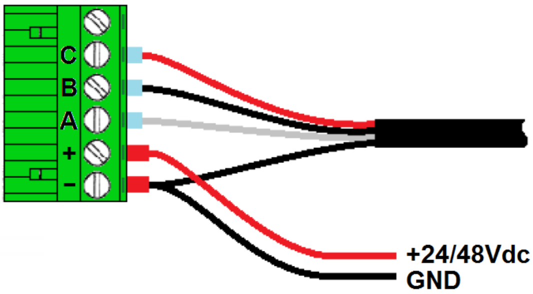
Technical Note

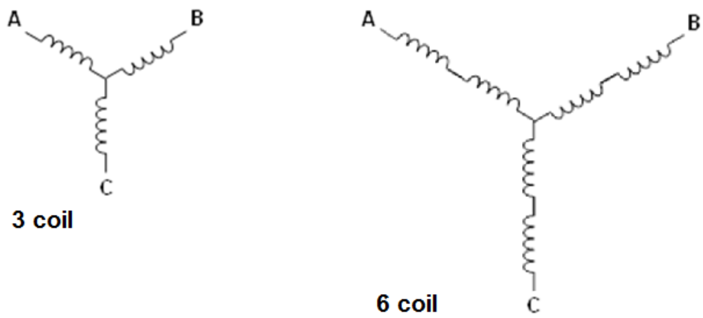
Phase Detection

Multipole concept

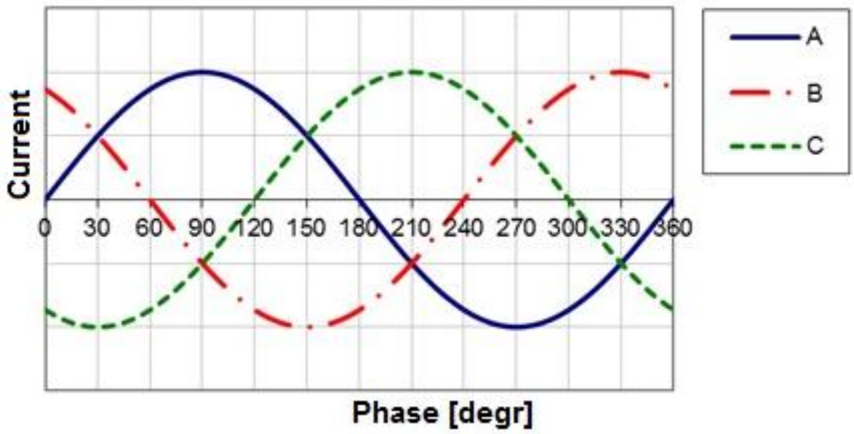
A SMAC actuator with 3 motor wires (see image→) is a multipole (also the term 3-phase actuator is used). The 3 phases are connected to the green connector at A, B and C.
The 3 phases connect to the motor coils which are in star configuration.
If the motor is a 3 coil version, each leg of the star configuration contains one coil.



If the motor is a 6 coil version, each leg of the star configuration contains two coils (in series, see ↓).



3 sinusoidal shaped voltages are applied to the 3 phases to make the actuator move. One full sine wave (360 ^{o}) is equal to one electrical revolution. The sine waves have a phase difference of 120 degrees as shown in the plot below.



Phase Detection

When an multipole actuator is started, the actual position in relation to the phase position is unknown. So, this relation needs to be determined at start-up of the controller. This can be done using:

- Forced Phasing
- Known Phase position
- Phase Vector Method

Forced Phasing

The default method to do phasing is Forced Phasing. Forced phasing applies a voltage to each phase while the other two phases are connected to ground. This way the actuator is going in course steps in the extend / retract direction. If two consecutive powered phases result in a correct encoder position difference, the phasing is completed. The phasing position is then known to the controller. The number of consecutive steps is 6 (3 in retract direction and 3 in extend direction). If the correct phasing is not found, a phasing error is reported (0xFF04).

To program a forced phasing it the following program line is required.

