

## **Graphical User Interface Product Simulator for Motion Control of Machine Path**

**Teerasak Asvahem<sup>3</sup>, Claudio Campana<sup>2</sup>, Devdas Shetty<sup>1</sup>**

**1. Vernon D. Roosa Professor in Manufacturing Engineering**

**2. Research Engineer, Mechanical Engineering**

**3. Graduate Student, Mechanical Engineering**

**College of Engineering, University of Hartford, West Hartford, CT 06117, USA**

**Tel: 860 768 4615, Fax 860 768 5073**

### **Abstract:**

This paper examines the current status of machine tool control simulation and proposes an integrated methodology of motion control based on three dimensional real time simulator. This technique enables the development of Graphical User Interface (GUI) allowing engineers to design customized virtual instruments and test panels. The additional advantages are simulation and visual representation of the control providing flexibility for modification of the PLC motion control system. The paper also looks at the inadequacies of CNC systems and the needs in tool path verification and optimization. The major contribution of this paper is the demonstration of a system that has the ability to control a machine without conventional CNC-G Code. The key features of this method are the hardware in the loop simulation, interface with the data acquisition and control board, and reproduction of tool path simulation.

### **Introduction**

Advances in computer-aided engineering (CAE) have brought a new paradigm in the last decade so that current Computer-Aided Design systems (CAD) have the capability for sophisticated geometry and assembly of part models. The assembly can be visualized, evaluated, and modified as a whole, and its motion can be simulated. It could become a virtual evaluation tool for assessing product reliability and performance if its real time computational efficiency is improved. CAE is used in the design process to guide design decisions, saving valuable time and costs associated with redesign. These trends in engineering have converged to a new engineering concept: virtual engineering. In essence, virtual engineering is simulation-based engineering. Virtual engineering is a simulation-based method to help the designer and engineer to make decisions and establish controls. The virtual environment is a computational framework in which the geometric and physical properties of real system are accurately simulated. It includes simulation of various engineering activities, such as machining, assembly, production-

line operation, inspection, and evaluation, as well as the design process. Thus virtual engineering can extend to an entire product development and production cycle. After the prototype is validated, the production system is modeled and operated virtually. These simulations yield an optimized final prototype and production procedures that serve as the basis for a physical system. [1] While the hardware advances in the personal computer have driven significant performance improvements and cost reductions compared to traditional stand-alone instruments, it is the highly productive, integrated software that empowers hundreds of thousands of engineers and scientists.

## Background

Normal procedure adopted in a conventional CNC machining system is to identify the machining tool path based on the part geometry requirements. The part-geometry is converted into machine identifiable G-codes that provide functions for the CNC machines to follow. Several CNC interpreters are in use in industry and educational training. This paper examines the use of Graphics User Interface based LabVIEW code (*developed by National Instruments*). The paper details a methodology of applying “*Motion Assistant*” for the main purpose of machine tool control tasks such as linear interpolated movement, circular and contour generation. The front panel can be customized with knobs/ buttons/dial, and graphs to emulate control panels of traditional instruments, or visually represent the control and operation of processes. The results of implementation are shown with the of a 2-axis table top milling system that has provision to display the creation of geometry with single point tool in synchronization with the table’s movement.

Process verification is one of the most important purposes of digital simulation. . With digital simulation, the user can verify the tool path. Applications involving LabVIEW can provide a powerful graphical development environment for signal acquisition, measurement, and data presentation, with flexibility of a programming language without the complexity of traditional development tools.

## Principles of Motion Control System

A typical motion control system (with a position feedback loop) consists of the following components: **Host** - which contains the motion controller - commands the desired target position; **Motion Controller** - which is the brain of the of the system takes that target position and converts it into a voltage reference (+/- 10V); **Amplifier** - that generates the current required to drive the motor - takes the voltage supplied by the motion controller and supplies the required current to the motor. (*Figure 7*); **Motor** - that converts electrical energy into mechanical energy - produces the required torque to move to the desired target position (*Figure 6*); **Position Sensor** - senses the position of the motor and reports the result to the controller (*Figure 8*), thereby closing the loop to the motion controller (*Figure 1*).

