cādence®

Brushless DC Motor Model

Although PSpice is designed as an electronic circuit simulator, you can also use it to simulate mechanical or electromechanical systems. Analog Behavioral Modeling makes simulating mechanical systems much simpler. An example of an electromechanical system which can benefit from PSpice simulation is a Brushless DC motor.

Although PSpice is designed as an electronic circuit simulator, you can also use it to simulate mechanical or electromechanical systems. Analog Behavioral Modeling makes simulating mechanical systems much simpler. An example of an electromechanical system which can benefit from PSpice simulation is a Brushless DC motor.

Brushless DC motors are used in computer disk drives and other applications where precise control of motor operation is required. A brushless DC motor is built like a stepping motor. It has a permanent magnet rotor attached to the motor shaft, and several electromagnets arranged around the stator. Each of these electromagnet windings must be driven in sequence to make the motor shaft turn, a process called commutation. Commutation must be synchronized with the motor shaft angle to make the motor turn at the desired speed and direction. Implementing a commutation strategy and motor control system usually requires both analog and digital circuit elements.

Because the motor is part of this closed loop control system, it is important to have an accurate model of its mechanical and electrical behavior.

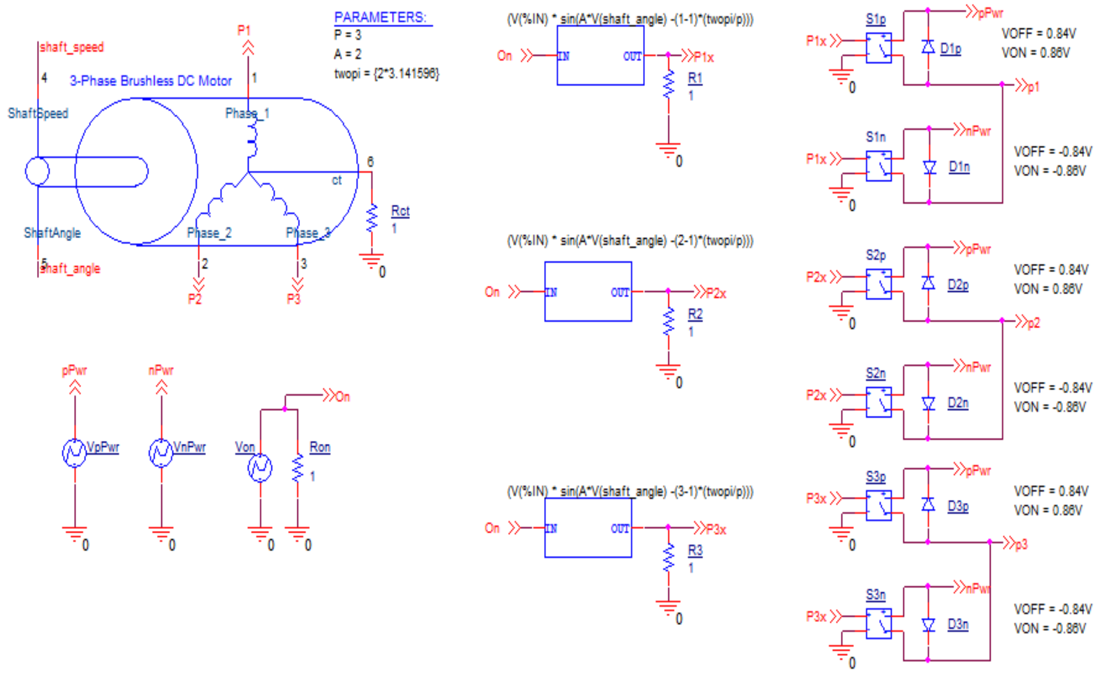

Figure 1: Brushless DC Motor Model Circuit

Modeling the Mechanical Properties

Note: The equations used in this document to explain motor behavior are from the book *Brushless Motor System Design and Analysis (Dr. C.K. Taft, Dr. R.G. Gauthier, & Dr. T.J. Harned, "Brushless Motor System Design and Analysis", University of New Hampshire Press 1988).*

