A Trajectory Tracking Controller Design for a Nonholonomic Mobile Robot

Xiaoming Lang
ECE517
2013, Fall
Abstract

The robot in project is from a research project in Agricultural and Biological Engineering Department at UIUC. They use the robot to collect data from the fields by controlling it to move automatically. The objective of this project is to design a trajectory tracking controller for this mobile robot. Since this robot is widely used in research as a platform to implement some top layer designs, such as navigation, path planning and bug algorithms, we can rarely reach the bottom layer to directly design the DC motor controller. However, it is a good platform we can use what we learn to know better of adaptive control. In this report, I will first introduce the system and problem. Then, according to kinematic model, I will design a kinematic model controller. After that, I will estimate some parameters by using Extended Kalman Filter. Finally, I will introduce a controller design for the dynamic model. There must be some places not proper or precise, I hope you can point out and give some advice.

Introduction

The robot is Pioneer 3-AT produced by Adept Mobilerobots Inc. The parameters and brief introduction can be found in Appendix. The image of this robot is shown in Figure 1. We notice that the four wheels cannot rotate, so the system is a nonholonomic system, which makes the control problem more complicated.

![Figure 1](image)

A system is nonholonomic means the state of the system depends on the path taken to achieve it. In mathematics, a general system can be expressed in form of

\[ \dot{x} = F(x,u,t), \]
\[ H(x,\dot{x},t) = 0, \]

where \( x \in R^n \) is system state; \( u \in R^n \) is control input of system; \( H(x,\dot{x},t) = 0 \) is constraint of the system. If there exists a function \( G(x,t) \) (not constant) such that

\[ \frac{dG(x,t)}{dt} = H(x,\dot{x},t), \]

then the system is holonomic system. The corresponding constraints are called holonomic constraints. Conversely, the system is nonholonomic. In robotics, a system
is nonholonomic if the controllable degrees of freedom are less than the total degrees of freedom. In this particular system, it simply means, the robot cannot move towards the direction perpendicular to the wheels, which is shown in Figure 2.

![Figure 2](image)

The difficulties on Non-holonomic system are that, the Nonholonomic system does not satisfy the Brockett condition, so there does not exist smooth time invariant state feedback control such that can make system asymptotically stable. So this project comes up with a controller design for this nonholonomic (nonlinear) system.

**Kinematic Model and Controller Design**

A simple graph of robot is shown in Figure 3. There are some parameters we will use in the following discuss.

![Figure 3](image)

where $r$ is the radius of the wheels; $R$ is the distance between the centroid of the robot and the geometric center of the robot; $d$ is the distance between the wheels and the central line. $P_c$ is the centroid, and $P$ is the geometric center.

Moreover, before we move forward, we need to make some assumptions,

1. Velocities of both wheels on each side to be equal
2. Robot runs stable on a horizontal surface
3. No deformation on ground and tires
4. Mass distribution is uniform

The position of the robot can be expressed in Figure 4.
The robot possesses three degrees of freedom in its positioning which are represented by a posture.

\[
p = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix}
\]

where the heading direction \( \theta \) is taken counterclockwise from the x-axis. In Figure 4, \( P_r \) is the reference position; \( P_c \) is the real or control position. We can define the error position \( P_e = P_r - P_c \).

If the derivatives of \( x \) and \( y \) exist, \( \theta \) is not an independent variable anymore, because the constraint,

\[
\dot{x} \cos \theta + \dot{y} \sin \theta = 0
\]

In kinematic model, we choose the input as

\[
z = \begin{bmatrix} \omega_1 \\ \omega_2 \end{bmatrix},
\]

where \( \omega_1 \) is the angular velocity of the right side wheels, \( \omega_2 \) is the angular velocity of the left side wheels, which are also functions of time.

Then we can derive the kinematic model,
\[
\begin{bmatrix}
\dot{x} \\
\dot{y} \\
\dot{\theta}
\end{bmatrix}
= \dot{p} = J(p)z = \frac{r}{2}
\begin{bmatrix}
\cos \theta & \cos \theta \\
\sin \theta & \sin \theta \\
\frac{1}{d} & \frac{1}{d}
\end{bmatrix}
\begin{bmatrix}
o_1 \\
o_2
\end{bmatrix}
\]

The error position \( p_e \) is,
\[
p_e = \begin{bmatrix}
x_e \\
y_e \\
\theta_e
\end{bmatrix}
= T_e(p_e - \hat{p}_e)
= \begin{bmatrix}
cos \theta_e & \sin \theta_e & 0 \\
-\sin \theta_e & \cos \theta_e & 0 \\
0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
x_e - x_c \\
y_e - y_c \\
\theta_e - \hat{\theta}_e
\end{bmatrix}
\]