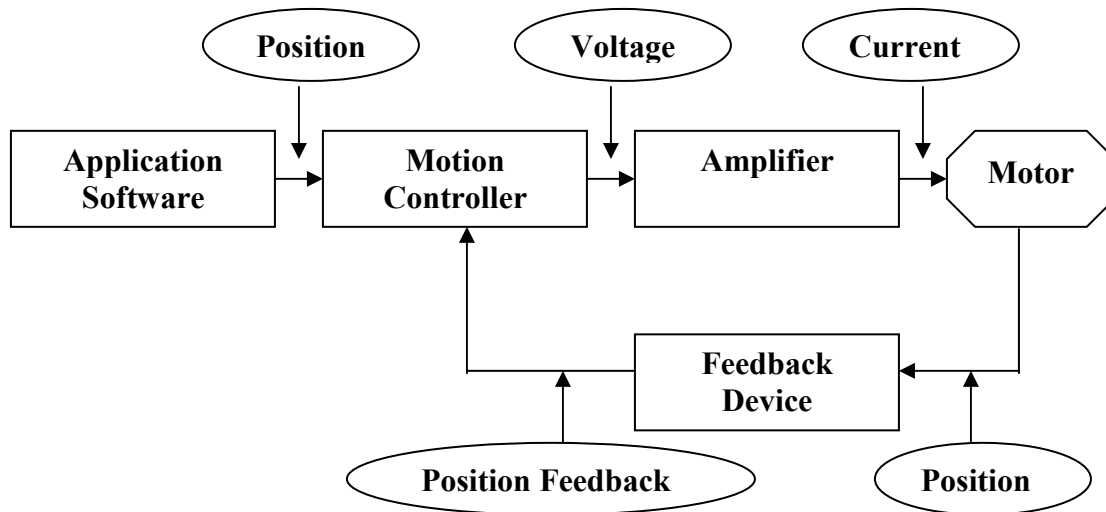


Figure 1 Elements of a Motion System

### Motion System Software

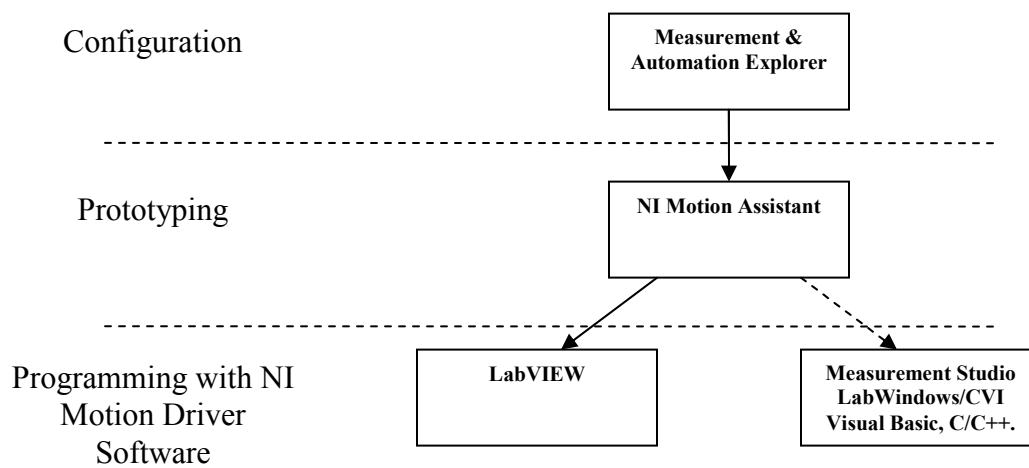


Figure 2 Motion Control Software Overview

While LabVIEW has the performance and flexibility of general purpose programming languages, such as Visual Basic or Visual C/C++, it is also a high-level rapid development environment for measurement and automation applications. In contrast with conventional programs, LabVIEW uses icons to represent functions and connect icons together to determine the flow of data through your program. It is analogous to flow-charting the code as one develops it. It has all the breadth and depth of a general-purpose programming language and is easy to use in a cost effective manner.