10	0-1	PhaseDetect	Forced,Time=,Current=,Tolerance=

The PhaseDetect needs to be programmed before the homing routine. If the parameters of the PhaseDetect are left empty, the preset values from the config file are used. The parameters of the forced phasing have the following meaning:

Time = The time phasing waits after applying a phasing step. With the 6 steps the phasing process can take 6 times this time. This time is used to wait for the vibration reduction which occurs due to the step. Generally the default value is 2500 msec.

Current = The current applied during the phasing process. Using motor rated current (1000 ‰) is in general a good value. If you apply a higher value, the I2T can be triggered. Higher currents give higher frequency vibrations. A too low current value gives inaccurate phase detection.

Tolerance = The Tolerance of the position difference between two steps. If the tolerance is exceeded, the phasing is retried up to 6 steps. Generally the default value is 10%.

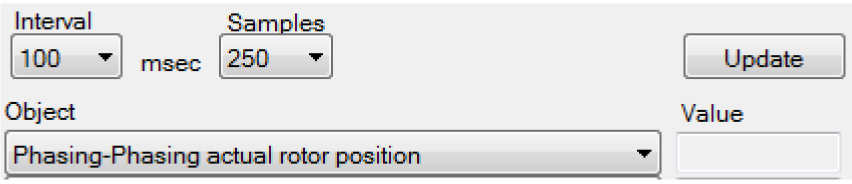
The parameters can be adjusted in cases of a return spring, a large added moving mass or other situations that make the phasing more difficult. In these situations you can increase the Current to 1500 or 2000 ‰. If that is insufficient, the Tolerance can be increased to 20%.

Known Phase Position

If the actuator is in a known position at start-up of the controller, a PhaseDetect is not necessary. In such a case it is possible to use the Initial position always known method. This method only requires providing the initial rotor position.

11	0-1	PhaseDetect	Initial_position_allways_known,Rotorpos=54500

This rotor position is the phase angle of the actuator at its actual location. This value is expressed in counts. The value 0 corresponds with the phase position 0 degree. The value 65535 corresponds with the phase position 360 degree. The actual phase position can be found by doing a regular PhaseDetect, then switching the motor off (with motor off button, not by disconnecting the power to the controller) so that the actuator can go to the initial position which it will have at power up of the controller. When it has reached this position, check the Phasing actual rotor position as shown below (press the update button to refresh the value)



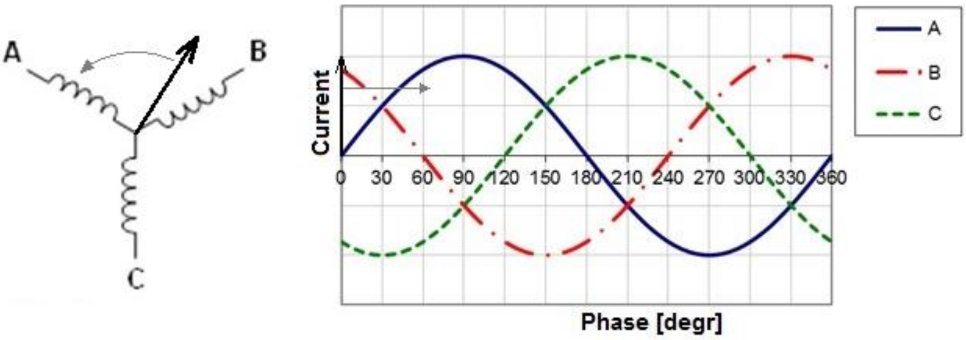
The resulting Phasing actual rotor position is the value that you should use as the Rotorpos in the Initial_position_always_known method.

Note: If the actuator is not in the correct position during start up, no error is reported. The actuator will have an incorrect phasing that in best case results in lower forces or high force ripple. Worst case it will not even move in correct directions.

Phase Vector Method

The most complex method is the phase vector method. It is not a universal method for all applications since the success of the method depends for a great deal on the application details. This method is created to have a phase detect without vibrating motion and to allow the axis to be on any position at start up. The intention is to move only in retract direction and as soon as the end-stop is reached, set the phasing using the known phase position method. There are various ways to solve this case. In the description on the next page one possibility is shown which is suitable for an actuator without a spring, without payload and in horizontal orientation.

