Parkinson, Simon, Longstaff, Andrew P., Allen, Gary, Crampton, Andrew, Fletcher, Simon and Myers, Alan

Hierarchical Task Based Process Planning For Machine Tool Calibration

Original Citation

Parkinson, Simon, Longstaff, Andrew P., Allen, Gary, Crampton, Andrew, Fletcher, Simon and Myers, Alan (2011) Hierarchical Task Based Process Planning For Machine Tool Calibration. Proceedings of The 29th Workshop of the UK Planning and Scheduling Special Interest Group (PlanSIG2011). pp. 53-60. ISSN 1368-5708

Hierarchical Task Based Process Planning For Machine Tool Calibration

S. Parkinson, A. P. Longstaff, G. Allen, A. Crampton, S. Fletcher, A. Myers

Centre for Precision Technologies, School of Computing and Engineering, University of Huddersfield, Queensgate, Huddersfield, HD1 3DH, UK

simon.parkinson@hud.ac.uk

Abstract

A literature search has indicated that artificially intelligent planners have not previously been used to address the planning problem of machine tool calibration, even though there are potential advantages. The complexity of machine tool calibration planning requires the understanding and examination of many influential factors, such as the machine's configuration and available instrumentation. In this paper we show that machine tool calibration planning can be converted into a Hierarchical Task Network by the process of task decomposition. It is then shown how the Simple Hierarchical Ordered Planner architecture can be used to provide all the identified complete process plans in a given time frame, and secondly, how the branch-and-bound optimisation algorithm can find the optimal solution in the same frame. The results for generating the process plans and optimal process plans for both a three and five axis machine are evaluated to examine the planner's performance.

Introduction

Generating process plans automatically is a challenging yet advantageous quality. The economic advantages are seen as significant to engineers (Satyandra, 1998). This is because the ability to create both an efficient and complete process plan can result in minimising the risk of problems occurring that could ultimately result in excessive expenditure. This is true for the process of machine tool calibration planning (Bringmann et al., 2008).

The requirement to manufacture more accurate parts and minimise manufacturing waste is resulting in the continuing requirement for machine tools which are more accurate. Therefore, machine tool calibration is required regularly to gain an understanding of a machine's capability. When planning a machine tool calibration, an engineer will derive a calibration plan based on many influencing factors. For the work undertaken within this paper, we are only concerned with (1) the machine's configuration of constituent parts, (2) the errors associated with the machine, (3) the available instrumentation, and (4) scheduling and resource constraints. Other constraints, for example, the possibility of different test methods, have been excluded in an attempt to identify a simplified set that allows for the creation of an initial prototype. The complexity and quantity of knowledge that is required and processed during machine tool calibration planning is sufficient to require the use of a computational reasoning.

The way that the process of machine tool calibration can be broken down into smaller tasks makes it well suited to being represented by a Hierarchical Task Network (HTN). An HTN planner will recursively decompose nonprimitive tasks into smaller subtasks until primitive tasks are reached which can be performed directly using planning operators (Nau et al., 2003). The literature suggests that HTNs have been widely used as a planning technique because they are a convenient way to write problem-solving recipes that correspond to how a human domain expert would think about solving the problem (Ghallab et al., 2004). The Simple Hierarchical Ordered Planner 2 (SHOP2) is a domain-independent planning system that allows for the implementation of a domain-specific problem-solving planner (Goldman, 2011). For this reason, the SHOP2 system has been selected for use.

In this paper, a solution to the problem of machine tool calibration planning is presented by adopting a cross-discipline approach to develop a Hierarchical Task Network (HTN) which is implemented using the SHOP2 architecture. First, a literature review of HTN's being applied to engineering planning problems is presented. Next, the problem of machine tool calibration planning is described in more detail. This leads to the implementation of an initial HTN system using SHOP2. The results of the prototype solution are presented and discussed describing the scope for future work.

Literature survey

There is currently an absence of any literature indicating the advancement of process planning for machine tool calibration. For this reason, planning advancements in other engineering processes of a similar nature are examined to identify any intelligent approaches.

