A Mechatronics Approach to Design a General CNC Controller

Omar. A. Al-Sharif [1]*, Abderady Okasha[1] , Mohamed Shaheen[1]

1 Department of Mechatronics Engineering, Faculty of Engineering, October 6 University, 6th of October City, 12585, Giza, Egypt
* Email of Corresponding Author: omarahmed.eng@o6u.edu.eg

Abstract

This study introduces the design and implementation of a general CNC controller for 3-axis CNC and 3D printer machines using LabVIEW software. The controller features an interpreter module for translating G&M code into machine and control commands. This module processes NC code generated by CAM systems or manually by the user, extracting vital data such as motor velocities, positions, spindle speed, and coordinate modes for simulation and control purposes. Additionally, we developed a user-friendly man-machine interface (MMI) to improve usability and monitor machine operations and program execution. This research highlights the importance of integrating mechanical, electronic, and software systems, emphasizing the role of mechatronics system design in modern manufacturing.

Keywords: CNC controller, CAD/CAM, G code, Interpreter, Interpolator.

I. INTRODUCTION

Computer Numerical Control (CNC) machines are integral to manufacturing automation. These systems exemplify mechatronics, seamlessly integrating mechanical, electronic, and software components [1, 2]. CNC machines are programmed using G-Code, a Motion Control Command Language (MCCL) that guides the movement of tools in CNC machines, 3D printers, and robot arms [3, 4].

The concept of Numerical Control (NC) was first applied to machine tools in 1947, following World War II. The Massachusetts Institute of Technology's Servo Laboratory developed the first 3-axis milling machine in 1952. From 1960 to 1980, advancements in semiconductors, integrated circuits, and microprocessors led to the development of Softwired NC or CNC systems, which rely on computer and software technology. Today, the terms NC and CNC are often used interchangeably [5].

The digital manufacturing process, illustrated in Figure 1, comprises several steps. Computer-aided design (CAD) software creates 2D or 3D part geometries. Computer-Aided Process Planning (CAPP) determines the necessary machining parameters. At the same time, Computer-Aided Manufacturing (CAM) generates machine tool paths based on CAD and CAPP information [6, 7].

NC program processing employs two methods: interpretation and compilation. The interpretation method processes code line-by-line, sending corresponding commands to control loops. In contrast, the compilation method processes the entire program before execution. While interpretation offers simplicity and ease of implementation, compilation allows for efficient, error-free program execution [8].

Open interpreter and controller software can enhance CNC machine productivity by facilitating the integration of new functional modules, whether developed by end-users or acquired commercially [9]. This approach improves overall machine performance, efficiency, and user-friendliness. It also enables CNC machine tool builders to rapidly develop and integrate new functional modules without complex custom control functions.

The open architecture of such systems offers advantages in maintenance and upgrades. Unlike conventional CNC machines with closed architecture, which have limited upgradability and can only be maintained by manufacturers, open architecture CNC machines allow for easy maintenance and upgrades by end-users [8]. This reduces downtime and maintenance costs while extending the machine's operational life.

The benefits of open interpreter and controller software extend beyond CNC machines to the entire manufacturing industry. Open software CNC machines contribute to more flexible and cost-effective manufacturing processes by integrating new features and using standard components, resulting in increased efficiency, improved quality, and reduced production costs [25].

This research presents a general CNC controller designed using the Laboratory Virtual Instrument Engineering Workbench (LabVIEW). The controller is tested on a general three-axis CNC machine and subsequently implemented on 3D printers and 3-axis CNC machines. It offers users the flexibility to modify or add new machines by adjusting specifications such as motor type (stepper or DC), control loop type (open or closed), and mechanical drive (lead screw or pulley diameter). Additionally, users can choose to simulate the G-code program before execution or perform simultaneous simulation and implementation.