One of the key components of any motion control system is software as shown in *Figure 2*. Motion control software needs to be flexible, easy to use, and integrate well with other I/O pieces in the machine. The flexibility of the software helps determine the types of applications one can

create and the ease of use helps one determine how fast one can create those applications. NI Motion software tools offer an ideal combination of flexibility and ease-of-use.

### Configuration of Measurement & Automation

One of the challenges in creating a motion system is testing each of the components without creating custom programs. Using the National Instruments Measurement & Automation Explorer configuration environment, its software can easily test and configure the systems without any programming, reducing setup time. Some of the parameters one can configure and test include limit switch settings, motor type, trajectory settings, velocity, and acceleration. Once it is configured these parameters, its can save the configuration for use in the application.

Figure3 shows the configuration screen.

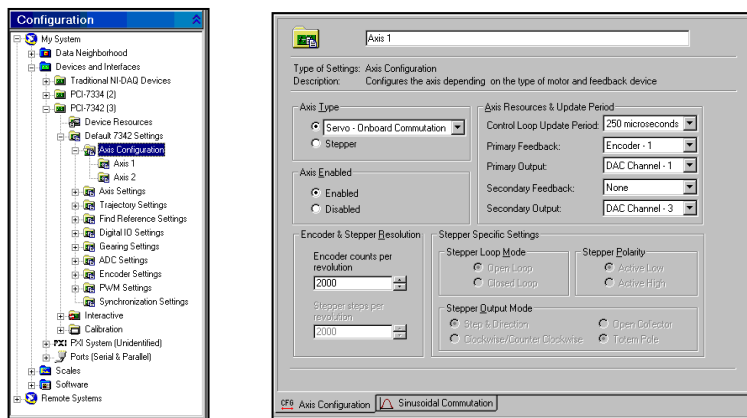
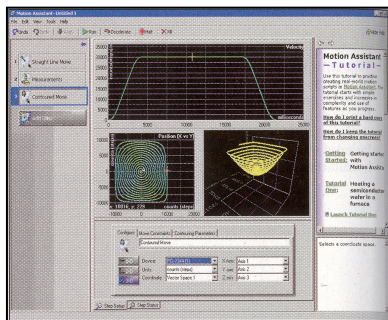


Figure 3 Configure Measurement & Automation Explorer

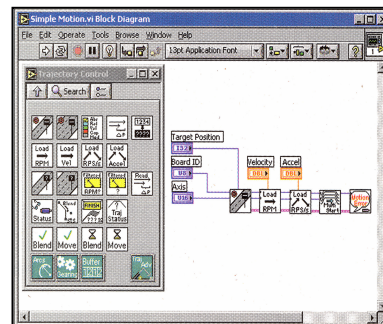
Prototyping Motion Control Tool Path or Motion Application

to

Programmable



NI Motion Assistant



LabVIEW Code Created

Figure 4 LabVIEW Code Generations

## Prototype-NI Motion Assistant

NI Motion Assistant is the tool to help one develop and prototype motion applications. It has an interactive environment where one can create motion profiles specifying parameters such as the type of move (for example, linear, or circular), velocity, and position without using any G-Code. When the sequences of moves are completed, one can generate a LabVIEW block diagram that performs the sequence configured. With this feature, one can quickly develop motion applications and then integrate them with the rest of the system. It also generates a “code recipe” that lists all of the functions as well as the parameters needed to create the same motion sequence in other programming languages, such as LabWindows/CVI and Measurement Studio for Visual Basic and Visual C/C++. In addition, the ability to graphically edit motion paths, as well as a 3D preview window provides more flexibility. (Figure 4)

## Integration of Motion Driver Software

The motion VIs include a wide range of fully functional single and multi-axis examples that one can link together and customize to meet the specific needs of any application. It can be customized for user interfaces. Further, it is possible to handle advanced motion control applications by combining the motion VI tools with LabVIEW controls so that the user can display data acquisition, process control, and image processing systems.

