UNIVERSITY OF HUDDERSFIELD

CONSTRUCTION OF MACHINE TOOL CALIBRATION PLANS USING DOMAIN-INDEPENDENT AUTOMATED PLANNING

by

SIMON LLOYD PARKINSON

A thesis submitted in partial fulfilment for the degree of Doctor of Philosophy

in the
School of Computing and Engineering
Centre for Precision Technologies

March 2014

Copyright Statement

Abstract

The evolution in precision manufacturing has resulted in the requirement to produce and maintain more accurate machine tools. This new requirement coupled with desire to reduce machine tool downtime places emphasis on the calibration procedure during which the machine's capabilities are assessed. Machine tool downtime can be as much as £120 per hour and is significant for manufacturers because the machine will be unavailable for manufacturing use, therefore wasting the manufacturer's time and potentially increasing lead-times for clients. In addition to machine tool downtime, the uncertainty of measurement, due to the schedule of the calibration plan, has significant implications on tolerance conformance, resulting in an increased possibility of false acceptance and rejection of machined parts.

Currently calibrations are planned based on expert knowledge and there are no intelligent tools aiding to produce optimal calibration plans. This thesis describes a method of intelligently constructing calibration plans, optimising to reduce machine tool downtime and the estimated uncertainty of measurement due to the plan schedule. This resulted in the production of a novel, extensible domain model that encodes the decision making capabilities of a subject expert. Encoding the knowledge in PDDL2 requires the discretization of non-linear resources, such as continuous temperature change.

Empirical analysis has shown that when this model is used alongside state-of-the-art automated planning tools, it is possible to achieve a reduction in machine tool downtime greater than 10% (12:30 to 11:18) over expert generated plans. In addition, the estimated uncertainty due to the schedule of the plan can be reduced by 59% (48 µm to 20 µm). Further experiments on a PC architecture investigate the trade-off when optimising calibration plans for both time and the uncertainty of measurement. These experiments demonstrated that it is possible to optimise both metrics reaching a compromise that is on average 5% worse that the best-known solution for each individual metric. Additional experiments using a High Performance Computing architecture show that on average optimality of calibration plans can be improved by 4%; a potential saving of 30 minutes for a single machine and 10 hours for a company with 20 machines tools. This could incur a financial saving in excess of £1200 saving.

Acknowledgements

This thesis was written while studying in the Engineering Control and Machine Performance Group (ECMPG) of the University of Huddersfield. I am very grateful to both for their generosity and financial support. Without their support I would have been unable to write this thesis and present myself as a PhD candidate.

Firstly and foremost I would like to express my great gratitude to Dr. Andrew P. Longstaff for his committed supervision throughout the entire duration. His continued dedication and guidance have ensured that this work is completed on time.

I would also like to thank all other members of my supervision team; Dr. Andrew Crampton, Dr. Simon Fletcher, Dr. Gary Allen, and Professor Alan Myers. Each has provided valuable input, both professional and personal. In addition I would like to express great thanks to Dr. Peter Gregory of Teesside University. His input has been hugely beneficial throughout my PhD.

Additionally, I would like to thank all other members and students of the School of Computing and Engineering for providing a constant source of entertainment and assistance when required.

Most importantly I would like to thank my family for their constant support through what has been a busy and difficult journey.

Contents

List of Figures

List of Tables

Abbreviations

AI	**A**rtificial **I**ntelligence
ADL	**A**ction **D**escription **L**anguage
CAD	**C**omputer-**A**ided **D**esign
CAM	**C**omputer-**A**ided **M**anufacturing
CAPP	**C**omputer-**A**ided **P**roduction **P**lanning
CNC	**C**omputer-**N**umerical **C**ontrol
COLIN	**CO**ntinuous-**LIN**ear numeric change
EUROPA	**E**xtensible **U**niversal **R**emote **O**perations **P**lanning **A**rchitecture
GIPO	**G**raphical **I**nterface for **P**lanning with **O**bjects
GUI	**G**raphical **U**ser **I**nterface
HPC	**H**igh **P**erformance **C**omputing
HTN	**H**ierarchical **T**ask **N**etwork
IDE	**I**ntegrated **D**evelopment **E**nvironment
IMACS	**I**nteractive **M**anufacturability **A**nalysis **C**ritiquing **S**ystem
IPC	**I**nternational **P**lanning **C**ompetition
IPPS	**I**ntegrated-**P**rocess **P**lanning and **S**cheduling
ISO	**I**nternational **S**tandards **O**rganisation
KE	**K**nowledge **E**ngineering
LP	**L**inear **P**rogramming
LPG-td	**L**ocal search for **P**lanning **G**raphs with **t**imed initial literals and **d**erived predicates
LPRPG	**L**inear **P**rogramming alongside **R**elaxed **P**lanning **G**raph
MARIO	**M**ashup **A**utomation with **R**untime **I**nvocation and **O**peration
Metric-FF	**Metric**-**F**ast-**F**orward
NA	**N**umerical **A**nalysis

OCL	**I**nternational **P**lanning **C**ompetition
OCL$_h$	**I**nternational **P**lanning **C**ompetition for **h**ierarchical models
OPTIC	**O**ptimising **P**references and **TI**me-Dependent **C**osts
PC	**P**ersonal **C**omputer
PDDL	**P**lanning **D**omain **D**efinition **L**anguage
POPF	**P**artial **O**rder **P**lanner using **F**orward-chaining
SHOP2	**S**imple **H**ierarchical **O**rdered **P**lanner
SHOP2$_{PDDL+}$	**S**imple **H**ierarchical **O**rdered **P**lanner **P**lanning **D**omain **D**efinition **L**anguage
QGG	**Q**ueensGate **G**rid
SRDT	**S**hort **R**ange **D**istance with **T**ransducer
STRIPS	**S**tanford **R**esearch **I**nstitute **P**roblem **S**olver
TEA	**T**emporally **E**xtended **A**ctions
TEG	**T**emporally **E**xtended **G**oals
TEIS	**T**emporally-**E**xtended **I**nitial **S**tates
TIL	**T**imed **I**nitial **L**iteral
TM-LPSAT	**T**emporal **M**etric - **L**inear **P**rogramming **SAT**isfiability
UML	**U**nified **M**odelling **L**anguage
UPMurphi	**U**niversal **P**lanner **M**urphi
VHPOP	**V**ersatile **H**euristic **P**artial **O**rder **P**lanner
6DOF	**S**ix **D**egrees **O**f **F**reedom

Symbols

Symbol	**Meaning**
a, A	Action, Set of actions
s, S	state, Set of State
$\rightarrow$	State-transition functions
$\sum = (S, A, \rightarrow)$	State-transition system
P	Planning problem
o, O	operator, Set of operators
$precond(o)$	Preconditions of an operator or action
$effect(o)$	Effect of an operator or action
π, Π	plan, Set of plans
$\Gamma(s)$	Set of immediate successors of s
$\mathbb{R}$	Set of real numbers
u_{DEVICE}	Uncertainty due to the measuring device in (µm)
$u_{M,DEVICE}$	Uncertainty due to temperature measurement of the measuring device (µm)
$u_{E,DEVICE}$	Uncertainty due to the expansion coefficient of the measuring device (µm)
L	Measurement length in (m)
$U_{CALIBRATION}$	Expanded uncertainty according to the calibration certificate (µm)
k	Coverage factor
$\Delta L_{MISALIGNMENT}$	Difference between measured and actual in (µm)
γ	Angle of misalignment
$u_{MISALIGNMENT}$	Uncertainty of measurement due to misalignment in (µm) due to misalignment of device with the axis
$u_{M,MACHINE\,TOOL}$	Uncertainty due to temperature measurement of the machine tool in (µm)
$u_{E,MACHINE\,TOOL}$	Uncertainty due to uncertainty of the expansion coefficient of the machine tool in (µm)

α	Expansion coefficient of the machine tool in (µm/(m.°C))
$R(\theta)$	Error range of expansion coefficient of measuring device in (µm/(m.°C))
ΔT	Difference to 20°C in degrees Celsius, $\Delta T = T - 20°C$
	Temperature in degrees Celsius (°C)
E_{VE}	Range from performing a drift test in micrometers (µm)

Chapter 1

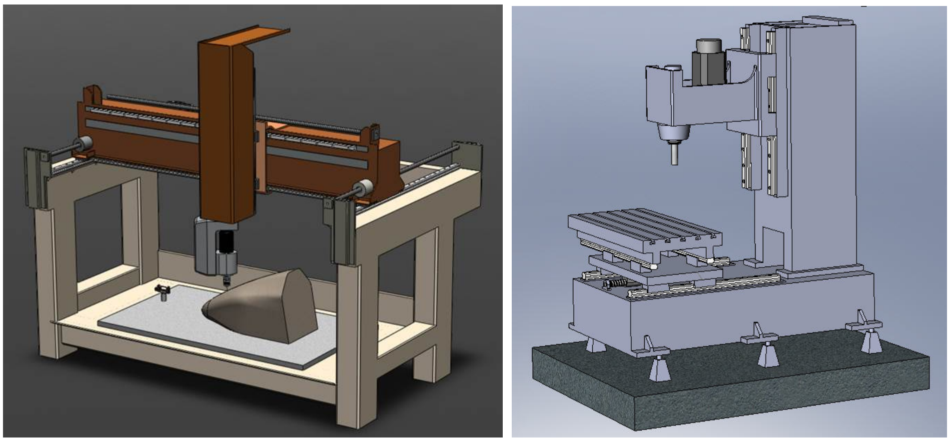
Introduction

1.1 Introduction and Overview

1.1.1 Machine Tools

A machine tool is a mechanically powered device used during subtractive manufacturing to cut material. The design and configuration of a machine tool is chosen for a particular role and is different depending, amongst other things, on the volume and complexity range of the work-pieces to be produced. A common factor throughout all configurations of machine tools is that they provide the mechanism to support and manoeuvre the cutting tool around the work-piece, although sometimes the work-piece moves around the cutter. The physical manner by which the machine moves is determined by the machine's kinematic chain. The kinematic chain will typically constitute a combination of linear and rotary axes.

Figure 1.1(a) shows an example five-axis gantry machine tool that has three linear and two rotary axes which are uses to move the tool around the work-piece (kinematic chain illustrated in Figure 2.3). Typically, this machine will be used to machine heavy, large volume work-pieces. Figure 1.1(b) shows an alternative design of a three-axis C-frame machine tool. This particular machine tool configuration consists of three linear and no rotary axes and will be used to machine smaller, less complex work-pieces than the five-axis machine.



(a) Five-axis machine tool (b) Three-axis machine tool

Figure 1.1: Example three- and five-axis machine tools

In a perfect world, a machine tool would be able to move to predicable points in three-dimensional space, resulting in a machined artefact that is geometrically identical to the designed part. However, due to tolerances in the production of machine tools and wear during operation, this is very difficult to achieve. Pseudo-static errors are the geometric positioning errors resulting from the movement of the machine tool's axes that exist when the machine tool is nominally stationary. Machine tool calibration is the process of quantifying these errors so that predictions as well as improvements of part accuracy can be made.

CNC machine tools are at the root of most metal working manufacturing systems, and are used to improve machining accuracy, lead time and cost. Therefore, the ability and availability of the machine tool should be improved in order to meet the various requirements.

1.1.2 Machine Tool Calibration

Machine tool calibration is the process of assessing a machine tool's manufacturing capabilities that includes error classification, measurement and analysis. Performing a machine tool calibration contributes to understanding and improving the machine's accuracy by providing detailed analysis of the machine's geometric capabilities which can subsequently be used to determine corrective action, and provide confidence that a given asset is capable of machining a part within a predefined tolerance.

1.1.3 Calibration Planning

Planning a full machine tool calibration requires consideration of many different influencing aspects. Decision-making made when planning for a calibration will determine the plan's duration and quality. This thesis defines a plan's duration as the time taken to execute the sequence of measurements when calibrating a machine tool. A plan's quality is in respect of the measurement's quality, including the uncertainty of measurement. Machine tool calibration can logically be broken down into sub-processes that are fundamental to the temporal and measurement quality criteria.

