Stepper Motor

Stepper motor suitable for whole-, half- and micro-stepping representation

Library: Simscape / Electrical / Electromechanical / Reluctance & Stepper

Description

The Stepper Motor block represents a stepper motor. It uses the input pulse trains, A and B, to control the mechanical output according to the following equations:

$$e_A = -K_m \omega \sin(N_r \theta)$$

$$e_B = K_m \omega \cos(N_r \theta)$$

$$\frac{di_A}{dt} = (v_A - Ri_A - e_A)/L$$

$$\frac{di_B}{dt} = (v_B - Ri_B - e_B)/L$$

$$J\frac{d\omega}{dt} + B\omega = T_e$$

$$T_e = -K_m\left(i_A - \frac{e_A}{R_m}\right)\sin(N_r\theta) + K_m\left(i_B - \frac{e_B}{R_m}\right)\cos(N_r\theta) - T_d\sin(4N_r\theta)$$

$$\frac{d\theta}{dt} = \omega$$

where:

- e_A and e_B are the back electromotive forces (emfs) induced in the A and B phase windings, respectively.
- i_A and i_B are the A and B phase winding currents.
- v_A and v_B are the A and B phase winding voltages.
- K_m is the motor torque constant.
- N_r is the number of teeth on each of the two rotor poles. The **Full step size** parameter is $(\pi/2)/N_r$.
- R is the winding resistance.
- L is the winding inductance.
- R_m is the magnetizing resistance.
- B is the rotational damping.
- J is the inertia.
- ω is the rotor speed.
- Θ is the rotor angle.
- T_d is the detent torque amplitude.

- T_e is the electrical torque.

If the initial rotor is zero or some multiple of (π/2)/N_r, the rotor is aligned with the phase winding of pulse A. This happens when there is a positive current flowing from the **A+** to the **A-** ports and there is no current flowing from the **B+** to the **B-** ports.

Use the Stepper Motor Driver block to create the pulse trains for the Stepper Motor block.

The Stepper Motor block produces a positive torque acting from the mechanical **C** to **R** ports when the phase of pulse A leads the phase of pulse B.

Averaged Mode

If you set the **Simulation mode** parameter to `Averaged`, both for a Stepper Motor block and for the Stepper Motor Driver block that controls it, then the individual steps are not simulated. This can be a good way to speed up simulation. In Averaged mode, under nonslipping conditions, the motor and driver are represented by a second-order linear system that tracks the specified step rate. The demanded step rate is determined directly from voltage across **A+** and **A-**. So, for example, a voltage of +10 V across the **A+** and **A-** terminals is interpreted as a step rate demand of 10 steps per second. See the Stepper Motor Driver block reference page for more information on how to connect the driver block to your step angle controller.

Averaged mode includes a slip estimator to predict whether the stepper motor would have slipped if running in Stepping simulation mode. Slip is predicted if the motor torque exceeds the **Vector of maximum torque values** parameter value for longer than one step period, the step period being determined from the current step rate demand. Upon detecting slip, the simulation will proceed or stop with an error, according to the **Action on slipping** parameter value. If you choose the action that lets the simulation continue, note that simulation results may be incorrect. When slipping occurs, the torque generated by the motor is not generally the maximum available torque; the maximum torque is only achieved if the stepper controller detects slip and adjusts the step rate command accordingly.

The dynamics of the equivalent second-order system are determined from the values that you specify for the **Approximate total load inertia** and **Maximum step rate command** parameters. It is important that you set as accurate values as possible for these parameters, so that the step rate command is tracked, and the block does not generate false slipping warnings or errors.

If you run the motor in Averaged mode with the optional thermal ports exposed (see Thermal Ports and Effects), then heat is added to the thermal ports, assuming that the windings are always powered even when the step rate command is zero. The block makes adjustments for half stepping and for reduced torque (and winding currents) at higher speeds. For these adjustments to be correct, the **Vector of maximum torque** parameter values must be correct. For half stepping, at zero speed the heat generated by the block is the average of that generated when stopped at a half step and at a full step.

To validate Averaged mode model configurations where you predict slip to occur, compare results with the same simulation performed in stepping mode.