After taking the derivative of \( p_e \), we can get the system description,
\[
\begin{bmatrix}
\dot{x}_e \\
\dot{y}_e \\
\dot{\theta}_e
\end{bmatrix}
= \dot{p}_e
= \begin{bmatrix}
v_r \cos \theta_e \\
v_r \sin \theta_e \\
\omega_r
\end{bmatrix}
+ \begin{bmatrix}
y,e - \frac{r}{2d} & -\frac{y_e - r}{2d} & \omega_1 \\
-x,e - \frac{x,r}{2d} & \frac{x,e + \omega_r}{2d} & \omega_2 \\
-\frac{r}{2d} & \frac{r}{2d} & 0
\end{bmatrix}
\begin{bmatrix}
o_1 \\
o_2
\end{bmatrix}
\]

So the problem turns to design an input so that the system state can be tracked, i.e. \( p_e \) will approach to 0 in finite time.

\[
z = \begin{bmatrix}
o_1 \\
o_2
\end{bmatrix}
= \begin{bmatrix}
v_r \cos \theta_e + \frac{K_1}{r}x_e + d \omega_r + rd(K_2y_e + K_3 \sin \theta_e) \\
-v_r \cos \theta_e - \frac{K_1}{r}x_e + d \omega_r + rd(K_2y_e + K_3 \sin \theta_e)
\end{bmatrix}
\]

where \( K_1, K_2, K_3 \) are gains larger than 0.

By using this input, the first term \( v_r \cos \theta_e \) can be cancelled by this input. And other terms are just feedback coefficients, which we can change to adjust the system to be stable. Then we need to prove the system is stable under the input we chose by using Lyapunov method.

The Lyapunov function we choose is,
\[
V = \frac{1}{2}(x_e^2 + y_e^2) + \frac{1 - \cos \theta_e}{K_2}
\]
\[
\dot{V} = \dot{x}_e x_e + \dot{y}_e y_e + \dot{\theta}_e \sin \theta_e + \frac{\dot{\theta}_e \sin \theta_e}{K_2}
\]
\[
= -K_1x_e^2 - \frac{v_r K_1 \sin^2 \theta_e}{K_2} \leq 0
\]

The following result demonstrates that the uniformly asymptotically stability around \( p_e=0 \) under some conditions.

By linearizing the differential Equation around \( p_e=0 \), we can get
\[
\dot{p}_e = A p_e
\]
where

\[
A = \begin{pmatrix}
-K_1 & \omega_r & 0 \\
-\omega_r & 0 & v_r \\
0 & -v_r K_2 & -v_r K_3
\end{pmatrix}
\]

Assume that (a) \( v_r \) and \( \omega_r \) are continuous, (b) \( v_r, \omega, K_1 \) and \( K_3 \) are bounded, and (c) derivatives of \( v_r \) and \( \omega_r \) are sufficiently small. Then, \( A \) is continuously differentiable and is bounded. The characteristic equation for \( A \) is

\[
a_3 s^3 + a_2 s^2 + a_1 s + a_0 = 0
\]

where

\[
\begin{align*}
a_3 &= 1 \\
a_2 &= K_2 v_r + K_1 \\
a_1 &= K_2 v_r^2 + K_1 K_3 v_r + \omega_r^2 \\
a_0 &= K_1 K_2 v_r^2 + \omega_r^2 K_3 v_r
\end{align*}
\]

Since all coefficients \( a_i \) are positive and \( a_1a_2-a_0a_3>0 \), the real parts of all roots are negative through the Routh-Hurwitz Criterion. Therefore, Under these conditions, \( Pe=0 \) is uniformly asymptotically stable over \([0, \infty)\).

Then we can simulate to see whether the result we get is appropriate. I select 3 different values of \( K \). Figure 5, 6 and 7 show the results of my simulation.
From the results, we can see only see that, larger $K_3$ makes the system to be more stable. However, not all values of $K$’s can make system asymptotically stable. Only with more results, can we analyze the effects of the coefficients more precisely. The oscillations occurred because of the existence of cos and sin terms. But it will make the system not stable in the sense of Lyapunov.

Parameter Estimation

In this part, I want to estimate the parameters $r$ and $d$. Since the system is nonlinear and nonholonomic, I desire to use extended Kalman Filter. First, take the input as $(4,2)$, so $r$ and $d$ can be explicitly showed and measured in the result. Following the extended Kalman Filter design process, the system is expressed as,

$$
\dot{x} = 3r \cos \theta \\
\dot{y} = 3r \sin \theta \\
\dot{\theta} = \frac{r}{d}
$$

with arbitrary initial conditions. We also know that the $r$ and $d$ are unknown, but constants. So we have,

$$
\frac{dr}{dt} = 0 \\
\frac{dd}{dt} = 0
$$

Then we can construct the Kalman Filter

$$
\dot{x} = f(\hat{x}) + K[x - \hat{x}] \\
K = PC^T \tilde{R}^{-1} \\
\dot{P} = AP + PA^T + \tilde{Q} - PC^T \tilde{R}^{-1} CP
$$
where

\[
\begin{bmatrix}
x \\
y \\
\theta \\
r \\
d
\end{bmatrix}
\]

\[\begin{align*}
x & = f(x) \\
y & = h(x) = (x, r, d)^T
\end{align*}\]

\[
A = \frac{\partial f}{\partial x}
\]

\[
L = \frac{\partial f}{\partial \omega}
\]

\[
C = \frac{\partial h}{\partial x}
\]

\[
M = \frac{\partial h}{\partial \nu}
\]

\[
\tilde{Q} = LQL^T
\]