The following list describes an abstraction where the calibration process has been broken down logically into five sub-processes:

1. **Geometric error identification** is where the machine's configuration will be analysed to determine the geometric errors and which are required to be measured. Literature suggests that machine tool geometric errors are well understood [1, 2].

2. **Instrumentation and test method selection** is where the most suitable measurement method and instrumentation is chosen where the duration and measurement quality are the criteria. International Standards [3] and advancements in state-of-the-art instrumentation [4] can have significant influence over instrumentation and test method selection.

3. **Scheduling** is closely linked with all other sub-processes. This process will be dependent on how geometric error components contaminate each other, the temporal aspects of the selected method and instrumentation and the environmental conditions in which the chosen combination will result in the best overall measurement quality [5].

4. **Measurement** is the process where each error component is measured using the chosen test method and instrumentation at the scheduled time.

5. **Analysis and reaction** is the final process in the chain where the measurements are evaluated and corrective action is taken. For example, through the use of compensation techniques [6].

It is noticeable that these five sub-processes are tightly coupled, and to a certain extent should be treated collectively as one process. It is possible to manually treat the individual processes collectively or individually and produce valid calibration plans. The difficulty of either approach being that producing complete and optimal calibration plans is a much more complicated task. For example, consider the following scenario:

A calibration is being planned for a three-axis machine tool with a collective total of twenty-one geometric errors. There are two available measurement techniques for each geometric error, and there are two types of instrumentation available for each measurement technique. This would result in a set of eighty-four measurements available for selection to calibration the machine. If not constrained, the total number of calibration plans (permutations) is factorial (i.e $84! = 3.31424013 \times 10^{126}$).

The above example is 'simplistic' and omits possibilities such as the requirement to perform multiple, sub-measurements within the calibration of an error component. However, the example demonstrates the magnitude of calibration planning and highlights that the decision-making required for a human is cumbersome and makes it difficult to reach optimality.

At present, there is no standard way to plan a full machine tool calibration and plans are usually created *ad hoc* or in the order they were done in the past. Performing calibration plans in the same order each time can potentially bring improved consistency between calibrations allowing for better comparisons to be made.

1.1.4 Autonomous Planning and Scheduling

Planning is an abstract, explicit deliberation process that chooses and organises actions by anticipating their expected outcome. Automated planning is a branch of Artifical Intelligence (AI) that studies this deliberation process computationally and aims to provide tools that can be used to solve real-world planning problems [7].

Domain-independent planning is a form of planning where a piece of software (planner) takes as input the problem specification and knowledge about the domain in the form of an abstract model of actions. Searching for solution plans is a PSPACE hard problem [8]. PSPACE describes the computational complexity associated with decision problems that can be solved by a Turing machine using a polynomial amount of space. One key difficulty encountered with domain-independent planners is the very broad range of planning problems which could be presented, making any guidance strategy needing to be effective across the potential range of problems.

Advances in domain-independent research resulted in the formation of the International Planning Competition (IPC) [9] where state-of-the-art planners try to solve an ever increasing set of complex benchmark problems. The birth of the IPC brought a standardised formalism for describing planning domains and problems that could be used to make direct comparisons between the performance of planners. Therefore, supporting faster progress in the community. This formalism is called the Planning Domain Definition Language (PDDL) [10] and has gone through many revisions where new features, allowing for more expressive domain modelling, have been added.

1.2 Context

This thesis focuses on development, comparison and validation of automated planning models against expert knowledge. In this thesis, expert knowledge has been broken down into the two areas of industrial and academic. Industrial knowledge has been obtained from a machine tool metrology company operating predominantly in the area of machine tool calibration. Academic knowledge has been obtained from world leaders in machine tool metrology, some of whom are on International Standards comities. Collectively, their knowledge is accurate and comprehensive for the development and validation of automatically constructed calibrations plan. However, this knowledge base can easily be expanded to include the knowledge of experts with different opinions.

1.3 Scope, Motivation and Aim

Aim: To provide a method of automatically constructing machine tool calibration plans, minimising machine tool downtime and the uncertainty of measurement due to the schedule of the calibration plan.

The motivation behind this thesis is to produce a method of automatically constructing calibration plans where the machine tool is out of production for as little time as possible and there is a high confidence that the calibration result will be as precise an evaluation as possible. These two objectives can be conflicting, so the planner needs to be appropriately informed.

The scope of this work is to examine and understand how, and to what extent, automated planning and scheduling can optimise machine tool calibration planning to minimise machine downtime and measurement uncertainty. The focus is on using traditional and prevalent measurement instrumentation where scheduling can have significant impact, rather than simplifying the problem by using instrumentation that can measure multiple degrees-of-freedom simultaneously. This philosophy is well-justified in terms of normal industry practice and is extensible to situations where multi-degrees-of-freedom instruments are used for other tasks than pseudo-static geometry, such as thermal and non-rigid measurement planning. In terms of automated planning and scheduling, offline planning is considered where both the problem and domain are predefined.

1.4 Objectives

The objectives in respect to the machine tool metrology community are:

- The first objective is to develop a method of machine tool calibration planning that is capable of temporal reduction, thus reducing machine tool downtime
- Secondly, a method of machine tool calibration is required that is capable of reducing the uncertainty of measurement due to the ordering of the plan, thus increasing the value of the process.

The objectives in respect to the AI planning community are:

- Firstly, produce a model so that state-of-the-art, domain-independent automated planning tools can produce both valid and temporally optimal plans.
- Secondly, a model that encodes a way of reducing the uncertainty of measurement while considering environmental temperature as a suitable case study.

Both the temporal and uncertainty of measurement optimisation models can be used for benchmarking planner performance, helping to motivate planner development.

1.5 Contributions

The thesis presents several contributions to knowledge for both the automated planning and machine tool metrology community.

The contributions to the machine tool metrology community are:

- The use of automated planning to reduce the duration of calibration plans, thus reducing machine tool downtime. This is achieved appropriately selecting the error to measure, measurement equipment and the measurement order.
- The application of automated planning to reduce the uncertainty of measurement due to the scheduling of the calibration plan. This consists of discritizing the continuous, non-linear change in temperature into discrete, linear change that can then be interpreted using state-of-the-art automated planners.
- Providing a method of optimising calibration plans for downtime and uncertainty of measurement. This is achieved by combining both temporal (downtime) and uncertainty of measurement optimisation model.
- Comparisons between industrial and academic experts has highlighted the different philosophies behind machine tool calibration.

The contributions to the AI planning community are:

- A Hierarchical Task Network model and a series of problem instances to represent the problem of machine tool calibration. The domain and problem instances can be used as benchmarks in HTN research.
- A PDDL2.2 temporal model representing the process of machine tool calibration. The domain and problem instances can be used in future International Planning Competitions to motivate planner development.
- Method of encoding the square root function in PDDL2.2 by using a Babylonian preprocessor method.
- A PDDL2.1 model implementing the process of planning for a machine tool calibration, optimising based on the estimated uncertainty of measurement, machine tool downtime, and the average of both (multi-objective).

1.6 Thesis Structure

The next chapter (Chapter 2) in this thesis provides the theoretical background into both the temporal and uncertainty of measurement aspects of machine tool calibration. This includes a survey of the literature to determine the current state-of-the-art.

Chapter 3 investigates different planning technologies, and their applicability to machine tool calibration planning. This leads to the background of automated planning and the current state-of-the-art in automated planning tools.

In Chapter 4, an investigation is performed into automatically constructing machine tool calibration plans, whilst considering temporal optimisation. Following this, Chapter 5 investigates the feasibility of applying automated planning technology to reducing the uncertainty of measurement as a result of the calibration plan schedule.

Finally, Chapter 6 summaries the work presented in this thesis followed by a critique of it, highlighting novel aspects and motivating future research.

Chapter 2

Machine Tool Calibration

This chapter provides the background theory of machine tool calibration relevant to this thesis. It starts by providing the theory of machine tool geometric errors and their measurement. Next, the decision-making process required for producing a full machine tool calibration plan, optimised for temporal and uncertainty of measurement reduction is discussed.

2.1 Motion Errors Principle

Predetermined machine tool movement can only be achieved by deterministic and controlled machine tool motion. Motion errors are deviations from the expected machine's tool path as a result of geometric errors in the movement of the machine tool's axes throughout the working volume [1]. They cannot be eliminated, due to necessary clearances of moving parts, but they can be controlled and quantified. Motion errors in a machine tool will transfer to the machine's cutting profile and thus result in a deviation from the planned cut. The number of individual errors is determined by the machine's configuration of linear and rotary axes. Knowing the machine's accuracy for high precision manufacturing is essential [11]. Motion errors have a repeatable and non-repeatable element. The repeatable element, or systematic error, can be corrected using compensation techniques [2]. Postlethwaite, et. al. [6] demonstrate one successful implementation of error motion compensation through either a personal computer or open architecture numeric controller. Non-repeatable elements are difficult to compensate for and must to be minimised through good design.

2.1.1 Linear Axes

Error	X	Y	Z
Positioning	E_{XX}	E_{YY}	E_{ZZ}
Straightness	E_{YX}	E_{XY}	E_{XZ}
Straightness	E_{ZX}	E_{ZY}	E_{YZ}
Pitch	E_{BX}	E_{AY}	E_{CZ}
Yaw	E_{CX}	E_{CY}	E_{AZ}
Roll	E_{AX}	E_{BY}	E_{CZ}
Non-orthogonality between Y and Z			E_{C0Y}
Non-orthogonality between Z and X	E_{B0Z}		
Non-orthogonality between Z and Y		E_{A0Z}	

TABLE 2.1: Linear geometric errors for a three-axis machine

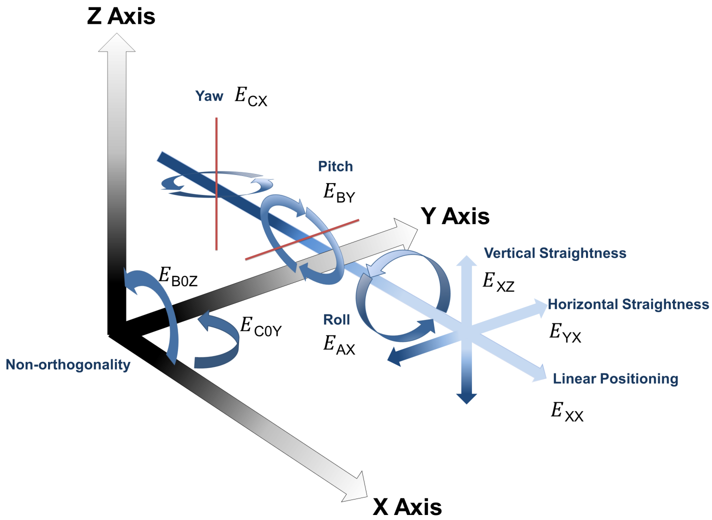


FIGURE 2.1: Linear motion errors for an X-axis

Movement of a body on a linear axis in three-dimensional space will have six-degrees-of-freedom (6DOF) [1, 2]. This refers to three translation errors (positional and two straightness), and three rotational errors (pitch, roll and yaw). For example, Figure 2.1 shows the movement of a body along the X-axis. In addition to the 6DOF, each linear axis of movement will have a non-orthogonal error with each nominally perpendicular axis. For a three-axis machine tool, there would be a total of twenty-one geometric error components. Table 2.1 shows the errors for a three axis, cross-table machine tool, as well as the ISO notation [12] for each error.