The basic equation which describes the mechanical system is:

$Ttotal = J \cdot d\ 2\theta/dt\ 2$

Where,

- $Ttotal$ is the total torque, including friction, applied to the motor shaft from all sources (g.cm)
- J is the mechanical system moment of inertia (g.m.sec^2)
- θ is the motor angle shaft (radians)

$Ttotal$ can also be expressed as:

$Ttotal = 2\pi JdS/dt$

Where, S is the shaft speed (rev/sec) represented by the equation:

$S = Cd\upsilon/dt$

You can implement the equation for $Ttotal$ expressed in terms of S by modeling torque as a current and the moment of inertia, $2\pi J$, as a capacitor. This gives the shaft speed as the voltage across the capacitor.

Modeling torque as a current and moment of inertia as a capacitor is convenient because you can model any additional mechanical system moment of inertia as an additional capacitor in parallel with the first one. Also, you can add various torque and drag forces as parallel current sources making the model easier to use in a system. You can use the equation for the shaft speed again on the circuit equation for a capacitor to give the shaft angle as the voltage across a capacitor which has a current equal to the shaft speed applied to it. Here is a PSpice sub-circuit for the motor's mechanical system:

```
.subckt motor_mech shaft_speed shaft_angle
+ params:
+ J= .30 ; moment of inertia of rotor (g*cm*sec*sec)
+ twopi = {2 * 3.141596}
Cmotor shaft_speed 0 {J*twopi} ; Inertia
Gintegrate 0 shaft_angle_intg VALUE = {V(shaft_speed)}
Cintegrate 0 shaft_angle_intg {1/twopi} IC=0.0
Rdummy2 0 shaft_angle_intg 1e12 ; (otherwise floating)
Ecopy shaft_angle 0 VALUE = {V(shaft_angle_intg)}
; Copy the voltage
Rdummy3 shaft_angle 0 1 ; Make sure there is a load
.ends
```

To use the model, apply a current proportional to the shaft torque between nodes `SHAFT_SPEED` and `0 (1 amp = 1 g.cm)`. The voltage on `SHAFT_SPEED` will correspond to the shaft speed `(1 volt = 1 rev/sec)`, and the voltage on `SHAFT_ANGLE` will be the shaft angle `(1 volt = 1 radian)`.

You now need to model the mechanical losses of the motor. The simplest are the linear losses, damping and eddy current losses.

They are described by the equation:

$Tdamping = 2\pi(B \cdot S)$

- B is the damping and eddy current losses (g·cm·sec/rad)
- S is the shaft speed (rev/sec)

Translating to the specified model units, you get:

$$\mathbf{I(SHAFTSPEED) = 2\pi(B \cdot V)(SHAFTSPEED)}$$

This is just the equation for a resistor attached to node `SHAFT_SPEED` and ground, with a value of `1/(2πB)`.

```
.PARAM B=.36;Damping and eddy current losses (g*cm*sec/rad)
Reddy shaft_speed 0 {1/(B*twopi)}
```

Another mechanical loss is the frictional loss. This loss is a fixed torque which opposes the direction of rotation. To model this loss, use a table to specify the shape of the loss curve, and an Analog Behavioral Modeling current source to multiply the loss curve by the loss factor *F (g·cm).*

```
.PARAM F = .72 ; Friction losses (constant
torque loss) (g*cm)
Gdrag shaft_speed ld1 VALUE = {F * V(drag)} nonlinear drag
Ldummy1 ld1 0 100mH ; force timestep control
Edrag2 drag 0 TABLE {V(shaft_speed)}=(-.001, -1) (.001, 1)
Rdummy1 drag 0 1
```

The `Ldummy1` inductor is inserted in series with the Gdrag current source to help PSpice do timestep size control. Since inductor is in series with the current source it has no effect on the current (torque) produced by Gdrag. But the voltage across it reflects the derivative of the current and allows PSpice to do timestep size control on that derivative. When using the Analog Behavioral Models, it is often a good idea to place an inductor in series with a controlled current source, or a capacitor in parallel with a controlled voltage source, to help PSpice with timestep control.