The paper is structured as follows: Section 2 provides an overview of the NC program structure. Section 3 discusses key mathematical operations in CNC, Digital Differential Analyzer (DDA) techniques, interpolation, and open and closed-loop servo systems. Section 4 presents the main CNC controller modules, and Section 5 concludes the paper

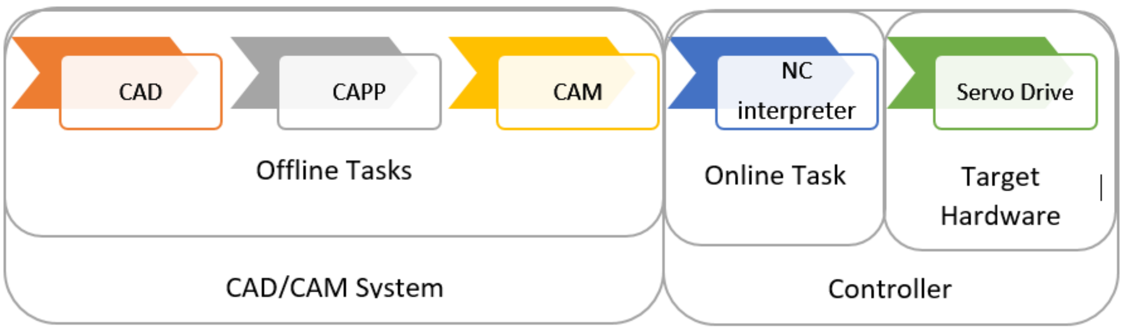

Figure 1. Digital manufacturing process

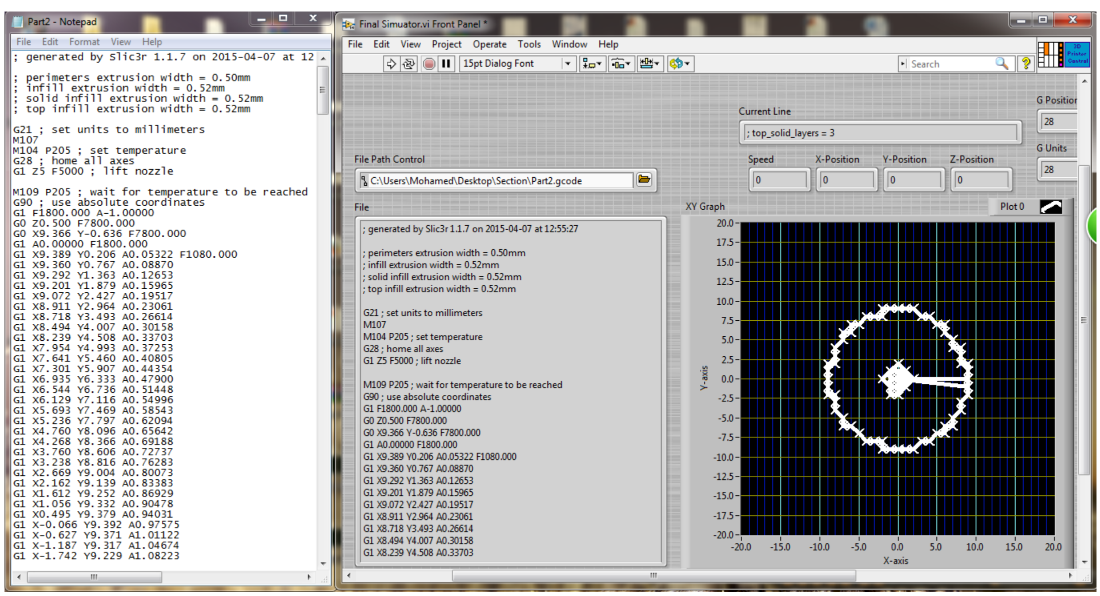

Figure 2. Man-machine interface

I. NC PROGRAM STRUCTURE

G-code or NC program interpreter in software that translates the NC program into a control command, therefore, we have to give a brief introduction to the NC program structure and contains. Most modern NC programs are organized in word address format, as shown in Figure 3. The word address format contains several different word commands to indicate various programming parameters. The NC program can differentiate between the other words and assign them to different addresses in the computer memory [9].

Word	Word	Word	Word
Address Number	Address Number	Address Number	Address Number
N 10	G 01	X -30	S 2000
Block			

Figure 3. NC program block

NC programs use the following code words in programming block statements:

- Preparatory function codes or G codes are any word in an NC program that begins with the capital letter G. Generally, G codes are used to program machine operational modes such as coordinate systems, interpolation.
- Coordinate values (X, Y, Z, a, b, c) to specify the desired motion of the machine tool relative to a workpiece.
- M codes or miscellaneous function codes are any word in an NC program that begins with the capital letter M. M codes carry miscellaneous information such as tool change, coolant, and spindle control.