2.1.2 Rotational Axes

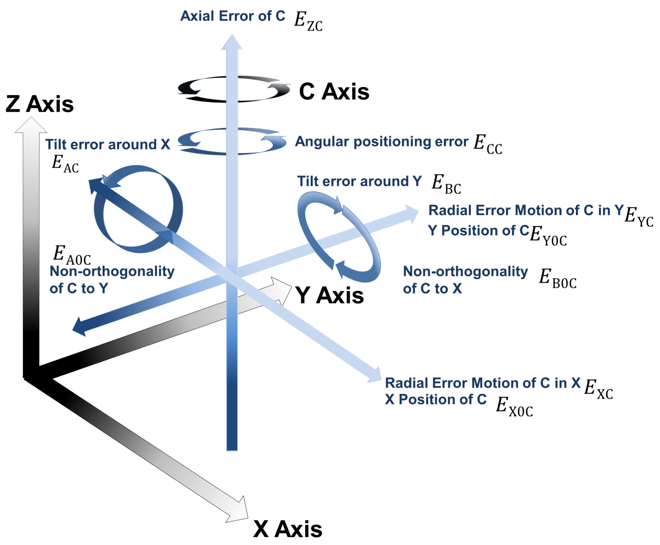


FIGURE 2.2: Rotary motion errors

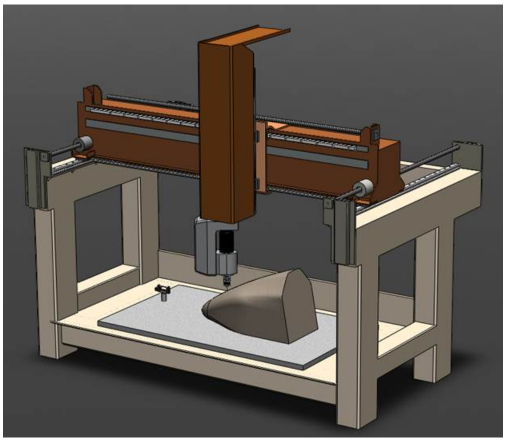
Error	C	B
Radial error motion	E_{XC}	E_{YB}
Radial error motion	E_{YC}	E_{ZB}
Axial error motion	E_{ZC}	E_{XB}
Tilt error motion	E_{AC}	E_{CB}
Tilt error motion	E_{BC}	E_{BB}
Angular positioning error motion	E_{CC}	E_{BB}
Location errors	E_{X0C}	E_{Z0B}
Location errors	E_{Y0C}	E_{Y0B}
Non-orthogonal	E_{A0C}	E_{C0B}
Non-orthogonal	E_{B0C}	E_{B0B}

TABLE 2.2: Rotary geometric errors for a five-axis gantry machine

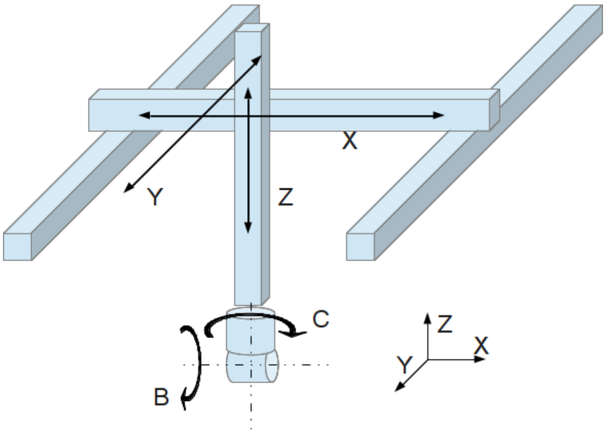
Rotational body movement in three-dimensional space will have six motion errors and two location errors. In addition, a rotary axis will also have two non-orthogonal errors. Figure 2.2 illustrates the errors of a rotational axis (C-axis) for a five-axis machine tool. Additionally, Table 2.2 shows the motion errors for the C- and A-axis of a five-axis gantry type machine tool.

2.1.3 Propagation and Interrelated Errors

While there are some very common machine tool designs, such as the three-axis C-frame (Figure 2.3(b)), there are many other configurations, some of which are bespoke configurations based on the customer's requirement. Moriwaki [13] identifies that there



(a) Machine tool



(b) Kinematic chain

FIGURE 2.3: Geiss five-axis machine tool

are two hundred and sixteen possible configurations of five-axis vertical machine centres alone, not taking into consideration different configurations of horizontal and gantry machine centres. The two hundred and sixteen possible configurations differ in the way that the machine supports and moves its axes around the work-piece, or that the work-piece moves around the machine.

The configuration of the machine tool typically constitutes a combination of linear and rotary axes. This combination will determine the kinematic chain, the geometric errors, and their propagation. An example five-axis machine tool is shown in Figure 2.3(a). This machine tool has a configuration of three linear axes and two rotary axes and its kinematic chain is illustrated in Figure 2.3(b).

It is important to consider the stacking order of the machine's axes to determine how the geometric errors manifest through the kinematic chain. For example, consider the stacking of the X and Y-axis seen in Figure 2.3(a) and 2.3(b). Any roll error (E_{AX}) of the X-axis will be amplified by the distance from the centre of rotation to the tool/work-piece interface causing a positioning error in the Y- and Z-xis directions. This distance changes with Z-axis position and is known as the Abbé offset [1]. Altering the B- and C-axis positions will change the Abbé offset and the resulting error. This has implications on the calibration plan as consideration needs to be made regarding the order in which these measurements are taken, and any effect that it might have on the quality of the measurement value.

2.1.4 Environmental Temperature

Changing environmental temperature will cause thermal expansion of the machine tool, work-piece and potentially any measurement equipment. Thermal expansion of the machine tool when the work-piece has a different coefficient of thermal expansion or the machine structure is asymmetric can result in non-linear thermal expansions, resulting in complex dynamics between the work-piece and machine tool [1]. When measuring a machine tool error it is important to take into consideration the temperature since it will significantly affect the results.

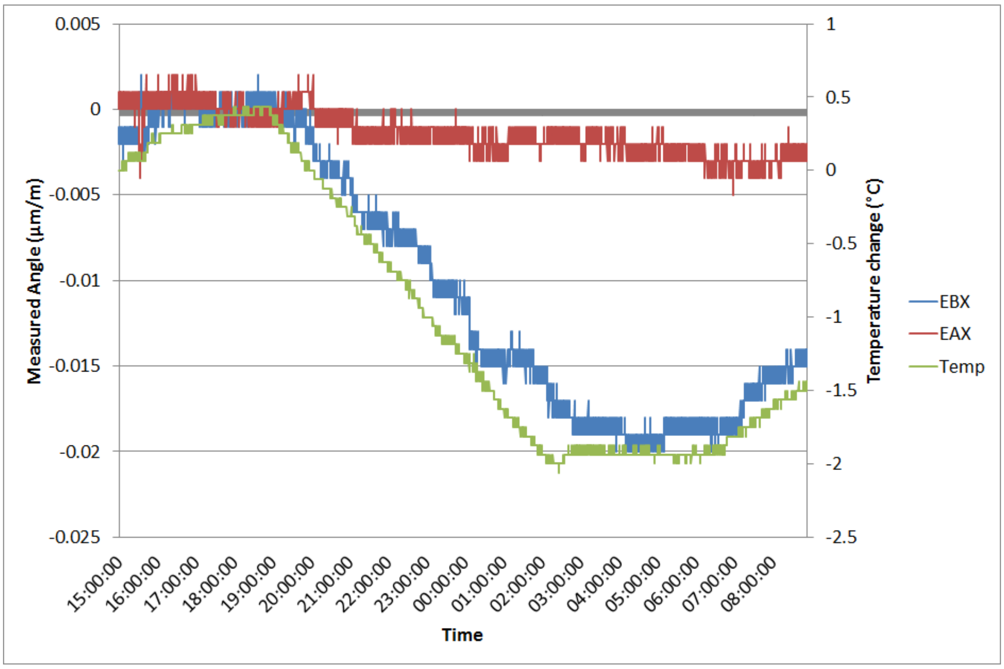


FIGURE 2.4: Effect of temperature on E_{BX} and E_{AX}

An example error induced by environmental temperature fluctuations is shown in Figure 2.4, where the E_{BX} of a C-type machine tool changes by around 2µm/m for approximately a 2°C ambient temperature change. Such an error will affect machining, but will also, depending upon the instant of measurement, affect the expected or calibrated performance of the machine.

Methods of modelling thermal aspects of machine tools and their environments have made it possible to apply corrective compensation. Mian et. al. [14] develop an offline modelling technique using finite element analysis that can predict the machine's thermal expansion. In addition, they provide a method to predict sensitive points where temperature sensors should be placed to provide real-time compensation [15]. Information can be collected from these sensors and then used when planning a machine tool calibration.

2.1.5 Tolerance of Errors Components

A tolerance is the permissible limit for a physical dimension [12]. Typically, machine tool manufacturers will provide the tolerances for the error components based on the client's requirements [1]. Each error components will have a tolerance to which the measurement result will be compared. Out-of-tolerance error components will result in the requirement for corrective action. Whereas results that are within the tolerance will require no further action.

2.2 Generic View of Geometric Error Measurement

In the previous section, the geometric errors for both linear and rotary axes have been defined. In this thesis, a generic process of measuring these errors is discussed. The motivation behind this view-point is not to limit the work to a predefined set of measurement techniques. Instead, the developed work should be adaptable to advancements in measurement technology. In this section, common decision criteria for choosing a measurement technique and instrumentation for all machine tool geometric error measurements are discussed.

2.2.1 Method of Measurement

Selecting the method of measurement is important to ensuring the correctness of the measurement procedure for feed axes. International Standards Organisation (ISO) provide guide 230 part 1, 2 and 7 [3, 12, 16] to govern the correct assignment and use of measurement methods and instrumentation. Compliance to ISO guides strives to improve the quality of the measurement as well as allowing for better comparisons to be made between machines in the knowledge that the measurements have been taken using similar methods and traceable equipment. In the following section, decision criteria for selecting the method of measurement are discussed.

2.2.1.1 Direct or Indirect Measurement

Direct measurement is where the error component of interest is measured directly, whereas indirect measurement is where a error component is deduced from measuring other errors. For example, one direct method of measuring non-orthogonality is to use a Short Range Displacement Transducer (SRDT) and a granite square, taking measurements in the two axes within the non-orthogonal plane. An alternative indirect measurement techniques would be to measure non-orthogonality by using the double ball bar method where we measure the circularity of a circle in a plane and extract the non-orthogonality by using the lengths of the two diagonals [11]. Multilateration measurement [17] is another example of indirect measurement where individual axis errors are deduced from a comprehensive measurement plan.

2.2.1.2 Sampling, Interval, Dwell and Feedrate

Different measurement methods require a different number of samples i (targets) to be taken. ISO 230 part 2 [3] recommends that for axes of travel up to 2000mm, a minimum of five target positions per metre and an overall minimum of five target positions should be selected. For axes of travel greater than 2000mm, a sampling interval $p = 250$mm should be used. The nominal interval p (stepsize) between the two targets, and based on ISO recommendation [3], should also include a random number, r, with $\pm$ the amplitude of possible periodic errors, to ensure that they are adequately sampled. If no information is available regarding the periodic error, r should be within a magnitude $\pm 30\%$ of p. Equation 2.1 shows the general form of the target positions.

$$p_i = (i - 1)p + r \tag{2.1}$$

Recommendation governing the location of the first and last position, dwell time and feedrate are less clear. ISO 230 part 2 suggests that this criteria is to be agreed between the supplier and the manufacturer [3]. This can prove problematic because deciding the the location of the positions, dwell time and feedrate can be dependent on the machine's configuration, use, and previous calibration results. Expert knowledge is required to decide these parameters effectively.

2.2.2 Instrumentation

Instrumentation will be used to perform the measurement and acquire the result. The state-of-the-art in instrumentation is continuously changing, therefore the selection of more efficient and accurate instrumentation can have a significant impact on the time taken to perform the measurement and its quality. The selection of instrumentation is closely linked with the method of measurement. For example, Figure 2.5 shows an illustration take from ISO230 part 2 [3] of measuring straightness using laser interferometry. Using this method of measuring straightness using a laser interferometer clearly requires the use of a laser interferometer and requisite optics. However, if more than one interferometer or short or long range optics are available, choosing the one best suited to the measurement and the desired level of accuracy is important.

