NISTIR 4832



# ALTERNATIVE DESIGNS OF A REAL-TIME ERROR CORRECTOR FOR MACHINE-TOOLS WITH "ENCODER" POSITION FEEDBACK

Kenneth W. Yee

U.S. DEPARTMENT OF COMMERCE
Technology Administration
National Institute of Standards and Technology
Manufacturing Engineering Laboratory
Automated Production Technology Division
Gaithersburg, MD 20899



NIST

NISTIR 4832

# ALTERNATIVE DESIGNS OF A REAL-TIME ERROR CORRECTOR FOR MACHINE-TOOLS WITH "ENCODER" POSITION FEEDBACK

**Kenneth W. Yee**

U.S. DEPARTMENT OF COMMERCE
Technology Administration
National Institute of Standards
and Technology
Manufacturing Engineering Laboratory
Automated Production Technology Division
Gaithersburg, MD 20899

April 1992

**U.S. DEPARTMENT OF COMMERCE**
**Barbara Hackman Franklin, Secretary**

**TECHNOLOGY ADMINISTRATION**
**Robert M. White, Under Secretary for Technology**

**NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY**
**John W. Lyons, Director**

## Introduction

As a part of a Quality in Automation project [1,2,3], the Manufacturing Engineering Laboratory at the National Institute of Standards and Technology (formerly the National Bureau of Standards) has implemented fast part-probing and real-time error compensation on a computer-numerical-control (CNC) machine tool using the existing machine-tool controller. Many machine-tool errors are repeatable and predictable. Therefore, we can compensate for these systematic errors during the process, reducing the errors in the finished part [4,5,6]. Machine-tool errors may be fixed geometric errors, such as lead-screw nonlinearity and straightness errors in the ways, or thermally induced errors such as expansion and distortion of the machine bed and lead-screw, or spindle-head growth. Error compensation allows parts to be produced with errors approaching the repeatability limits of the machine tool, throughout the operating temperature range, even during the thermal warm-up period. Fast part-probing is on-machine probing with a touch-trigger probe using a probing feed rate of 2.5 m/min which is 20-times faster than the rate commonly used. This saves time and allows more on-machine part data to be taken without excessive waste of machine time.

We have inserted a *black box* called the Real-Time Error Corrector (RTEC) between the position feedback elements of the axes of the machine tool and the machine-tool controller (MTC) as shown in figure 1. This device independently and simultaneously counts the signals from the *encoder-type* feedback element for each axis to produce the machine position and also includes a parallel interface to send this position data to an IBM compatible PC-type computer which calculates the required error-compensation for each axis as a function of current position and temperature and records the axes' positions at probe trip-points. The device allows (in a way that is invisible to the machine tool) real-time correction of positions of the machine-tool axes and fast part-probing without intrusion into the MTC.

The RTEC is simply *plugged-in* between the position feedback device and the MTC. A minimum of hardware and software is MTC-specific. Fast probing is facilitated by high-speed capture of the axes' positions in response to the probe trip. The same circuitry that is necessary for compensation is used, therefore the fast-probing capability is a *cost-free* feature of the RTEC. Compensation is implemented by altering the position feed-back signals passed on to the MTC. The MTC position count is changed which *tricks* the machine in to going to a slightly different physical position than normal. The amount the tool-path is altered is precisely controlled according to the number of correction counts calculated for each axis by the PC for the current positions and machine temperature. The first RTEC design has been previously described [7]. Some further details of the first design, the "Original RTEC," will be given and alternative designs proposed. Some implementation and application considerations will be discussed.

## Position Feedback Signals

Machine-tool position feedback has traditionally been obtained from rotary resolvers or encoders coupled to the ball-screws which move the carriage or table. Resolvers produce two sinusoidal waveforms which change in relative phase by 360° for one revolution. Resolvers are not commonly used in *modern* machines and will not be addressed further. Encoders produce two square-wave outputs (A and B shown in figure 2a) that are displaced in phase by 90° (one-quarter cycle). A rotary encoder typically produces 1000 to 10 000 cycles for one revolution. The signal which leads in phase changes with the direction of motion, i.e., A leads B for positive axis motion. The position resolution is also a function of the ballscrew pitch and the gear ratio between the two. Other position feedback devices produce signals which emulate a resolver or encoder. The trend today is to use high-resolution encoders or glass scales with *encoder-type* signal conditioning. A quadrature (quad-) decoder produces a signal from A&B which is counted (and a direction signal) to produce the axis position with four times the resolution of the encoder square wave. Submicrometer resolution is now common for machine tools. For each axis, the *encoder* resolution, length of travel and the maximum velocity must be known to design the RTEC.

