COURSE Review 1
MOBILE ROBOTICS
CENG 5437-01, CENG 4391-02 SPRING 2021
1. Introduction to the Course To the videos – History, Cars, Applications. ROS robots
2. Physics, Inertia, URDF models
3. Selecting Motors for Wheeled Robots
4. PID control of Wheeled Robots
SHAKEY STARTS THE REVOLUTION!
Figure 2. A decomposition of a mobile robot control system based on task achieving behaviors.
Why Learn Physics - Prof - When would we use this stuff??

Why Learn Physics - Prof - When would we use this stuff??
SHOULD HAVE LISTENED TO THE PHYSICS PROF!
Inertia

4. Inertia and URDF Chapter 2 in Textbook (5435 Web – Text)

Stopping Distances

MotorSelection Videos
<?xml version='1.0'?>
<robot name="dd_robot">
  ...
  <inertial>
    <mass value="5"/>
    <inertia ixx="0.13" ixy="0.0" ixz="0.0"
            iyy="0.21" iyz="0.0" izz="0.13"/>
  </inertial>
  ...
  <inertial>
    <mass value="0.5"/>
    <inertia ixx="0.0001" ixy="0.0" ixz="0.0"
            iyy="0.0001" iyz="0.0" izz="0.0001"/>
  </inertial>
</link>
  ...
  <inertial>
    <mass value="0.5"/>
    <inertia ixx="0.01" ixy="0.0" ixz="0.0"
            iyy="0.005" iyz="0.0" izz="0.005"/>
  </inertial>
  ...
  <inertial>
    <mass value="0.5"/>
    <inertia ixx="0.01" ixy="0.0" ixz="0.0"
            iyy="0.005" iyz="0.0" izz="0.005"/>
  </inertial>
  ...
</robot>
THINK AHEAD

Loss in KE indicates 71MPH at crash!!
Example System

- Customer driven requirements include:
  - Max load weight (65kg)
  - Battery voltage (168Vdc)
  - Throughput, longest move time (≤ 40s)

https://www.youtube.com/watch?v=VJFDU31LQGM&app=desktop
Example System

- Autonomous Warehouse Fork Lift For Picking Up Pallets and Transporting Them

![Diagram showing the components of the robot and load, including controls, traction drive, lift drive, batteries, a load of 65 kg max, lift platform of 4 kg, ballscrew driven lift, 6" diameter wheels (one driven by traction motor).]

**TOTAL WEIGHT (ROBOT AND LOAD)= 418 pounds → 190 kg mass**
Control_PID_TLH

PID_EXAMPLES&Video

StateSpace

MotorControl_DC_Klafter

PWM_Eshed

Feb 8.
If the input $R(s)$ is constant, we say the control system is a *regulator* since the object is to maintain the output at some constant value in the presence of disturbances. Temperature control or speed-control closed-loop systems are of this type. Other control systems are designed to allow the output to follow some time function as input. Such systems that control mechanical position or motion are called *servomechanisms*.

The transfer function $Y(s)/R(s)$ for the system in Figure 1.7 is derived by observing that

\[
Y(s) = G(s)E(s)
\]
\[
E(s) = R(s) - H_f(s)Y(s)
\]

and eliminating $E(s)$ from these equations to yield
\[ g = 500 \text{ so from Equation 1.16 } M_{web} \]

\[ \Gamma_{k_1} = \text{Gain}_1 = \frac{Y(s)}{R_{CS1}} = \frac{500}{1 + 500(0.09)} = \frac{500}{46} = 10.87 \quad \text{5 pts} \]

\[ \Gamma_{k_2} = \text{Gain}_2 = \frac{Y(s)}{R_{CS1}} = \frac{450}{1 + 450(0.09)} = \frac{450}{41.5} = 10.84 \quad \text{5 pts} \]

\[ \Delta \Gamma_k = \frac{10.87 - 10.84}{10.87} = \frac{0.0306}{10.87} = 0.0028 \]

\[ \text{or } 0.28\% \quad \text{5 pts} \]

\[ \Delta g = \frac{500 - 450}{500} = \frac{50}{500} = 0.100 \]

\[ \text{or } 10\% \quad \text{5 pts} \]