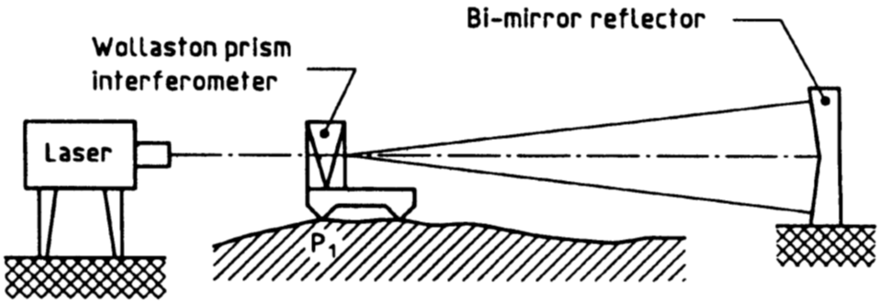


FIGURE 2.5: Measuring straightness using laser interferometry (ISO230 part 2, 2006)

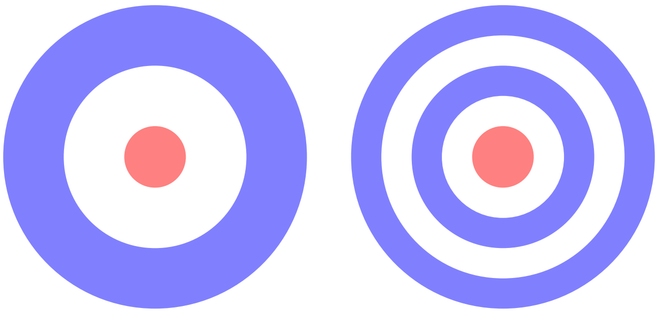
To illustrate the impact of using different instrumentation to perform the same measurement, consider the following example of measuring straightness using three possible different set-ups for measuring the straightness of a linear axis: (1) mechanical straight edge and SRDT, (2) laser interferometer, and (3) taut wire. All three methods are suited better to different measurements. The laser interferometer will be better suited to measuring long axes of travel, whereas the taut wire technique is better suited to measuring machine tools where a laser cannot be used because of physical restrictions or constantly changing environmental conditions. For short axes of travel, the mechanical straight edge and SRDT can be much easier to set-up and performed than other methods.

2.2.2.1 Degrees of Freedom

Instrumentation will measure in one or more Degrees Of Freedom (DOF) depending upon the method of measurement. It is most common that instrumentation can measure in only one DOF. For example, when measuring non-orthogonality using a SRDT and a granite square, the SRDT can only measure in one DOF. However, advancements in state-of-the-art instrumentation have made it possible to measure multiple error components simultaneously. For example, the Wyler Leveltronic inclination measurement device can measure in 1DOF. However, the recently developed Wyler Zeromatic inclination measurement device can simultaneously measure in 2DOF [18]. Using this instrument, it is possible to measure both the pitch and roll angular deviation of a linear axis simultaneously.

2.2.2.2 Resolution, Accuracy and Precision

When selecting instrumentation to use, there are parameters that should be considered to help improve the quality of the measurement and determine the instrument's fitness for purpose. The following list discusses this criteria:



(a) Good resolution (b) Higher resolution

FIGURE 2.6: Resolution

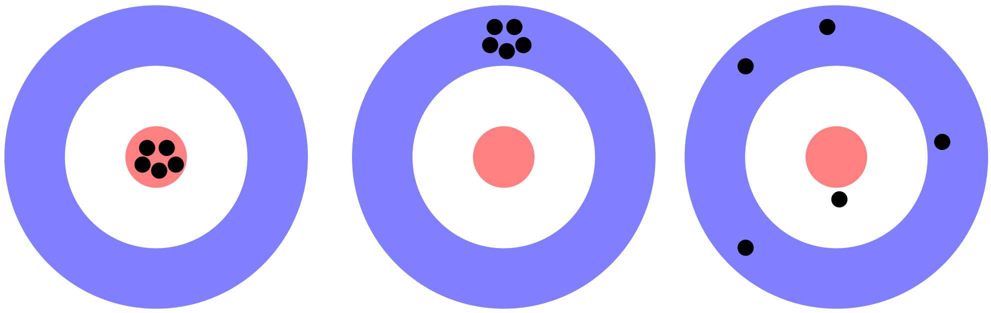
1. **Resolution** is the smallest detectable increment that the device can measure. Figure 2.6 shows two different resolutions. The first (Figure 2.6(a)) is a resolution good enough for the required measurement, and the second showing a higher resolution (Figure 2.6(b)). It is important to consider resolution with respect to the measurement tolerance. However, using instrumentation with too high a resolution can result in the unnecessary use of expensive equipment which could be

better used elsewhere (in general, selecting an instrument with a higher resolution is good practice). Also, there are instances where selecting an instrument with a high resolution can increase the time for the instrumentation to stabilise, making it difficult to read the value efficiently. An example of this is attempting to use a high resolution dial test indicator on a poor axis causing the indicator to vibrate making it impossible to read the value.

2. **Accuracy** is the closeness of a measurement to the actual value being measured. Choosing the correct instrumentation that has accuracy levels inside the measurement tolerance is essential. Failure to do could result in false acceptance or rejection of measured errors when compared to their tolerance.

3. **Precision** of measurement instrumentation is the degree by which the measurement can be repeated under changed conditions producing the same result.

When considering precision and accuracy, it is important to consider them together to maintain good measurement quality. Figure 2.7 illustrates this by showing three possible scenarios of accuracy and precision. The best case is to have high accuracy and good precision (Figure 2.7(a)). However, sometimes the instrumentation might not be capable of this. In which case, poor accuracy and good precision (2.7(b)) is better than poor accuracy and poor precision (2.7(c)) since the systematic error can be calibrated and the measurement result compensated accordingly.



(a) Good accuracy, good precision (b) Poor accuracy, good precision (c) Poor accuracy, poor precision

FIGURE 2.7: Accuracy Vs. Precision

2.2.3 Temporal Aspects

A machine tool will not be available for normal manufacturing while a calibration is taking place. For this reason, it is important to consider the temporal aspects when performing a measurement. Measuring an error component has several temporal implications [2]. The following list describes the different phases associated with all measurements.

1. **Set-up** of the equipment is normally a manual process where the instrumentation will be taken from its protective packaging and set-up on the machine for use. This duration includes time taken for fine tuning of the instrumentation (e.g laser beam alignment) it can also include the time taken for the instrument to stabilise in terms of self-heating and stabilising to the environmental conditions, although with good planning can this can be achieved "offline" without the need for the machine.

2. **Measurement** of the component error can be manual or automated, but either way it will still require time to complete. During measurement, the measurement data as well as any necessary environmental data will be recorded. Additionally the duration will be affected by the sampling, interval, dwell and feedrate (Section 2.2.1.2)

3. **Removal, Adjustment and Reposition** of equipment are post-measurement durations. Removal is simply the time taken to remove the instrumentation and package it suitable for storage. Adjustment and repositioning are durations for when the instrumentation is require to be adjusted to measure another component error. For example, after measuring linear positioning using a laser interferometer the optics could be changed and the laser realigned without having to go through the complete set-up. Repositioning is where the instrument needs moving to perform another measurement for the same component error. For example, when measuring straightness of a long axis using a SRDT and a granite straight edge it is possible that the straight edge will need to be repositioned multiple times to cover a sufficient amount of travel. Another example is using a granite square and SRDT; the square can be adjusted to measure another axis, taking less time than setting the equipment up in the first instance.

2.3 Uncertainty of Measurement

Uncertainty of measurement is a parameter associated with the result of a measurement that characterises the dispersion of the values that could reasonably be attributed to the measurand [19]. For example, a thermometer might have an uncertainty value of ±0.1°C. Therefore, it can be stated that when the thermometer is displaying 20°C, it is actually 20°C ±0.1°C with a confidence level of 95% where the confidence level is determined by the distribution and knowledge of the system. The confidence value states the certainty that the true value is within the margin [20]. In the previous example, there is a 95% certainty that the temperature is between 19.9°C and 20.1°C. Quantifying and reducing uncertainty of measurement is an important task since it is both required to be reported on the calibration certificate. More importantly, it is required to determine whether the measurement is suitable to establish whether the machine is capable of meeting its tolerances.

2.3.1 Tolerance Conformance

Tolerances in the machine's positioning capabilities are particularly important because they are transferred directly to the achievable tolerance of the work-piece during machining. High accuracy and precision manufacturing such as the aerospace industry have tight, micron-level tolerances. Therefore, reducing the estimated uncertainty of measurement will have an effect on tolerance evaluation, and can help reduce repeating measurements and false rejections. .

Figure 2.8 illustrates the conformance (green) and non-conformance (red) zones based on the uncertainty value and the lower and upper tolerance limit [21]. The remainder is uncertain. From this illustration false acceptance and rejections can be visualised. False acceptance could occur if the measurement value is out-of-tolerance but the uncertainty of measurement brings it into tolerance. Conversely, false rejection could occur if the measured value is in tolerance but the uncertainty of the measurement makes it out-of-tolerance. Therefore, only measurement values that fall within the conformance zone are certain, within the given confidence level, to be within the tolerance. Minimising the uncertainty of measurement can increase the conformance zone reducing false acceptance and rejection.

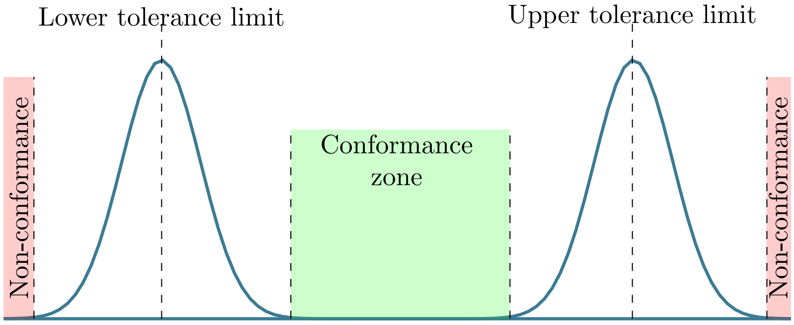


FIGURE 2.8: Conformance and non-conformance zones for two-sided tolerance.

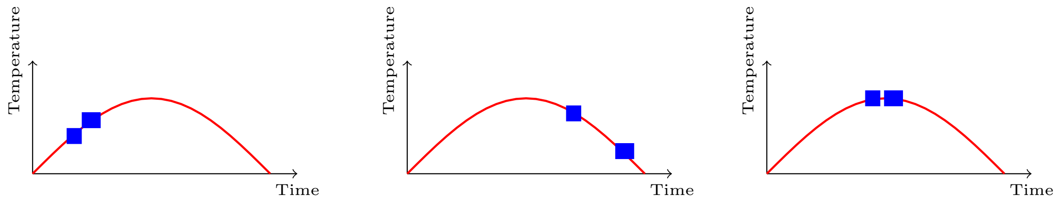
For example, using the example of the thermometer in Section 2.3, if a temperature was required to be between 20 ±2°C it would only conform if the reading on the thermometer was between 18.1°C and 21.9°C. This is because the device has a 0.1°C uncertainty. Likewise if the uncertainty were ±0.5°C, the reading will only conform if it is between 18.5°C and 21.5°C. Therefore, a tolerance of ±2°C is reduced to conformance of±1.5°C.

2.3.2 Contributors

The uncertainty of measurement u_c is a combination of the individual uncertainties u_i from (1) the measurement's environment, (2) the measurement method, and (3) the machine [20, 22]. In this section, a selection of individual contributors that affect the overall uncertainty of measurement are discussed.

2.3.2.1 Effect of Environmental Temperature

Environmental temperature is an important aspect of estimating the uncertainty of measurement, especially when estimating for interrelated measurements. Figure 2.9 illustrates three potential scenarios that occur when planning for the effect of temperature on interrelated measurements. All three scenarios only consider two measurements for illustration purposes. A full calibration plan will consist of considerable more interrelated measurements. Figure 2.9(a) illustrates temperature rise during the measurement of two interrelated measurements. Conversely, Figure 2.9(b) illustrates temperature decrease while taking two interrelated measurements. Figure 2.9(c) illustrates the ideal case when both interrelated measurements are taken where the temperature has stabilised. For example, measuring both straightness errors E_{ZX} and E_{YX} are interrelated with measuring the non-orthogonal error E_{COY}. Both straightness errors are required when calculating the non-orthogonal error as they need to be removed from the values measured using the mechanical square and SRDT. In the ideal case, both straightness error measurements will be followed consecutively by the non-orthogonal measurement where the variation in temperature is at its lowest, as in Figure 2.9(c). If the three interrelated measurements take place when the temperature is either increasing or decreasing it can have adverse effects on the estimated uncertainty because as the temperature changes, so will the error value.