Another torque is the magnetic detent torque which tends to align the rotor magnetic poles with the stator poles. This torque is periodic, and is described by the equation:

$$Tdetent = D \cdot \sin(Nd \cdot A \cdot \theta)$$

Where,

- *D* is the magnetic detent torque (g·cm)
- *A* is the number of north poles on the rotor
- *Nd* is an integer determined by the number of stator slots and the structure of the rotor

You can translate this directly into a behavioral modeling current source:

```
.PARAM D = 2.9 ; Magnetic detent torque (g*cm)
.PARAM A = 2 ; number of north poles on the rotor
.PARAM P = 3 ; number of stator phases
Gdetent shaft_speed ld2 VALUE={D *sin(2*A*P*V(shaft_angle))}
```

All of these together give you the following model for the mechanical part of the motor:

```
 .subckt motor_mech shaft_speed shaft_angle
+ params:
+ J= .30 ; moment of inertia of rotor (g*cm*sec*sec)
+ B= .36 ; Damping and eddy current losses
; (linear torque with speed) (g*cm*sec/rad)
+ F= .72 ; Friction/drag losses (constant torque losses)
; +(g*cm)
+ D= 2.9 Magnetic detent torque (g*cm)
+ A= 2 ; Number of north poles on the rotor
+ P= 3 ; Number of phases (if you change this you need
; to add more windings to the motor subckt.)
+ twopi = {2 * 3.141596}
Cmotor shaft_speed 0 {J*twopi} ; Inertia
Reddy shaft_speed 0 {1/(B*twopi)} ; Linear losses
Gdrag shaft_speed ld1 VALUE = {F * V(drag) ; non-linear drag
Ldummy1 ld1 0 100mH ; force timestep control
Gdetent shaft_speed ld2 VALUE = {D * sin(2*A*P*V(shaft_angle))}
; detent
Ldummy2 ld2 0 100mH ; force timestep control
Edrag2 drag 0 TABLE {V(shaft_speed)} = (-.001, -1) (.001, 1)
Rdummy1 drag 0 1
Gintegrate 0 shaft_angle_intg VALUE = {V(shaft_speed)}
Cintegrate 0 shaft_angle_intg {1/twopi} IC=0.0
Rdummy2 0 shaft_angle_intg 1e12 ; (otherwise floating)
Ecopy shaft_angle 0 VALUE = {V(shaft_angle_intg)
; Copy the voltage
Rdummy3 shaft_angle 0 1 ; Make sure there is a load
.ends
```

Modeling the Electrical Properties

Now you need to model the electrical properties of the stator windings. The properties which are required for a first order model are:

- Winding inductance
- Winding resistance
- Winding capacitance
- Winding mutual inductance
- Winding back EMF
- Torque on the rotor from the winding current

The first four of these are simple electrical properties of the winding which are modeled directly by PSpice. The last two require a behavioral model. Dr. Taft et al provide the following equations for back emf and torque:

$$V_{bn} = C_b \cdot S \cdot sin\ (A \cdot \theta - (N - 1)2\ \pi / P)$$

$$T_{dn} = C_t \cdot i_n \cdot sin(A \cdot \theta - (N - 1)2\ \pi / P)$$

Where,

- V_{bn} is the back EMF voltage for the phase n winding
- C_b is the back emf voltage constant (volts·sec/rev)
- T_{dn} is the drive torque from the phase n winding
- C_t is the torque constant (g·cm/amp)
- i_n is the current in the phase n winding (amp)
- S is the shaft speed (rev/sec)
- A is the number of north poles on the rotor

