

# LMDX

## Command and Parameter Manual

2016.11.10.

This manual is applicable for LMDX driver with firmware version 2016.10.25.

## CONTENTS

0. Terminologies .....	3
1. Common syntax.....	5
2. Commands explanation.....	5
2.1 Motion related commands 'PA', 'PR', 'AA', 'DL'.....	6
2.2 'FA', 'KV', 'CC', 'BF' .....	8
2.3 Coordinate display 'DD', 'N', 'DR', 'FF', 'RX', 'RY'.....	10
2.4 Errors 'DE', 'ER', 'FR' .....	11
2.5 ADC calibration commands 'CL', 'CLS', 'NN'.....	12
2.6 Zero electrical angle commands 'NP', 'NPS'.....	13
2.7 Alignment commands 'LNS', 'LNP', 'LN', 'SL' .....	14
2.8 Motor parameters 'DM', 'PX', 'MX', 'RF', 'AM', 'SF', 'LT', 'NB', 'MM', 'AS'.....	14
2.9 Homing commands 'GSS', 'GP', 'GS' .....	19
2.10 Compensation of position sensor 'CD', 'SF', 'ST'.....	20
2.11 IO related commands 'WN', 'PN', 'DN' .....	22
2.12 Power on setting 'RP' .....	23
2.13 Pulse mode command 'SD' .....	24
2.14 Motion programming commands '@', 'DP', '!', 'PS' .....	24
2.15 Display of position sensor signals 'G', 'GC'.....	25
2.16 PID gains 'KX', 'KY', 'KF' .....	25
2.17 Flash ROM read/write 'RE', 'WE' .....	26
2.18 Motion modes 'FX'.....	26
2.19 Driver's version 'VER' .....	26
2.20 Reset driver 'RES' .....	26
2.21 DSP time parameters 'DS', 'TI', 'SP' .....	27
2.22 Error compensation.....	27
2.23 In-Position 'INP' .....	28
2.24 Data capturing function 'NS', 'CB', 'O' .....	29
2.25 Software Limits 'LMT'.....	30
Figure 1 Definition of axes in LMSP series motors.....	5
Figure 2 An example of circular arc movement.....	7
Figure 3 Multi-point continuous motion.....	10
Figure 4 Diagram of RF parameter .....	16
Figure 5 Phase compensation table of field weakening .....	17
Figure 6 Modified current according to velocity .....	17
Figure 7 Current limit at alarm .....	18

## 0. Terminologies

### Stand-alone mode

With help of high speed DSP, HIWIN's LMDX series driver is capable of path planning. Without motion control card, it is possible to issue motion commands through RS-232.

### Pulses mode

Motion controller sends pulses to driver. Pulses correspond to position commands. When driver receives a pulse, motor would move a specified distance accordingly. Motion controller is responsible for path generation. In acceleration phase, frequency of sending pulses becomes higher. In constant speed phase, frequency remains the same.

### Control points

When executing multi-point continuous motion in stand-alone mode, the starting point and the terminal point of every motion line segment are called control points.

### Smooth movement

When executing multi-point continuous motion in stand-alone mode, motor can stop on every control point and then go on to the next motion line segment. Therefore the path can have turning points(called not smooth), and LMDX driver provides smooth motion in which control points are not passed through. It will smooth the original turning point of multi-point continuous motion.

### Mechanical pitch

In Hiwin's LMSP series motors, there are teeth or grids on the surface of stator. Mechanical pitch is the distance between two consecutive teeth or grids. It is 0.64mm for LMSP.

### Current Harmonics

In motor control, different current harmonic ratios are applied to change the motor commutation current. When an ideal motor is supplied with sinusoidal commutation current, it moves proportionally. However, actual coils of motors might include some manufacturing faults and motor deviates from ideality. To improve this, it might be useful to change the harmonic ratio of commutation current. Hiwin LMDX driver provides possibility of adjusting up to the 7<sup>th</sup> harmonic ratio.

### Field weakening

At high speed, motor does not provide enough force or torque due to back EMF of motor. Therefore, high speed is not easy to achieve. In order to overcome this

situation, field weakening technique is used to increase the achievable max speed. (Note: Efficiency and maximum thrust force will be decreased.)

## Alignment

In LMSP series motors, there are teeth or grids on the surface of stator (platen). There are also teeth structure on force modules and sensors in forcer, therefore motor must align with grids before servo control (Servo on). This is an important step before servo control.

## Grid zero

In LMSP series motors, there are teeth or grids on the surface of stator (platen), and there are also teeth structure on force modules in forcer. After force modules are supplied with current of zero electrical angle, forcer's teeth and stator's teeth attract each other. And the corresponding balance point is called grid zero.

## Sub-period

The position sensor is sensing tooth by tooth. The position measured inside a tooth is called sub-period.

## Motion buffer

In stand-alone mode, all unfinished motion commands are stored in motion buffer. Motion buffer may contain many motion commands, commands will be executed one after another.

## Position counters

The memories for actual positions of X- and Y-axis inside LMDX driver are called position counters.

## X1- and X2-axis

In LMSP series motors, there are two sensors for X-axis. One is X1-axis sensor and the other is X2-axis sensor.

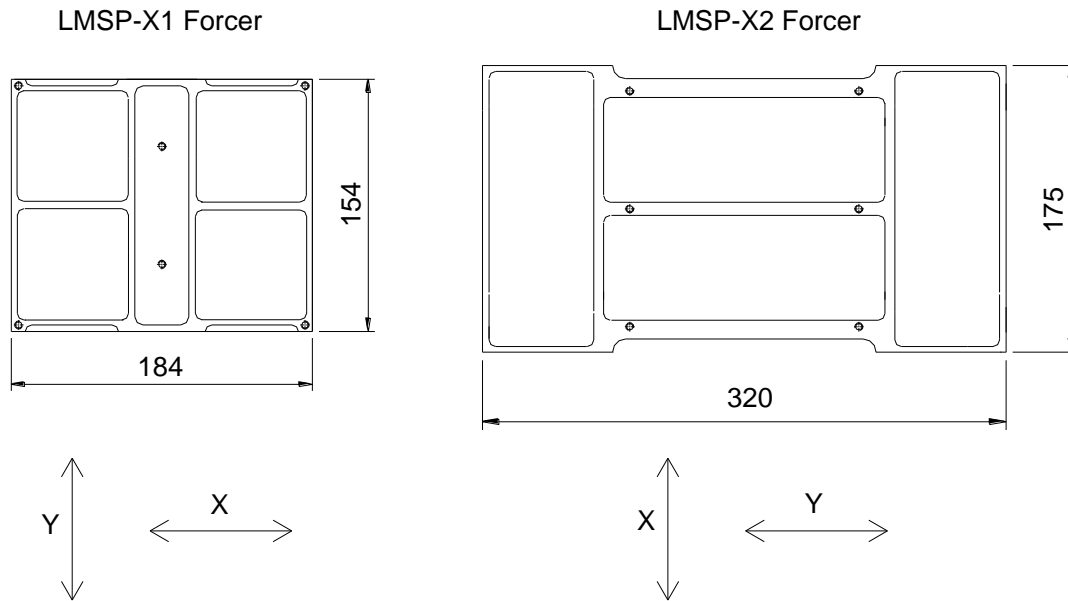


Figure 1 Definition of axes in LMSP series motors

## 1. Common syntax

It is possible to issue commands through RS-232, the settings of the corresponding communication port are 8 data bits, 2 stop bits, odd parity, and baud rate from 2400 up to 115200 baud. (See command 'SP')

All commands are case insensitive.