Significant effort has been spent in the improvement of automated planning techniques for industrial applications. There have been many successful implementations within mechanical engineering. The Interactive Manufacturability Analysis Critiquing System (IMACS) was developed to evaluate the manufacturability of machined parts and to suggest improvements to increase the ease of manufacture (Satyandra, 1998). The system processes the geometric features of a Computer Aided Design (CAD) model to determine the required machining operations. The authors have identified the complexities with populating a general purpose planner with domain-specific knowledge. Instead, they integrate the domain-specific knowledge into the planning algorithms themselves. The finished IMACS made use of an HTN planning system using a depth-first branch-and-bound search strategy to find the optimal complete process plan.

A similar application named the computer-aided process planning (CAPP) system was also developed to find both a complete and optimal solution for the manufacturing of a part based on (1) a description of the blank part, (2) description of the finished part, (3) available resources, and (4) technical knowledge (Deák et al., 2001). The CAPP system is represented in HTN form by using the SHOP architecture. The motivation behind the selection of an HTN is very similar to that as IMACS. It was found that traditional general purpose planners did not allow for the specification of the domain-specific knowledge.

In conclusion, it is evident that significant work throughout the 1990s has been performed to optimise the process of manufacturing parts, which has been largely successful. The significance of earlier work can be seen in that many commercially available Computer Aided Manufacture (CAM) packages now implement intelligent functionality to improve the part's design and proposed machining operations to reduce both manufacturing time and cost (Delcam, 2011).

Previous work has shown that the process of machine tool calibration can be represented in first-order logic (Parkinson et al., 2011) to provide a means of modelling all the possible tests that can be performed during the calibration of a specific machine tool. However, this knowledge needs interpreting to decide on the most feasible set of tests to reach the state of having a calibrated machine.

Machine tool calibration

As previously identified in the introduction, machine tool calibration is based on many influencing factors. For the context of this paper, the following section contains enough information regarding the influencing factors of machine tool calibration to allow the reader to understand the planning problem in sufficient detail.

Machine configuration

A machine can be designed and constructed in many different ways to perform its task. Figure 1 shows a machine tool with three perpendicular linear axes, while Figure 2 shows a gantry machine tool with three perpendicular linear axes and two rotary axes. In addition to the number of linear and rotary axes, the configuration (stacking) of these axes can cause errors to propagate differently throughout the machine. The configuration of a machine tool will determine how many error components it has. While there are a few common machine configurations, there are a lot of different configurations which require in-depth consideration to identify all their error components.

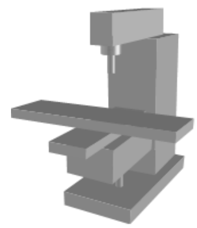

Figure 1 - Three-axis machine tool

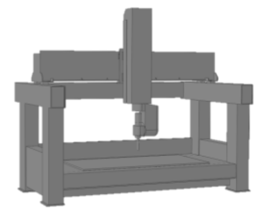

Figure 2 - Five-axis machine tool

Machine errors

The configuration of the machine's constituent parts determines the potential geometric errors that a machine might have. The geometric errors associated with linear and rotary axes are well known (Bohez et al., 2007). For example, a linear axis will have six error components (six-degrees-of-freedom) plus a squareness error with the perpendicular axis, which is illustrated in Figure 3. From this it is possible to deduce that a three axis machine tool will have in total 21 geometric errors (Ramesh et al., 2000)

A machine tool will, however, actually experience more error sources such as thermal, dynamic and non-rigid (Mekid, 2009). For the scope of this paper, only the calibration planning problem for geometric errors in machine tools is considered.

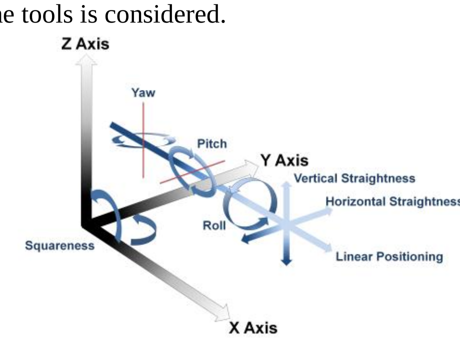

Figure 3 - Six-degrees of freedom and squareness errors for the X-axis of a machine tool with three perpendicular linear axes

Instrumentation

The extensive variety of instrumentation available for performing a machine tool calibration adds complexity to deciding the optimum solution when measuring each error component. There are many different reasons why a specific instrument might be selected. The following list supplies two sample Key Performance Variables (KPVs) which would influence the instrumentation selection.