- N is the phase number (1, 2, 3 in this example)
- P is the number of motor phases

Keeping in mind that the sine terms of the equations are the same, and adding the other four electrical properties of the motor windings, you come up with the following model for the motor:

```
*The motor with both ends of each coil available.
  *
  * Phase 3 coil ----------------+---+
  * Phase 2 coil --------+---+ | |
  * Phase 1 coil +---+ | | | |
  * | | | | | |
  .subckt bldcmtr p1a p1b p2a p2b p3a p3b shaft_speed shaft_angle
  + params:
  + J= .30 ; moment of inertia of rotor (g*cm*sec*sec)
  + B= .36 ; Damping and eddy current losses
  * (linear torque with speed) (g*cm*sec/rad)
  + F= .72 ; Friction/drag losses (constant torque losses)
  * (g*cm)
  + D= 2.9 ; Magnetic detent torque (g*cm)
  + A= 2 ; Number of north poles on the rotor
  + P= 3 ; Number of phases (if you change this you need
  * to add more windings to the motor subckt.)
  + CL=3mh ; winding inductance (Henrys)
  + CR=6ohm ; winding resistance (Ohms)
  + CC=.001uf ; winding capacitance to ground (Farads)
  + CM=.5 ; adjacent winding mutual coupling factor
  + Cb=.12 ; Back EMF constant (Volt*sec/rev)
  + Ct=300 ; Torque constant (g*cm/amp)
  + twopi = {2 * 3.141596}
  * Model each winding. The inductor must be here so we can include
  * mutual inductance. The other effects are modeled in
  * motor_winding
  Lwinding1 p1a p1x {CL}
  R_snub_1 p1a p1x {1K*twopi*CL}
  * ; snubbing resistor to limit coil Q
  x1 p1x p1b shaft_speed shaft_angle motor_winding
  + params: N=1 A={A} P={P} CL={CL} CR={CR} CC={CC}
  + CM={CM} Cb={Cb} Ct={Ct} twopi={twopi}
  Lwinding2 p2a p2x {CL}
  R_snub_2 p2a p2x {1K*twopi*CL}
  * ; snubbing resistor to limit coil Q
  x2 p2x p2b shaft_speed shaft_angle motor_winding
  + params: N=2 A={A} P={P} CL={CL} CR={CR} CC={CC}
  + CM={CM} Cb={Cb} Ct={Ct} twopi={twopi}
  Lwinding3 p3a p3x {CL}
  R_snub_3 p3a p3x {1K*twopi*CL}
  * ; snubbing resistor to limit coil Q
  x3 p3x p3b shaft_speed shaft_angle motor_winding
  + params: N=3 A={A} P={P} CL={CL} CR={CR} CC={CC}
  + CM={CM} Cb={Cb} Ct={Ct} twopi={twopi}
  * Model the mutual inductance here.
  * (For three phase, all windings are adjacent to each other.)
  k1 Lwinding1 Lwinding2 {Cm}
```

```
k2 Lwinding2 Lwinding3 {Cm}
k3 Lwinding3 Lwinding1 {Cm}
* Model the motor mechanical system.
x4 shaft_speed shaft_angle motor_mech
+ params: J={J} B={B} F={F} D={D} A={A} P={P} twopi={twopi}
.ends
*
* The motor winding
*
* This models the electrical properties of the windings,
* and creates the torque "current" which is delivered to
* the mechanical model.
* Mutual inductance is modeled in the motor subcircuit,
* so the inductance must be there also. The inductance
* must be in series with this model.
.subckt motor_winding winding1 winding2 shaft_speed shaft_angle
+ params: N=1 A=2 P=3 CL=3mh CR=6ohm CC=.001uf CM=.5 Cb=.12 Ct=300
+ twopi={2*3.141596}
* The electrical model: backemf, resistance, and capacitance
Ebackemf winding1 3 VALUE = {Cb * V(shaft_speed) * V(factor)}
Vsense 3 4 0v ; measure winding current
Rwinding 4 winding2 {CR}
* Place half the winding capacitance at each end of the winding
C1 winding1 0 {CC/2}
C2 winding2 0 {CC/2}
* The mechanical model: torque created by this winding
Gtorque 0 shaft_speed VALUE = {Ct * I(Vsense) * V(factor) }
* The shaft angle function for this phase.
Efactor factor 0 VALUE = + {sin(A*V(shaft_angle) - (N-1)*(twopi/P))}
Cdummy factor 0 10uf ; force timestep control
.ends
```