FEEDBACK IS GREAT
Example 5.15  Reduction of a Second-Order Equation

Consider the second-order equation

\[ m\ddot{x}(t) + b\dot{x}(t) + kx(t) = f(t). \]

Using the principles just defined, set \( x_1(t) = x(t) \) and \( x_2(t) = \dot{x}(t) \), so the first-order system becomes

\[
\begin{align*}
\dot{x}_1(t) &= x_2(t), \\
\dot{x}_2(t) &= -\frac{k}{m}x_1(t) - \frac{b}{m}x_2(t) + \frac{f(t)}{m}.
\end{align*}
\]

The matrix equation is

\[
\dot{x}(t) = \begin{bmatrix} 0 & 1 \\ -\frac{k}{m} & -\frac{b}{m} \end{bmatrix} x + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \frac{f(t)}{m}.
\]
For $x + 16 \dot{x} + 64x = u(t)$

$\ddot{x}(t) = \begin{bmatrix} 0 & 1 \\ -64 & -16 \end{bmatrix} x + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u(t)$

Output is $x(t)$

$x(t) = \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix}$
**Linearization**

Suppose the acceleration of an automobile can be described by the equation of motion

\[ M \frac{dv(t)}{dt} = cu(t) - \alpha v^2(t), \]  \hspace{1cm} (12.33)

where the first term represents the acceleration caused by the engine at a throttle setting \( u \) and the second term is the drag caused by air resistance. Since this force is proportional to the square of the speed, the equation is nonlinear. Solving this by numerical techniques would not be difficult if the constants
Take \((U_0, V_0)\) to be the “operating point,” so that the car travels at a constant speed \(V_0\) for a constant throttle position \(U_0\). Inserting this condition into Equation 12.33 yields

\[
M \frac{dV_0}{dt} = cU_0 - \alpha V_0^2(t) = 0,
\]

or \(V_0 = \sqrt{cU_0/\alpha}\). Let us assume that a small change in the throttle position leads to a small change in speed. Thus, we set

\[
u(t) = U_0 + \Delta u, \quad v(t) = V_0 + \Delta v
\]

and substitute again into the equation of motion. The result is

\[
M \frac{d}{dt} [V_0 + \Delta v] = c[U_0 + \Delta u] - \alpha [V_0 + \Delta v]^2.
\]

Using the result that \(cU_0 = \alpha V_0^2\) and expanding the terms but neglecting the second-order term \(-\alpha \Delta v^2\) leads to the approximation

\[
M \frac{d}{dt} [\Delta v] \approx c[\Delta u] - 2\alpha V_0 \Delta v.
\]

Writing \(\Delta v = v_a(t)\) and \(\Delta u = u_a(t)\) to indicate the linearized variables, the original differential equation of motion described by Equation 12.33 becomes the linear differential equation

\[
M \frac{dv_a(t)}{dt} + 2\alpha V_0 v_a(t) = cu_a(t).
\]
TWO-DIMENSIONAL TAYLOR SERIES

The notion of sequences and series of functions of a single variable as described in Chapter 6 can be extended to functions of several variables. For example, the power series expansion for a function of two variables $F(x, y)$ is

$$F(x, y) = \sum_{n=0}^{\infty} f_n(x, y) = f_0(x, y) + f_1(x, y) + \cdots + f_n(x, y) + \cdots,$$  \hspace{1cm} (12.35)

with the terms

$$f_n(x, y) = c_{n,0} x^n + c_{n,1} x^{n-1} y + \cdots + c_{n,n-1} x y^{n-1} + c_{n,n} y^n.$$
1. Automated Delivery


Feb 22

Starship Autonomous Food Delivery Robots Deployed at University of Houston

StarshipPatentUS10732641.pdf

MOBILE ROBOT SYSTEM AND METHOD FOR GENERATING MAP DATA USING STRAIGHT LINES EXTRACTED FROM VISUAL IMAGES

Applicant: Starship Technologies OÜ, Tallinn (EE)

Inventors: Ahti Heinla, Tallinn (EE); Kalle-Rasmus Volkov, Tallinn (EE); Lindsay Roberts, Tallinn (EE); Indrek Mandre, Tallinn (EE)

Assignee: STARSHIP TECHNOLOGIES OÜ, Tallinn (EE)

Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.

