

## *Application notes for AKD2G™*

### **First servodrive AKD2G tuning**

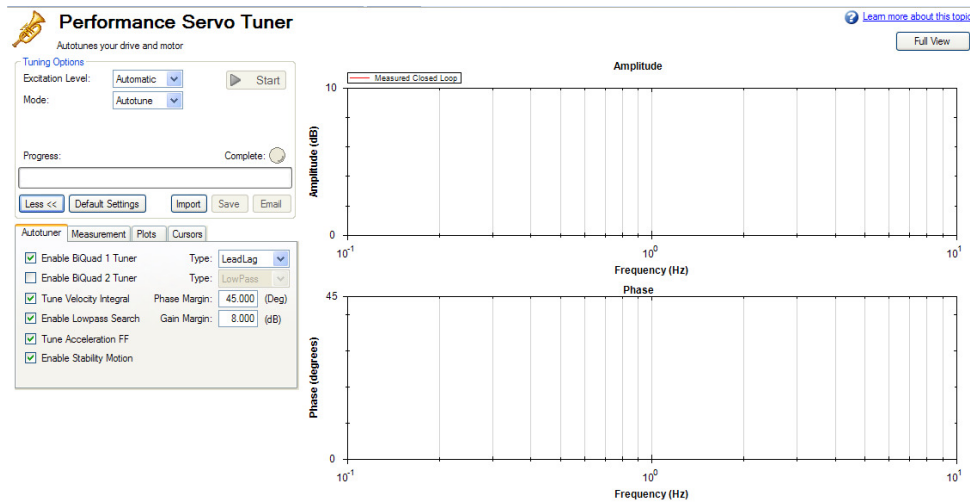
Users facing the tuning of an AKD2G servodrive for the first time will have to use the Workbench configuration software tool which, for novice users or users of the Servostar series drive, may confuse them given the difference in the graphical interface and the many features introduced.

This note will guide you in setting a general configuration indicating the most significant parameters to be configured (v. [Manual Tuning](#))

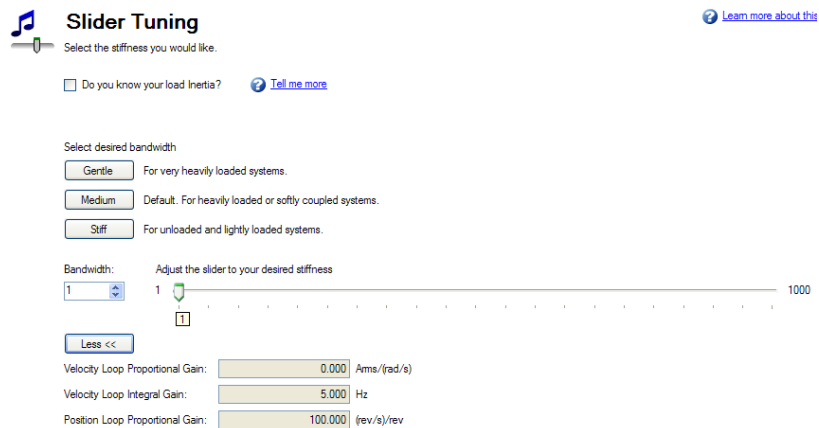
## Tuning Mode

Workbench allows different tuning mode.

The more experienced user will appreciate *Performance Servo Tuner*



while the less experienced user can easily use the *Slider Tuning*.



However, both in the first case and in the second one may encounter difficulties linked to the mechanical and kinematic characteristics of the load to be handled (vertical load, limited stroke, resonance frequencies, protection intervention, etc.). All this can make it complicated if not impossible to perform an automatic tuning.

Given the large number of parameters of the AKD2G, only the parameters necessary for the regulation will be illustrated and set below to obtain a first reliable operation applicable to any mechanical and kinematic situation.

For the moment we leave out the *Performance Servo Tuner* and *Slider Tuning*.

It is possible that the suggested values will have to be modified later, both to obtain a fine tuning of the kinematics and in case of kinematic problems.

## Manual Tuning

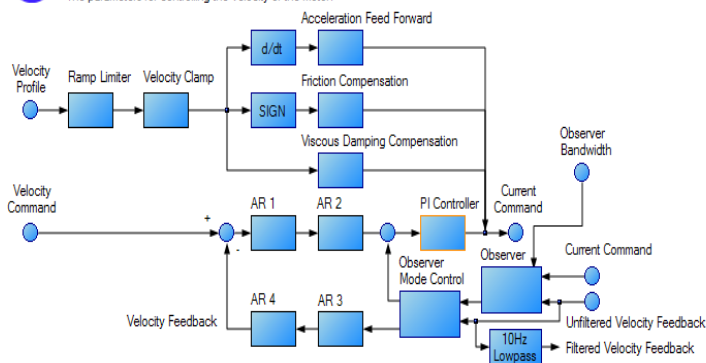
The parameters set below refer to the speed and position loop. The user can adopt the suggestions by acting on the parameters shown in the screens of the windows involved in the various functions or manually using the terminal function.