The words in the block should be as organized and clear as possible [10]. Therefore, the words in a block should be arranged as shown in Figure 4.

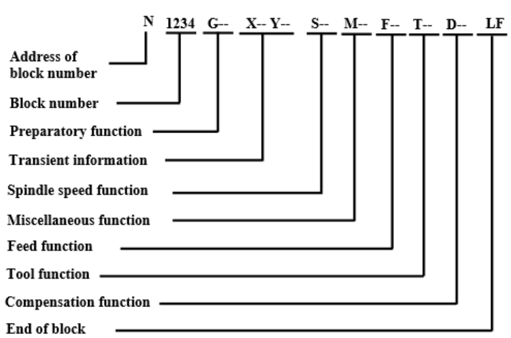

Figure 4. Word sequence in NC program block

II. TOOLPATH AND MOTION CONTROL

The main function of the CNC system is to generate coordinated movement for each driven axes-of-motion to achieve a desired cutting tool path relative to the workpiece [11]. Therefore, the most important and critical tasks in CNC machines are tool path and motion control, which includes interpolation, servo control, and drive of the motion axes [12].

A. Interpolation algorithms

Interpolation techniques are methods for estimating values between two known values. Interpolation can be either point-to-point (PTP) or contouring, and a CNC controller should support both types.

PTP interpolation is straightforward and involves providing the final position coordinates to the control system. In contrast, contour interpolation rasterizes contour lines of a segment map with a value domain and then cyclically computes motion set points. In contour interpolation, the machine path typically consists of linear and circular segments. At each segment, the interpolator provides the machine axes with the instantaneous position and speed for the control loops.

From a reference signal perspective, interpolation techniques can be classified as either reference pulse or reference word types. In pulse reference interpolation, the computer generates a sequence of reference pulses for each axis, with one pulse equating to one basic length unit (BLU). [11] The accumulated pulses represent the distance, and the frequency of the pulses represents the velocity, depending on the interpolation loop execution time. Conversely, reference word interpolation continuously provides velocity set points to the servo system [12].

1) Linear interpolation algorithm

Linear interpolation can be done by Digital differential analyzer (DDA) techniques. DDA is an algorithm for digital integration and generates pulse references varying in frequencies.

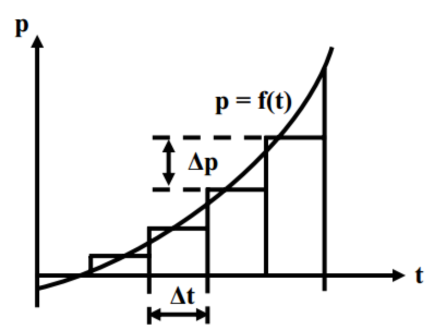

Figure 5. Backward Euler Integration

As shown in figure 5, using Euler approximation

$$z(t) = \int_0^t p\, dt \tag{1}$$

The value of z at $t = k\Delta t$ t is denoted byz_k, the digital form of Eq. (1) is

$$z_k = z_{k-1} + \Delta z_k \tag{2}$$

Where

$$\Delta z_k = p_k \Delta t \quad (3)$$

$$q_k = q_{k-1} + p_k \quad (4)$$

2) *Circular interpolation algorithm*

Circular interpolation, the simultaneous motion of two different axes, draws an arc, as shown in Figure 6, at a constant tangential velocity V [6]. The axial velocities can be given by the following equations:

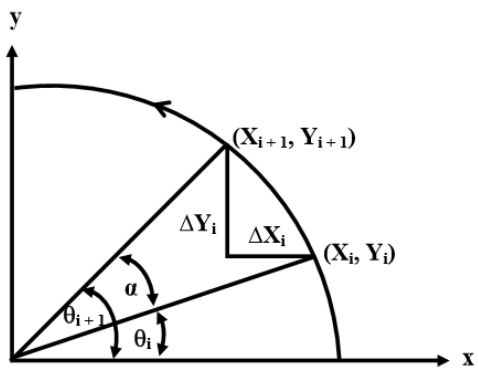

Figure 6. Circular interpolation

$$v_x(t) = v_0 \sin\theta(t) \quad (5)$$

$$v_y(t) = v_0 \cos\theta(t) \quad (6)$$

Where

$$\theta(t) = \frac{v_0 t}{R} \quad (7)$$

The velocities *Vx* and *Vy* are computed by the circular interpolator and supplied as reference inputs to the computer-closed loops. The interpolator converts the arc into a polygon inscribed on an arc or a pixel arc, with the resolution of the converted arc depending on the angle α, as shown in Figure 7. At each iteration, the interpolator calculates a new position and velocity for each axis and provides these values to the controller set points.

