Chapter XIII
Development of an Integrated, Adaptable CNC System

ABSTRACT

In order to prepare manufacturing companies to face increasingly frequent and unpredictable market changes with confidence, there is a recognized need for CNC machine tools to be further advanced so that they become more integrated with design models and adaptable to uncertain machining conditions. For a CNC system to be able to access any design information, this design information has to be at the task-level, that is what-to-do. For a CNC system to produce the final part, it has to turn the task-level information into method-level information which effectively is the machine control data. These topics are discussed at the beginning of this chapter.

The rest of the chapter discusses a CNC native database used for converting the task-level data to method-level data, the methodology of converting the task-level data to method-level data, and implementation of the methodology to a conventional CNC machine that employs G-codes. Again both STEP-NC (ISO 14649-1, 2003) and function blocks (IEC 61499, 2005) are used.

TASK-LEVEL DATA VS. METHOD-LEVEL DATA

An important feature of the STEP-NC concept is that of “machine tool independency”; this makes STEP-NC codes interoperable across various CNC systems. This is because a STEP-NC data model mainly captures the task-level or the what-to-do information. Although it is possible to define data at the method-level or the how-to-do level, such as machine tool
trajectory, the main aim of STEP-NC is to allow these decisions to be made by a STEP-NC-enabled controller. This way, STEP-NC part programs may be written once but can be used on different machine tools providing that the machine tool has the required process capabilities. Figure 13.1 shows these two categories of data defined in the STEP-NC data model. The first two columns depict the manufacturing task information. The process-level data describe abstract manufacturing tasks at the macro-level. The geometry-level data are represented in terms of manufacturing_features. The how-to-do data are also divided into two types. Machine tool core data spell out the manufacturing requirements. Machine tool auxiliary data are method-level data and they are in fact defined in the STEP-NC data model as optional data. Both categories of how-to-do data take different forms when different machine tools are used.

The central issue is therefore the ‘transition’ from the task-level data to the method-level, or from the what-to-do data to the how-to-do data. Since STEP-NC is utilized as a CNC machining data model, implementation of STEP-NC is effectively a process of adapting its data model for different CNC systems. This is illustrated in Figure 13.2.

Such a system may have three stages. First of all, a native version of STEP-NC program is generated based on the information in a generic STEP-NC program. Then, low-level, local NC commands can be generated based on the native STEP-NC information. This provides a direct interface with the targeted CNC machine, hence a “CAM-CNC transition”. This transition is intended to be hidden away from the user and ideally synchronised with the subsequent execution process. Finally, a STEP-NC enabled controller executes the STEP-NC program through the above native CNC commands.

**GENERATE A NATIVE STEP-NC PROGRAM**

The key in this phase is to “map” a generic STEP-NC program to a native one. In other words, the main task is to convert what-to-do information in a generic STEP-NC program,
Addressing Privacy in Traditional and Cloud-Based Systems
www.igi-global.com/article/addressing-privacy-in-traditional-and-cloud-based-systems/105484?camid=4v1a

Advanced Technologies for Transient Faults Detection and Compensation
www.igi-global.com/chapter/advanced-technologies-transient-faults-detection/69367?camid=4v1a