

Title: MultiFlex ETH 1000 Series Brushless/Linear AC Motor Commutation
Products(s): All MultiFlex ETH 1400, 1440, 1800, 1840 motion controller models
Keywords: sine, commutation, AC motor, linear motor
ID#: TN1070
Date: Nov. 12, 2008

Summary

PMC's MultiFlex ETH 1000 Series Ethernet motion controller models MultiFlex ETH 1400, 1400, 1800 and 1840 can provide direct AC sine commutation for brushless AC rotary and linear motors. This tech note describes AC sine commutation and the procedure for setting up and initializing the commutation of brushless and linear motors.

More Information

The following Figure 1 illustrates the basic components of a brushless AC motor. Three coils are mounted 120 degrees apart. When current is applied to a coil it will attract the nearest rotor magnet. If current is applied to the phase U coil, the rotor will turn clockwise until magnet **A** is aligned with the coil (at a detent). This simple brushless motor is similar to a how a stepper motor operates, turning on one coil will cause the shaft to rotate by one magnet pole.

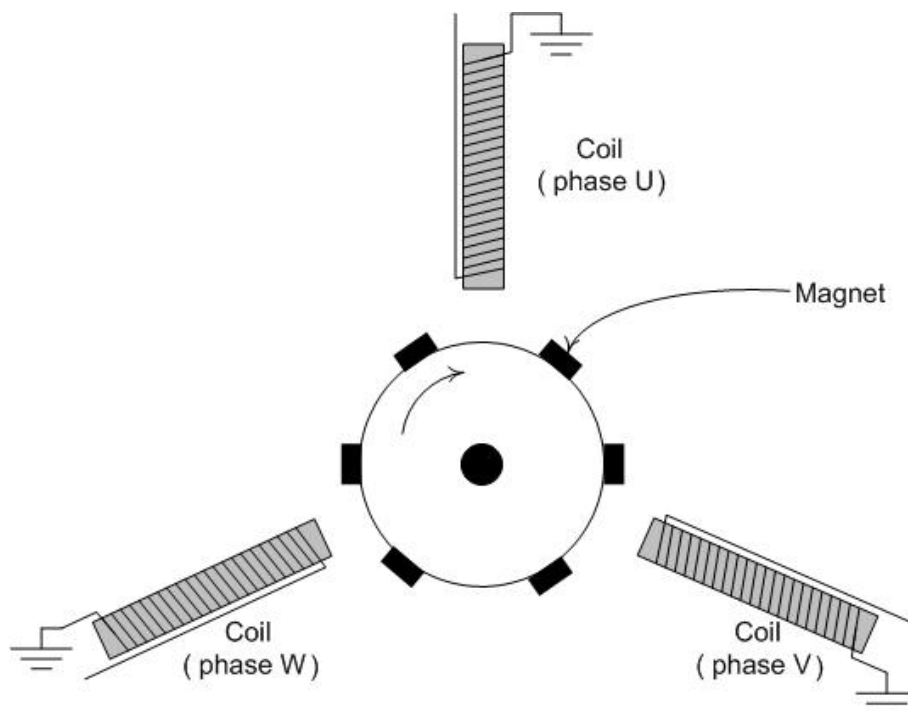
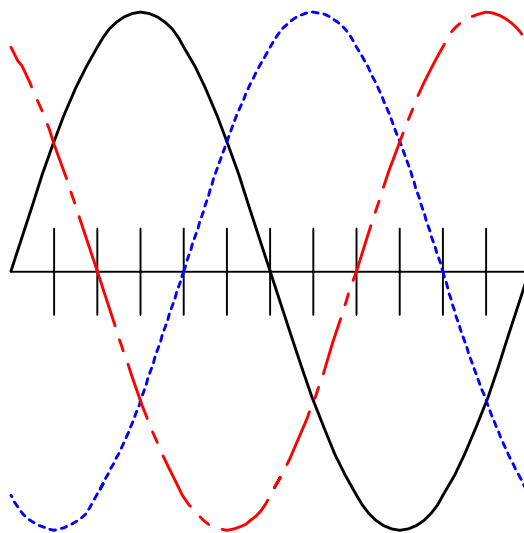


Figure 1. Brushless AC Motor

When using sinusoidal commutation to drive a three-phase brushless AC motor, currents are applied to each of the three coils at the same amplitude, phase shifted by 120 degrees, in a continuous manner. This provides for a more constant torque angle and much smoother motor movement. Figure 2 shows the outputs of a sine commutation motion controller for one commutation cycle. For the example brushless motor in Figure 1, it would take three commutation cycles to complete one rotation of the motor shaft.

