Realization of the 5-Axis Machine Tool Digital Twin Using Direct Servo Control from CAM

Roby Lynn[1]
George W. Woodruff
School of Mechanical Engineering
Georgia Institute of Technology
Atlanta, GA, USA

Mukul Sati
School of Interactive Computing
Georgia Institute of Technology
Atlanta, GA, USA

Tommy Tucker
Tucker Innovations, Inc.
Charlotte, NC, USA

Jarek Rossignac
School of Interactive Computing
Georgia Institute of Technology
Atlanta, GA, USA

Christopher Saldana
George W. Woodruff
School of Mechanical Engineering
Georgia Institute of Technology
Atlanta, GA, USA

Thomas Kurfess
George W. Woodruff
School of Mechanical Engineering
Georgia Institute of Technology
Atlanta, GA, USA

ABSTRACT

This paper describes an architecture for control and monitoring of a 5-axis computer numerical control (CNC) machine tool directly from a computer-aided manufacturing (CAM) system without reliance on the text-based G-Code toolpath definition format that is currently standard in industrial practice. Instead of defining a toolpath as a set of geometric primitives as is done with G-Code, this architecture utilizes a high-speed bidirectional data pathway between the CAM system and the CNC machine to transfer dense time samples of axis position information between the CAM system and the servo controllers of the machine tool's motion control system in near-realtime. Time samples of axis position are created using a time-optimal trajectory planning algorithm instead of a proprietary trajectory planning strategy that is common in industrial CNC systems. The developed architecture is machine agnostic, and can be used both for enhanced control of machine tool motion and powerful visualization and analysis tasks. An implementation of the system using an open-source machine tool controller known as Machinekit is presented, and a Digital Twin of the machine tool is constructed in the CAM system and shown to be capable of visualizing the as-executed toolpath during machine operation.

KEYWORDS: Computer-aided manufacturing, computer numerical control, cyber-physical systems, Digital Twin, G-code, time optimal trajectory planning, servo control

1 INTRODUCTION

The current state of data communication for CNC machine tools relies on a more than 50 year old format known as G-Code, which is a text-based programming language used to convey movement primitives between a toolpath planning system and the controller of a CNC machine. G-Code allows unidirectional data transfer between the CAM system and the CNC, but in-process monitoring of the machine must be accomplished using separate protocols, such as MTConnect or Object Linking and Embedding for Process Control Unified Architecture (OPC UA) [1]. Familiarity with numerous control and communication standards is required for technicians and manufacturing engineers to be proficient at implementation of a CNC machining process.

The creation of G-Code from a CAM system necessitates translation of geometric entities into the text-based format that consists of lines, arcs, splines, and other motion primitives, each of which is completely defined by endpoints and other parameters. It is the responsibility of the CNC system to realize motion of the cutting tool from information contained in the G-Code commands, and machine tool vendors often use proprietary trajectory planning algorithms to accomplish this task [2], [3]. As a result, some amount of information is lost in translation from the CAM system to the CNC: although the CAM system may be capable of computing desired velocity, acceleration, jerk, and higher positional derivatives along a toolpath, this information cannot be conveyed to the machine tool using G-Code [4], [5]. Additionally, monitoring of actual machine tool motion is hampered by a vendor's implementation of a chosen communication protocol (e.g., MTConnect),

[1]Corresponding Author E-mail: roby.lynn@gatech.edu

which may be insufficient for engineers and supervisory staff to evaluate an as-executed toolpath [6], [7]. This paper describes the development and implementation of a tightly-coupled CAM-CNC architecture that enables more complete control and monitoring of a machining process from within a CAM system. The implementation presented in this paper uses the SculptPrint voxel-based CAM system in conjunction with the PocketNC, which is a desktop-sized machine tool powered by the open-source Machinekit platform.

2 SYSTEM ARCHITECTURE

2.1 SculptPrint: Voxel-Based Computer-Aided Manufacturing

This research uses a CAM software package known as SculptPrint, which leverages the high-performance computing capability of modern graphics processing units (GPUs) to perform automated toolpath generation and analysis for multi-axis machine tools [8]. Part geometry in SculptPrint is represented using voxels, which are the three-dimensional analog to two-dimensional image pixels. The use of voxel models enables creation of high-density toolpaths that can be used to machine intricate and organic shapes that would be difficult to create with traditional CAM [9].