Implementing a Commutation Strategy - Testing the Model

To test the motor model you need to implement a commutation strategy. You will use a simple strategy which works like electronic brushes and drives only one stator phase at a time. To do this, take the sine of the shaft angle with respect to each stator phase (p1x, p2x, and p3x). The sine wave is used to control switches for each phase, so that voltage is applied only to the stator phase that has the highest torque-to-current ratio. There are many other commutation strategies which can be used. The phases can be connected in a Y or delta, with two or more of the terminals connected to a supply or ground during each commutation interval.

```
* A test circuit for the motor
  .param twopi = {2*3.141596}
  .param P = 3 ; the number of phases
  .param A = 2 ;the number of north poles on the rotor
  * Connect one end of each phase winding to ct.
  x1 p1 ct p2 ct p3 ct shaft_speed shaft_angle bldcmtr
  + params: J=.30 B=.36 F=.72 D=2.9 A= {A} P= {P} CL=3mh CR=6ohm CC=.1pf
  + CM=.5 Cb=.12 Ct=300 twopi={twopi}
  rct ct 0 1 ;hook ct to ground through current measuring resistor
  * Make some brushes
  Ep1x p1x 0 VALUE = {V(on) * sin(A*V(shaft_angle) - (1-1)*(twopi/P))}
```

```
Ep2x p2x 0 VALUE = {V(on) * sin(A*V(shaft_angle) - (2-1)*(twopi/P))}
Ep3x p3x 0 VALUE = {V(on) * sin(A*V(shaft_angle) - (3-1)*(twopi/P))}
r1 p1x 0 1
r2 p2x 0 1
r3 p3x 0 1
S1p ppwr p1 p1x 0 switchp
S1n npwr p1 p1x 0 switchn
S2p ppwr p2 p2x 0 switchp
S2n npwr p2 p2x 0 switchn
S3p ppwr p3 p3x 0 switchp
S3n npwr p3 p3x 0 switchn
* 5v to drive, 0v to brake
Vppwr ppwr 0 PWL (0 5v .9 5v .901 0v 2s 0v)
Vnpwr npwr 0 PWL (0 -5v .9 -5v .901 0v 2s 0v)
* Clamping diodes to keep the kickback voltage down
D1p p1 ppwr dmod
D1n npwr p1 dmod
D2p p2 ppwr dmod
D2n npwr p2 dmod
D3p p3 ppwr dmod
D3n npwr p3 dmod
.model switchp vswitch (RON = .1 ROFF = 1e5 VON= .86 VOFF= .84)
.model switchn vswitch (RON = .1 ROFF = 1e5 VON=-.86 VOFF=-.84)
.model dmod D (RS = 10)
* "on" is used to enable the "brushes": 0 disconnects, 1 connects
* brushes to power.
Von on 0 PWL( 0,0 10ms,0 20ms,1 .8s,1 .81s,0 .9s,0 .91s,1)
ron on 0 1
.watch tran V([Shaft_Speed])
.tran 10ms 2s
.probe
.options acct
.end
```

Verifying Operation - Simulating the Model

Remember that simulating the motor takes lots of computer time (15 minutes on a 486-33 MHz PC), and using the .PROBE command to save all the internal voltages and currents takes up lots of disk space `(2.5 Mb)`. The simulation run applies `5v` to the motor brushes from 20ms to 0.8s. During this time the motor accelerates to about 25rev/sec. Power is disconnected from `0.8s` to `0.9s`, and the motor slows down due to friction losses. From `0.9s` to the end of the simulation, the motor brushes are connected to `0v` making the motor slow quickly, as the back EMF provides reverse torque.

