Step by step setup/tuning procedure for R2000 drives.

Introduction:

This document outlines how to setup and tune the Rutex brush motor drives R2010 (100V/20A), R2020 (200V/40A), brush-less drive R2030 (100V/20A) and the analogue output R2040 (+/-10V).

This document is targeted for new users of the Rutex products. The entire test should be run on the bench before installation of the servo drives on the machine. The purpose of this document is to familiarize the user with the R2000 servo drive prior to installation on a machine. It deals with following issues:

- Connection of the R2000 drive to the PC.
- Installation and the functionality check of test software such as R2xTune for Windows XP/2000.
- Connection of the different types of encoders to the R2000 servo drive as well as describing advantages and disadvantages of different types of encoders
- Connection of the servomotor, phasing the hall feedback / motor phases for brush-less drives and testing the DC output of the R2040 servo drive.
- And finally, tuning of the drive.
- Dealing with electrical noise and interference.

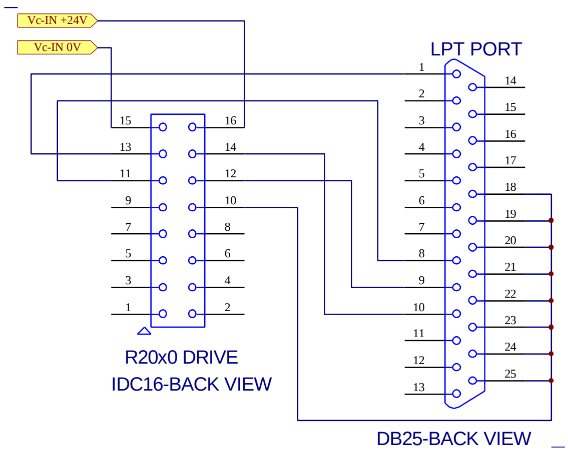
1. Connection of the drives:

The PC based test software communicates with the R2000 drives via the printer port. It uses a SPI protocol (Serial Peripheral Interface). The Step input is used as the Clock input and the Direction input is used as Data-in. There are two more lines required for PC/R2000 drive communication: Data-out and Stepper/SPI. The Stepper/SPI input to the drive is the signal, which directs the drive to switch between the standard Step&Dir mode and the SPI mode. The Data-out and Stepper/SPI lines have to be connected back to printer port pins either by a custom cable or jumpers on J2 of the R2110/R2120 motherboard – if a motherboard is used. Once the SPI tuning is done, these two lines can be disconnected from printer port pins and again used for their normal I/O functions.

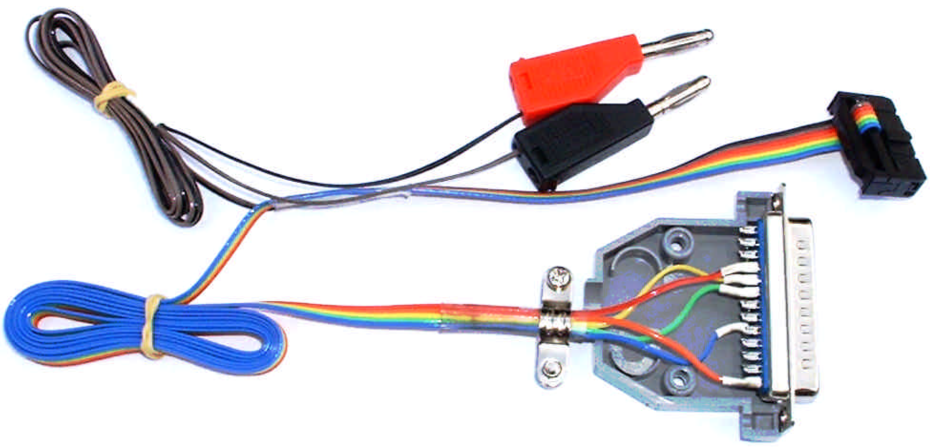
Start with one drive only.

If you are **not using the R21x0** (motherboard), then make a short *custom cable* having:

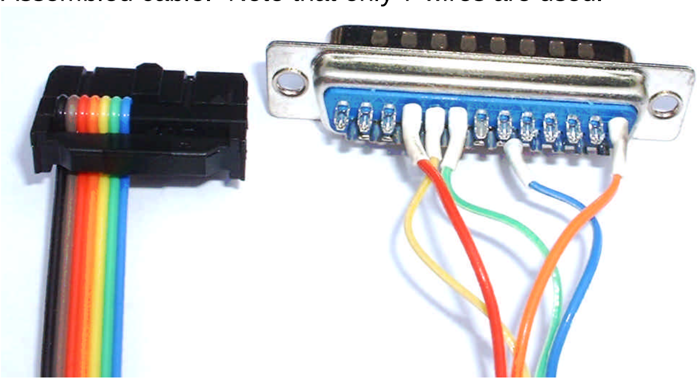
- DB25M (25 pin D connector available from your local computer/electronic store or www.didgikey.com p/n 1225M-ND)
- IDC16 cable (www.digikey.com p/n C3AAG-1636M-ND. Note that this cable comes with crimped connectors on both ends. You can cut off the connector from one end.



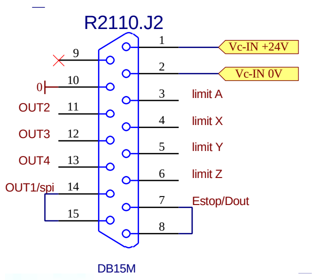
Above a custom made cable for tuning Rutex R20x0 servo drive configured for A-axis.

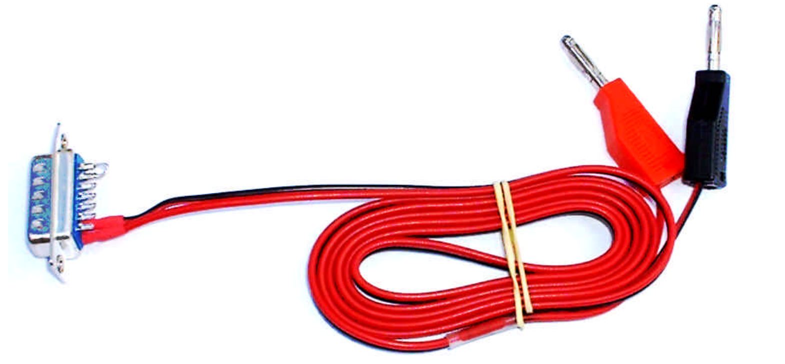


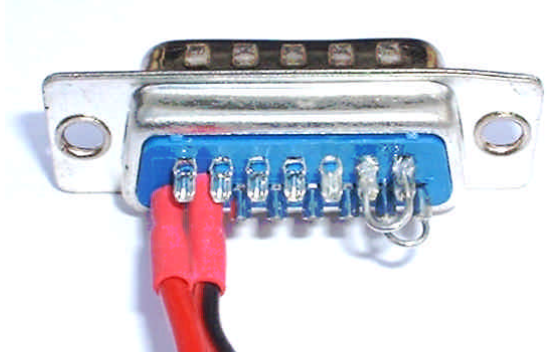
Assembled cable. Note that only 7 wires are used.