2.2 Machinekit: An Open-Source Machine Tool Controller

The overwhelming prevalence of proprietary machine tool control platforms has motivated the development of a Linux-based open-source alternative CNC known as Machinekit, a fully-featured software machine controller that is capable of controlling both simple and complex multi-axis machine tools [10]. Machinekit enjoys a thriving development community that is committed to creating a usable and fully-featured machine control environment [11]. This research relies on an embedded computer known as the Beaglebone Black (BBB), which runs Machinekit in conjunction with the Xenomai realtime (RT) Linux framework to provide reliable and deterministic control of a physical machine tool. The actual machine tool used for this work is the PocketNC, which is a 5-axis desktop-sized machine that is targeted at the maker community.

2.3 Time Parameterization of Toolpaths

An interactive design session in SculptPrint allows the user to describe toolpaths as the result of voxelized constructive solid geometry (CSG) operations [12]. The output of the session is a sequence of very finely spaced affine frames that represent sample orientations of the cutting tool. Additionally, SculptPrint also provides, for each sample orientation, the corresponding sample positions that the joint motors should track. This is done with the aid of a machine specific inverse kinematic model.

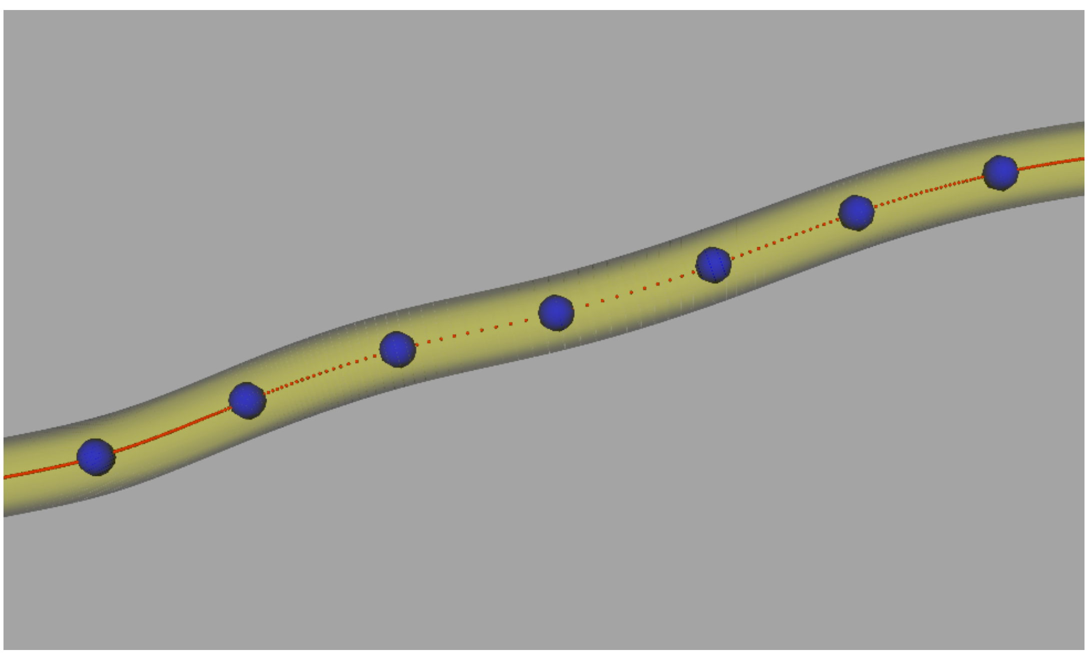


Figure 1: Position Samples and Resulting Time Parameterization

Typical motion planning algorithms work by blending G-code primitives. It would be possible to use Machinekit's motion blending algorithm by interpreting two consecutive joint samples as defining the directed edge that has the two samples as end points. Given the dense position sampling provided by SculptPrint, the direct control architecture instead directly fits and optimizes a spline that interpolates axis position samples using techniques presented in [13]. After the joint samples are broken into retraction-free sequences, the approach described by Pham is applied as follows: a cubic spline is fit to the joint samples with an a-priori prescription of individual joint velocities at the beginning and end of each sequence; then, the time paramaterization of the spline is optimized while being constrained to obey both path geometry and the manufacturer-specified velocity and acceleration bounds of the motors for each joint. The resulting motion can be sampled to obtain the position of each joint at the servo update rate. An example path parameterization is shown in Figure 1. The yellow tube is the optimized spline, the blue points are the position constraints obtained from the CAM system, and the red points are the actual servo samples in the part reference frame. Larger spacing between adjacent points indicates faster traversal of the cutting tool along the path. The very dense sampling ensures that the joint motions affect a motion of the cutting tool that adheres to the user's design intent.