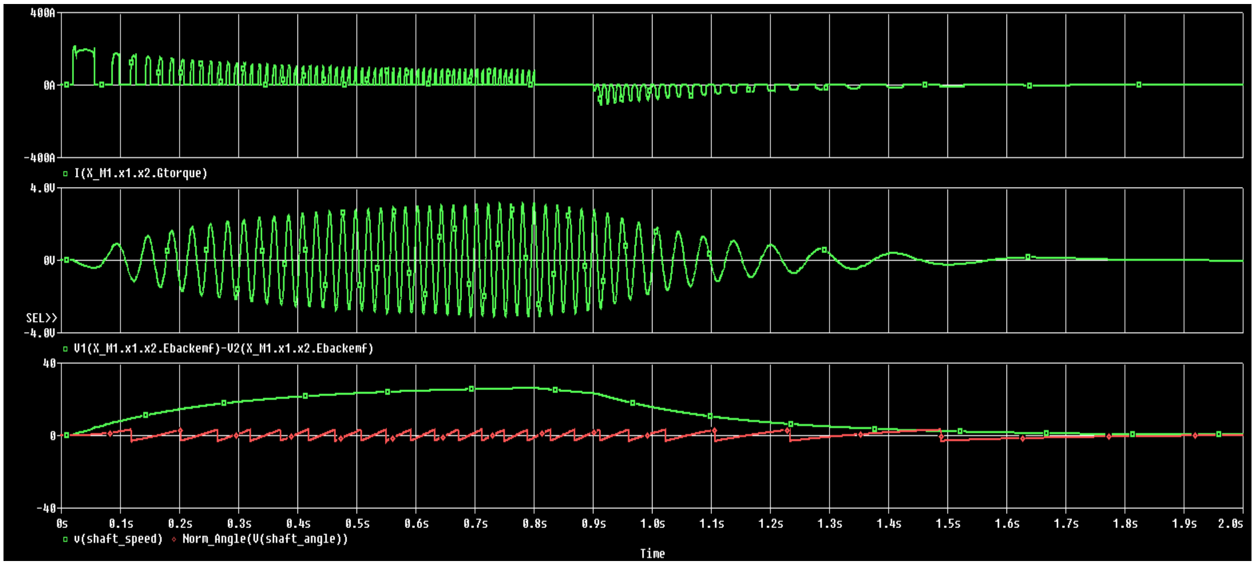

Figure 2: Simulation Results

In Figure 2, the top plot shows the motor torque delivered by the phase 2 stator winding. The middle plot shows the back EMF voltage for the phase 2 winding. Notice that as the motor speed increases, the back EMF increases and the torque decreases. The bottom trace shows the motor shaft speed `(1volt = 1rev/sec)` and the motor shaft angle `(1volt = 1radian)`. The shaft angle has been normalized to the range from -π to +π by the Probe macro Norm_Angle. The `Norm_Angle` macro is defined as:

```
Norm_Angle(a) = 2 · atan(sin(a/2))/cos(a/2))
```

Note: If the shaft angle is not normalized to the range from -π to +π, the shaft angle increases by 2π for each revolution.

A Brushless DC motor is a relatively complex electromechanical system to simulate in detail. But PSpice's Analog Behavioral Modeling feature allows you to model it in enough detail to verify the operation of an electronic motor control system.

Tip: The model described is included in the Model Library, misc.lib.

1766 12/13 CY/DM/PDF