When working normally, driver is ready to receive commands whenever the prompt '>' is displayed on the screen.

Parameters of a command are separated by blanks.

In case command accepts multiple parameters, if only parameters at the beginning are entered, not given parameters do not change.

If not specifically described in each command explanation, parameters are floating decimal numbers.

For example: '234.56', '15', '65000.1', '0'.

All commands should be ended with enter key or semicolon ';'. Notice that

1. If the command is accepted normally, the prompt '>' is displayed.
2. If the syntax of command is incorrect or the execute time is not suitable, the symbol '?' follows.
3. If the motion buffer becomes full, the symbol '!' appears. After some movements are executed and part of the motion buffer is cleared, it is again possible to input motion command.

In this manual, .1, .2, .3 are used to show the first, second and third value of the parameters. For example, FA.1 means the first value of FA-parameters, and so on.

## 2. Commands explanation

## 2.1 Motion related commands 'PA', 'PR', 'AA', 'DL'

There are four move commands. The move commands can not be executed in the following conditions:

- The driver is servo off.
- When the motor is homing.
- Tracking error is too big and the driver is in alarm status.
- In pulse mode.

### 2.1.1 Absolute positioning 'PA'

Usage:

' PA xxx, yyy ', where

xxx – final absolute position of X-axis, in  $\mu\text{m}$ , signed.

yyy – final absolute position of Y-axis, in  $\mu\text{m}$ , signed.

Example:

PA 12345, 23456;

After performance of this command, motor will be moved to the point with absolute coordinate of (12345, 23456).

### 2.1.2. Relative positioning 'PR'

Usage:

' PR xxx, yyy ', where

xxx – relative movement on X-axis, in  $\mu\text{m}$ , signed.

yyy – relative movement on Y-axis, in  $\mu\text{m}$ , signed.

Example:

(PA 50000, 50000;)

PR 10000, -22000;

Assumed that PA 50000,50000 has been executed, after executing PR 10000, -22000, it moves 10000  $\mu\text{m}$  on X-axis and -22000  $\mu\text{m}$  on Y-axis. Namely, the final absolute position coordinate is (60000,28000).

### 2.1.3. Circular / Arc movement in absolute coordinate 'AA'

Usage:

' AA xxx\_center, yyy\_center, angle; ', where

xxx\_center – absolute X-coordinate of the center of arc, in  $\mu\text{m}$ , signed.

yyy\_center – absolute Y-coordinate of the center of arc, in  $\mu\text{m}$ , signed.

Angle – angle of arc, in degrees(angle > 0 indicates counter-clockwise motor)

Example:

(PA 50000, 50000;)

AA 50000, 55000, 90;

Assumed that PA 50000,55000 has been executed, after issuing AA 50000,55000,90, motor moves along an arc with center of (50000,55000). The arc motion is shown in Fig. 2. This command is only for stand-alone mode.

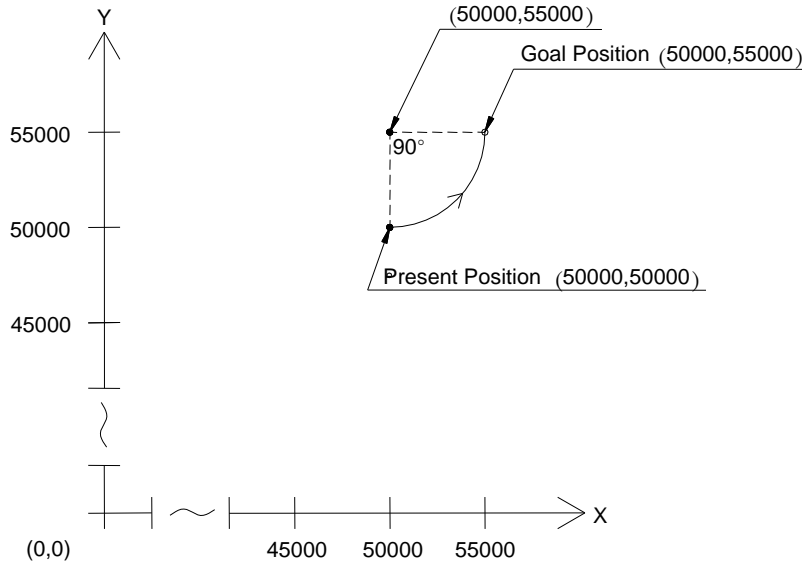


Figure 2 An example of circular arc movement

## 2.1.4. Pause motion for a specified time 'DL'

Usage:

'DL ttt' where

ttt - delay time(starting from finishing of the previous movement), in ms, only positive value.

Example:

PA 12345, 23456

DL 1500

PA 0 0

After movement to the point of (12345, 23456), it remains still for 1500 ms(1.5 sec.), then it proceeds to move to coordinate (0,0). This command is only for stand-alone mode.

## 2.1.5. Notice

After motion commands are issued, they are stored in motion buffer and executed consequentially. If the motion buffer is full, no further commands could be issued and it shows '!'.  
The state of motion buffer can be read by command 'BF'.

If there was a necessity to stop a drive immediately and to clear unexecuted motion commands stored in the motion buffer, please operate according to the following

sequences.

1. By a command 'KV 0' to lower trajectory speed down to 0, make sure that motor is motionless.
2. By a command 'BF 0' to clear all the rest commands inside the motion buffer.
3. By a command 'KV 1.0' to set nominal trajectory speed.

## 2.2 'FA', 'KV', 'CC', 'BF'

### 2.2.1. Velocity and acceleration command 'FA'

In stand-alone mode, motor moves with the velocity and acceleration specified by the command 'FA'. After setting parameters with 'FA', the following movements use the value until using 'FA' to change again. For every motion command, velocity and acceleration are both stored in the motion buffer.

Usage:

'FA p1, p2', where

p1 - velocity, in mm/sec

p2 - acceleration, in m/sec<sup>2</sup>

Both parameters should be positive.

Example:

FA 10, 1.5

PR 20000, -10000

Both the above commands are brought into motion buffer. And speed of the following movements remains the same until next usage of 'FA' to change.

### 2.2.2. Velocity scale command 'KV'

When a motion command is taken from motion buffer, its velocity is multiplied by the scale and then motion is executed.

Usage:

'KV p1' where

p1 - velocity scale,  $0.0 < p1 \leq 4.0$

Example:

After reception of a command 'KV 0.5', the velocity becomes the original value multiplied by 0.5. The acceleration for changing speed is same as that of 'FA' command. And velocity scale of the following movements remains the same until next usage of 'KV' to change.

