International Journal of Production Research

ISSN: 0020-7543 (Print) 1366-588X (Online) Journal homepage: www.tandfonline.com/journals/tprs20

Developing a virtual machining model to generate MTConnect machine-monitoring data from STEP-NC

Seung-Jun Shin, Jungyub Woo, Duck Bong Kim, Senthilkumaran Kumaraguru & Sudarsan Rachuri

To cite this article: Seung-Jun Shin, Jungyub Woo, Duck Bong Kim, Senthilkumaran Kumaraguru & Sudarsan Rachuri (2016) Developing a virtual machining model to generate MTConnect machine-monitoring data from STEP-NC, International Journal of Production Research, 54:15, 4487-4505, DOI: 10.1080/00207543.2015.1064182

To link to this article: https://doi.org/10.1080/00207543.2015.1064182

Published online: 27 Jul 2015.

International Journal of Production Research, 2016
Vol. 54, No. 15, 4487–4505, http://dx.doi.org/10.1080/00207543.2015.1064182

Taylor & Francis
Taylor & Francis Group

Developing a virtual machining model to generate MTConnect machine-monitoring data from STEP-NC

Seung-Jun Shin[a], Jungyub Woo[b]*, Duck Bong Kim[a], Senthilkumaran Kumaraguru[a,1] and Sudarsan Rachuri[a]

[a]*Systems Integration Division, Engineering Laboratory, National Institute of Standards and Technology, Gaithersburg, MD, USA;* [b]*Software and Systems Division, Information Technology Laboratory, National Institute of Standards and Technology, Gaithersburg, MD, USA*

(*Received 15 October 2014; accepted 11 June 2015*)

The ability to predict performance of manufacturing equipment during early stages of process planning is vital for improving efficiency of manufacturing processes. In the metal cutting industry, measurement of machining performance is usually carried out by collecting machine-monitoring data that record the machine tool's actions (e.g. coordinates of axis location and power consumption). Understanding the impacts of process planning decisions is central to the enhancement of the machining performance. However, current methodologies lack the necessary models and tools to predict impacts of process planning decisions on the machining performance. This paper presents the development of a virtual machining model (called *STEP2M* model) that generates machine-monitoring data from process planning data. The *STEP2M* model builds upon a physical model-based analysis for the sources of energy on a machine tool, and adopts STEP-NC and MTConnect standardised interfaces to represent process planning and machine-monitoring data. We have developed a prototype system for 2-axis turning operation and validated the system by conducting an experiment using a Computer Numerical Control lathe. The virtual machining model presented in this paper enables process planners to analyse machining performance through virtual measurement and to perform interoperable data communication through standardised interfaces.

Keywords: performance measurement; simulation model; STEP-NC; MTConnect; metal cutting; power consumption

1. Introduction

In manufacturing, automated measurement of process and resource performance is becoming increasingly important to improve efficiency of manufacturing processes and equipment (Muchiri and Pintelon 2008). In machining processes, the applied use of metrics allows observation and quantification of outcomes of a machine tool's actions. Primarily, the measurement of these metrics requires the collection and analysis of machine-monitoring data that record the events and movements of machine tool components in relation to planned machining operations. On the other hand, process planning decisions greatly influence efficiency of machining operations (Xu, Wang, and Newman 2011). Proper determination of process sequence and process parameter selection can occur during the process planning stage by measuring and analysing the machine-monitoring data to make machining performance better.

Energy-conscious process planning enables to perform machining processes with better energy efficiency, thereby significantly reducing the industrial consumption of energy (Newman et al. 2012). For this reason, previous works have developed predictive and optimisation models to improve energy-related metrics such as energy consumption and power consumption. Power consumption is a key dominant metric in the analysis of energy efficiency, and consists of the basic power consumed by a machine tool's actions and the cutting power needed to remove a workpiece (Kara and Li 2011; Diaz et al. 2013).

Mativenga and Rajemi (2011) proposed a methodology for selecting process parameters to minimise energy footprint using a tool life equation. Bhushan (2013) presented experimental investigations into the effects of process parameters and nose radius on power consumption and tool life. Yan and Li (2013) and Li et al. (2013), respectively, applied the grey relational analysis or the neural network to evaluate trade-offs between energy, material removal rate (MRR) and surface roughness. Iqbal et al. (2013) proposed a fuzzy rule-based model to find optimum process parameters in terms of energy consumption, tool life and MRR. Kant and Sangwan (2014) presented a statistical model to optimise power consumption and surface roughness in machine models. Velchev et al. (2014) analysed the influence of MRR and the

*Corresponding author. Email: jungyub.woo@nist.gov
[1]Present address: Indian Institute of Information Technology Design and Manufacturing (IIITD&M), Kancheepuram, Chennai, India

insert grades on specific energy consumption, i.e. the energy consumption for removing a unit-volume material, in turning machining. These previous works have shown good results in finding optimum process parameters in terms of energy and power consumption as well as productivity-related metrics. Their optimisation problems have been solved based on the predictive models constructed by using design of experiments. However, their experiment-based models have been created by statistical curve fitting, and they do not represent physical aspects of the energy uses inside a machine tool (Newman et al. 2012). Experiments should be repeated to generate their models that correspond to a different machine tool. Therefore, a physical model-based approach that can create machine-specific models based upon physical aspects rather than experimental aspects is necessary for improving energy efficiency in machining. This approach enables to integrate the consumed energy in machines into process planning as a new predictable criterion (Newman et al. 2012).

In addition, the impact of process planning decisions to machining performance is verified after the fact, i.e. when a machining process is completed. Due to this gap, the influence of process planning is unknown until one has acquired and analysed the monitoring data (Vijayaraghavan et al. 2008). Significant research has tried to make the influence of process planning known before the acquisition and analysis of the monitoring data. Much of the research have used Numerical Control (NC) part programmes, which are current machine-interpretable formats, as inputs to virtual machining models that can simulate machining performance at the post-processing stage, which converts process planning decisions to an NC part programme.

Avram and Xirouchakis (2011) proposed a theoretical model to evaluate mechanical energy requirements by using an Automatic Programming Tool file. Larek et al. (2011) described a discrete-event simulation to generate workpiece-specific power consumption from an NC part programme. Braun and Heisel (2012) developed a modular modelling approach to generate the estimated energy consumption from an NC part programme. Shao, Kibira, and Lyons (2010) developed a simulation tool to assess power consumption in a machining process. Dietmair, Verl, and Eberspaecher (2009) developed an electrical power consumption model of a machine tool, which depended on its operational state, kinematics and processes.

All these efforts have contributed to the success of using virtual machining models to estimate target metrics. However, as noted, their models are limited because they rely on an NC part programme as input data. Since the NC part programme is only a set of code instructions for a machine's actions and movements, it is rather difficult to identify the process sequence and the parameter selection decisions made during the process planning stage from the NC part programme alone (Shin, Suh, and Stroud 2007). Therefore, it would be advantageous to use process planning data rather than an NC part programme to evaluate accurately the impacts of process planning decisions.

Up to now, vendor-specific communication protocols were commonly used for representing and interpreting process plan and machine-monitoring data. Machine tool builders and controller providers create these proprietary data for their own black-box applications. Therefore, it is difficult to use the proprietary data as input to an outside application. This drives a need for a standardised communication interface for process plan and machine-monitoring data to provide interoperable data exchange in an open data sharing environment.

This paper presents a virtual machining model, called *STEP2M* model, which generates machine-monitoring data from process planning data instead of NC programming data. This virtual model builds upon a physical model-based analysis for the energy sources needed to run a machine tool. Also, this model uses standardised communication interfaces to allow open data exchange: ISO 14649 (STEP-NC) (ISO 2003) and MTConnect (AMT 2011). STEP-compliant data interface for Numerical Controls (STEP-NC) standardises process planning data and the MTConnect standardises machine-monitoring data. While the NC programming data provide the instructions and coordinate for machine components' motions, the STEP-NC planning data can provide the comprehensive inputs by specifying machining processes. The MTConnect is a set of open standards intended to foster greater interoperability between machine tools and applications by publishing data using internet protocol such as eXtensible Markup Language (XML) and Hyper Text Transfer Protocol (AMT 2011). We have developed a prototype system to validate the virtual machining model for a two axial Computer Numerical Control (CNC) lathe machine.

Section 2 of this paper describes the input of *STEP2M* model in the form of STEP-NC and the output in the form of MTConnect and the interoperability resolution of STEP-NC and MTConnect interfaces. Section 3 presents the logic for the *STEP2M* model, and Section 4 presents a prototype system with discussion. Finally, Section 5 outlines a summary and conclusions of the present work.

2. Data definitions

This section provides essential descriptions necessary for the input and output of the virtual machining model. In addition, because the virtual machining model will use standard-based heterogeneous data interfaces, i.e. STEP-NC for the input data and MTConnect for the output data, we should resolve their interoperability issues. For this purpose, this section introduces the literature that has resolved these interoperability issues. Then we figure out our research scope different with that of the literature in this section.

2.1 *Input data: STEP-NC*

STEP-NC provides a data model for seamless data exchange between Computer-Aided Process Planning, Computer-Aided Manufacturing and CNC (Xu et al. 2005). STEP-NC improves upon conventional NC codes by using an object-oriented *workingstep* entity to specify machining processes rather than workpiece-tool relative motions (Zhao, Habeeb, and Xu 2009). *Workingsteps* represent the essential building blocks of machining operations. *Workplan* organises descriptions and the sequence of the *workingsteps* (process sequence). In turn, each *workingstep* describes a single machining operation associating instances of a tool, a strategy, a technology and a machine function (parameter selection) with a unique instance of a single manufacturing feature (ISO 2003).

