



Motors and Control Systems for Precise Motion Control

In this article we discuss different types of electric motors for precise motion control (stepper motors, microstepper motors, servo motors) and the impact of the selected type of motor on a motion control system.

Introduction

Motion control is an important task in industrial automation systems. A typical motion control system comprises the following components:

- *Sensors* are used to receive feedback based on position, velocity or other characteristic;
- *Controller* is a “brain” of a control system that closes feedback loops;
- *Amplifier* translates low-power signal from a controller to drive a motor, sometimes providing a *transducer* function (e.g. PWM signal generation);
- *Actuator* (in our case – *electric motor*) is the actual device providing physical motion;
- *Transmission system* is used to convert torque and rotation speed, usually lowering speed and increasing torque.

Motion control systems can be classified on the basis of variable that is controlled (it can be, for instance, position control, velocity control or both).

Precise motion control is possible (to some extent) with the following types of motors:

1. *Stepper motors* and *microstepper motors* (it should be noted that microstepping is actually a method of controlling stepping motor).
2. *Servo motors*.

It should be noted that motor classification is based not only on the motor construction, but also on the control algorithms that are appropriate for the particular type of motor.

Stepper motors are frequently used for the task of motion control due to the control system simplicity (no complex feedback loop is required). However, they also have drawbacks that make them inappropriate in some applications: increased vibrations and lack of precision due to discrete positioning, lack of smooth motion at slow speeds and torque loss at high speeds.

Servo motors require more complex control system, but have a potentially broader range of applications.

Systems with Stepper Motors

Stepper Motors

As can be inferred from the name, a rotor in stepper motors moves itself in a step-by-step way. This

effect is achieved by sequentially turning on different stator phases in order to pull rotor to that position.

A simplified stepper motor diagram is shown on Figure 1. It can be seen that in this example there are 4 stator poles and 2 rotor poles. Real stepper motors have much more rotor poles.

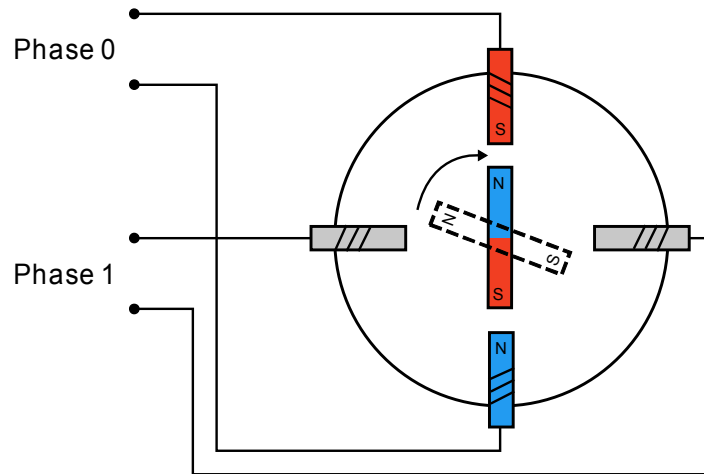


Fig. 1: A simplified diagram of a stepper motor

Stepper motors provide discrete positioning with typical steps 3.6° , 1.8° or 0.9° . Although this accuracy is sufficient for some applications, there are other drawbacks of stepping motors:

- lack of smooth motion, particularly obvious at slow speeds;
- significant loss of torque at high speeds;
- further positioning errors in the presence of static friction and/or external torque.

Some of these disadvantages can be eliminated by using a *microstepper motor* system.

