



AxeMotion

CNC Products
Real Time Systems

CPM Booster Card

Retrofit card for Isel IMC4 controllers.

User's Guide

CPM Booster Card

User's Guide

Version 1.0

January 2010

Retrofit card for Isel IMC4 controllers.

Hardware:	Copyright © 2008-2010	AxeMotion
Software:	Copyright © 2008-2010	Bertrand Lenoir-Welter
Manual:	Copyright © 2008-2010	Delft Spline Systems

All rights reserved.

CONTENTS

1	THE BOOSTER CARD	7
2	HARDWARE SETUP	8
3	SOFTWARE SETUP	10
4	OPERATION	12
	Appendix A - CNC control theory	13
	Appendix B - Card settings	16
	Appendix C - Card specifications	19

CHAPTER 1: THE BOOSTER CARD

The CPM machine is one of the most successful machines made by German manufacturer Isel. The machine features a closed cabinet built of metal, which is a stable frame for the machine and guarantees a safe operation as milling is possible only with the door closed. Ballscrews and bearings are well-protected from chips and dust. The CPM series are based on the IMC4 stepper motor controller: a proven design with a track record that shows a great reliability.

Drawback of this IMC4 controller is that it is not a very fast controller: between every two movements the cutter will briefly stop. So at the end of a movement the machine will decelerate, stop for a moment and then again accelerate for the next movement. As a result the machine moves rather jerky when executing series of G1 movements (linear interpolations). The machine makes a very typical “staccato” rumbling noise that will make the complete table underneath vibrate with the machine.

This jerky movement style comes with serious disadvantages:

- It takes a lot of time. Especially when the movements are short: then the machine does not even come close to the nominal feedrate as it is decelerating and accelerating all the time.
- The resulting surface quality is not very good
- Machining metals is not possible.

Second drawback of the IMC4 is that communication with the PC a serial cable is used (RS232). This is a rather obsolete method: most current PCs no longer have a serial port present. For communication with the PC current devices use USB.

Stepper motor control has evolved a lot since the IMC4 controller was designed: fluid movements and a machine that keeps up speed are very well possible nowadays.

Axemotion is a specialist in controlling stepper motor based machines. Based on this knowhow, Axemotion has developed an plug-in card for the IMC-4 controller that changes it to an up-to-date stepper motor controller. This card is called the **CPM Booster card**, and you new are reading the manual for this card.

The CPM Booster card features:

- USB Interface
- up to 4 axes controlled simultaneously
- look-ahead buffer (250 vectors).
- supports all IMC4 functionality

The Booster card will not function with the Isel Remote or RemoteWin software: it needs to use Kay or Galaad as control software. This is because the software must be able to do the inverse kinematics: analyze the movements and determine the optimum speeds at each moment. Also see Appendix A of this manual.

Installing this card will save a lot of time for NC files with loads of very small G1 movements. Reducing time with 50% is easily possible, in some cases the time gain will even be higher !

An extreme example was a 8 Mb NCP file for a small object (jewelry, ca 40x50x10 mm). Very accurate toolpaths with G1 movements only: small distance between the toolpaths and small stepsize. A conical cutter with a 0.1 mm tip was used. Here the difference between the original IMC4 and the Axemotion booster was enormous:

- IMC4 242 minutes (so 4 hours 2 min)
- AxeMotion 40 minutes

So a milling time reduction of over 80 % !!!

Check the videos on www.spline.nl to see the different movement style. You can see the same NC file (a bottle model) being machined with the original IMC4 (51 minutes) and with a Boosted version (26 min). Note that seeing and also hearing the difference on the video will tell you more than only these numbers.

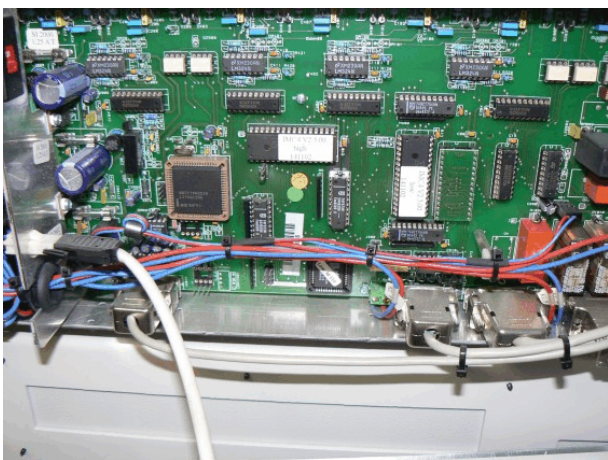
For NC files with long movements (say G1 movements of 20 or more mm long no time gain will be present: then also with the original IMC4 the machine will be on speed most of the time. And the maximum speed (50 mm sec for CPM machines with a 10 mm ballscrew pitch) is caused by the maximum rotation speed of the stepper motors and cannot be improved.

CHAPTER 2: HARDWARE SETUP

How to insert the IMC4 Booster card into the IMC4 controller.