2.4 Direct Servo Control Scheme

Communication of command and feedback information between the CAM system and the RT CNC system is accomplished according to Figure 2. The non-RT CAM system resides on a standalone workstation PC, and serves as both the toolpath planner and the operator interface for the machine tool. The user interacts with SculptPrint to perform process planning and

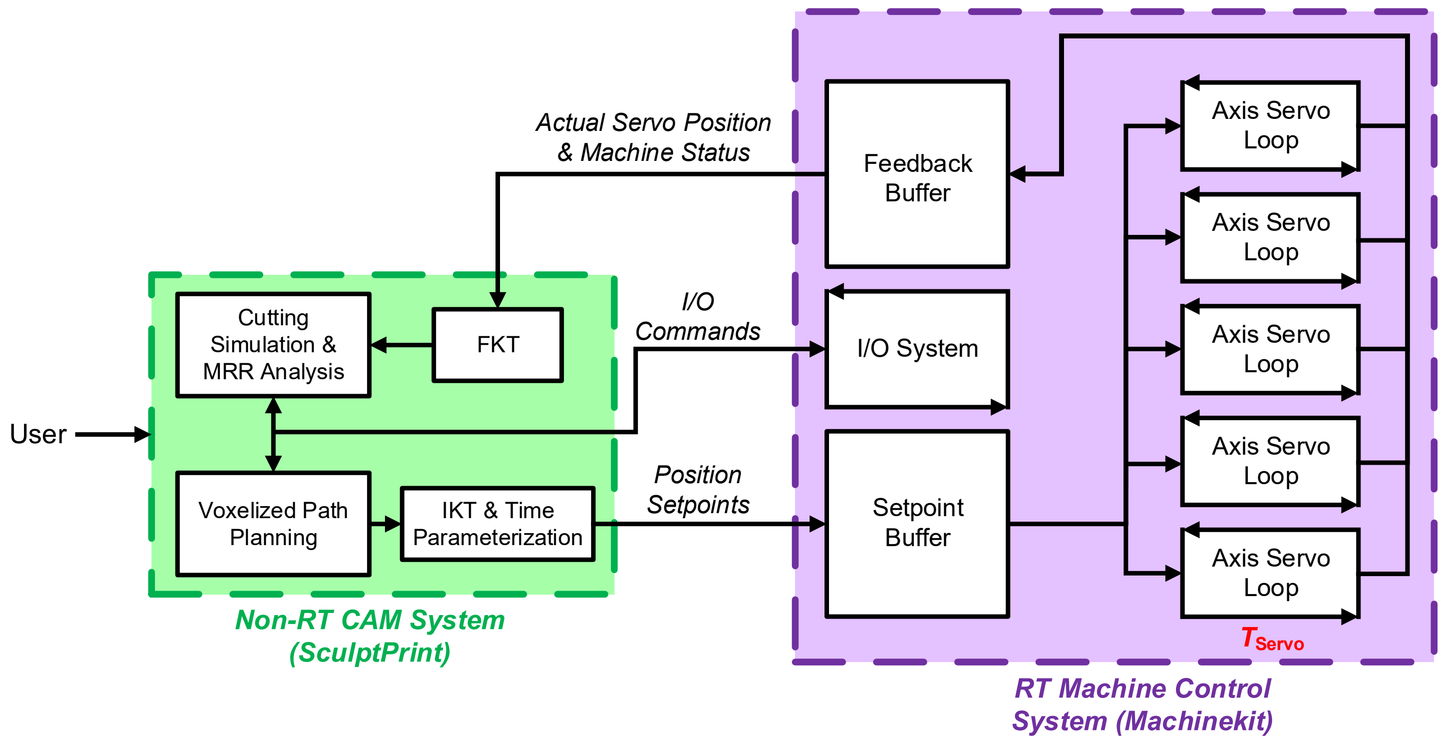


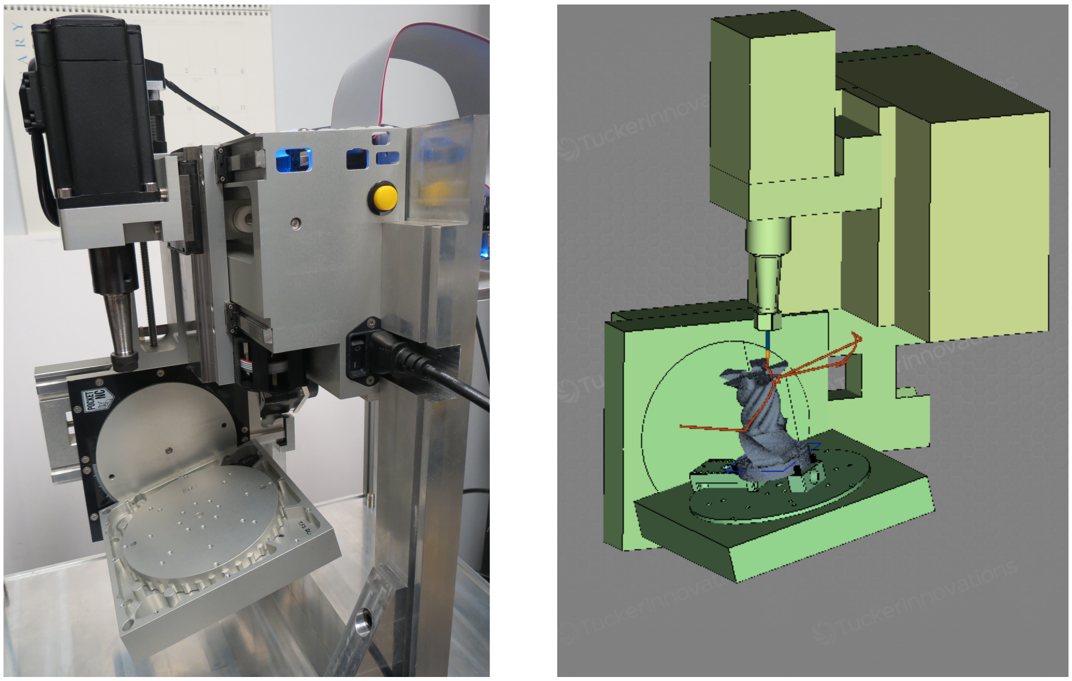
Figure 2: Direct Control System for a 5-Axis Machine Tool

analysis, and the resulting toolpaths consist of pose samples in the part coordinate frame which the tool should track. These poses are then converted to axis positions using the Inverse Kinematic Transformation (IKT) derived from the geometry of the machine. The resulting axis position commands are transformed to time-based position samples using the method presented in Section 2.3 to create axis position setpoints that can be written to each axis servo controller at T_{Servo}, the update rate of the servo system. Upon execution of each setpoint, the servo loops record the actual axis positions, also at T_{Servo}, which are passed back to the CAM system and converted to poses that are visualized in the part coordinate frame using the Forward Kinematic Transformation (FKT) for the machine tool. This architecture thus consists of a machine-agnostic control and monitoring environment that can be used to execute and monitor toolpaths for any machine tool configuration in near-realtime.

Communication of RT data between the RT machine controller and the non-RT CAM system is accomplished using setpoint and feedback buffers whose fill level can float to absorb non-deterministic latencies introduced by the connection of the RT and non-RT subsystems. The CNC system consumes one position setpoint per axis from the setpoint buffer every T_{Servo} and supplies one position sample per axis to the feedback buffer every T_{Servo}. The CAM system must maintain the proper fill level of the setpoint buffer to ensure that it is not exhausted during machine operation (which would cause a cessation of movement) and must also consume position samples from the feedback buffer quickly enough so that the buffer does not overflow. In this research, communication was performed using an Ethernet connection between the BBB and the network interface of the CAM workstation.

