CS 470

Motion control in BZRFlag

Daqing Yi

1 Robot motion

The motion in robotics usually consists of two components, which are *planning* and *control*. The planning component generates a motion reference based on a given task. The motion reference can either be a sequence of waypoints or a trajectory. The control component then drives the robot moving along the generated reference.

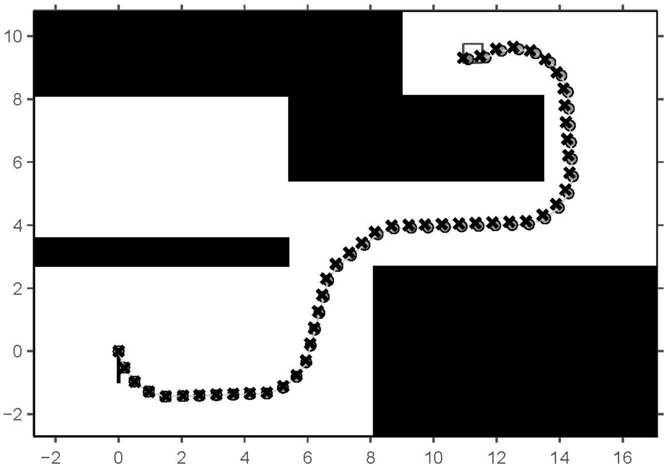


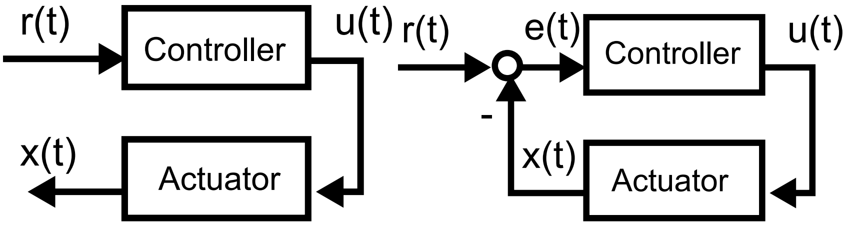
Figure 1: Planning and control in robot motion.

There are two types of controlling approaches, *feedforward*[1] and *feedback*[2]. A feedforward controller directly converts the motion reference into control commands by the dynamics

[1] http://en.wikipedia.org/wiki/Feed_forward_(control)

[2] http://en.wikipedia.org/wiki/Feedback

model of the actuator. The feedforward controller provides a simple mechanism, which usually is applied to the system that needs fast response. It only works well when the dynamics model is simple and has no uncertainty. While a feedback controller calculates the state error by comparing the current state of the actuator with the motion reference, and use the state error to determine the control commands according to the dynamics model of the actuator. It has better control precision and is more robust to the motion noise. Figure 2 shows two types of controllers.



(a) Feedforward system

(b) Feedback system

Figure 2: Control system.
$r(t)$ is the reference, $u(t)$ is the input and $x(t)$ is the state.

2 PID CONTROLLER

The PID controller is one of the most popular feedback controller[3]. The PID controller combines the *proportion* of the error, the *derivation* of the error and the *integral* of the error to generate the input to the actuator, as in Equation (1).

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{d}{dt} e(t). \tag{1}$$

In a computer-based controller, the running of the control algorithm is commonly driven by a digital timer [4]. It means that the control algorithm runs in a discrete-time manner instead of a continuous-time manner. Define the interval of the digital timer as ΔT. The measure of time can be simplified into the count of the time steps k. A discrete-time PID controller, which approximates the (continuous-time) PID controller, is written as

$$u(k) = K_p e(k) + K_i \sum_{i=1}^{k} e(i) \Delta T + K_d \frac{e(k) - e(k-1)}{\Delta T}. \tag{2}$$

In the lab, we only implement the discrete-time PD controller, which means that $K_i = 0$.

$$u(k) = K_p e(k) + K_d \frac{e(k) - e(k-1)}{\Delta T}. \tag{3}$$

Similarly, by setting $K_d = 0$, we have a P controller.

[3] http://en.wikipedia.org/wiki/PID_controller

[4] http://en.wikibooks.org/wiki/Control_Systems/Digital_and_Analog

3 Robot motion in BZRFlag

In BZRFlag, the task is having a tank grabbing the opponent's flag and bringing it back to the base. We select potential fields to provide the motion reference. We use PD controller to drive the tank moving along the motion reference.
The motion state of a tank consists of the location in x-axis $x(t)$, the location in y-axis $y(t)$ and the orientation $\theta(t)$. The dynamic model of the tank is

$$\begin{aligned} \dot{x}(t) &= v(t)\cos(\theta(t)); \quad (4)\\ \dot{y}(t) &= v(t)\sin(\theta(t)); \quad (5)\\ \dot{\theta(t)} &= \omega(t). \quad (6) \end{aligned}$$

Presenting it in a discrete-time way, we have

$$\begin{aligned} \Delta x(k) &= x(k+1) - x(k) = v(k)\cos(\theta(k))\Delta T; \quad (7)\\ \Delta y(k) &= y(k+1) - y(k) = v(k)\sin(\theta(k))\Delta T; \quad (8)\\ \Delta\theta(k) &= \omega(k)\Delta T. \quad (9) \end{aligned}$$

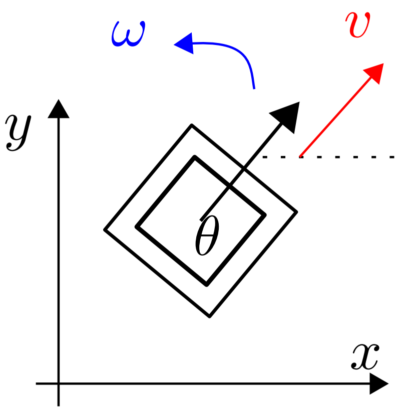


Figure 3: The motion of the tank.

How to design the controller to generate the linear velocity $v(k)$ and the angular velocity $\omega(k)$ for the tank movement? There are a few ways of robotic motion by using potential fields. Three ways are given as following.

3.1 Feedforwad control using potential field

The simplest way is using the gradient vector of the potential field as the input to the tank, which forms a feedforward model. When the tank is at location $[x(k), y(k)]^T$ at time step k, we can have the gradient of the potential field at location $[x(k), y(k)]^T$ as a vector

$[\Delta_x p(x(k),y(k)), \Delta_y p(x(k),y(k))]^T$, in which

$$\Delta_x p(x(k),y(k)) = \frac{\partial p(x(k),y(k))}{\partial x}, \quad (10)$$
$$\Delta_y p(x(k),y(k)) = \frac{\partial p(x(k),y(k))}{\partial y}. \quad (11)$$

The gradient can be directly used as the input to the actuator, the length of the gradient as the linear velocity $v(k)$ and the direction of the gradient as the angular velocity $\omega(k)$. We write the input as

$$v(k) = \sqrt{\Delta_x p(x(k),y(k))^2 + \Delta_y p(x(k),y(k))^2}; \quad (12)$$
$$\omega(k) = \arctan\left(\frac{\Delta_y p(x(k),y(k))}{\Delta_x p(x(k),y(k))}\right). \quad (13)$$

This way avoids using a PD controller. If you carefully select parameters of the potential field, the tank might be pushed to the goal by the force of the gradient. But it cannot move nicely along where the gradient of the potential field directs.