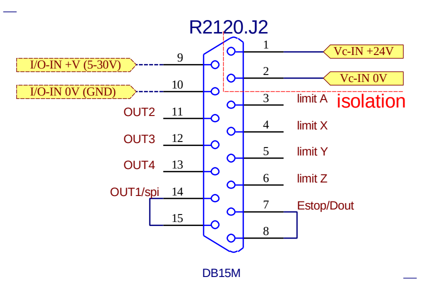
If you are **using a R2110 or R2120 motherboard,** then you will have to make a connection to a DB15M connector (www.didgikey.com p/n 1215M-ND for example) for supply and two (jumpers) links. Please follow the schematic below. Then you will also need a straight through cable (DB25M-DB25F) for connection between the PC and R21x0 motherboard. This cable is sometimes called as printer extension cable. Use quality-shielded cable not longer then 2 meters (www.digikey.com p/n AE1012-ND).







Above is an SPI cable for R2110 non-isolated motherboard.



The picture above shows the connection for cable for R2120 opto-isolated motherboard. Note that during testing a single 24V power supply can be used. In normal operation two floating isolated power supplies should be used to preserve the opto-isolation.

Note that this is a minimal connection for Setup&Tune (SPI mode). Please refer to the datasheets R20x0.PDF and R21x0.PDF for description of pin-out for digital I/O connection for limit switches and relay outputs etc.

If you are using the R21x0 motherboard, then plug one R20x0 servo drive into motherboard – preferable in slot 4 (A-axis). Connect DB15 connector into the motherboard and connect DC24V to Vc-IN terminals (pin.1&2 of R21x0.J2). Observe the correct polarity – **do not use AC voltage**. The minimum voltage is DC18V and maximum is DC35V.

If you are using the custom cable as described above, then connect the supply to DC24V power supply. Any voltage between DC18V to DC35V is fine. You can use a low cost 300mA or more wall outlet power supply available from Radio Shack or Jameco. **Do not connect any AC voltage directly to the R20x0 drives!**

Do not connect the encoder, motor or the DC motor power supply to the Vm terminals at this stage.

Make sure that the LED on the R20x0 drive is flashing (about four flashes per second) after the power is switched on.

NEVER 'hot-swap' drives in a motherboard. Always disconnect Vc supply when plugging or unplugging servo drives.

2. Test software:

At the time of writing of this document there are three versions of tuning software for R20x0 drives (WinXP/2000). They can be downloaded from Rutex website.

(1) http://www.rutex.com/zip/R2xTuneVB6.msi

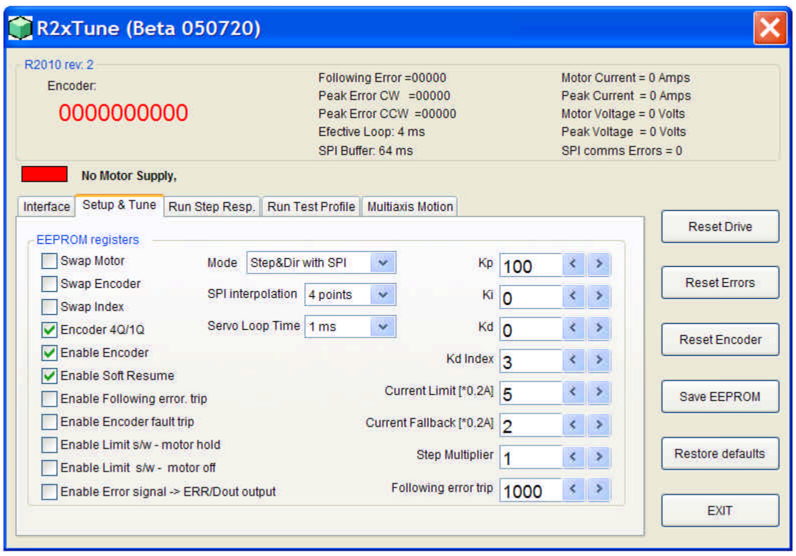
(2) http://www.rutex.com/zip/R2xTune.msi

(3) http://www.rutex.com/zip/R2xSetup.msi

all of them are functionally same, with slightly different GUI (graphics user interfaces). The (1) (2) versions are preferred – easier to use GUI, but there was a report that (2) it does not work properly with certain video card. Version (1) runs fine on XP only. The (3) version seems to be stable, but it runs under the Microsoft's dotNET framework. If the dotNET framework is not installed on your PC, then the program will report an '0xc0000135 error' and you will have to download and install the dotNET framework from Microsoft or MSDN website. Just search the Microsoft website or Google for DotNETfx.EXE – it is about a 20MB file. Never install both versions of the software at the same time. Always un-install one before installing second one. All the reference below is to (2) version.

Download the R2xTuneVB6.msi or R2xTune.msi or R2xSetup from Rutex homepage or use the direct link above. Note that the link is case sensitive. The MS Windows should automatically start installation after downloading. If not, click on the file with right button of your mouse and scroll down to 'Install'. Follow the installation instruction – do not forget to restart the computer! The program can be uninstalled from Control Panel -> Add or Remove programs – if needed.

Switch on the Vc power to the drive and run the R2xTune program. Make sure that no encoder, Vm or motor is connected at this stage – only the 24V Vc, two links on DB15 and cable between printer port and motherboard or drive. The program should indicate that drive has been found with default values.



It is important to get the communication between PC and R20x0 servo drive correct – in fact, there is absolutely no point to go any further steps, unless the communication is established. If program indicates *"Drive not found"* or *"SPI error"* then there is a problem, which must be fixed before going any further.

Description of R2xTune program behavior: The program polls the data from drive about every 100ms and then refreshes the information on the screen. The very first information that the program reads from the drive is the drive type. If the program receives a one of the valid drive types (2010, 2020 etc.) then it reads rest of the packet. Although, if the program receives 0xFFFF or 0x0000 (Data line permanently high or low) then the program assumes that there is no drive connected. Alternatively, if the program receives an invalid drive type, then it assumes that the packet is corrupted and indicates SPI error. In case of the SPI error or Drive not found, the program goes in stand-by mode for 1 second and then tries to read the packet again.