$$\cos\theta(i+1) = A\cos\theta(i) - B\sin\theta(i) \quad (8)$$

$$\sin\theta(i+1) = A\sin\theta(i) - B\cos\theta(i) \quad (9)$$

Where the coefficients A and B are given by

$$A = \cos\alpha \qquad B = \sin\alpha \quad (10)$$

$$\theta(i+1) = \theta(i) + \alpha \quad (11)$$

The corresponding segments are terminated at the point $X(i+1), Y(i+1)$ which is approximated by

$$X(i+1) = R(i)\cos\theta(i+1) \quad (12)$$

$$Y(i+1) = R(i)\sin\theta(i+1) \quad (13)$$

Substituting Eq(8) and Eq(9) into Eq (8) and Eq(9) yield:

$$X(i+1) = AX(i) - BY(i) \quad (14)$$

$$Y(i+1) = AY(i) + BY(i) \quad (15)$$

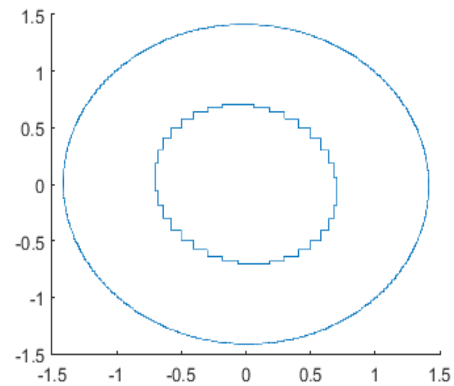

Figure 7. Simulated circular interpolation algorithm

Once the angle α is determined, the iterative interpolation process starts as the following:

At each point, X(i), y(i), the interpolator calculates the new points on x(i+1) and x(i+1) on x-y coordinates according to Eq (16) and Eq (17), the length segments given by the following two equations:

$$\Delta X(i) = X(i+1) - X(i) = (A-1)X(i) - BY(i) \quad (16)$$

$$\Delta Y(i) = Y(i+1) - Y(i) = (A-1)Y(i) + BX(i) \quad (17)$$

The corresponding velocities are

$$v_x(i) = K\Delta X(i) \quad (18)$$

$$v_y(i) = K\Delta Y(i) \quad (19)$$

Where $K = V/R\alpha$, the K is constant, which means calculated only once at the beginning of each interpolation process, the values given by Eq(16) and Eq(17) are the incremental position and those of Eq(18) and Eq(19) are the corresponding velocities for the new segment lengths.

B. Servo control

1) Stepper motors

A stepper motor is a pulse-driven motor that converts the angular position of the rotor into steps [8]. The stepper motor is usually an open loop control, as shown in figure 8. For this reason, stepper motors are a convenient option in many positions and speed control loops. In the case of using stepper motors, the NC controller generates step and direction; one pulse rotates the motor shaft on the step. The number of steps per revolution is supplied by the manufacturer, usually 200 step/ rev. The motor rotation angle can be given by α = 360/NPR, degrees where NPR is the number of pulses per revolution. If the motor receives "n" number of pulses, then the total angle θ is,

$$\theta = n\frac{360}{NPR} \quad (20)$$

The displacement x along an axis can be calculated by.

$$x = p\, t_n \frac{n}{NPR} \quad (21)$$

Where p is the pitch of the lead screw and t_n is the number of threads. Number of revolutions per minute N is given by,

$$N = \frac{60f}{NPR} \quad (22)$$

f is the pulse frequency. Linear velocity,

$$V = pN \quad (23)$$

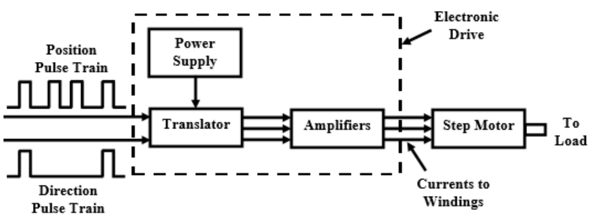

Figure 8. Open loop CNC system

2) DC Servo motor

DC motors offer a simple construction and are easy to control. The power supplied to the motor can be adjusted to regulate its speed. This adjustment can be done manually using a potentiometer or automatically using Pulse Width Modulation (PWM). However, unlike stepper motors, DC motors require a closed-loop system to minimize the error between the desired position (setpoint) and the actual position reached. As illustrated in Figure 9, the closed-loop system typically relies on feedback mechanisms to ensure accurate positioning.