## Altering the Position Signal

A simplified block diagram of an RTEC is shown in figure 3. A commercially available quad-decoder and single-chip microcomputer (microcontroller) are used for each machine axis. The microcomputer sends the total position count for the axis to the PC through a parallel interface. Twenty bits, for example, are required for a one-meter long axis with one micrometer resolution--a million counts. The size of the position counters and/or the parallel I/O interface for each axis is one of the few hardware design parameters which must be tailored to the specific machine-tool design. (Compatibility with the encoder/MTC electrical interface must be ensured. Differential line-driver circuits are commonly used, though single-ended logic levels may be used.) The microcomputer also alters the position feedback signal passed through to the MTC according to the correction commands from the PC. In the probing mode, the axis position at probe trip is sent to the PC. Compensation is not done in real time during probing. The probe-trip positions can be corrected by the amount of the predicted errors off line.

Adding a pair of pulses which overlap by half their width, as shown in figure 2b, with A leading B will cause the axis position count in the MTC to increase by four (for the usual sign convention). This means the MTC believes the axis position is four counts further in the positive direction of motion than it really is. Therefore, the axis will stop a distance corresponding to four counts in the negative direction from the programmed position. If a pair of encoder pulses were deleted as shown in figure 2c, the axis would stop four counts in the positive direction from the programmed position. The signals can, of course, be altered while the axis is in motion and the MTC will immediately alter the path from the programmed positions.

Since the encoder-signal frequency inherently varies from zero, when the axis is not moving, to some maximum frequency depending on the resolution and the maximum velocity, quad-decoders are designed to respond only to the state (high or low) and the order of transitions of the two signals A&B (i.e., they are independent of time or frequency of the signals). This characteristic makes it possible to subtract counts from the MTC position by **adding** a pair of pulses with B leading A (the opposite phase from figure 2b). This simplifies the implementation algorithms which will be discussed later. A pair of narrow pulses, with the phase chosen depending on addition or subtraction of counts, can be inserted whether the axis is moving slow or fast. The velocity, along with the resolution, determines the square-wave frequency (i.e., the width of the encoder pulses). The added pulses must not be too narrow for the quad-decoders in the MTC to respond. Decoders are typically designed for maximum square-wave frequencies of 100 kHz to 1 MHz which look like strings of 5 $\mu$s to 0.5 $\mu$s wide pulses. The "Original RTEC" design used 2 $\mu$s wide pulses which is the fastest the (12 MHz) 8051 family microcomputer used can produce. The demonstration turning-center has glass scales and a position resolution of one micrometer. For this machine, the rapid traverse rate of 7 m/min, produces a maximum *encoder* output frequency of about 30 kHz. Rapid traverse rates twice this and position resolutions approaching one-tenth this value, which are on the horizon, will produce signals up to 500 kHz. The speed of the microcomputer and quad-decoder must be chosen based on the highest frequency expected.

## The "Original RTEC"

The "Original RTEC" design uses several restrictions to simplify the implementation. Short fixed-width pulses are always added to the A&B signals. The phase is chosen depending on whether addition or subtraction is required. Pulses are added only when both square waves, A&B, are in the low state. By adding short pulses, the job of inserting a correction count is over in a short, finite time. Deleting pulses is time dependent, varying with axis velocity. If a pulse is deleted as the axis is coming to a stop, an ambiguity results. Under this condition, the microcomputer could be waiting *forever* for the pulse to end. Implementing subtraction by inserting pulses, with B leading A, eliminates this problem. Also, in this design the *encoder* square-wave signals from the position feedback are not actually passed through the microcomputer, as shown in figure 3--they pass through a logical "OR" circuit which allows the microcomputer to add pulses to the string through another input. Except when inserting pulses, the microcomputer does not do anything to the A&B signals. The signals just go through the "OR" circuit to the MTC. At other times, the microcomputer is free to do other tasks such as read the quad-decoder output and send the axis position count to the PC in real time. These restrictions simplify the algorithm and reduce the speed requirements of the microcomputer. Even so, it is not as fast as required and additional compromises had to be made.