Fault finding hints:

?? Go to your PC BIOS (usually press F2 or DEL during boot-up) and check that the printer port is set to LPT1: (0x378 hexadecimal) and that it is set as a standard printer port (SPP) or bi-directional, although it should work in ECP mode as well.

?? Make sure that no Windows printer, including virtual such some versions of Acrobat Writer, or any other device driver is using the LPT1: port. Go the Settings -> Printers and check that none of the printers is using LPT1 port. Also check the Control Panel -> System -> Device Manager.

?? Some of the laptops are not able to produce 5 Volt level signals – the output is 3.3 Volts. The input to R20x0 servo drive is an Schmitt trigger with a logical low at 1.7 Volts and logical high at 3.3 Volts. Some of the 3.3 Volts printer ports might just work, but there is no room for noise – you might experience SPI errors once the motor and Vm are connected. Use a desktop computer or an R2120 opto-isolated motherboard. The R2120 is compatible with 3.3 and 5 Volt printer ports.

?? Make sure that the cable between the PC and motherboard/drive is a good quality, shielded cable no longer than 6-ft.

?? By far, the most common problem is incorrect wiring. Check it again.

?? Try to run the program from a different PC or reinstall the XP/W2K and try to run R2xTune on a fresh “un-polluted” Windows operating system.

?? Try to run the other version of tuning program (2.1.).

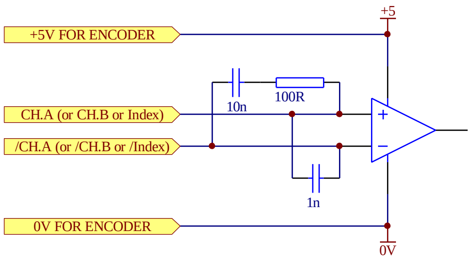
?? If the above fails, contact Rutex.

3. Connection of Encoders:

The differential encoder is much preferred over the single ended encoders. It provides much better noise immunity than the other signal types offer.

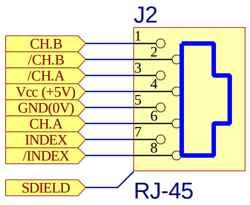
R20x0 drives accept signals from standard 5V two channel incremental encoders. Incremental encoders have 2 channel square wave outputs shifted 90deg. Both channels are identical – they can be swapped. Quite often encoders have one more output which is the Index pulse. The index pulse is a one pulse per revolution marker and it is usually used for homing or synchronizing more drives for one axis. The R20x0 boards can use the index pulse via SPI – the encoder reading is latched into Index Counter. Use of index is optional. Encoders can have *single ended* or *differential* outputs.

See the picture below, which illustrates the internal circuitry of the encoder and hall effect sensors inputs of the R20x0 drives and the encoder and hall sensors RJ45 connector pin-layouts on the R20x0 boards.



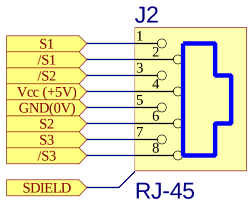
Simplified one channel input on the R20x0

Encoder Feedback



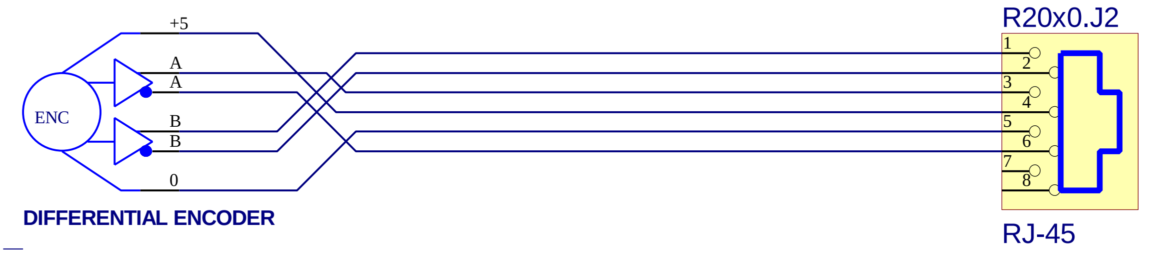
R2010/R2020/R2030/R2040

Hall Sonsors Feedback



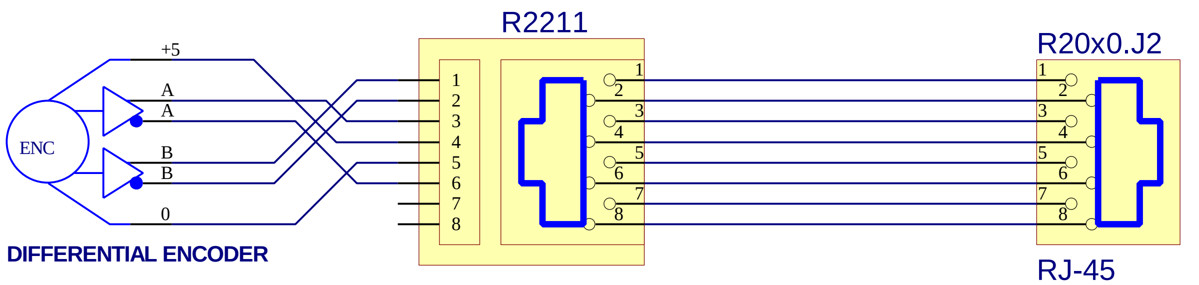
R2030 only

Differential output encoders have two complementary wires for each channel. If one is logic high (5V), then its compliment is logic low (0V). These encoders have 6 or 8 wires – depending on whether or not there is an Index pulse. There are two wires for the power supply and two wires for each channel. Differential encoders are suitable for long encoder cables and high noise environments and in general are more desirable than single ended encoders. See the picture below for connection.

