Fuzzy Control of a Mobile Robot for the Push-a-Box Operation

Yoshikuni Okawa
Osaka University
Yamadaoka, Suita
Osaka 565, Japan

Jun Aoki
Komatsu Ltd.
1200 Manda, Hiratsuka
Kanagawa 254, Japan

Abstract
We are contriving to build an intelligent robot. As a sample of the intelligent actions of robots, we consider the situation where a mobile robot pushes a square box. The operation of pushing a thing is unstable, in that once a small disturbance forces the system leave from the stable position, the pushing operation itself makes the deviation wider.

The state space is divided into 25 subregions, in each of which a control rule is specified manually. To avoid crisp division of the control space, we adopt the fuzzy control logic. Various possibilities are summed up by membership functions. We prepare three membership functions one of which is made active according to the distance from the goal being near, medium, or far.

We actually built a mobile robot, and have done many experiments. The results show that the proposed algorithm gives the expected results, although the treated situation is strongly limited, since this is the first step to make robots treat this type of intelligent jobs.

1 Introduction
Many research results have been reported on the path finding problem or the obstacle avoidance for mobile robots (see for example, Iyenger and Elfes, 1991). In the path finding, usually, algorithms to find a suitable route from the start position to the goal are discussed. The major target of the obstacle avoidance is to find obstacles on the pathway and to avoid them to reach the objective place. In those conventional research areas only the positional change of mobile robots has been investigated.

In this paper, apart from the mere navigation of mobile robots, we deal with the problem to make a mobile robot do some meaningful jobs. As an example of meaningful jobs, we take the operation to push a box by a mobile robot. You have probably had some experiences on pushing furniture in your home, or seeing a big ship pushed to a pier by tug boats. The operation of pushing a thing to change its place is one of the basic functions of our daily activities.

Moreover, from the standpoint of the mechanical dynamics this is classified as an unstable operation, in that we have to exert continuous control to maintain the object on its right truck, while dragging or carrying is considered stable. To make a mobile robot do such a complex job is a newest and exciting research theme in robotics, especially when we consider intelligence for robots.

Mason (Mason, 1986) first reported his research result on the pushing operation by a manipulator. He proposed an algorithm to determine which direction, clockwise or counterclockwise, a pushed object rotates, and also to calculate its rotating center. Peskin and Sanderson (Peskin and Sanderson, 1988) dealt with the grasp by a manipulator where pushing decreases positional uncertainty of a part to be grasped. Pham et al. (Pham et al., 1990) actually calculated the rotating center of a rectangular plate, and proved their result coincides with the Mason’s theorem. Mayeda and Wakatsuki (Mayeda and Wakatsuki, 1991) reported their result on pushing a box along a wall. Franchi et al. (Franchi et al., 1991) are doing the manipulator pushing experiments with a TV camera for position detection. Chen and Hwang (Chen and Hwang, 1991) wrote a paper on the path finding problem in the situation with movable obstacles, where a mobile robots sometime collides with obstacles, and then goes a while pushing the obstacles. Spreng (Spreng, 1991) proposed to use small pushes for the test to identify the object to which a mobile robot happens to bump in its course of actions. Yoshikawa and Kuris (Yoshikawa and Kuris, 1991) did experiments using a fixed pusher to locate the center of friction of a box.

In our research, we assume a flat floor for the mobile robot’s running environment. A robot is given an order to move a box from one place to another. For the control we adopt a fuzzy control algorithm. We have actually built a mobile robot, and done experiments. Our algorithm is effective in our stated situation, but also needs further generalization.
2 Statement of the problem

The control of a mobile robot usually has at least two different levels: the planning stage, and the real-time execution stage of that plan. We focus our attention to the latter: the real-time control problem of a mobile robot. We consider a robot's trajectory is calculated elsewhere, and given apriori to the controller.

Our mobile robot has four wheels, as shown in Fig. 1. The wheels are placed at each apex of a diamond, that is, two free casters are in the front and the rear center, and two driving wheels are at the right and left middle positions. A driving wheel is powered by an independent D. C. motor. Thus the steering vector consists of two input voltages to those driving wheels. If the left and right input voltages are equal, the robot goes straight. If they are different, it makes a left or a right turn.

A bar is fixed on the front frame of the mobile robot like a unicorn. This is called a pushing rod. A small square plate is attached to the rod at the top via a rotating joint. The plate is called a contact plate. A contact plate can be freely rotated in between the robot and the pushed object, whose relative angle is measured by a potentiometer. The angle is called a pushing angle and written as $\theta$.