The 8051 microcomputer can output a pair of two-microsecond wide pulses which overlap by half. However, there is overhead time to get ready before they actually appear. A hardware "AND" circuit is used to determine the time when A&B are "low." This reduces the decisions made by the microcomputer and saves time. Even so, there is a minimum 4-$\mu$s delay from recognizing the "A&B low" signal until a pulse can be added to A or B, 3 $\mu$s for the pulses and 2 $\mu$s after the pulses which is a total of 9 $\mu$s. At 30 kHz, A&B are low for only about 8 $\mu$s so the routine may run into the next signal transition of A or B. Fortunately, it is not necessary to input pulses when the machine is at rapid traverse since practical cutting speeds are at least an order of magnitude slower, i.e., errors cutting air are not errors in the part. Since there is a ramp down in feed rate at the end of rapid moves and a cutting feed-rate move in air before the tool engages the part, there is ample time to update the correction counts before cutting begins. The time between "A&B low" signals is timed by the microcomputer to assure that the machine is moving slowly enough to insert pulses. The pulses are always inserted immediately after a transition in A or B. This ensures that the inserted pulses will not overlap a *real* encoder transition. An alternative is to inhibit the real transitions during the period a pulse is being inserted, perhaps using 3-state or latch logic circuits.

On any machine worthy of error compensation, the errors are small and slowly changing. Very few correction counts are necessary over a practical cutting path. It would be unusual to find a machine with very large yet repeatable errors. The maximum error for a one-meter long axis might be 240 $\mu$m. If the error is linear, this is only 4 $\mu$m (one correction count for the demonstration machine) every 16 700 $\mu$m of axis travel. Parts made on this machine are likely to only use one-quarter of the axis travel or less. Thus only 15 correction counts might be required over the length of a part. The algorithm has been written so that the PC sends the RTEC the total number of correction counts required for the current axes positions and machine temperature. It is unimportant if the changes in correction during rapid moves are not implemented as long as the RTEC catches-up before the tool enters the part.

This "Original RTEC" method has some limitations. The correction resolution is four times the position resolution of the machine since "full" encoder cycles are always inserted. The machine cannot be corrected when it is not moving unless it happens to stop with both signals A&B in the low state, for the axes that need correction, or the axis-position servos have some instability that results in dither (slight instability in position) that causes this state. A *good* position servo might dither only one count which may not include the A&B low state. The same is true for a machine move where an axis has a fixed programmed position but may require corrections on that axis. This condition is common for turning and facing cuts on a turning center, or milling rectangles (aligned with the machine axes) on a machining center. The servo gain or some amplifier zero-offset might intentionally be altered to cause the required dither. The correction resolution is inherently four-times the position resolution since full *encoder cycles* are added.

The advantage of this "Original RTEC" design is that the microcomputer needs to work *fast*, only when inserting pulses and does nothing to the A&B signals which simply pass though the "OR" circuit when no correction is taking place. The microcomputer has time to read the quad-decoder counts and keep track of *revolutions* to keep total count of the axis travel from machine home.

## Alternate RTEC "Design A"

There are alternative RTEC designs or algorithms which have various tradeoffs in performance limitations and microcomputer speed requirements. The simplest change, "Design A", still does not pass the encoder signals through the microcomputer, yet eliminates the requirement that pulses simulating full encoder cycles be inserted only when A&B are low. The added *pulses* are still inserted in a hardware "OR" or "AND" circuit as in the "Original RTEC." There are only four combinations of states of the A&B signals (each can only be high and low). *Encoder* cycles can be added even if the starting point is not A&B low. This means that instead of always inserting two positive-going pulses overlapping by half their width, the pulses may be of either polarity, but would still overlap by half.