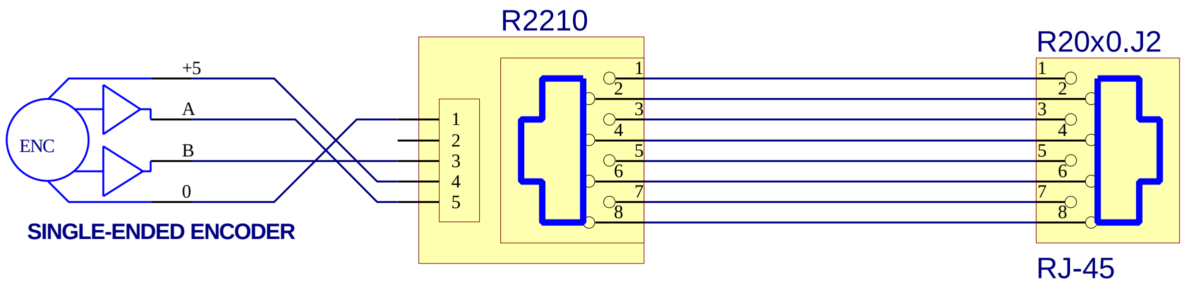


Encoder Interface boards: Rutex offers two interface boards for convenience of the users to simplify the encoder's connection. The R2210 is for single ended encoders. The R2211 is for differential encoders. Both of these boards can be used with standard CAT5 or CAT6 computer network cables. Note that the encoder connector on the R20x0 boards is shielded and support shielded or un-shielded network cables. The un-shielded cables are available from most computer stores. The shielded cables are not very common and you may research the web for supplier (for example: http://store.a2zcable.com/networking-shielded-network-cables.html). The shielded cable should be used for noisy environment for lengths 10-ft (3 meters) or more.

Connection of differential encoder using R2211 interface: The R2211 is a passive breakout board. It simply routs the connection from RJ45 connector to screw-in terminal block matching pin to pin on the connector. Pin one is a square pad on PCB and the identification of the signals is etched on the bottom side of the PCB.



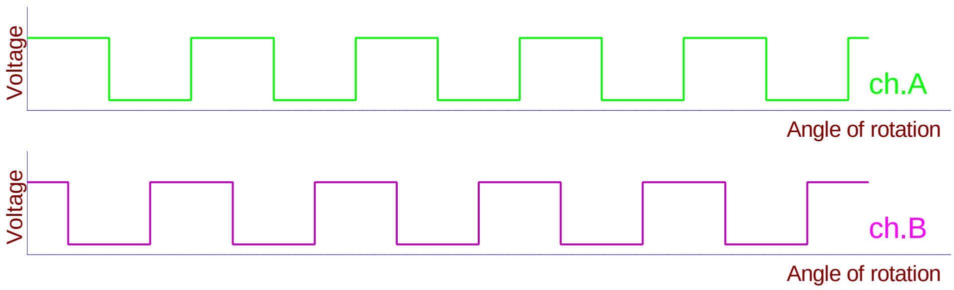
Connection of single ended encoder using R2210 interface: The **single ended output** encoders have only one wire for each output. They have 4 or 5 wires - two for supply, one for A, one for B and one for I (index where applicable). Single ended encoders can be of different types such as *CMOS*, *TTL* and *OC (open collector)*. The difference is in the amount of current they can sink (internally connect to ground) and source (internally connect to +5vdc) - encoder output current and voltage for logic low and logic high. The R2210 is an active board and it converts the single ended signal into differential signal. In can be used with CMOS, TTL or OC type of 5V single ended encoders. The high voltage single ended encoders could be used with voltage level translator – contact rutex for details.



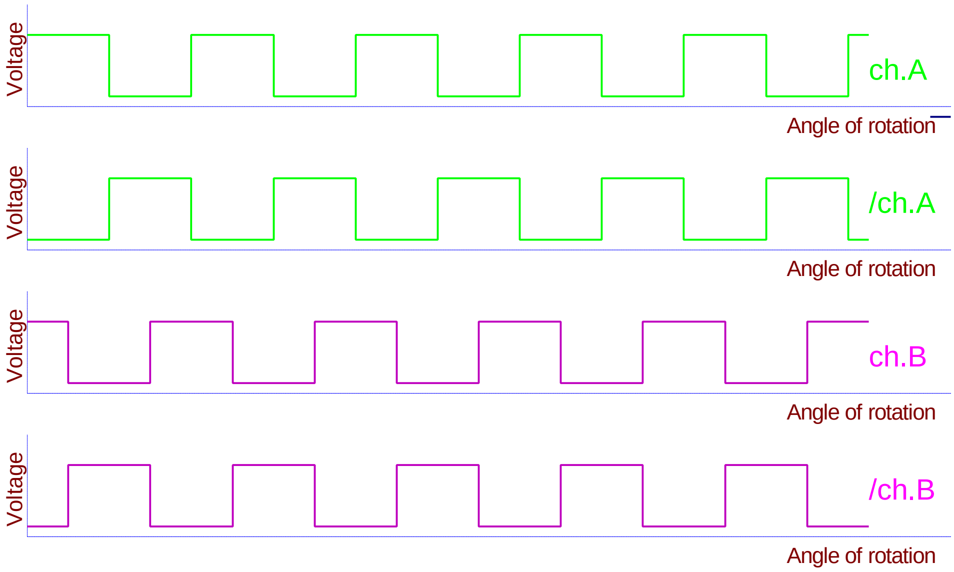
Using US-Digital's encoder interface EA-D8: US-digital offers an encoder interface with a RJ45 connector but unfortunately this interface is not compatible with standard CAT-5 network cables. The CAT-5 cable is using twisted pairs between wires number 1&2, **3&6, 4&5,** 7&8. The US-Digital interface is using twisted pairs between 1&2, **3&4, 5&6,** 7&8. A custom made cable must be used to match the EA-D8 interface with the R20x0 servo drives. As of June 1, 1008, US Digital has discontinued this part. They are no longer available.

Connect your encoder per one of the configurations above with a short cable. Do not connect the motor or the supply for the motor to the Vm connector as yet. Connect Vc power to the drive. Start the test program (R2xTune). Manually rotate the shaft of the encoder and observe that the position feedback from the encoder is changing proportionally to the displacement of the shaft. If it does change, then you can go to step number 4.

Fault finding hints: Encoder problems are the most common problems with the whole setup. You should check the signal directly on the CAT-5 encoder cable. The signal must look like the pattern below:

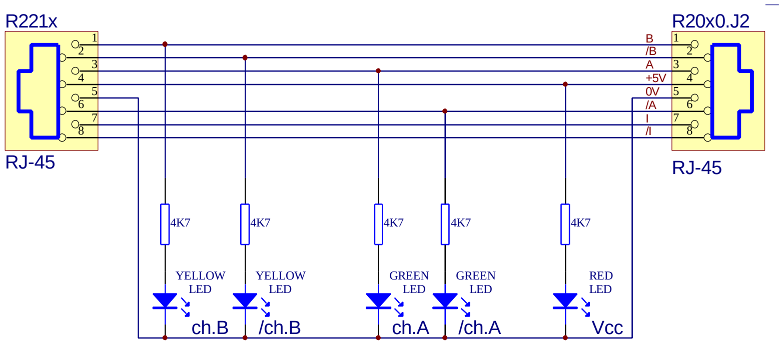


Output of single ended encoder.