(a) (b) (c)

FIGURE 2.9: Three example temperature scenarios

2.3.2.2 Measurement Method and Instrumentation

Different measurement methods can have different uncertainties because of their principle of operation, but also because of their complexity and difficulty to set-up and perform. For example, aligning a laser over a long distance when measuring straightness would have a larger uncertainty when compared to the same measurement over a short distance. Included in the uncertainty for the measurement method is the uncertainties for any instrumentation. Instrumentation used should have a calibration certificate that can be used in estimation calculation.

2.3.2.3 Machine Tool

The machine tool itself has uncertainty contributors that must be included in the estimation. One contributor that requires consideration is the uncertainty due to the coefficient of expansion of the machine tool $u_{E,MACHINETOOL}$ from 20°C, and its measurement $u_{M,MACHINETOOL}$, which is the uncertainty due to the temperature measurement that includes the uncertainty of the temperature measurement device and the point of measurement.

Additionally, interrelated measurements require detailed consideration. As described in Section 2.1.3, geometric errors can manifest through the kinematic chain to be evident in other geometric errors. This means that any error component that propagates through to other error component should be included in their measurement's uncertainty estimation. If the propagating error has not been measured yet, then an estimated uncertainty as recommended in ISO [23] should be used.

2.3.3 Estimation

Taking into consideration the measurement method, measurement equipment and both the surroundings, it is possible to produce a method that can be used to estimate the uncertainty of measurement. In this section, a method to calculate the estimated uncertainty is described. Additionally, the contributing factors to the uncertainty of measurement are discussed.

One known method, recommended by ISO, involves combining the individual uncertainties using the root of the sum of squares to produce a combined uncertainty u_c. In this thesis, Equation 2.2 as described by ISO [23] is used for calculating u_c.

$$u_c = \sqrt{\sum u_i^2} \tag{2.2}$$

Where u_c is the combined standard uncertainty in micrometers (µm), and u_i is the standard uncertainty of uncorrelated contributor, i, in micrometers (µm).

The next stage is to calculate the expanded uncertainty U which specifies the uncertainty value and a confidence level. To calculate this, the combined standard uncertainty is multiplied by an appropriate coverage factor. This factor represents the confidence level and the probability distribution of the combined standard uncertainty. When it can be assumed to be normal, the value of $k = 2$ defines an interval having a confidence level of approximately 95%. The k factor for more critical applications, such as measurements taken in the pharmaceutical sector where the implications of uncertainty of measurement can be critical, have a value of $k = 3$ having a confidence level of approximately 99.7%.

2.3.4 Example: Uncertainty of Non-orthogonal Measurement

The uncertainty of non-orthogonal measurement is presented as an example of how uncertainty can propagate through interrelated geometric errors. ISO 230 part 1 [12] defines non-orthogonality as "the difference between the inclination of the reference straight line of the trajectory of the functional point of a linear moving component with respect to its corresponding principle axis of linear motion and (in relation to) the inclination of the reference straight line of the trajectory of the functional point of another linear moving component with respect to its corresponding principle axis of linear motion."

The non-orthogonal error can be measured using a variety of instrumentation and test methods. In this example the method of measuring non-orthogonality considered is using a mechanical square and SRDT. Non-orthogonality is contaminated by other geometric errors, in particular the straightness of each axis in the plane of measurement. When measuring non-orthogonality it is important to consider the uncertainty of measurement arising due to the change in the straightness of axes between measurements.

Non-orthogonality between two nominally perpendicular axes can be measured using a mechanical square and SRDT. This method involves placing a mechanical square in the plane of the non-orthogonal error so that it is perpendicular with the two linear axes. A SRDT will then be attached the the spindle of the machine tool and readings will then be taken at designated points along the perpendicular face of the mechanical square. The measured values then require processing using mathematical techniques such as least square to remove misalignment of the mechanical square with the reference axis of the machine tool. In this section a few of the contributing uncertainties are included to demonstrate possible sources that contribute to the uncertainty of measurement. The uncertainty of the SRDT ($U_{Device\ SRDT}$) can be calculated using the calibration certificate and Equation 2.3. The uncertainty of the expansion of the mounting post ($U_{SRDT\ POST}$) is calculated based on the known uncertainty of measurement due to the coefficient of thermal expansion.

$$U_{Device\ SRDT} = \frac{U_{CALIBRATION}}{k} \tag{2.3}$$

Similarly, the uncertainty for the mechanical straight edge ($U_{Device\ STR}$) can be calculated using Equation 2.4 and the straight edge's calibration certificate.

$$U_{Device\ STR} = \frac{U_{CALIBRATION}}{k} \tag{2.4}$$

As illustrated in the measurement data shown in Figure 2.10, the effect that temperature has on the machine tool ($u_{E,MACHINETOOL}$) is acquired by monitoring the relationship between straightness movement and temperature. The effect of temperature on this straightness can then be used when estimating the uncertainty of the non-orthogonal measurement. If empirical data is not available, values can be acquired from ISO230 part 9 [23] where it is recommended that the ISO tolerance for straightness deviation is taken as a first estimate. The combined standard uncertainty (u_c) can then be calculated using Equation 2.2.

To minimise the uncertainty of measurement, the change in machine temperature between straightness and non-orthogonal measurements should be minimal. Ideally this is achieved by having a stable environment. However, when it is not possible it can be achieved by scheduling the tasks accordingly.

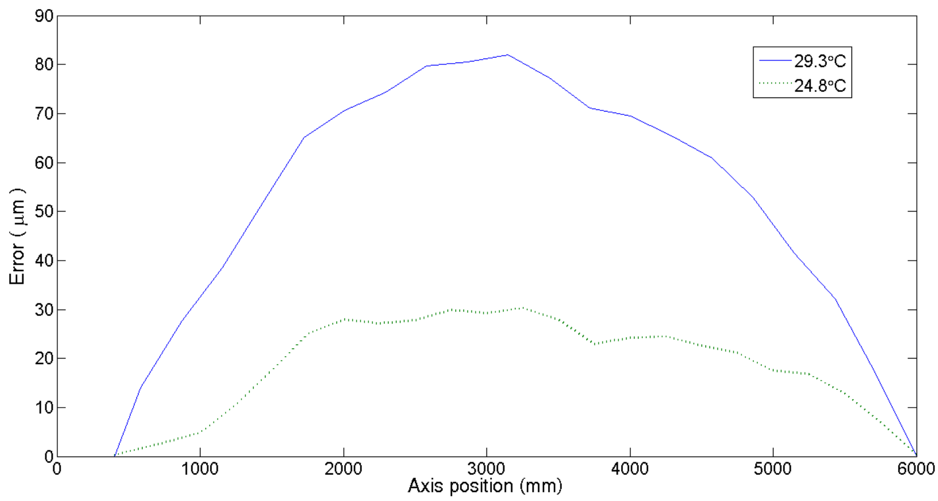
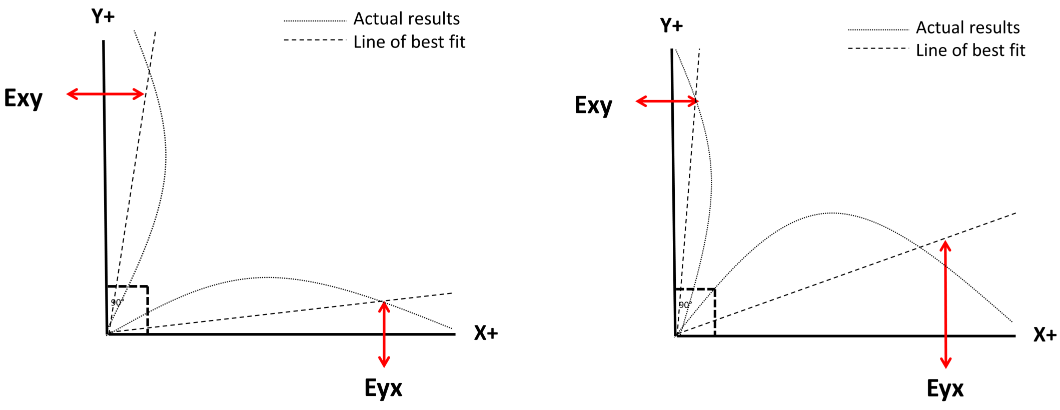


FIGURE 2.10: Y-axis straightness change (E_{XY}) due to temperature

Figure 2.10 shows a real-world example of the Y-axis straightness in the X-axis direction measured on a gantry milling machine at two different temperatures. From this figure, it is evident that the error quadruples with a 4.5 ^{o}C increase in temperature. Figure 2.11(a) and 2.11(b) show different non-orthogonality results for the same measurement taken in the two different temperature conditions. In Figure 2.11(a) and 2.11(b) the solid, curved lines represent the actual measurement data and the straight dashed lines represent a line of best fit. The shape of the line displaying the measured data shows how the non-orthogonal error between the two axes change throughout the travel of the axes. Figure 2.11(a) was performed during stable temperature, whereas Figure 2.11(a) was performed in conditions where the temperature is increasing, resulting in the E_{YX} straightness error increasing. The measuring order of the influential errors will affect the estimated uncertainty. For example, if the non-orthogonality measurement is taking place after the measurement of all the other errors, not only their uncertainties, but also any uncertainty in their change over the time period should be included when calculating the non-orthogonality uncertainty of measurement. If the non-orthogonality measurement is taking place without that of the other errors, the uncertainty calculation would have to either include the maximum permissible value for each error, or be calculated once these values and their uncertainties are known. This value can be acquired by monitoring the change in angular and straightness error with respect to temperature over a given time period.



(a) (b)

FIGURE 2.11: Influence of other geometric errors on the XY non-orthogonality error at different temperatures

2.4 Calibration Planning

Previously in this chapter, the decision making aspects involved when planning for a machine tool calibration have been discussed individually. In the next section, the process of calibration planning which combines the following functions into one unified process.

1. **Identification** of geometric errors to measure based on the machine's configuration and use.
2. **Optimal** test method and instrumentation to select based on temporal and uncertainty criteria.
3. **Scheduling** against changing environmental temperature to produce optimum calibration plan.

2.4.1 Calibration Philosophies

The philosophy for calibrating machine tools can differ greatly depending on the accuracy requirement of the application [24]. Additionally, the use of the machine tool can significantly affect the calibration philosophy. Many areas of manufacturing such as medical and aerospace are machining parts to micron-level tolerances, therefore their machines must be accurate enough to achieve this.

The motivation for performing a machine tool calibration can also greatly affect the structure of the calibration plan. In an industrial setting, strong temporal restrictions will govern the structure of a calibration plan [25], whereas in a research environment, measurement quality might be more important and temporal restrictions will be relaxed. Additionally, in industry sometimes achieving calibration certification is the main philosophy, and the cost and quality is of little concern as long as certification can be achieved.

2.4.2 Expert Calibration Comparison

Little comparison between expert calibration plans has been made. For this reason, this section contains a controlled case study performed to compare the calibration plans produced by both an industrial and academic expert. In comparison, evaluation of both expert's plans will take place to examine instrumentation selection and measurement ordering. This casestudy is a snapshot of two measurements and is not an exhaustive survey. However, the snapshot highlights the key differences stemming from the different calibration philosophies.