The signals are defined by a state table:

| State | A | B |
|---|---|---|
| 1 | 0 | 0 |
| 2 | 1 | 0 |
| 3 | 1 | 1 |
| 4 | 0 | 1 |

Only one signal changes per state which is called a Gray code. Positive axis motion (figure 2a) moves forward in the state table, while negative axis motion (figure 2d) moves backwards in the state table. The "Original RTEC" was built with the simplest assumption that you make any correction in state 1 where A&B are zero. A positive pulse is a change from zero to one and back to zero. The state table shows that A and B overlap by half, assuming equal time in each state. Moving backwards in the table, B goes to a logical "1" or high signal-level before A (B leads A). If the starting state is not zero-zero, the *pulses* for addition are changes in state through the next three states and back to the starting state, whatever polarity each state may be. Similarly, for subtraction, they are changes backwards in the table through the next three states ending at the starting state. This design eliminates the problem of not being able to insert corrections if the axis does not pass through the A&B low state. It is still limited by the correction resolution being four-times the position resolution. The microcomputer again only has to work *fast* during the insertion of *pulses* and can do other tasks. It takes the same time for the microcomputer to insert the encoder cycles as in the "Original RTEC" design. The state of A&B (which of the 4 possibilities) must be determined. This is likely to take additional overhead time, regardless of whether the determination is done by hardware or software. The speed of the axis must be still checked. Pulses cannot be inserted if the signals are too fast, and some means must ensure that a real transition does not occur during the insertion process.

## Alternate RTEC "Design B"

"Design B" is more flexible because it passes the A and B signals through the microcomputer (as shown in figure 3) with the capability of altering them at will by any multiple of the position resolution. Again the changes in A&B are only a step forward or backwards in the state table, but they may now be increments of only one step rather than four. The microcomputer keeps track of the total counts added or subtracted which can be in the hundreds. A position count and a correction count are now the same size. The disadvantage is that the microcomputer must be significantly faster since, even if no corrections are made, it must read and write the states of A and B just to pass the *encoder* signals through to the MTC. The circuit can never make a mistake or the MTC position count will be accidentally changed or the MTC may recognize an illegal state and generate a fatal error, typically a *servo fault* message. The inserted counts (transitions) would still have to be done at a time that ensures that a real count will not overlap.

For *encoder* square-wave frequencies of interest, a microcomputer at least an order of magnitude faster than the 8051 (which has a one-microsecond simple-instruction execution time) is required. A 100-kHz *encoder* signal which is moderate by today's standards has a transition in A or B every 2.5 $\mu$s. Current generation digital signal processors (DSPs) have the speed capability to handle these signals. Even though they are primarily designed to do fast computations, the faster cycle-times allow faster input and output (I/O). The disadvantage of DSPs when compared with the 8051 microcomputer which has instructions to operate on I/O bits, is that I/O, like most computers, is by 8-bit byte. This means that additional masking instructions are required to isolate changes in a single bit. Commonly available DSP cards are not intended for intensive parallel I/O but for computation. A high-speed RTEC of "Design B" would require a custom designed circuit with the required parallel I/O, for both the A&B *encoder* signals and the parallel axis-position count.

Newer quad-decoders have higher-resolution counters internally, or cascadeable external counters that do not require the microcomputer to total the *revolutions*. The later models of those used in the "Original RTEC" (which had only 12 bits), however still require software reading of two 8-bit bytes for each reading which must be done by the microcomputer. If the A&B signals are passed through the microcomputer, it may not even be able to stop to service the low bytes of the position output. An additional microcomputer for all axes in the RTEC might be required.

## Alternate RTEC "Design C"

Another possibility, "Design C," eliminates some of the disadvantages of the "Original RTEC" design and has some of the advantages of "Design B." This design does not pass the A&B signals through the microcomputer, but uses digital inverters rather than "OR" circuits to alter the signals before they go to the MTC. From the state table, it is obvious that the change from one state to the next is caused by the inversion of one of the signals, since a change from low-to-high or high-to-low corresponds to a 0-to-1 or 1-to-0 change. Therefore, steps in the state table can be inserted by inverting one of the signals. Further examination shows that if both A&B are inverted, the signals will then pass through with the same A-leading-B or B-leading-A phase relationship as the *encoder* output, i.e., the microcomputer need not do anything to pass the signals through.

The following state table is the same as shown before with the position count included:

| State | A | A | Count |
|---|---|---|---|
| 1 | 0 | 0 | 0 |
| 2 | 1 | 0 | 1 |
| 3 | 1 | 1 | 2 |
| 4 | 0 | 1 | 3 |
| 5(1) | 0 | 0 | 4 |

When the signals reach the fifth state, which is the same as the first, the count is 4. If, while A&B are in the "0,0" state, we invert "A" and then invert "B" we have the following:

| State | A | B | Count | Comment |
|---|---|---|---|---|
| 1 | A | B | 2 | |
| | 1 | 0 | 4 | Invert A |
| | 1 | 1 | 2 | Invert B |
| (2) | 0 | 1 | 3 | 2 inverted |
| (3) | 0 | 0 | 4 | 3 inverted |
| (4) | 1 | 0 | 5 | 4 inverted |
| (5) | 1 | 1 | 6 | 5 inverted |

We have added two extra states which are forward moves in the original state table. This results in two counts being added. If B were inverted before A, we would find that two counts are subtracted.