Process sequence and parameter selection have an influence on the machining performance in the current CNC environment that passively executes code instructions converted from the process sequence and parameter selection (Xu, Wang, and Newman 2011). Since STEP-NC explicitly represents process sequence and parameter selection, STEP-NC-based process planning data can be useful for estimating machining performances.

2.2 *Output data: MTConnect*

MTConnect defines a common language and structure for communication of monitoring data from manufacturing equipment (Vijayaraghavan and Dornfeld 2010). The MTConnect architecture consists of: (1) a device, which is a machine tool, (2) an agent, which is a function object that collects data from the device and delivers it to the applications, (3) a client application, which is a user application that consumes the device's data and (4) a Lightweight Directory Access Protocol server, which is a server that translates device names for the agent (AMT 2011).

A probe document defines an information model for the data obtained from the machine tool. These data are available from the device's constituent component axes, spindles, programmes and control sequences (AMT 2012a). The agent receives a manufacturing data sample or events from the device, stores these data sample or events, and transmits them to a client application. A 'sample' datum is the value of a continuous data stream at a point in time. Position, acceleration, linear force and wattage are good examples of sample data. An 'event' datum describes an asynchronous change in state. A 'condition' datum reports the device's health and ability to function (AMT 2012a).

MTConnect-based data contents can provide essential data for measuring process and resource performance. For example, we can measure machining time from a data item called *accumulated time*. We can also measure energy consumption by extracting a time series data for *power* and integrating power values over machining time.

2.3 *Interoperability of STEP-NC and MTConnect*

For generating an MTConnect document from a STEP-NC part programme, we took a critical look into past efforts to develop solutions to the associated interoperability problems. Those problems stem from efforts to improve real-time process control by using on-line feedback of machine-monitoring data associated with process planning data (Wang, Xu, and Tedford 2007).

In the recent past, two studies have contributed toward showing feasibility of interoperability of these two standardised data-sets. Kadir et al. (2011) measured the coordinate origin of a worktable from MTConnect streaming data and re-assigned the origin to a STEP-NC programme. However, they focused only on the coordinate re-assignment of a STEP-NC part programme from MTConnect data as a feedback control for positional accuracy. Ridwan and Xu (2013) developed a methodology to optimise process parameters for minimal machining time and maximum surface quality in the process planning stage. In addition, they developed an advanced CNC system that enabled fuzzy-based feedrate optimisation through MTConnect data. However, they used MTConnect as a data interface to collect metrics (e.g. vibration, cutting power and feedrate) after the fact. Differently, our study aims to develop a virtual machining model to generate an MTConnect document from a STEP-NC part programme in the process planning stage before the actual machining process begins.

3. *STEP2M* model

This section describes the logic of the virtual machining model (*STEP2M* model) that generates an MTConnect XML document from a STEP-NC part programme. This paper describes the logic mainly for a 2-axis turning process. The abstract idea of the logic can be widely applied to other machining processes. This will be more discussed in Section 4.4.

3.1 *Logical flow of* STEP2M *model*

STEP-NC data organise a sequence of machining operations statically in terms of *workingsteps*. Meanwhile, MTConnect generates continuous snap shots of a machine tool's actions and events of time domain. To transit static data to dynamic data, the virtual machining model first interprets STEP-NC data and transforms it to an NC part programme. Second, it generates a machine tool's events and movements, matched with sequential execution of the NC programme. Third, it reasonably generates kinematics (e.g. locations due to acceleration, steady state and deceleration) and dynamics (e.g. power consumed by servomotor and cutting operations) corresponding to the events and movements. Fourth, it documents all the machine tool's machining data-sets, complying with MTConnect protocol.

For this purpose, the virtual machining model needs machine tool-specification data in addition to the STEP-NC data because the machine tool specification influences the simulation of dynamics and kinematics of a machine tool. The machine tool-specification data define the capability and performance of a machine tool's main body and its constituent components. Moreover, it is necessary to select a numerical controller system so that we can design the NC part programme to be fully compatible with its code scheme. Because a different NC system requires a different code scheme, there is a need to develop individual post-processors that can output NC programmes suitable for a selected NC system.

Figure 1 shows the logical flow to organise the functions and data flow of the *STEP2M* model. The functions are modularized into three modules in terms of their roles – 'STEP-NC interface', 'machining modeller' and 'MTConnect interface'. Since the modules transact the data in the form of objects and files, the modular approach allows one to design flexibly and extensibly the internal functions and logics in the modules. The following sub-sections describe the details of the three modules.

3.2 *STEP-NC interface*

For interfacing STEP-NC data to a specific machine tool, it is necessary to transform a STEP-NC part programme to a machine-interpretable format for controlling a machine. The 'STEP-NC interface' module parses a STEP-NC part programme and generates its associated NC part programme. The inputs to this module are a STEP-NC part programme and the choice of a specific NC system. The outputs are an NC part programme and STEP-NC objects. The following sub-sections explain the logic and data input/output for each function.

3.2.1 *STEP-NC interpretation*

'STEP-NC interpretation' parses a STEP-NC part programme and instantiates entities and attributes of the part programme into a STEP-NC object model. Figure 2 shows the STEP-NC object model that identifies a minimum data-set used at the following sub-sections.

A top-level entity *project* (an entity, an attribute or a type in the *STEP2M* model) contains two objects called *workpiece* and *workplan*. *Workpiece* is a data object to represent parametric geometry and material property of a raw material.

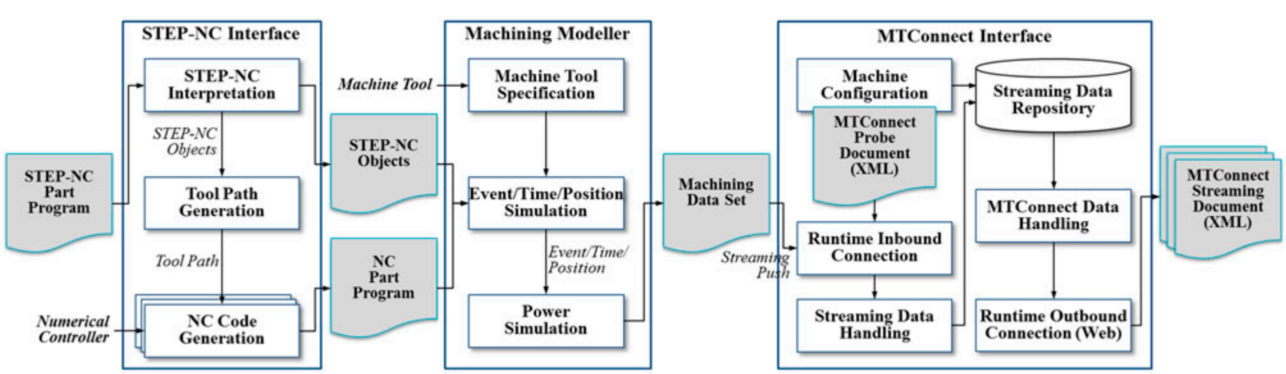


Figure 1. A logical flow of the *STEP2M* model.

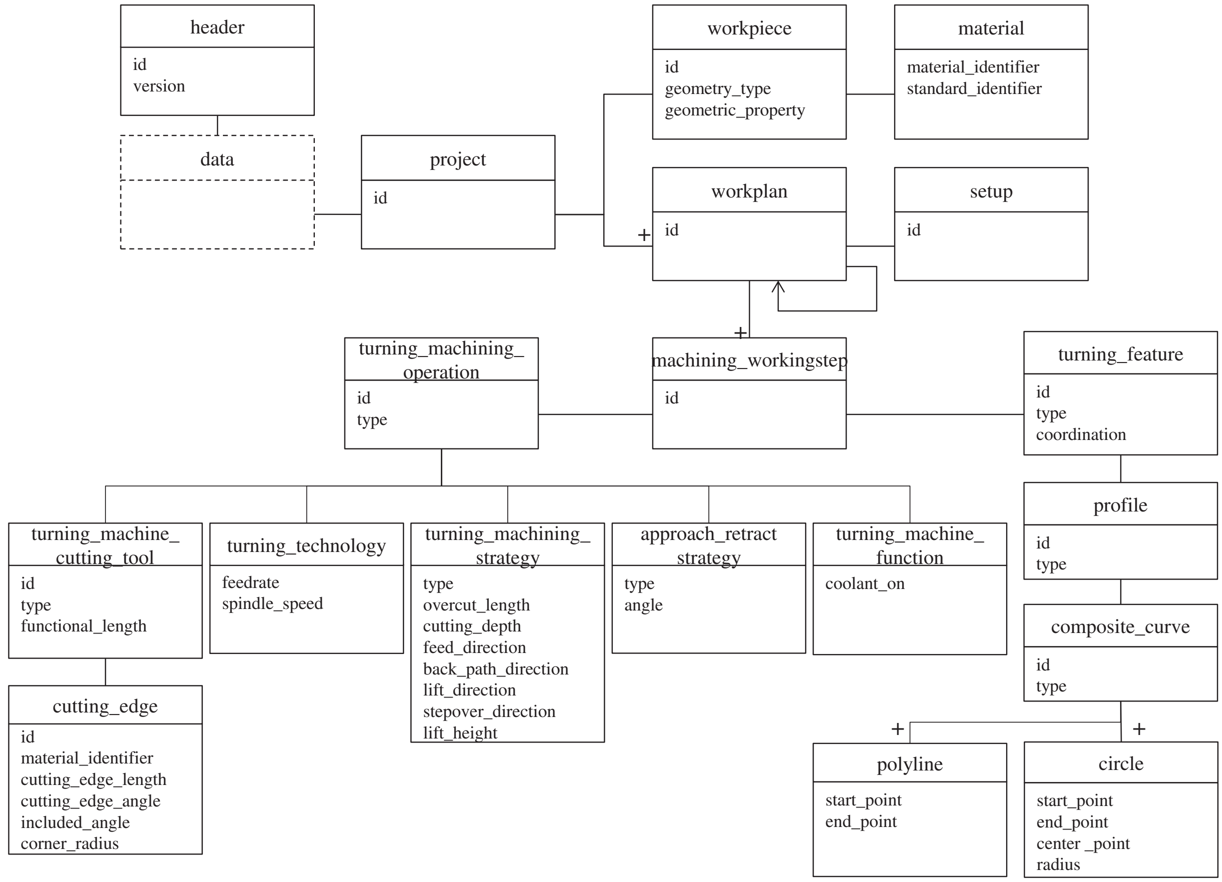


Figure 2. A STEP-NC object model for a turning operation.