The basis of this method is that the phase vector is used in an open loop (Motion Mode nr = -1). This mode of operation will apply a defined voltage on the coils resulting in a current through the coils. (current = length of the vector in the images below). This mode will generate a rotation of this vector in the electrical star configuration of the coils. This way the actuator will move in a retract motion with a fixed velocity of one electrical revolution in 6 seconds (60 degr/sec).

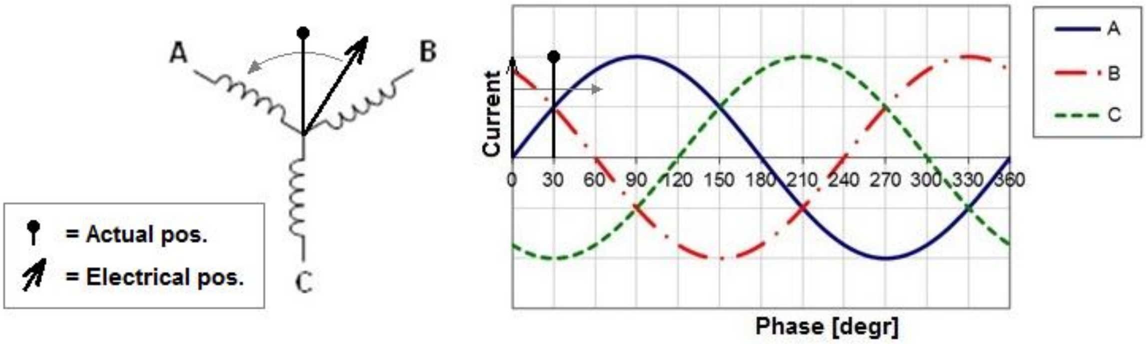


The program to create this motion is shown in the text block below. It is the macro 0 part of the program as it is saved as .LCC or .SCC file. This example program can be found in the example programs (phasing vector method in Demo-examples).

```
MacroNumber     0 // Macro 0 starts automatically at power up
PhaseDetect     Initial_position_allways_known,Rotorpos=  // disable the autophase sequence
SetVariable     Var=Modes_of_operation(0x006060),Constant,Const=-1 // set the open loop mode (motion without encoder feedback)
SetVariable     Var=Target_duty(0x002F01),Constant,Const=700 // set voltage on coils (Full Vcc = 32767)
SetVariable     Var=Controlword(0x006040),Constant,Const=6
SetVariable     Var=Controlword(0x006040),Constant,Const=7
SetVariable     Var=Controlword(0x006040),Constant,Const=15
Macro           Jump,MacroNumber=1
```

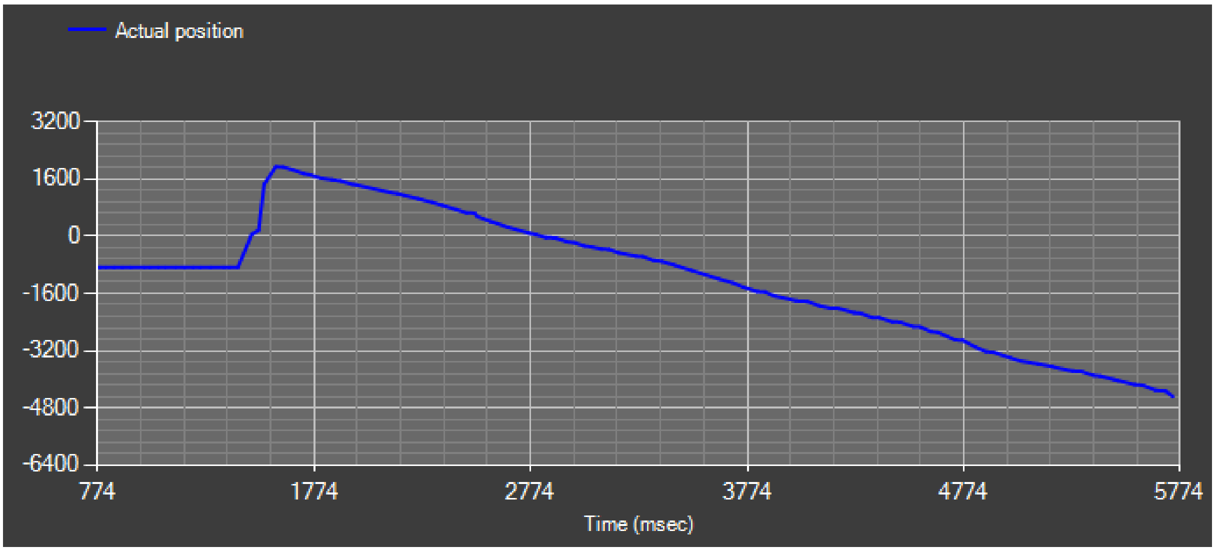
The sequence of control word 6, 7 and 15 is used to change the status from any state to start the open loop motion. In this example the voltage (target duty) is set to 700/32767 (≈ 0.02) times the supply voltage of the controller. This value should be just above the voltage required to overcome the friction. The jump in the last line is to make the program continue to the next step.