We assume the following restrictions hold for our mobile robot's running environment.

- A mobile robot runs on a wide, free and flat floor (like a gymnasium). No obstacle exists.
- A robot knows its present position.
- A frictional force from floor is uniform and constant.
- A contact plate always keeps a firm contact with the object. No slippage or no separation is assumed to occur during the pushing operation.
- The pushing angle $\theta$ can be measured.
- The trajectory a robot must follow is given by an ordered set of positions.

Thus defined situation is probably the simplest one ever considered. Our policy is to start from the minimum core, and gradually proceed to the more complicated cases.

3 Analysis of the control system

A trajectory a mobile robot has to follow is given by an ordered set of positions $\{g_i = (x_i, y_i); i = 0, 1, ..., n\}$. The first position $g_0$ is called the start, and the last $g_n$ is the goal. Other positions are called the intermediate goal positions. A box is placed at $g_0$. A robot starts
to push the box toward g₁. If some condition is satisfied at g₁, the robot replaces g₁ by g₂, and continues to push the box to g₂. When the box reaches to gₙ, the control ends.

The basic configuration of our control system is depicted in Fig. 2. B is the geometrical center of an object, and g₂ is an active intermediate goal. The angle of the object to the robot is the object angle θ. The angle to see the goal gₙ from the box is the goal angle φ. The distance between the goal and the object is written as d. Thus, the state vector of this control system is (θ, φ, d).

Intuitively, to minimize the necessary energy or the travelling time to the goal, the robot has to proceed straight to the goal, or mathematically θ = φ = 0. We call this ideal state the straight state. This straight state can not be easily realized. When the system takes, for example, the position shown in Fig. 2, the question is what control is necessary for this state of the robot and a box. This is our control problem.

Let us write two voltages to the left and the right motor as v_L and v_R. Let us also write

\[ v = v_L + v_R \]

Then, v represents the linear speed of the robot along the trajectory, and w determines the radius of circular movement. Since v can not directly control θ or φ, we assume the speed v is a given constant. Thus, we have only one control w left to control θ and φ. Naturally we can not control two independent state variables by one steering variable. For example there is the case where, if we try to decrease θ, it increases φ, or vice versa. We have to depend on physical meanings of the controlled system, rather than the mathematical treatments.

First, we consider the role of the distance d. As an example, consider the case of an operator driving, say a fork-lift truck. When he is at a distant position from the goal, and finds himself not in the right direction, he will not steer in haste, since he has a plenty of time. But near the goal, he has to correct the state quickly. Certainly, the distance from the goal is not the major factor, but it affects the control moderately. We divide the distance into three ranges; Near (NE), Medium (ME), and Far (FA). The control w has three distinct policies; Left Turn (LT), Straight (ST), and Right Turn (RT). We have 12 rules for real-time control.

Parameter selection rules

- Rule 1 If \( d \) is NE, then select the parameter set 0.
- Rule 2 If \( d \) is ME, then select the parameter set 1.
- Rule 3 If \( d \) is FA, then select the parameter set 2.

Stability rules

- Rule 4 If \( θ \) is RL, then \( w \) is RT
- Rule 5 If \( θ \) is LL, then \( w \) is LT

Goal seek rules

- Rule 6 If \( θ \) is RS and \( φ \) is RL or RS, then \( w \) is ST
- Rule 7 If \( θ \) is RS and \( φ \) is ZO, or LS, or LL, then \( w \) is RT
Table 1 The decision table for the real-time control.

<table>
<thead>
<tr>
<th>θ</th>
<th>LL</th>
<th>LS</th>
<th>ZO</th>
<th>RS</th>
<th>RL</th>
</tr>
</thead>
<tbody>
<tr>
<td>φ</td>
<td>LT</td>
<td>LT</td>
<td>ST</td>
<td>ST</td>
<td>RT</td>
</tr>
<tr>
<td>RL</td>
<td>LT</td>
<td>LT</td>
<td>ST</td>
<td>LT</td>
<td>ST</td>
</tr>
<tr>
<td>RS</td>
<td>ST</td>
<td>ST</td>
<td>ST</td>
<td>ST</td>
<td>ST</td>
</tr>
<tr>
<td>ZO</td>
<td>ST</td>
<td>ST</td>
<td>RT</td>
<td>RT</td>
<td>RT</td>
</tr>
<tr>
<td>LS</td>
<td>ST</td>
<td>ST</td>
<td>RT</td>
<td>RT</td>
<td>RT</td>
</tr>
<tr>
<td>LL</td>
<td>ST</td>
<td>ST</td>
<td>RT</td>
<td>RT</td>
<td>RT</td>
</tr>
</tbody>
</table>

Fig. 3 The membership function for the object angle.

- Rule 8 If θ is ZO and φ is RL, then w is LT
- Rule 9 If θ is ZO and φ is RS, or ZO, or LS, then w is ST
- Rule 10 If θ is ZO and φ is LL, then w is RT
- Rule 11 If θ is LS and φ is RL, or RS, or ZO, then w is LT
- Rule 12 If θ is LS and φ is LS, or LL, then w is ST

The above 12 rules are listed in Table 1, which we call the decision table for real-time control.

In the fuzzy control theory the observed θ and φ are distributed into each of the defined categories by their membership functions. The membership function used for an angle has the structure shown in Fig. 3. This membership function has five parameters; (a1, ..., a5) for θ, and (b1, ..., b5) for φ. Thus, we have following three parameter sets:

- Parameter set 0 \((a_{10}, ..., a_{50}, b_{10}, ..., b_{50})\)
- Parameter set 1 \((a_{11}, ..., a_{51}, b_{11}, ..., b_{51})\)
- Parameter set 2 \((a_{12}, ..., a_{52}, b_{12}, ..., b_{52})\)

We select one from above three parameter sets according to the distance \(d\). The precise rules are:

- Rule 1' If \(0 \leq d < d_1\), then select the parameter set 0 for the angle membership functions.
- Rule 2' If \(d_1 \leq d < d_2\), then select the parameter set 1 for the angle membership functions.
- Rule 3' If \(d_2 \leq d\), then select the parameter set 2 for the angle membership functions.

where \(d_1\) and \(d_2\) are the prespecified thresholding values. The structure of the membership function to decide the control \(w\) is depicted in Fig. 4. This function has two parameters; \(c_1\) and \(c_2\).

Fig. 4 The membership function for the decision process.

Using a sample case, I will explain the computing procedure. The present state of the control system is assumed to be like the one shown in Fig. 5. Clearly, θ is minus (to the right from the reference axis a), and φ is plus (to the left of the reference axis b). Intuitively speaking, if θ and φ are both small, then the robot could push the box straight in the next short time interval. If θ or φ is not small, the robot must turn to the right, or push the box to the right direction. If the robot pushes the box to the right in an appropriate angle, then the box will rotate.
to the left (or it rotates counter-clockwise.) If the box rotates counter-clockwise, the goal angle has a chance to decrease favorably in this case. This indicates that there are some occasions for the robot to correct both $\theta$ and $\phi$ by a single control.

Fig. 5  A sample position used to explain the decision process.

We will describe the exact computing process. The mobile robot has, by the assumption stated in Section 2, always its present position. The coordinate of the intermediate goal now heading is given. By the geometrical relation the robot can calculate the distance $d$ and the goal angle $\phi$ using its present position and the measured $\theta$. Assume the measured $\theta_{mes}$ is located at the point indicated in Fig.

Fig. 6 Transformation of the measured object angle to the applicability by the membership function.

6. Then, the robot calculates the applicability of the statement "$\theta$ is ZO" is $m_1$, and that of "$\phi$ is RS" $m_2$.

The similar reasoning process is applied to $\phi$. The result is, for example, the applicability of the statement "$\phi$ is ZO" is $n_1$, and "$\phi$ is LS" $n_2$. Then by the Rule 9, in the cases of "$\theta$ is ZO and $\phi$ is ZO", and "$\theta$ is ZO and $\phi$ is LS", the robot understands it has to go straight. Write the applicability of the go-straight control as $\alpha_{ST}$. Then, we can calculate $\alpha_{ST}$ by

$$\alpha_{ST} = \frac{m_1n_1 + m_1n_2}{2}$$

Also by the Rule 7, in the cases of "$\theta$ is RS and $\phi$ is ZO", and "$\theta$ is RS and $\phi$ is LS", the robot realizes it has to turn right. The applicability is computed by

$$\alpha_{RT} = \frac{m_1n_1 + m_2n_2}{2}$$

Now, see Fig. 7. Thus computed $\alpha_{ST}$ and $\alpha_{RT}$ define two shaded areas in two triangles; ST and RT. The centroid of these two areas is computed, and its location is shown as the point P in Fig. 7. The horizontal axis is $u$. We assume $v = u_L + u_R$ is given as the average speed of the robot. From thus computed $u$ and the already known $v$, the robot is able to compute $u_L$ and $u_R$.

5 Simulation

Computer simulation is a valuable tool, even though actual experiments have their own meanings. We derive the equations of motion for the situation discussed in the previous section. The symbols related in mechanical behaviors of a mobile robot, an object, and motors are briefly interpreted in the following paragraphs.
1. symbols related to a mobile robot

\( v_1 \) velocity of translation

\( \omega_1 \) angular velocity of rotation

\( \theta_1 \) angle of the axis \( x_1 \) from \( X \)

\( D_L, D_R \) driving force of two wheels

\( N_{1L}, N_{1R} \) frictional force from floor

\( -F_x, -F_y \) forces from a pushed object

\( I_1 \) moment of inertia

\( M_1 \) mass

\( N_1 \) force from floor

\( r \) radius of a driving wheel

2. symbols related to a pushed object

\( v_2 \) velocity of translation

\( \omega_2 \) angular velocity around \( \theta_2 \)

\( \mu_{21} \) friction coefficient from floor

\( F_x, F_y \) forces from a mobile robot

\( I_2 \) moment of inertia

\( M_2 \) mass

\( N_2 \) force from floor

3. symbols related to a motor

\( i \) counter electromotive force

\( k \) voltage to current ratio

\( u_L, u_R \) voltage to two driving wheels

\( \mu_3 \) frictional coefficient of a wheel

\( I_3 \) moment of inertia

\( N_{3L}, N_{3R} \) frictional forces to wheels

Equations of motion related to robot's motion are given by

1. translation

\[
M_1 \dot{v}_1 = -N_{1L} - N_{1R} + F_x
\]

\[
M_1 \dot{w}_1 = D_L + D_R - F_y
\]

2. rotating motion

\[
I_1 \dot{\omega}_1 = a(D_R - D_L) + bF_x
\]

Equations of motion for a pushed object are given by

1. translation

\[
M_2 \begin{pmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{pmatrix} \begin{pmatrix} v_1 \omega_1 - b \omega_1 \\ v_1 - b \omega_1 \end{pmatrix} + \begin{pmatrix} d_1 \omega_2 - d_2 \omega_2 \\ d_3 \omega_2 - d_4 \omega_2 \end{pmatrix} = \begin{pmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{pmatrix} \begin{pmatrix} F_x \\ F_y \end{pmatrix} - \mu_{21} N_{2x}
\]

2. rotating

\[
M_2 \left( \frac{d_2(v_1 - b \omega_1) - d_4(v_1 \omega_1 - b \omega_1)}{d_2 + d_4} \right) + I_2 \dot{\omega}_2 = \mu_{21} N_{2z} \sqrt{d_2^2 + d_4^2} \cos(\psi + \tan^{-1} \frac{d_2}{d_4})
\]

Deleting \( F_x \) and \( F_y \) from robot's equations of motion (1), (2) and object's equations (3), (4), we have a set of equations of motion for a combined dynamics of both a robot and an object.

1. translation

\[
(M_1 + M_2) \dot{v}_1 = D_L + D_R + M_2 b \omega_1 + M_2 \left( \frac{d_1 \omega_2 + d_3 \omega_2 - \mu_{21} g \cos \psi}{\cos \theta} \right) \sin \theta - \left( -d_2 \omega_2 + d_4 \omega_2 - \mu_{21} g \sin \theta \right)
\]

2. rotation

\[
I_2 \dot{\omega}_1 = a(D_R - D_L) + M_2 b(v_1 \omega_1 - b \omega_1) - M_2 \left( \frac{d_1 \omega_2 + d_3 \omega_2 - \mu_{21} g \cos \psi}{\cos \theta} \right) \sin \theta - \left( -d_2 \omega_2 + d_4 \omega_2 - \mu_{21} g \sin \theta \right)
\]

\[
I_2 \dot{\omega}_2 = -M_2 \left( \frac{d_2(v_1 - b \omega_1) - d_4(v_1 \omega_1 - b \omega_1)}{d_2 + d_4} \right) + \mu_{21} M_2 \sqrt{d_2^2 + d_4^2} \cos(\psi + \tan^{-1} \frac{d_2}{d_4})
\]

where \( \psi \) is given by

\[
\psi = \tan^{-1} \frac{b_1 \sin \theta + v_1 \cos \theta + d_2 \omega_2}{-b_1 \cos \theta + v_1 \sin \theta - d_4 \omega_2}
\]

The equations of motion for a D.C. motor is given by

\[
T + I_3 \phi_c + \phi = k u
\]

where \( T \) is output torque of a motor. Geometry of a wheel gives us the following equation for torque balance.

\[
T = (D + \mu_3 N_3) \tau
\]

The following relations hold between wheels', \( \phi_L \) and \( \phi_R \) and robot's velocities \( v_1 \) and \( \omega_1 \).

\[
r \phi_L = v_1 - a_1 \omega_1
\]

\[
r \phi_R = v_1 + a_1 \omega_1
\]

Substituting eqs. (7), (8), and (9) into (5) and (6), we have

1. translation

\[
(M_1 + M_2 + \mu_3 N_3) v_1 + 2 \mu_3 b (v_1 - u_R) - \mu_3 (N_{3L} + N_{3R}) + M_2 b \omega_1 + M_3 (d_1 \omega_2 + d_3 \omega_2 - \mu_3 g \cos \psi) \sin \theta - \left( -d_2 \omega_2 + d_4 \omega_2 - \mu_3 g \sin \theta \right)
\]

2. rotation

\[
(M_1 + M_2 + \mu_3 N_3) \dot{v}_1 + \mu_3 M_2 \sqrt{d_2^2 + d_4^2} \cos(\psi + \tan^{-1} \frac{d_2}{d_4})
\]

Equations (10) and (11) represent total dynamics of our system including a robot, a box and two driving motors.

6 Experiments

We built an experimental mobile robot, whose photograph is shown in Fig. 8. A pushing rod is fixed on the front frame, on the top of which a contact plate is attached. The contact plate can rotate freely around the vertical axis of the rotating joint. The relative angle of the contact plate to the pushing rod is measured by a potentiometer and read into the control computer via an A-D converter. Two D.C. motors drive two wheels independently. A rotary encoder measures the rotational angle of the driving wheel.
Four microcomputers are installed in the robot. They make, so called, a functional parallel processing system. The first computer controls the speeds of two driving wheels. The second computes incrementally the position of the mobile robot using the information from two rotary encoders. The third calculates $\phi$ and $d$ from the measured $\theta$ and the given $\gamma$, and calculates a control from the decision table or the fuzzy control rule. The fourth is the master computer and executes the overall control. The necessary programs are down loaded via this master controller from a workstation.

First, as a preliminary experiment, fundamental runs along a short straight line, or around a circle have been tested. During these running experiments three parameter sets for the angle membership functions are determined. This is the learning phase.

Next, as the test phase, we assume a corridor like path, and pick up an appropriate point set from the path. This point set is given to the robot, and experiments are carried out. The computed positions of the robot are stored in the memory of the third computer, and after the experiment ends the record is recovered in a workstation in our laboratory. This process is repeated. One sample of the experimental results is shown in Fig. 9. Although small fluctuations are seen, the robot follows a specified path. The results are satisfactory within the limits we imposed.

7 Conclusion

We have proposed a method to make a mobile robot push a box from one place to another. Since this is the first step of the research, we strongly restricted the environment.

We have analyzed the situation and found that the system under consideration has a single control variable against two independent controlled variables. Since the system is indeterministic, we adopt the fuzzy control theory. We divide the state space into small sub-regions, and in each subregion we specify the necessary control, which is stated in the form of IF-THEN rules. We construct membership functions for input variables, and output control, by which two or more rules are combined to make a unique solution. The number of the basic rules is 12.

We have built an experimental robot and done many experiments. In the learning phase the various parameters are adjusted. Final parameters are used in the test experiments. We confirmed that the proposed system works as planned.

The situation considered here is too restrictive to apply the results to actual situations, although it is natural as the first step. The future direction is to widen the applicability or to build a system which is able to do the work in more realistic circumstances. The possible research theme or natural extensions of this research are:

- Consider the situation where a robot may bump obstacles.
- Consider the situation where the frictional force from floor is variable.
- A robot has to arrange the objects in a specified manner at the goal position.
- Consider the situation where a robot may stop and off the box, and once again continues to push the box.
- Give the robot a capability to plan the trajectory.
- etc.
It is clear that we have many and exciting research themes in this field.

8 References


Fig. 9 One sample of the experimental results in a corridor-like path. The path is 3m wide, while the robot is 1m wide.