Output of differential output encoder.

The best way to check the signals is to use 4-channel oscilloscope, although the 2-channel can be used as well. A single channel oscilloscope is not much help. If you have no access to oscilloscope you can make a little test jig that can be connected inline with the encoder.



Differential encoder test jig.

Note that the LEDs will not light up very brightly. The LED current is less than 1mA. Rotate the encoder very slowly. A typical 500-line encoder will change the status of the LEDs every 0.18deg. Measure the voltage for logic low and logic high. It should be more than 4V for a high and less than 1V for a logic low when the shaft is not being rotated.

4.1. Connection of the DC PM brush motor – R2010, R2020 only:

Do not connect the motor to the drive unless you are sure that the R20x0 board is communicating with the PC and the encoder is working properly. Initially, use a lower motor voltage and have the power-on switch close to your hand at all times. A one-horse power motor can do plenty of damage – *just imagine a full size horse out of control.*

Make sure that default values are loaded – click on “Restore Defaults” (Kp=100, Ki=0, Kd=0 etc). Connect the motor and supply for the Vm to R20x0 board. You can use DC24V for Vm even for a 200-volt motor. You will get enough motion for testing, but you will not get the full speed or torque.

The motor should stay steady after the power is switched on. Carefully try to displace the shaft by hand and the motor should “fight” you back proportionally to the displacement. If it runs away for second or two, then “fights “ you back with a huge dead band, then you have to swap the polarity of the motor or encoder. You can swap the wires or click on the Checkbox “Swap Motor” or “Swap Encoder”. Do not swap both the motor and encoder, just one of them. If the motor breaks into oscillation, lower the Kp value by adjusting the Kp and apply power again.

If you are doing a retrofit of a machine and you do not know the terminal resistance of the motor (the resistance of the motor’s armature), and the motor is over .3 hp for a R2010 or 2030 drive, or over 1 hp for a R2020 drive, for warrantee purposes, **you must install** a “braking” resistor in series with one of the wires connecting the motor to the drive. If you do not know the terminal resistance, you may measure it with a quality digital volt/ohm meter and estimate it by taking the reading of the ohmmeter and subtracting 1 ohm. Then add the necessary braking resistor. The R2010 drive must “see” a motor resistance of .03 ohms per volt being supplied to the motor (Vm). The R2020 drive must see a motor resistance of .015 ohms per volt. If the drives do not see these minimum amounts of resistance the drive will eventually fail, and may drive an axis out of control when it does. Read the **R2000FAQ.PDF** “Motor and Drive Matching” document downloadable from the Rutex.com web site listed under the R2000 series products to learn why this is necessary and how to calculate whether you need a braking resistor if you have the motor’s specifications. If the MOSFETs in a drive become shorted and blow while you are tuning it, you will need to send Rutex a copy of the motor’s specification with the blown drive and indicate the braking resistor used for the drive to be covered by the warrantee.

Now you can go on to step 6. – Tuning of the drive.

4.2. Connection of the DC PM brush-less motor to the R2030:

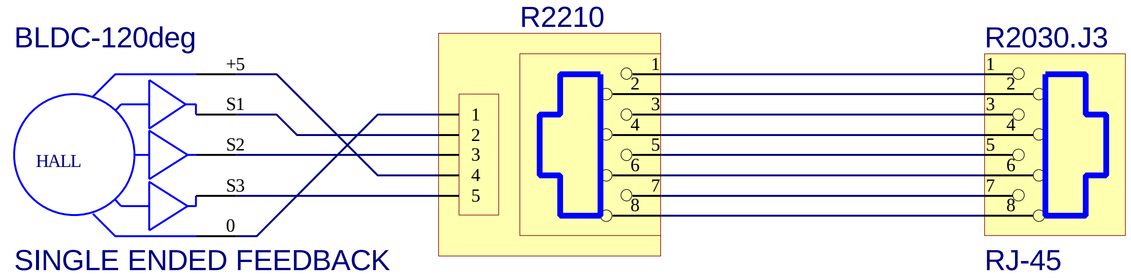
Determinate if the motor feedback is 60-deg or 120-deg. Most of the motors are 120-deg.
Table below shows the 6-step communication feedback S1-S2-S3

Rotation angle (electrical)		S1 60-deg	S2 60-deg	S3 60-deg		S1 120-deg	S2 120-deg	S3 120-deg
0-deg		0	0	0		0	0	1
60-deg		1	0	0		1	0	1
120-deg		1	1	0		1	0	0
180-deg		1	1	1		1	1	0
240-deg		0	1	1		0	1	0
300-deg		0	0	1		0	1	1

As you can see from above the 60-deg motor NEVER goes through 0-1-0 and 1-0-1 sequence. The 120-deg motor NEVER goes through 0-0-0 and 1-1-1 sequence. You can use this rule to determinate type of your motor. Use oscilloscope or LED test jig similar as per 3. Note that typical 4-pole motor has two 360-deg electrical revolutions per one 360-deg mechanical revolution.

If the feedback from your motor is 60-deg then connect a jumper, JP1, to link two pads on the R2040 board. The jumper is located in the middle of the R2030 board next to the crystal. Alternatively, if the commutation feedback is a differential output, then you simply swap the polarity of one feedback channel and then the motor will ‘become’ 120-deg. If your motor is 120-deg, then you do not have to change anything.

Connect hall sensors to the R2030 drive as per picture below:



Do not connect the incremental encoder at this stage.

Note that for the 120-deg motors the sequence S1, S2, S3 does not have to match. They can be freely swapped at this stage. You will match the phases of motor to match to commutation feedback. Although for the 60-deg motor the S2 feedback must be connected to S2 input. Only the S1 and S3 can be freely swapped.

Note that the commutation feedback input to R2030 is identical to encoder input. Follow the instruction in section 2 how to connect different types of feedback output.

Connect the Vc supply to the drive and start the test program (R2xTune). Do not connect the motor or Vm at this stage. Firstly, make sure that *all* default values are loaded – click on "Restore Defaults" (Kp=100, Ki=0, Kd=0 etc). Once the default values are loaded, then adjust the Kp=0, Kd=0 and Ki between 5 to 10.

Phasing of the motor:

Set the current limit in the R2030 drive to minimum (via the R2xTune) and use current limiting power supply (bench top laboratory supply). If you have no current limiting power supply then connect in series with +Vm a 10 ohms 50 Watts or more power resistor as well as Amperemeter. Use lower Vm – about 15-25 Volts is fine for test. Incorrectly phased motor can draw plenty of current and that can damage the motor or drive.