Now if we should invert "A," then invert "B" while A&B are in the "1,0" state we have the following:

| State | A | B | Count | Comment |
|---|---|---|---|---|
| 1 | 0 | 0 | 0 | |
| 2 | 1 | 0 | 1 | |
| | 0 | 0 | 0 | Invert A |
| | 0 | 1 | -1 | Invert B |
| (3) | 0 | 0 | 0 | 3 inverted |
| (4) | 1 | 0 | 1 | 4 inverted |
| (5) | 1 | 1 | 2 | 5 inverted |

Each inversion result in a step backward in the original state table which subtracts two counts, but the inverted signals following result in addition. The general algorithm is that if A&B are the same, invert A then B to add two counts, or invert B then A to subtract two counts. However, if A&B are different, invert B then A to add two counts, or invert A then B to subtract two counts.

With "Design C" we now can add or subtract in increments of two position counts while the "Original RTEC" was limited to increments of four. It is possible to insert counts in any A&B state, however what must be done to add or subtract is a function of the state. The resolution is not as fine for as "Design B," however, the major advantage is that the signals do not pass through the microcomputer so very-high processor speed is not required. Except when incrementing counts, the microcomputer is free to read the quad-decoder and send position to the PC. The inversions can readily be implemented by switching an inverter circuit into or out of the A&B circuits. This can be done with 3-state logic circuits.

Actually, each of the four possible states do not have to be recognized. Only if A&B are at the same or different levels must be determined. This can be implemented in hardware by an "exclusive OR" circuit. Like the "Original RTEC," a decision must be made as to which order to output the A&B changes in order to add or subtract. Also, the period of the "exclusive OR" output can be monitored to ensure that there is time to complete the inversions. Somewhat less time is required to complete the insertion since the microcomputer must only output an invert command not a pulse. Changes should be made immediately after a transition of the "exclusive OR" to ensure that a real transition does not overlap. This design can be implemented with the 8051 microcomputer used in the "Original RTEC."

## Alternate Configurations

The RTEC need not be a stand-alone *black box* like the "Original RTEC." The required circuitry could readily be built on a PC computer card. Of course, the high-speed microcomputer and quad-decoder are still required for each axis. The computation required to determine the correction counts [4,5,6] is moderate and no challenge for 80286 or 80386 based PCs. Since the errors are small and slowly changing, for most purposes a new correction value every 1-2 mm of axis travel is sufficient. For practical cutting speeds, a value every 10-20 ms is sufficient. The position values and correction counts could now be passed directly via the PC bus, eliminating the parallel I/O interface.

## General Considerations

Regardless of the RTEC design implemented, every precaution must be taken to avoid mistakes. The errors that can be introduced into the machining process are **enormous**, while the errors that can be compensated are **small!** Signal integrity and data integrity must be ensured in all possible ways. In the "Original RTEC," many hand-shaking signals, data checks and limitations are employed. Additionally, critical hardware to pass the A&B encoder signals through to the MTC is powered from the MTC. Failure of the PC and other parts of the RTEC cannot cause a machine-tool crash. This option does not exist for "Design B." The machine tool controller must be relied on to fail gracefully with loss of position feedback.

In the "Original RTEC," the interface to the PC's parallel I/O is completely asynchronous, i.e.,each runs at its own pace. For each axis there is a "Axis Position" word and "Correction Completed" byte to the PC. A "Correction Requested" byte and "Status" byte are received from the PC. The length of the axis-position word and the numerical convention (signed or unsigned) were chosen depending on the length of each machine axis and the direction of motion from machine home. A generic version with a *large* number of signed bits for all axes could be designed. "Data Valid" bits are included to ensure that readings are not taken while data is being updated. The "Correction Requested" is the total correction count required for the current axis position and temperature. This avoids the possibility of getting lost which might happen if incremental counts were passed. The RTEC reports back the "Corrections Completed" (correction counts inserted) as soon as implemented. The job is done when the requested and completed counts agree.