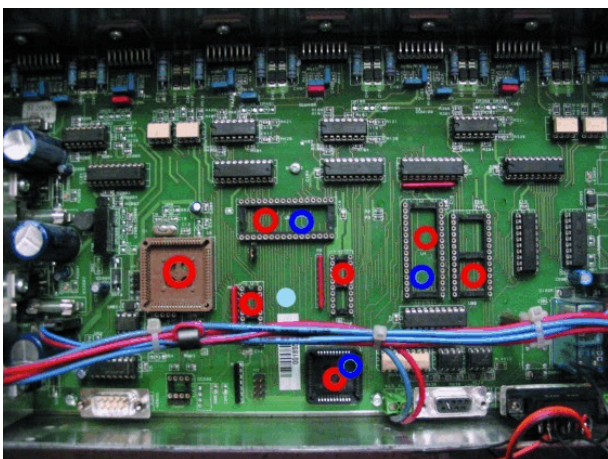
The IMC4 controller is located on the back side of your Isel CPM or ICP machine. You can see it after removing the back cover panel of the machine (six hexagon screws). Note that this panel is connected to the machine by an earth wire.

In order to insert the AxeMotion Booster card you also need to remove the back panel of the controller (three Phillips screws and some double sided tape). Carefully remove this panel in order not to damage the cable that connects the ventilator in the panel to the controller (you can disconnect the plug to completely loosen the panel).



Now you can see the PCB board of the IMC4 controller, as shown in the picture above.

From this controller you need to remove 7 chips: 2 square PLCs and 5 rectangular chips. Note that for removing the PLCs you need a tool called “PLCC extractor”. For the chips using a chip extractor is advised as well: when you correctly remove them (without bending the pins) you will be able to replace them later if needed. Make notes which chip was placed in which connector, and how it was orientated.



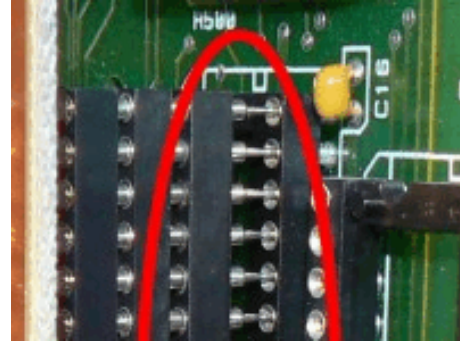
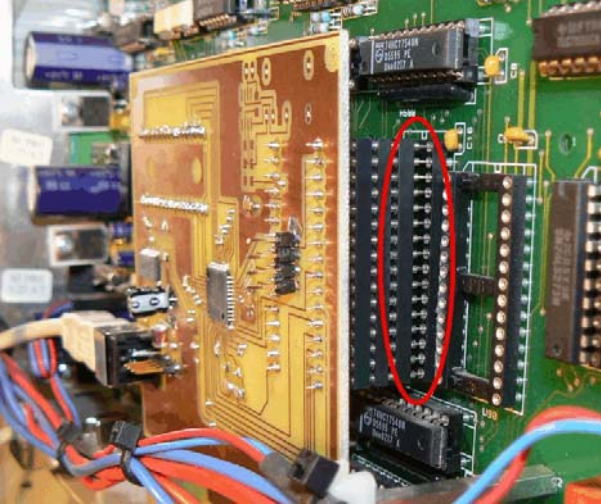
The picture above shows the 7 chips removed, indicated by 7 red circles.

You can also see 3 blue circles: the IMC4 Booster card needs to be pressed into these three chip connectors. To make this possible the booster card comes with some extra connectors to bridge the distance between the two boards.

Placing the board will prove to be the most difficult part of the complete installation procedure: normally when

pressing one chip down it is possible to gently rock the chip while pressing in order to make it slide home. As here three connectors are present this is not possible. And in addition it is not possible to see what you are doing as the Booster card completely blocks your view.

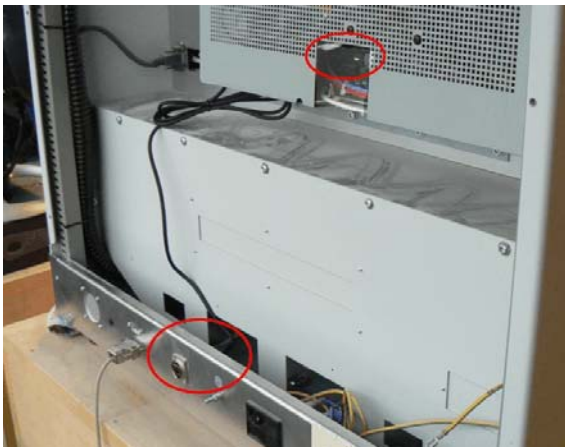
Check carefully if the pins of the two 32 pins (2x16) connectors are all well placed on the two 32 pins sockets of the IMC4. If that is OK you can firmly press on the "middle" of each these two connectors. All pins must be completely inserted into the socket. After that, the PLC connector will slide into position when pressing that part of the booster board.



As a result all pins should have been **completely** inserted, and the Booster card needs to be exactly parallel to the IMC4 main board. The booster board also may not be bended.

In case this plugging in has not been done completely correct you will get unpredictable behavior. In the picture above you can see a card that has **NOT** yet been fully plugged in (note that your booster card may have a different color).

Finally you can plug a standard USB cable into the IMC4 Booster board and lead that to the outside of the machine. Or you can mount a USB connector in the back wall of your machine, next to the other connectors. Connect the booster card with the inside of that connector using a short USB cable: you now can close the machine and use a second USB cable to connect PC to machine. See the two pictures below.



CHAPTER 3: SOFTWARE SETUP

2.1 How to install the Drivers for the Booster Card.

Do NOT use a Win95, Win98 or Win ME based PC for this card: the USB system on these PCs is too old and you will run into problems when using the Booster card. Use Win 2000, XP, Vista or newer.

The IMC4 Booster Card makes your milling machine a USB device, so you will need USB drivers in order to use it. Unfortunately these drivers do not yet come with a Setup program: you need to manually install them.

You can download the driver files from <http://www.axemotion.com> (Download menu)

There you can find a file called "AxMo 2-n.nn.nn.zip" where n.nn.nn is the version number. Must be 2.02.04 or newer. Unpack this file and remember into which directory.

Next, when you connect the USB cable from the Booster board to your PC, Windows will start the New Hardware found wizard. Make this wizard look in the directory just mentioned and use the driver it will find there. It might be that this needs to be done twice as the driver comes in two parts.

2.2 How to configure Kay to work with the Booster Card.

Your Isel machine has been delivered with a Kay CD. At least when you have bought the machine from Delft Spline Systems. In case you used the Remote or RemoteWin control software you also need to buy a Kay license to be able to use the booster card. As a first step you will need to install Kay from this CD: later updates will only work when a valid Kay license is present already.

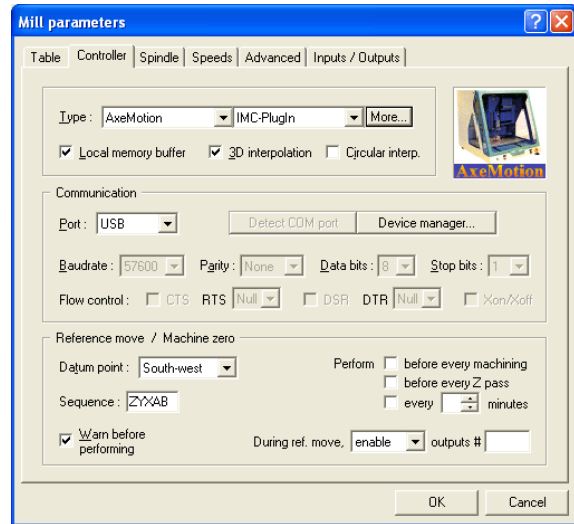
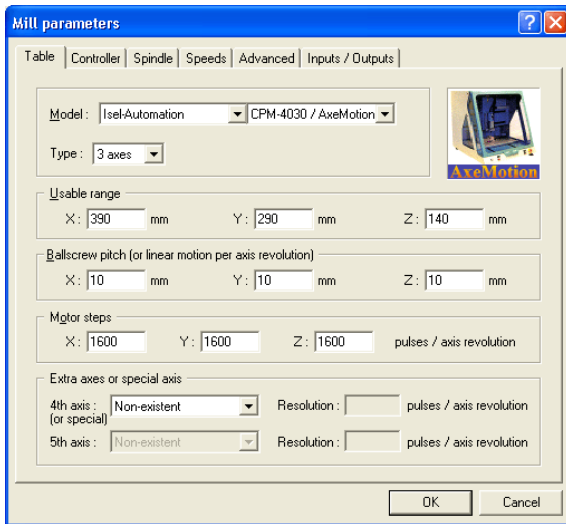
Next you need to download and install such update, as the version on your CD may not yet support the booster card. For downloading please visit www.galaad.net Go to the Download page and download an "Update for main modules" in your preferred language. Unpack this ZIP file into the directory of your original Kay installation, which by default will be C:\Program Files\Galaad\
Just over-write the old files with the new versions.

Now you are ready to start.

In contrast to how the machine behaved earlier, you cannot yet press the green Power button on the front of your machine to power up the IMC4 controller: Kay will now tell you when this can be done.

Start Kay, cancel the file load (or load a small NC file) and open the Machine Parameters (Parameters menu, Machine ...). It might be that these will be opened automatically when Kay cannot find the currently selected machine (Note that the dialog is called "Mill parameters").

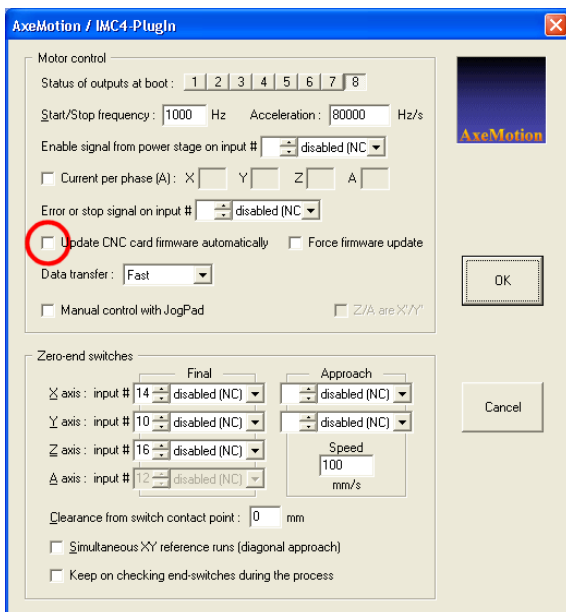
It may be that in a moment you will see a message that Kay is automatically updating the firmware of the CNC Card. That is OK: as the software changes more often than the hardware this is way to keep them both up-to-date.



The first tab of this dialog is called Table and is shown above left.

Here select your machine: make sure to choose the type with “/ AxeMotion” added to the original name. You also need to check the values for the ballscrew pitch: most CPM machines have 10 mm per revolution, it might be that Kay incorrectly sets these to 4. If so then you need to change 4 to 10 for all three axes. Or of course the other way round when you have a machine with 4 mm screws. Motor Steps needs to be 1600 for all three axes (detailed information: the Isel stepper motors have 200 steps, however are operated using a 1/8 step microstep resolution which results in 1600 steps). In case you are using a fourth axis, the Resolution needs to be 26284 (type 1:16) or 78792 (type 1:50),

The second tab page is called Controller, as shown above right, and the settings should be correct automatically: Controller type is AxeMotion IMC-Plugin and Port is USB. Now press the button “More” just right of the controller type IMC-Plugin.



The AxeMotion sub-dialog shown above will pop up.

Important here is to check the option “Update CNC card firmware automatically”. This is a great option, as it allows you to renew the firmware on your booster card. Downloading new software is easier than ordering a new card, so this is an easy way to keep your card updated. It is important to check this option as your card may have been in stock for some time and thus may have old firmware.

All other settings in this dialog should be automatically correct, at least for machine with 10 pitch ballscrews. If you have a machine with **4 or 5 mm pitch** please check Appendix B of this manual for more detailed information. In case of 5 mm pitch use the advised values for 4 mm. Important: for such machine you also need to change the maximum speed of the machine in Kay (Speeds tab): needs to be 20 mm/sec. The default value of 50 mm/sec is for 10 mm screws and will cause problems.

CHAPTER 4: OPERATION

Using the machine:

When you have completed all steps of the Setup as described in the previous chapters, you are ready to use your optimized machine. You can use Kay just as before (see the Kay manual).

The only difference when starting is that you now need to first start Kay and only when Kay asks you press the green Power button of your CPM machine.

When using Kay, two more difference can be observed now the booster card has been installed, and these two differences are in fact disadvantages.

Previously the movement information was sent to the machine line by line: so only after completing a movement the next command was sent over (while the machine briefly halted). Because of this Kay always knew the current position of the cutter, and could update the drawing to exactly follow the real cutter.

Now the movement information is sent to the booster card in large chunks, which is needed in order to keep up speed. So Kay no longer knows the position of the cutter at any given moment, so the drawing no longer is synchronized with the cutter movement. The booster card also allows sending data while the machine is milling. Previously the IMC4 could either communicate or machine.

The second change is while interrupting the program: Previously it was possible to press the red stop button on the Kay screen at any moment, briefly stop (say to remove the chips) and the continue on the same location. As Kay no longer knows the exact position this is no longer possible.

Stopping at any moment still works (using the red stop button), however when resuming the cutter will restart at the last rapid movement. This might even be the begin of the NC file. This is needed as kinematics calculations for all moves are now related: the instant speed when stopping was the result of all previous vectors with their own calculated speeds. So, the only possibility is resume at the last tool-up position, so to make a fresh start after the last Rapid move.

This is a disadvantage indeed, we hope that it will be possible to find a way to improve this situation.

The Pause button on the Kay screen will not properly function either: the movement will indeed be paused, however Kay will wait until the next Rapid movement before Pausing, which may take very long.

Still we feel that the advantages clearly outweigh these two problems, and as you now know about them you can learn how to deal with them.

Please note that kinematics calculation in Kay does not consider the **rotary axis**. So the toolpaths should be in XYZ, with increments of A in-between (in DeskProto this means using strategy “parallel to X-axis”).

If you use A in any other way, so let the A-axis and one or more other axes move at the same time, then the kinematics for these axes is no longer integrated: only done for X, Y and Z, not for A. So when A moves faster than the other axes you may lose steps. Not necessarily though: several users are known to do so without problems: it will depend on the toolpath characteristics.

Once you start machining you will immediately hear the difference: instead of the low rumbling noise that made both machine and table vibrate, while machining you will now hear music !

Machining time will be much lower for accurate NC files, that is for files with many small movements.

APPENDIX A: CNC control theory

Stepper motors are controlled by sending pulses to the motor. A pulse will cause the motor to rotate one step (or in fact two pulses: one for the step and one for the rotation direction). The number of pulses determines the total angle, the number of pulses per second determines the rotation speed. This rotation then is converted to a linear motion (for X, Y or Z), which in most cases is done by a screw. That is in fact a large bolt / nut combination: when the bolt is turned the nut will be translated. The screws can either have a trapezium shape or use balls for less friction. The CNC controller determines how many pulses are sent to each motor, and so the distance and speed of each axis. For almost any movement this will be different for X, Y and Z.