1. The time to install and align the equipment may be lower
2. The resolution and accuracy of the instrument might be greater

For example, measuring the y-axis linear positioning error using the Renishaw XL-80 laser interferometer would require the configuration of the optics as seen in Figure 4. Next, measuring the y-axis pitch error would require the use of the optics aligned as seen in Figure 5. However, because the optics' base and the laser are already aligned, it is possible to carefully exchange the optics with only a small adjustment.

Given that the number of potential KPVs for selecting a given instrument is large, the work undertaken in this paper will only be concerned with the time required to install and setup the instrument, and the time to adjust the equipment from a previous setup.


Figure 4 – Linear position optics


Figure 5 - Pitch optics

Scheduling

Once a decision has been made to establish which instrument is to be used to measure each error component, the ordering of these measurements needs to be decided. As previously highlighted, there are many cases where the instrumentation will only need to be readjusted slightly to allow the measurement of two different error components. For this reason, finding the optimal sequence of measurements can reduce the time taken to perform the calibration by saving on instrumentation setup time.

HTN implementation

As identified in the introduction, the planning problem of machine tool calibration is well suited to being represented as an HTN. The following section shows how machine tool calibration was broken down into smaller tasks to create an HTN.

Task decomposition

Task decomposition is the process of breaking tasks into smaller tasks until primitive actions are reached. Figure 6 shows the abstract task decomposition for calibrating a machine tool, which takes into consideration what has been regarded as the main calibration tasks. A description for each primitive subtask can be found in the following list:

1. Find all linear errors based on the machine's configuration.
2. Find all rotary errors based on the machine's configuration.
3. Find all cross-axis errors based on the configuration of the linear and rotary axes.
4. Select an error component for measuring.
5. Select the suitable equipment for measuring the error.
6. Setup the equipment in a suitable way to measure the error component.
7. Measure the error component using the instrumentation and the current setup.

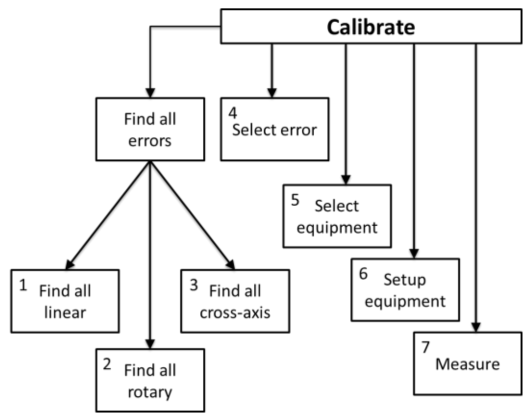

Figure 6 - Task decomposition tree

The process of performing this manual task decomposition to convert the nonprimitive task of machine tool calibration into the primitive tasks will serve as the basis for creating an HTN network.

System definition

An HTN planning problem is a 4-tuple

$$P = (s0, w, O, M)$$

Where $s0$ is the initial state, w is the initial task network, O is the set of operators, and M is the set of HTN methods. Applying this to the planning problem of machine tool calibration would mean that $s0$ is the initial non calibrated state of the machine tool, and w is the initial task network for performing the calibration. O would be the set of operators which describe how to perform a primitive task which cannot be decomposed any further to reach the state of a calibrated machine. M is the set of methods which perform the task decomposition based on a logical precondition.

Initial state

The initial state definition $s0$ can be regarded as the facts that describe the current planning problem. As previously described, a primitive version of machine tool calibration can be represented in first-order logic (Parkinson et al., 2011). An expansion of this work is used here to define the initial state. The following shows a sample of the facts that describe $s0$:

```
;;Axis
(axis X)

;;Axis Type
(linear X)

;;Linear geometric error + cost in priority
(linear-geometric-error PITCH 10)

;;Equipment + setup and adjust time (mins)
(equipment LASER 10 5)

;;Measurement + cost of performing (mins)
(measures PITCH LASER 10)
```

The size of $s0$ for representing a three axis machine tool which is used for testing in section contains a total of 34 facts. For a comprehensive representation, additional parameters would be included. For example, axis length, feed rate, number of targets, dwell time, etc. These additional parameters will be included once a working prototype has been achieved.

Initial task network

The initial high level task network w for performing a machine tool calibration is simply:

```
(perform-calibration)
```

In practice, it is highly possible that the initial task network might be more detailed than this. It is possible that there will be machine-specific preconditions that must be considered.

