Chapter 2

Planning the Process

Features

Thinking Before Integrating With Motion Control
How to Define the Problem
An Example Design Thought Process
Discussion

Thinking Before Integrating With Motion Control

The purpose of this chapter is to outline a method of design thinking required to integrate motion, machine, and control (software). There have been many occasions where I have been called to a job site to correct design failures that would not have occurred if the engineer had properly *defined the problem*. Properly defining the problem and thinking out design issues are topics discussed in this chapter.

System Integration in the motion control field consists of breathing life into stagnate assemblies of steel, wire, and electronic components. It is irrelevant whether the project is budgeted at hundreds of dollars or millions. The basic principles involved in the approach and solution of the *design thought process* will not differ significantly.

Always bear in mind that making something move is easy, controlling the move is more difficult, but integrating that move into a precision machine product that must work with other sections that both move and remain stationary is what requires good design *technique* and *method.*

Design begins with DEFINITION,

proceeds to CALCULATION,

continues with INTEGRATION,

and ends in CELEBRATION!

When completed, if the designer can view the *Big Picture,* all of the puzzle pieces will fit, and the machine will work! Coordinating <u>all</u> phases of the project (mechanical, electrical, control, and software) will insure project success. Pointing fingers or blaming *Murphy* for bad assumptions, incomplete assessments, or lack of communication will not "hold water" if you intend to be a proficient designer. By adhering to the ***Six P Rule*** . . .

P
roper
lanning
revents
_ss
oor
erformance

. . . you'll rarely come up short.

Viewing the ***Big Picture*** simply means to look and to see beyond the specifications, to envision the variables that count and the variables that do not. You need to be able to visualize the final product in action while it's still in the development stage. As with any learned behavior, the method(s) you choose to use will become intuitive. Learning proper thinking habits when problem-solving today will save you time and money tomorrow.

Everyone has the potential to be proficient at their chosen career, but it is important to shed the "It-can't-be-done" attitude, and "I'm-not-smart enough" thinking. You must train yourself to think positive, and to think in general terms—with the emphasis on **positive** and **in general**. You must rid yourself of the most significant restriction any designer can acquire - TUNNEL VISION.

> ***Tunnel vision*** *is defined as answering questions you think were asked. It is choosing solutions based on shotgun principles rather than relying on a thoughtful analysis of the process.*

Do *YOU* have tunnel vision? Try this simple test. What is your first **verbal** response to the directive. . .

Define . . . Motion Control?

If you responded with a specific answer about the use of motors, servos, robotics, cylinders, computer control, etcetera, you probably have it! If you responded in general terms with something resembling . . . being in control of motion, controlling movement, or coordinating moving and stationary objects, and such . . . you probably see the *Big Picture.*

Note that the question did not ask for something specific (although *thinking* about specifics is OK). The question asked something that has no boundaries. Motion control is a field without boundaries! Big picture thinking is not limited by boundaries. Specific answers to general questions display a sense of personal restriction. This sense does not prevent you from creating, but it does *restrict* your creativity. When you place restrictions on yourself, you also place them on your projects. The guidelines to good critical thinking skills for design are:

- **Do not restrict yourself to specifics until necessary.** General considerations come first, and you will be quite busy with them. Accomplish the details later.
- **Do not read anything into the problem you are facing.** People tend to heap problems on top of problems by trying to control variables that will not affect the project outcome (the "what-if" syndrome).
- **Maintain a *positive general* outlook.** I received a bookmark while a senior in high school (which I still have). It states . . .

If you think you're beaten, you are.
If you think you dare not, you don't.
If you'd like to win, but think you can't,
it's almost a cinch you won't.
If you think you'll loose, you've lost.
For out of the world you'll find success
begins with a fellow's will,
It's all in your state of mind.

- **Read and think.** The most important element in problem solving is *common sense* and a good understanding of the basic laws of your field.
- **Remember what you have seen, what has worked, and what has not.** Shed the *pride-of-authorship*, and the *one-upmanship* attitudes. If a more expedient product is available, use it. If your method will **almost** do the job, find a better one that will **completely** do the job. Your objective is to *get the job done*, not to be a hero. Do not reinvent the wheel. Build a library of methods, and learn from others who have already been down those roads.
- **Ask questions.** The only foolish people are those who *know everything.*
- **Take nothing for granted.** If something can fail, it will—in spite of *Murphy. It is the OVERSIGHTS that cause a system to fail. Consider* ***all*** *of the possibilities . . .* ***the Big Picture.***
- **Be persistent.** Nothing is given; you must work hard for good results.

The eight points above establish a base that we can draw solutions from. (I doubt very seriously if anyone ever solved any problem without some form of knowledge base and/or long hard hours of thought.)