Workplan combines a series of object called *machining_workingstep* in a linear order (ISO 2004a). *Machining_workingstep* consists of two objects – *turning_feature* and *turning_machining_operation* (ISO 2005).

Turning_feature is a data object that expresses the volume removed from a workpiece, i.e. what-to-make. This object describes a desired shape manufactured by a machining operation. The object constitutes a series of composite curves chosen in the form of either a polyline or a circle. *Turning_machining_operation* is another data object that describes the details of machining operations performed by the machine tool, i.e. how-to-make. This object specifies entities and attributes of *machining operation*, *cutting tool*, *machining technology*, *machining strategy*, *approach/retract strategy* and *machine function*.

3.2.2 *Tool path generation*

'Tool path generation' creates a tool path list, the so-called Cutter Location (CL) path list, by using STEP-NC objects. This CL path list is an aggregation of each sequential cutter position and its instruction (rapid or interpolation trajectory). The *STEP2M* model needs this function because tool path specification is 'optional' in the current STEP-NC (Shin, Suh, and Stroud 2007). The following items are the detailed STEP-NC objects necessary for creating the CL path:

- *workpiece: geometry_type, geometric_property*
- *feature: type, coordination*
- *profile: type*
- *composite_curve: type*
- *polyline: start_point, end_point*
- *circle: start_point, end_point, centre_point, radius*
- *operation: type*

- *turning_machine_cutting_tool: id, type*
- *cutting_edge: corner_radius*
- *turning_machining_strategy: type, overcut_length, cutting_depth, feed_direction, back_path_direction, lift_direction, stepover_direction, lift_height*
- *approach_retract_strategy: type, angle*

'Tool path generation' goes through the procedure as follows (Shin et al. 2009): (1) extract a removal delta volume from an original machining feature; (2) offset the delta volume by a corner radius of a cutting tool; (3) generate scan lines for actual cutting paths; (4) generate net cut paths based on offset lines and scan lines; (5) interlink net cut paths by approach/retract paths (paths to reach the first cut or to retreat after the last cut) and transition paths; and (6) connect a net cut path with the next net cut path.

A tool path object model represents the tool path generated. This object model defines five selective path types – *approach*, *retract*, *rapid*, *interpol_nocontact* and *interpol_contact*. *Approach*, *retract* and *rapid* types correspond with rapid trajectories which support actual cutting paths. *Approach* and *retract* are defined relative to *approach_retract_strategy*. The *rapid* type maps with lift, backward and stepover paths to connect actual cutting paths in a serial order. An interpolation trajectory is the tool path involved in actual cutting, and can have two types including *linear_interpolation* and *circular_interpolation*, depending on whether a cutting tool contacts with a workpiece or not. The parameter to separate the two paths is *overcut_length*, which is the distance that the tool travels through the air before contacting the workpiece. The reason of the separation comes from occurrence of cutting power during only *interpol_contact*, which is necessary for use in 'power simulation'.

3.2.3 *NC code generation*

An NC system requires an input of an NC part programme that codes a CL path list and its associated modal codes (e.g. spindle movement, feed movement, tool assignment and miscellaneous function). The 'NC code generation' function plays a role on a post-processor, and thus creates an NC part programme interpretable and executable by an NC system from the CL path list and the STEP-NC objects. The following items describe the necessary STEP-NC objects:

- *turning_technology: feedrate, spindle_speed*
- *turning_machine_function: coolant_on*
- *turning_machine_cutting_tool: id, type*

Figure 3 presents an example of the method to transform STEP-NC objects and a CL path list into an NC part programme in the case of an NC controller. We can transform the attributes of *feedrate* and *spindle_speed* in the *turning_technology* entity to F-code and S-code, respectively. *Coolant_on* in *turning_machining_function* corresponds to M-code (M08 turned on and M09 turned off). T-code indicating a tool number assigned on a turret can be extracted from an identification attribute of *turning_machine_cutting_tool*. After setting these modal codes, we map a coordinate of a tool path segment to a coordinate of a line segment of an NC part programme. *Rapid* tool path type matches with G00, *linear_interpolation* to G01, respectively. Clockwise *circular_interpolation* matches with G02 and counter clockwise *circular_interpolation* to G03, respectively.

As mentioned in Section 3.1, the selection of different NC systems needs multiple post-processors that define separate NC code schemes available to CNC machines. Each post-processor maps the STEP-NC objects to the designated code scheme, but the mapping rule described above is identical.

Compared with a STEP-NC part programme (see the STEP-NC part programme in Figure 3), an NC part programme inevitably loses much process plan information. As shown in the NC code (see the NC part programme in Figure 3), it is difficult to figure out process sequence and parameter selection directly. For example, we can determine cutting depth directly from a STEP-NC programme. Meanwhile, we need to infer the cutting depth from the pitch of axial movements in the NC code. This data loss shows us why process planning data are better input data than those of an NC programme.

3.3 *Machining modeller*

An NC part programme has a static feature that is independent of the time domain. On the other hand, machine-monitoring data are dynamic, time-sensitive and related to a machine tool's actions. Thus, the static data should be transited to the dynamic data in order to generate machine-monitoring data from the input of an NC part programme. 'Machining modeller' simulates time and event according to the sequential execution of an NC part programme. In addition, it forecasts the events related to dynamics and kinematics of each machine component over time.

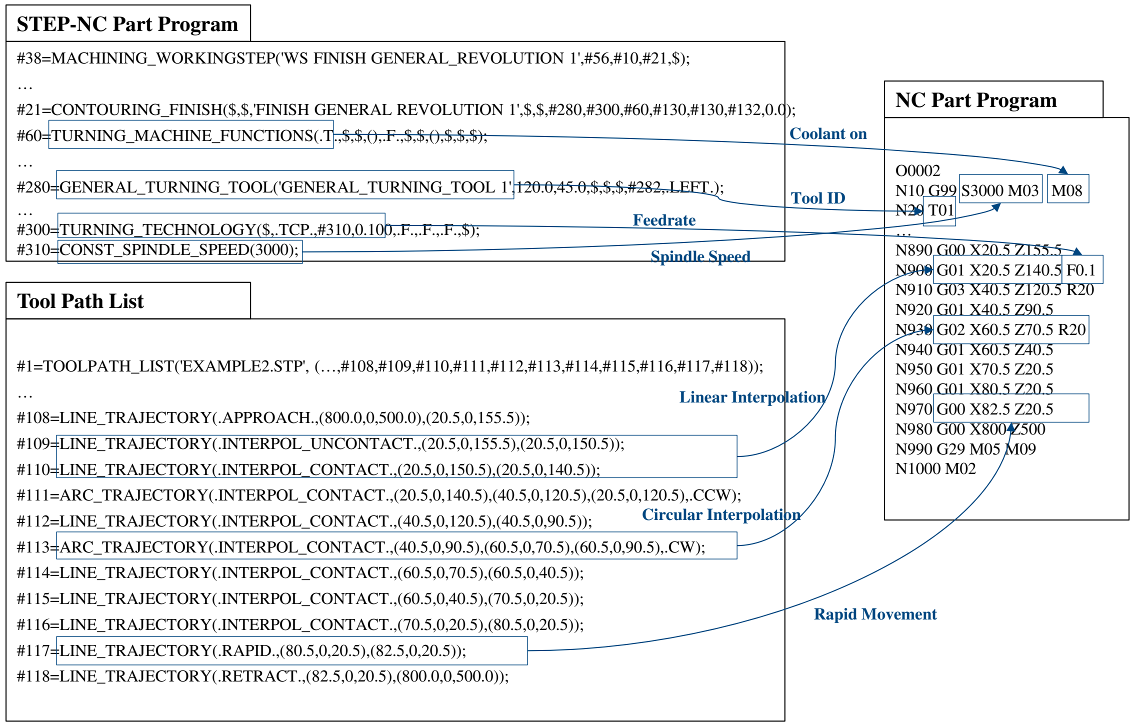


Figure 3. An example of NC code mapping.

The inputs to this module are an NC part programme, a machine tool specification and the collection of STEP-NC objects. The output is a data-set composed of time-series positions for linear (*Z*- and *X*-) axes components in a machine tool and time-series powers for relevant machine components including *Z*- and *X*- axes, a rotary axis, a coolant auxiliary system and a main body system of a machine tool. The Appendix 1 lists equations derived for and used in 'machining modeller'.