There are 6 possible combinations of motor phases and only one is correct. For example, if the three phase wires from the motor are Red, White, and Black then the possible combinations are:

#1	Red	White	Black
#2	Red	Black	White
#3	White	Black	Red
#4	White	Red	Black
#5	Black	Red	White
#6	Black	White	Red

The idea is to try them all to find "best" one. You can find one or two, which might seem to be correct, but when you find the right one you'll notice the difference.

Make sure that encoder is not connected, Kp = 0, Kd = 0 and Ki = 5 to 10. PC does not have to be connected at this stage. Try the #1 combination and then connect at the **same time** the Vm and Vc to the drive. Within few seconds (between 1 to 10) the motor should start moving, slowly accelerating and smoothly increasing the torque and current. Check the torque with your hand and make sure that there is no ripple in torque or current. Incorrectly phased motor can stall in certain position or have very high torque and current ripple during motion. Try all 6 combinations to make sure that you found the right one.

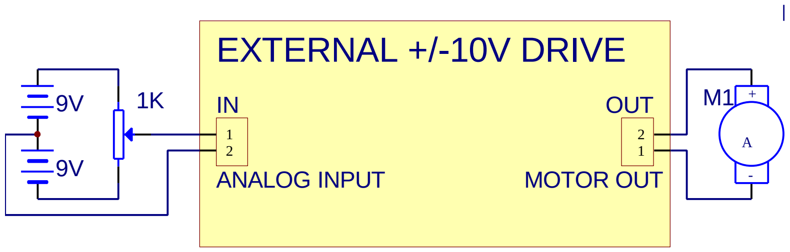
The encoder paragraph 2 explains why the Ki cause the motion of the motor. Alternatively you could use a small jog from R2xTune (1 - 5 steps) to simulate the error for Ki accumulator. Once the commutation feedback and motor are correctly phased, then you can follow the procedure for the brush motors:

4.1. Connection of the DC PM brush motor – R2010/R2020 only: to re-connect the encoder and verify polarity of encoder.

4.2. Connection of external +/-10V servo drive to R2040:

Do not connect the external 3rd party servo drive amplifier to the R2040 unless you are sure that R2040 board is communicating with the PC and the encoder is working properly.

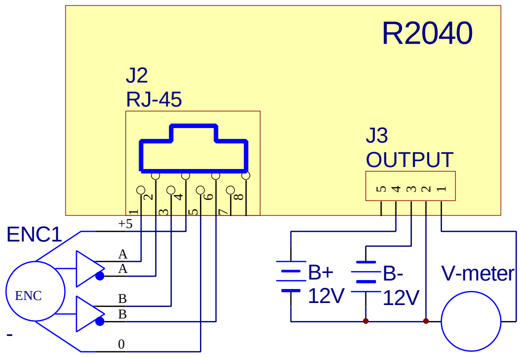
Tune the external servo drive as per the manufacturer's specification. You may want to test the drive before connection to the R2040 using, for example, two 9V batteries and 1 kilo-ohms potentiometer. See the picture below. Some analog servo drives must be set up in the current or torque mode for successful tuning with an R2040. Check the analog drive specification to see how to set it up in the current or torque mode. It may require the installation of slightly different jumper settings, potentiometer settings or resistors inside the analog drive.



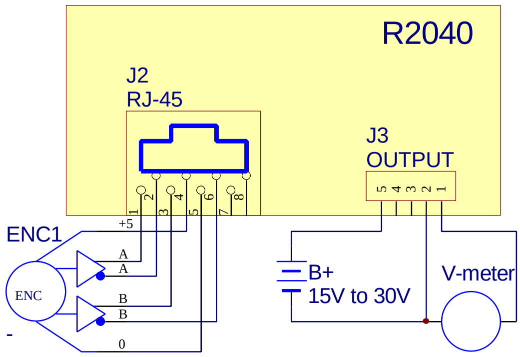
Testing 3rd party analogue servo drive.

Make sure that the default values are loaded in the R2040 drive: Kp is set between 100 to 200, Ki=0 and Kd=0, Step=1 (multiplier).

Connect the Multi-Meter to the output of R2040 board. And either single 15-30V DC supply or regulated complementary +/-10 to 15 Volts (+/-12V nominal).



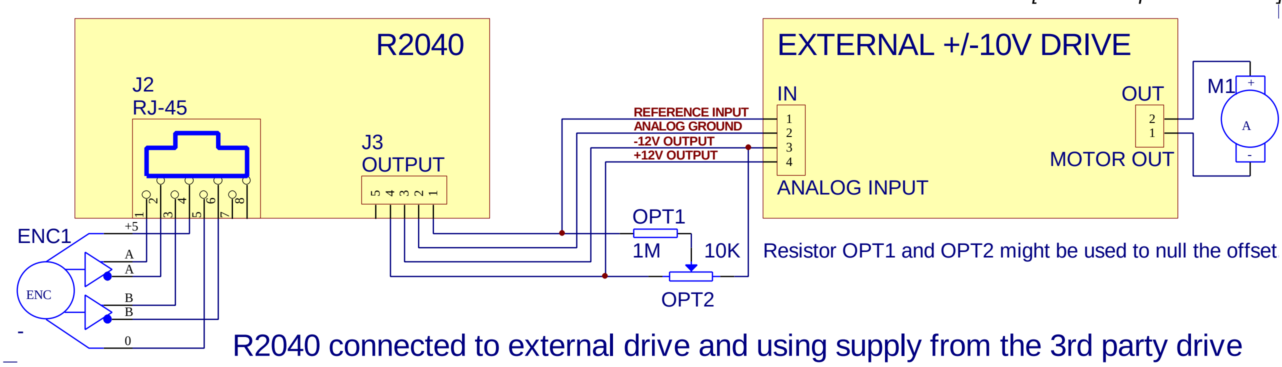
R2040 using dual supply for output



R2040 using single supply for output

Connect the Vc power to R2040 servo board and monitor the voltage on your Multi-Meter. It should be close to zero (typical offset is less than 50mV). Then rotate the shaft of the encoder by hand and the output voltage should be proportional to the angle of displacement reaching close to or just above +/-10V.

Having tested the external servo drive and the R2040 then we can connect the drives together.