Q and R are the nominal noises, we choose 1 for simplicity. The matrix becomes,

\[
A = \begin{bmatrix}
0 & 0 & -3\hat{r}\sin\hat{\theta} & -3\cos\hat{\theta} & 0 \\
0 & 0 & 3\hat{r}\cos\hat{\theta} & -3\sin\hat{\theta} & 0 \\
0 & 0 & 0 & \frac{1}{d} & -\frac{\hat{r}}{d^2} \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0
\end{bmatrix}
\]

\[
C = I_{5x5}
\]

We can set arbitrary initial conditions for x, r and d. Then let the system track the values, so we can get the estimated parameters as the errors of states and parameters approaching 0. Figure 8 shows the result of the simulation. From Figure 8, we can see that the errors approach 0, so we estimated the parameters successfully.
Dynamic Model and Controller Design

In this part, I try to design a controller for the dynamic model. First, I will derive the dynamic model for the robot system. Generally, the model of nonholonomic system mobile robots are expressed as,

\[
\ddot{q} + V_m(q, \dot{q}) \dot{q} + F(q) + G(q) + \tau_d = B(q) \tau - A(q) \lambda.
\]

Where \( M \) is a symmetric, positive definite inertia matrix, \( V \) is the centripetal and coriolis matrix, \( F \) denotes the surface friction, \( G \) is the gravitational vector, \( \tau_d \) denotes bounded unknown disturbances including unstructured unmodeled dynamics, \( B \) is the input transformation matrix, \( \tau \) is the input vector, \( A \) is the matrix associated with the constraints, and \( \lambda \) is the vector of constraint forces. We consider that all kinematic equality constraints are independent of time, and can be expressed as follows

\[
A(q) \dot{q} = 0
\]

The nonholonomic mobile robot is transformed to and divided into the following two equations,

\[
\dot{q} = S(q) \omega \\
\bar{M} \ddot{\omega} + \bar{V} \omega + \bar{G} = \bar{B} T
\]

For the robot we use, we just simplify the situation. We assume that, there is no disturbance, No sliding friction loss, input is directly the torque applied on the wheels, moves only on horizontal surface. Then we can come up with the corresponding coefficients.
\[
\tilde{M} = \begin{bmatrix}
\frac{r^2}{4d^2} (md^2 + I) + I_w & \frac{r^2}{4d^2} (md^2 - I) \\
\frac{r^2}{4d^2} (md^2 - I) & \frac{r^2}{4d^2} (md^2 + I) + I_w
\end{bmatrix}
\]

\[
\tilde{V} = \begin{bmatrix}
0 & \frac{r^2}{2d} m_r R \dot{\theta} \\
\frac{r^2}{2d} m_r R \dot{\theta} & 0
\end{bmatrix}
\]

\[m = m_c + 4m_w\]

\[I = m_c R^2 + 4m_w d^2 + I_c + 4I_m\]

where \(m\) is the total mass, \(m_w\) is the mass of the wheels, \(I\) is the inertia moment of the corresponding axis. \(T\) denotes the torques operating on the wheels, also the inputs of the system.

After rearrange the equations, we can see the common expression.

\[
\begin{bmatrix}
\dot{x} \\
\dot{y} \\
\dot{\theta} \\
\dot{\omega}_1 \\
\dot{\omega}_2
\end{bmatrix}
= \begin{bmatrix}
\frac{r}{2} \cos \theta & \frac{r}{2} \cos \theta \\
\frac{r}{2} \sin \theta & \frac{r}{2} \sin \theta \\
\frac{r}{2d} & -\frac{r}{2d} \\
\frac{r}{2d} & -\frac{r}{2d}
\end{bmatrix}
\begin{bmatrix}
\omega_1 \\
\omega_2
\end{bmatrix}
+ \tilde{M}^{-1} \begin{bmatrix}
T_1 \\
T_2
\end{bmatrix}
\]

Our objective is to select proper input to make this system stable. There are several papers discussing this problem, but there is no general ways to come up with the solution. Now I just introduce a solution as,

\[
T = \tilde{B}^{-1} (-K_m \omega_e + Y \hat{p} - (\frac{\partial V}{\partial q} \tilde{S})^T)
\]

The details about this result can be found in the reference [5].