3.3.1 *Machine tool specification*

As mentioned in Section 3.1, there is a need to specify the capability and performance of a machine tool. Table 1 presents the properties of the machine tool specification for a CNC lathe. We can use these properties to calculate time-series positions and their associated power values for machine components.

3.3.2 *Event/time/position simulation*

The 'event/time/position simulation' function simulates events and their times of machine component's motions and then calculates axial positions as a function of time. 'Time' is a timestamp for the NC code execution, which enables point-in-time measurement of data items that are changing continuously. 'Event' contains not only discrete machine-state change, like coolant on-off and spindle on-off, but also axial-movement changes such as acceleration, steady state and deceleration. 'Position' describes a space-time coordinate data along two linear axes of the machine tool.

Figure 4(a) and (b), respectively, illustrate an example of an NC part programme and its tool movement. Figure 4(c) shows velocity profiles of the linear and rotary axial components of the tool movements shown at (1), (2) and (3) in Figure 4(b). The following descriptions explain the logic of event, time and position simulation.

3.3.2.1 *Event, time and position of linear (Z-/X-) axes.* We assume linear velocity has a trapezoidal profile when a tool is moving toward a designated point, as shown in Figure 4(c) (Avram and Xirouchakis 2011). A given feedrate NC code instruction assigns velocities. In the case of G00 codes such as (1) and (3), the velocity will match with a constant value defined in the machine tool specification. However, in the case of G01 code such as (2), the velocity will match with a moving speed defined by *feedrate*.

Table 1. The property list for machine tool specification.

Component	Property	Unit
Main body system	Basic power	W
Coolant system	Cooling power	W
Linear axis	Acceleration/deceleration coefficient	m/s^2
	Rapid moving speed	m/s
	Pitch of feed screw	m/rad
	Friction coefficient in guide	–
	Table mass	kg
	Gravity	m/s^2
	Friction coefficient in bearing	–
	Feed screw diameter	m
	Gear reduction ratio	–
	Lead screw mass	kg
	Coupling inertia	$kg*m^2$
	Acceleration of table	m/s^2
	Viscous damping coefficient	Nm/(rad/s)
	Pre-load force	N
	Servomotor efficiency	–
Rotary axis	Rotor diameter	mm
	Rotor length	mm
	Friction coefficient	Ns/mm^2
	Gap between rotor and stator	mm
	Application inertia	$kg*m^2$
	Rotor inertia	$kg*m^2$
	Acceleration/deceleration coefficient	rad/s^2
	Friction torque in the front bearing	Nm
	Friction torque in the rear bearing	Nm
	Spindle motor efficiency	–

We can calculate an acceleration/deceleration time from dividing a moving speed by an acceleration/deceleration coefficient (Appendix #1 and #2). The steady time is the time taken by the axes to travel the remaining distance subtracting the distances moved during acceleration and deceleration by the moving speed (#3). We can obtain the position along a linear axis by integrating velocity in time (#4).

3.3.2.2 *Event and time of rotary (C-) axis.* When a spindle rotation starts with the M03 code and ends with the M05 code as in Figure 4(a), it also assumes a trapezoidal velocity profile as in Figure 4(c). The spindle quickly reaches angular velocity specified by the S-code. We can calculate an acceleration/deceleration time by dividing the angular velocity by an angular acceleration/deceleration coefficient (#5 and #6). The steady time is the remaining time from the total time (#7).

3.3.2.3 *Event and time of coolant system.* A coolant system controls coolant injection and the circulation for its reuse. The coolant system is turned-on with an input of M08 code and turned-off with an input of M09 code, as shown in Figure 4(a). The coolant operation time is the difference between the turned-on and turned-off times.

3.3.2.4 *Event and time of main body system.* The main body system includes control unit, fans, and so on (Larek et al. 2011). Operation of a main body assumedly starts with an input of programme identification such as O0001 of Figure 4(a) and turns off with an input of M02 code as in Figure 4(a). The main body operation time is also the difference between the turned-on and turned-off times. The power consumption of the main body system excludes the power consumed by the individual axis motors during motion, but includes the power consumed by them in idling.

3.3.3 *Power simulation*

Power indicates the amount of energy consumed per unit time by the machine components. Because each machine component has its unique power requirement for its independent actions, 'power simulation' should simulate power consumption for each machine component. The simulation model for the power consumption of linear and rotary

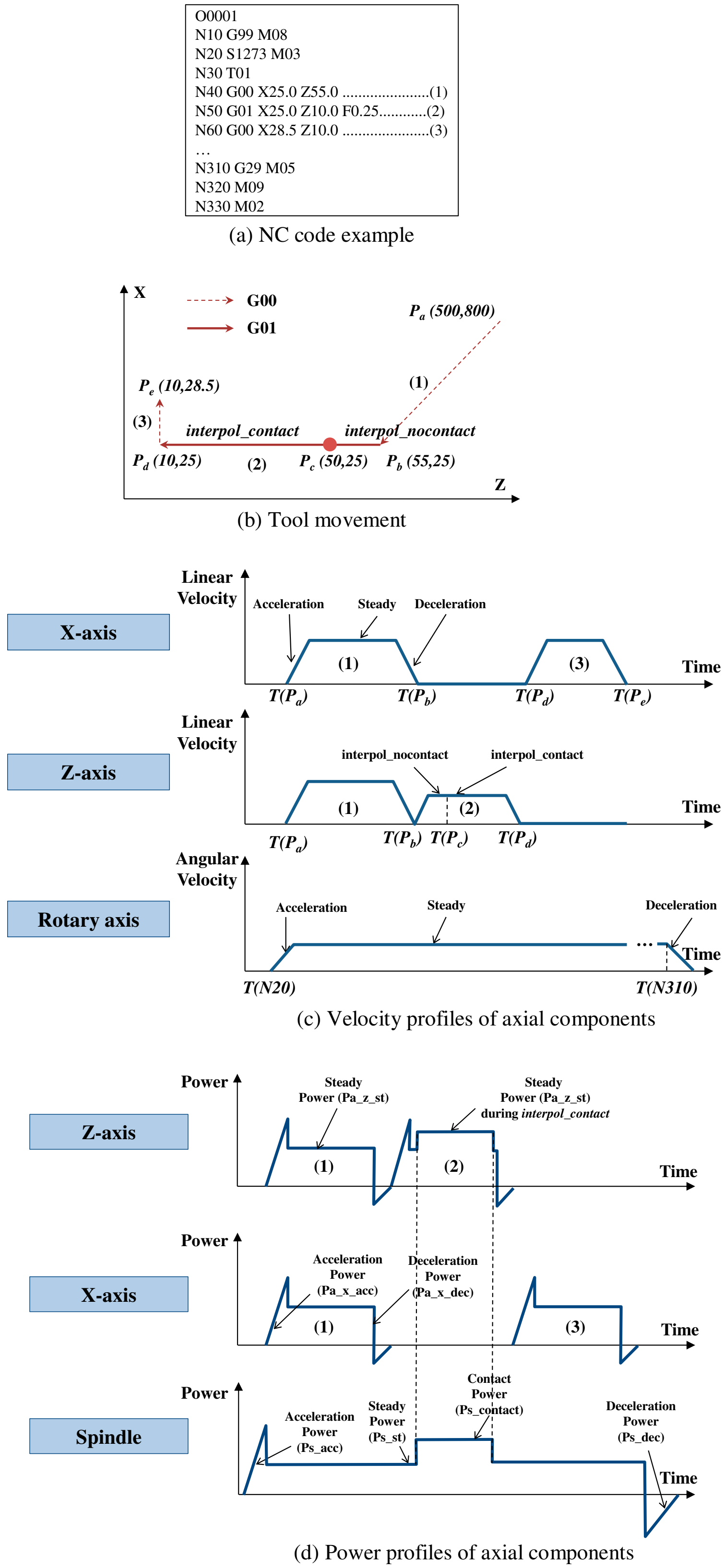


Figure 4. An example of NC code, its tool movement, velocity and power profiles.

motions adapts the basics of the metal cutting theory (Voss 2007; Avram and Xirouchakis 2011; Altintas 2012). The power consumed by the main body and coolant system will remain unchanged over time as defined in the machine tool specification.

3.3.3.1 *Power consumption for linear (Z-/X-) axis motion.* Power is required to drive the linear servomotors that control the linear motion along each axis. The power profile of a single NC code command for linear movement consists of acceleration, steady state and deceleration. As shown in Figure 4(d), the servomotor requires a steady torque to overcome the friction losses in guide ways, bearings and gears (#8). The servomotor requires enhanced power and torque during its acceleration and reduced power and torque during deceleration phases, as compared to that required for steady motion. The deceleration torque and power assumedly goes back to the power supply, and thus the deceleration power takes negative values (Avram and Xirouchakis 2011). Appendices #9 and #10 describe the details.

Power consumption during the steady state will vary for cutting and non-cutting motions as specified with the current NC code instruction. For example, the G00 instruction, shown at (1) and (3) in Figure 4(a), requires less power and torque than the G01 instruction, shown at (2) in Figure 4(a), which requires cutting force, feed force, and vertical force (#8). We use the Kienzle equation in the calculation of these forces (Klocke 2011). The machine tool specification assigns the power and torque for the non-cutting phase or air cutting. The power and torque during the phase will be greater by the amount required for the cutting forces, as explained in #8. The maximum power during acceleration/steady state/deceleration phases is the value obtained by multiplying the assigned torque with the angular velocity and dividing further by the servomotor's efficiency (#11). These maximum power levels can parameterize the time series of power consumption for a linear axis motion.