## Components of Motion Control

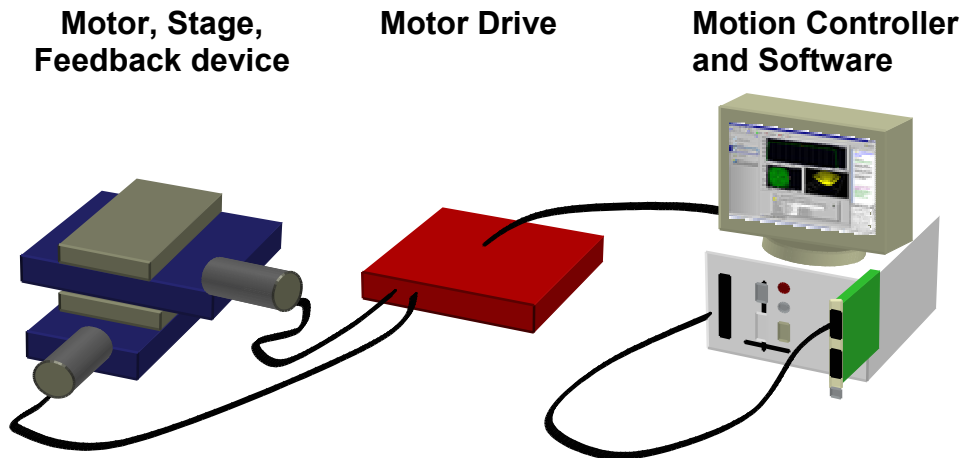


Figure 5 Components of Motion Control

### Motorized Stage:

The combination of motor and the mechanical stage provides the required actuation. *Stages* are mechanical platforms that are available in many varieties and are mechanical devices. Those most commonly used to move or position objects from one location or orientation to another by rotary motion from motor, and transfer it to useful linear motion. When choosing a stage, some of

the basic specifications are: Resolution, Length of travel, Repeatability, Load capacity, and Motor type

The peripheral devices working in association with control systems are:

- Encoders
  - Incremental encoders
- Analog Feedback
  - Potentiometer
  - Other Sensors (torque, pressure, etc...)
- Other Feedback Sources
  - IMAQ
  - DAQ

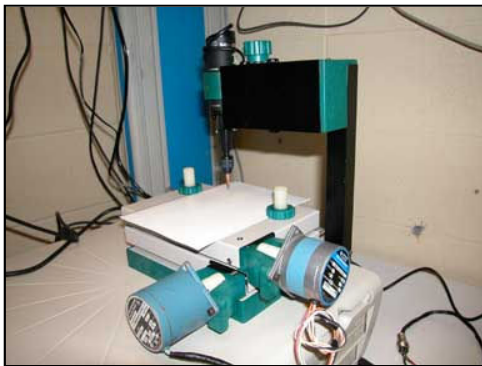


Figure 6 2-axis tabletop milling system with stationary tool    Figure 7 NI driver and Amplifier (MID7604/2)

Mechanical positioning components typically have end of travel limit switches, as well as a home switch for calibrating the zero position. (Figure 8) These limit switches can either be mechanical, where the switch is physically tripped as a stage moves over it, or electrical (hall effect sensor, phototransistor, etc). These limit-switches provide feedback to the motion control board for providing safe and accurate operation. An incremental quadrature encoder can accurately count up or down, depending on the direction of movement. However, it has no idea where the motion actually starts.

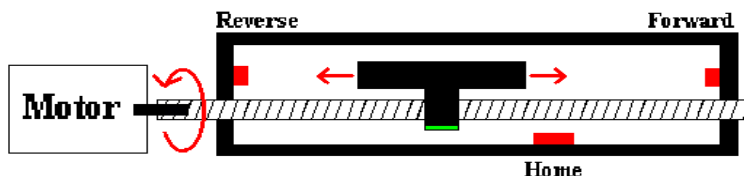


Figure 8 Limits Switch Diagram

- End of travel limits for preventing damage to the system
- Home switches for locating reference position
- Position breakpoint output for triggering
- High-speed position capture input for correlating data with positions

## Motor drive

The power drive must match the motor actuator design capabilities and provide the correct peak current, continuous current, and voltage in order to drive the motor. If one drive supplies too much current, it risks damaging the motor. If one voltage is too low, it will not be able to run the motor at the speed it is capable of.