### Velocity Loop



#### Velocity Loop

The parameters for controlling the velocity of the motor.



### PI Controller

Velocity Clamp	AR Filter	PI Controller	Observer	Source	Status
Proportional Gain: <input type="text" value="0.500000"/> Ams/(rad/s)					
Integral Gain: <input type="text" value="10.000000"/> Hz					
Integrator Mode: <input type="text" value="1 - Integrators Alternating"/>					
<input type="radio"/> Velocity Integrator <input checked="" type="radio"/> Position Integrator ⚠ Integrator activation driven by integrator threshold.					
Integrator Threshold: <input type="text" value="15'000.000000"/> rpm					
Trajectory Command: <input type="text" value="0.000000"/> rpm					

The integral gain will initially be set to 10 Hz.

$$\text{AXIS1.VL.KI} = 10$$

The proportional gain value can initially be calculated with this formula:

$$\text{AXIS1.VL.KP} = 500 * J_{\text{tot}} / K_t$$

where:

$$J_{\text{tot}} = \text{load inertia} + \text{motor inertia (Kg m}^2\text{)}$$

$$K_t = \text{motor torque constant (Nm/A)}$$

If the inertia of the load compared to the motor shaft is not known, set a value from those in the table present according to the rated power of the motor power P<sub>n</sub>:

P <sub>n</sub> < 1kW	VL.KP=0.02
1kW < P <sub>n</sub> < 2kW	VL.KP=0.08
2kW < P <sub>n</sub> < 8kW	VL.KP=1
P <sub>n</sub> > 8kW	VL.KP=4

Starting from these initial values, increase AXIS1.VL.KP until the motor starts to vibrate, then reduce it.

**AR Filter**

If the motor has high resolution feedback (optical sin-cos encoder, Hiperface, EnDat 2.1 oe 2.2 full digital) set up filters in this way:

AR2,AR3 e AR4 = *Unity Gain*  
AR1 = *BiQuad*

In summary:

AXIS1.VL.ARTYPE1 = 4  
 AXIS1.VL.ARTYPE2 = 0  
 AXIS1.VL.ARTYPE3 = 0  
 AXIS1.VL.ARTYPE4 = 0  
 AXIS1.VL.ARZF1 = 200  
 AXIS1.VL.ARPF1 = 160  
 AXIS1.VL.ARPQ1 = 0.707  
 AXIS1.VL.ARZQ1 = 0.707

Select AR Type:

AR 1	BiQuad
AR 2	Unity Gain
AR 3	Unity Gain
AR 4	Unity Gain

Filter Type: 4 - BiQuad

Numerator: Frequency: 200.000 Hz Q: 0.707

Denominator: Frequency: 160.000 Hz Q: 0.707

If the motor has low resolution feedback (TTL square wave encoder, resolver, SFD3) of Hiperface DSL encoder set up filters in this way:

AR2 = *Unity Gain*  
AR1, AR3 e AR4 = *BiQuad*

In summary:

AXIS1.VL.ARTYPE1 = 4  
 AXIS1.VL.ARTYPE2 = 0  
 AXIS1.VL.ARTYPE3 = 4  
 AXIS1.VL.ARTYPE4 = 4  
 AXIS1.VL.ARZF1 = 200  
 AXIS1.VL.ARPF1 = 160  
 AXIS1.VL.ARPQ1 = 0.707  
 AXIS1.VL.ARZQ1 = 0.707  
 AXIS1.VL.ARZF3 = 5000  
 AXIS1.VL.ARPF3 = 1000  
 AXIS1.VL.ARPQ3 = 0.707  
 AXIS1.VL.ARZQ3 = 0.707  
 AXIS1.VL.ARZF4 = 5000  
 AXIS1.VL.ARPF4 = 1000  
 AXIS1.VL.ARPQ4 = 0.707  
 AXIS1.VL.ARZQ4 = 0.707

Select AR Type:

AR 1	BiQuad
AR 2	Unity Gain
AR 3	BiQuad
AR 4	BiQuad

Filter Type: 4 - BiQuad

Numerator: Frequency: 200.000 Hz Q: 0.707

Denominator: Frequency: 160.000 Hz Q: 0.707

Select AR Type:

AR 1	BiQuad
AR 2	Unity Gain
AR 3	BiQuad
AR 4	BiQuad

Filter Type: 4 - BiQuad

Numerator: Frequency: 5'000.000 Hz Q: 0.707

Denominator: Frequency: 1'000.000 Hz Q: 0.707

Select AR Type:

AR 1	BiQuad
AR 2	Unity Gain
AR 3	BiQuad
AR 4	BiQuad

Filter Type: 4 - BiQuad

Numerator: Frequency: 5'000.000 Hz Q: 0.707

Denominator: Frequency: 1'000.000 Hz Q: 0.707

**Observer (to be use mainly with low resolution feedback)**

Velocity Clamp	AR Filter	PI Controller	Observer	Source	Status
Observer Enabled: 1 - Enabled					
Observer Gain: 1'000.000000 (rad/s^2)/A					
Observer Bandwidth: 500.000000 Hz					

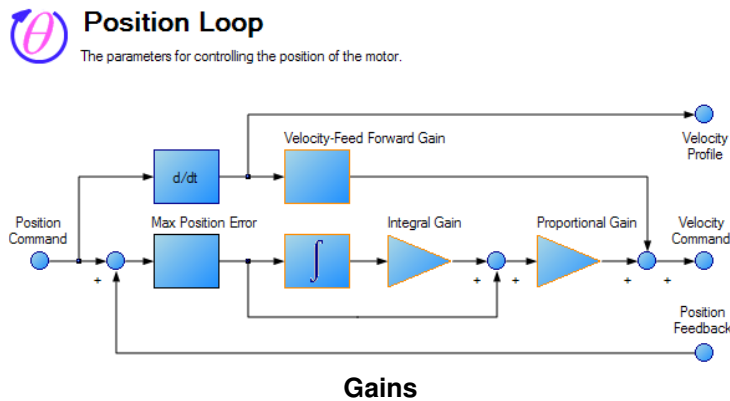
Observer Mode                    AXIS1.OBS.ENABLE       = 1  
 Observer gain                    AXIS1.OBS.KO           =  $K_t/J_{tot}$  (motor torque constant divided by total inertia)  
 Observer Bandwidth            AXIS1.OBS.BW         = 500

If you do not know the inertia of the load, use the inertia of the motor multiplied by 2 for the calculation:

$J_{tot} = J_m * 2$

**Position loop**

Skip this part if the AKD operating mode is different from *Position Mode*.



**Gains**

Gains	Command Prefilter	Limiter	Source	Status
Proportional Gain:		10.000000	Hz	Integral Output Saturation: 3'000.000000 rpm
Integral Gain:		0.000000	Hz	
Integrator Mode:		1 - Integrators Alternating		
		<input checked="" type="radio"/> Velocity Integrator		
		<input checked="" type="radio"/> Position Integrator		
		<input type="checkbox"/> Integrator activation driven by integrator threshold.		
Integrator Threshold:		15'000.000000	rpm	
Trajectory Command:		0.000000	rpm	
Feed Forward Gain:		1.000000		

Set the initial parameters like this:

AXIS1.PL.KP                    = 10  
 AXIS1.PL.KI                    = 0  
 AXIS1.VL.KVFF                 = 1

## Limits

Set the limits as follows:

AXIS1.IL.LIMITP = x (application requirements in A)  
 AXIS1.IL.LIMITN = -x (application requirements in A)  
 AXIS1.VL.VFTHRESH = y (application requirements, e.g. motor max speed multiplied by 1.2)

### Limits

This page shows all the drive limits in one place.

<b>Current Limits</b>		
Positive Peak Current:	<input type="text" value="18.000"/>	Ams
Negative Peak Current:	<input type="text" value="-18.000"/>	Ams
Dynamic Brake Peak Current:	<input type="text" value="1.000"/>	Ams

<b>Velocity Limits</b>		
Positive Speed Limit:	<input type="text" value="15'000.000"/>	rpm
Negative Speed Limit:	<input type="text" value="-15'000.000"/>	rpm
User Over-Speed Limit:	<input type="text" value="8'000.001"/>	rpm
Overall Over-Speed Limit:	<input type="text" value="3'600.000"/>	rpm

<b>Position Limits</b>		
Maximum Position Error:	<input type="text" value="65'536.000"/>	Counts16Bit
HW Positive Limit Switch:	<input type="text" value="0 - Not Configured"/>	
HW Negative Limit Switch:	<input type="text" value="0 - Not Configured"/>	
SW Limit Switch 0: <input type="checkbox"/>	<input type="text" value="0.000"/>	Counts16Bit
SW Limit Switch 1: <input type="checkbox"/>	<input type="text" value="1'048'576.000"/>	Counts16Bit

**Motor and Drive Limits**

These limits are set automatically through thermal protection:

## Further considerations

It should be clear that the values of the speed gains (AXIS1.VL.KP and AXIS1.VL.KI) and of the position (AXIS1.PL.KP) will be optimized according to the result to be achieved. AXIS1.VL.KP is proportional to the total inertia so it can be increased in steps until reaching a high motor noise, and then decrease it to the value of the previous step. AXIS1.PL.KP can instead be increased until the desired tracking error is reached, considering that a too high value may cause overshoot of speed or oscillations at low speed. Finally, AXIS1.VL.KI must be commensurate with the overall inertia of the system (lower than 10Hz for high inertia, higher but lower than 50Hz for low inertia).