### 2.2.3. Smooth motion command 'CC'

The capability of command 'CC' is to smooth motion without passing through control points when executing multi-point continuous motion.



## Usage:

'CC p1 p2' where

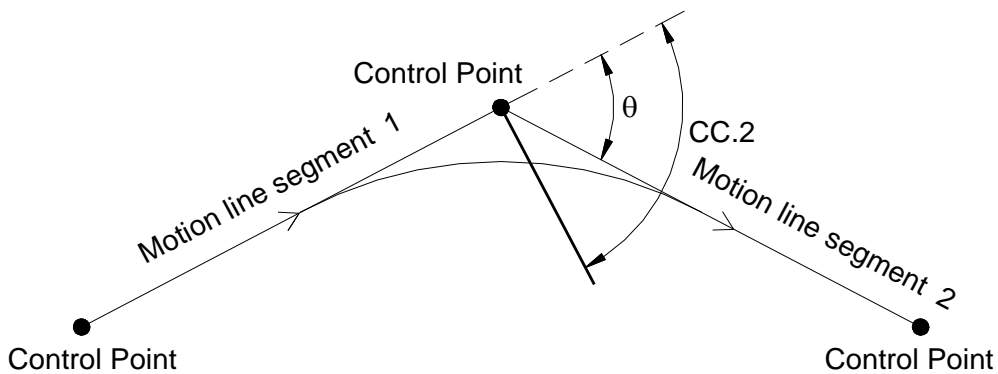
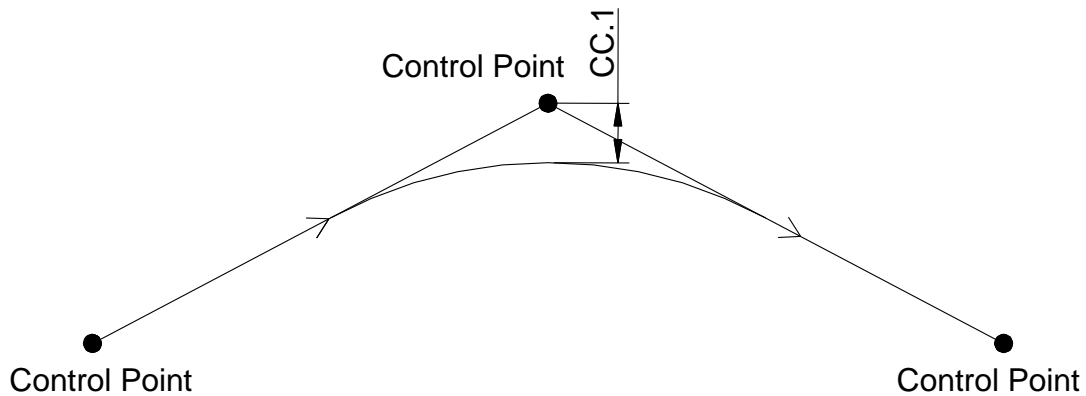
p1 - ( $\mu\text{m}$ ) maximum deviation of distance from control point in the process of smooth motion. According to setting, the movement may not run through the control point. The parameter must be positive.

When  $p1=0$ , the movement will certainly run through the control point (so called not smooth motion).

p2 - maximum angle of direction change, which allows smooth motion. If the angle of direction change exceeds p2, the movement will certainly run through control point (so called not smooth motion).

Both parameters should be positive.

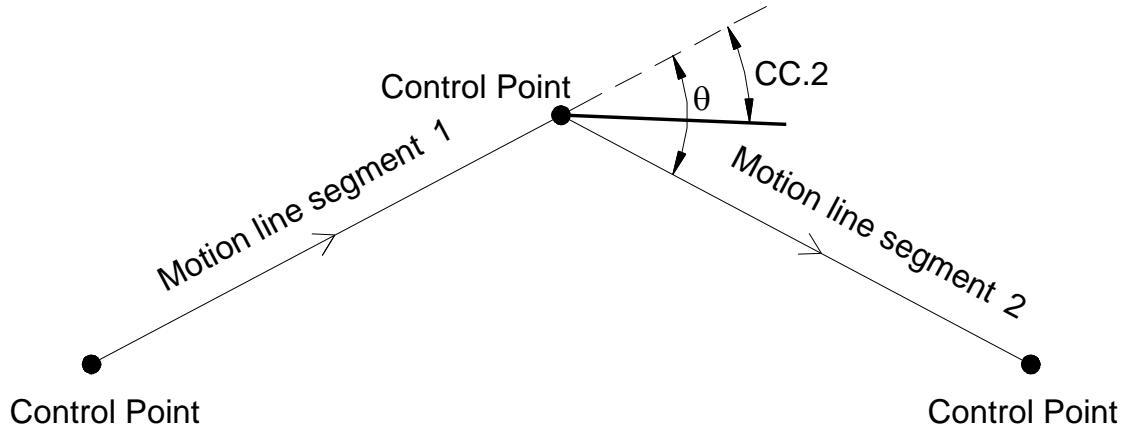
When smooth movement applied, if necessary, it will slow down near the coming control point according to p1.



$\theta$ : angle of direction change

CC.2: maximum angle of direction change

If  $CC.2 > \theta$ , there is smooth motion



If  $CC.2 > \theta$ , the movement will certainly run through control point and there is not smooth motion.

Figure 3 Multi-point continuous motion

**2.2.4. Motion buffer size command 'BF'**

Usage:

'BF 0' motor stops immediately (no deceleration), and unexecuted motion commands are cleared.

'BF' it shows the internal motion buffer memory being occupied, displays a decimal number (0 to 255)

- 0 – it means all motion commands have been executed, and the motor stops.
- 255 – it means all motion buffer memory have been taken up, so it does not accept any new motion command.

**2.3 Coordinate display 'DD', 'N', 'DR', 'FF', 'RX', 'RY'**

**2.3.1. Display of command position 'DD'**

Usage:

After executing the command 'DD', "*p1 p2*" will be displayed

- p1 – X-axis command position, in  $\mu\text{m}$
- p2 – Y-axis command position, in  $\mu\text{m}$

**2.3.2. Binary transfer of X, Y command position 'N'**

Usage:

'N' it transfers 10 bytes of binary data,

"B1 B2 B3 B4 B5 B6 B7 B8 B9 B10 ", followed by (CR, LF) (carriage return, line feed)

B1 B2 B3 B4 – command position of X-axis, in  $\mu\text{m}$ , in format int32 (32-bit integer)

- B1 – lowest byte
- B4 – highest byte

B5 B6 B7 B8 – command position of Y-axis, in  $\mu\text{m}$ , in format int32

B5 – lowest byte

B8 – highest byte

B9 B10 – two bytes of CRC, calculated by summing bytes of B1~B8 and only lower 2 bytes are considered as CRC

B9 – low byte

B10 – high byte

### 2.3.3. Display real position (feedback position)'DR'

Usage:

After executing the command 'DR', "*p1 p2 p3*" will be displayed, where

p1 – feedback position X1, in  $\mu\text{m}$

p2 – feedback position X2, in  $\mu\text{m}$

p3 – feedback position Y, in  $\mu\text{m}$

### 2.3.4. Display sub-period 'FF'

Usage:

After executing the command 'FF', "*p1 p2 p3*" will be displayed, where

p1 – sub-period X1, in  $\mu\text{m}$

p2 – sub-period X2, in  $\mu\text{m}$

p3 – sub-period Y, in  $\mu\text{m}$

### 2.3.5. Reset position counters 'RX', 'RY'

Usage:

'RX' – after execution, motor moves to the nearest grid zero. Then value of command position counter of X becomes 0.