- The motor drive must be compatible with the motor selection.
  - For example, 2-phase stepper motors require 2 phase stepper motor drives, etc...
- The motor drive must provide adequate:
  - Peak current
  - Continuous current
  - Voltage

## Controller:

The motion controller acts as the central component of the system and performs vital functions such as: Calculation of the trajectories for each commanded move; Provision of the torque commands to the motor drive; Monitoring of the limits and emergency stops for supervisory control as well as closing the PID loop.

Several different motion controllers including the high performance NI 7344 series for stepper and servo motors as well as the low cost NI 7334 controllers for stepper motors are available. There is also a variety of software options available for LabVIEW, C/C++, Visual Basic, and Measurement Studio/CVI.

The basic components of a motion controller fall into three categories:

- Supervisory control -- Does all the command sequencing and coordination
- Trajectory generator -- Generates the motion profile on the fly (multi-axes)
- Control loop -- Does the fast, closed-loop control with position, velocity, and trajectory maintenance on multiple axes simultaneously

Figure 9 shows the motion controller components and how they interact:

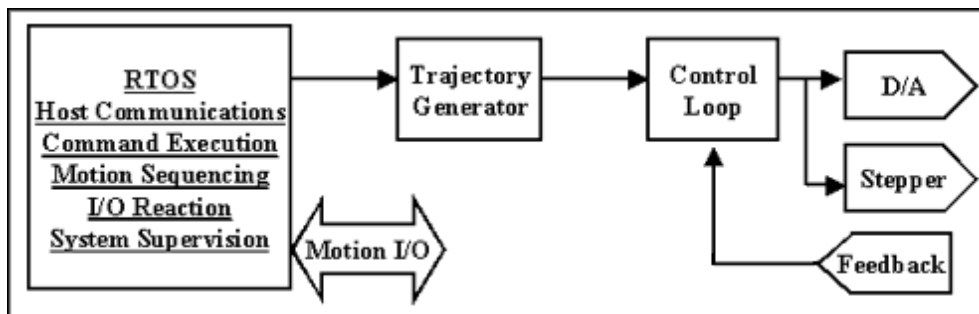


Figure 9 Overview of communication between RTOS and Motors

## Features of a Simple Motion Trajectory

The motion trajectory describes the profile of the control or command signal output by the motion controller board to the driver/amplifier, resulting in a motor/motion action that follows the profile.

The intelligent motion controller calculates the segments of a motion profile trajectory based on the parameter values programmed by the user for the move selected. The desired target position, maximum target velocity and acceleration values are used by the trajectory algorithm to determine how much time is spent in the three primary segments of a move, acceleration, constant velocity and deceleration. In the acceleration segment, motion begins from a stopped position or from a previously in-process move, and follows a prescribed acceleration ramp until the speed reaches the target velocity for the move (*Figure 10*).

Motion continues at the target velocity for a prescribed period until the controller determines that it is time to begin the deceleration segment to slow the motion to a stop exactly at the desired target position. If a move is short enough that the deceleration beginning point occurs before the acceleration has completed, then the profile will appear triangular instead of trapezoidal, and the actual velocity attained may fall short of the desired target velocity.

Enhancements to the basic trapezoidal trajectory include adjustments in acceleration and deceleration so that fine control over the shape is attained. This facilitates tailoring of the performance of a motion trajectory based upon the inertial, frictional forces, motor dynamics and other mechanical limitations in motion systems.

