CONCEPTS, INC.

The Optional Stop

© Copyright 2002, CNC Concepts, Inc.

Fall 2002 **Tips For Fanuc Control Users From CNC Concepts, Inc.** Issue 53

44 Little Cahill Road *Cary, IL 60013* *Ph: (847) 639-8847 FAX: (847) 639-8857*

Helping management understand the CNC programming process

by Mike Lynch

This article was inspired by a letter I recently received from Tom Stanley of VAE Nortrak, Ltd. Here is a portion of his letter:

Dear Mr. Lynch,

Have you ever considered writing a short book for supervisors and managers? Let me explain. We have purchased a number of books, videos and courses from you over the years and been successful in applying the principles that you teach, at least as far as the machinists go. On the other hand, our company has problems with the conversion of 2D drawings into useable programs. Our management doesn't understand why they cannot hire a young engineer straight from school or a machinist off the shop floor, give them a week's training in the CAM software and have them produce programs right away (especially when the new hire already knows G-code).

When we introduce a CAM system in new plants I'm often sent to these plants for a week at a time to bring the new people up to speed. But if the person doesn't perform right away, or if another opening comes up, they move the programmer I've taught into another job and bring in someone new (starting from scratch again). Management finds that the CNC machines' productivity is not as expected, but can't seem to figure out why.

What I am looking for is a book to explain to management and supervisors what this process is, helping them understand what happens from the issuing of a 2d drawing to the downloading of a CNC program into a machine. It should also emphasize how important this process is in the efficient creation of a finished workpiece from the 2d drawing. If management and supervisors better understood this process, they might give more attention to the training of their CNC programmers. Currently, my impression is that most managers see the 2d drawing issued, it goes into a black box, and a program comes out the other side; and because of their lack of knowledge, they tend to view the process as unimportant in the overall productivity of the plant.

Yours truly,
Tom Stanley, Industrial Engineer,
VAE Nortrak Ltd.

I agree that in many companies, management people (plant managers, shop superintendents, foremen, etc.) do not understand the process of CNC program development – at least not to the extent that they should. I'll also add to this list of ill-informed people: design engineers, manufacturing engineers, schedulers, quality control people, and just about anyone else not working directly with the CNC machine tools. And since managers are the people that decide who receives training and to what extent training is done, a company's productivity – if not its very survival – is commonly placed in jeopardy due to this ignorance.

While this subject may eventually form the basis of a book, in this relatively short article I intend to present some general information related to what management must know about the CNC programming process and give some examples of why each function is important. Along the way I'll also describe some attributes of a proficient CNC programmer. Note that this article primarily targets management people. As CNC people, our normal readership will easily understand all points being made. Again, the goal is to help you inform management people. You should be able to get the ball rolling by simply giving your management this article to read.

Management must, of course, be interested in what you have to say. I've found that progressive managers are very interested in maintaining the highest productivity levels and

In This Issue:

Subscription Information:
The Optional Stop is published quarterly by CNC Concepts, Inc. and is distributed free of charge to people downloading it from our website (www.cncci.com). Back issues are available, but a $10.00 charge per issue will apply to any back isues not currently published on our site. Back issues 1-50 are available for a total of $199.00.

have already done what it takes to learn what they need to know about CNC to be effective managers. But in companies that don't have interested managers, there may not be much you can do to educate them. We're assuming your management is interested in improving the CNC environment and that they need/want your help. In any event, concerned CNC people should be ready and willing to relate what they know about CNC to their management in an understandable manner, helping them make informed decisions about the CNC environment. As we describe each topic, CNC people should be thinking about how the topic applies to their own company, and how best it can be related to management.

The role of a CNC coordinator

The issue that Tom Stanley raises is that often management does not understand the responsibilities of the various CNC people in the company - specifically programmers. And if management does not understand the responsibilities of CNC people, they cannot solve problems in the CNC environment in an appropriate manner. In one company I have visited, for example, management was convinced that the unacceptable scrap rate coming from a particular CNC cell was being caused by their CNC operators. Management felt that these operators were unfamiliar with their responsibilities and needed more training – which was the reason why I was brought in. Once I began working with the operators, however, I found that they were among the most knowledgeable CNC operators I'd ever worked with. They understood every aspect of CNC machine operation. On the other hand, the *process* being used to machine the workpieces was so poor that even these highly skilled operators could not make good parts! (By the way, the process engineer responsible for this cell was more than happy to let upper management continue believing that operators were to blame for the scrap, even though I'm sure this person knew better.) Again, it was the ignorance of upper management that caused them to incorrectly diagnose the problem.