R2040 connected to external drive and using supply from the 3rd party drive

The motor should stay steady after the power is switched on. Try to displace the shaft by hand and motor should "fight" you back proportionally to the displacement. If it runs away for second or two, then "fights " you back with huge dead band, then you have to swap the polarity of the encoder:

If using a single ended encoder then swap the ch. A with ch. B

If you are using a differential output encoder then swap ch. A with ch. A/ OR ch. B with ch. B/. Do not swap both.

The supply for the opto-isolated output for R2040 usually can be 'tapped' from external analog servo drive. It can be either the complementary dual +/-10V to +/-15V (+/-12V nominal) or single ended unregulated +15V to +30V DC. Do not connect the Vc supply of the drive (supplying digital circuitry and encoder) together with the Vout supply.

5. Tuning software:

The tuning software R2xTune is used to setup and tune R2000 series of servo drives. There are several configuration flags (check-boxes) as well as PID variables to be adjusted with this software. Additionally, there are some flags, which are not user adjustable such as under and over voltage detection.

Check-boxes:

- *Swap Motor:* Swapping polarity of the motor. Same as manually swapping motor wires
- *Swap Encoder:* Swapping polarity of the encoder. Same as manually swapping the wires for channel A and B.
- *Swap Index:* Swapping polarity of the encoder's index pulse.
- *Encoder 4Q/1Q:* If checked then the encoder is decoded in 4-quadrature fashion 1 optical line = 4 encoder pulses. If un-checked then 1 optical line = 1 encoder pulse.
- *Enable Encoder:* Encoder is enabled when this flag is checked (default). Encoder might need to be disabled during phasing of the hall feedback of the brush-less motor.
- *Enable Soft Resume:* If checked, then the drive resumes slightly "softer" from error or limit switch condition. The current slowly increases to full value and the PID accumulators are reset.
- *Enable Following Error Trip:* If checked, then the drive switches off the motor if the following error exceeds the pre-set trip value.
- *Enable Encoder Error:* If checked, then the encoder error detection is enabled. The drive turns off the motor output if there is no encoder activity for a period of ¼ of a second, when the drive expects to see the encoder activity. The encoder error detection can detect an encoder fault. This prevents motor run-away and stalled rotor conditions.

- *Enable Limit s/w – motor hold:* If checked, then the motor abruptly stops when limit switch is closed. This flag should be used only on small machines where abrupt stop will not cause mechanical damage.

- *Enable Limit s/w – motor off:* Preferred method of using limit switches. If the limit switch is closed then the motor output is switched off. No braking force is applied, so there needs to be room for the axis to coast to a stop, or a spring mechanism on the end of the travel to stop the motion before the axis runs into the axis physical end of travel.

- *Enable Error signal -> Err/Out output:* This flag must **NOT** be set during SPI tuning. Setting this flag will corrupt multi-drive SPI communication. If this flag is set (checked) then, in case of an error (5.2.7, 5.2.8 etc.), the ERR/Dout pin (open collector) is pulled down. ERR/Dout pin can be used to drive relay to trip E-stop circuitry. The ERR/Dout signals from all drives are tied together, and any one of them can trip the E-stop. This flag can be set once the drives are tuned and ready to be used with Step&Dir software.

Pull down menus:

Mode:

- *Step&Dir with SPI:* Default. It should be used for standard Step&Direction mode as well as during SPI tune.

- *SPI only – Master:* Step&Dir mode is disabled. Only one Master can be used in a system. In this mode the Stepper/SPI line is used as a servo loop time synch *output* to synchronize servo loop time of the other servo drives.

- *SPI only – Slave:* Step&Dir mode is disabled. There must be one Master in a system. In this mode the Stepper/SPI line is used as a servo loop time synch *input* to synchronize the servo loop time with master servo drive. The value set in servo loop time (5.3.2.) of the slave is disabled.

Servo Loop Time*:* The servo loop time can be set from 125 micro-seconds up to 13.1072 milli-seconds (200ns * 2^16). The servo loop time could be as low as 60 us if synchronized with an external clock (SPI only – slave mode). It is quite often incorrectly assumed that the fastest loop time is the best. The servo loop time should be proportional to time constant of the motor. There is not much point for the servo drive to 'recalculate' everything if the motor mechanically did not respond yet to previous calculation. Secondly, with very short servo loop time compared to motor's time constant, the Kd term will loose it's meaning and it will only act as a Kp term. Typically, the 125us or 250us servo loop time should be set for small servomotors – usually NEMA23 or smaller. The large servo motors are bit slower and the loop time should be set to 500 us or 1 ms. The optimal loop time for non-servo motors (such as typical DC motors for electric wheelchair or treadmill) used for servo application could be in the milli-seconds range.

SPI interpolation: It can be set from 1 point up to 128 points (2^n). SPI interpolation is used only in SPI mode. In Step&Dir mode this value is ignored. During SPI motion, the value written in the Motion Command (a Velocity Word) is divided by SPI interpolation factor making the Effective Loop Time = SPI interpolation * Servo Loop Time. During tuning/setup, the SPI interpolation should be set so that the SPI buffer is large enough to be able to run smoothly. The SPI buffer should be set at 16 ms minimum, but preferably at 64 or 128 ms (Effective Loop Time set to 4 or 8 ms).

Variables:

Kp: Proportional gain of PID filter. Proportional gain is directly proportional to the torque. It is the main force in the whole PID loop. The Kp provides a multiplier for the position error between the commanded position (from the CNC control) and the counter, which tracks the feedback from the encoder.

Ki: Integral gain of PID filter. It is like a flywheel in a toy car. You have to push it harder (more Kp) to move it, but it runs smoother than a toy car without a flywheel. The I part of the PID filter integrates the following error and this integral is directly added to the P term and keeps the following error small during the motion or in steady condition. In fact, correctly tuned Ki can keep the following error within +/- 1 encoder count during constant velocity motion (or in steady condition) while the motor can deliver full torque. The other great advantage of Ki is that it can make from a simple trapezoid velocity profile a profile very similar to a typical "S" profile. A trapezoid velocity profile is when the motor acceleration is constant (an increment of velocity) until it reaches the desired velocity. In S profile, the acceleration at the start is rather low, and then it exponentially goes up till it reaches the maximum possible acceleration. Then, just before reaching the desired velocity the acceleration gradually decreases. You can see the typical S velocity profile on the faces of the astronauts in the space rocket while taking off.

Kd: Differential gain of PID filter. It is like oil in a shock absorber. It makes the servo settle in the desired velocity.

Kd Index: part of the differential gain. Differential part is determinated from $Error_{N} - Error_{(N\text{-}KdIndex)}$. This variable tells the PID loop how far back in the recent servo loop cycles to consider when computing the amount of damping to use. The higher the number (say for example 3) that is specified, the further in the past it looks, thus the damping may be reduced. The lower the number entered, the fewer previous cycles it considers, so that the damping may in effect be higher.