For the "Original RTEC" on the demonstration turning center, the corrections counts are in one signed byte which allows $\pm 127$ counts of 4 $\mu$m or $\pm 508$ $\mu$m ($\pm 20$ mil) which is more than the expected correction for any axis. However, an additional safe guard is used since a data glitch could cause a 500 $\mu$m error in the part. Since the machine errors are small and slowly changing, a window is placed on the incremental change allowed in the "Corrections Requested," e.g., $\pm 3$ counts since the normal increment expected is $\pm 1$. If the increment is greater, the RTEC does not implement the request and issues an error message that means "I Don't Believe You." If the "Corrections Requested" returns within the allowable range in the next two readings, they will be accepted. Otherwise, the RTEC will issue a fatal error message "I Still Don't Believe You" which will cause the PC to "Hold" the machine tool and issue a warning to the operator. For cases where a larger correction increment is required (e.g., at restart with a hot machine), a "Large Correction" signal bit from the PC tells the RTEC to accept the increment.

The start-up procedure must be carefully controlled. With the machine axes at a predetermined position, e.g., machine home, the quad-decoder counters including any count kept by the microcomputer must be zeroed. The PC program should not allow anything further until it has received the expected zero or home position values from the RTEC.

The "Original RTEC" also included a hardware switch to prevent any corrections from being inserted. This ensures that the machine position is unaltered from its own performance. An error message is sent from the RTEC if a nonzero "Correction Requested" is received from the PC while the corrections are disabled.

In the probing mode, which is signaled by the PC to the RTEC via the "Status" word, real-time error compensation is not implemented. There are two reasons for this. The PC and microcomputer in the "Original RTEC" are marginally fast enough to implement compensation at the 2.5 m/min feed rate used in fast probing. The main reason is that if there is an error in the compensation system, it would repeat in the probing and not be *seen* until the part was inspected off-line. Compensation for the geometric-thermal errors can be applied to the probing data, off-line, before the data is used to determine the deviations from the nominal part dimensions.

## The "Right Way"

The right way to implement both error compensation and fast probing is in the MTC itself. External *black boxes* or cards are a necessary evil since these capabilities do not generally exist in commercial MTCs on the market today. A few MTCs are known to have the capability of fast probing. This capability requires that the probe-trip initiate the capture of the axes' positions in hardware, as in the "Original RTEC" or by computer interrupt rather than as part of a polling loop. A response in microseconds is required rather than in milliseconds.

The only error compensation designed into all commercial MTCs is lead-screw compensation. This system only corrects for room-temperature geometric errors in position (linear displacement error) along the axis travel divided into some number of segments. An additional correction parameter for each axis needs to be included in the position loop calculations. The error computation as a function of position and temperature might require an additional processor in the MTC. This built-in compensation capability is on the horizon.

## References


1. "Progress Report of the Quality in Automation Project for FY88," C. D. Lovett ed., NISTIR 89-4045, National Institute of Standards and Technology, Gaithersburg, MD., April 1989.

2. "Progress Report of the Quality in Automation Project for FY89," T. V. Vorburger, B. Scace eds., NISTIR 4322, National Institute of Standards and Technology, Gaithersburg, MD., May 1990.

3. "Progress Report of the Quality in Automation Project for FY90," M. A. Donmez ed., NISTIR 4536, National Institute of Standards and Technology, Gaithersburg, MD, May 1990.

4. Donmez, M. A., Blomquist, D. S., Hocken, R. J., Liu, C. R., and Barash, M. M., "A General Methodology for Machine Tool Accuracy Enhancement by Error Compensation," Prec. Eng., Vol. 8, No. 4, pp 187-196, 1986.

5. Donmez, M. A., Liu, C. R., and Barash, M. M., "A Generalized Mathematical Model for Machine Tool Errors," Modeling, Sensing, and Control of Manufacturing Processes, K. Srinivasan ed., The American Society of Mechanical Engineers, New York.

6. "A Real-Time Error Compensation System for a Computerized Numerical Control Turning Center," Donmez, M. A., Lee, K., Liu, C. R., and Barash, M. M., Proc. IEEE Int. Conf. on Robotics and Automation, San Francisco CA April 1986.

7. "Implementing Fast Part Probing and Error Compensation on Machine Tools," Yee, K. W., and Gavin, R. J., NISTIR 4447, National Institute of Standards and Technology, Gaithersburg, MD, October, 1990.