The first suggestion I offer is to assign someone in your company the role of CNC coordinator. I strongly feel that every company should have a CNC coordinator, regardless of company size or the formality of the title. In a large company with many CNC machines, the related responsibilities may require a full time position. In smaller companies, this person can double as a CNC programmer, trainer, or setup person. The CNC coordinator should, of course, have a good understanding of the company's CNC equipment and all facets of its use. They should also understand how CNC fits into the company's overall manufacturing methods. And they should have good communication skills to relate what they know and draw people together. Indeed, the primary role of the CNC coordinator should be to facilitate communication and understanding among people in all areas of the company – especially when it comes to those areas that affect the productivity of CNC machine tools.

A CNC coordinator should be involved with helping everyone in the company (including management people) understand the responsibilities for the company's CNC people. Indeed, the CNC coordinator may even be the person actually assigning responsibilities. If the company mentioned above would have had a CNC coordinator, the scrap problem would have been correctly diagnosed as being related to a poor process.

The impact of a programmer on CNC machine productivity

It should almost go without saying that a CNC programmer will directly affect the productivity of the machine/s they program. But as Tom Stanley points our, decisions made by some managers don't reflect a thorough understanding of this point. So let's list a few specific areas. Note that many managers complain about the productivity related to these areas, but never address the programmer's lack of understanding related to them (maybe they don't know the programmer is responsible for them?), so the programmer doesn't receive the training needed to improve relative to these productivity issue/s.

Efficiency of program execution – Proficient programmers know the tricks and short-cuts that make CNC machines execute programs as efficiently as possible. Examples include: programming miscellaneous functions (M codes) within motion commands, combining machine functions when feasible and safe, and programming efficient motion commands.

Appropriateness and efficiency of process – Most metal-cutting CNC machines hold several cutting tools. And most companies expect their CNC programmer to develop the step-by-step machining order (process) for workpieces they program. The process is of paramount importance to the success of the job. A good, efficient process will make good parts (easily held within all tolerance bands) in an operator-friendly manner. A poor process, at best, will be inefficient and difficult for the CNC operator to run. And an incorrect process will make it impossible to machine good parts.

Appropriateness of work holding setup design – While some companies employ tool designers for the purpose of purchasing and/or designing fixtures and other workholding devices, many companies assign this task to the CNC programmer. Even if a separate tool designer is employed, the task of documenting the workholding setup falls to the CNC programmer. The CNC programmer must be well versed with the workholding devices your company uses. Appropriate workholding setups allow optimum cutting conditions to be safely used, meaning efficient machining.

Appropriateness of cutting tool selection – CNC programmers are commonly responsible for selecting the cutting tools used in their programs. The decisions related to cutting tool selection must be appropriate to, among other things, the materials being machined, lot sizes to be run, and the rigidity of the workholding setup and CNC machine tool itself. Inappropriate tool selection leads to, at best, inefficient machining. At worst, it can lead to damaged or scrap workpieces and possibly damaged cutting tools.

Efficiency of cutting conditions – Most companies leave it to the CNC programmer to determine cutting conditions for the programs they create. Of course, appropriate cutting conditions lead to efficient machining. Cutting condition selection that is too aggressive leads to shortened tool life, higher tooling costs, and possibly damage to tools, workpieces, and machines when tools break. Cutting condition selection that is too conservative leads to inefficient machining. Note that cutting condition selection tends to be a very subjective topic. It is not unusual to see two or more programmers in a company programming dramatically different cutting conditions, even when using the same cutting tools and machining the same materials held in the same workholding devices! Unless you provide training, how can you ensure that *all* programmers use appropriate cutting conditions?

Effectiveness of documentation – Programmers are commonly responsible for providing documentation for the programs they develop. If left to their own devices, documentation provided by programmers can range from barely adequate (or inadequate) to overkill. Documentation must be appropriate to the skill level of the person using the documentation and the complexity of the job. It should target the lowest skill level of person you expect to perform the task described by the documentation. Poor documentation results in wasted time, duplication of effort, scrap, and mistakes. Appropriate documentation leads to efficient setups and production runs. Note that many companies provide great setup documentation for the setup person but inadequate (or no) production run documentation for the lesser skilled CNC operator.