The velocity of a brushless AC motor is controlled by the amplitude of the sinusoidal outputs and the phase relationship of the three outputs does not change with velocity. As the amplitude increases, the motor velocity will increase which in turn will cause the encoder frequency to induce a corresponding increase in the sinusoidal commutation output frequency.

Rotor Angle (in degrees)	0	30	60	90	120
Commutation Angle (in degrees)	0	90	180	270	360



————— Phase U
 - - - - - Phase V
 - · - · - Phase W

Verifying basic amplifier and motor operation



For connector, pinout and wiring information, consult the user manual or installation manual that corresponds to your controller model.

Before configuring closed loop sine commutation servo control, the controller can be used to move the motor in an open loop manner using the methods described here. The output voltages of phases U and V are defined by the following expression:

$$\text{Output} = T_{q_{\max}} * \sin(\theta + \Delta)$$

$T_{q_{\max}}$	=	parameter <i>n</i> of SQ (maximum torque)
θ	=	encoder commutation angle
Δ	=	parameter <i>n</i> of aLAn or aLBn (commutation offset)

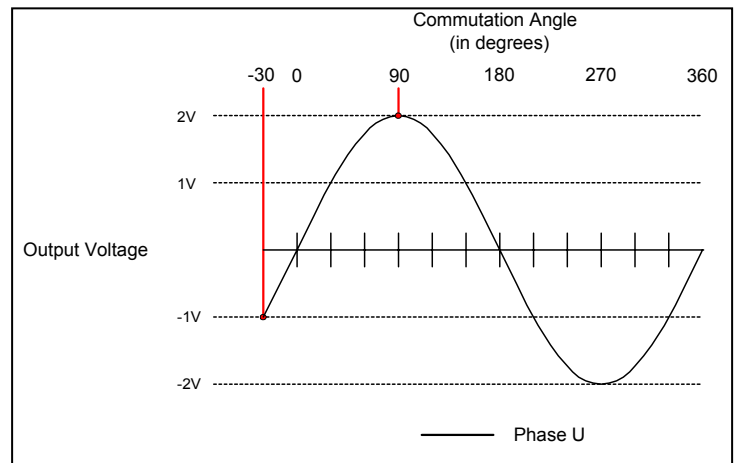
The **LAn** Load commutation phase offset A and the **LBn** Load commutation phase offset B commands are used to establish these values. The units for parameter *n* of the LA and LB commands are in degrees of commutation angle.

The LA and LB commands set the offset of the angle of the corresponding commutation phase signal – U and V, respectively – and have the effect of setting the amplitude of that signal to the product of the sine of that angle and the current torque setting. For example, by setting the offsets and torque as follows,

LA parameter <i>n</i> = 90	1LA90
LB parameter <i>n</i> = -30	1LB-30
Torque command = 2.0 volts	1SQ2.0
Enable sine commutation mode (w/ encoder commutation angle disabled)	1OM8000

$$\begin{aligned} \text{Output U} &= T_{q_{\max}} * \sin(\theta + \Delta_A) \\ \text{Output U} &= 2 * \sin(\theta + 90) \\ \text{Output U} &= 2 * 1 \\ \text{Output U} &= 2.0 \text{ volts} \end{aligned}$$

$$\begin{aligned} \text{Output V} &= T_{q_{\max}} * \sin(\theta + \Delta_B) \\ \text{Output V} &= 2 * \sin(\theta - 30) \\ \text{Output V} &= 2 * -0.5 \\ \text{Output V} &= 1.0 \text{ volts} \end{aligned}$$





As shown in the example above, the Output Mode command, **OM**, is used to control the commutation characteristics of the axis. The parameters to the OM command are:

aOM0	sine commutation mode disabled
aOM08000 ¹	sine commutation mode enabled – encoder angle disabled
aOM0C000	sine commutation mode enabled – encoder angle enabled

When in sine commutation mode, the encoder angle should be disabled for open-loop operations such as the encoder resolution test and zero-phase detent-seek. The encoder angle must be enabled for closed-loop commutation to begin.

Setting the phase U and V outputs to non-zero values using the *aLAn* and *aLBn* commands should cause the amplifier to apply current to the motor windings, causing the motor to move a *detent* position. After the motor moves, set the outputs to 0.0 volts by issuing the Set torQue command with parameter n = 0.