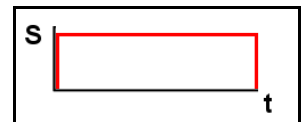
With Stepper motors it is possible to lose steps: when pulses are sent to the motor but for some reason the rotation cannot be executed (the axis may be blocked), then the step is lost. As there is no feedback from motor to controller the controller has no way of knowing that this has happened. It just assumes that the movement has been properly done while in fact it has not: the step is lost. So all subsequent movements will be done on an incorrect position (the machine does not "know" where it is, it only adds up movement commands to calculate the current position).

So far this is all standard, and it will apply to any CNC controller for stepper motors. The difference is about how high speeds can be achieved. Any toolpath will start at speed 0, and for any milling machine it is impossible to instantaneously change from 0 to a high speed. Not only for milling machines: also for say a car it is impossible to speed up from 0 to 100 in zero seconds.

This is where the CNC controller will make a big difference.

Very old CNC controllers can perform one single type of movement: the flat vector, where the speed remains unchanged from the start point to the end point. Of course this can work only with motors which have enough torque and when the speed is very slow (in theory this cannot work at all, so we may assume that the acceleration time in fact is a bit above 0).

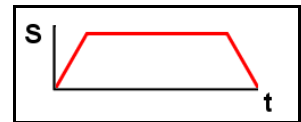
See the picture on the right, where the speed S is drawn against the time t .



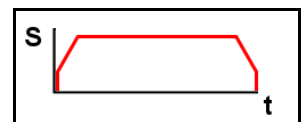
Old CNC controllers can perform accelerations, what we call "vectors with ramps". The target speed can be high because the motor accelerates before reaching that target speed, and later decelerates so it reaches the end point at slow speed. So the speed scheme is trapezoidal.

However: when the vector is short, then the acceleration and deceleration ramps that correspond to the target speed are longer than the vector itself. Then the motors are beginning to decelerate at midpoint, before having reached the target speed. Consequently the average speed is lower than the target speed. In a high resolution curve (the vectors are very small) the average speed thus is much slower than expected. This is the major cause of the machining time being much higher than expected.

Isel IMC4 controllers can perform only vectors with ramps and nothing else: so in-between each two movements the cutter will briefly halt.

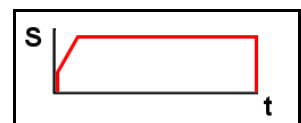


For most controllers, when making an acceleration ramp the vector does not start moving from null speed: the acceleration slope starts from a given speed which corresponds to the **Start/Stop frequency** (F_{ss}). This is a very important parameter. F_{ss} is given in Hz instead of speed units because it corresponds to a motor, not a linear motion. Of course the deceleration ramp leads to that same F_{ss} speed. So F_{ss} is the gap between immobility and rotation. If it is too low, the acceleration slope is longer than needed, which is not dangerous. On the other hand, if it is too high, the motors may be unable to catch the acceleration slope and stall: they remain still and all steps are lost.



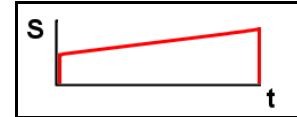
Recent CNC controllers can perform both vectors with ramps and vectors without ramps. Of course we cannot send a vector without ramps for a high speed (greater than F_{ss}) when the motors are still. It is the responsibility of the driver software to send a vector without ramps **only** if the motor is currently turning at a speed which is not too far from the speed of the new vector. I should talk about "speeds"

because each motor is independent and does not know what its colleagues do. So for the new vector the speed of each motor must almost match the speed of that motor for the previous vector, because the ramps need to be



equal for all motors.

Your **new Axemotion Booster controller** features an even more advanced type of vector: a "slope vector" with different speeds at begin and end. So now the speed for the new vector can exactly match the speed of that motor for the previous vector. This allows make a speed curve when chaining vectors, where other controllers must make stairs.



The purpose of the **kinematics function** is to calculate these matching speeds.

The basic form of kinematics calculation is based on the angle between the two consecutive vectors. It (still) is used by many CNC drivers. The trick is to calculate the angle (2D or 3D) between two consecutive vectors, and reduce the passing speed inversely proportional. At 90° , the passing speed is null, at 0° it is the same as the target speed. This works fine with big curves, but unfortunately it gives poor result with small curves. If you consider a 180° curve of 100 mm radius, made of 180 vectors, then the angle between two consecutive vectors is very small (only 1°) so you can imagine that the machine can go fast through it. But if the radius is 10 mm, indeed you cannot get through at high speed - though the vectors have the same angles. Early versions of Galaad/Kay were adding a radius variable, but this is not the best solution because motors are independent from one another.

The kinematics calculation that is currently used in Kay and Galaad consists in searching for speed gaps between consecutive vectors. Of course, any path starts and ends with a gap since the speed is null at the start and the end of the path.. When there is a difference of speeds between two vectors, that difference has influence before and after, the range being defined by the acceleration capabilities of the motors. Due to this range of influence, the behavior of the machine will not be the same in a large and in a small curve. This is the theory; the algorithm is in fact much more complicated (in fact by far the most complicated algorithm in Galaad). Changing the speed of one motor which has a critical chaining to perform of course induces the same proportion of change for the other motors, which can then get into a critical situation themselves. So these are parallel calculations with cross influences, not very far from fuzzy logic.