Other areas a CNC programmer will impact – The CNC programmer must have a good working knowledge of almost all departments within the company. In some cases, this is necessary when the programmer needs to get a clarification. In others, a proficient programmer may be able to help with the manufacturing process as a whole. For example, a programmer should have easy access to the design engineering department. A good programmer will be able to point out questionable dimensions or tolerances, and in some cases may be able to spot design mistakes. In many companies, the programmer must work closely with the quality control department in order to provide good setup and production run documentation for gauges and other measuring tools used from job to job. The programmer may have to work closely with the production control department in order to provide new programs where they are required on a timely basis. Frankly speaking, there aren't many areas of a manufacturing company with which a CNC programmer doesn't have need to have a good relationship.

Programming methods

Generally speaking, programs are prepared in one of three ways.

Manual programming – With this method, a programmer writes the program in the same language a CNC machine will execute the program. This form of programming is commonly called "G code programming". It is this kind of programming that allows a programmer to be the most intimate with the CNC machine tool, commanding *every* action that the machine must perform. It's great for simple work, or when production quantities are so high that every last second of program execution must be eliminated from the machining cycle. It is also important to understand G code when programs must be edited at the machine (again, the CNC machine runs the program from G code). This method of programming, of course, requires the programmer to know all CNC words and commands needed for each machine they program. As the number of machines they must program grows, so does the difficulty related to remembering the various words and commands for each machine.

Computer aided manufacturing (CAM) system programming – Since manual programming is so tedious, most manufacturing companies use some kind of CAM system to help with the programming process. A good CAM system (one that's appropriate for the machines and processes in your company) will eliminate much of the drudgery of programming. But remember, a CAM system is only a tool used by the programmer – and there's nothing magical about a CAM system. A good CAM system will minimize the math required to prepare programs and will allow the programmer to work at a higher level than when manual programming. CAM systems can also help with the selection of cutting conditions, though the programmer is still ultimately responsible for cutting condition selection. Though this is a bit of an over-simplification, the programmer will import a workpiece drawing (from the company's computer aided design [CAD] system) and specify the machining operations to be performed. The CAM system will then create the G code program – just like one written manually – for the particular CNC machine to be used for the job (the CAM system handles G code variations among machines). Again, think of a CAM system as nothing more than a tool the programmer uses – a glorified calculator. The programmer must still develop the machining process, select the cutting tools, select the workholding device, design the setup, develop all documentation, and perform all efficiency-related tasks mentioned earlier. And note that CAM systems are notorious for creating rather inefficient CNC programs – so the programmer must still understand G code in order to tweak the program at the machine for optimizing purposes.

Shop floor programming – Also called conversational programming, this form of programming is commonly done when there is no time or personnel to prepare programs while the machine is running production (all new jobs, short cycle times, and small lots). This is commonly the case in job shop and tool-room environments. A special CNC control must be used on each machine and the CNC operator creates the program right at the machine. You can think of a conversational control as being like a single-purpose CAM system. Again, this kind of control eliminates much of the drudgery of manual programming, but the programmer must still perform the tasks described earlier.

We cannot stress enough the fact that the actual act of programming is but one of the many functions a programmer performs. In fact, most experienced programmers would agree that the act of programming is the simplest part of their job! The real work lies in developing workable processes, developing documentation, selecting workholding tools, cutting tools, and cutting conditions. I've talked to managers that actually believe that the programmer's job is limited to using the CAM system owned by the company – or that the CAM system will somehow eliminate the programmer's need to know about other manufacturing-related functions.

Program creation from start to finish

As per Tom Stanley's request in his letter, here we list the various tasks a programmer must complete before acceptable workpieces can be run on the CNC machine. We do so in the approximate order the tasks are performed. Note once again that companies vary when it comes to what they expect of their CNC people – but rest assured that these tasks must be completed (by someone) before production can be run. We're assuming that things go smoothly during this process. Problems a programmer encounters during programming (like dimensioning problems on the print) will add to the tasks we list.

Print evaluation – The programmer must study the print for the workpiece to be programmed. The effort required, of course, depends upon the complexity of the workpiece and the drafting methods used by the design engineer.