Thermal Ports and Effects

The block has three optional thermal ports, one for each of the two windings and one for the rotor. These ports are hidden by default. To expose the thermal ports, right-click the block in your model, and then from the context menu select **Simscape** > **Block choices** > **Show thermal port**. This action displays the thermal ports on the block icon, and adds the **Temperature Dependence** and **Thermal Port** tabs to the block dialog box. These tabs are described further on this reference page.

Use the thermal ports to simulate the effects of copper resistance and iron losses that convert electrical power to heat. If you expose these ports, winding resistance is assumed linearly dependent on temperature, and is given by:

$$R = R_0\,(1 + \alpha\,(T - T_0)) \qquad (1)$$

where:

- R is the resistance at temperature T.
- R_0 is the resistance at the measurement (or reference) temperature T_0. Specify the reference temperature using the **Measurement temperature** parameter.

- α is the resistance temperature coefficient, which you specify with the **Resistance temperature coefficients, [alpha_A alpha_B]** parameter. A typical value for copper is 0.00393/K.

The block calculates temperature of each of the windings and the rotor by

$$M\frac{dT}{dt} = Q$$

where

- M is the thermal mass. Specify this value for the windings using the **Winding thermal masses, [M_A M_B]** parameter, and for the rotor using the **Rotor thermal mass** parameter.
- T is the temperature. Specify the initial values for the windings using the **Winding initial temperatures, [T_A T_B]** parameter, and for the rotor using the **Rotor initial temperature** parameter.
- Q is the heat flow, which is calculated from the iron losses of the windings:

  $$Q_A = {i_a}^2R_A(1 - \rho_m/100)$$
  $$Q_B = {i_B}^2R_B(1 - \rho_m/100)$$
  $$Q_R = Q_A(\rho_m/100) + Q_B(\rho_m/100)$$

  where ρ_m is the percentage of magnetizing resistance associated with the rotor. Specify this percentage using the **Percentage of magnetizing resistance associated with the rotor** parameter.

Predefined Parameterization

There are multiple available built-in parameterizations for the Stepper Motor block.

This pre-parameterization data allows you to set up the block to represent a specific supplier component. To load a predefined parameterization, click on the "Select a predefined parameterization" hyperlink in the Stepper Motor block mask and select the specific part you want to upload from the list of available components.

> **Note**
>
> Predefined parameterizations of Simscape components use available data sources for supplying parameter values. Engineering judgement and simplifying assumptions are used to fill in for missing data. As a result, deviations between simulated and actual physical behavior should be expected. To ensure requisite accuracy, you should validate simulated behavior against experimental data and refine component models as necessary.

Assumptions and Limitations

The model is based on the following assumptions:

- This model neglects magnetic saturation effects and any magnetic coupling between phases.
- When you select the **Start simulation from steady state** check box in the Simscape™Solver Configuration block, this block will not initialize an **Initial rotor angle** value between –π and π.
- To use Averaged mode, the Stepper Motor block must be directly connected to a Stepper Motor Driver block also running in Averaged mode.
- The Averaged mode is an approximation, and exact step tracking compared to the Stepping mode should not be expected.
- Slip detection in Averaged mode is approximate, and depends on a good estimate for load inertia and maximum step rate. Incorrect values may result in false slip detection.
- When simulating slip in Averaged mode, it is assumed that the stepper motor controller adjusts the step rate command so as to achieve maximum possible torque.

Ports

Conserving

expand all

`A+` — A-phase positive terminal
electrical

`A-` — A-phase negative terminal
electrical

`B+` — B-phase positive terminal
electrical

`B-` — B-phase negative terminal
electrical

`C` — Machine case
mechanical rotational

`R` — Machine rotor
mechanical rotational

`HA` — Winding A thermal mass
thermal

`HB` — Winding B thermal mass
thermal

`HR` — Rotor thermal mass
thermal

Parameters

expand all

Electrical Torque

`Simulation mode` — Simulation mode
`Stepping` (default) | `Averaged`

`Vector of rotational speeds` — Rotor speeds
`[0, 1000, 3000] rpm` (default) | positive vector

`Vector of maximum torque values` — Maximum torques
`[2, 2, 1] N*m` (default) | positive vector

`Action on slipping` — Slip response
`none` (default) | `warn` | `error`

`Approximate total load inertia` — Total load inertia
`1e-4 kg*m^2` (default) | positive number