With the kinematics calculation, the feed speed is no longer a constraint but becomes a simple target whenever possible. Galaad splits vector ends in small interpolated sub-vectors which will allow make speed stairs to help motors change speed along an artificial ramp, so there are no gaps. For most machine this artificial ramp will look like a staircase: a series of vectors of increasing or decreasing speed, each vector having a constant speed and no ramps. For your **AxeMotion** card the situation is better as here slope vectors can be used: one slope vector can replace a series of stairs, and the complete speed stairs will become a speed curve.

The drawback is the fact that we add many vectors to make these speed curves, and the transmission (from PC to controller) is more loaded. This adding of sub-vectors will of course only be done in case needed: when the calculation determines that a vector has passing speeds under Fss at both ends then there is no need to add vectors. Galaad / Kay then will simply send a vector with ramps so the accelerations will be processed locally

Because of the extra vectors a large communication bandwidth is needed, which in fact is big problem. Even on a USB port, transmitting data costs time, especially in conversational mode where telegrams sent to the port are short and interactive. USB communication taxes a fixed lapse of time at every transmission, so the telegram must be as big as possible. Unfortunately, the IMC4 Booster CNC controller does not have a huge buffer for storing received data (the original IMC4 does not have any buffer at all), and it is already performing motion which takes most of its CPU time.

The communication speed on the USB cable can be compared to a serial port at 115200 baud, which is fast but not light speed. If the path has thousands of fast mini-vectors, each vector requiring 2 or 3 co-ordinates and 1 speed information, then we may get into a situation where the machine crunches vectors faster than the transmission can send. At a moment, the local memory buffer will become empty and then the last executed vector will not be chained to the next one which is still being received. This induces a time gap where the chain is broken - and if the last vector ended at a high speed of course steps are lost. Steps are then lost as well (in the other direction) when the next vector starts, also at too high speed. The result is hard knocking, exactly like too high accelerations, though the problem is the transmission. At the next long vector, the buffer gets its advance since a mini-vector and a long vector cost the same transmission time.

To avoid this, Galaad sends a small pause as first command in the buffer, which gives some time for sending the next vectors though the tool yet did not start moving. This will happen at program start, and also after any rapid move half-way the path. The delay is calculated but the result is not perfect: sometimes it is too long (up to 3 seconds) and the user does not understand why the machine is not moving. It is difficult to explain that in fact the machine is receiving vectors ahead.

In short:

The AxeMotion cards allow Kay/Galaad to calculate an optimal feedrate for each vector: begin & end ramps are not made if not needed. The calculation is quite complicated and tries to make the machine move as fast as possible (the theoretical feedrate being a ceiling) without losing steps, i.e. without making frequency jumps that the inertia of motors & axes could not allow. So new vectors are added to build artificial acceleration and deceleration stairs or curves where needed (in case of big frequency jumps). The actual number of vectors may be easily multiplied by 10 or 20 if the path contains many sharp angles or small curves.

APPENDIX B: Card settings

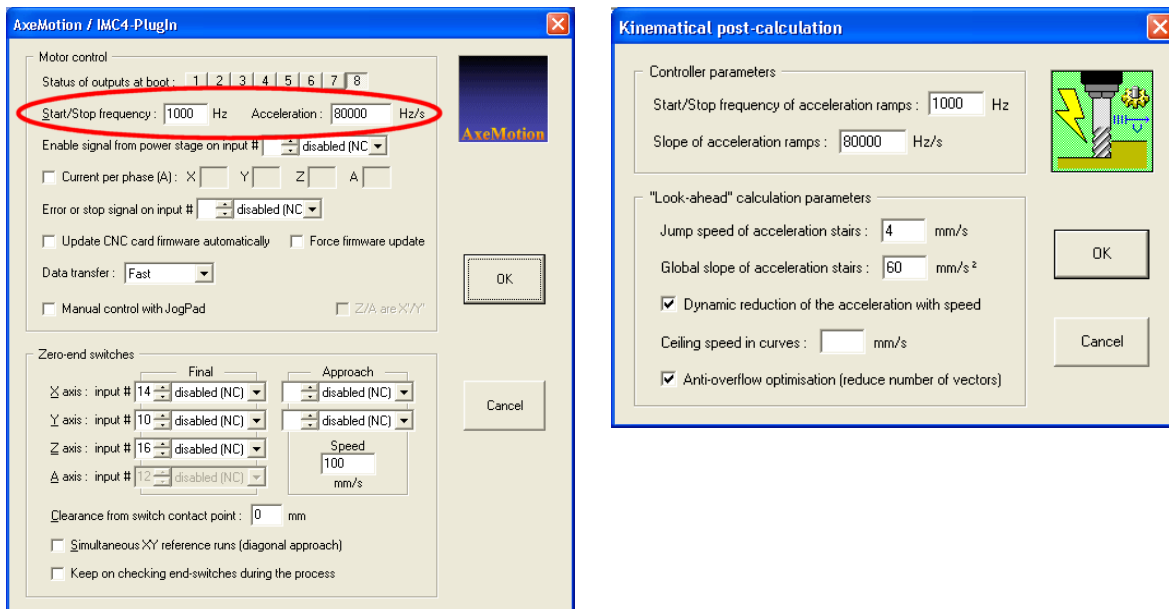
For the kinematics calculations done in Kay or Galaad many detail settings can be made. These settings should be OK on installation, still it may be needed for you to have some background information.