Process development – The programmer must determine what machining operations are to be performed on the workpiece (many programmers mark up the print for clarification). They must also develop a step-by-step machining order. Most programmers use a special form that lets them document a description for each operation, the cutting tool to be used, and cutting conditions for each operation. Again, we cannot stress enough how important the process is to the development of a safe, efficient CNC program. Even a poorly formatted CNC program can eventually be made to work if the process is good. But even a perfectly developed program (one that does exactly what the programmer intends) will fail if the process is bad.

Workholding device selection and workholding setup design – Based upon the workpiece to be machined, the programmer will decide what kind of workholding device will be used to secure the workpiece during machining. In some companies, this selection is quite simple, since only a handful of different workholding devices are used. In others, this task may be quite difficult, especially if special fixtures must be designed and built.

Cutting tool selection – Based upon the process developed earlier, the programmer will select the cutting tool to be used for each machining operation and determine where the tool will be placed in the machine's turret or magazine (tool station number).

Cutting condition selection – Based upon cutting tool material, workpiece material, and the rigidity of the workholding setup, the programmer will select speeds, feedrates, and depths-of-cut for the various machining operations to be performed. Progressive programmers will reference technical information supplied by cutting tool manufacturers for recommended cutting conditions. If qustions or problems arise, a technical representative from the tooling manufacturer should be called in to help. Too many programmers (especially those coming from a machinist's background) use a seat-of-the-pants approach to selecting cutting conditions. And since slow cutting conditions will almost always work, optimum cutting conditions are seldom achieved. Most cutting tool manufacturers offer (free) courses related to developing apporopriate cutting conditions for the cutting tools they sell.

Determine tool paths and calculate coordinates – Manual programmers must determine the tool path and set of coordinates for each tool being used in the program. Most programmers will do this prior to actually writing the CNC program. Note that even CAM system programmers will have to develop tool paths for tools used in the program (but the CAM system will calculate coordinates), meaning they still have to know how each style of cutting tool will machine the workpiece.

Create the CNC program – Manual programmers will write the program at G code level, usually using a pencil and paper first. This program will then be typed into a CNC text editor so it can be quickly transferred to the CNC machine tool. CAM system programmers will work at a higher level, but the end result (G code program) will be the same. The CAM system will automatically save the G code program as a text file that can be quickly transferred to the CNC machine.

Desk-check the program – By one means or another, the programmer should double-check their work. In the past, manual programmers would simply re-read the program – or give it to another programmer to check. But today, there are any number of tool path verification software programs to help with program checking. Most are relatively inexpensive, work at G code level, and will allow the programmer to view tool paths on the display screen of a personal computer. Note that many current model CNC controls also come with graphic tool path display features – but remember – the computer-based systems let the programmer check new programs off line, while the machine is in production. Almost all CAM systems also provide this kind of tool path display, but most only do so as the CNC program is created (most cannot show the tool path for an edited G code level program).

Develop setup documentation – Assuming someone else will be actually setting up the job (not the programmer), the programmer must provide documentation for the person that does. We're assuming that your jobs will be repeated in the future. The more a job is repeated, the more important it is that good documentation be provided. The setup person will (obviously) be the person using this documentation. Setup documentation includes instructions for how to place the workholding device, the location of program zero, cutting tool placement in the machine's turret or magazine, offset assignments, and anything else that helps the setup person understand how the setup is supposed to be made.

Develop production run documentation – Many companies allow lesser skilled people to actually run out the job once the setup is made (CNC operators). Though this is the case, many companies minimize the documentation they provide their operators. Appropriate production run documentation should include workpiece loading instructions, gauging instructions, tool life information, and a list of which offsets control critical dimensions.

Transfer the CNC program to the CNC machine – Most companies use some for of distributive numerical control (DNC) system for this purpose. With manually activated DNC systems, the setup person must first get the CNC control ready to receive the program. They must then walk to the DNC system computer and give the command to send the program to the machine. With automatic DNC systems, the setup person can give the command to transfer the program from the CNC control panel (no need to walk back and forth).

Make the setup – While the programmer may not be the one actually making setups, he or she will be directing the person that does, providing instructions and handling any questions or problems that come up when the setup is made. Truly, programmers must have a good relationship with setup people and operators. Tasks that must be completed before the program can be run include (possibly among other things): gathering all components needed to make the setup, mounting the workholding device, assignment of program zero, and assembly of, measurement of, offset entry for, and loading of cutting tools.