3.3.3.2 *Power consumption for rotary (C-) axis motion.* Mechanical factors that govern the power consumed by the spindle motor are its inertia, the type and size of bearings and their lubricants (Avram and Xirouchakis 2011). During spindle starting (enabled by M03 code), a spindle motor requires a large acceleration torque to accelerate a heavy spindle body to an assigned spindle speed (#12), and then sustains a steady torque to rotate at a constant speed (#13). The M05 demands a deceleration torque to stop spindle's rotation. The deceleration torque assumedly goes back to the power supply. During actual cutting, a steady power is the sum of the sustaining steady power (#14) and cutting power (#15). The power for a spindle motor is the product of the assigned torque with the angular velocity divided by the spindle motor's efficiency (#16).

The result of the 'machining modeller' module is the generation of a time-series data-set. For example, a resultant data-set includes time-series *Z*-/*X*- axes positions, *Z*-/*X*- axes power, spindle power, coolant power and main body power, corresponding to the NC part programme given in Figure 4(a). In addition, STEP-NC objects provide event and condition data items. For example, if *feed_direction* in the STEP-NC objects indicates (X, Y, Z) = (0, 0, –1), the direction of a *Z*-axis during the G01 code will be 'negative'. Finally, machine-monitoring data including all the samples, events and conditions are available to be inputted to the next 'MTConnect interface' module. We can append additional data items as necessary to measure machining performances. MTConnect allows this extensibility.

3.4 *MTConnect interface*

The machine-monitoring data-set delivered by 'machining modeller' is a data-set that records simulated values at different point in a machine tool's conditions. Based on the data-set, 'MTConnect interface' outputs an MTConnect-based streaming document when one issues a request. 'MTConnect interface' performs three tasks: (1) registration of machine specification, (2) runtime data collection and (3) MTConnect data request.

3.4.1 *Registration of machine specification*

This task registers the specification of a device – a machine tool – based on MTConnect Data Architecture. The architecture consists of devices, components and data items. Devices serve as the entities for representing existing machine tools. Devices can have data items subordinately associated with devices such as Availability, CommPort and Status, or components such as *X*-axis and *Z*-axis. Components represent sub-physical systems of the device, such as Axis, Controller, Sensor and Actuator. Data items are individual elements of information about the device or component, which include a type, an ID, a unit, a category and so on (AMT 2012b). Table 2 presents a list of the data items available from 'machining modeller'.

'Machine configuration' registers the specification of a device and its components as well as available data items. This function registers an MTConnect probe, which formalises the available data items, as mentioned in Section 2.2, on

Table 2. Available data items.

Data items	Type/Subtype	ID	Unit	Category
X_AXIS_POSITION_ACTUAL	POSITION/ACTUAL	x_axis_position_actual	Millimetre	Sample
X_AXIS_WATTAGE	WATTAGE	x_axis_wattage	Watt	Sample
Z_AXIS_POSITION_ACTUAL	POSITION/ACTUAL	z_axis_position_actual	Millimetre	Sample
Z_AXIS_WATTAGE	WATTAGE	z_axis_wattage	Watt	Sample
C_AXIS_WATTAGE	WATTAGE	c_axis_wattage	Watt	Sample
COOLANT_WATTAGE	WATTAGE	coolant_wattage	Watt	Sample
DEVICE_WATTAGE	WATTAGE	load_wattage	Watt	Sample

'streaming data repository'. The probe informs a client of data items that are available from the device. When the client issues a probe request, 'streaming data repository' delivers the probe to a client application through 'runtime outbound connection'. In addition, the probe can be used as an input to 'runtime inbound connection'. The data structure of the probe can also check whether the available data items are streaming-pushed from 'machining modeller'.

3.4.2 *Runtime data collection*

When a device is running, the device reports all the event, condition and sample data to the 'MTConnect interface' module. 'Runtime inbound connection' collects streaming data delivered from 'machining modeller'. 'Streaming data handling' translates the streaming data into the MTConnect-based streaming data registered by the MTConnect probe. 'Streaming data handling' delivers the MTConnect data to 'streaming data repository'. The data repository stores the MTConnect data to provide the data when a client issues a data request.

An important function of 'streaming data handling' is to assign a sequence number to each stream data line. The sequence number must be unique, starting from zero and assigned in monotonically increasing numbers as the data arrives at 'runtime inbound connection'. Figure 5 illustrates an example of streaming data flow from 'machining modeller' to 'streaming data repository' within 'MTConnect interface'. The streaming data are stored in the form of the schema defined in 'streaming data repository'. 'Streaming data handling' should translate each streaming datum into *DataItem* of a category such as *Event*, *Sample* and *Condition*.

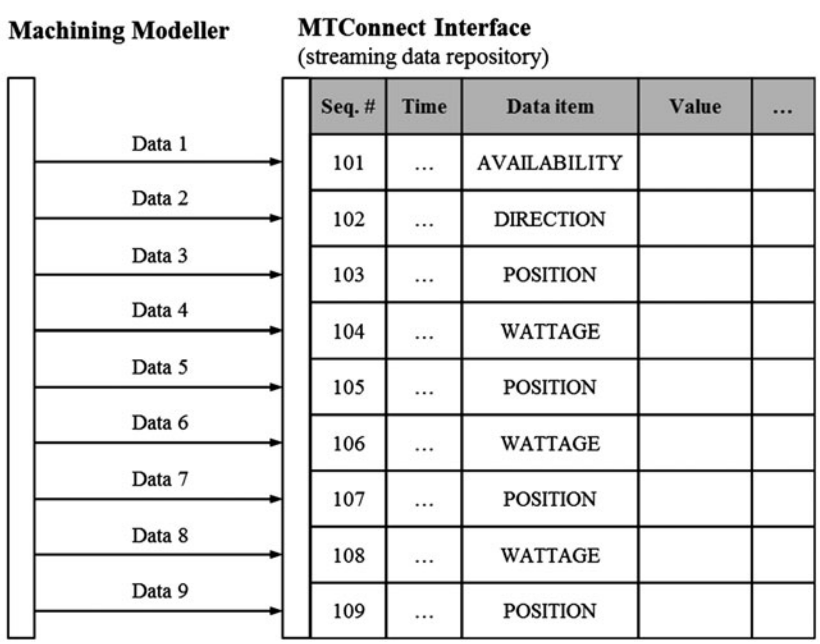


Figure 5. An example of streaming data flow.

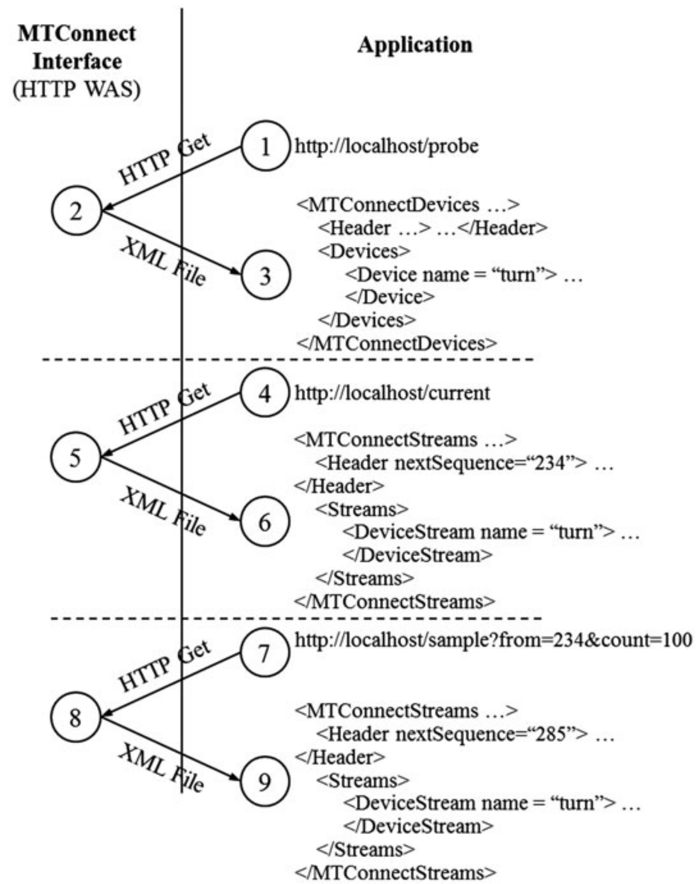


Figure 6. Communication between 'MTConnect interface' and an application.

3.4.3 *MTConnect data request*

When an client application issues a request, 'MTConnect data handling' collects the client-requested data from 'streaming data repository', formalises the data into an MTConnect document, and sends the document to 'runtime outbound connection'. Finally, 'runtime outbound connection' delivers an MTConnect document or a probe document to the client application.