Note that the actual maximum speed for the CPM and ICP machines is limited by the stepper motors that are used. These are hi-torque motors and cannot turn fast: 5 rev/s maximum, which for a 10 mm pitch means 50 mm/sec. So in fact the speed ceiling is not the card's maximum output frequency but the motors themselves.

The kinematics settings in Kay can be found in two different dialogs.

First the **AxeMotion dialog** that was already shown in Chapter 3 (in Kay: Parameters menu >> Machine parameters >> Tab Controller >> button "More" next to IMC-plugin). This illustration is repeated below left.

The second dialog with booster card settings is the **Kinematical post calculation dialog**, shown below right. You can find it via the fifth tab of the machine parameters: Advanced. Here you can call the extra dialog by enabling the checkbox "Enable kinematics calculation". When it is already enabled then first deselect and then again enable this checkbox: then the dialog with kinematics settings will pop up.



The two kinematics parameters in the AxeMotion dialog are the same as in the top frame of the kinematics post-calculation dialog box. They concern only the vectors with ramps, fully managed locally by the controller. These are used for isolated vectors (all movements in manual mode, and rapid positioning movements in automatic machining) or vectors between sharp angles where the calculation considers they have slow passing speeds at ends. These values are easy to setup: the manual drive at fast speed quickly shows if the acceleration is too smooth or if it is too hard (motors knock or stall).

- The Start/Stop frequency.

This parameter, also called Fss, has been explained in the previous Appendix. It should be set roughly to half of the resolution of the least multiplied motor, regardless of ball screws. For example, if you have 800 pulses/rev on XY and 1600 pulses/rev on Z, then it should be set to $800/2=400$ Hz. Stepper motors have their maximum torque at low speed, so they can skip instantly to the turning frequency entered here. If the motors have big torque, then it is possible to set it to a greater value, but anyway it is not very helpful. The common range between $0.5 \cdot \text{resolution}$ and $1 \cdot \text{resolution}$, so in the above example between 400 and 800 Hz. If the most critical motor knocks even with a weak acceleration, then Fss should be reduced (divide by 2 until it does not knock anymore).

On the CPM machine (1600 pulses/rev), Fss should thus be set to **800 Hz**. The motors have a big torque and the value can be higher, but there is no need for it. On 4 mm screws, in theory this value can be increased a bit, in practice the same value is used. In case the machine ‘knocks’ (makes funny noises) in manual drive then you can reduce in steps of 200. Note that for this test the manual speed has to be set on the maximum value.

- The Slope of **Acceleration** ramps.

This is the slope from Fss to target speed, starting from the above-mentioned Fss frequency until the demanded speed is reached (see Appendix A). If it is too high, then the motors stall (it will whistle) and refuse to turn (lose steps). If it is too low, then the motion is smooth and long: the machine will look heavy with soft accelerations.

On the CPM machine, the acceleration should be between **60000 Hz/s** and 80000 Hz/s with 10 mm screws, and between 120000 - 160000 Hz/s with 4 mm screws. The AxeMotion card cannot perform accelerations of more than 160 KHz/s, so it is useless to set it to a greater value.

In case the Acceleration frequency is too high this may result in losing steps during high load conditions, you then can try reducing to 75000 or even 50000. The manual drive says much in that matter, provided that you set the motion speed to the highest value.

There is a function for calibrating these two parameters in the Galaad program: "Parameters / Machine / Calibrate". It gives good results for maximum speed and local ramp parameters, but of course it is more fuzzy on look-ahead parameters because it cannot match all cases. Anyway the user should "listen to" his machine. The ear says a lot about motor knocks.

The bottom frame in the kinematics dialog box concerns the heart of the calculation, i.e. chaining vectors without ramps along curves. These are in fact the settings for **the look-ahead function**, that adapts the speed to the shape of the toolpath to come: high speed when paths will follow at low curvature, decelerate when a sharp corner is approaching.

- The **Jump speed** of acceleration stairs.

is the maximum gap that the most critical motor (the weakest or the one that drives the most loaded axis) can perform. It is given in speed units instead of Hz because the calculation applies on speeds, regardless of the gear factors. The jump speed is the height of the speed stairs. If it is too high, the motors will lose steps; if it is too low, there will be more vectors and the bandwidth will suffer.

On a CPM machine, I would set the jump speed to **4 mm/s** with 10 mm screws, and 6 mm/s with 4 mm screws. Sorry the ratio is not linear.

- The **Global slope** of acceleration stairs.

is the hardness of the acceleration, i.e. the angle of slope of the speed-curve / stairway. You can consider the tool is like a car in a crooked circuit: it must brake before a sharp corner and re-accelerate after. The global slope is the capability to brake and accelerate. If it is too low, the acceleration will take long and the average speed will be decreased; if it is too high, the motors will be unable to brake enough so steps will be lost in the curve.

On the CPM, acceleration slope should be about **60 mm/s²**. The torque of stepper motors decrease with speed, so the acceleration capacity is not stable; it depends on the speed. Galaad / Kay integrates this in the kinematics algorithm, but unfortunately the decrease is not linear. There is a stall after a given speed which depends on the motor and the voltage it receives (the higher the better, since the coil gets its charge faster). With 4 mm screws, the slope can be set to 80 mm/s².