Verify the program – If the programmer desk-checked the program with a tool path display system, the setup person can pretty much rest assured that the movements caused within the program will be acceptable. But this is about all they can be relatively sure of. Any number of other mistakes could cause the program to fail. For example, tools may not be properly placed, program zero may not be properly assigned, or cutting conditions may not be appropriate. For these reasons, the setup person must be extremely careful when running the program for the first time (even previously used programs). Machine functions like dry-run, single block, program check, and distance-to-go help setup people safely run a program for the first time, minimizing the potential for mishaps.

A note about trial machining - When the setup person sees a tolerance that is so small that they're worried about holding size, they will almost always have a way to make the cutting tool machine with a little excess stock, ensuring that the workpiece will not be scrapped when the tool cuts for the first time. After machining, the setup person will measure the workpiece and re-adjust so that the tool will machine within the tolerance band. Trial machining (also called gauge cuttin) takes time. Programmers should do all they can to help. A well trained programmer will included special trial machining commands within the program to facilitate the task of trial machining. This will minimize the potential for mistakes and reduce the time required for trial machining.

A note about proven programs – I've talked to managers that can't understand why problems are encountered and programs must be changed every time they are run. The comment "I thought we've run that job before", is a common one – as if just because a job has been run before means it should run flawlessly in the future. The act of "proving a program" – or creating "qualified programs" – doesn't just happen. A conscious effort must be made if you intend to ensure that a program will never have to be changed again in the future. In every case when an old program must be modified, something has changed since the last time the job was run. Maybe a cutting tool is setup differently today than it was three weeks ago and it can't cut at the same feedrate it could before. Maybe the material hardness has changed from the last time the job was run. Maybe a casting has more material on a surface to be machined than it did in the past. In order to truly qualify a program and eliminate the need to change it again, you must in turn eliminate those things that change from one time a job is run to the next. Some of these variations may be within the control of your CNC people. Maybe better documentation, for example, will eliminate the possibility for cutting tool variations. But other variations, like raw material variations, may be beyond the control of your CNC people. To hold your CNC people responsible for problems that are not within their control is unfair. In the case of the material hardness variations, for example, possibly your purchasing department is always buying material from the lowest bidder. The purchasing agent must understand that the money they're saving by buying from the lowest bidder is being lost (possibly many-fold) due to the time-consuming problems that are encountered every time a job is run.

First workpiece inspection – Again, this is commonly beyond the responsibilities of a CNC programmer. But during first workpiece inspection, problems may be found that require program modifications (especially with new jobs). Programmers must be readily available to make these modifications to avoid lost machine time.

Transfer program back to the DNC system – If changes are made during the program's verification or after first workpiece inspection, the modified program must be sent back to the DNC system. Most DNC systems will not overwrite the original program with the modified one. Instead it will be saved as a back-up file. It is commonly the programmer's responsibility to evaluate programs that have been sent back to the DNC system and overwrite originals as required. If they don't do this on a timely basis, of course, the old version of the program will be sent out to the machine the next time the job is run.

Bringing an entry-level programmer up to speed

At this point you should easily agree that a programmer's job involves much more than just learning how to run a CAM system. You should also agree that you cannot place a new person in a programming position (especially if the person comes from outside your company) and expect them to begin producing at optimum levels. A period of adjustment will always be necessary.

Most companies fill the CNC programmer position with people that come from one of three backgrounds. They may come from technical schools, they may come from other companies, or they may be promoted from within the company.

Training for many of the tasks described to this point is readily available from several sources. Technical schools, for example, offer training in basic machining practice, programming, setup, operation, and most areas needed by a CNC programmer. Most technical schools also offer CAM system training. Or you can send people directly to your CAM system supplier for training. My company (CNC Concepts, Inc.) offers CD-rom courses for basic CNC training as well as on more advanced CNC topics like setup reduction, cycle time reduction, and parametric programming. Most tooling manufacturers offer free training for the products they sell. And trade publications will keep your CNC people abreast of the latest developments in manufacturing technologies.

In addition to sources that supply generic training for common CNC fucntions, there are many company-specific functions that any new programmer will have to get familiar with before they can perform at peak levels. In many companies, managers (among other people) are so familiar with these company-specific functions that they take them for granted – and can't understand why newcomers have so much trouble with them.