'RY' – after execution, motor moves to the nearest grid zero. Then value of command position counter of Y becomes 0.

## 2.4 Errors 'DE', 'ER', 'FR'

### 2.4.1 Display of alarms in each axis 'DE'

Usage:

After executing the command 'DE', "*p1 p2 p3*" will be displayed, where

p1 – X-axis alarm

p2 – Y-axis alarm

p3 – rotational alarm

Alarm code:

0x800 – tracking error too big. Tracking error window is defined by 'ER' command.

0x400 - position command exceeds software limits (please refer to section 2.27, 'LMT' command)

When motor is in servo on state (FX1, alignment OK), if any of the above alarm happened, ALM (O3 signal) of IO would be ON. There are the following ways to clear the alarm: 'DE 0 0 0', 'BF 0' or 'GS' (also via HOME signal from IO), other commands such as 'FX1', 'FX2' and 'RES' also clear alarms.

## 2.4.2 Tracking error window 'ER'

Usage:

'ER p1 p2 p3', where

- p1 – permissible error for closed loop in X,Y direction, in  $\mu\text{m}$
- p2 – permissible rotational error, in  $\mu\text{m}$
- p3 – fatal rotational error, in  $\mu\text{m}$ . If this window is exceeded during closed loop control, it becomes fatal error.

## 2.4.3 Flags 'FR'

Usage:

'FR p1 p2 p3', where

- p1 – X-axis homing flag
  - 0=failure, (In this case, compensation table is not effective)
  - 1=success
- p2 – Y-axis homing flag
  - 0=failure, (In this case, compensation table is not effective)
  - 1=success
- p3 – determines whether automatic reset after fatal error is performed
  - 0=not reset automatically
  - 1= reset automatically

## 2.5 ADC calibration commands 'CL', 'CLS', 'NN'

### 2.5.1 Calibration of position sensor signals 'CL'

Usage:

'CL'

After executing the command, amplitude and offset of signals from position sensors are collected. Signals are collected for calibration while motor moves 2 mechanical pitches back and forth along axis X and Y.

### 2.5.2 Parameters for calibration 'CLS'

Usage:

'CLS p1 p2 p3 p4 p5 p6 p7 p8', where

- p1 – 'CL' calibration velocity, in mm/s
- p2 – determines whether 'CL' is to be performed during motion. 0 for no; 1 for yes; the following p3, p4 and p5 are related.
- p3 – filter factor, range 0..1.0, the smaller value, the stronger filtering.
- p4 – time interval for calibration, in ms
- p5 – minimum distance to travel for every calibration, in  $\mu\text{m}$

If the travelling distance on an axis is less than the minimum distance (set by p5) during p4 ms time, the calibration on that axis is not performed.

- p6 – maximum amplitude ratio. During each calibration, the measured signals with amplitude greater than  $2048/p6$  will be recorded. (please refer to the second parameter of command 'NN0'~'NN5')
- p7 – time delay before reading sensors for calibration, in  $\mu\text{s}$
- p8 – delay time for current rise, in  $\mu\text{s}$

### 2.5.3 Display result of calibration 'NN0'~'NN5'

Only after successful execution of 'CL', the displayed data are effective.

Usage:

'NNn p1 p2', where

n –

- 0,1 – A, B phase of X1-axis
- 2,3 – A, B phase of X2-axis
- 4,5 – A, B phase of Y-axis

- p1 – offset of signal from sensor (range -2048~2047)
- p2 – amplitude ratio of signal from sensor ( $2048/\text{unilateral amplitude}$ ) (range 1.0~10.0), the value is between  $2.0 \pm 0.3$  normally. Bigger value means weaker signal.

## 2.6 Zero electrical angle commands 'NP', 'NPS'

### 2.6.1 Searching data for zero electrical angle 'NP'

Usage :

Using this command, motor moves 1.5 teeth back and forth along X and Y axis and collects sub-period data of zero electrical angle.

### 2.6.2 Data of zero electrical angle 'NPS'

Usage:

'NPS p1 p2 p3'

- p1 – sub-period for zero electrical angle of X1-axis

p2 – sub-period for zero electrical angle of X2-axis

p3 – sub-period for zero electrical angle of Y-axis

The parameters will be used during execution of command 'LN' for alignment with grids.

## 2.7 Alignment commands 'LNS', 'LNP', 'LN', 'SL'

### 2.7.1 Storing parameters for alignment 'LNS'

Usage:

'LNS'

It stores all parameters of NN0~NN5 and NPS to flash ROM. The parameters are for alignment (see command 'LN').

### 2.7.2 Setting of alignment parameters 'LNP'

Usage:

'LNP p1 p2 p3 p4'

p1 – Permissible error of alignment, in  $\mu\text{m}$ . When the difference between feedback from position sensors and parameters of NPS are less than p1, alignment is successful.

p2 – maximum number of trying to rotate and align; namely, if alignment trial exceeds p2 times, motor will give up alignment.

p3 – numbers of counter-clockwise(or clockwise) rotation at alignment each time.

p4 – the current scale of Y-axis motor module at alignment (1 is maximum(3A),0 is minimal (no current))

### 2.7.3. Executing alignment 'LN'

Usage:

'LN' motor makes alignment with grids applying previously stored parameters NN0~NN5 and NPS. At execution, the forcer tries to rotate and make alignments. If alignment fails at clockwise rotation, it changes to rotate counter-clockwise and tries alignment again.