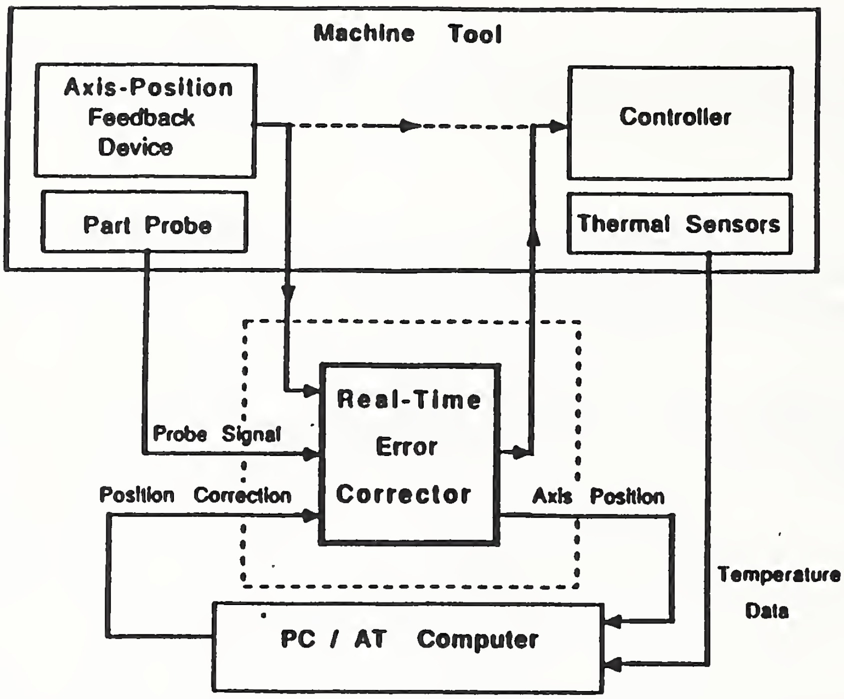


Figure 1. System Overview

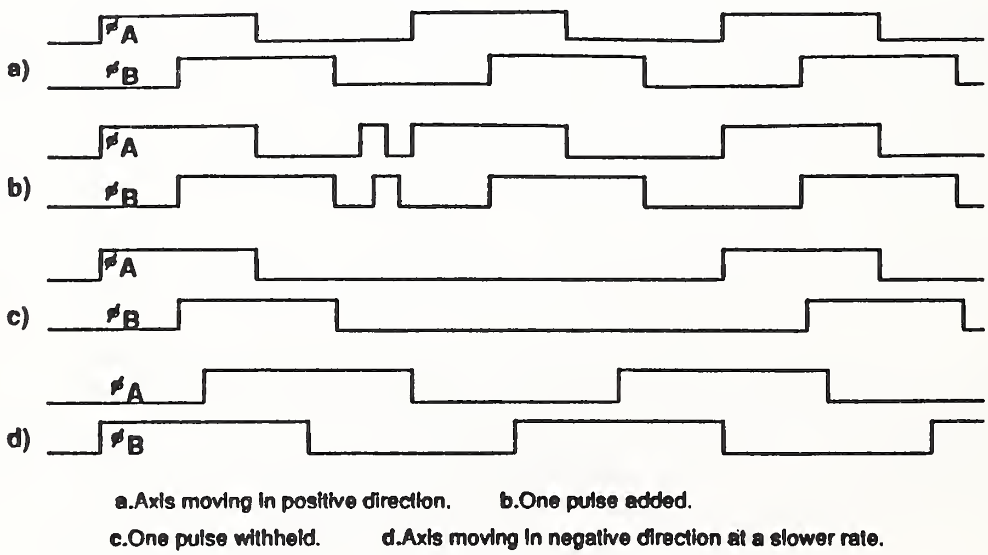


Figure 2. Encoder Signals

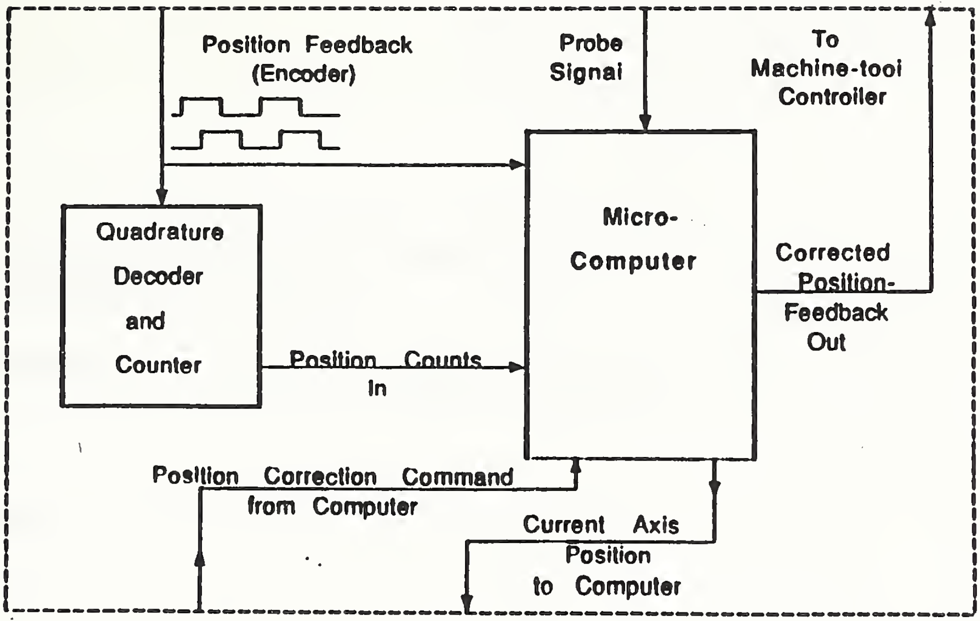


Figure 3. Generic Real-Time Error Corrector

NIST-114A
(REV. 3-90)

U.S. DEPARTMENT OF COMMERCE
NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY

# BIBLIOGRAPHIC DATA SHEET

1. PUBLICATION OR REPORT NUMBER
NISTIR 4832

2. PERFORMING ORGANIZATION REPORT NUMBER

3. PUBLICATION DATE
APRIL 1992

4. TITLE AND SUBTITLE
Alternative Designs of a Real-Time Error Corrector for Machine Tools with "Encoder" Position Feedback

5. AUTHOR(S)
Kenneth W. Yee

6. PERFORMING ORGANIZATION (IF JOINT OR OTHER THAN NIST, SEE INSTRUCTIONS)
U.S. DEPARTMENT OF COMMERCE
NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY
GAITHERSBURG, MD 20899

7. CONTRACT/GRANT NUMBER

8. TYPE OF REPORT AND PERIOD COVERED

9. SPONSORING ORGANIZATION NAME AND COMPLETE ADDRESS (STREET, CITY, STATE, ZIP)

10. SUPPLEMENTARY NOTES

11. ABSTRACT (A 200-WORD OR LESS FACTUAL SUMMARY OF MOST SIGNIFICANT INFORMATION. IF DOCUMENT INCLUDES A SIGNIFICANT BIBLIOGRAPHY OR LITERATURE SURVEY, MENTION IT HERE.)