3.2 Feedback control on only the orientation

A simple improved way is adding feedback control on the orientation of the tank. When the tank is in location $[x(k), y(k)]^T$ at time step k, we can have the direction of the gradient vector as the reference orientation, as in Figure 4.

$$\theta^R(k) = \arctan\left(\frac{\Delta_y p(x(k),y(k))}{\Delta_x p(x(k),y(k))}\right). \quad (14)$$

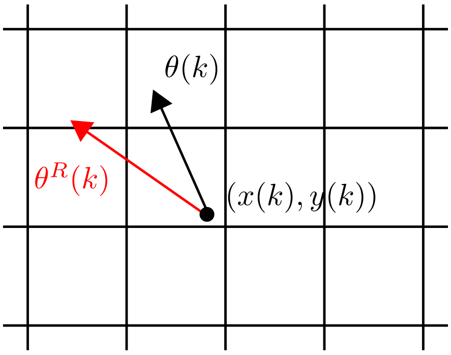


Figure 4: Get the reference orientation from the potential field.

Because we have the orientation of the tank $\theta(k)$, we can calculate the error of the orientation by

$$e(k) = \theta^R(k) - \theta(k). \quad (15)$$

By applying Equation (15) into Equation (3), we can have the PD controller for the angular velocity $\omega(k)$,

$$\omega(k) = K_p e(k) + K_d \frac{e(k) - e(k-1)}{\Delta T}. \tag{16}$$

The linear velocity $v(k)$ can either be a constant small value or the length of the gradient vector. This way only requires one state variable θ in the controller. It is easy to implement and to tune the parameter. The weakness of this way is from not considering the location information, which cannot support some complex potential fields and might be slow in final convergence.

Inherently, here we assume wherever the tank is, we can have the reference orientation from the gradient of the potential field. It implies a reference trajectory following, though the trajectory is only one dimension about the orientation.

3.3 Feedback control on the position and the orientation

We can also have a sequence of waypoints as the motion reference. The motion process then becomes sequentially visiting the waypoints. The state variables include the location.

There are a few ways of extracting waypoints. A straightforward one is that when the tank is in location $[x(k), y(k)]^T$ at time step k, a close location along the current orientation $\theta(k)$ can be selected as the reference location $[x^R(k), y^R(k)]^T$. The gradient vector of this location is then the reference orientation $\theta^R(k)$. Figure 5 illustrates an example.

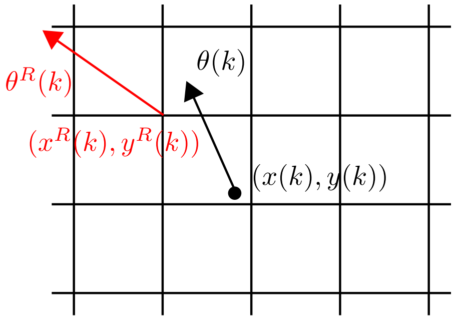


Figure 5: Get the reference orientation and position from the potential field.

Because linear velocity indicates the change of the distance and the angular velocity indicates the change of the orientation, we define the error as

$$\vec{e}(k) = \begin{bmatrix} \sqrt{(x^R(k) - x(k))^2 + (y^R(k) - y(k))^2} \\ \theta^R(k) - \theta(k) \end{bmatrix}. \tag{17}$$

By applying Equation (17) into Equation (3), we can have the PD controller for the linear velocity $v(k)$ and the angular velocity $\omega(k)$,

$$\begin{bmatrix} v(k) \\ \omega(k) \end{bmatrix} = K_p \vec{e}(k) + K_d \frac{\vec{e}(k) - \vec{e}(k-1)}{\Delta T}. \tag{18}$$

If you have the gradient vectors for a grid discretization of the map for visualization, you can also select the gradient vector of the closest position in the grid as the reference location and reference orientation. But you need to make sure that at the next step, the closet position will be changed so that there will be a new reference location and a new reference orientation. If not, the tank might be stuck with the same location as the reference.

4 Conclusion

All the ways are extracting motion reference from the potential field. By the type of the motion reference, the controller is then designed. Tuning the parameters of the controller leads the tank moving swiftly as expected. Don't let the above mentioned ways limit your creativity. You can find better ways to integrate potential field with PD controller for the planning and control of the robot motion.