3 THE 5-AXIS MACHINE TOOL DIGITAL TWIN

The direct control system architecture in Figure 2 was implemented using the SculptPrint CAM system, a collection of Python applications, and changes to core Machinekit code. The Python scripts performed both time parameterization of SculptPrint toolpaths and high-speed bidirectional machine communication. A CAM workstation with an NVIDIA Quadro M5000 GPU was used to run both SculptPrint and the supporting Python scripts, and communicated with the PocketNC with a direct (i.e., switchless) Ethernet connection. The PocketNC used to validate this system is shown in Figure 3(a). One Python script was responsible for wrapping the time parameterization algorithms provided by [13]. The generation servo position samples were sent by a separate multithreaded Python script over a transmission control protocol (TCP) connection to the machine; the script also received machine position feedback from a user datagram protocol (UDP) socket listener on the CAM PC. The feedback samples were relayed to SculptPrint and used to move the axes of the interactive machine model and to generate the orange tool tip trace as the machine was running, as shown in Figure 3(b). To limit computational load and bandwidth consumption, the update rate of the Digital Twin in the CAM system was controllable by the user.



(a) *5-Axis Desktop-Sized CNC Machine* (b) *Interactive Representation in CAM*

Figure 3: Machine Tool Digital Twin

4 CONCLUSIONS

This paper described a CNC control architecture and its realization to create a tightly-coupled CAM-CNC system that enables enhanced control and monitoring of a 5-axis machine tool directly from a CAM system. This work lays the foundation for a new machine tool control strategy in which the traditional generation and transfer of G-Code to a CNC is no longer required; instead, machinists and manufacturing engineers need only manipulate a CAM system to create, execute, and analyze toolpaths in an interactive fashion. Future work will investigate both automated process plan generation and enhanced trajectory planning strategies that are realizable by controlling point spacing between axis position samples.