Microstepping

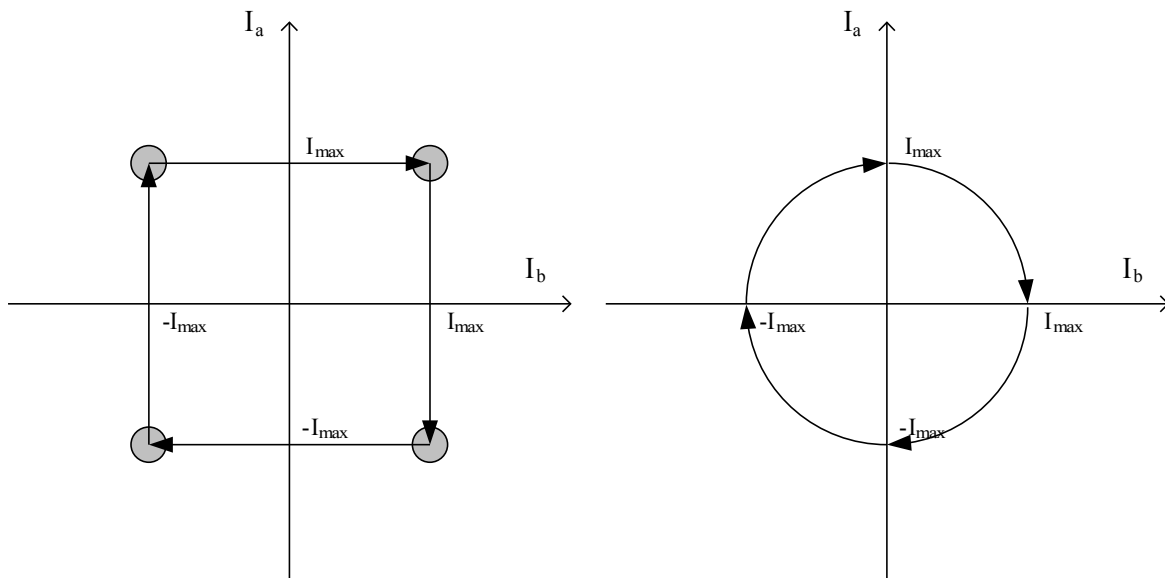


Fig. 2: Conventional stepping motor (left) and microstepping motor (right) phase diagrams

In conventional stepping motors, the currents in windings are switched instantly. Microstepping is a

technique that employs the possibility to provide partial (less than 100%) currents to the windings in order to increase position accuracy. Usually the positioning step is further divided into 256 parts, which can increase positioning accuracy (in the absence of external torque) to 0.007° and also increase motion smoothness.

The phase diagram for conventional full-stepping and microstepping are shown on Figure 2. It is assumed that we have 2 stator poles. In the case of a conventional stepper motor a current in one of the windings is instantly switched to the opposite to make a step. In the case of microstepping the winding currents are changed smoothly. The most widely used microstepping method is the so-called *sine-cosine microstepping*:

- $I_a = I_{max} \sin(\varphi)$ – current through the first winding;
- $I_b = I_{max} \cos(\varphi)$ – current through the second winding.

Ideally, by specifying φ we can set a rotor to any position between poles. In practice, however, this effect is limited due to static friction and load torque.

So, microstepping provide increased positioning accuracy and motion smoothness as compared to conventional stepping motors.

Controlling Stepper Motors

There are two ways to control a stepper motor (either in the conventional full-stepping mode or in the microstepping mode): an *open-loop control* and a *closed-loop control*.

An *open-loop control system* (i.e. without sensors) is most widely used for stepper motors. It is based on the fact that we set the position of the rotor by setting the winding currents, so normally we don't need any sensors to know rotor's position.

The main advantage of the open-loop control system is its simplicity. However, it is vulnerable to a step skipping (when the rotor doesn't move itself to the desired position in time before the next step begins) and a loss of precision due to the external factors such as static friction, which is significant in microstepping mode.

The *closed-loop control system* for stepper motor eliminates the risk of step skipping. Closed-loop system uses some sort of rotary encoder to know actual position of the rotor.

Systems with Servo Motors

Servo motors are characterized by their ability to any position (limited only by the sensitivity of the sensors). It uses several stator phases to arbitrarily set magnetic flux.

For example, in order to set a magnetic flux in the three-phase servo motor one need to flow the following currents through the stator windings:

- $I_a = I_0 \sin(\theta)$,
- $I_b = I_0 \sin(\theta - 120^\circ)$,
- $I_c = I_0 \sin(\theta + 120^\circ)$,

where θ is an *electric angle*, which can be calculated using the following formula:

$$\theta = K\varphi,$$

where φ is a mechanic angle and K is a *reduction ratio* equal to the number of rotor poles halved.

In order to produce a motion (not just hold a rotor in a static position), there must be an angle between the rotor's and stator's fields. To maximize the efficiency factor this angle should be close to 90° :

$$\theta = K\varphi + 90^\circ.$$

In this case, I_0 directly influences the torque.

For example, consider a simplified motor diagram on Figure 3.