Servo motors are automatic devices specifically designed for precise positioning control. They achieve this by incorporating a combination of components: a DC motor, a gearbox, a position sensor (often an encoder), and a control circuit. The control circuit continuously monitors the position of the motor shaft using the feedback from the sensor. Based on this information, it adjusts the direction and speed of the DC motor to reach the desired position with high accuracy. This makes servo motors ideal for applications where precise movement control is critical.

In situations where a CNC system encounters resistance to movement, such as during milling and turning operations, closed-loop control becomes essential. DC or AC servo motors are employed alongside feedback devices. This combination ensures that the machine tool or worktable reaches the intended position with the necessary precision.

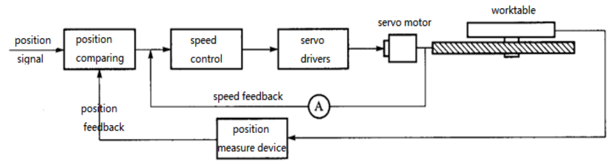

Figure 9. Closed loop CNC system.

C. Motion axis drive

Motion axis drives convert rotary motion into linear motion for various axes within a CNC machine. Two common types of motion axis drives are:

1) Power screw-based linear drive

These drives are widely used in CNC machines because they effectively convert rotary to linear motion. They consist of a screw spindle and a nut attached to the moving part (worktable or another axis) of the machine. The pitch (distance traveled per revolution) and number of threads on the power screw are crucial specifications that need to be provided to the NC controller for accurate positioning.

2) Belt drive

Belt drives offer a simpler and more cost-effective alternative to gears. They typically consist of an endless belt and two pulleys on parallel shafts. Various belt types exist, including flat, timing, and V-belts. Timing belts, such as small laser cutters, are commonly used in light-duty CNC machines. When using a belt drive, the user must specify the pulley diameter to the CNC controller program for proper axis movement calculations.

III. CNC CONTROLLER

The CNC controller serves as the brain of the digital manufacturing process. It comprises two main modules: hardware and software. The software module can be further divided into two sub-modules: interpreter and control loops.

The interpreter functions as a translator, converting NC program blocks (read from memory) into control commands for execution by either open-loop or closed-loop control systems. It processes the interpreted data by performing various mathematical operations, including interpolation, coordinate mode calculations, and other necessary functions (see Figure 10 for a visual representation of the main NC controller processes).

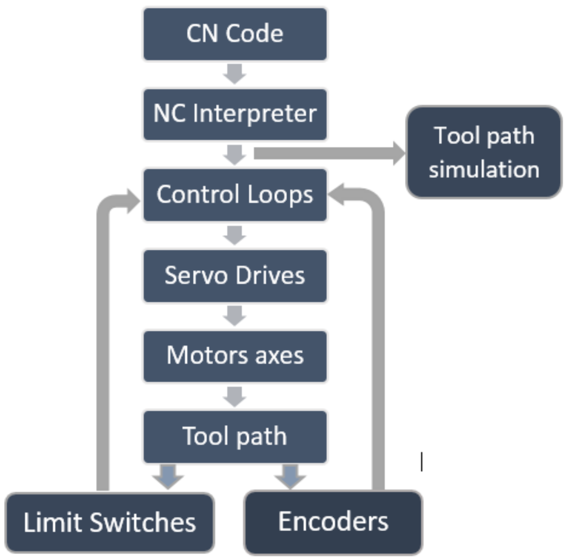

Figure 10. The NC controller processes.

The interpreter consists of several functions, or as they are called in LabVIEW, sub-VIs. The first is the input module, where the user can upload a text file containing the NC program to the NC controller by providing a file path or writing G-code blocks directly on the NC controller (NC program section), as shown in Figure 2.