1SQ0

Encoder Resolution Test

The following set of MCCL macros can be used to verify proper connections and operation of the amplifier and motor, as well as confirming the encoder resolution provided by the motor manufacturer. These macros have the effect of setting the U and V commutation signals to 0° and -120°, respectively, and setting the maximum torque to 1.0 Volt. The commutation offset registers are then both advanced by 15° and the encoder position reported by the **TP** command. This process is repeated 24 times to provide a full 360° commutation cycle. The difference between the initial and final position represents the encoder resolution in counts/commutation cycle. This value will be used to load the encoder resolution register using the **LR** command, discussed below.

```
; encoder resolution test section
; MC90 initializes registers
; MC100 performs iterations
;
MD90,MC10,MC12,1SQ0,1QM,AL0,AR100,AL-
120,AR101,AL15,AR102,AL1,AR103,AL23,AR104,AL0,AR105

MD100,1MN,1LA@100,1LB@101,1SQ@103,WA.5,1TP,MJ101
MD101,AL@100,AA@102,AR100,AL@101,AA@102,AR101,1SQ0,MJ102
MD102,AL@105,AA1,AR105,IG@104,BK,NO,MJ100
```

¹ 08000 is a hexadecimal parameter. The HM (hex mode) command must first be issued in order for the controller to correctly interpret this value. DM (decimal mode) will return the controller to the default setting.

Issuing the call to macro 90 (MC90) will initialize the working registers. The call to macro 100 (MC100) will perform the iterations required to move the motor through one commutation cycle (1/3 of a rotation of a rotary motor).

Zero-phase Detent seek and closed-loop initialization

Figure 3 illustrates a linear AC motor stage which will be used as an example in the following description of commutation parameter initialization. The stage contains the 3-phase coil assembly that establishes position as a result of interaction with the fixed magnetic fields along the track.

The vertical lines along the track represent locations of zero-phase² detent positions. The distance between these detent positions is the commutation pitch which represents a 360° electrical cycle and is usually on the order of 2" in length. The encoder is typically on the opposite side of the stage and may have resolutions of 50,000 – 250,000 counts per commutation pitch.

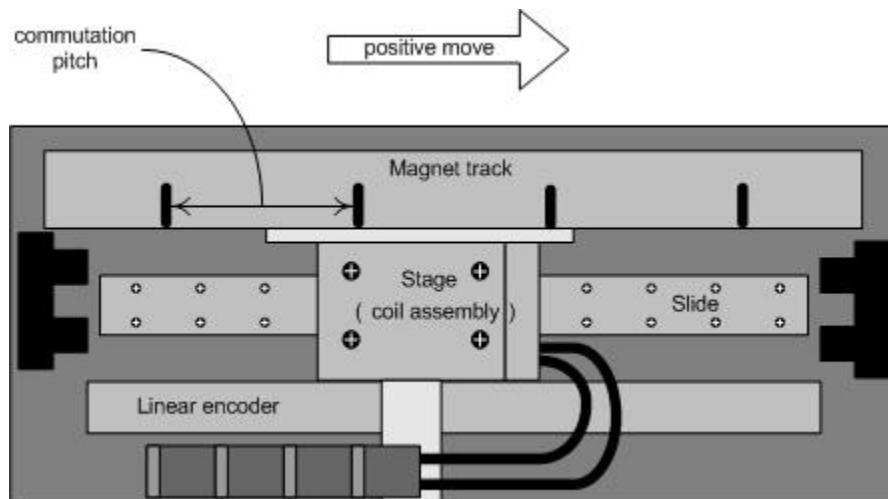


Figure 2. Linear AC Motor Stage

The stage should be placed somewhere in mid-track to insure that the ensuing motion will not cause travel to one of the stops. The following MCCL macros can be used to seek one of the zero-phase detent positions.

```

; drive initialization section
; MC99 performs seek to zero-phase detent, drive reset and commutation
; angle retardation to prepare for position mode
;
;
MD99,1QM,1SQ0,1MN,MC96,WA.5,MC94,WA.5,1RT,WA.5,MC95,WA0.5

```

² zero-phase corresponds phase U offset equal to 0° and phase V offset equal to -120°