2.4.2.1 Calibration Scenario

The considered problem is the calibration of a five-axis gantry machine as seen in Figure 2.3. In total, the machine has 41 pseudo-static geometric errors. Each linear axis has the component errors that can be seen in Figure 2.1 and each rotary axis has the component errors that can be seen in Figure 2.2. In addition to the geometric errors of the linear (X-, Y- and Z-axis) and the rotary (C- and A-axis) pseudo-static geometric errors, the spindle (S-axis) errors will also be considered. The spindle can be considered as an additional rotary axis requiring the measurement of the:

1. Spindle centre of rotation in X and Y.
2. Spindle axial run out in Z.
3. Spindle radial run out in X and Y.
4. Spindle taper run out.

2.4.2.2 Expert Plans

Figure 2.12 shows the ordering and expected duration of the two expert calibration plans. The first by an industrial expert (orange), and the second by an academic expert with extensive experience in on-machine measurement (blue). These plans have been validated by performing the measurements to verify their feasibility.

It is noticeable from Figure 2.12 that both the industrial and academic calibration plans have differences in terms of ordering, test duration and equipment selection.

Firstly, it is necessary to establish the difference related to the different motivation behind performing the calibration. The industrial calibration plan is ordered in the way that the geometric errors manifest. This method allows them to correct an error that they might discover during their work, minimising the effect that the modification has on the errors that have already been tested. The academic's motivation is different, they

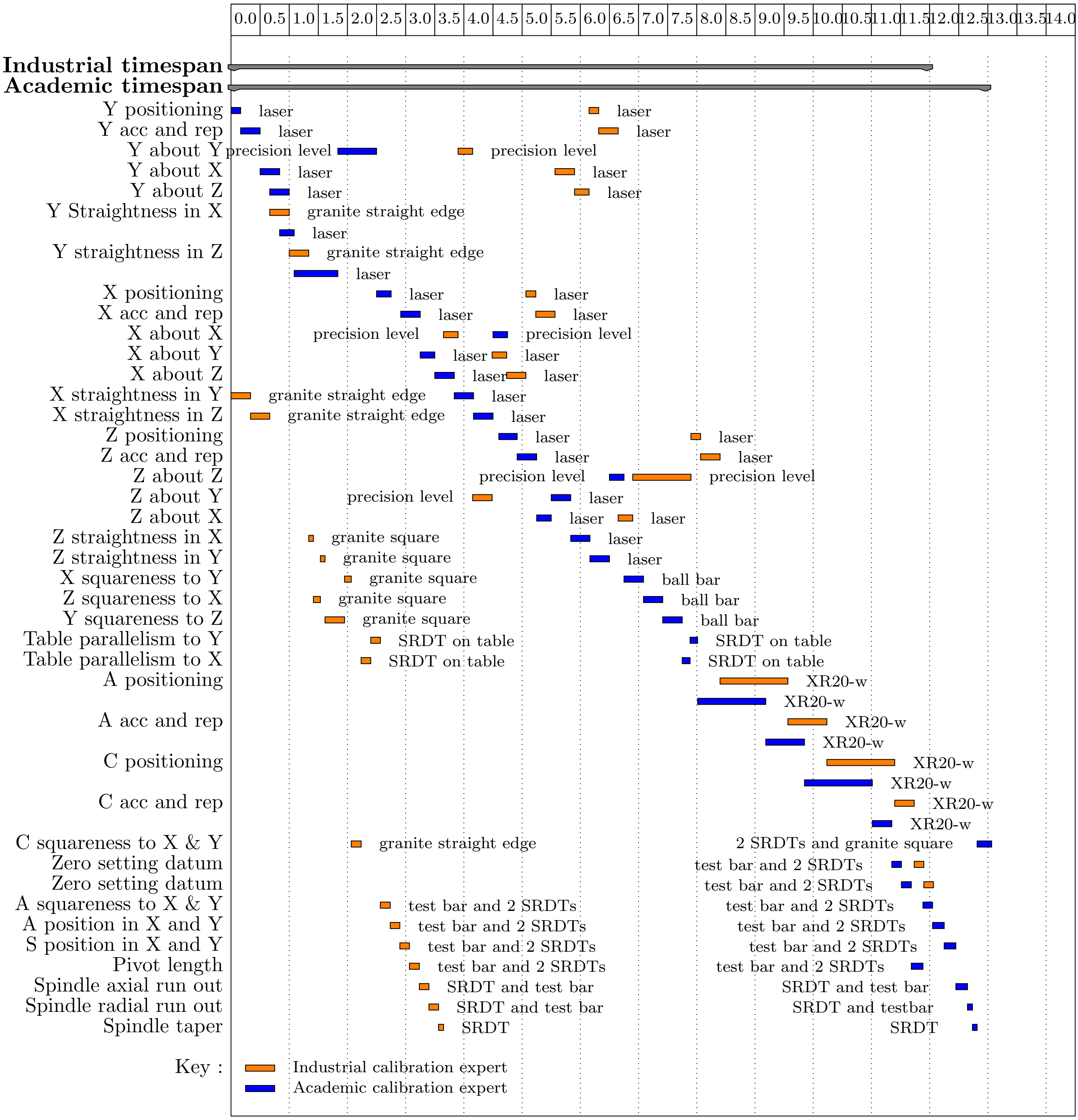


Figure 2.12: Industrial and academic calibration plan comparison

will perform all the measurements first and then analyse the data before recommending any corrective action.

The industrial calibration plan is also subject to the resource constraints of other concurrent calibration jobs, so company-wide resource allocation can have a significant impact on the produced calibration plan. It is also possible that the academics' might be working on more than one calibration job at one time, but at the time that the calibration plan was produced for the five-axis machine in question, they were not. Additionally, the academic calibration plan was produced under the psychological reason of performing the measurements in the most convenient order, measuring the largest axes first.

It is also evident from Figure 2.12 that the industrial calibration plan contains the use of a granite straight edge to test for the straightness component errors because they are more comfortable with it. The academics, on the other hand, use the laser interferometer because using the granite straight-edge for a machine tool with a large axis travel will take more time, whereas the laser can measure an axis with a longer travel without adjustment. Another difference is the selection of the equipment for measuring the non-orthogonal errors. The industrial calibration plan contains the use of a granite square, meanwhile the academic's plan makes use of a ball bar. This is due to the ball bar being more convenient to use for the academics, and that it possesses the capability to also capture data regarding the dynamic errors of the machine tool.

2.4.3 State-of-the-art in Calibration Planning

The complexity associated with machine tool geometric error measurement [1, 2] and the desire to reduce measurement uncertainty [5, 26] and machine tool downtime are well known for individual measurements. However, surveying the literature suggests that less well known is the potential to reduce machine tool downtime and the uncertainty of measurement by intelligent construction of the calibration plan. In this section, the state-of-the-art in terms of error identification, measurement techniques and instrumentation, and intelligent calibration planning are discussed.

2.4.3.1 Method and Measurement

Bringmann et al. [5, 26] have identified that current ISO 230 part 2 [3] is based on sequential testing of single geometric component errors. However, an exception is made for ISO 230 part 4 [27] where several machine errors are tested together while the machine tool is performing multi axis movement. Bringmann et al. [5] then continue to describe the importance of interrelated errors using the example of linear yaw deviation effecting the non-orthogonality measurement at different positions in the plane of non-orthogonality measurement. The authors identify that this process is time consuming, and in response have shown the calibration of a machine tool using a 3D-ball plate where the amplification of interrelated measurements can be identified. However, when such approach cannot be used, they suggest using a Monte Carlo simulation that uses an approximation of the machine tool, the measurement and the machine's performance after calibration to estimate the uncertainty of measurement. Performing the Monte Carlo simulation sufficiently often will produce a distribution for the uncertainty of the identified errors. This work succeeded in producing optimal measurement plans when considering interrelated measurements by suggesting the use of a 3D-ball plate, or measurement uncertainty of measurement reduction using Monte Carlo simulation. In one example, the uncertainty of measurement for the X-axis linear positioning error E_{XX} is reduced from 30 µm to 10µm The limitation of this work is that is that it is concerned with achieving the best possible measurement sequence with respect to the uncertainty of measurement at all costs, ignoring machine tool downtime.

Muelaner et al. [28] produced a method of large volume instrumentation selection and measurability analysis. This work is not explicitly for machine tool calibration, but does considers the suitability of instrumentation based on measurement method and instrumentation criteria. This implementation results in a prototype piece of software capable of finding the best instrumentation and measurement method from an internal database. Although this work is capable of always finding the optimum selection based on the predefined criteria, it pays no consideration to temporal aspects. Additionally, the produced model and software does not take any consideration to interrelated measurements, allowing for optimal sequencing.

Recent advancements in measurement instrumentation have demonstrated how multiple error components can be measured simultaneously using the same instrument. These techniques can simplify the calibration planning process as the calibration will require less time to complete, making the duration between measurements lower. Therefore, the likelihood of being able to schedule the measurements to happen over a duration that is temperature-stable is increased. However, this significantly depends on the machine tool's environment. For example, the API XD™ [29] allows for measuring all 6DOF simultaneously for one linear horizontal axis from a single set-up.

Other methods of machine tool calibration include being able to measure indirectly all the geometric error components simultaneously. One such method is the Etalon laserTRACER [4] which has a linear measurement resolution of 0.001µm for measuring axes up to 15 m in length. The laserTracer tracks the actual path of the machine tool throughout the entire working volume. This is done by attaching a reflector on the machine tool at the tool fixing point. From the acquired information, the system can perform a full calibration of multiple axis Cartesian machines. This includes all six-degrees-of-freedom and the non-orthogonal error. Using this method to calibrate a machine tool reduce the requirement for the use of multiple instrumentation and measurement methods, therefore, the type of calibration planning discussed in this thesis is reduced. However, due to the expensive cost of such equipment, the majority of machine tool owners and providers of calibration services will not yet own such a device.

2.4.3.2 Calibration Planning

As previously discussed, ISO guides are available to suggest measurement techniques to use when performing machine tool calibrations. However, one limitation of these guides is that they concentrate on calibration as a sequence of separate measurements, rather than the view-point of a set of closely linked measurements. The view-point of considering calibration as a sequence of separate measurements is adequate for performing calibrations. However, it is difficult to consider optimality of the calibration plan in this way. There is an absence of literature surrounding machine tool calibration suggesting that research has been performed into treating machine tool calibration as a set of close linked measurements, leading to optimality in terms of machine tool downtime and the uncertainty of measurement.

2.4.3.3 Calibration Software

There are software packages provided by machine tool metrology companies that are capable of data capture and analysis. One example is the software for communicating with the Renishaw QC-20W ballbar [30]. This software is capable of assisting with measurement frequency, data acquisition, reporting and viewing the measurement history of a specific machine tool. Software packages like these are essential for collecting and processing the data from complex measurement instrumentation. However, these packages take no consideration to the scheduling for a full machine tool calibration when using multiple instrumentation. The literature suggests that no package is currently available for planning a full machine tool calibration to determine the optimal sequence of measurements.

In the metrology community, there are many commercial software packages that are available to aid with the management of device and instrumentation calibration. Typically, these packages will provide the means of regulating the frequency of calibration, acquisition and reporting of the data. These features often aid a company that operates in a heavily regulated industry to maintain ISO compliance. For example, Beamax CMX is a universal software package that can be configured to manage all types of calibration [31]. This software package is beneficial for a wide range of calibration processes, However, the extent to which it attempts to optimise the calibration plans is not clear. From the literature surrounding this tool, there is no evidence to suggest that it is capable of intelligently reasoning with available instrumentation to measure error components while considering the external environment.

2.5 Chapter Summary

In this chapter, the complexity involved within machine tool calibration has been described. This includes describing machine tool configurations, geometric errors, their contribution to tolerance conformance, and how environment temperature affects them. This resulted in the production of a generic view of error measurement. The uncertainty of measurement was then discussed, detailing how it can be estimated. This includes the consideration of the effect of changing temperature. An example of measuring non-orthogonality using a short range displacement transducer is provided to illustrate the process.

Following this, calibration planning was discussed, detailing different calibration philosophies and their effect on the structure of the calibration plan. A comparison between two different experts, who have two different philosophies, was then performed. It was found that the industrial expert will structure their calibration plan to measure the error components that can affect the measurement of other error components first. This allows them to make corrections after measurement which could have otherwise influenced the result of subsequent measurements. The academics follow the philosophy of measuring all the error components before analysing and implementing any corrective action.