The parser is the main module in the interpreter. It comprises both simple lexical scan and syntax analysis functions. It reads the NC program block by block, searches for and identifies predefined words and addresses such as G90, M, X, Y, etc., and executes each word and address sub-VI. If an error occurs during the interpretation process, the program halts and informs the user about the error and its location. Subsequently, the NC program exports all extracted data to the appropriate sub-VI to execute the required machining command. The output of this module sends control loop setpoints and other machining commands to the simulation sub-VI and/or the control loops.

The second module of the CNC controller consists of the control loops, which process the positions and velocities of the motors, spindle speed, and other commands and send appropriate control signals to motor drivers. The user must select either a stepper or a DC motor. For stepper motors, the CNC interpreter output signals are pulse and direction. For DC motors, the CNC controller sends PWM signals and receives encoder pulses to determine the corresponding machine tool position. In both cases, the user must provide the CNC controller with parameters such as BLU (Basic Length Unit) or the number of encoder pulses per revolution in the closed-loop case. Additional simple control loops, such as limit switches, can be connected to the computer through the CNC controller hardware interface or directly between the motor and motor driver to cut the connection when the limit switch is pressed.

IV. Conclusion

From a control perspective, the primary function of the CNC system is to generate coordinated movement for each driven axis to achieve the desired cutting tool path relative to the workpiece. The CNC controller has been designed and successfully implemented to meet these requirements on a 3D CNC machine. The controller consists of two main modules: the interpreter and the control loops.

The interpreter acts as a translator, transforming user-defined NC programs (generated by CAM software or written directly) into control commands for execution. Understanding the NC program structure was essential for designing the interpreter's functionality. Following this analysis, the interpreter module, including functionalities such as parsing, linear and circular interpolation, absolute and incremental movement handling, error management, and other relevant functions, was developed and coded using a graphical language for the simulation and execution of the NC program.

The second module of the CNC controller is the control loop, which processes the positions and velocities of the motors, spindle speed, and other commands from the interpreter and sends appropriate control signals to the motor drivers.

This research demonstrates the successful implementation of a comprehensive CNC controller that effectively bridges the gap between NC programming and physical machine control. The modular design of our controller allows for flexibility and adaptability, making it suitable for various CNC applications. Future work could focus on optimizing the controller's performance, expanding its compatibility with different types of CNC machines, and incorporating advanced features such as real-time error compensation and adaptive control strategies.

V. References

[1] Diegel, O., S. Singamneni, and A. Withell. "A Mechatronics Approach to Rapid Product Development: A Case Study." Mechatronics and Machine Vision in Practice, 2008. M2VIP 2008. 15th International Conference on. IEEE, 2008.

[2] Shetty, Devdas, Lou Manzione, and Ahad Ali. "Survey of Mechatronic Techniques in Modern Machine Design." Journal of Robotics 2012 (2012).

[3] Smid, Peter. CNC programming handbook: a comprehensive guide to practical CNC programming. Industrial Press Inc., 2003

[4] Xu, Xun. Integrating advanced computer-aided design, manufacturing, and numerical control: principles and implementations. Information Science Reference-Imprint of: IGI Publishing, 2009.

[5] Suh, Suk-Hwan, et al. Theory and design of CNC systems. Springer Science & Business Media, 2008.
[6] Mattson, Mike. CNC programming: principles and applications. Cengage Learning, 2009.
[7] Yusof, Yusri, and Kamran Latif. "Survey on computer-aided process planning." The International Journal of Advanced Manufacturing Technology 75.1-4 (2014): 77-89.
[8] Lai, Xiaoyan. "A design of general compiler for NC code in embedded NC system." Industrial Electronics and Applications (ICIEA), 2014 IEEE 9th Conference on. IEEE, 2014.
[9] Kibbe, Richard R., et al. Machine tool practices. Pearson, 2014.
[10] Krar, Steve, and Arthur Gill. Computer numerical control programming basics. Industrial Press Inc, 1999.
[11] Koren, Y., and O. Masory. "Reference-pulse circular interpolators for CNC systems." Journal of Engineering for Industry 103.1 (1981): 131-136.
[12] Koren, Y., and O. Masory. "Reference-word circular interpolators for CNC systems." ASME Journal of Engineering for Industry 104 (1982): 400-405.
[13] Zribi, M., and J. Chiasson. "Position control of a PM stepper motor by exact linearization." IEEE Transactions on Automatic Control 36.5 (1991): 620-625.