### 2.7.4 'SL'

Usage:

'SL' the command will execute of 'CL', 'NP', 'LNS', 'WE' commands in sequence. (It stores data for alignment)

## 2.8 Motor parameters 'DM', 'PX', 'MX', 'RF', 'AM', 'SF', 'LT', 'NB', 'MM', 'AS'

### 2.8.1. Motor directions 'DM'

Usage:

'DMn p1', where

n –

1 – X1-motor force module

2 – X2-motor force module

3 – Y1-motor force module

4 – Y2-motor force module

p1 – motor direction 1 (positive) or – 1 (negative)

Alternate usage:

'DM p1 p2 p3 p4', where

p1 – motor direction for X1-motor force module; 1 (positive) or – 1 (negative)

p2 – motor direction for X2-motor force module; 1 (positive) or – 1 (negative)

p3 – motor direction for Y1-motor force module; 1 (positive) or – 1 (negative)

p4 – motor direction for Y2-motor force module; 1 (positive) or – 1 (negative)

## 2.8.2. Sensor direction and motor mechanical pitch 'PX'

Usage:

'PX p1 p2 p3 p4'

p1 – definition of X1 sensor direction; 1 (positive) or – 1 (negative)

p2 – definition of X2 sensor direction; 1 (positive) or – 1 (negative)

p3 – definition of Y sensor direction; 1 (positive) or – 1 (negative)

p4 – motor mechanical pitch, in  $\mu\text{m}$  (fixed as 640)

## 2.8.3. Control force direction for close loop control 'MX'

Usage:

'MX p1 p2 p3'

p1 – control force direction for X-axis closed loop; 1 (positive) or – 1 (negative)

p2 – control force direction for Y-axis closed loop; 1 (positive) or – 1 (negative)

p3 – control force direction for rotational axis closed loop; 1 (positive) or – 1 (negative)

The three parameters are usually set to 1, do not change.

## 2.8.4. Factors for rotational control 'RF'

Usage:

'RF p1 p2 p3 p4'

p1 – factors for rotational control by X force modules; between –1 and 1

p2 – factors for rotational control by Y force modules; between –1 and 1

The rotational factors (p1 and p2) decide output of force modules. If p2 equals 0, Y-axis force module won't contribute to rotational motion.

p3 –  $L_y / L_{hx}$ , and sign of p3 must be set the same as p2

$L_y$  is the center distance of Y force modules along X-axis

$L_{hx}$  is the center distance of X1,X2 sensor along Y-axis

	$L_y$	$L_{hx}$	$L_y / L_{hx}$
LMSPX1	109	113	0.9646
LMSPX2	72	248	0.29
LMSPX3	280	247	1.1336

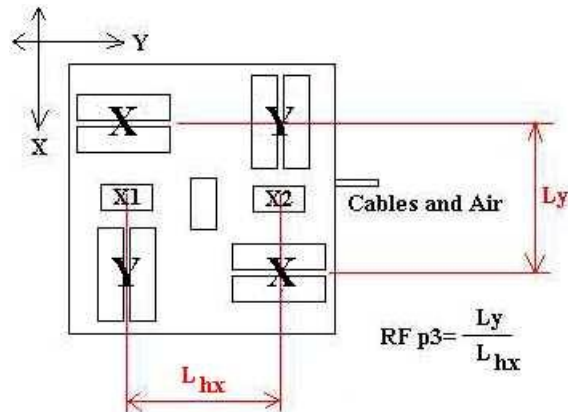


Figure 4 Diagram of RF parameter

p4 – sets effective force modules at alignment, default value is 0

0 – X force modules effective

1 – Y force modules effective

p5 – distance between X1 and X2 sensors, in mm

LMSPX1 forcer: 113

LMSPX2 forcer: 248

LMSPX3 forcer: 247

## 2.8.5. Current amplitude and SIN table 'AM'

Usage:

'AM p1 p2 p3'

p1 – current amplitude for servo-off mode (FX 0); the value is between 0 and 1; 1 corresponds to 3A

p2 – current amplitude for open loop mode (FX 2); the value is between 0 and 1; 1 corresponds to 3A

p3 – 1/8 length of sine table for sensor; the value is between 16 and 512

## 2.8.6. Current harmonic 'SF0'

Usage:



'SF0 p1 p2 p3 p4 p5 p6 p7'

This is used to set harmonics of output current

p1 – 1st harmonic (set to 1)

p2 – 2nd harmonic

p3 – 3rd harmonic

p4 – 4th harmonic

p5 – 5th harmonic

p6 – 6th harmonic

p7 – 7th harmonic

## 2.8.7. Field weakening 'LT'

Usage:

'LT p1 p2 p3 p4 p5'

p1 – starting velocity, in mm/s (for to Figure 5, Figure 6)

p2 – final velocity, in mm/s (for to Figure 5, Figure 6)

p3 – starting current; between 0 and 1 (for Figure 6)

p4 – final current; between 0 and 1 (for Figure 6)

p5 – final angle; in degree (for Figure 5)

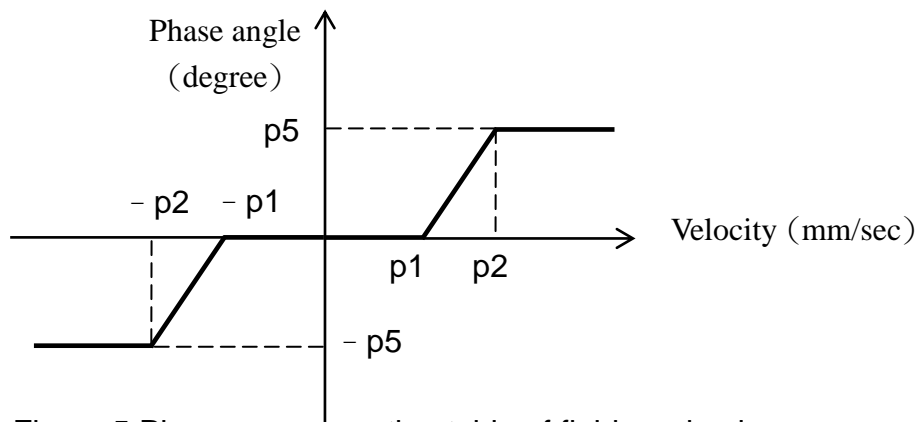


Figure 5 Phase compensation table of field weakening

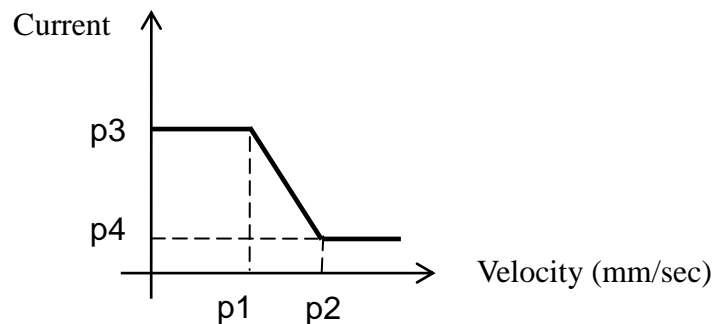


Figure 6 Modified current according to velocity

## 2.8.8. 'NB'

Usage:

'NB p1 p2 p3 p4 p5'

p1 – For stand-alone mode and demonstration program only. Delay time of internal program running after error, in ms; If during program running, error is too big, motion will be stopped for time p1, then running will continue.

The following four parameters are used in Figure 7:

p2 – V1; in mm/s

p3 – V2; in mm/s,  $V1 < V2$

p4 – Ampl\_pos, between 0 and 1

p5 – Ampl\_neg, between 0 and 1, value is given positive, LMDX deals with it as negative value internally.