`Maximum step rate command` — Maximum command frequency
`10 Hz` (default) | positive number

`Phase winding resistance` — Phase resistance
`0.55 Ohm` (default) | positive number

`Phase winding inductance` — Phase inductance
`1.5e-3 Ohm` (default) | positive number

`Motor torque constant` — Torque constant
`0.19 N*m/A` (default) | positive number

`Detent torque` — Torque variation amplitude
`0 N*m` (default) | positive number

`Magnetizing resistance` — Phase magnetizing resistance
`inf` (default) | strictly positive number

`Full step size` — Step size
`1.8 deg` (default) | positive number

Mechanical

`Rotor inertia` — Rotational inertia
`4.5e-5 kg*m^2` (default) | zero or positive number

`Rotor damping` — Rotational damping
`8.0e-4 N*m/(rad/s)` (default) | zero or positive number

`Initial rotor speed` — Initial speed
`0 rpm` (default) | real number

`Initial rotor angle` — Initial angle
`0 rad` (default) | real number

Temperature Dependence

This set of parameters appears only for blocks with exposed thermal ports. For more information, see Thermal Ports and Effects.

`Resistance temperature coefficients, [alpha_A alpha_B]` — Temperature coefficients
`[.00393, .00393] 1/K` (default) | positive vector

`Measurement temperature` — Reference temperature
`25 degC` (default) | real number

Thermal Port

This set of parameters appears only for blocks with exposed thermal ports. For more information, see Thermal Ports and Effects.

`Winding thermal masses, [M_A M_B]` — Winding thermal masses
`[100, 100] J/K` (default) | positive vector

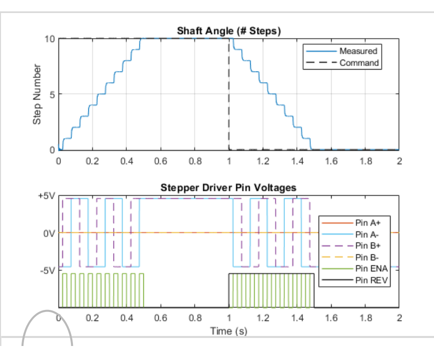
`Winding initial temperatures, [T_A T_B]` — Initial winding temperatures
`[25, 25] degC` (default) | real number

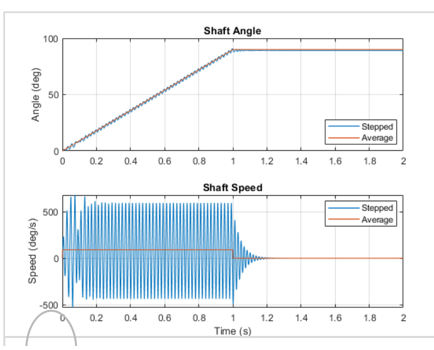
`Rotor thermal mass` — Rotor thermal mass
`50 J/K` (default) | positive number

`Rotor initial temperature` — Initial rotor temperature
`25 degC` (default) | real number

`Percentage of magnetizing resistance associated with the rotor` — Magnetizing resistance rotor percentage
`90` (default) | positive number between `0` and `100`

Model Examples





Stepper Motor with Control

How to use the Stepper Motor Driver and Stepper Motor blocks together to implement a controlled permanent magnet stepper motor. The model

Stepper Motor Averaged Mode

The Stepper Motor simulating in Stepping and Averaged simulation modes. The purpose of Averaged mode is faster simulation for any

References

[1] M. Bodson, J. N. Chiasson, R. T. Novotnak and R. B. Rekowski. "High-Performance Nonlinear Feedback Control of a Permanent Magnet Stepper Motor." IEEE Transactions on Control Systems Technology, Vol. 1, No. 1, March 1993.

[2] P. P. Acarnley. *Stepping Motors: A Guide to Modern Theory and Practice*. New York: Peregrinus, 1982.

[3] S.E. Lyshevski. *Electromechanical Systems, Electric Machines, and Applied Mechatronics*. CRC, 1999.

Extended Capabilities

C/C++ Code Generation
Generate C and C++ code using Simulink® Coder™.

See Also

Stepper Motor Driver | Unipolar Stepper Motor

Introduced in R2008a