REFERENCES

[1] M. Helu, T. Hedberg, and A. B. Feeney, “Reference architecture to integrate heterogeneous manufacturing systems for the digital thread,” *CIRP Journal of Manufacturing Science and Technology*, 2017. [Online]. Available: http://ws680.nist.gov/publication/get%7B%5C_%7Dpdf.cfm?pub%7B%5C_%7Did=922753.

[2] X. Beudaert, S. Lavernhe, and C. Tournier, “Direct trajectory interpolation on the surface using an open CNC,” *International Journal of Advanced Manufacturing Technology*, vol. 75, no. 1-4, pp. 535–546, 2014, ISSN: 14333015. DOI: 10.1007/s00170-014-6134-7.

[3] K. Erkorkmaz and Y. Altintas, "High speed CNC system design. Part I: Jerk limited trajectory generation and quintic spline interpolation," *International Journal of Machine Tools and Manufacture*, vol. 41, no. 9, pp. 1323–1345, 2001, ISSN: 08906955. DOI: 10.1016/S0890-6955(01)00002-5. [Online]. Available: http://www.sciencedirect.com.prx.library.gatech.edu/science/article/pii/S0890695501000025.

[4] Y.-F. Tsai, R. T. Farouki, and B. Feldman, "Performance analysis of CNC interpolators for time-dependent feedrates along PH curves," *Computer Aided Geometric Design*, vol. 18, pp. 245–265, 2001. [Online]. Available: www.elsevier.com/locate/comaid.

[5] X. W. Xu, H. Wang, J. Mao, S. T. Newman, T. R. Kramer, F. M. Proctor, and J. L. Michaloski, "STEP-compliant NC research: the search for intelligent CAD/CAPP/CAM/CNC integration," *International Journal of Production Research*, vol. 43, no. 17, pp. 3703–3743, Sep. 2005, ISSN: 0020-7543. DOI: 10.1080/00207540500137530. [Online]. Available: http://www.tandfonline.com/doi/abs/10.1080/00207540500137530.

[6] R. Lynn, A. Chen, S. Locks, C. Nath, and T. Kurfess, "Intelligent and Accessible Data Flow Architectures for Manufacturing System Optimization," in *IFIP Advances in Information and Communication Technology*, vol. 459, 2015, pp. 27–35, ISBN: 9783319227559. DOI: 10.1007/978-3-319-22756-6_4. [Online]. Available: http://link.springer.com/10.1007/978-3-319-22756-6%7B%5C_%7D4.

[7] R. Lynn, W. Louhichi, M. Parto, E. Wescoat, and T. Kurfess, "Rapidly Deployable MTConnect-Based Machine Tool Monitoring Systems," in *2017 ASME Manufacturing Science and Engineering Conference (MSEC)*, vol. 3, 2017, pp. 1–10, ISBN: 9780791850749. DOI: 10.1115/MSEC2017-3012.

[8] Tucker Innovations Inc, *SculptPrint - The Subtractive 3D Printing Application*. [Online]. Available: www.sculptprint3d.com.

[9] R. Lynn, M. Dinar, N. Huang, J. Yu, J. Collins, C. Greer, T. Tucker, and T. Kurfess, "Direct Digital Subtractive Manufacturing of Functional Assemblies Using Voxel-Based Models," *ASME Journal of Manufacturing Science and Engineering*, 2017.

[10] J. Albus and R. Lumia, "The Enhanced Machine Controller (EMC): An Open Architecture Controller for Machine Tools," *Journal of Manufacturing Review*, vol. Vol. 7, no. No. 3, pp. 278–280, 1994. [Online]. Available: https://www.nist.gov/publications/enhanced-machine-controller-emc-open-architecture-controller-machine-tools.

[11] *Machinekit Integrator Manual*. [Online]. Available: http://www.machinekit.io/docs/index-integrator/.

[12] M. M. Hossain, T. M. Tucker, T. R. Kurfess, and R. W. Vuduc, "Hybrid dynamic trees for extreme-resolution 3d sparse data modeling," in *Parallel and Distributed Processing Symposium, 2016 IEEE International*, IEEE, 2016, pp. 132–141.

[13] Q.-C. Pham, "A general, fast, and robust implementation of the time-optimal path parameterization algorithm," *IEEE Transactions on Robotics*, vol. 30, no. 6, pp. 1533–1540, 2014.