<b>Types of Profiles Available</b>	<b>Description</b>
<ul style="list-style-type: none"><li>• Simple point-to-point move</li><li>• Vector point-to-point move</li><li>• Blended motion</li><li>• Contouring/Arbitrary paths</li></ul>	<ul style="list-style-type: none"><li>• Target Position, Velocity, Acceleration</li><li>• Target Position (X, Y, Z), Vector Velocity, Vector Acceleration</li><li>• Target Position A, Target Position B, blend factor</li><li>• [Buffer of points]</li></ul>



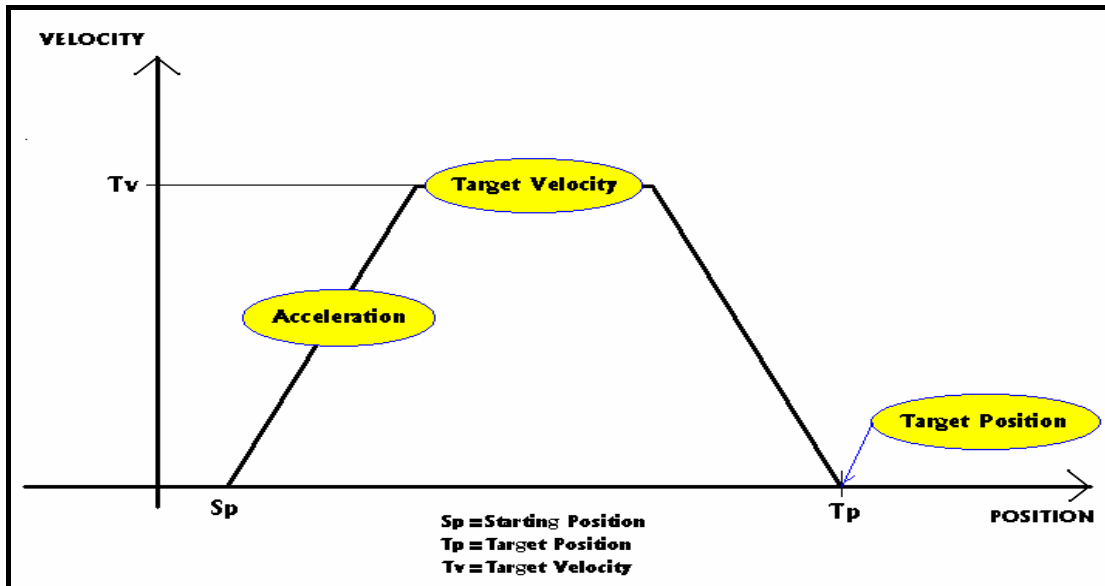


Figure 10 Motion trajectories at Starting Position, Acceleration and deceleration to the Target Position.

### Application in Educational Environments

#### Example *Parts inspection using vision with continuous motion*

This application moves an XY stage under a camera through a specific path comprised of separate moves without stopping between move segments. The device being tested is placed on the XY stage. The stage is moved in a pattern such that the entire region of interest (ROI) is covered for the camera.

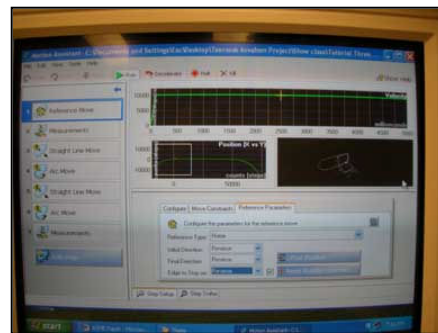
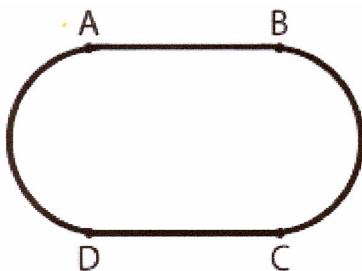