- **Dynamic reduction** of the acceleration with speed.

This option makes the reduction of both jump speed and global slope depend on the path speed.

The CPM motors cannot turn very fast and limit the maximum speed 50 mm/s with screws of 10 mm or 20 mm/s with screws of 4 mm. Faster is physically impossible for these motors. As the dynamic reduction option applies to machines which run at 100 mm/s and above, it is not important for the CPM. Nevertheless by default it is set to **Active**.

- The **Ceiling speed** in curves.

Parameter to avoid getting too fast in curves. When the maximum speed for the machine has been correctly set this parameter can be ignored for CPM machines. The kinematics calculation is supposed to take care of this maximum. So as default the edit box for this parameter is left **blank**.

- The **Anti-overflow optimization**.

This option shortcuts mini-vectors which are not significant and only take bandwidth. A non significant vector makes an angle of less than 3° with the previous one and has a small length. When the option is active, Galaad

shortcuts vectors according to speed: a vector of 0.02 mm will be ignored at 10 mm/s, a vector of 0.1 mm will be ignored at 50 mm/s (roughly). The vectors that are ignored cannot be longer than 1 mm and cannot make a direction change of more than 5°. Since on a CPM the maximum speed is only 50 mm/sec this option will not be very useful for most NC files, still by default it is **Active**.

Conclusion:

If you lose steps in manual drive, then reduce the Start/Stop frequency and possibly the Slope of acceleration. If you lose steps during the milling process, then reduce the Jump speed and Global slope values.

Of course, if you use a machine that has more powerful motors or less inertia, then you can increase all kinematics parameters and consequently the actual machining time can be reduced a lot. If the kinematics parameters are correct it should be impossible to go faster without losing steps. The CPM machine has high-torque motors that unfortunately cannot turn fast (and this is why they fit the machine with 10 mm ballscrews), and their torque decreases a lot with speed, which ceilings the jump speed. Furthermore the axes of the machine are somewhat heavy. This is a good warranty for less vibrations when milling metal, but it requires much torque to make the axes accelerate and decelerate. These machine characteristics set the maximum speed and acceleration that can be achieved, and are a match with the parameter settings mentioned above.

Racing car comparison:

The best comparison for kinematics in a machine tool is a car in a racing circuit (this is simple 2D but expandable to as many axes as you want), that car having exactly the same capabilities for accelerating and braking. If a curve is sharp, then the car can run through it only at low speed. It must brake before and accelerate after, the braking distance depending on the speed it had before the curve. If you think your curves are passed too slowly, then yes the machine is slowing too much and that means the braking distance is too long. Increasing the **Slope of acceleration** parameter reduces the braking distance and the curve is entered faster. Of course, if it is too high, the tool gets out of the path and this is a loss of steps. The value you have to find is the most powerful acceleration that does not induce loss of steps. Of course it is better to keep a small margin so there are no risks of passing the limits. To test you can use a crooky path with some long straight lines and see if the machine motion is not too hard nor too soft.

The above will be applied when you machine for instance a rectangle: four lines at 90 degree angles. Kay / Galaad will use classical vectors with ramps because begin- and end-speeds of a given vector are both under the Start/Stop frequency (Fss), and all kinematics is managed locally (by the controller card).

For a rectangle with rounded corners the speed at the end of the straight lines may be faster than the Start/Stop frequency, and thus only vectors at constant speed may be used (so without ramps to Fss). Kay / Galaad will add vectors that make the stairs of an acceleration slope, the height of the stairs being the **Jump speed** which is actually a gap of frequency between two consecutive vectors. For the Axemotion card instead of a speed stairs a speed curve can be used. Axes are no more considered together but one by one, because the X motor is not concerned by the frequency gaps of the Y motor, etc. Like coordinates, we can split the tangent speed in speed components for each axis, and any speed reduction for a motor has influence on the speed of other axes. This is where the calculation becomes tricky. The only thing that must absolutely be avoided is that one of the motors makes a speed jump higher than its capabilities. This would surely induce a loss of steps for that motor. The advantage of this approach is that it is valid whatever the number of axes.

In a few words: the acceleration slope parameter is the braking capability of the car, and the jump speed parameter the gap between two gears in the gear-box.

APPENDIX B: Card specifications

Main characteristics of the AxeMotion CPM Booster card:

- 4 Axis CNC module with USB interface, directly plugged on the IMC4 motherboard
- USB powered for high-rate data transmission.
- Up to 4 axis, with fast interpolation on 2, 3 or 4 axis simultaneously
- Advanced kinematics possibilities
- 35 KHz maximum output frequency
- Look-ahead buffer (250 vectors)
- Extension to 8 Inputs / 8 outputs (with an extension card)
- One analog 0-10V output
- Connector for AxMoBus extension
- Compatible with Kay and Galaad.

The following Isel machine have the build-in IMC-4 controller and can be upgraded with this Booster card:

- CPM 2018 / CPM 3020 / CPM 4030
- ICP2015 / ICP 3020 / ICP 4030.
- GFM 4433

For more information:

www.axemotion.com - Hardware
www.galaad.net - Control Software
www.deskproto.com - 3D CAM software.