Operators

An operator is a description of how to perform a primitive task, which cannot be decomposed further. An operator's description is:

```
(:operator h P D A [c])
```

Where `h` = head, `P` = preconditions, `D` = delete list, `A` = add list, `c` = optional cost.

The following set O contains the operators that are required for the HTN implementation.

```
(:operator    (!select-error ?a ?e ?c)
              ((meas_required ?x ?y ?c ))
              ()
              ((meas_selected ?a ?e ))
              (*1 ?c))

(:operator    (!select-equip ?a ?e ?i ?mc ?c ?ac)
            ((meas_selected ?a ?e ))
              ()
              ((equip_selected ?a ?e ?i ?mc ?c ?ac)))

(:operator (!set-up-equip ?a ?e ?i ?mc ?c )
              ((equip_selected ?a ?e ?i ?mc ?c ?ac))

              ()
              ((equip_setup ?a ?e ?i ?mc))
              (* 1 ?c))

(:operator    (!adjust-equip ?a ?e ?i ?mc ?c ?pe
              ?pmc ?ac)
              ((equip_selected ?a ?e ?i ?mc ?c ?ac))
              ()
              ((equip_setup ?a ?e ?i ?mc))
              (* 1 ?ac))

(:operator    (!measure ?a ?e ?i ?mc )
              ((equip_setup ?a ?e ?i ?mc )(equipment
              ?i ?c ac)
              (equip_selected ?a ?e ?i ?mc ?c ?ac))
              ((meas_required   ?a ?e
              ?c)(meas_selected ?a ?e)
              (equip_selected ?a ?e ?i ?mc ?c ?ac)
              (equip_setup ?a ?e ?i ?mc ))
              ()
              (* 1 ?mc))

(:operator    (!!assert ?g)
              ()
              ?g
              0)

(:operator    (!!remove ?g)
              ?g
              ()
              0)
```

Methods

This section contains the set of methods M for performing the task decomposition in the HTN. The methods can been seen in the decomposition tree shown in Figure 6. Other methods can be seen here which are responsible for keeping track of the current error, instrumentation and instrumentation setup selection. A method's description is:

$$(:method\ h\ [n_1]\ C_1\ T_1\ [n_2]\ C_2\ T_2..\ [n_k]C_k\ T_k)$$

Where `h` = head, n_i = name for each succeeding C_i T_i pair, C_i = precondition, T_i = task list (tail).

```
(:method    (perform-calibration)
            ()((find-all-required)(calibrate)))

(:method    (find-all-required)
            ((linear ?a)(linear-geometric-error ?e
            ?c)
            (not(meas_required ?a ?e ?c)))
            ((!!assert ((meas_required ?a ?e ?c)))
            (find-all-required))
            ((linear ?axis)(cross-axis-error ?e
            ?c)(not(meas_required ?a ?e ?c)))
            ((!!assert ((meas_required ?a ?e ?c
            )))(find-all-required))
            ((rotary ?axis)(rotary-geometric-error ?e
            ?c)
            (not(meas_required ?a ?e ?c)))
            ((!!assert ((meas_required ?a ?e ?c)))
            (find-all-required))
```

```
            nil
            nil)

(:method    (calibrate)
            ((meas_required ?a ?e ?c)
            (not(meas_selected ?a ?e ))
            (not(measured ?a ?e )))
            ((!select-error ?a ?e ?c )(select-
            equipment)(calibrate))
            nil
            nil)

(:method    (select-equipment)
            ((meas_selected ?a ?e )(equipment ?i ?c
            ?ac)
            (measures ?e ?i ?mc)
            (not(equip_selected ?a ?e ?i ?mc ?c
            ?ac)))
            ((!select-equip ?a ?e ?i ?mc ?c ?ac)
            (set-up-equipment)(select-equipment))
            nil
            nil)

(:method    (set-up-equipment)
            ((equip_selected ?a ?e ?i ?mc ?c ?ac)
            (not(previous_error ?a ?pe ?i ?pmc))
            (not((equip_setup ?a ?e ?i ?mc))))
            ((!set-up-equip ?a ?e ?i ?mc ?c)(measure-
            error)
            (set-up-equipment))

            ((equip_selected ?a ?e ?i ?mc ?c ?ac)
            (previous_error ?a ?pe ?i ?pmc)
            (not((equip_setup ?a ?e ?i ?mc))))
            ((!adjust-equip ?a ?e ?i ?mc ?c ?pe ?pmc
            ?ac)
            (measure-error) (set-up-equipment))
            nil
            nil)

(:method    (remove-previous)
            ((previous_error ?a ?e ?i ?mc))
            ((!!remove((previous_error ?a ?e ?i
            ?mc)))
            (remove-previous))(not(previous_error ?a
            ?e ?i ?mc))
            nil)

 (:method   (measure-error)
            ((equip_setup ?a ?e ?i ?mc)(meas_required
            ?a ?e ?c))
            ((!measure ?a ?e ?i ?mc )
            (remove-previous)
            (!!assert((previous_error ?a ?e ?i ?mc)))
            (!!remove ((meas_required ?a ?e ?c))))
            nil
            nil)))
```

Branch-and-bound

The branch-and-bound algorithm is used for finding the lowest cost solution to optimisation problems (Nau et al., 2003). The computational expense for exploring every potential partial plan to find the optimal complete solution can be large, or even infinite. For example, a machine tool with three linear axes which each have six error components plus three squareness would result in the generation of 21 calibration tasks. There is then a potential 21^{21} sequences. To find the plan with the lowest cost, each of potential plans must be explored and evaluated. The number of potential sequences will increase with the addition of different instrumentation and measurement techniques. SHOP2 allows for the use of the branch-and-bound algorithm without any change to the HTN domain or problem specification.

Cost calculation

A SHOP2 operator also expresses a cost for performing the primitive task. The operators used in the machine tool calibration HTN have a cost assigned which is originally acquired from the initial state facts. The motivation behind an operator's cost is explained below:

1. Error selection – this is the importance of an error component. An error component that is regarded as having a high significance, or that should be measured first, is assigned a lower cost value.
2. Equipment setup – the cost in minutes that is required for setting up the instrumentation out of the box.
3. Equipment adjustment – this is the cost in minutes for adjusting the equipment. For example, realigning the optics of a laser interferometer.
4. Performing the measurement – this is the cost in minutes for measuring the error component using the selected equipment.

The implementation of an operator's cost can allow the branch-and-bound algorithm to find the optimal solution in a lower computational time.

Results

To evaluate the HTN's performance, empirical observations have been made using the two following planning problems. For the scope of the work presented in this paper, the selection of the planning problem and assigned cost values is arbitrary and not comprehensive. The cost values do however correctly show that some error components are more important than others and that different instrumentation requires a different length of time for setting up, adjusting and taking the measurement.

1. A machine tool with three linear axes. Each linear axis will have six geometric plus one squareness error components. There are a total of five different instruments available, and each error component can be measured by using at least two of the available instruments. The size of $s0$ for this problem is 53.
2. A five axis machine tool with three linear and two rotary axes. Each linear axis will have six geometric plus one squareness error components, and each rotary axis will have nine error components. There will also be a total of five different instruments available, and each error component can be measured by using at least two of the available instruments. The size of $s0$ for this problem is 99.

The experiments were carried out on an Ubuntu 10.04 virtual machine with 2GB of RAM and two cores of the host's AMD Phenom™ II X4 970 assigned. SHOP2 (v2.8.0) was executed in the Steel Bank Common Lisp environment (v1.0.51).

Plan exploration

Executing the HTN with both the three- and five-axis planning problems will result in the generation of all the complete potential plans. The HTN was executed initially to return the first complete plan. Next the HTN was executed in five seconds increments up to sixty seconds. SHOP2 returns information for each execution regarding the number of complete plans found, and the minimum and maximum cost.

As seen in Figure 7, it is noticeable that the number of complete plans generated for the three-axis machine is more than twice that of the five-axis machine. This highlights the higher computational effort for more complex problems. Figure 8 also shows the efficiency increase in terms of the time saved when comparing the first identified plan with the plan of the lowest cost discovered within the specified timeframe. For the tests that are executing in 5 second intervals, the plan with the lowest cost stabilise at 200 minutes for a three-axis machine after exploring 574 plans, and 18 minutes for a five-axis machine in just 50 plans. This shows that with no optimisation, the lowest cost plan from the 60 second period was discovered in 15 seconds, and 10 seconds for the five-axis machine.

Plan optimisation

Next, the same experiment was performed with the addition of the branch-and-bound optimisation. This is done by specifying the `:optimize-cost` flag in the problem definition. It is evident from Figure 9 that the number of complete plans generated in the allocated time frame is much lower with the use of the branch-and-bound algorithm.