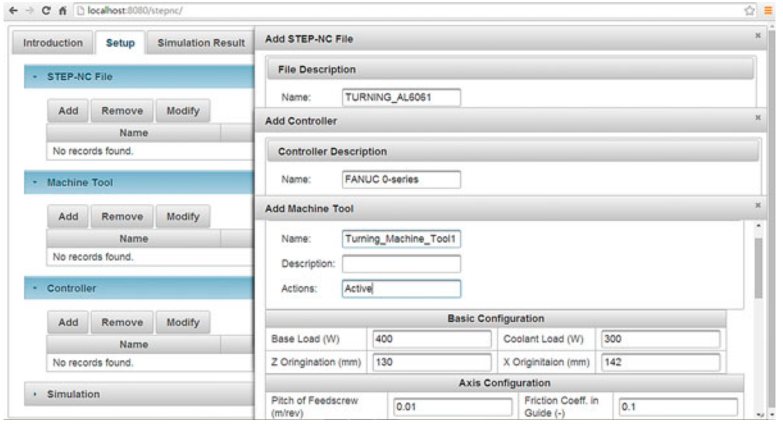
Figure 6 illustrates the communication between 'MTConnect interface' and a client. The 'MTConnect interface' generates an MTConnect document when a client calls *probe*, *current* and *sample* requests. When the client requests a *probe*, 'MTConnect interface' provides an MTConnect probe document, as shown in procedures 1 to 3 of Figure 6. When the client requests a *current*, this module retrieves the most recent values or states of the data item of the device at a single time step and records these in an MTConnect streaming document, as in procedures 4 to 6 of Figure 6. When the client requests a *sample*, this module retrieves *Samples*, *Events* and *Conditions* as a time series and records these in an MTConnect streaming document, as in procedures 7 to 9 of Figure 6.

4. Implementation

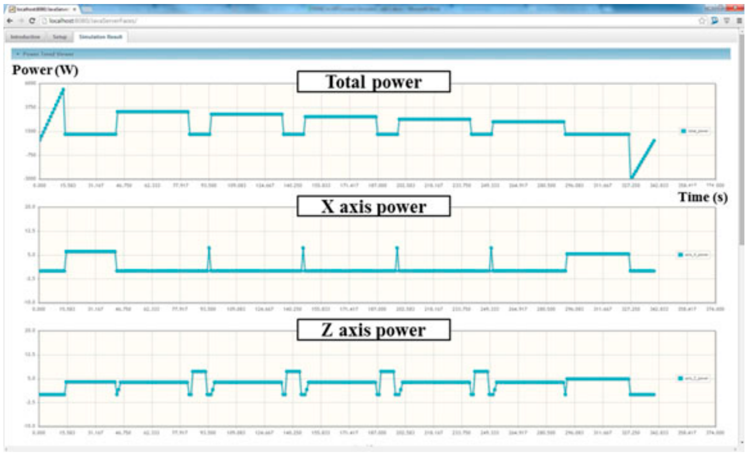
We have developed a prototype system to validate feasibility and performance of the *STEP2M* model for a 2-axis turning operation. The prototype system uses JAVA as a programming language, Prime Face as a web interface and Tomcat as an MTConnect server, respectively. Section 4.1 describes STEP-NC part programming, machine tool specification and the resulting machining simulation. Section 4.2 explains the generation of an MTConnect streaming document. Section 4.3 presents the result of validation. Section 4.4 discusses the result of the implementation.

4.1 *Process planning & machining simulation*

Figure 7(a) presents the user interface for choosing a STEP-NC part programme as an input, an NC system and a machine tool specification. The STEP-NC part programme, for example, would assign aluminium alloy (AL6061) as



(a) A STEP-NC part program input and a machine tool specification



(b) Line charts for machine components' powers

(c) An MTConnect XML file

Figure 7. Implementation results of the *STEP2M* prototype system.

material_identifier, *general_revolution* as *turning_feature*, *contouring_rough* as *turning_machining_operation*, *unidirectional_turning* as *turning_machining_strategy*. In addition, the STEP-NC part programme would assign 0.25 mm/rev, 166.5 rad/s, 2.5 and 2.0 mm as main process parameters – *feedrate_per_revolution*, *spindle_speed*, *cutting_depth* and *overcut_length*, respectively. In addition, we select an NC system and identify a turning machine tool. We instantiate the machine specification properties of Table 1 for the identification of a turning machine tool. Figure 7(b) gives a screen shot of visual presentation of the simulated power of *X*-axis, *Z*-axis and total power over time.

4.2 *MTConnect streaming*

The prototype system pushes the streaming data that represent machine-monitoring data into ‘MTConnect interface’. The ‘MTConnect interface’ stores the streaming data in the form of the schema defined in ‘streaming data repository’. The schema conforms to the available monitoring data-set, as shown in Table 2.

A client application receives the probe document provided by the prototype from the MTConnect agent. The application interprets this probe and requests MTConnect streaming data from the agent. Figure 7(c) shows an MTConnect XML file retrieved by the ‘sample’ query. The ‘sample’ query allows the agent to give the client an MTConnect streaming document that contains the values during an indicated time interval.

4.3 *Comparison with experimental data*

To validate the *STEP2M* model, we have compared the power simulations of the *STEP2M* model with the power measurement in 2-axis turning operation by conducting an experiment in a CNC lathe. A power metre measures the summed power consumption by all the machine components. The process plan for the machining process matches the specifications of the STEP-NC part programme used in Section 4.1.

Figure 8 provides a time-series chart that plots the simulated and measured power over time. The trend of simulated power over time matches with measured power. We can analyse the five discrete points marked in Figure 8 as follows:

(1) When a spindle starts rotation, a momentary power peak occurs due to the spindle’s quick response.
(2) A coolant power is added when a coolant system is turned-on.
(3) Simulated power is different from the measured power at the collision point between a tool insert and a workpiece because measured power consumes a momentary power by the restricted force that occurs in the collision of the two metals.
(4) Cutting power is consumed when a tool insert contacts a workpiece, but gradually decreases as the workpiece diameter decreases.

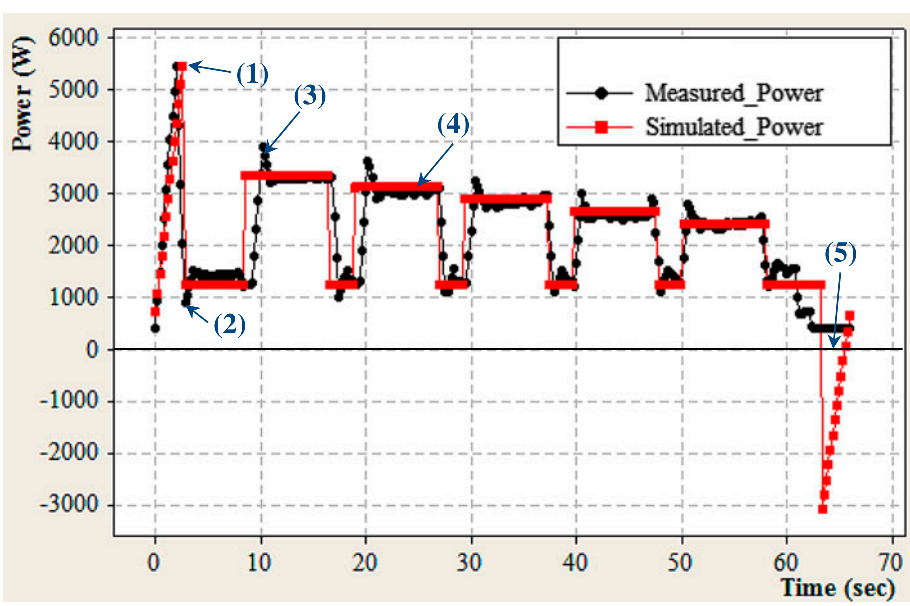


Figure 8. A line chart for measured power and simulated power.

(5) Simulated power takes a large negative value when spindle stops which is quite different from the measured transient response of the experiments. The reason for this is due to the assumption in 'machining modeller'. In actual machining case, the spindle decelerates gradually due to bearing friction and therefore the spindle motor does not consume power. Our 'machining modeller', on the other hand, specifies consumption of power to assure the quick stop of the heavy spindle.

4.4 Discussion

(1) Extensibility towards multiple metrics: the *STEP2M* model first makes an attempt to generate a time-series power profile on a machine because power consumption is an important metric for energy-conscious process planning. Power consumption closely relates to calculate major machining performances such as machining time and energy consumption. For example, machining time is the duration between power turned-on and turned-off, as shown in Figure 8, and energy consumption is the integral of power with time. Also, it helps obtain the relevant metrics including position, feed, force and torque because these metrics are essential to calculate power consumption, as shown in Appendix 1. We can also include other metrics such as temperature, noise and electrical resistance. For this case, we need to develop physical models necessary for simulating target metrics, similarly with the model for power consumption. We then can pile these models in the location of 'power simulation' in Figure 1. As mentioned in Section 3.1, our modular approach supports the extensibility towards multiple metrics.

(2) Extensibility towards multi-axis machining: we can extend the *STEP2M* model for 2.5-Dimensional milling machines as well as Multi-Channel Complex Machines (MCCM), which supports turn-mill machining with synchronised machining modes including one feature simultaneous, two feature simultaneous and parallel machining (Shin et al. 2009). For 2.5D milling machines, we need to develop the STEP-NC interface and the power model for milling, and add *y*-axial available data items in the MTConnect probe document. We can develop the STEP-NC interface for milling by using the STEP-NC data model for milling (ISO 2004b). Several research works have successfully demonstrated the implementation of milling STEP-NC interfaces (Xu, Wang, and Newman 2011). In addition, open source tools like ISO 14649 Tool Kit help easy implementation of the milling interface (NIST 2009). Developing the milling power model differs in the additional consideration of the number of cutters, cutting width and up- or down- milling. The two former will make different cutting force distributions in the 3D coordinate; whereas, the latter up- or down-milling makes an opposite cutting force profile (Altintas 2012). The 3D coordinate system demands the addition of *y*-axial available data items in the MTConnect probe document to represent *y*-axial position and power values. We are currently developing the *STEP2M* model for milling by this approach.