The call to macro 99 that performs the detent-seeking utilizes calls to other macros defined below. The encoder angle must be disabled to allow the offset registers to determine the phase currents, thereby insuring an accurate positioning of the stage at the detent position. Once the stage has moved to this location, the offset registers have their phases retarded by 90° to provide for the necessary advance in commutation angle when a non-zero torque command is applied to the axis.

```

RM1

DM
MD1, HM
MD2, DM

HM
MD3, 1OM0A000           ; comm reset
MD4, 1OM0C000           ; comm_angle = enc
MD5, 1OM8000            ; comm_angle = 0

DM
MD8, 1LA-90, 1LB-210, 1LR60700 ; ph_u = -90 ph_v = -210 encoder repeat = 60700
MD9, 1LA0, 1LB-120, 1LR60700 ; ph_u = 0 ph_v = -120 encoder repeat = 60700

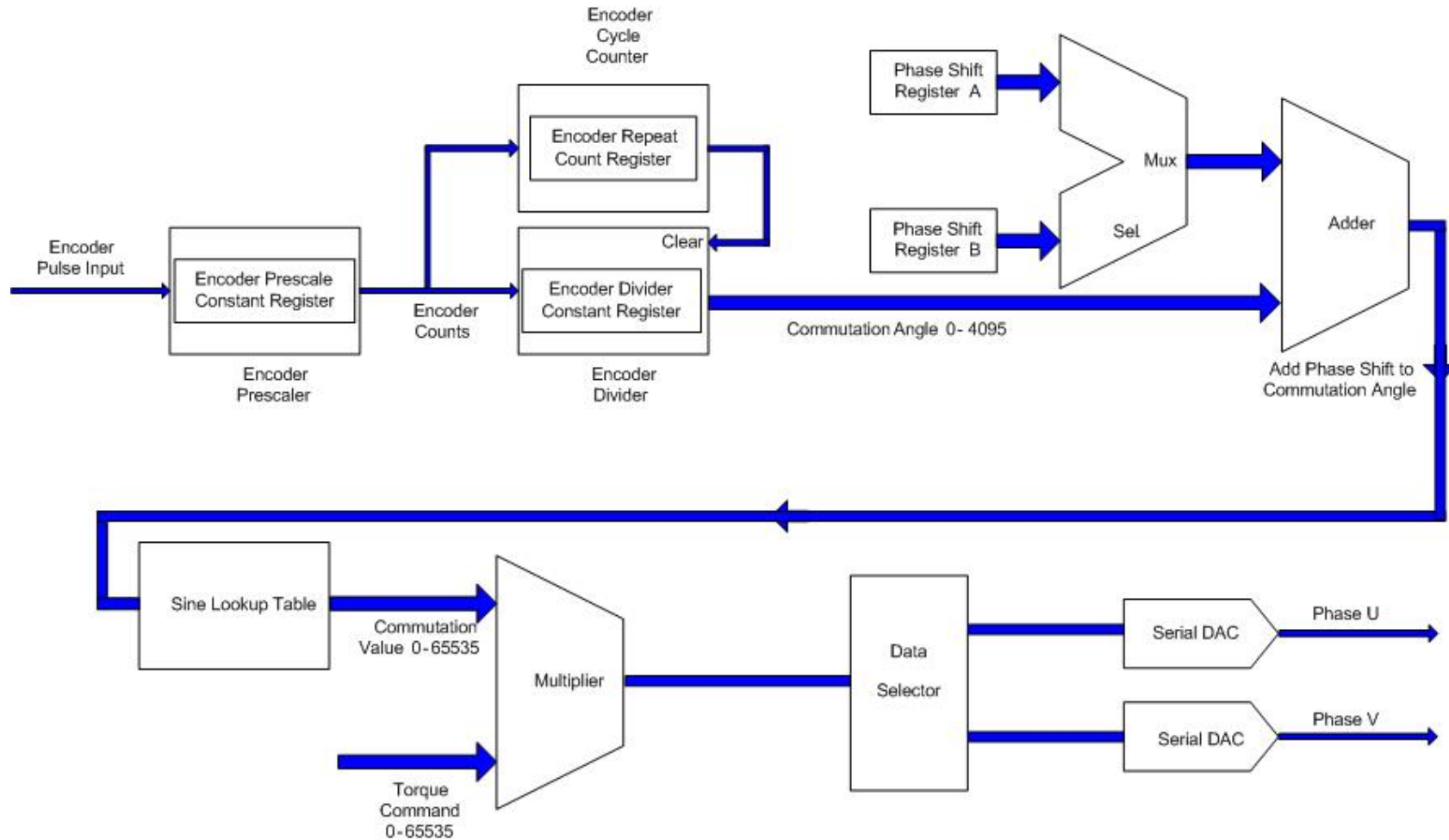
MD10, MC1, MC3, MC2      ; call reset
MD11, MC1, MC4, MC2      ; call comm_angle = enc
MD12, MC1, MC5, MC2      ; call comm_angle = 0

MD94, 1SQ1, 1WA.5, 1SQ0, 1WA1, 1TN ; apply 1V torque to seek zero-phase detent
MD95, MC10, MC11, MC8    ; initialize registers for position mode
MD96, MC10, MC12, MC9    ; initialize registers for seek zero-phase detent

```

The final step in preparing for closed-loop sinusoidal commutation and servo position moves is to load the encoder resolution count register with the correct value. The example above uses the **LR** command to initialize this value to 60700 counts/commutation cycle. The LR command performs the necessary computations to initialize the encoder prescale, divide count and repeat constant registers based upon the definition of encoder counts/commutation cycle.

Commutation Circuit Block Diagram



Commutation initialization and closed loop motion

Once the initialization steps outlined above have been performed, the controller is now ready to commutate the motor. Prior to executing any Position or Velocity mode moves the user must first tune the PID loop parameters which can be facilitated by PMC's Servo Tuning application, shown in Figure 5. Following is a sample MCCL macro that can also be used to define the servo parameters explicitly by command line.

```

; drive tuning parameter section for Trilogy 210-2 stage with Copley 7225AC amplifier
; MC500 initializes axis to following parameters:
;
;
;      Kp = 0.0126
;      Kd = 0.0570
;      Ki = 0.001
;      Accel = Decel = 10.0E6
;      Velocity = 5.0E6
;
;
MD500,1SG0.0126,1SD0.0570,1SI0.001,1SA10000000,1DS10000000,1SV5000000

```

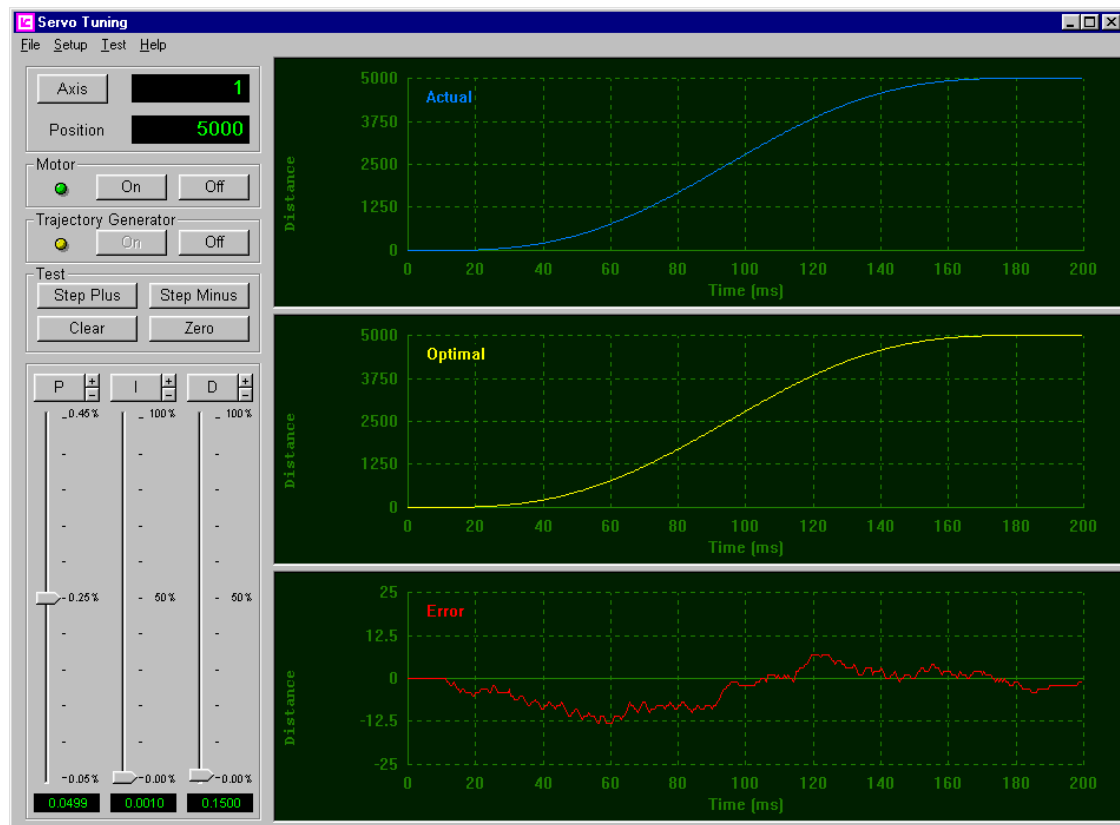


Figure 3. Servo Tuning Application