It is also noticeable in Figure 9 that the number of optimised plans for the three-axis machine rises quickly, peaking at 22 before rapidly dropping to 6 where it stabilises. For the five-axis machine, the number of plans fluctuates between a maximum of 6 and a minimum of 2. This behavior is because the branch-and-bound optimisation is continuously trying to identify partial plans of a lower cost. Once a lower cost partial plan is identified, the algorithm will then explore it to find a complete plan that is of an overall lower cost than the previous plan. Figure 10 shows the increase in efficiency for the discovered plans. It is evident that the time saved for both the three- and five-axis machines increases gradually within the first 10 seconds. The time saved then stabilises for both the problems until 25 seconds for the three axis machine, where it reaches an efficiency saving of 19 minutes. The five axis problem increases rapidly until it stabilises with an efficiency gain of 74 minutes in 50 seconds of execution time.

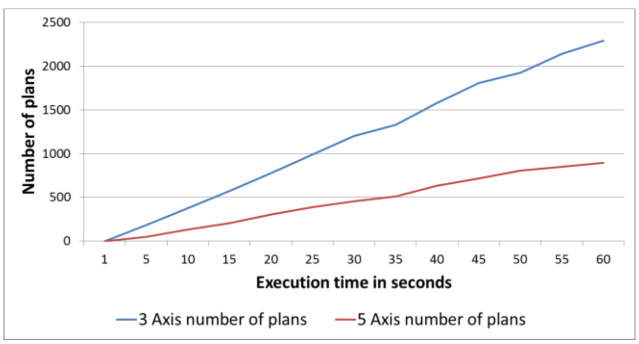

Figure 7 - Plan exploration

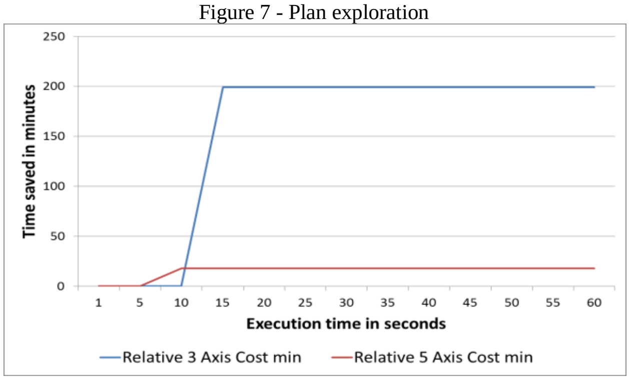

Figure 8 - Efficiency

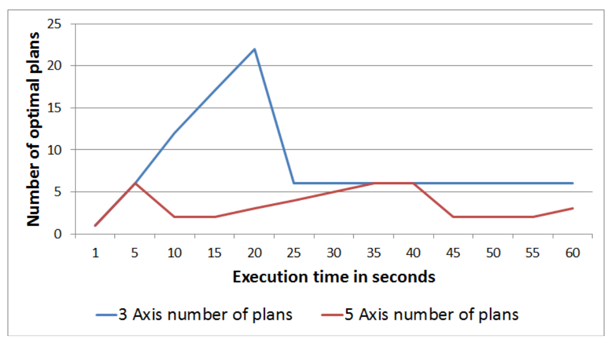

Figure 9 - Plan exploration (optimised)

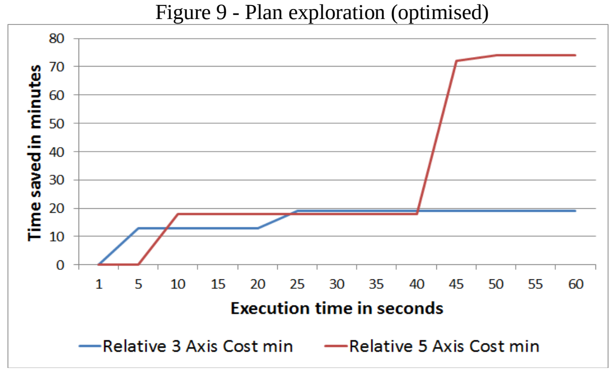

Figure 10 – Efficiency (optimised)

Plan comparison

In comparison, the number of plans generated when using the branch-and-bound optimisation algorithm is significantly lower. However, the number of explored plans is insignificant providing that the identified plans are the most efficient.

It is evident from Table 1 that the first identified plan for the three-axis machine when using the branch-and-bound algorithm has a lower cost by 186 minutes. The initial cost for a five-axis machine has the same cost for both tests. As seen in Table 2 the difference between the identified lowest cost plans in the whole sixty second period is 6 minutes for a three-axis machine, and 56 for a five-axis machine. This shows that the branch-and-bound algorithm can identify plans of a lower cost within the sixty second period even if the efficiency gain is only small.

Plan	**First plan cost**	**First optimised plan cost**	**Difference in minutes**
3-axis	2745	2559	186
5-axis	4777	4777	0

Table 1 - Comparison of the first identified plan

Plan	**Lowest cost plan**	**Lowest optimised cost plan**	**Difference in minutes**
3-axis	2546	2540	6
5-axis	4759	4703	56

Table 2 - Comparison of the identified lowest cost plan

Table 3 shows the execution time taken to identify the plan with the lowest cost with and without the use of the branch-and-bound optimisation. It is noticeable that the plans of a lower cost are discovered in the last third of the allocated time frame, and in the first quarter without the optimisation. Even though the time taken to find the optimal is 35 seconds longer for both problems when using the branch-and-bound optimisation, the overall efficiency gained makes its use beneficial. It is also evident that the cost reduction for the five-axis problem when using the branch-and-bound optimisation is higher than the three-axis problem. This potentially indicates that the efficiency of the optimisation algorithm increases as the problem's complexity also increases.

Problem	**Not optimised time**	**Optimised time**
3 axis	15	50
5 axis	10	45

Table 3 –Comparison of the execution time to find the lowest cost plan

Plan justification

The optimisation of the produced plan should result in an ordered set of tasks that exhibit the same or better time-saving decisions that a domain expert would make when creating a calibration plan. To examine whether this is true, an extract from the optimised plan for the three-axis machine problem was examined to highlight the justification behind the time-saving decisions that were made. This will validate the ability to encode expert knowledge and decision making skills in an HTN.

Comparing the first identified plan against the optimised plan will highlight the different ordering of tasks which results in the optimisation. From this difference in ordering, it is possible to derive the reasoning that resulted in the HTN making these decisions. Table 4 shows an extract taken from the optimised calibration plan for a three-axis machine and provides justification for the ordering.

Test	**Instrumentation use**	**Justification**
X Positioning	Laser interferometer	The positioning error of the X axis in this case has a high importance
X Straightness in Z	Laser interferometer	The laser is already aligned parallel to the X axis. From this setup the optics can be changed, therefore, saving time.
X Straightness in Y	Laser interferometer	
X about Y (Yaw)	Laser interferometer	
X about X (Roll)	Electronic level	Instrumentation is quicker to use than the laser
X about Z (Pitch)	Electronic level	The electronic level is already setup.

Table 4 – Extract from the optimised HTN plan with justification for the provided selection.

After examining the ordering of tests and their justification, it is evident that both the knowledge and decision making skills of a domain expert can be encoded and automated using a HTN. The justifications provided in Table 4 state the logical reasoning behind the test's ordering. This includes reasoning about which is the most efficient instrumentation to be used based on the previous and next error component.

Plan validation

The validity of the ordering can be verified by observing that the same decisions have been made in a traditional handmade plan for the same three-axis machine problem.
An extract from a calibration plan for a three-axis machine tool that was produced by a domain expert is shown in Table 5. When comparing this extract with the automated plan extract shown in Table 4 it is noticeable that the domain expert has made similar decisions regarding the sequencing of tests based on the premise of trying to minimise instrumentation setup and adjustment time.

Even though comparing an extract from an expert's calibration plan against the one generated by the HTN only provides for a very simplistic initial validation procedure, it does provide enough validation to warrant the continuation of this project by highlighting that calibration planning can be successfully automated without the loss of expert knowledge.

Test	Instrumentation used	Justification
X axis positional accuracy	Laser interferometer	
X axis accuracy and repeatability	Laser interferometer	The laser is already aligned parallel to the x-axis
X about X (Roll)	Electronic level	Depending on whether the optics will be realigned for the vertical Z, or removed for the Y. If removed for the Y, then remove the laser and setup the electronic level.
Z axis positional accuracy	Laser interferometer	
Z axis accuracy and repeatability	Laser interferometer	The laser is already aligned parallel to the z-axis
Squareness of X axis to Z axis	Ballbar or granite square	If the machine was also being tested for dynamic errors, then using the ballbar would be beneficial and save time later on.