Applied No.: 15/968,802

Filed: May 2, 2018

Field of Classification Search
CPC .. G05D 1/0274; G05D 1/0212; G05D 1/0088; G05D 1/0251; G05D 2201/0216;

References Cited

U.S. PATENT DOCUMENTS

FOREIGN PATENT DOCUMENTS

OTHER PUBLICATIONS
ABSTRACT

A mobile robot is configured to navigate on a sidewalk and deliver a delivery to a predetermined location. The robot has a body and an enclosed space within the body for storing the delivery during transit. At least two cameras are mounted on the robot body and are adapted to take visual images of an operating area. A processing component is adapted to extract straight lines from the visual images taken by the cameras and generate map data based at least partially on the images. A communication component is adapted to send and receive image and/or map data. A mapping system includes at least two such mobile robots, with the communication component of each robot adapted to send and receive image data and/or map data to the other robot. A method involves operating such a mobile robot in an area of interest in which deliveries are to be made.

28 Claims, 7 Drawing Sheets
1. 3_Transforms_Rotationsscan0001.pdf
2. 4_RotateQuaternion0001a.pdf

   OurBook - Show pose Quaternions (See 5435 – Text) Page 84-85 ROS Pose/Roomba Topics

   4a_PG 84 Text Pose of a Turtlebot Robot and its Topics.pdf

PRESENTATION 2.

1. 5_DD_references2_15.pdf
2. 6_Steering_Odometry_references.pdf
1. Rotation by an angle $\alpha$ about the $x$-axis:

$$R_{x,\alpha} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \alpha & -\sin \alpha \\ 0 & \sin \alpha & \cos \alpha \end{bmatrix}; \quad (3.29)$$

2. Rotation by an angle $\phi$ about the $y$-axis:

$$R_{y,\phi} = \begin{bmatrix} \cos \phi & 0 & \sin \phi \\ 0 & 1 & 0 \\ -\sin \phi & 0 & \cos \phi \end{bmatrix}; \quad (3.30)$$

3. Rotation by an angle $\theta$ about the $z$-axis:

$$R_{z,\theta} = \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix}. \quad (3.31)$$
2. Rotate \([1, 2, 3]\) \(60^\circ\) ccw around \(z\) axis. 

\[ \begin{align*} 
(1, 2, 3) & \rightarrow (1', 2', 3') \\
2 & \rightarrow \frac{1}{2} \\
(1, 2) & \rightarrow (\sqrt{3}, 1, 0) \\
(0, 1, \frac{3\sqrt{3}}{2}) & \rightarrow (0, 1, 0) \\
(0, 0, \frac{3\sqrt{3}}{2}) & \rightarrow (0, 0, \frac{3\sqrt{3}}{2}) \\
\end{align*} \]

\[ \begin{align*} 
\tan \theta & = \frac{\sqrt{3}}{3} \\
\theta & = 24.5^\circ \\
\end{align*} \]
Matrix Rotation

$x$-axis rotation:
\[
\begin{pmatrix}
1 & 0 & 0 \\
0 & \cos \alpha & \sin \alpha \\
0 & -\sin \alpha & \cos \alpha
\end{pmatrix}
\]
\[
x = 1 \\
y = \cos \alpha + \sin \alpha \\
z = -\sin \alpha + \cos \alpha
\]

$y$-axis rotation:
\[
\begin{pmatrix}
\cos \beta & 0 & -\sin \beta \\
0 & 1 & 0 \\
\sin \beta & 0 & \cos \beta
\end{pmatrix}
\]

$z$-axis rotation:
\[
\begin{pmatrix}
\cos \gamma & \sin \gamma & 0 \\
-\sin \gamma & \cos \gamma & 0 \\
0 & 0 & 1
\end{pmatrix}
\]