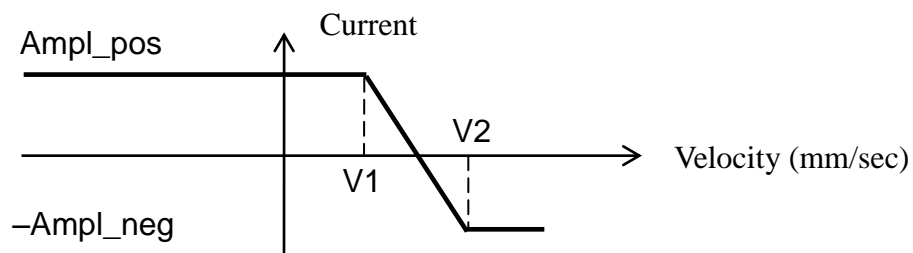


Figure 7 Current limit at alarm

## 2.8.9. Moment limits 'MM'

Usage:

'MM p1 p2 p3'

p1 – upper limit of moment, between 0 and 1.5; If moment exceeds p1 due to external torque, the output of force module will be reduced to value p3, and it remains in abnormal state of excess torque.

p2 – lower limit of moment, between 0 and 1; If moment becomes less than p2 during abnormal state of excess torque, the abnormal state will be finished. The output of force module will be set at nominal value (defined by 'LT' command)

p3 – maximum forces of force module during abnormal state of excess torque, the value is between 0 and 1.

## 2.8.10. Definition of axis directions 'AS'

Usage:

'AS p1 p2'

p1 – direction of X-axis; 1 (positive) or - 1 (negative)

p2 – direction of Y-axis; 1 (positive) or - 1 (negative)

Executing the command will change the definition of direction for each axis. This has to be done in open loop mode. Modifying the parameters of 'DM', 'PX', 'RF' directly has the

same effect.

Note that before using 'AS', parameters of 'DM', 'PX', 'RF' must be set correctly in advance.

After 'AS' is executed, command 'SL' must be issued to update related data.

## 2.9 Homing commands 'GSS', 'GP', 'GS'

The way LMSP finding its home position is through finding the "wall" on sides of stator. The following commands define conditions for finding the "wall".

### 2.9.1. Homing parameters 'GSS'

Usage:

'GSS p1 p2 p3 p4 p5 p6 p7'

p1 – stop force; the value is between 0 and 1

p2 – free force; the value is between 0 and 1

p3 – stop error; in  $\mu\text{m}$

p4 – free error; in  $\mu\text{m}$

p5 – homing velocity, in mm/s

p6 – stop noise time, no unit

p7 – free noise time, no unit

(see also GS command)

### 2.9.2. Homing parameters 'GP'

Usage:

'GP p1 p2 p3 p4 p5 p6 p7 p8 p9 p10'

p1 – homing direction of X-axis(direction of searching wall); 1 (positive) or - 1 (negative)

p2 – homing direction of Y-axis(direction of searching wall); 1 (positive) or - 1 (negative)

p3 – back distance after wall is found for X-axis, unit in mechanical pitch

p4 – back distance after wall is found for Y-axis, unit in mechanical pitch

p5 – X-coordinate at home, in  $\mu\text{m}$

p6 – Y-coordinate at home, in  $\mu\text{m}$

p7 – homing mode of X-axis

1 – home according to the nearest grid zero

0 – home according to grid zero in negative direction

p8 – homing mode of Y-axis

1 –home according to the nearest grid zero

0 –home according to grid zero in negative direction

p9 – display of X-axis sub-period of previous wall finding process

p10 – display of Y-axis sub-period of previous wall finding process (see also GS command.)

During homing for X-axis, if  $p9 > 480$  or  $p9 < 160$  (for motor pitch 640), it is recommended to set  $p7=1$ , else 0; For Y-axis, p10 and p8 are also set similarly.

### 2.9.3. Homing 'GS'

Usage:

'GS' executes homing, according to parameters of 'GSS' and 'GP'.

During homing, at first motor will search wall in the negative direction of X-axis. There are two conditions for finish of searching wall: motor output force becomes more than stop force (GSS.1) or position error becomes more than stop error (GSS.3). If the above two conditions are satisfied for more than stop noise time (GSS.6), motor starts to move in opposite direction until free conditions are satisfied. There are also two conditions for leaving wall: motor output force becomes less than free force (GSS.2) and position error becomes less than free error (GSS.4). After the above free conditions are satisfied for free noise time (GSS.7), motor stops and then moves to home, the moving distance is defined by 'GP' command. After X-axis homing is finished, Y-axis homing is performed similar to X-axis. Besides using GS command, user can execute homing via HOME (I1 input signal) of IO. Positive edge trigger executes homing. Similarly, use negative edge trigger to stop homing.

When succeed, it shows "OK.", otherwise, it prints "Err" with the following error codes,

- 1 – motion has been finished, but the wall has not been found.
- 2 – there was any key pressed on keyboard
- 3 – error happened during the process
- 4 – not in close loop
- 5 – already found

## 2.10 Compensation of position sensor 'CD', 'SF', 'ST'

### 2.10.1. Phase offset compensation 'CD'

Usage:

'CDn p1'

n –

1 – X1-axis

2 – X2-axis

3 – Y-axis

p1 – the difference between the phase shift of A-, B-signal and 90 degrees, in degree,

signed

0 – means the phase shift of A-, B-signal is 90 degree

Example:

If the phase shift between A- and B- phases is 80 degrees, the value to set is –10.

Alternate usages:

'CD p1 p2 p3'

p1 – the difference between the phase shift of A-, B-signal (X1-axis) and 90 degrees;  
in degree

p2 – the difference between the phase shift of A-, B-signal (X2-axis) and 90 degrees;  
in degree

p3 – the difference between the phase shift of A-, B-signal (Y-axis) and 90 degrees; in  
degree

## 2.10.2. Harmonic compensation 'SF'

Usage:

'SFn p1 p2 p3 p4 p5 p6 p7'

n –

1 –harmonic compensation for X1 sensor

2 –harmonic compensation for X2 sensor

3 –harmonic compensation for Y sensor

p1 – 1st harmonic (set to 1)

p2 – 2nd harmonic

p3 – 3rd harmonic

p4 – 4th harmonic

p5 – 5th harmonic

p6 – 6th harmonic

p7 – 7th harmonic

## 2.10.3. Step for position table generating 'ST'

Usage:

'STn p1'

n –

1 – X1-axis

2 – X2-axis

3 – Y-axis

p1 – step for position table generation, in  $\mu\text{m}$ , the default value is 0.2  $\mu\text{m}$ . Small value increases time for table calculation. If too big, during position table generating, it will automatically recalculate with 10% less value and shows message "ST n p1",

where

n –

1 – X1-axis

2 – X2-axis

3 – Y-axis

p1 – new step for position table generation

After changing parameters in command 'CD', 'SF' and 'AM.3', position tables are recalculated automatically. After execution, use 'WE' to save parameters.