When this open loop is started, the actual position of the actuator will not be exactly the same as the electrical position (see image below).



In cases the electrical position is a little ahead of the actual position; the actual position will move towards the electrical vector and follow it from that point on.

In case the actual position is a little ahead of the electrical position, the actual position will move in the opposite direction towards the electrical position and will follow it from that point on (see position plot below).



This will lead to a movement in the extend direction that can be as big as 45% of the full electrical revolution distance, for an LCA25 that could be a distance of up to 24 mm.

Since we want to reduce the extend movement, we need to switch off the motor as soon as we detect a forward movement. The movement check is done in macro 1 as shown in the text block below.

```
MacroNumber    1 // Check motion direction
If        Actual_velocity,Higher,Threshold=0,Then_Macro_Call=2,Else_Macro_Continue // if moving forward go to macro 2
If        Actual_velocity,Lower,Threshold=-200,Then_Macro_Jump=3,Else_Macro_Continue // If retracting go to macro 3
Macro       Repeat,RepeatCount=0
```

The first check is to see if the velocity is forward (>0 counts/sec) and if so it will call macro 2. Then the check is done if the velocity is backward (<-200 counts/sec). If so it will jump to macro 3 and continue the phasing process. If both are not the case it will keep checking the velocity by repeating macro 1.

In Macro 2 the actuator is de-activated for 1500 msec (= 1/4th revolution of the phasing vector). Then the motor is activated again using the control word sequence 6, 7, 15. This way the phase position has moved for 1/4th of a revolution without powering the unit.

```
MacroNumber     2 // motor off due to forward movement
Motor           Off
Wait            Time,Timeout=1500
SetVariable     Var=Controlword(0x006040),Constant,Const=6
SetVariable     Var=Controlword(0x006040),Constant,Const=7
SetVariable     Var=Controlword(0x006040),Constant,Const=15
Macro           Return
```

The Return command will take care that the program continues where the call came from, in the middle of macro 1.

Once the actuator has reached the threshold velocity for detecting retract motion, it jumps to macro 3. In Macro 3 the target duty (applied voltage) is increased to make sure it has sufficient force to move the full stroke to the retract position. Then a check is done if it reaches the full retract end stop. A wait time is added to ensure that vibration is not playing any role in the detection.

```
MacroNumber     3 // check if velocity = 0, retract position reached
SetVariable     Var=Target_duty(0x002F01),Constant,Const=5000 // set voltage on coils (Full Vcc = 32767)
Wait            Time,Timeout=500
If              Actual_velocity,Higher,Threshold=-5,Then_Macro_Jump=4,Else_Macro_Continue // If retract pos reached, go to macro 4
Macro           Repeat,RepeatCount=0
```

Then a check is done if the velocity is about 0 (>-5 counts/sec). If so, it will jump to macro 4, if not it will repeat this loop.

In Macro 4 the actuator is at the full retract end stop position. At this position the phasing position is known. The Phasing initial rotor position (object 0x2100 sub 0x05) needs to contain the correct phasing position for the full retract location. This object is a non-volatile variable that can be part of the config file.

```
MacroNumber     4 //  set phase position at retract pos, do homing
PhaseDetect     Initial_position_allways_known,Rotorpos= // set phasing position at full retract.
Homing          Indexpulse,Positive,Acc=,Vindex=,Timeout=,Offset=
PositionMove    Absolute,Target=0,Vel=,Acc=,Change_immediate
Wait            Trajectory_generator_ready
```

After phasing with initial position always known the homing against retract end-stop is executed with standard config file parameters. After finding the index a position move is added to go to the exact zero position. The result of macro 4 is that the phasing and homing has been done.