Final rotation matrix:
\[
\begin{pmatrix}
\cos \beta \cos \gamma & -\cos \beta \sin \gamma & \sin \beta \\
\sin \alpha \sin \beta \cos \gamma + \cos \alpha \sin \gamma & -\sin \alpha \sin \beta \sin \gamma + \cos \alpha \cos \gamma & -\sin \alpha \cos \beta \\
-\cos \alpha \sin \beta \cos \gamma + \sin \alpha \sin \gamma & \cos \alpha \sin \beta \sin \gamma + \sin \alpha \cos \gamma & \cos \alpha \cos \beta
\end{pmatrix}
\]
Example of unit quaternion

Our notation:

- Rotation of vector $\mathbf{v}$ is $\mathbf{v}'$
  - Where $\mathbf{v}$ is: $\mathbf{v}=ai+bj+ck$
- By quaternion $q=(w,u)$
  - Where $w=1$
- Vector (axis): $\mathbf{u}=i+j+k$
- Rotation angle: $120^\circ = (2\pi)/3$ radian ($\theta$)
- Length of $\mathbf{u}=\sqrt{3}$
- If we rotate a vector, the result should be a vector.
Example of quaternion rotation (cont’d)

- So to rotate \( \mathbf{v} \):
  \[
  \mathbf{v}' = q \mathbf{v} q^* 
  \]

- Where \( q^* \) is conjugate of \( q \):
  \[
  q^* = \frac{(1-i-j-k)}{2} 
  \]

So now we can substitute \( q \mathbf{v} q^* \) to matrix form of quaternion
TurtleBot's odometry

The odometry data to determine position and orientation can become very inaccurate as the TurtleBot moves a long distance. The inaccuracy can be due to errors in the robot's parameters such as incorrect wheel diameters used in calculation of distance or due to the uneven driving surfaces causing the wheel encoders to output inaccurate data. A comprehensive discussion of odometry is found in the paper Measurement and Correction of Systematic Odometry Errors in Mobile Robots by Johann Borenstein and Liqiang Feng. The paper can be found at the following site: http://www-personal.umich.edu/~johannb/Papers/paper58.pdf.
DD Robot moves in x,y
Turns around z-axis
As you can see there: [http://www.ros.org/doc/api/geometry_msgs/html/msg/PoseWithCovariance.html](http://www.ros.org/doc/api/geometry_msgs/html/msg/PoseWithCovariance.html) the covariance is a float[36] a 6x6 matrix. Diagonal terms are the trust you have in your sensor for each Dof. You have 6 Dof, position (x, y, z) and orientation (x, y, z) even if you can see the orientation in Quaternion. You can estimate your sensor or algorithm accuracy with experiment. If you see your data are good for 1cm in translation and 0.1 radian in rotation you can use this matrix:

\[
\begin{bmatrix}
0.01 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\
0.0 & 0.01 & 0.0 & 0.0 & 0.0 & 0.0 \\
0.0 & 0.0 & 0.01 & 0.0 & 0.0 & 0.0 \\
0.0 & 0.0 & 0.0 & 0.1 & 0.0 & 0.0 \\
0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\
0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.05
\end{bmatrix}
\]

If you have no information for one Dof you can put a huge value.
twist:
  twist:
    linear:
      x: 0.0
      y: 0.0
      z: 0.0
    angular:
      x: 0.0
      y: 0.0
      z: -0.00174532925199
    covariance: [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0]
---
Pose pose

# Row-major representation of the 6x6 covariance matrix
# The orientation parameters use a fixed-axis representation.
# In order, the parameters are:
# (x, y, z, rotation about X axis, rotation about Y axis, rotation about Z axis)
float64[36] covariance

The PoseWithCovariance sub-message records the position and orientation of the robot while the TwistWithCovariance component gives us the linear and angular speeds as we have already seen. Both the pose and twist can be supplemented with a covariance matrix which measures the uncertainty in the various measurements.

THE DD-ROBOT STATE VECTOR IS X,Y AND Φ
TO PRESENTATION 2

DD-ROBOT (FEB 22) AND SENSORS TO MARCH 8TH.