How to Define the Problem

Good thinking skills provide us with the means to figure out just what it is we are trying to do. I am amazed how some designers immerse themselves in a problem without clearly understanding the **goals** or **objectives.** In order to complete any project, you can basically take one of two approaches:

- Go directly from point *A* to point *B*.
- Take the scenic route.

Taking the scenic route may be acceptable when you are on vacation. You expect to spend extra time and money for sideline excursions. However, when designing, the direct approach should be your only choice; it saves both time and money. Never forget who is financing the project. You may be an employee, but it does not give you a license to indulge. Your employer is paying you just as any other contractor.

Exactly how do you define a problem furnished in a specification? The best answer is to digest the problem statement, and restate it in progressively simpler terms (see Chapter 1). Eliminate the flowery descriptive language and all non-related verbiage, and the real problem to be solved will emerge in the end.

Let's try one . . .

Example: The Flying Cutoff Machine

Specification . . .

The material to be cut is in constant motion at 60 fpm. The cutter is mounted on a carriage; which, when at the matched linear velocity of the material, will signal a crosscut to be performed on the material. We need to ensure that the specified cut length (8 ft.) is within a maximum of ±1/32 inch. The cutter carriage travel is 5 ft. (60 inches).

Questions to resolve:

- What is the problem in its simplest form?
- What are the desired results: accuracy, repeatability, or both?

Define the problem . . .

We need to achieve material (line) velocity, make a cut, stop, then return the carriage to its zero position *before* the next point to cut passes the zero position (8 sec. intervals).

Now simplify . . .

Match speed, cut, stop, return to zero.

Notice that only four operational requirements are listed in the problem definition. Nothing is expressed about tolerances, backlash, weights, product stability, gearing, following error, control type, resolution, or any other specifics. They will work their way in or out of the problem's solution as it develops.

Develop the solution . . .

The weak-link variable in the operation is the material cutting speed. Just how fast will the crosscutter tooling or material allow the cross cut to be made? The material may have a cutting speed restriction; or, on the other hand, the cutter may have a speed restriction (such as a water jet). If the cutting speed is restricted, the carriage may crash into the far hard-stop before completing the cut. Therefore, knowing just how fast the cut can be made will govern the length of the machine. The maximum cut length for any material velocity can be determined by knowing the overall travel length of the machine, the acceleration and deceleration rate of the cutter carrier to get to speed, and the return to Home (zero) profile time.

Discussion . . .

In this example, we learned by talking with the customer, the material vendor, and the mechanical design engineer that the cut can be made in 2.5 seconds (2.5 feet of carriage forward travel). Therefore, we know that 2.5 feet of carriage travel remains.

To allow crash limit switches to adequately stop the motion (for instance, in the event of a failed cut-complete signal from the crosscutter unit), we will allow six inches of travel at each end or ten percent of the 60 inch carriage travel available (an experience guideline).

By simple arithmetic:

$$\textit{Carriage travel remaining} = 60 - 30 - 6 - 6 = 18 \textit{ inches}$$

There is nine inches of travel remaining for the acceleration and deceleration profiles. This presents a problem at the crash switch points because we have only six inches.[1] We will split the difference and allow 7.5 inches for the acceleration and deceleration profiles. Forward, and reverse limit over-travel protection will also be 7.5 inches. The carriage return transition-to-zero velocity can then be set to a reasonably high speed to ensure locating the carrier to the zero position before the next cut. If we return the carrier to the zero position at a speed of 120 fpm (twice the material line speed), we will maintain a comfortable time margin before the next cut. Also, we discovered that the return speed required will actually be less than the selected 120 fpm, but if we use this two-times-the-line-speed value as a guideline, the carrier motor will allow a comfortable speed margin increase for future situations.

The next step is to actually profile the move operations, both cutting and return-to-zero, to find the worst case RMS torque loading for motor, drive and gearing selection. At this point, it is prudent to solicit carriage weights and other mechanical information from the mechanical engineer, so that as we profile, we can insure that the system mechanics can withstand the inertial torque-loading.

If we design the system control for ***repeatability***, then all repeatable mechanical and electrical error (such as following error, backlash, leadscrew/belt/rack, and pinion error) can be adjusted out in the system operation. Figure 2.1 illustrates the difference between *accuracy* and *repeatability*. Box B

---

[1] Some designers allow the carrier to decelerate to a stop when a limit switch is tripped.

shows the target holes largely within the target boundaries, but can you adjust for the error in accuracy? What about the target holes surrounded by box A? They are not very accurate, but they are repeatable; therefore, with adjustment they can be made accurate.

