(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
14 February 2008 (14.02.2008)

PCT

(10) International Publication Number
WO 2008/019340 A2

(51) International Patent Classification: **Not classified**

(21) International Application Number: PCT/US2007/075280

(22) International Filing Date: 6 August 2007 (06.08.2007)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
60/821,481 4 August 2006 (04.08.2006) US

(71) **Applicant** *(for all designated States except US)*: **HURCO COMPANIES, INC.** [US/US]; One Technology Way, Post Office Box 68180, Indianapolis, IN 46268 (US).

(72) **Inventors; and**

(75) **Inventors/Applicants** *(for US only)*: **OKRONGLI, Stefan** [DE/US]; 2901 West 61st Street, Indianapolis, IN 46268 (US). **GORGOL, Robert, J.** [US/US]; 1721 Shorte Drive, Indianapolis, IN 46214 (US). **SHKILER, Aleksandr** [US/US]; 6383 Monitor Drive, Indianapolis, IN 46220 (US). **TRAICOFF, Gerald** [US/US]; 14927 Windmill Drive, Carmel, IN 46033 (US). **LANKALAPALLI, Kishore** [IN/US]; 13340 Sedgwick Lane, Westfield, IN 46074 (US).

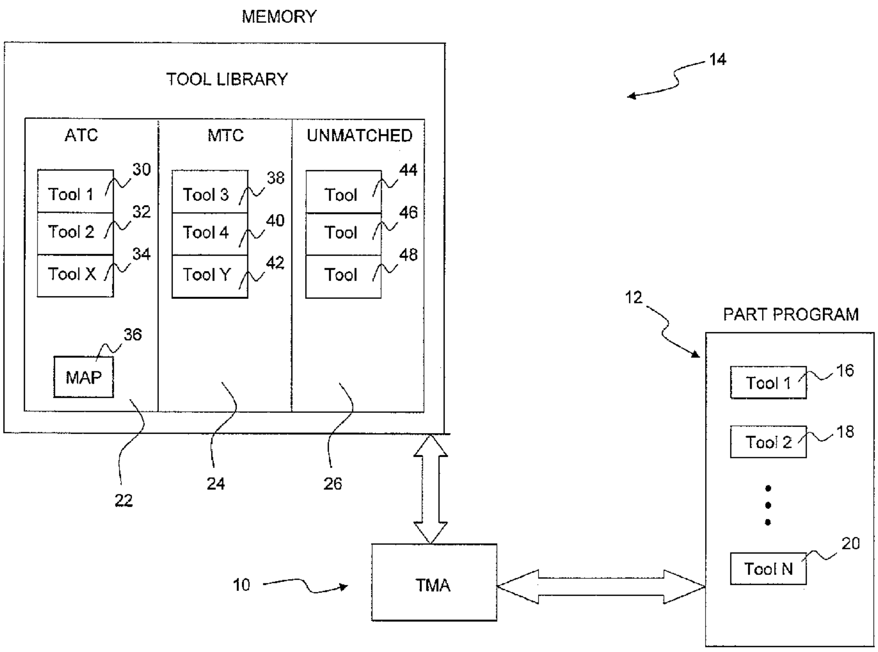
(74) **Agents: NULL, Robert, D.** et al.; Baker & Daniels LLP, c/o Robert D. Null, 300 N. Meridian Street, Suite 2700, Indianapolis, IN 46204 (US).

(81) **Designated States** *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BH, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RS, RU, SC, SD, SE, SG, SK, SL, SM, SV, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) **Designated States** *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IS, IT, LT, LU, LV, MC, MT, NL, PL, PT, RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:
— *without international search report and to be republished upon receipt of that report*

(54) **Title:** SYSTEM AND METHOD FOR TOOL USE MANAGEMENT



(57) **Abstract:** The present invention provides a system and method for tool use management wherein a CNC machine retains information identifying the tools associated with the machine as well as their current locations (if any), and executes an algorithm for determining the source tools needed by a part program and matching the source tools with the available tools.

WO 2008/019340 A2

SYSTEM AND METHOD FOR TOOL USE MANAGEMENT

Related Applications:

This application claims the benefit of U.S. Provisional Application Serial No. 60/821,481, filed on August 4, 2006, the entire disclosure of which is expressly incorporated herein by reference.

Field of the Invention:

The present invention relates generally to managing the use of tools loaded onto or associated with a computer numerically controlled ("CNC") machine, and more particularly to algorithms for matching tools required by a program for creating a machined part to tools available for use by a particular machine or at least described by data accessible by the machine.

Background of the Invention:

CNC machines use various tools (e.g., drills, end mills, reamers, taps, etc.) held by a movable, rotating spindle under the control of a program to form material such as metal into predetermined shapes or "parts." Often several different tools are required to create a part from stock material, each tool performing a function as specified by the part program. Many CNC machines have an associated automatic tool changer ("ATC") to speed the process of changing tools during execution of a part program. As is known in the art, such ATCs have many tool stations, each holding a particular tool which the ATC automatically indexes to a loading position for mounting to the spindle as required by the part program. Tools may also be changed manually. Generally, tools for manual loading that are used frequently by a particular machine are kept near the machine and are often mounted in tool holders. The locations of these tools for manual loading are collectively referred to as the manual tool changer ("MTC"). The various tools associated with a particular machine, either mounted in the spindle, mounted in the ATC, or available in the MTC, are referred to herein as the available tools.

Conventionally, to run a part program on a particular CNC machine, the operator had to determine from the part program the tools necessary to complete the part, and ensure that those tools were loaded, for example, in the ATC tool station locations expected by the part program. Alternatively, if a required tool was already loaded in the machine, but in a location different from that expected by the part program, the operator could modify the part program to reflect the actual location of the tool. Either way, the operator had to determine what tools were needed, identify the available tools, manually match the available tools with the needed tools, and correlate the actual locations of the available tools with the location designations of the needed tools specified by the part program. This process is slow and subject to error.

Summary of the Invention:

The present invention provides a system and method for tool use management wherein the CNC machine retains information identifying the available tools for the machine as well as their current locations, and executes an algorithm for determining the tools needed by the part program (hereinafter, "source tools") and matching the source tools with the available tools. Each machine's memory (or a distributed memory, as further described below) is populated with tool information (including the actual location of the tool) when the available tools are initially loaded onto, replaced, or associated with the machine, and the part program is modified based on that tool information to correlate the source tools with the available tools. Consequently, the part program may be executed on any of a plurality of machines, each having a different configuration of available tools, without manually changing the locations of the available tools or manually modifying the part program to reflect the locations of the available tools.

The machine's memory may also retain information describing tools that are not currently available at the machine. Source tools may be matched to these unavailable tools to perform virtual operations on the machine. Alternatively, the operator may be informed that a tool required by a part program is unavailable at the machine, but defined by information in the machine's memory. The operator may then obtain the tool and add it to the available tools for the machine. The data maintained in the machine's memory describing the available tools and the defined, but unavailable tools (collectively referred to herein as "defined tools") includes information about the physical characteristics and/or use of the tools. Throughout this description, this collection of data is referred to as "the tool library."

In another embodiment of the invention, source tools that are not identically matched to defined tools are identified as unmatched tools. These unmatched tools may be matched to similar, but not identical, defined tools of the same tool type. The matching of the defined tools to the unmatched tools may be performed by the operator using the graphical user interface.

In yet another embodiment of the present invention, the algorithm for identifying the source tools and matching the source tools to the defined tools is used during importation of defined tools into the tool library to compare the imported defined tools with the previously defined tools to ensure that duplicates are not imported.

The above mentioned and other features of this invention, and the manner of attaining them, will become more apparent and the invention itself will be better understood by reference to the following description of embodiments of the invention taken in conjunction with the accompanying drawings.

Brief Description of the Drawings:

Figure 1 is a conceptual diagram of software components associated with the present invention.

Figure 2 is a flow diagram of a tool management algorithm ("TMA") according to the present invention.

Figure 3 is a flow diagram of a process control routine included in the TMA of Figure 2.

Figure 4 is a flow diagram of a tool matching algorithm included in the TMA of Figure 2.

Corresponding reference characters indicate corresponding parts throughout the several views. Although the drawings represent embodiments of the present invention, the drawings are not necessarily to scale and certain features may be exaggerated in order to better illustrate and explain the present invention.

Detailed Description of Embodiments of the Invention:

The embodiments disclosed below are not intended to be exhaustive or to limit the invention to the precise forms disclosed in the following detailed description. Rather, the embodiments are chosen and described so that others skilled in the art may utilize their teachings.

Figure 1 conceptually depicts a tool management algorithm ("TMA") 10, a part program 12, and a machine memory 14. TMA 10 is described in greater detail below. Part program 12 represents code for operation on a CNC machine to create a part. Part program 12 includes a plurality of commands and other information used by the machine to move the spindle or the part, adjust the speed of spindle rotation, change tools, etc. The commands and other information may be organized into sections or blocks representing operations 16, 18, 20 to be performed on the part. Each operation 16, 18, 20 may include a numeric reference to a tool (e.g., tool 1, tool 2, tool n) needed to perform the corresponding operation 16, 18, 20. Operations 16, 18, 20 further includes information describing the physical characteristics of the tools, including the tool type and geometry, and information describing the manner in which the tool is to be used. The tool use information may include tool feed and speed specifications. As is known in the art, tool feed specifies how quickly the tool may be moved across the material, and tool speed specifies how quickly to tool should be rotated. These parameters affect, among other things, the throughput of the machine and the surface finish of the part. Unless otherwise stated or made clear by context, references throughout this description to "source tools" are short hand for information about a tool associated with a particular operation 16, 18, 20, including its numeric reference and the data describing its physical characteristics and use specifications. It should be understood that while only three operations 16, 18, 20 are shown in Figure 1, any number of operations may be included in part program 12.

Machine memory 14 conceptually includes a tool library, including defined tools as is further described below, which are depicted in Figure 1 as arranged in ATC grouping 22, MTC grouping 24, and unmatched grouping 26. Groupings 22, 24, 26 are merely intended to indicate that certain of the defined tools in the library are associated with the ATC or the MTC, and other defined tools (unmatched grouping 26) are not associated with the ATC or the MTC. ATC grouping 22 is depicted as including defined tools 30, 32, 34. MTC grouping 24 is depicted as including defined tools 38, 40, 42. Unmatched grouping 26 is depicted as including source tools 44, 46, 48. Defined tools 30, 32, 34, 38, 40, 42, 44, 46, 48 include the same types of information as source tools (i.e., numeric references and data describing physical characteristics and use specifications). Accordingly, unless otherwise stated or made clear by context, references throughout this description to "defined tools" are short hand for information about a tool associated with the machine, including its numeric reference and data describing its physical characteristics and use specifications.

Machine memory 14 further includes a map 36 associated with ATC grouping 22. Map 36 includes information describing the physical position in the machine's ATC of the physical tools described by defined tools 30, 32, 34 of ATC grouping 22.

As should be apparent to one skilled in the art, memory 14 may be a distributed memory including multiple memory devices at various physical locations which are accessible by the machine (or multiple machines), either directly or over a wired or wireless network. Alternatively, memory 14 may reside entirely on the machine. As is further described below, defined tools 44, 46, 48 (only three shown to simplify the description) of unmatched grouping 26 are source tools from program 12 that do not match any defined tools in ATC grouping 22 or in MTC grouping 24 of the tool library. Even if defined tools 44, 46, 48 remain unmatched, the data may be used by the machine to perform virtual operations as is further described below.

When part program 12 is loaded onto a machine (e.g., transferred from a portable media to the machine controller or otherwise received by the controller from a source location), TMA 10 is executed. Various steps of TMA 10 are depicted in flow diagram 50 of Figure 2. Initially, TMA 10 processes the content of part program 12 to identify the source tools associated with operations 16, 18, 20. In this example, the source tools specified for operations 16, 18, 20 have numeric references 1, 2, and N, respectively. TMA 10 generates a listing of the source tools to be processed during loading of part program 12, including their numeric references and data describing their physical characteristics and use specifications. At step 52, TMA 10 determines whether the source tools need to be matched to defined tools included in the tool library. Initially, none of the source tools have been matched, so control is passed to step 54. TMA 10 then accesses source tool 16 and calls (step 56) a process control routine 58 (Figure 3) to process source tool 16.

Referring now to Figures 3 and 4, step 60 of routine 58 simply indicates that the input parameter for routine 58 is source tool 16. At step 62, routine 58 calls the tool matching algorithm 64 depicted in Figure 4 to evaluate defined tools 30, 32, 34 of ATC grouping 22. In general, algorithm 64 identifies any defined tools in ATC grouping 22 that satisfy the requirements of source tool 16 (i.e., defined tools that match source tool 16). Step 66 indicates that the input parameters for algorithm 64 are source tool

16 and defined tools 30, 32, 34 of ATC grouping 22. At step 68, algorithm 64 accesses the first defined tool of ATC grouping 22 (i.e., defined tool 30). At step 70, algorithm 64 determines whether all of defined tools 30, 32, 34 in ATC grouping 22 have been processed. At this stage of processing, the result is "no." Accordingly, control is passed to step 72. At step 72, algorithm 64 determines whether the tool type of defined tool 30 matches the tool type of source tool 16. As indicated above, each source tool and each defined tool includes information identifying the type of tool (e.g., drill, end mill, etc.) described by the associated data. In one embodiment of the invention, if the tool types do not match, then there is no point in further investigating the physical characteristics of the tools to determine if they match. As such, if the answer to step 72 is "no," then control is passed to step 74, which causes algorithm 64 to access the next defined tool 32 in ATC grouping 22. Otherwise, control is passed to step 76, where the physical geometries of source tool 16 and defined tool 30 are compared.

At step 76, algorithm 64 accesses information in source tool 16 describing a plurality of physical characteristics of the physical tool specified by source tool 16. For example, source tool 16 may specify a cut diameter value, a shank diameter value, a flute length value, etc. Each of these physical characteristics may be compared to the corresponding data in defined tool 30 to calculate a compatibility index for defined tool 30. The method of computing a compatibility index may vary for each tool type. In some cases, one or more threshold conditions must be satisfied during the comparison in order for it to proceed. For example, when comparing data associated with drills, step 76 may require that the diameters differ by no more than 0.000001 mm, and that the spin directions be identical. Further comparison is skipped unless both threshold conditions are met. In one embodiment of the invention, step 76 compares physical characteristics and computes the compatibility index by applying weighting factors to one or more of these geometry comparisons. For example, some of the physical characteristics may be considered more critical than others, and thus may have a heavier weighting factor. The weighted comparisons result in a compatibility index value, for example, a value between zero and one.

At step 78, the compatibility index value is compared to a threshold value. It should be understood, however, that in embodiments where a compatibility index is not computed, but rather tool matching criteria are applied to the tools under consideration, step 78 is a determination of whether the matching criteria are met. In one embodiment of the invention, during the automatic matching process described herein in association with loading of part program 12, the threshold value to which the compatibility index is compared is one, signifying an identical match of source tool 16 with the defined tool currently under consideration. It should be understood, however, that one of ordinary skill in the art may implement a different threshold value to permit matches that are less than identical. Step 78 determines whether the compatibility index value is equal to or exceeds the threshold (or whether the matching criteria are met, as the case may be). If not, then control passes to step 74 where the next defined tool 32 of ATC grouping 22 is accessed. If, on the other hand, the compatibility index value is equal to or greater than the threshold (or the matching criteria are met, as the case may be), control is passed to step 80, where algorithm 64 identifies defined tool 30 as a match with source tool 16 before accessing defined tool 32 at step 74.

When algorithm 64 accesses defined tool 32 of ATC grouping 22, control is returned to step 70 where algorithm 64 determines whether it has processed all of the defined tools in ATC grouping 22. Algorithm 64 continues stepping through defined tools 30, 32, 34 of ATC grouping 22 and comparing them to source tool 16 in the manner described above until all of the defined tools of ATC grouping 22 have been processed. As algorithm 64 processes defined tools 30, 32, 34, it adds defined tools that match source tool 16 (if any) to the list created by step 80. Eventually, all of defined tools 30, 32, 34 of ATC grouping 22 are processed, and the result of step 70 is "yes."

At step 82, control is returned to process control routine 58 of Figure 3. Having completed step 62, routine 58 performs step 84 which again calls tool matching algorithm 64. During this execution of algorithm 64, all of defined tools 38, 40, 42 of MTC grouping 24 are processed in the manner described above with reference to ATC grouping 22. Any matches are added (at step 80) to the matched tools identified during processing of ATC grouping 22. When all of defined tools 38, 40, 42 of MTC grouping 24 have been processed, control is again returned at step 82 to process control routine 58.

At step 88, process control routine 58 calls tool matching algorithm 64 to process unmatched tools 44, 46, 48 in unmatched grouping 26. It should be understood that multiple part programs 12 may concurrently reside on the machine. In the process of loading these part programs 12, source tools may be added to unmatched tool grouping 26 if they do not match a defined tool in ATC grouping 22 or MTC grouping 24. Later loaded part programs 12 may require source tools that match the unmatched tools from earlier loaded part programs 12. While unmatched tools 44, 46, 48 are not physical tools currently available for use in cutting operations, they may still be used for virtual operations or added, for example, to the MTC if a physical tool corresponding to the unmatched tool is obtained for use. The processing of unmatched tools 44, 46, 47 is the same as that described above with regard to ATC grouping 22 and MTC grouping 24. As should be apparent from the foregoing, if a source tool is matched to an unmatched tool in unmatched tool grouping 26, then the source tool is not added to unmatched tool grouping 26. On the other hand, if a source tool does not match any of defined tools 30, 32, 34 of ATC grouping 22, defined tools 38, 40, 42 of MTC grouping 24, or unmatched tools 44, 46, 48 of unmatched grouping 26, then the source tool is added to tool library as an unmatched tool. In this manner, TMA 10 prevents addition of duplicate unmatched tools to unmatched grouping 26. When a part program 12 is unloaded from the machine, TMA 10 determines whether any unmatched tools from the program are still needed by other part programs. If so, then those unmatched tools are left in unmatched grouping 26. Otherwise, they are removed.

In one embodiment of the invention, at step 90 the various matched tools stored at step 80 of algorithm 64 are sorted to determine the best match with source tool 16. As a result of matching algorithm 64 as described above, all of the matched tools may be identical in geometry to source tool 16 (i.e., if the threshold value to which the compatibility index value is compared is set to one). In other embodiments, the matched tools may not be identical geometric matches. In that case, the primary sorting criteria in step 90 may be the compatibility index, which relates to the quality of the geometric match. Where all of the matched tools are identical in geometry to source tool 16 (or where there is a tie in the compatibility index sorting), the matched tools may be automatically sorted by location. In one embodiment, the location order in terms of preference for sorting is the spindle, the ATC, the MTC, then unmatched tool grouping 26. The spindle is preferred because the tool is already loaded. The ATC is the next preference because the tool will be automatically loaded. The MTC is the next preference because the tool is available for manual loading onto the spindle. Unmatched grouping 26 is the next preference because the unmatched tools, although not physically available for use, are at least geometrically characterized and can be used in virtual operations including verification of part program 12.

Referring back to Figure 3, at step 92 of process control routine 58, control is returned to TMA 10 at step 94 of Figure 2. At step 94, TMA 10 determines whether all of the sorted and matched tools have been processed. At this point in the process, none of the matched tools have been processed, and control is passed to step 96 where the first matched tool is accessed from the sorted list. If no matches were identified during processing of the defined tools and unmatched tools, or if all of the matched tools had been processed, then the result of step 94 would be "no," and source tool 16 would be added to unmatched grouping 26 at step 98.

For the purpose of explanation, assume that multiple defined tools matched source tool 16 during the processing described above. Assumer further that the first matched tool after sorting is defined tool 32, which has a numeric reference of tool 2. Step 100 determines whether the defined tool currently under consideration (here, defined tool 32) has already been matched to a source tool as is further described below. In this example we assume that defined tool 32 has not been associated with any source tool. Accordingly, at step 102 TMA 10 automatically correlates defined tool 32 with source tool 16 of part program 12.

In another embodiment, the numeric reference for the tools to be matched may be used as the criteria for designating a matching tool. For example, if multiple defined tools satisfy the matching criteria, any defined tool having a numeric reference that is identical to the numeric reference of the source tool may automatically be designated a matched tool. As with all matched tools, the designated matched tool cannot thereafter be matched with another source tool, even if all matching criteria (except numeric tool reference) are met.

In some circumstances, part programs require the use of two identical tools. For example, an end mill of a particular geometry may be required for rough cutting the part, while another end mill of the identical geometry is required for finish cutting. Where this situation exists, TMA 10 needs to ensure that the same matched tool is not associated with both source tools. Step 100 of algorithm 64 provides this feature. In one embodiment, the first source tool would be correlated with the first matched tool accessed in step 96. At step 100, TMA 10 determines that the first matched tool has not yet been matched to a source tool. Thus, at step 102, the first matched tool is correlated with the first source tool.

The second source tool is then processed through steps 52, 54, and 56. As the second source tool is identical to the first, the same matched tools should be identified during this processing, and they will be sorted in the same order. Accordingly, at step 96 TMA 10 will access the first matched tool. At step 100, TMA 10 will determine that the first matched tool has already been associated with a source tool. TMA 10 will then return to step 94. Assuming additional matched tools were identified by tool matching algorithm 64, the result of step 94 will be "yes," and the next matched tool will be accessed at step 96. At step 100, TMA 10 will determine that the second matched tool has not been associated with a source tool. Finally, at step 102 TMA 10 will correlate the second matched tool with the second source tool. In this manner, a defined tool on the machine will not be correlated to two separate, but identical source tools required by part program 12.

After a matched tool is correlated with source tool 16 in the manner described above, TMA 10 returns to step 52 to determine whether additional source tools required by part program 12 need to be matched. Source tool 18 will be accessed at step 54. At step 56, TMA 10 calls process control routine

58, which executes tool matching algorithm 64 for each defined tool of ATC grouping 22 and MTC grouping 24, and each unmatched tool of unmatched grouping 26 in the manner described above. In one embodiment of the invention, one of the sorted matched tools identified during this processing (if any) is then correlated at step 102 with source tool 18 of part program 12. In another embodiment, all of the matched tools identified are displayed to the operator for manual selection. If no matched tools are found, then source tool 18 is added to unmatched grouping 26 at step 98. The numeric reference for source tool 18 when added to unmatched grouping 26 will be the numeric reference associated with source tool 18 (i.e., tool 2) if that numeric reference is not already associated with another defined tool in tool library, otherwise it will be the next available numeric reference.

As should be apparent from the foregoing, each of the source tools of part program 12 is processed in the above-described manner to be either matched to a defined tool or unmatched tool in tool library or stored in unmatched grouping 26. Finally, at step 52 TMA 10 will determine that all of the source tools of part program 12 have been processed, and control will pass to step 104 where TMA 10 replaces the original numeric tool references associated with the source tools with the numeric references of the correlated matched tools. When performing this function, TMA 10 first identifies, for each source tool, all operations 16, 18, 20 in part program 12 that use the source tool. Then, TMA 10 replaces the original tool references with the matched tool references identified above. The following example illustrates this process.

Assume in part program 12, three operations exist, and they originally reference the following source tools:

Operation 1: Source tool 1 (roughing); Source tool 2 (finishing)
Operation 2: Source tool 3 (roughing); Source tool 2 (finishing)
Operation 3: Source tool 1 (roughing); Source tool 4 (finishing)

At step 104 of Figure 2, TMA 10 identifies the operations in part program 12 that use each of the originally referenced source tools:

Source tool 1: Operation 1 (roughing), Operation 3 (roughing)
Source tool 2: Operation 1 (finishing), Operation 2 (finishing)
Source tool 3: Operation 2 (roughing)
Source tool 4: Operation 3 (finishing)

Assume further that during operation of TMA 10 as described above, the following numeric references associated with matched tools were identified:

Source tool 1 matched to matched tool 4
Source tool 2 matched to matched tool 1
Source tool 3 matched to matched tool 2
Source tool 4 matched to matched tool 10

After all of the operations referencing source tools are identified as outline above, the original source tool references are replaced in the following manner:

Replace source tool 1 with matched tool 4 in operation 1 (roughing) and operation 3 (roughing)
Replace source tool 2 with matched tool 1 in operation 1 (finishing) and operation 2 (finishing)
Replace source tool 3 with matched tool 2 in operation 2 (roughing)
Replace source tool 4 with matched tool 10 in operation 3 (finishing)

In this manner, TMA 10 avoids replacing a matched tool reference with another matched tool reference. After this replacement process, TMA 10 ends at step 106.

In one embodiment of the invention, during the above-described replacement process, TMA 10 checks the tool use information (i.e., the feed and speed information) corresponding to the source tools to determine whether that information was manually programmed into part program 12. Some operators modify the default feed and speed information provided with the tool. During creation of part program 12, the feed and speed information may either be imported, for example, from memory 14 when the tool is defined, or manually entered. If the information is manually entered, then a flag is set in association with the information to indicate manual entry. TMA 10 identifies these flags in making the manual entry determination. In one embodiment of the invention, if the feed and speed information was manually entered, then TMA 10 retains it. If the feed and speed information was not manually entered, then TMA 10 replaces it with the feed and speed information associated with the matched tool during the replacement process described above.

When one or more source tools are added to unmatched grouping 26, an "unmatched tool(s)" message is displayed to the operator after part program 12 is loaded. In one embodiment of the invention, the operator has the option of selecting a tool review screen, which lists all of the source tools and the location of their matches (i.e., the spindle, the ATC, the MTC, or in unmatched grouping 26). If no matched tool was found, the source tool is identified as an "unmatched tool." The operator may match unmatched tools with similar, but not identical, defined tools in ATC grouping 22 or MTC grouping 24. When the operator selects (e.g., clicks on) an unmatched tool on the tool review screen and activates a "find matches" command, TMA 10 is again executed for the selected unmatched tool. For this unmatched tool operation, however, TMA 10 begins execution at step 60 of process control routine

58 and ends execution at step 92 as described below. Additionally, the threshold associated with step 78 of tool matching algorithm 64 is set to zero instead of one.

At step 60, the input parameter is the unmatched tool. Step 62 calls tool matching algorithm 64 of Figure 4 to locate matching tools in ATC grouping 22 in the manner described above. In one embodiment, at step 86, instead of computing a compatibility index value for each defined tool 30, 32, 34 of the same tool type as the unmatched tool, algorithm 64 sets the compatibility index to zero and assigns a percentage match based on the diameter of the defined tool 30, 32, 34. For example, a defined tool having a diameter that is identical to the diameter specified for the unmatched tool is assigned a percentage match of 100%. A defined tool have a diameter twice the size of the diameter specified for the unmatched tool data block is assigned a percentage match of 200%, and so on.

As the threshold for step 78 is zero during the unmatched tool operation, every defined tool 30, 32, 34 processed by step 78 will be added to the list of matched tools. This same process is performed at steps 84, 86, and 88 of process control routine 58 for defined tools 38, 40, 42 of MTC grouping 24. The list of matched tools may then be sorted at step 90 of process control routine 58 in order of percentage match. Finally, the sorted list of matched tools is displayed to the operator, along with the locations (i.e., the ATC or the MTC) of the matched tools. The operator then selects a matched tool from the list and it is used to replace the previously unmatched source tool required by part program 12 in the manner described above.

As was indicated above, a tool matched to a source tool required by part program 12 need not have a corresponding physical tool on the machine (i.e., a tool in the spindle, the ATC, or the MTC). If the operator executes part program 12 and a source tool has been correlated to an unmatched tool of unmatched grouping 26, then TMA 10 will display a message to the operator indicating that a physical tool is not present for one or more of the source tools. The operator may load a physical tool on the machine by placing it, for example, in the ATC and creating a corresponding defined tool in the tool library. Alternatively, the operator may select an option of defining the tool and adding it to the MTC for manual loading at the appropriate time. Even if the operator does not obtain a physical tool, the operator may cause the machine to perform virtual operations using the unmatched tool such as a graphical verification of part program 12.

In addition to its use during loading of part program 12 and manual matching of unmatched tools, TMA 10 may be used when defined tools are added to the tool library. The tool library may be updated with new defined tools from part programs or back-up files, with defined tools located in a central library server, etc. Regardless of the source of the defined tool for importation into the tool library, TMA 10 may be executed to prevent importation of duplicate defined tools.

For example, an operator may identify a source tool in a part program that the operator would like to import to tool library of a particular machine. Instead of executing the tool matching process of TMA 10 described above, the operator may execute a tool importation process wherein all source tools in the part program are imported into tool library if they do not already exist in tool library. After assembling the list of source tools referenced in part program 12 in the manner described above, TMA 10 accesses the first source tool (steps 52 and 54), and calls process control routine 58 (step 56). The input parameter at step 60 of routine 58 is the first source tool. At step 62, TMA 10 calls tool matching algorithm 64 to compared defined tools 30, 32, 34 of ATC grouping 22 to the first source tool in the manner described above. In this application of TMA 10, the compared tools must be identical to be considered a match. If a matched tool is identified, the first source tool is not imported into the tool library because it would constitute a duplicate. If no matched tool is identified, defined tools 38, 40, 42 of MTC grouping 24 are processed (step 84) in the same manner. Again, if a match is found, the source tool is not added to the tool library. If none of the defined tools in ATC grouping 22 or MTC grouping 24 match the source tool, it is added to the tool library. This procedure is repeated for the remaining source tools.

While this invention has been described as having exemplary embodiments, the present invention may be further modified within the spirit and scope of this disclosure. This application is therefore intended to cover any variations, uses, or adaptations of the invention using its general principles. Further, this application is intended to cover such departures from the present disclosure as come within known or customary practice in the art to which this invention pertains.

CLAIMS

1. A method of correlating tools associated with a cutting machine and tools specified by a program executed by the machine to form a part, the method including the steps of:

identifying source tools specified by the program;

comparing each source tool with defined tools associated with the machine;

designating matched tools, which are defined tools that match the source tools during the comparing step; and

replacing source tool references in the program with references to corresponding matched tools.

2. The method of claim 1, wherein the comparing step includes the step of comparing a tool type identifier of the source tool with tool type identifiers of the defined tools.

3. The method of claim 1, wherein the comparing step includes the step of comparing a physical characteristic of the source tool with a physical characteristic of each defined tool.

4. The method of claim 3, wherein the comparing step includes the step of computing a compatibility index value based on the physical characteristic comparison and comparing the compatibility index value to a threshold value.

5 The method of claim 1, wherein the comparing step includes the step of computing a compatibility index value by applying weighting factors to a plurality of comparisons of physical characteristics of a source tool with physical characteristics of a defined tool.

6. The method of claim 5, wherein the designating step includes the step of determining that the compatibility index value at least meets the threshold value.

7. The method of claim 3, wherein the designating step includes the step of determining that the physical characteristic of a defined tool matches the physical characteristic of the source tool.

8. The method of claim 7, wherein in the determining step, the physical characteristics match when they are within a specified tolerance of one another.

9. The method of claim 1, further including the step of sorting matched tools.

10. The method of claim 9, wherein the sorting step includes using matched tool compatibility index value as a sorting criterion.

11. The method of claim 9, further including the step of determining a matched tool to correlate with a source tool based on an outcome of the sorting step.

12. The method of claim 1, wherein the replacing step includes the steps of identifying any references in the program to a first source tool and replacing all reference to the first source tool

with references to a matched tool before replacing references to a second source tool with references to a matched tool.

13. The method of claim 1, wherein the replacing step includes the step of replacing tool use information associated with a source tool with tool use information associated with a matched tool.

14. The method of claim 13, including the step of identifying manually entered tool use information associated with the source tool.

15. The method of claim 1, wherein the comparing step includes the steps of comparing each source tool to defined tools corresponding to physical tools available for use by the machine and comparing each source tool to defined tools that do not correspond to physical tools available for use by the machine.

16. The method of claim 1, further including the step of displaying a message to an operator when no defined tools match the source tool.

17. The method of claim 1, wherein in the designating step, a defined tool matches a source tool only if a physical characteristic of the defined tool is identical to a physical characteristic of the source tool.

18. The method of claim 1, further including the steps of displaying a plurality of defined tools that do not match a source tool and enabling an operator to select a defined tool from the plurality of defined tools to designate as a matched tool.

19. The method of claim 1, wherein a defined tool is designated a matched tool only after determining that the defined tool had not previously been designated a matched tool.

20. The method of claim 1, further including the step of storing a source tool in a memory as a new defined tool associated with the machine if the defined tools associated with the machine do not match the source tool.

21. The method of claim 1, further including the step of loading the program into a memory associated with the machine.

22. The method of claim 20, wherein the storing step is performed as to a particular source tool only after the step of determining that the particular source tool does not match any other source tool stored in the memory.

23. The method of claim 1, further including the step of accessing a map that associates tool references to physical locations of corresponding tools.

24. The method of claim 1, further including the step of adding a source tool to a memory associated with the machine if the source tool does not match the defined tools.

25. The method of claim 1, wherein the designating step includes the steps of determining that none of the defined tools match a source tool, displaying a plurality of defined tools as options for correlating to the unmatched source tool, and enabling the operator to select a displayed defined tool as a matched tool.

26. The method of claim 1, further including the step of using a matched tool to perform a virtual operation.

27. A computer readable medium having computer-executable instructions for correlating tools associated with a machine tool system with tools specified by a program for the system to use in forming a part, the computer executable instructions including:

an algorithm for identifying source tools specified by the program;

a tool matching algorithm for comparing each source tool with defined tools associated with the machine, and designating as matched tools any defined tools that are acceptably similar to the source tools; and

an algorithm for replacing source tool references in the program with references to corresponding matched tools.

28. An apparatus for machining a part using a plurality of tools associated with the apparatus as specified by source tools including in a program processed by the apparatus, the apparatus including:

a spindle;

an automatic tool changer configured to load tools onto the spindle; and

a memory including

a tool management algorithm for identifying source tools specified by the program,

a tool matching algorithm for comparing each source tool with defined tools associated with the machine, and designating as matched tools any defined tools that are acceptably similar to the source tools, and

an algorithm for replacing source tool references in the program with references to corresponding matched tools.

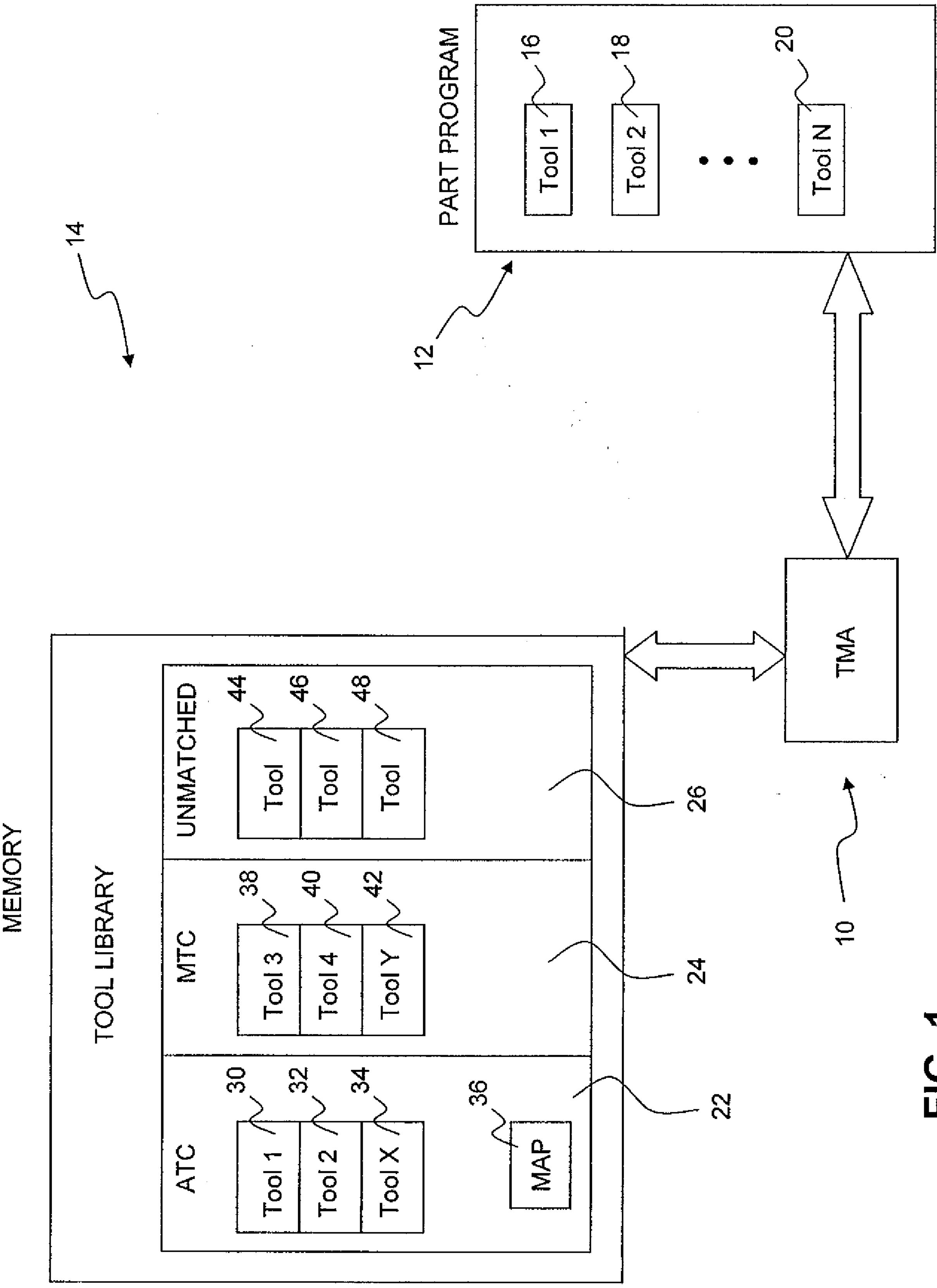

PART PROGRAM
Tool 1
Tool 2
Tool N
16
18
20
12
14
MEMORY
TOOL LIBRARY
ATC
Tool 1
Tool 2
Tool X
MAP
30
32
34
36
22
MTC
Tool 3
Tool 4
Tool Y
38
40
42
24
UNMATCHED
Tool
Tool
Tool
44
46
48
26
TMA
10

FIG. 1

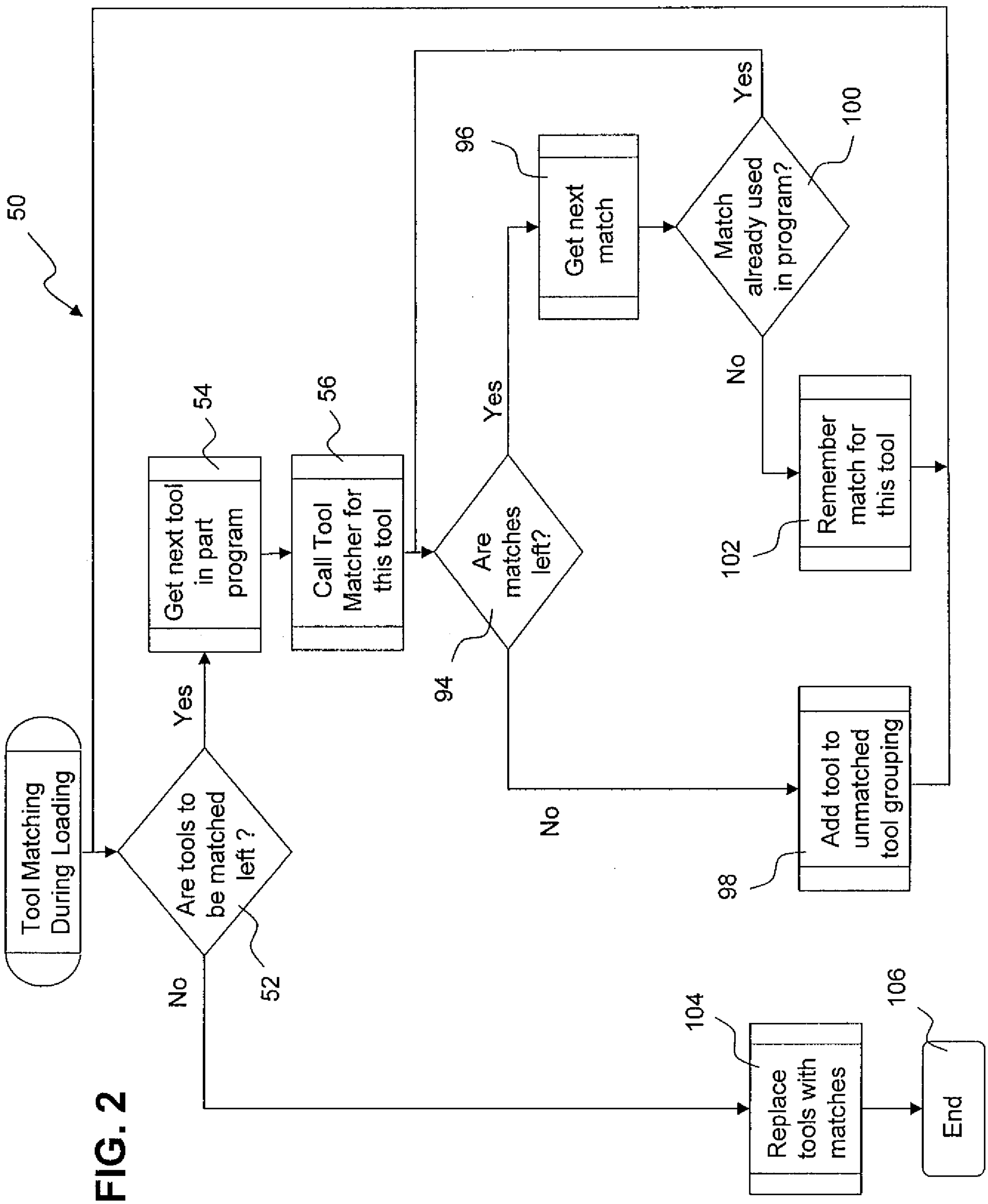

50
Tool Matching During Loading
Are tools to be matched left ?
52
Yes
No
Get next tool in part program
54
Call Tool Matcher for this tool
56
Are matches left?
94
Yes
No
Get next match
96
Match already used in program?
100
Yes
No
Remember match for this tool
102
Add tool to unmatched tool grouping
98
Replace tools with matches
104
End
106

FIG. 2

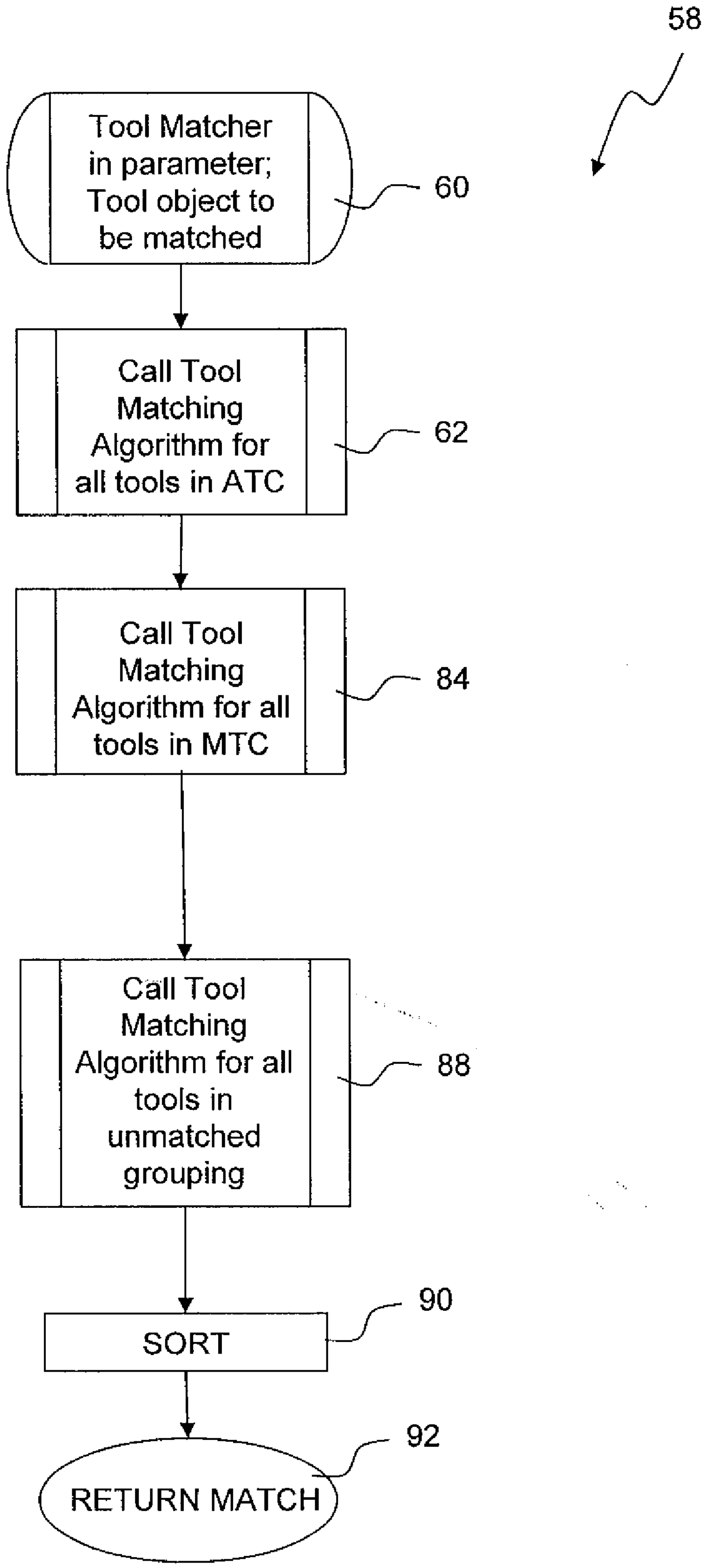

58
Tool Matcher in parameter; Tool object to be matched
60
Call Tool Matching Algorithm for all tools in ATC
62
Call Tool Matching Algorithm for all tools in MTC
84
Call Tool Matching Algorithm for all tools in unmatched grouping
88
SORT
90
RETURN MATCH
92

FIG. 3

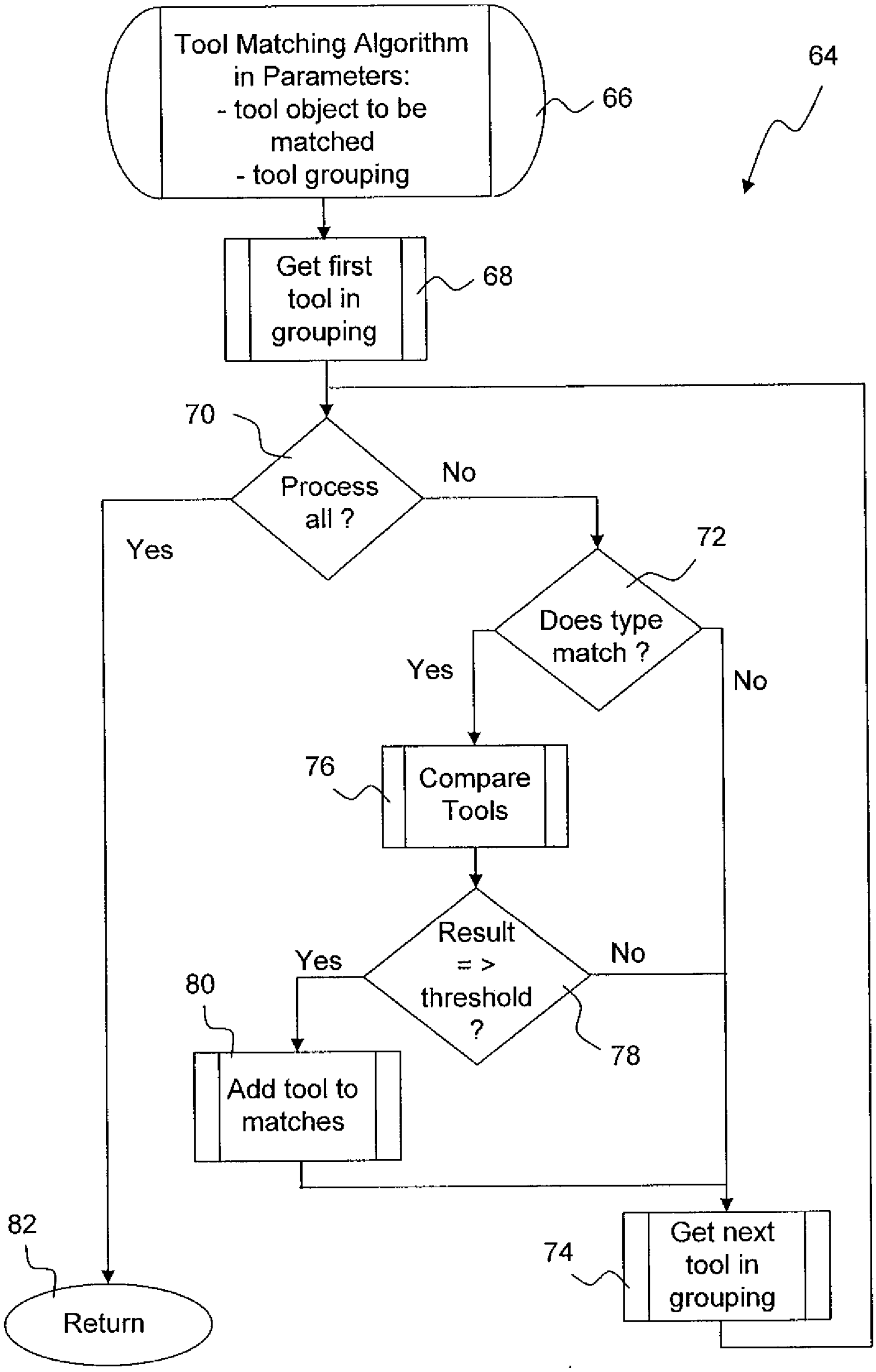

Tool Matching Algorithm in Parameters: - tool object to be matched - tool grouping
64
66
Get first tool in grouping
68
70
Process all ?
No
Yes
72
Does type match ?
Yes
No
76
Compare Tools
Result = > threshold ?
Yes
No
80
78
Add tool to matches
82
74
Get next tool in grouping
Return

FIG. 4