The MCCM should consider the synchronised modes and the slope machining where milling features in a turn-mill part are defined on arbitrary planes on the 3D coordinate system. Both turning and milling STEP-NC interfaces are required to convert turning and milling STEP-NC objects into several NC programmes that are separately assigned to multiple turrets. The integration of the two STEP-NC interfaces has been introduced by the previous work in (Shin et al. 2009). We then need to apply the turning and milling machining modellers to calculate values of event, time, position and power, which correspond to each NC programme assigned to an individual turret. The summation of the power values at each timestamp derives the time-series power profile on the MCCM. For the slope machining, a transfer matrix considering tool rotational angle transforms the local coordinate of the milling feature to calculate power values on the XY plane (Shin et al. 2009). Then its reverse matrix retransforms the calculated values on the local coordinate of the milling feature.

5. Conclusion

This paper presented the efforts to develop the virtual machining model that generates machine-monitoring data directly from process planning data. The *STEP2M* model used standardised data interfaces – STEP-NC and MTConnect – to represent process planning and machine-monitoring data. The *STEP2M* model described in this paper allows one to: (1) predict machining performance during the process planning stage, and (2) through standardised interfaces, perform interoperable data communication in an open data sharing environment. The *STEP2M* model results showed its applicability with conducting an experiment on a CNC lathe.

The MTConnect data generated through the *STEP2M* model finds its usefulness in generating data-sets with different input conditions necessary for performing data analytics. Examining and drawing conclusions about raw data is the key to data analytics for enhancing machining performance. The virtual but universal measurement device developed in this paper can provide the large-size raw data necessary for data analytics and eliminate the need for large number of experiments with substantial reduction in cost and effort in data collection. Additionally, the use of STEP-NC and MTConnect data is valuable in data analytics because it helps discover meaningful relations in cause and effect factors. The data content of STEP-NC covers cause factors that relate to process planning. On the other hand, the data content of MTConnect includes fundamental machine-monitoring data to identify effect factors. Thus, the *STEP2M* model can provide a pairwise set of cause and effect data to support efficiently data analytics modelling works. We can use these pairwise data-sets to create numerical relationships between the cause and effect data. These models enable manufacturers to respond proactively by identifying anticipated measured values. We then apply these numerical models into predictive control and optimisation to improve machining performances. Using the *STEP2M* model, we are currently developing predictive and optimisation models that can make better machining performances in machining (Shin, Woo, and Rachuri 2014).

Some limitations of this paper are that the current model can only simulate power and position for 2-axis turning machining, and that the model does not account for disturbances and disruptions as well as other transient responses such as vibration, noise and tool wear. Future work includes the extension of virtual machining models toward multi-axis machining, and the improvement of virtual machining models to accommodate spindle deceleration and other transient responses of a machine tool.

Disclosure statement

Any mention of commercial products is for information only; it does not imply NIST recommendation or endorsement, nor does it imply that the products mentioned are necessarily the best available for the purpose.

References

Altintas, Yusuf. 2012. *Manufacturing Automation: Metal Cutting Mechanics, Machine Tool Vibrations, and CNC Design*. Cambridge: Cambridge University Press.

AMT (The Association for Manufacturing Technology). 2011. *MTConnect Standard Part 1 – Overview and Protocol*. McLean, VA: The Association for Manufacturing Technology.

AMT (The Association for Manufacturing Technology). 2012a. *MTConnect Standard Part 2 – Components and Data Items*. McLean, VA: The Association for Manufacturing Technology.

AMT (The Association for Manufacturing Technology). 2012b. *MTConnect Standard Part 3 – Streams, Events, Samples and Condition*. McLean, VA: The Association for Manufacturing Technology.

Avram, Oliver Ioan, and Paul Xirouchakis. 2011. "Evaluating the Use Phase Energy Requirements of a Machine Tool System." *Journal of Cleaner Production* 19: 699–711.

Bhushan, Rajesh Kumar. 2013. "Optimization of Cutting Parameters for Minimizing Power Consumption and Maximizing Tool Life during Machining of Al Alloy SiC Particle Composites." *Journal of Cleaner Production* 39: 242–254.

Braun, S., and U. Heisel. 2012. "Simulation and Prediction of Process-oriented Energy Consumption of Machine Tools." Paper presented at 19th CIRP International Conference on Life Cycle Engineering, Berkeley, May 23–25.

Diaz, Nancy, Annabel Jondral, Sebastian Greinacher, David Dornfeld, and Gisela Lanza. 2013. "Assessment of Lean and Green Strategies by Simulation of Manufacturing Systems in Discrete Production Environments." *CIRP Annals – Manufacturing Technology* 62: 475–478.

Dietmair, Anton, Alexander Verl, and Philipp Eberspaecher. 2009. "Predictive Simulation for Model Based Energy Consumption Optimisation in Manufacturing System and Machine Control." Paper presented at 19th International Conference on Flexible Automation and Intelligent Manufacturing, Middlesbrough, July 6–8.

Iqbal, Asif, Hong-Chao Zhang, Lu Lu Kong, and Ghulam Hussain. 2013. "A Rule-based System for Trade-off among Energy Consumption, Tool Life, and Productivity in Machining Process." *Journal of Intelligent Manufacturing*. doi:10.1007/s10845-013-0851-x.

ISO (International Standards Organization). 2003. *ISO 14649: Industrial Automation Systems and Integration – Physical Device Control – Data Model for Computerized Numerical Controllers, Part1: Overview and Fundamental Principles*. Geneva: International Standards Organization.

ISO (International Standards Organization). 2004a. *ISO 14649: Industrial Automation Systems and Integration – Physical Device Control – Data Model for Computerized Numerical Controllers, Part 10: General Process Data*. Geneva: International Standards Organization.

ISO (International Standards Organization). 2004b. *ISO 14649: Industrial Automation Systems and Integration – Physical Device Control – Data Model for Computerized Numerical Controllers, Part 11: Process Data for Milling*. Geneva: International Standards Organization.

ISO (International Standards Organization). 2005. *ISO 14649: Industrial Automation Systems and Integration – Physical Device Control – Data Model for Computerized Numerical Controllers, Part 12: Process Data for Turning*. Geneva: International Standards Organization.

Kadir, Abdul, Aini Zuhra, Xun Xu, and Firman Ridwan. 2011. “Employing Actual Shop-floor Status and Machine Tool Kinematic Data Model in Machining Simulation.” *International Journal of Integrated Engineering* 3 (1): 32–37.

Kant, Girish, and Kuldip Singh Sangwan. 2014. “Prediction and Optimization of Machining Parameters for Minimizing Power Consumption and Surface Roughness in Machining.” *Journal of Cleaner Production* 83: 151–164.

Kara, S., and W. Li. 2011. “Unit Process Energy Consumption Models for Material Removal Processes.” *CIRP Annals – Manufacturing Technology* 60: 37–40.

Klocke, Fritz. 2011. *Manufacturing Processes 1: Cutting*. Berlin: Springer-Verlag.

Larek, Roland, Ekkard Brinksmeier, Daniel Meyer, Thorsten Pawletta, and Olaf Hagendorf. 2011. “A Discrete-event Simulation Approach to Predict Power Consumption in Machining Processes.” *Production Engineering* 5 (5): 575–579.

Li, Li, Fei Liu, Bing Chen, and Cong Bo Li. 2013. “Multi-objective Optimization of Cutting Parameters in Sculptured Parts Machining Based on Neural Network.” *Journal of Intelligent Manufacturing*. doi:10.1007/s10845-013-0809-z.

Mativenga, P. T., and M. F. Rajemi. 2011. “Calculation of Optimum Cutting Parameters Based on Minimum Energy Footprint.” *CIRP Annals – Manufacturing Technology* 60: 149–152.

Muchiri, P., and L. Pintelon. 2008. “Performance Measurement Using Overall Equipment Effectiveness (OEE): Literature Review and Practical Application Discussion.” *International Journal of Production Research* 46 (13): 3517–3535.

Newman, S. T., A. Nassehi, R. Imani-Asrai, and V. Dhokia. 2012. “Energy Efficient Process Planning for CNC Machining.” *CIRP Journal of Manufacturing Science and Technology* 5: 127–136.

NIST (National Institute of Standards and Technology). 2009. *ISO 14649 Tool Kit*. Accessed January 09, 2015. https://code.google.com/p/iso-14649-toolkit/

Ridwan, Firman, and Xun Xu. 2013. “Advanced CNC System with in-process Feed-rate Optimisation.” *Robotics and Computer-Integrated Manufacturing* 29: 12–20.

Shao, Guodong, Deogratias Kibira, and Kevin Lyons. 2010. “A Virtual Machining Model for Sustainability Analysis.” Paper presented at ASME 2010 International Design Engineering Technical Conference & Computers and Information in Engineering Conference, Quebeck, August 15–18.

Shin, Seung-Jun, Suk-Hwan Suh, and Ian Stroud. 2007. “Reincarnation of G-code Based Part Programs into STEP-NC for Turning Applications.” *Computer-Aided Design* 39: 1–16.