Designing with repeatability will also allow encoder resolution to be relatively loose (required tolerance divided by 10). Since the specification calls for ±1/32 (.0375) inch, our encoder need only be:

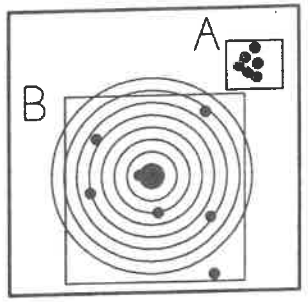

Figure 2.1 Which box can be adjusted in, Box A or Box B?

$$\frac{0.0375 \ \mathit{inch}}{10} = 0.00375 \ \mathit{inches/count}$$

In this example, we designed the system control to count the mainline material encoder pulses in *real-time* rather than using the *updating* technique. This will require a pulse and direction type encoder (a pulse for the interrupt handler jump, and direction for the software counting scheme).

The frequency of the encoder pulse signal will be:

$$\left(\frac{60 \ \mathit{fpm}}{60 \ \mathrm{sec}}\right)\left(\frac{12 \ \mathit{inches}}{0.00375}\right) = 3200 \ \mathit{Hz}.$$

For *real-time* operation, this will yield a worst case repeat interrupt handling time of 312 microseconds. If we design the software to spend no more than 150 microseconds in the interrupt handler we can insure that the system will work and hold to specifications. The machine will also maintain a comfortable margin of available speed increase (one of those inevitable *creeping feature-isms*). In addition, if we size the motors properly, the only obstacle to system performance will be the *cutter* capability, as stated previously.

> ***Creeping feature-isms*** *are usually paid for out of project profit money. They are the after thoughts that come about once the machine has been completed (and generally shipped). If you can determine these features in the beginning, the project will be more of a dream than a nightmare.*

Had we originally started to design this system from the control end, many needless variables (i.e., following error and backlash) might have found their way into the equation. Since we are always applying the motion in the same direction, we never had to consider these variables. Also, the system encoder resolution would probably have been tighter, requiring an updating approach rather than a *real-time* approach. The updating approach for position counting ***is not*** more accurate than *real-time* methods in this machine. The *real-time* approach will give us increased control by being able to monitor the machine (and do I/O handling) on every encoder count (0.00375 inches of movement). Think about it.

Notice that throughout this example, we avoided the temptation to include elements of the design that have no direct bearing on the problem definition (such as switches, buttons, lights, and computer). Consequently, we could concentrate strictly on the problem.

We must concern ourselves with *operation* first. As we think about other operational items (like safety), we simply note them down so as not to disturb the current design direction. As the design develops, the control, software, and interaction with the mechanics will arrive at a point that will insure first time startup success.

The design flow to this point has generated the following information:

- System resolution = 0.00375" per count.
- Encoder style: pulse and direction.
- *Real-time* software interrupt counting.
- 150 microseconds maximum interrupt time period.
- 7.5" accel/decel profiles.
- 7.5" limit over-travel.

The next design objective is to get the product and machine weights to profile the system for motor and motor amplifier selection. The control can also be selected at this point, since we know how the software will function. With the problem now under control, we can begin listing the items required to make the system functionally complete:

- Operator entry terminal
- Switches, buttons, I/O
- Type of computer
- Motors, drives
- Power supplies
- Cabinets

Discussion

The above example described the actual design methodology used to develop a generic angle/carriage saw system currently operating in over 30 locations around the world. These systems range in cost from $100,000 for a single axis angle saw, to $500,000 for a four-axis double cutter carriage saw. All of these systems are controlled by an 8052 microprocessor and use the same software control. Two types of data entry panels were used to interface the operator to the system. One is a *Transterm* model TT6-00 two-line LCD display with membrane keyboard and the other a *Xycom* 14-inch RGB color monitor with membrane keypad. A screen handling program was written for each and can be selected with a board-level jumper.

The systems are capable of cutting material at speeds up to 200 fpm (the maximum capability of the software was measured at 350 fpm.) with repeatability better than 1/16 inch. All of the systems use a base operating resolution between 0.002 and 0.00488 inches-per-encoder-count (the resolution is software configurable). The units were designed to cut in a length mode, seam mode, FIFO mode, and remote DNC drop via RS232/RS485 communication link. The reliability of each system was engineered to have less than one 8-hour down period per year (2080 hours), but to date, *none* have had any electrical failures in over five years of 16 hour-a-day operation (over 20800 electrical on-line hours per machine).

In conclusion, putting thought into the product up front saved time, money, and multiple versions of the same control/software idea. Remember:

Defining the problem is not magic . . . it's necessary.

Good problem definition reduces design effort, simplifies design method, and allows you to look more closely at what is important. The integration of software, control, movers, and mechanics will flow if you do your homework.

Notes: