

All ▾  [ADVANCED SEARCH](#)

Conferences > 2022 26th International Confe... 






Design of one degree of freedom rotating platform applied in leader-follower autonomous robot control

Publisher: IEEE [Cite This](#) [PDF](#)

Wojciech Kowalczyk ; Arpit Joon [All Authors](#)

1
Cites in
Paper


61
Full
Text Views

[More Like This](#)

Need Full-Text
access to IEEE Xplore
for your organization?
[CONTACT IEEE TO SUBSCRIBE >](#)

Research papers

IEEE Xplore® Browse ▾ My Settings ▾ Help ▾ Institutional Sign In 

All ▾

Q

ADVANCED SEARCH






Conferences > 2023 27th International Confe... ?

Leader Following Control of Non-holonomic Mobile Robots Using EKF-based Localization

Publisher: IEEE [Cite This](#) [PDF](#)

Arpit Joon ; Wojciech Kowalczyk [All Authors](#)


12
Full
Text Views

Need Full-Text
access to IEEE Xplore
for your organization?
[CONTACT IEEE TO SUBSCRIBE >](#)

More Like This

Research papers

IEEE Xplore® Browse ▾ My Settings ▾ Help ▾ Institutional Sign In 

All ▾

Q

ADVANCED SEARCH

Conferences > 2023 27th International Confe... ?

Distributed Control for Teams of Non-holonomic Mobile Robots Executing Competitive Tasks

Publisher: IEEE [Cite This](#) [PDF](#)

Wojciech Kowalczyk ; Arpit Joon [All Authors](#)

4

Full Text Views

R

©

Need Full-Text
access to IEEE Xplore for your organization?
[REQUEST A FREE TRIAL >](#)

More Like This

Thanks

THANK YOU FOR LISTENING

Email id's:

joonarpit@gmail.com

arpit.joon@doctorate.put.poznan.pl