## 2.11 IO related commands 'WN', 'PN', 'DN'

Commands 'WN', 'PN' are brought into motion buffer before performance. (similar to 2.1 motion commands)

### 2.11.1. Waiting for specific input 'WN'

Usage:

'WN p1'

p1 –IO input value, minimum is 0, maximum is 255 (decimal number); the command is used to wait IO input state. If input state matched with p1, it proceeds to execute the next command in motion buffer. For preparing p1, transform the value into binary, definitions of each bit are as follows :

bit0 - IO pin I1(HOME)	bit4 - IO pin I5
bit1 - IO pin I2	bit5 - IO pin I6
bit2 - IO pin I3	bit6 - IO pin I7
bit3 - IO pin I4	bit7 - IO pin I8(RESET)

Example: p1 = 2, then

bit0 - 0	bit4 - 0
bit1 - 1	bit5 - 0
bit2 - 0	bit6 - 0
bit3 - 0	bit7 - 0

(0 – low level, 1 – high level)

This command is only for stand-alone mode.

### 2.11.2. Output command 'PN'

Usage:

'PN p1'

p1 – state of IO output pin. The command is used to send signal to IO. For preparing p1, transform the value into binary, definitions of each bit are as follows :

Example: p1 = 16, then

bit0 – 0	bit4 - 1
bit1 – 0	bit5 - 0
bit2 – 0	bit6 - 0
bit3 – 0	bit7 - 0

(0 – low level, 1 – high level)

This command is only for stand-alone mode.

### 2.11.3. Display of I/O state 'DN'

Usage:

'DN p1 p2'

p1 – decimal value of output state, minimum is 0, maximum is 255

p2 – decimal value of input state, minimum is 0, maximum is 255

There are 8 inputs and 8 outputs for IO. Two of inputs are already used. Includes Home and RESET. Also five of the outputs are already used. Includes In-Position, SVRDY, ALM, HOMED and HOMING.

### 2.12 Power on setting 'RP'

Usage:

'RP p1 p2'

p1 – decimal number ( – 1 to 63 )

For understanding, the number must be transformed into binary, meaning of each bit are as follows:

- | bit | function  |
|-----|---|
| 0   | closed-loop mode after power-on (same as FX 1)          |
| 1   | open loop mode after power-on (same as FX 2)            |
| 2   | position sensor signals auto-calibration (same as 'CL') |
| 3   | automatic homing (same 'GS')                            |
| 4   | automatic alignment with grids (same as 'LN')           |
| 5   | automatic execution of motion program (same as '!')     |

If p1 is set as –1, system loads default values for all parameters.

Each bit corresponds to each function, when bit set as 1, the corresponding function is enabled; bit set as 0, it is disabled.

Example:

To make the motor to execute automatic homing, automatic alignment in closed-loop mode after power-on, p1 should be set as binary (011001) namely decimal (25); Similarly if automatic execution of motion program is wished, then it would be 57 (binary 111001).

p2 – KV.1 increase or decrease its value with this p2 factor, each time '+' or '-' was pressed during program execution. The value is between 1 and 10.

## 2.13 Pulse mode command 'SD'

Usage:

'SD p1 p2 p3 p4 p5 p6'

p1 – 0 stand-alone mode; 1 pulse mode; 2 CW/CCW mode; 3 AqB mode

p2 – speed for moving to the nearest grid zero, when Reset signal of pulse interface is engaged, in mm/s

p3 – after the reset signal of pulse interface is engaged, it delays p3, then Ready signal becomes high, in ms

p4 – pulse weight of X-axis; the distance to travel in X-axis for each pulse it receives, signed, in  $\mu\text{m}/\text{pulse}$

p5 – pulse weight of Y-axis; the distance to travel in Y-axis for each pulse it receives, signed, in  $\mu\text{m}/\text{pulse}$

p6 – acceleration factor for feedforward, the value is between 0 and 1

## 2.14 Motion programming commands '@', 'DP', '!', 'PS'

### 2.14.1. Loading motion program '@'

Usage:

'@

*first line of program*

....

*last line of program*

@'

Motion programs can include motion-, display- or other commands. The capacity of memory for program is 32k bytes.

The program is stored in flash ROM immediately after input.

### 2.14.2. Display motion program 'DP'

Usage:

'DP'

Press the “Enter” key to stop display, and press any key to continue displaying next page.

### 2.14.3. Execution of motion program '!'

Usage:

'!'

The following characters could be pressed during execution:

+, – increases/decreases velocity (see RP.2)



- P        pauses/continues the execution of motion programs
- Enter    cancels the execution of motion programs

## 2.14.4. Debugging motion programs 'PS'

Usage:

'PS p1 p2'

- p1 - 1 - step by step execution of motion program is enabled. (Press any key except "Enter" to proceed to the next program line)
- 0 - disable debugging. Motion program runs continuously. (This is the default value after power-on)
- p2 - setting whether echoing program lines
  - 1 - echo
  - 0 - no echo

## 2.15 Display of position sensor signals 'G', 'GC'

### 2.15.1 single display 'G'

Usage:

'G p1 p2 p3 p4 p5 p6'

- p1 – phase A value of X1 sensor (ADC0)
- p2 – phase B value of X1 sensor (ADC1)
- p3 – phase A value of X2 sensor (ADC2)
- p4 – phase B value of X2 sensor (ADC3)
- p5 – phase A value of Y sensor (ADC4)
- p6 – phase B value of Y sensor (ADC5)

The value is between –2048 and 2047.

### 2.15.2 Continuous display 'GC'

Usage:

'GC p1 p2 p3 p4 p5 p6'

This is similar to command 'G', except that it displays continuously, press "Enter" key to stop display.

## 2.16 PID gains 'KX', 'KY', 'KF'

Usage:

'Kn p1 p2 p3 p4 p5'

- n - X - X-axis
- Y - Y-axis
- F - rotational movement

- p1 – P gain (proportional gain)
- p2 – I gain (integral gain)
- p3 – D gain (differential gain)
- p4 – acceleration factor
- p5 – filter factor for position sensor. Range 0 to 1.0. The smaller value, the stronger filtering

## 2.17 Flash ROM read/write 'RE', 'WE'

### 2.17.1. 'RE' – reads all stored parameters from flash ROM

Usage:

'RE'

This function is automatically executed after power on, if nothing wrong with flash ROM.

### 2.17.2. 'WE' – writes all parameters to flash ROM

Usage:

'WE'

## 2.18 Motion modes 'FX'

Usage:

'FX p1'

p1: 0 – servo off mode

1 – closed-loop control mode. After switching to this mode, motor will automatically make alignment to grids.

2 – open loop mode

If motor becomes in servo on state (FX 1 finished), LMDX will output SVON(O4 signal) of IO to notice user the state. If it is low, motor is in servo off state or open loop mode.

## 2.19 Driver's version 'VER'

Usage:

'VER p1'

Firmware version. Please do not change it.

## 2.20 Reset driver 'RES'

Usage:

'RES'

It is similar to reboot LMDX. User can input positive edge trigger signal from RESET (I8 signal) of IO to execute reset function.