Table 5 - Extract from a handmade plan with justification for the provided selection.

Conclusion

The work undertaken in this paper has shown how the process of machine tool calibration can be broken down by task decomposition to create a suitable HTN. The developed HTN was written in the common LISP format for execution in the SHOP2 architecture. Making use of a well-tested HTN architecture like SHOP2 means that no effort is wasted in implementing the HTN algorithm itself. It was then shown how the SHOP2 architecture can be used to execute problems against the created HTN. Two basic problem definitions of different complexities were created for testing the HTN's performance. The first was for a three-axis machine tool and the second a five-axis machine tool.

Each problem was tested by executing the HTN for durations in five second increments up to sixty seconds. This allowed for the evaluation of the quantity of complete plans found, and the minimum and maximum potential cost. In addition, this cycle was also performed using the branch-and-bound algorithm as a search optimisation strategy. After analysing the results it was evident that the use of the branch-and-bound algorithm improved the performance for both the three- and five-axis planning problems. It is suggested that the cost reduction provided by the branch-and-bound optimisation increases as the problems complexity also increases.

The results show that an HTN is viable solution for machine tool calibration process planning. The next stage is to develop the HTN further to include a more comprehensive representation of machine tool calibration. This would include better consideration of the instrumentation setup based on the machine's characteristics. Currently the HTN only looks for the solution with the lowest cost in terms of time and does not provide a solution for deciding which measurement techniques should be used. The HTN will require expanding to include and process a higher quantity of knowledge to remove this problem, and improve efficiency in terms of measurement traceability, repeatability and uncertainty. Verification of the complete HTN will not only include the evaluation of its efficiency, but the comparison the proposed plan against the plan of a subject expert. This way we can establish confidence in the HTN's planning power.

References

BOHEZ, E. L. J., ARIYAJUNYA, B., SINLAPEECHEEWA, C., SHEIN, T. M. M., LAP, D. T. & BELFORTE, G. 2007. Systematic geometric rigid body error identification of 5-axis milling machines. *Computer-Aided Design,* 39**,** 229 - 244.

BRINGMANN, B., BESUCHET, J. P. & ROHR, L. 2008. Systematic evaluation of calibration methods. *CIRP Annals - Manufacturing Technology,* 57**,** 529-532.

DEÁK, F., KOVÁCS, A., VÁNCZA, J. & DOBROWIECKI, T. P. 2001. Hierarchical Knowledge-Based Process Planning in Manufacturing. *Proceedings of the IFIP TC5 / WG5.2 & WG5.3 Eleventh International PROLAMAT Conference on Digital Enterprise - New Challenges: Life-Cycle Approach to Management and Production* Kluwer, B.V.

DELCAM. 2011. Knowledge-Based CAD/CAM: Separating the Hype from True Machining Intelligence. Available: http://featurecam.com/general/events/techart/knowbased.asp [Accessed 2/10/2011].

GHALLAB, M., NAU, D. & TRACERSO, P. 2004. *Automated Planning Theory and Practice* Morgan Kaufmann Publishers.

GOLDMAN, R. P. 2011. Documentation for SHOP2. University of Maryland.

MEKID, S. 2009. *Introduction to precision machine design and error assessment*, CRC Press.

NAU, D., AU, T., LLGHAMI, O., KUTER, U., MURDOCK, J., WU, D. & YAMAN, F. 2003. SHOP2: An HTN Planning System. *Journal of Artificial Intelligence Research* 20**,** 379-404.

PARKINSON, S., LONGSTAFF, A. P., CRAMPTON, A., ALLEN, G., FLETCHER, S. & MYERS, A. 2011. Representing the Process of Machine Tool Calibration in First-order Logic. *17th International Conference on Automation and Computing (ICAC'11).* Huddersfield.

RAMESH, R., MANNAN, M. A. & POO, A. N. 2000. Error compensation in machine tools -- a review: Part I: geometric, cutting-force induced and fixture-dependent errors. *International Journal of Machine Tools and Manufacture,* 40**,** 1235 - 1256.

SATYANDRA, K. G. 1998. IMACS: A Case Study in Real-World Planning. *In:* DANA, S. N. & WILLIAM, C. R. (eds.)