Your company's corporate identity – Again, companies vary dramatically when it comes to how they apply their CNC machine tools. And we cannot stress enough the importance of a programmer's understanding your company's identity. There are many features available to CNC programmers. But what's right for one company will not be appropriate to another. In order to make wise decisions, a CNC programmer must have a thorough understanding of what your company is trying to accomplish in its manufacturing environment. Consider, for example, one very important factor that contributes to your company's corporate identity: typical lot size. To program appropriately, your programmer must consider lot size when making just about every programming decision (cutting tools to use, cutting conditions, fixturing, documentation, etc.). Inappropriate decisions in this area will result in lost time and effort. If, for instance, a programmer programs a job as a low runner, when in reality, there are thousands workpieces to run, the job won't run as efficiently as it should (the programmer may use high speed steel tools instead of carbide tools, for example). This is but one of the countless factors that contribute to your corporate identity. Others include typical lead time, amount of repeat business, tolerances held, materials machined, and size & complexity of work. Just about anything that affects the way your company does business should be considered a factor contributing to your company's corporate identity.

Your products – Your company makes something. You're probably so familiar with your product line (or the component workpieces you produce) that you don't even give it a second thought. You've probably even developed your own set of slang words or special terminology to talk about your products. New people, regardless of how much previous experience they have, will need to learn about your products and workpieces.

Your design engineering methods – While you may think that your workpiece drawings follow standard drafting methods, it's likely that your design engineers are using techniques that are not universal. Maybe they dimension a whole family of workpieces using one drawing. Maybe they minimize views to reduce paper use. Or maybe they don't dimension or tolerance in a consistent manner. In any event, it's likely that a newcomer will have to learn your company's specific design engineering methods.

Your methods – Though you may not think so, your company has certain ways of doing things that differ from the ways other companies would handle your problems. Your machines, tooling, processes, and personnel dictate how these methods have developed over the years – that is – it is an evolutionary process. And again, different companies evolve differently. Any newcomer will need to learn your methods, as well as why your company has evolved as it has.

Your machines – While CNC machine tools made by different machine tool builders behave in much the same manner, new CNC people will need to learn your specific machines, regardless of how much previous experience they have.

The problem with task-specific training

Most forms of training are *task-specific.* With this kind of training, a student is taught to perform specific tasks, limited by the scope of the class. In our basic CD-rom courses, for example, we teach the programming, setup, and operation tasks for CNC machining centers or turning centers. And as with any task-specific training, we limit the scope of our courses to include only those topics that are appropriate to the course content.

We assume, for example, that the student understands basic machining practices as it relates to the kind of machine tool being used. We assume that the student understands shop math, and can use shop gauging tools. We assume they understand workholding tools, cutting tools, and cutting conditions. And we assume that the student understands how CNC is being utilized by the company they're working for. Again, we're teaching CNC usage – not basic machining practices or company philosophies.

The problem with any task-specific training is that the student won't get the whole picture in a task-specific course. And since companies vary so dramatically when it comes to how they utilize their CNC machine tools, no single CNC course would work for all companies. As you now know, CNC programming requires a mastery of several disciplines. Management must think of any task-specific training course as but a single building block that makes up the foundation for a programmer's proficiency. If any building blocks are missing, the programmer will not be able to function at peak levels.

When you consider training anyone that requires the diverse knowledge of a CNC programmer, you must develop an appropriate foundation. You can, for example, send your CNC programmer to the training course offered by your CAM system supplier – and eventually the programmer will master the CAM system. But as stated, the CAM system is but a small part of what your programmer must know. CAM system training is but one of the many building blocks needed to form the foundation for a good CNC programmer.

Conclusion

In most companies, the largest individual productivity gain that can be achieved is related to improving the proficiency of people performing key tasks. Given the impact a CNCprogrammer can have on the productivity of each CNC machine, and in turn, the overall productivity of your company, a case can be made that your CNC programmers are among the most important people in your company. Improving their proficiency through training will likely result in tremendous productivity gains!

M01

Product Corner

With schools reopened for the fall session, we just want to remind instrcutors that are using our CNC Curriculums to regularly check the **Curriculum Update page** on our website. From our home page (www.cncci.com), click *CNC Curriculums* on the navigation menu. At the bottom of the page that comes up, click the link to *curriculum update information.* You'l find some important information, including correcctions to known problems and product update notices.

M01

CNC CONCEPTS, INC.
44 Little Cahill Road
Cary, IL 60013
Ph: (847) 639-8847
Fax: (847) 639-8857

The Optional Stop Newsletter Enclosed!

Mailroom: PAID SUBSCRIPTION ENCLOSED. Please assure delivery to addressee. Thank You!