The literature survey suggests that although both industrial and academic experts are currently producing valid machine tool calibration plans, there is little evidence to suggest that they are considering optimisation. It has also been identified that there is a desire to minimise machine tool downtime during calibration and to improve the machine's accuracy. From these observations, it has been established that the potential benefit from developing a method to automatically produce optimised machine tool calibration plans warrants further investigation.

Chapter 3

Planning Techniques to Aid Calibration

3.1 Process Planning

Process Planning techniques are commonly used in project planning as they focus on the selection and allocation of resources to achieve a desired goal. In the manufacturing environment, process planning typically deals with the construction of instructions to manufacture a part [32]. Computer-Aided Production Planning (CAPP) aims to improve the process planning and achieve more effective use of manufacturing resources [33]. In manufacturing, CAPP is used in Computer-Aided Design (CAD) and Computer-Aided Manufacturing (CAM) to help planning for design, manufacturing and assembly tasks. For example, planning the cutting paths for a machine tool. In CAD and CAM tools, CAPP is embedded as intelligent algorithms that are capable of performing planning tasks, whilst trying to find the optimum solution. One recent example presented by Chen et. al [34] is the application of CAPP for tool path generation of complex shoe moulds for numerically controlled machine tools. The approach provides a quicker and more robust way to generate NC tool paths than traditional approaches.

More recently, agent-based approaches have been applied to process planning and scheduling in manufacturing. An agent is a computer program that is designed to operate autonomously, perceive their environment, adapt to change, and create and peruse goals [35]. One such implementation of autonomous agent-based approach is the Integrated Process Planning and Scheduling (IPPS) system for machine job planning and scheduling [36]. The system implements a job and machine agent that represent their role in the manufacturing system respectively. In addition, there is an optimisation agent which aims to identify the optimum plan. Their system produces plans that have a shorter makespan (duration) that previous methods.

3.2 Automated Planning

Automated planning in engineering has been around since the nineteen nineties. Khoshnevis and Chen [37] developed a method of automated planning and scheduling while considering the assignment of resources. Their work presents a software tool that considers the assignment of resources as a multi-criteria decision-making problem. Their work was a successful demonstration of how automated process planning and scheduling can be implemented to enhance productivity.

Automated planning is particularly attractive as a solution for machine tool calibration because previous applications in engineering have shown that it can potentially provide a method of overcoming planning and resource allocation complexities, while finding the most efficient solution.

In this chapter, a review into techniques, languages and knowledge engineering tools is provided. This review will then allow for justified decisions to be taken regarding the techniques and tools used in this thesis.

3.2.1 Conceptual Model for Classical Planning

To explain the basic concepts of autonomous planning, a conceptual model is provided based on the state-transition system [7]. A state-transition system is a 3-tuple $\sum = (S, A, \rightarrow)$ where $S = (s_1, s_2, \dots)$ is a finite set of states, $A = (a_1, a_2, \dots)$ is a finite set of actions, and $\rightarrow: S \times A \rightarrow 2^s$ is a state-transition function.

A classical planning problem for a restricted state-transition system $\sum = (S, A, \rightarrow)$ is defined as a triple $P = (\sum, s_0, g)$, where s_0 is the initial state and g is the set of goal states. A solution P is a sequence of actions $(a_1, a_2, \dots, a_k)$ corresponding to a sequence

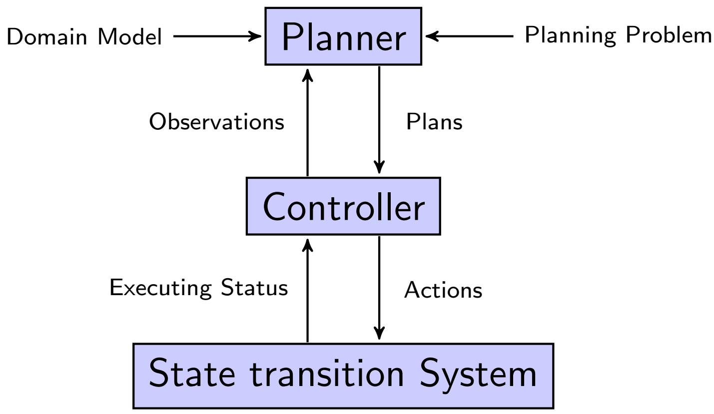


FIGURE 3.1: A conceptual model of AI planning

of state transitions $(s_1, s_2, \ldots, s_k)$ such that $s_1 \Longrightarrow (s_0, a_1), \ldots, s_k \Longrightarrow (s_{k-1}, a_k)$, and s_k is the goal state.

In AI planning, when planning for a complex problem, it can become practically impossible to represent explicitly the entire state space since the number of states can potentially increase exponentially. In classical planning, the state of the world is represented by a set of first-order predicates which are set true or false by an operator o (synonymous with action). An action has three elements: (1) a parameter list that is used for identifying the action, (2) a list of preconditions $precond(o)$ that must be satisfied before the action can be executed, and (3) an effect $effects(o)$ that contains a list of predicates that represent the resulting state from the execution of this action.

A full conceptual model for planning is shown in Figure 3.1 (Modified from [7]). The model has three parts: (1) a planner, (2) a controller, and (3) the state-transition system. The planner generates a plan (sequence of actions) for a specified problem model by using the domain model. A controller observes the current state of the system from the state-transition function and chooses an action that is generated by the planner based on the domain model. The state-transition system progresses according to the actions that it receives from the controller. The state-transition system is a form of "online" planning because it can also progress due to an unpredictable exogenous event. However, this thesis is considered with "offline" planning where nothing changes during planning and exogenous events are predetermined.

In this conceptual model of planning, the planner is kept logically separate from the domain model. This is called domain-independent planning, where the development of the planning tool is separate from the development of the domain model.

3.2.2 Deterministic and Non-deterministic Domain models

When the application of an action to a state can result in a single successor state, the state-transition system is deterministic [35]. In a deterministic model, every action will only result in one successor state for each applied action. In a non-deterministic model, the application of an action can result in multiple possible successor states [35]. In a non-deterministic model, applying an action in the state-transition system would result in a list of possible successor states, where the list is always greater than one.

3.3 Searching for Solutions

In automated planning, searching for a solution is a fundamental part of problem solving. Automated planning tools implement search algorithms to find sequence of actions to solve a planning problem. Either uninformed or informed search algorithms traverse through the state space to find a solution, π, or a set of solutions, Π [35]. Informed search algorithms traverse through the state space making informed decisions regarding some information or heuristic estimation function. To search through the state space, the search algorithm generates a search tree in the form of a graph. The current state, s, will be expanded by applying actions that result in the generation of a successor states. The set of immediate successor states can be referred to as the search space fringe. In the remainder of this section, different search algorithms are discussed to provide the background of state space search.

3.3.1 Uninformed Search Strategies

Two prominent uninformed search strategies in computer science are breadth- and depth-first search [35]. Breadth-first search iteratively expands all the nodes at the current fringe before expanding to the next level until there are no more states left to expand or the goal state is reached. Using Big O notation [35], if d is the depth of the search tree and b is the branching factor of the search tree, then the required storage is $O(b^d)$.

Depth-first search iteratively deepens the search tree by selecting the first node in the fringe and expanding it to the next level until either the solution is found or there is no successor to the current state in the fringe. The algorithm will then backtrack one level in the search tree and expend any alternative successor nodes. The space and time complexity for depth-first search is $O(b^d)$.

3.3.2 Informed Search Strategies

An informed search strategy applies specific knowledge other than the definition of the planning problem to guide the expansion of the search tree using a heuristic function ($h(n)$). The heuristic function will return the cheapest path from the state at node n to a goal state [35]. This heuristic function can form a "greedy" search because it attempts to expand the nodes that are closest the goal on the basis that it will result in a solution quickly. There is a wide range of available informed search strategies. However, there is a lot of research being performed to develop ones that are more efficient. A* [38] is a kind of best-first search that avoids expanding nodes that appear expensive by the evaluation function.

The branch-and-bound optimisation algorithm is another informed search strategy [7]. During search, an upper bound, λ, representing the cost of the optimal solution found so far is maintained. When a new node, u, is visited, a lower bound heuristic function $l(u)$ is used to calculate the cost of the plan currently being explored. If $l(u) > \lambda$, then the algorithm prunes node u in the knowledge that it will not result in a solution with a lower cost than λ.

3.4 Beyond Classical Planning

In many planning problems, the path (sequence of actions) to achieve the goal is irrelevant. For example, using automated planning for a factory-floor layout. This involves deciding what machine goes where based on some optimisation function. Using classical planning to plan for this problem would produce a sequence of actions to reach the goal. However, in this particular problem only the final configuration is required, not the order in which they are added. In this section, a different class of algorithms are discussed that are not concerned with node paths.

3.4.1 Local Search

Local search algorithms operate by keeping the search in the neighbourhood of the current state of the world and move to a neighbouring state that looks promising. Unlike classical planning algorithms, local search is not systematic. This introduces two new attributes: (1) low memory consumption, and (2) the ability to find solutions in very large (infinite) search spaces where systematic algorithms are not suitable. Additionally, local search algorithms are useful for solving optimisation problems. The hill-climbing algorithm [35] is an example of local search. A fundamental to hill-climbing is that it always moves in the direction of increasing value and does not maintain a search tree. However, hill-climbing can find a non-optimal solution. For example, because hill-climbing has no backtracking technique, it will get stuck at local maxima. This is where the algorithm is drawn up to a peak, but then cannot find a way to retrace and find the global maximum. Backtracking is a technique used in search algorithms which incrementally builds candidates to the solution, abandoning each partial candidate as soon as it has been determined that it can does not lead to a valid or optimal solution [35].

3.5 Planning with Time and Resources

Planning algorithms that have been discussed so far only have the implicit representation of time. The sequence of states and actions are instantaneous state transitions where the planning goal may be constrained by time but contain no implicit representation of time. These algorithms are useful for studying the computational aspects of planning. However, for many real-world applications they are not sufficient. In many real-world applications of planning, actions will occur over a time span. This introduces the complexity that actions should no longer have just preconditions and effects, they should also have conditions that prevail while the action is taking place. In this section, fundamental methods of planning with time and resources that are pertinent to this thesis are presented.

3.5.1 Representation of Time

Temporal planning involves reasoning with temporal references which are entailed by actions, events and time periods during which propositions hold. Typically, a temporal planner will reason about time using a temporal database that maintains temporal references for every domain proposition that varies in time. Temporal references are time periods during which a proposition holds or time points at which a state variable changes and are represented as an instant or interval. An instant is a variable ranging over the set $\mathbb{R}$ of real numbers which represents a point in time. An interval i is a pair (t_1, t_2) of instants, such that $t_1 \leq t_2$ [39, 40]. For example, consider planning for machining a feature on a CNC machine. t_1 would be the instant at which the machine starts machining the feature, t_2 would be the instant when the CNC machine has finished machining the feature, therefore, i_1 is the interval (t_1, t_2) corresponding to the machining the feature. The instants t_1, t_2 and interval i_1 are temporal references that specify when domain propositions are true.

Relation	Symbol	Inverse	Example
$before(i_1, i_2)$	$<$	$>$	
$equal(i_1, i_2)$	$=$	$=$	
$meets(i_1, i_2)$	m	m'	
$overlaps(i_1, i_2)$	o	o'	
$during(i_1, i_2)$	d	d'	
$starts(i_1, i_2)$	s	s'	
$finishes(i_1, i_2)$	f	f'	

TABLE 3.1: Thirteen possible relationships

Allen's interval-based temporal logic framework uses instants and intervals along with thirteen basic relations that can hold between two intervals [41]. Table 3.1 shows the thirteen possible relationships between two intervals. Disjunctions are allowed between the relationships for greater expressive power (i.e. $\{<\} \cup \{=\} = \{\leq\}$). Therefore, the ('in') predicate can be defined as:

$$in(t_1, t_2) \Leftrightarrow (during(t_1, t_2) \vee starts(t_1, t_2) \vee finishes(t_1, t_2))$$

where $\Leftrightarrow$ = equivalence and $\vee$ = logical disjunction (or).