Current Limit: A positive whole number can be set here, where the final limiting point is multiplied by 0.2Amps. For example, a value set to 20 represent current limiting point of 4 Amps (20 * 0.2 = 4.0).

Current Fallback: If the drive limits the current for about 2 seconds to the value set in (5.4.5) then the current limiting point is set by the fallback value. For example, the Current Limit is set to 4 Amps (20); the Current Fallback is set to 1 Amp (5). In over-current condition the drive limits the motor current to 4 Amps for 2 seconds and then the current limit is dropped down to 1 Ampere. The motor has to draw less then half of the fallback value current for 2 seconds to reset the Fallback condition. In general the Current Fallback feature greatly protects the motor/drive from overheating. If the Fallback feature is not desired, then the Current Limit and Current Fallback value should be set to be same.

Step Multiplier: It can be set from 0 to 255. The step multiplier is used only in Step&Dir mode. During SPI motion the value is ignored. Preferred value is 1 and value set to 0 disables the motion in Step&Dir mode. For example, if the step multiplier is set to 10 then each one step on the Step&Dir input represents the motion of 10 steps from the encoder (2.5 optical lines), although the 10x step multiplier decreases the resolution of the system by 10. You should match the step multiplier with the maximum required speed of your system. For example, running software with 45kHz sampling frequency can generate a maximum pulse train of 45,000 samples per second, or 2,700,000 steps per minute. Using a 500 optical line (2,000 quadrature pulses) encoder and a step multiplier set to 1, the maximum available speed is 1350 RPM (2700000/2000). If the step multiplier is set to 10 the maximum speed is 13,500RPM.

Following Error Trip: If the Following Error – a difference between commanded position and encoder feedback exceed the preset value and the 5.2.7. flag is set, then the motor output is switched off. The following error trip value can be set from 2 up to 30,000.

6. Tuning of drives:

Overview: The PID tuning is bit of art. You may spend days to find 'the best' tune-up and never be fully satisfied and then again, most likely, you will be able to tune drive within few minutes to satisfactory results. The PID algorithm has several variables and you might get very similar results with different combinations of these variables.

The tuning is very much motor/machine/encoder dependant. There is no universal procedure. Before you start to tune a drive you should mechanically disconnect motor from the machine and you should have the E-stop switch very close to your hand. An incorrectly tuned drive can violently oscillate which can do mechanical damage to the machine/motor or overheat the drive. It is also a good idea to start the tuning at a lower voltage (if you can drop down the voltage) to familiarize yourself with tuning. There is much less mechanical/electrical energy.

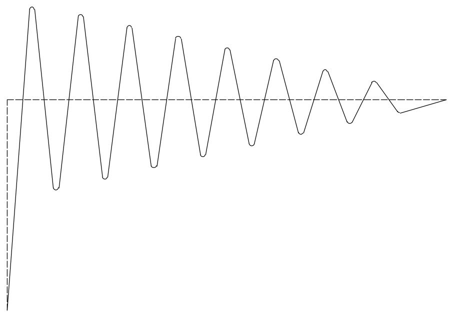
The goal in the adjustment of the drive is to find the best combination of variables such as the Servo Loop Time, Kp, Kd, Kd Index and Ki. After changing the value of a variable the Step Response and Test Profile should be run to view the progress of adjustment.

Results: (Or, how the properly tuned drive should look like.)

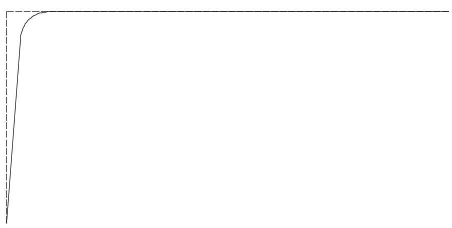
Unfortunately optimizing one parameter compromises other. It is like tuning a car engine for best power and best economy at the same time. It cannot be done. The engine tuning has to be compromised or optimized for one of them only. The best way to learn how to tune a drive is to actually do it. Most people learn to use a combination of factors to guide them in tuning, such as sound and feel as well as the step response and short trapezoidal moves that can be made using the tuning software.

Set the Servo Loop Time: The Servo Loop Time is the first parameter that should be set. For fast small motors start with 250us. For large servomotors use 1 ms. See section #5 for details. Make sure that the Ki and Kd are 0 and Kp is reasonably low (for example, 100). The current limit should be set reasonably high for example 5 times the expected continuous current.

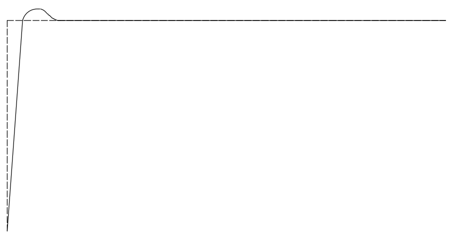
Adjust Kp: Slowly increase the Kp and run the Step Response. You should get the graph that eventually stabilizes itself. You might need to increase the number of samples to, say, 5000, to see the whole graph – see below. Do not allow the motor to oscillate for more than a couple of seconds or damage to the drive could occur.



Adjust the Kd and Kd Index: Increase the Kd with the optimal Kd Index value. Whenever you change the Ki Index, then you might need to start with Kd again from zero. Try to adjust the optimal damping as shown below.



Adjust the Ki: Adding Ki to the system will result again in overshoot. A very low Ki (usually lower then 50) does not do a very impressive job. The Ki usually must be well over 50 for the system to work properly. With Ki in the system, the Kp and Kd might need to be increased as well. The Step Response graph for correctly tuned drive should look like below.



Step Response tuning is a commonly used method of tuning a servo system. In tuning a Rutex drive, the tuning software induces a very large following error through the spi communication with the servo drive and then the user monitors the settling time of the correction on the monitoring screen. The aim is to adjust the drive for shortest possible settling time and acceptable error, but keep in mind that some overshoot is normal in this tuning method. The size of the Step should be large enough to let the motor reach its full speed. Usually the step has to be between ¼ to 1 turn of the motor shaft. This tuning method does not employ a trapezoid profile with the programmed acceleration/deceleration, and this is why the motor might overshoot. If the step (or induced following error) were very low or provided correctly programmed acceleration / deceleration, then the motor should not overshoot. A properly tuned servo, matched with a control that employs acceleration and deceleration will not overshoot, but the tuning software allows the user to tune the servo system under somewhat extreme conditions.

(End of the document)