In order to find the phasing position at full retract the macro 5 is added. Macro 5 will do a standard phasing and makes a softland against the endstop. At the end stop position the actual phase is read and copied to Phasing initial rotor position. Macro 5 is only required once, at setting up the actuator.

```
MacroNumber     5 // Initial determination of the retract phase position
PhaseDetect     Forced,Time=,Current=,Tolerance= // default phase detect
Softland        Negative,Vel=10000,Acc=,Force=1000,Error=1000,Apply_force // Find Full retract position
Wait            Time,Timeout=100
SetVariable     Var=Phasing-Phasing_initial_rotor_position(0x052100),Variable,Var1=Phasing-Phasing_actual_rotor_position(0x062100)
GetVariable     Var= Phasing-Phasing_initial_rotor_position(0x052100)// display retract phase pos.
Motor           Off
```

In order to write all registers to NVRAM you need to push the button “Save in non volatile” in the Tuning tab of LCC control center (or SMAC control center).

What if the Program of the phase vector does not work?

The program holds various parameters which can be adjusted. These parameters and can be adjusted to the needs of the application.

Target duty (macro 0; 3rd function): Numerical range is 0 to 32767. Realistic range is 200 to 2000. The value represents the voltage on the actuator, 32767 is equal to the supply voltage to the controller. The value needs to be just above the friction level so that extend moves will not be too powerful. This Target duty is only used to detect motion direction. A value of approx. 2% of full voltage is generally sufficient for most actuators. In that case the Target duty is 700. After that the Target duty in Macro 3 is used.

Target duty (macro 3; 1st function): Numerical range is 0 to 32767. Realistic range is 1000 to 10000. The value represents the voltage on the actuator, 32767 is equal to the supply voltage to the controller. It is used to move to full retract in a robust way. Too low values causes the full retract position to be found at too early locations. Too high values will possibly damage the actuator coils. Calculate the safe voltage carefully since the I2T is not operational in this motion mode. The calculation is started by checking the motor rated current of the actuator (see tuning tab). Then you need to calculate the motor rated voltage by multiplying the motor rated current with the motor coil resistance (measure or look up in specs). Then determine the target duty that represents this voltage using the supply voltage to the controller.

Example:
Motor rated current = 500 mA
Motor Coil resistance = 18 Ohm
Controller Supply Voltage = 24 Volt

Motor rated Voltage = I x R = 0.5 x 18 = **9 V**
Max Target duty **=** 32767 * Motor rated voltage/Control supply Voltage **=** 32767 * 9/24 = **12287**

The value of the target duty has effect on the strength of the motion in macro 0 up to macro 3. This means that both the strength of the movements forward and backward during the first part of the phasing sequence are determined by this value.

Actual velocity lower threshold (macro 1; 2nd function): Realistic Range is -100 to -200000. The value represents the velocity at which the retract motion is detected. It depends on the encoder resolution. If, for example, the encoder resolution is 5 micron and the full electrical revolution (pole period) is 52.5 mm (like the LCA25), you can calculate the average velocity for the retract move. With the 6 seconds for one electrical revolution the velocity is 52.5/6 = 8.75 mm/sec which is 1750 counts/sec. The value entered as lower threshold for the velocity should be significantly lower than this value (about 20% of the nominal value works fine, being 350 counts/sec in the example) in order to create a robust detection, even in the case that the actuator is at full retract position at start up.

Time out (macro 2; 2nd function): Realistic Range is 100 to 2500. The value represents the wait time after incorrect direction detection. If going in the extend direction, the motor is switched off for a period to activate the motor again. You make this value shorter if you have an actuator that has difficulties to start moving. Shortening the time has also the consequence that the extend direction is briefly activated more often. In general a good value lies between 1000 and 2000 msec.