Figure 11 the Monitor of Motion Assistant Window

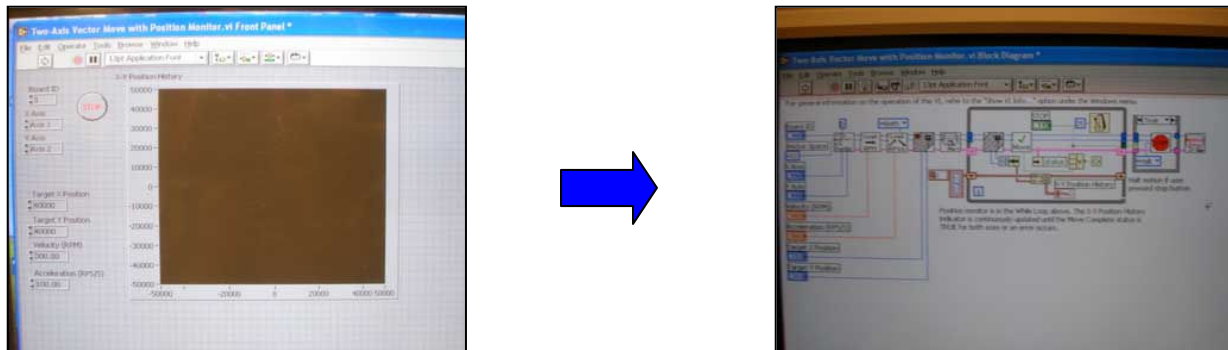
The system is comprised of an XY stage with a camera mounted on top.

The motion part of this application is prototyped by creating a script in Motion Assistant using the following steps:

1. **Reference Move step**—Moves the XY stage to its initial starting position by using the find home move type.
2. **Measurements step**—Placeholder for the code that starts image acquisition.
3. **Straight-line Move step**—Moves from starting position A to position B.
4. **Arc Move step**—Creates an arc from position B to position C.
5. **Straight-Line Move step**—Moves from position C to position D.
6. **Arc Move step**—Creates an arc from position D back to the starting position A.
7. **Measurements step**—Placeholder for the code that ends image acquisition.

After successful movements, one can generate code for the LabVIEW diagram and front panel.

**Example Front panel of Two-axis movement with Position Monitor and the Block diagram.**



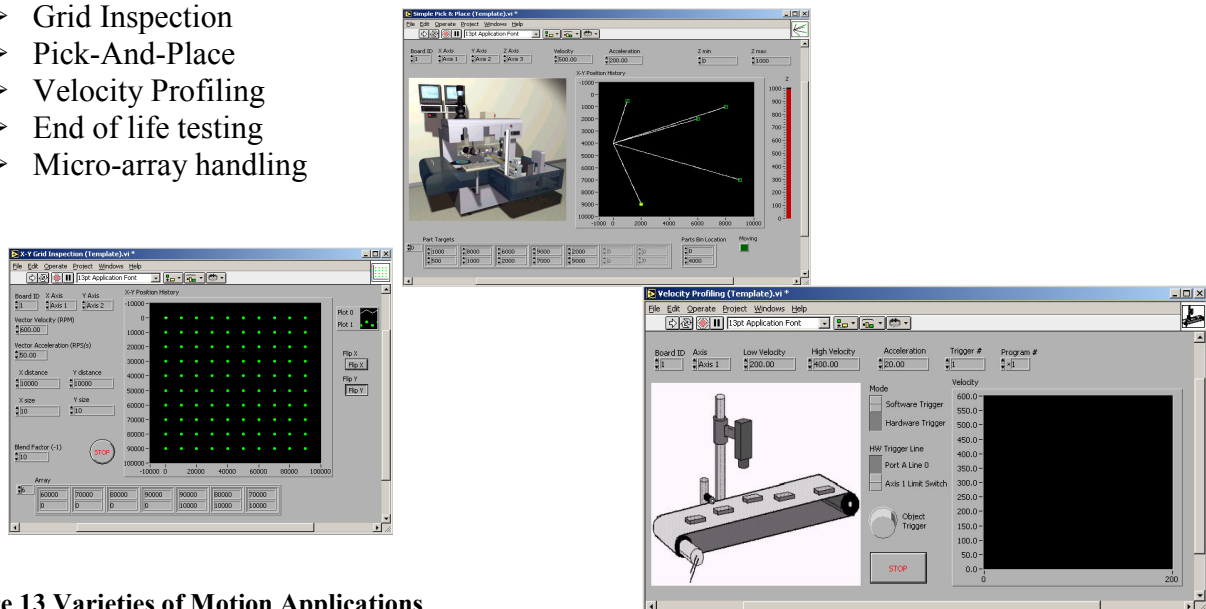
**Figure 12 Two-Axis Vector Move with Position Monitor and Block Diagram**