Shin, Seung-Jun, Jumyung Um, Joo-Sung Yoon, Suho Jeong, Jae-Min Cha, Suk-Hwan Suh, and Dae-Hyuk Chung. 2009. “Developing ISO 14649-based Conversational Programming System for Multi-channel Complex Machine Tools.” *International Journal of Computer Integrated Manufacturing* 22 (6): 562–575.

Shin, Seung-Jun, Jungyub Woo, and Sudarsan Rachuri. 2014. “Predictive Analytics Model for Power Consumption in Manufacturing.” *Procedia CIRP* 15: 153–158.

Velchev, Stefan, Ivan Kolev, Krasimir Ivanov, and Simeon Gechevski. 2014. “Empirical Models for Specific Energy Consumption and Optimization of Cutting Parameters for Minimizing Energy Consumption during Turning.” *Journal of Cleaner Production* 80: 139–149.

Vijayaraghavan, Athulan, and David Dornfeld. 2010. “Automated Energy Monitoring of Machine Tools.” *CIRP Annals – Manufacturing Technology* 59: 21–24.

Vijayaraghavan, Athulan, Will Sobel, Armando Fox, David Dornfeld, and Paul Warndorf. 2008. “Improving Machine Tool Interoperability Using Standardized Interface Protocols: MTConnect.” Paper presented at 2008 International Symposium on Flexible Automation, Atlanta, June 23–26.

Voss, Wilfried. 2007. *A Comprehensible Guide to Servo Motor Sizing*. Greenfield, MA: Copperhill.

Wang, Hongqiang, Xun Xu, and J. Des Tedford. 2007. “An Adaptable CNC System Based on STEP-NC and Function Blocks.” *International Journal of Production Research* 45 (17): 3809–3829.

Xu, Xun, H. Wang, J. Mao, S. T. Newman, T. R. Kramer, F. M. Proctor, and J. L. Michaloski. 2005. “STEP-Compliant NC Research: The Search for Intelligent CAD/CAPP/CAM/CNC Integration.” *International Journal of Production Research* 43 (17): 3703–3743.

Xu, Xun, Lihui Wang, and Stephen T. Newman. 2011. “Computer-aided Process Planning – A Critical Review of Recent Developments and Future Trends.” *International Journal of Computer Integrated Manufacturing* 24 (1): 1–31.

Yan, Jihong, and Lin Li. 2013. “Multi-objective Optimization of Milling Parameters – The Trade-offs between Energy, Production Rate and Cutting Quality.” *Journal of Cleaner Production* 52: 462–471.

Zhao, Yaoyao F., Salah Habeeb, and Xun Xu. 2009. “Research into Integrated Design and Manufacturing Based on STEP.” *The International Journal of Advanced Manufacturing Technology* 44: 606–624.

Appendix 1

Number	Component/ domain	Item	Formula	Variable
1	Linear/time	Acceleration time (t_{L_ACC}, sec)	$t_{L\_ACC} = \frac{v_i}{a_{L\_ACC}}$	a_{L_ACC} (m/s^2): linear acceleration coefficient v (m/s): moving speed i: selection of x-axis or z-axis
2	Linear/Time	Deceleration time (t_{L_DEC}, sec)	$t_{L\_DEC} = \frac{v_i}{a_{L\_DEC}}$	a_{L_DEC} (m/s^2): deceleration coefficient
3	Linear/Time	Steady time (t_{L_ST}, sec)	$t_{L\_ST} = \frac{1}{v_i}\left(L_i - \frac{1}{2}v_i(t_{L\_ACC} + t_{L\_DEC})\right)$	L (mm): tool path distance
4	Linear/Position	Length from a start point (L, mm)	$\text{if } 0 \le t < t_{L\_ACC},\ L_i(t) = \frac{1}{2}a_{L\_ACC}t^2$ $\text{else if } t_{L\_ACC} \le t < t_{L\_ACC} + t_{L\_ST},\ L_i(t) = \frac{1}{2}a_{L\_ACC}t_{L\_ACC}{}^2 + v_i(t - t_{L\_ACC})$ $\text{else if } t_{L\_ACC} + t_{L\_ST} \le t \le t_{L\_ACC} + t_{L\_ST} + t_{L\_DEC},\ L_i(t) = \frac{1}{2}a_{L\_ACC}t_{L\_ACC}{}^2 + v_i t_{L\_ST} + \frac{T}{2}(2v_i - a_{L\_ACC}T)$	t (s): timestamp $T = t - t_{L\_ACC} - t_{L\_ST}$
5	Rotary/Time	Acceleration time (t_{R_ACC}, sec)	$t_{R\_ACC} = \frac{\lvert w_s - w_{s-1}\rvert}{a_{R\_ACC}}$	a_{R_ACC} (m/s^2): rotary acceleration coefficient w_s (rad/s): angular velocity
6	Rotary/Time	Deceleration time (t_{R_DEC}, sec)	$t_{R\_DEC} = \frac{\lvert w_{s+1} - w_s\rvert}{\lvert a_{R\_DEC}\rvert}$	a_{R_DEC} (m/s^2): deceleration coefficient
7	Rotary/Time	Steady time (t_{R_ST}, sec)	$t_{R\_ST} = t_{TOT} - t_{R\_ACC} - t_{R\_DEC}$	$t_{TOT} = \sum_{i=1}^{N} t_{line,i}$ N: number of NC part program lines before another S-code appears
8	Linear/Power	Steady torque (T_{sr}, Nm)	$F_z = k_{z\,1.1}h^{(1-m_z)}$ $F_f = k_{f\,1.1}h^{(1-m_f)}$ $T_{gf} = \frac{h_p \times \mu_{gf} \times [(m_t + m_w) \times g + F_z]}{2\pi}$ $T_{lf} = \frac{\mu_b \times d_p \times (F_f + F_p)}{2}$ $T_f = \frac{h_p \times F_f}{2\pi}$ $T_{sr} = \frac{T_{gf} + T_{lf} + T_f}{r_g}$	h_p (m/rad): pitch of feeds crew μ_{gf}: friction coefficient in guide m_t (kg): table mass m_w (kg): workpiece mass g (m/s^2): gravity F_z (N): vertical force ($k_{z1.1}$: coefficient, $1 - m_z$: exponent, h: chip thickness = cutting depth) F_f (N): feed force ($k_{f1.1}$: coefficient, $1 - m_f$: exponent, h: chip thickness = cutting depth) F_p (N): pre-load force μ_b: friction coefficient in bearing d_p (mm): feed screw diameter T_{gf}: torque friction in a guide way T_{lf}: torque lost in bearing T_f: torque by the cutting force r_g: gear reduction ratio

(*Continued*)

Appendix 1. (*Continued*)

Number	Component/ domain	Item	Formula	Variable
9	Linear/Power	Acceleration torque (T_{d_acc}, Nm)	$T_{d\_acc} = J_e \times \frac{dw}{dt} + B \times w + T_{sr}$	J_e (kg/m^2): summation of inertia dw/dt (rad/s^2): angular acceleration B (Nm s/rad): viscous damping coefficient w (rad/s): angular velocity
10	Linear/Power	Deceleration torque (T_{d_dec}, Nm)	$T_{d\_dec} = -J_e \times \frac{dw}{dt} + B \times w - T_{sr}$	
11	Linear/Power	Power (P_{ax}, W)	$P_{ax\_acc} = T_{d\_acc} \times \frac{w}{\eta}$ $P_{ax\_st} = T_{sr} \times \frac{w}{\eta}$ $P_{ax\_dec} = T_{d\_dec} \times \frac{w}{\eta}$	η: servomotor efficiency
12	Rotary/Power	Acceleration torque (T_{a_sp}, Nm)	$T_{a\_sp} = (J_{\text{app}} + J_{\text{rotor}}) \times 2\pi\alpha$	J_{app} (kgm^2): application inertia J_{rotor} (kgm^2): rotor inertia α (rev/s^2): angular acceleration
13	Rotary/Power	Steady torque (T_{run}, Nm)	$T_{\text{shear}} = \frac{\pi^3 D^3_{\text{rotor}} L_{\text{rotor}} F_{r\_\text{coeff}} N}{60 \times gap}$ $T_{\text{run}} = T_{\text{frfb}} + T_{\text{frrb}} + T_{\text{shear}}$	D_{rotor} (mm): rotor diameter L_{rotor} (mm): rotor length F_{r_coeff} (Ns/mm^2): friction coefficient N (rad/s): spindle speed gap (mm): gap between rotor and stator T_{shear} (Nm): shear torque T_{frfb} (Nm): friction torque in the front bearing T_{frrb} (Nm): friction torque in the rear bearing
14	Rotary/Power	Steady power (P_{s_st}, W)	$P_{s\_st} = \frac{T_{\text{run}} w}{\eta}$	w (rad/s): angular velocity η: spindle motor efficiency
15	Rotary/Power	Cutting power ($P_{s\_contact}$, W)	$F_c = k_{c\,1.1} h^{(1-m_c)}$ $P_{s\_\text{contact}} = \frac{2\pi F_c v_c}{\eta}$	F_c (N): resultant cutting force ($k_{c1.1}$: coefficient, $1-m_c$: exponent, h: chip thickness = cutting depth) v_c (mm·rad/min): tangential cutting speed
16	Rotary/Power	Power (P_s, W)	$P_{s\_acc} = \frac{(T_{a\_sp} + T_{\text{run}})w}{\eta}$ $P_{s\_cut} = P_{s\_st} + P_{s\_\text{contact}}$ $P_{s\_dec} = \frac{(-T_{a\_sp} + T_{\text{run}})w}{\eta}$	