## 2.21 DSP time parameters 'DS', 'TI', 'SP'

### 2.21.1 DSP frequency 'DS'

Usage:

'DS p1'

p1 - DSP frequency (MHz), default is 150, do not change

### 2.21.2 DSP interrupt time 'TI'

Usage:

'TI p1'

p1 - DSP interrupt time ( $\mu$ s), default is 250, do not change

p2 - noise time for WN command, in ms

### 2.21.3 RS232 baud rate 'SP'

Usage:

'SP p1'

p1 - There are 1200, 2400, 4800, 9600, 19200, 38400, 56000, 57600 and 115200 for RS232 setting, default value is 115200, please do not change.

## 2.22 Error compensation

### 2.22.1. Linear error compensation 'TKX', 'TKY', 'TKA', 'TKB'

Usage:

'TKn i M dat0 dat1 ... datN', where:

n - X - X-axis forward

Y - Y-axis forward

A - X-axis backward

B - Y-axis backward

i - specify the  $i$ -th data to begin with data input (minimum is 0, If  $i = 0$ , input data begins with *dat0*)

M - total number of input data

dat0 .... datN - compensation data table, maximum of N is 599

Compensation data table is explained as follows: (all the data are in  $\mu$ m except the first value *dat0*)

For the compensation table to be effective, homing process must be done successfully at first. See also 'GS' and 'FR'.

*dat0* – the first data is not a compensation value, it is used to indicate the period of compensation, this is also the measurement pitch when collecting error compensation values, its unit is 0.64 mm. For one compensation table, there is only one such period. If setting it to a value less than 1, compensation table will be disabled.

For example, if measurement is made every 0.64 mm, *dat0* is set 1. And if measurement is made every 6.4 mm, *dat0* is set 10.

- dat1* – error at period\*0, i.e. error of the initial point, usually set to 0
- dat2* – error at period\*1
- dat3* – error at period\*2
- dat4* – error at period\*3
- dat5* – error at period\*4
- .....
- datN* – error at period \*(N – 1), the maximum of N is 599

When preparing the compensation table values, use the actually measured errors, e.g., If laser interferometer reports position of 29997  $\mu\text{m}$  while it has been commanded to move to 30000  $\mu\text{m}$ , then -3  $\mu\text{m}$  is input to build the table. Movement in the negative coordinate, will not be compensated. Bidirectional compensation can be applied to acquire better accuracy. Note that all the compensated values are integer only and in the range of -127 to 127.

## 2.22.2. Display of error compensation table 'NKX', 'NKY', 'NKA', 'NKB'

Usage:

'NK*n* i M', where:

*n* – X – X-axis forward

Y – Y-axis forward

A – X-axis backward

B – Y-axis backward

*i* – display begins with the *i*-th data in compensation data table (minimum is 0)

*M* – total number of data to display

For example, "NKX 0 10" shows the error compensation values "*dat0 dat1 dat2 ... dat9*".

## 2.23 In-Position 'INP'

Usage:

'INP p1 p2 p3'

p1 – setting linear tracking error window for X-Y axes, the unit is  $\mu\text{m}$ .

p2 – delay time for In-Position, the unit is ms.

p3 – setting angular tracking error window for rotational axis, the unit is arc sec.

After path planning is finished, the absolute value of linear tracking error in X-Y axes becomes within p1, and the absolute value of angular tracking error in rotational axis becomes within p3, then the errors are within window. This status must last for a time of at least p2, so that In-Position becomes true. This signal outputs from In-Position (O6 signal) of IO. Please refer to IO section of 'LMDX User Guide'.

## 2.24 Data capturing function 'NS', 'CB', 'O'

### 2.24.1 Data capturing setting 'NS'

Usage:

'NS p1 p2 p3 p4 p5 '

It is used to set the physical quantities to be captured into driver's buffer. Up to 3 physical quantities could be captured at one time. If only one quantity to capture, set p2 and p3 to 0. If only two quantities, set p3 to 0. For p1, p2, p3, please refer to the number in the following table:

p1 – first pointer to a physical quantity

p2 – second pointer to a physical quantity

p3 – third pointer to a physical quantity

p4 – how many data totally to capture for all physical quantities ; max. 8000

p5 – every p5 interrupts, do capture; (sampling frequency)

Number	Variable	Format	Explanation
1	Analog[0]	Integer	Sensor signal, X1 A
2	Analog[1]	Integer	Sensor signal, X1 B
3	Analog[2]	Integer	Sensor signal, X2 A
4	Analog[3]	Integer	Sensor signal, X2 B
5	Analog[4]	Integer	Sensor signal, Y A
6	Analog[5]	Integer	Sensor signal, Y B
7	ABS_LIN[0]	Float	Feedback X1-position
8	ABS_LIN[1]	Float	Feedback X2-position
9	ABS_LIN[2]	Float	Feedback Y-position
10	s_dem[0]	Float	Command X-position
11	s_dem[1]	Float	Command Y-position
12	XN_1[0]	Float	Error rotational, (n-1)
13	XN_1[1]	Float	Error X, (n-1)
14	XN_1[2]	Float	Error Y, (n-1)
15	YN[0]	Float	Force rotational
16	YN[1]	Float	Force X

17	YN[2]	Float	Force Y
18	XXN[0]	Float	Error rotational
19	XXN[1]	Float	Error X
20	XXN[2]	Float	Error Y
21	A0[1]	Float	Feed-forward X
22	A0[2]	Float	Feed-forward Y
23	A0[2]	Float	Feed-forward Y

For example:

"NS 7 9 0 2000 1" This command will set the driver, so that the feedback position of X1 and Y will be captured. And there would be totally about 2000 captured data, including X1 and Y. And every 1 CPU interrupt cycle, data will be captured.

If you set p4 to be bigger, there would be more captured data, which means also longer data transfer time later.

The capturing will occur every time when you run with "PR" or "PA" commands in stand alone mode.

It is also possible to do capture in pulse command mode: The capturing will occur every time when you enter "CB" (Clear Buffer) command.

### **2.24.2 Clear Buffer 'CB'**

Usage:

'CB '

To do capture in pulse command mode: The capturing will occur every time when you enter 'CB' command. Enter 'CB' and very quickly after that, send pulses to move the motor. Be sure that there is as little time as possible between your pressing enter key and sending pulses.

### **2.24.3 Download captured data to PC 'O'**

Usage:

'O '

After data are captured, it is possible to use this command to download the data to DspHost.exe via RS-232. After pressing 'O' and enter key, DspHost.exe will start downloading and show a progress bar.

## **2.25 Software Limits 'LMT'**

Usage:

'LMT p1 p2 p3 p4'

p1 - limit for X-axis positive direction.

p2 - limit for X-axis negative direction.

p3 - limit for Y-axis positive direction.

p4 - limit for Y-axis negative direction.

If motion command is out of the above range, an alarm code of 0x0400 is generated. Motor stops immediately and it will not receive any further motion commands. To clear this alarm, please refer to section 2.4.1. After clearing the alarm, only motion commands to leave software limits can be accepted, otherwise alarm will happen again.