**Conclusion**

- For Kinematic Model, the controller design is simpler because we can set velocity or angular velocity directly.
- For Dynamic Model, we can only come up with controller design for some specific models.
- For more precise modeling, we should consider many more coefficients. I think the most important factor we should involve is slip rate, \(\lambda\). (ratio of actual velocity and output velocity)
- Other factors like friction and noise would also increases the difficulties of the problem
Reference


Appendix

Pioneer 3-AT

Pioneer 3-AT is a small four-wheel, four-motor skid-steer robot ideal for all-terrain operation or laboratory experimentation. The Pioneer 3-AT comes complete with one battery, emergency stop switch, wheel encoders and a microcontroller with ARCO5 firmware, as well as Pioneer SDK advanced mobile robotics software development package.

Pioneer research robots are the world’s most popular intelligent mobile robots for education and research. Their versatility, reliability, and durability have made them the preferred platform for advanced intelligent robotics. Pioneers are pre-assembled, customizable and upgradeable, and rugged enough to last through years of laboratory and classroom use.

Product features and benefits

- **Easy to Use** - Comes fully assembled and integrated with its accessory packages.
- **Reliable** - Construction is durable and rugged. Easily handles the small gaps, minor bumping, jarring, or other obstacles that hinder other robotic platforms. Some Pioneer robots have been in service for over 15 years.
- **Pioneer Software Development Kit** - All Adept MobileRobots platforms include Pioneer SDK, a complete set of robotics applications and libraries that accelerate the development of robotics projects. Pioneer SDKs backed by our product support team.
- **Customizable** - Easily customize by choosing from dozens of supported and tested accessories that integrate with your robotic platform. Additional help is available for future upgrades or added accessories.
- **Reference Platform** - Pioneer robots are a standard in intelligent mobile platforms. Search your preferred robotics journal or conference listings to find many examples of Pioneer platforms in research applications.
- **Technical Support** - Pioneer software and hardware comes fully documented with additional help available through our product support team.
Pioneer 3-AT

Dimensions (mm)

- 277
- 122
- 566
- 358
- 497
- 381
- 268
- 1

Core Software - included with all research platforms

ARIA provides a framework for controlling and sensing data from all MobileRobots platforms, as well as many accessories. Includes open-source infrastructure and utilities useful for writing robot control software, support for networked robots, and an extendable framework for client-server network programming.

MobileSim open-source simulator which includes all MobileRobots platforms and many accessories.

MobileEye graphical user interface client for remote operation and monitoring of the robot.

MapGen 3-Basic tool for creating and editing map files for use with ARIA, MobileSim, and navigation software.

SONAR provides sonar-based approximate localization and navigation.

Accessory Support Software - bundled with purchase of robotic accessory

ARINL enables robot, laser-based autonomous localization and navigation.

MOGS fuse robot and DGPS sensor data to guide your mobile robot outdoors.

Robotic Arm Support Pioneer arms are packaged with integrated software support.

Speech Recognition and Synthesis Library: Easy-to-use C++ development library for speech recognition based on the open source sphinx system. Speech synthesis (text-to-speech) based on Cepstral synthesizer.

ACTS Color Tracking System: Software application to read images from a camera and track the position and size of multiple color regions. Information can be incorporated into your own software via ARIA.

Optional Industrial Grade Internally Mounted Computers

<table>
<thead>
<tr>
<th>Mamba E8X-37 Dual Core 2.36 GHz - 2-8 GB RAM</th>
</tr>
</thead>
<tbody>
<tr>
<td>6 X USB2.0 Ports</td>
</tr>
<tr>
<td>2 X PCI/1394+ Slots</td>
</tr>
<tr>
<td>4 X RS-232 Serial Ports</td>
</tr>
<tr>
<td>2 X 10/100/1000 Ethernet Ports</td>
</tr>
<tr>
<td>Onboard Audio &amp; Video</td>
</tr>
<tr>
<td>Solid state Drive</td>
</tr>
<tr>
<td>Optional Wireless Ethernet</td>
</tr>
</tbody>
</table>

More Information:
See our website www.mobilerobots.com for a full range of supported accessories or contact our sales department to discuss your application.

Adapt Technology, Inc. 10 Columbia Drive, Amherst, NH 03031
Tel: 603-881-7060 Email: sales@mobilerobots.com
www.mobilerobots.com

Specifications subject to change without notice.

©2011 Adapt Technology, Inc. All Rights Reserved. The information provided in this communication is discussed in the property of Adapt Technology, Inc. and is protected by copyright and other intellectual property laws. In addition, we reiterate our policy to not specify any Adapt Technology, Inc. products as the marks and property of Adapt Technology, Inc. (and may be registered trademarks). All other trademarks or trade names are the property of their respective holders.