### Applications in Motion Control

There are several other types of motion applications that can be performed with the same basic steps that we've just outlined (*Figure 13*). For example, a grid inspection type of motion is very common. With an overhead camera, actuator, or sensor, a stage moves back and forth in a raster scanning motion to cover all points in a grid. For example, this template may be used for filling or sampling test tubes, inspecting IC labels or legs, or inspecting pixels under a microscope. A pick-and-place application involves moving from a parts bin to various part targets. At both the bin and the part target, a third axis of motion drops to 'pick' or 'place' a part.

In a velocity profiling application, a conveyor is driven in velocity mode under a camera or sensor. As an object passes under the camera, the belt decelerates so the inspection can occur with greater detail. After the object has passed, the belt speeds up again to maximize throughput.

- Grid Inspection
- Pick-And-Place
- Velocity Profiling
- End of life testing
- Micro-array handling



**Figure 13 Varieties of Motion Applications**

## Conclusion.

This new simulation technique enables the development of Graphical User Interface (GUI) is a powerful interactive way. It assists the machine operators in the creation of interactive motion. The system can run quickly. Moreover, since LabVIEW is a multiplatform programming language which means we can take the LabVIEW source code we have written on one platform and reuse it on any of the other supported platforms. The new concept has key capabilities for any remote field monitoring and control application that must pass data to a centralized reservoir or keep operators up-to-date on system status.

## References

- [1] Kunwoo Lee, "Virtual Engineering" Principle of CAD/CAM/CAE systems," pp.432-433, 1999.
- [2] Masayoshi Hashima, Yosuke Senta, and Yuichi Sato, "Design And Manufacturing Methodology For Mechanical Systems Using Virtual Product Simulator" Proceedings of DETC'02 ASME 2002 Design Engineering Technical Conferences and Computer and Information in Engineering Conference, Canada, September 29-October 2, 2002.
- [3] LabVIEW, <http://WWW.NI.com/>
- [4] James Madison, "CNC Machining Hand Book," Industrial Press Inc, NY.
- [5] David J. Ritter, "LabVIEW Essential Techniques GUI," McGraw-Hill, NY.
- [6] Peter Smid, "CNC Programming Handbook," Industrial Press Inc, NY.
- [7] Yusuf Altintas, "Manufacturing Automation" Cambridge University Press
- [8] S C Black, V Chiles, A J Lissaman, S J Martin, "Principles of Engineering Manufacture," ARNOLD, NY.
- [9] Serope Kalpakjian, "Manufacturing Processes for Engineering Materials," Addison-Wesley Publishing Company.

## Biographical

### DEVIDAS SHETTY

Devdas Shetty is Vernon D. Roosa Professor of Manufacturing Engineering and Associate Dean of the College of Engineering, University of Hartford, Connecticut. He also serves as the Director of the Engineering Applications Center, which is an affiliate structure with the regional industries. Prof. Shetty has published widely and has books in the area of Mechatronics Systems Design and Product Design. His areas of expertise are Mechatronics, Manufacturing and Product Design.

### CLAUDIO CAMPANA

Claudio Campana is currently working as a Research Engineer at the Engineering Application Center of the University of Hartford. He received his Bachelors degree from Boston University and Masters from University of Hartford in Mechanical Engineering. His areas of expertise are CAD/CAM and Mechatronics.

### TEERASAK ASVAHEM

Teerasak Asvahem is graduate student in Mechanical Engineering of the University of Hartford. He received his Bachelors degree from Mahanakron University.