As a part of a Quality in Automation project, we have implemented fast part-probing and real-time error compensation on a computer-numerical-control (CNC) machine tool using the existing machine-tool controller. We have inserted a black box called the Real-Time Error Corrector (RTEC) between the "encoder" position feedback elements of the axes of the machine tool and the machine-tool controller. Further details of the "Original RTEC" will be given and three alternative designs proposed. Some implementations and application considerations are discussed.

12. KEY WORDS (6 TO 12 ENTRIES; ALPHABETICAL ORDER; CAPITALIZE ONLY PROPER NAMES; AND SEPARATE KEY WORDS BY SEMICOLONS)
compensation; correction; encoder; error; geometric; machine tool; thermal

13. AVAILABILITY

- [X] UNLIMITED
- [ ] FOR OFFICIAL DISTRIBUTION. DO NOT RELEASE TO NATIONAL TECHNICAL INFORMATION SERVICE (NTIS).
- [ ] ORDER FROM SUPERINTENDENT OF DOCUMENTS, U.S. GOVERNMENT PRINTING OFFICE, WASHINGTON, DC 20402.
- [X] ORDER FROM NATIONAL TECHNICAL INFORMATION SERVICE (NTIS), SPRINGFIELD, VA 22161.

14. NUMBER OF PRINTED PAGES
16

15. PRICE
A02

ELECTRONIC FORM