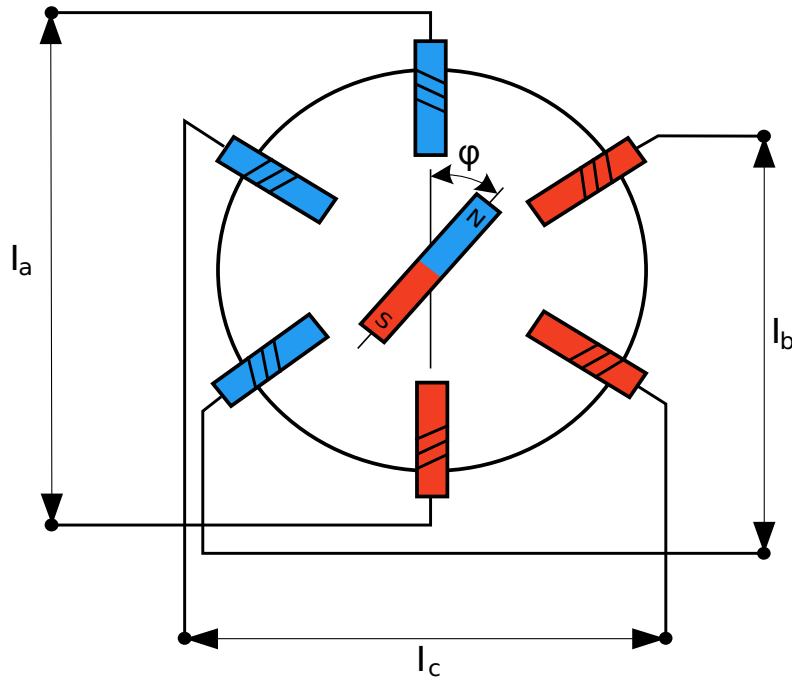


Fig. 3: A simplified diagram of a servo-motor (three phases, two poles)

Since $K = 1$, currents are in this case calculated using $\theta = \varphi + 90^\circ$, i.e.:

- $I_a = I_0 \sin(\varphi + 90^\circ)$,
- $I_b = I_0 \sin(\varphi - 30^\circ)$,
- $I_c = I_0 \sin(\varphi + 210^\circ)$.

In order to rotate in the opposite direction, I_0 should be negative.

PID Algorithm for Controlling Servo Motors

Servo motors are controlled with *closed-loop* systems, so, the position and/or velocity sensors are needed. Rotary encoders or resolvers can be used for this purpose. Precise angle sensors are usually constructed as sine-cosine sensors with rotating transformers.

The *PID algorithm* (proportional-integral-derivative) is widely used in industrial control systems. The *PID controller* observes the positioning error (difference between the setpoint and the actual position) and tries to eliminate this difference. In order to provide a smooth motion one needs simply to change the setpoint smoothly.

Let $e(t)$ be a positioning error. Then the I_0 current can be calculated using the formula:

$$I_0(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{\partial e}{\partial t}.$$

K_p , K_i and K_d are proportional, integral and differential coefficients respectively. Each of them plays an important role:

1. A *proportional term* is a direct response to the positioning error, trying to pull a motor to the setpoint. However, too big K_p can decrease system stability and lead to overshoot.

2. An *integral term* eliminates static error (due to static friction, external torque or other reasons) by constantly increasing I_0 until the positioning error is 0. Too big K_i is even worse than too big K_p and leads to overshoot and oscillation.
3. A *differential term* reduces overshoot and oscillation by countering the rapid position changes. Too big differential term will increase motor's response time and reduce stability by amplifying noise.

In practical systems a few additional issues should be addressed:

- $e(t)$ going to the input of the controller must be limited, especially for the integrating circuit (otherwise too big $e(t)$ will make a system unstable).
- $e(t)$ going to the input of the *differentiator circuit* must be filtered to decrease noise (in the simplest case, a lowpass IIR filter can be used for this purpose).
- If one needs to provide a precise *motion* (not only positioning), a PID controller can be combined with an open-loop controller for better results.

Conclusion

Stepper motors and servo motors can be used for precise motion control. A stepper motor is a cheaper solution which doesn't require complex control algorithms. On the other hand, servo motors are more powerful and can provide more smooth and precise motion and positioning.

1-CORE Technologies is a leading Russian electronics design company specializing primarily in the fields of digital communications, digital signal processing and automatic control.

Address: 24 Radio str., Moscow, Russia, 105005

Tel.: +7 (495) 921-79-91

Web: <http://www.1-core.com>

E-mail: sales@1-core.com