Additionally, mutual exclusion between intervals can be expressed where no instant can occur at the same time. For examples, it is possible to ensure that instants t_1, t_2, t_3

are mutually exclusive by using the framework. This can be achieved by using the $before(i_1, i_2)$ relationship:

$$before(t_1, t_2) \wedge before(t_2, t_3) \Rightarrow before(t_1, t_3)$$

where $\wedge$ = logical conjunction (and) and $\Rightarrow$ = implies.

3.5.2 Concurrency, Coordination and Synchronisation

This section uses the notation proposed by Alan's in his work on internal-based logic [41]. Concurrency is a key aspect to temporal planning because it defines what can and cannot happen simultaneously. Actions can only happen simultaneously if they do not conflict with each other. For example, one action cannot delete a concurrent action's preconditions or effects. Two actions that are concurrent with each other have a relationship of $\{=, o, o', d, d', s, s', f, f'\}$ (Table 3.1). In this thesis, this notation represents all the possible temporal relationships between two intervals. For example, when measuring a machine tool many measurements can be executed concurrently but are not required to be executed concurrently to perform a valid calibration. An example is the measurement of linear positioning using a laser interferometer of the X-axis (E_{XX}), and providing there are no physical restrictions or interference, concurrently measuring the roll of the X-axis (E_{CX}) using an electronic level.

Coordination is where actions can, and sometimes must happen together and interact with each other. A good example is when lifting a bowl of liquid. When lifting, both sides must be lifted evenly to avoid spilling the liquid. Both the lift left and lift right actions interact to keep the bowl level. Coordinated actions have an interval relationship of $\{o, o', d, d'\}$.

Synchronisation is the same as coordination, however, precise timings are essential to the effects of actions. For example, in automated assembly lines it is essential that the joining of two parts and the arrival of the relevant fixing happen at exactly the right time to ensure the correct outcome. Synchronisation actions have an interval relationship of $\{=, s, s', f, f'\}$.

3.5.3 Temporal Problems

In temporal planning, two different classifications of temporal planning problems exist. These are Temporally Extended Actions (TEA) and Temporally Extended Goals (TEG). TEA is the extension of a classical planning problem (Section 3.2.1) with the extension of activities taking a duration to have their expected effects. An important aspect of TEA planning problems it that the goal and initial state are the same as in classical planning. TEG problems are no longer final states, they are trajectories through the state-space. For example, a goal might require a proposition to be true over a specified time interval or achieved by a fixed deadline. Temporally Extended Initial States (TEIS) is a further extension where predictable exogenous events can be expressed. For example, a predicate might become true during a predefined interval. An example of this is the predictable exogenous event of the workshop heating turning on and off during the day. Another example is the time of sunrise and sunset. Both these examples affect the environmental temperature, the temperature of the machine tool and any instrumentation.

3.5.4 Durative Actions

Durative actions provide the method to model temporal actions by associating durations with actions. A durative action is an extension of an action used in classical planning where the effects change the state-space. The term “blackbox” is adopted to describe durative actions that have preconditions and effects, but no means of defining what is happening during their execution. This results in a restrictive concurrency model where only actions that do not interfere at all can be executed together. “Blackbox” actions do not allow for coordination and do not support actions that make a fact true only during their execution.

A more expressive formulation of durative actions defines conditions to hold at the start, end and for the whole duration. These later are called invariants. This formulation also allows for facts to be true at the start and end of a duration. For example, consider the actions of driving a car from a start location to an end location. There should be a precondition that the car is at the start location. However, once the driver starts the engine, engages the gear, releases the brake and starts moving it will no longer be at the start location, so it should be removed using a start delete effect. While the car is driving to the end location, an invariant should be used to ensure the engine remains on and there should be an end effect to assert that the car and driver are at a new location once it stops. This allows for expressing concurrency and coordination since the state of the world is known during the execution of an action

3.5.5 Planning with Resources

Resources are an essential aspect of planning as they provide the means to model quantitative and qualitative change. Quantitative resources are associated with consumption and production, which can be discrete or continuous. They may be consumed by the passage of time and be exchanged with other resources. Fuel for an aircraft is a good example of a quantitative resource. Qualitative resources are represented by the state of an object, such as the availability of a machine.

Within the research community there is less agreement, when compared to the planning problem, as to exactly what the scheduling problem is. However, there is agreement as to what the class of scheduling problems entail [39]. Planning is the construction problem, identifying which actions should be used to reach a goal without breaking any logical constraints. Scheduling is an optimisation problem, deciding when actions should occur without breaking any temporal constrains. Scheduling can also be defined as the allocation of resources over time.

Scheduling takes the view that resources should be expressed explicitly and reasoned with directly. However, planning takes a different view of resources. In planning, resources are modelled implicitly and there is no distinction between an object acting as a resource or part of the planning problem. For example, a machine tool might be seen as a resource when machining a set of features, but could also be part of the goal if it is required to finish machining in a particular state.

Representing resources implicitly makes it difficult to do any specific reasoning with them and realise the use of alternative resources. However, by not representing the resources explicitly, the system has to discover them. If it can do this successfully, it is able to find resources that the domain designer did not realise.

3.6 Hierarchical Task Networks

Hierarchical Task Network (HTN) closely relates to classical planning in that each state of the world is represented by a set of atoms. Actions are deterministic and modify the state of the world. However, HTN planners differ from classical planners in how they plan and what they plan for [7]. In HTN planning, the aim is not to achieve a set of goals but is to perform a set of tasks. The input to an HTN planning system is a set of operators and a set of methods, each of which is a description of how the task can be decomposed in to a set of subtasks. As illustrated in Figure 3.2, HTN planning is performed by recursively decomposing non-primitive tasks into small subtasks, until primitive tasks are reached that can be performed directly without using a planning operator.

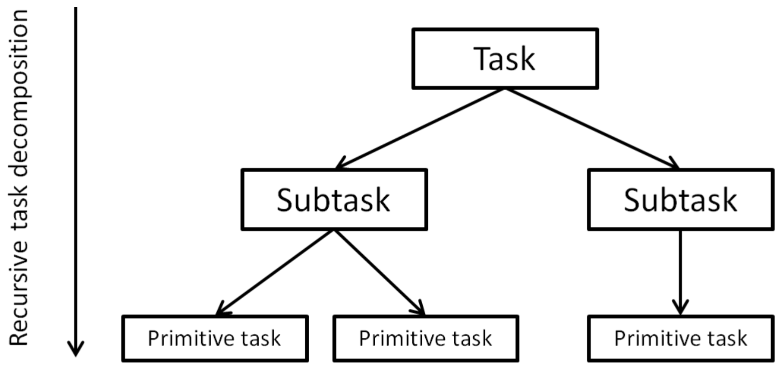


FIGURE 3.2: Illustration of a HTN system showing recursive task decomposition

HTN planning systems are more widely used for practical applications. The main reason behind this is they provide the means to write a problem-solving domains that closely mimic how a human expert would solve the problem. Within an engineering context, 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 [42]. The system processes the geometric features of a CAD model to determine the required machining operations. The authors have identified the complexities 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 CAPP based 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 [43]. The CAPP system is represented in HTN form by using the SHOP architecture [44]. The motivation behind the selection of an HTN is very similar to that of IMACS. It was found that traditional general purposes planners did not allow for the specification of the domain-specific knowledge.

3.7 Planning Domain Definition Language

The Planning Domain Definition Language (PDDL) was released in 1988 By Drew McDermott [10] and has since become widely adopted by researchers. PDDL is based around the Stanford Research Institute Problem Solver (STRIPS) [45] and Action Description Language (ADL) [46] for classical planning. The main differences between the notations of STRIPS and ADL are: (1) in ADL it is possible to have both positive and negative literals, whereas it is only possible to have positive literals in STRIPS, and (2) in STRIPS unmentioned literals are false (closed-world assumption), whereas in ADL all unknown literals are unknown (open-world assumption). The main motivating factor for this is that the languages were used in the first International Planning Competition (IPC) [9]. The initial version of PDDL was specified for early IPCs and had the level of expressibility required for classical planning.

A PDDL problem is comprised of two parts. Firstly, the domain that consists of predicates and operators, and secondly the problem definition, consisting of the initial and goal state. In this section, the evolution of the PDDL language to include expressibility for many real-world planning problems is discussed.

3.7.1 PDDL2.1, 2.2 and 3.0

PDDL2.1 is an extension of PDDL1.0 that includes a durative action model [47] and was the official language in the 3rd IPC [48]. Temporal aspects are expressed through action durations where conditions and effects are specified to hold either at the start or at end of the actions. Invariants are conditions that can hold throughout the entire action's duration.

PDDL2.1 introduces the notation of numeric variables (fluents) to represent non-binary resources such as time. These fluents become part of the state representation as well as propositions. Fluents can be used in both preconditions and effects. The effects use operators `(scale up, scale down, increase, decrease and assign)` to modify the value of the fluent by the binary functions `(+, -, /,*)`. Comparisons between fluents is performed by using comparators $(\leq,<,=,>,\geq)$ between functions of fluents and real numbers.

For the purpose of the planning competition, PDDL2.1 was split into five levels [47], with the fifth representing the full PDDL semantics. The following list describes the four levels where each level extends the previous:

1. **Level 1** Original PDDL with STRIPS and ADL.
2. **Level 2** Addition of numeric fluents and the ability to test and modify their values instantaneously.
3. **Level 3** Actions can represent time (Durative actions).
4. **Level 4** Effects happening during the execution of an action (Continuous effects) where a numeric fluent is modified by some function of time since starting the action.

The language was then further extended to PDDL2.2 [49] for the IPC held in 2004 [50] to include two new features; (1) Derived Predicates and (2) Times Initial Literals (TIL). Derived predicates account for the possibility of a proposition becoming true or false based on other propositions. These are implemented based on "if then" rules. For example it is now possibly to express, "If Fred is employed as an electrician, and all electricians are working on a site in Huddersfield, then Fred is working in Huddersfield".

Timed Initial Literals (TILs) cater for the specification of predictable exogenous events in the initial state. For example, it is possible to state that a factory will open at 08:00 and close at 19:00. TILs are outside the control of the planner, but are predictable and known in advance.

PDDL2.2 problems can be compiled down to PDDL2.1 problems [51]. However, TILs are a polynomial compilation, whereas derived predicates can potentially lead to an exponential growth in the number of actions required. The method of compiling down TILs involved a process called clipping actions. Given the predefined times $t_1, \ldots, t_n$ when a predicate $p_1, \ldots, .p_n$ will change, a collection of durative actions, $d_1, \ldots, d_n$ are created that will occur for the durations $t_1, t_2 - t_1, \ldots, t_n - t_{n-1}$.

PDDL2 has a TEA implementation of temporal planning problems. However, just as it is possible to model TILs in PDDL2.1, it is also possible to model other TEG features, such as temporal constraints and deadlines [51].

In PDDL2, durative actions increase the complexity of the planning problem because they can be of four different forms [47]:

1. **Fixed** The duration of the action is fixed and is the same for all instantiations.
2. **Statically Computed** The duration of an action is dependent on the described parameters and not the state of the world.
3. **State Dependent** The duration of an action will change dependent on the state of the world.
4. **Variable** The duration of the action is dependant on how long the action is executed.

Further extensions include PDDL3.0 which introduced state-trajectory constraints and preferences [52]. This extension implemented the preference based notation of "hard goals" that must be achieved in a valid plan, as well as "soft goals" that are desirable to achieve, but do not necessarily have to be achieved. When planning with the notation of hard and soft goals, the planner should satisfy, in addition to hard constraints, as many soft constraints as possible. PDDL3.1 is the most recent extension to include object-fluents where a fluent can not only be numerical, it could also be an object-type [53].

3.7.1.1 PDDL+

PDDL+ is the extension of PDDL2 that caters for modelling continuous processes and events [54, 55]. The key to this extension is the ability to model the interactions between agent's behaviour and change that are initiated by the agent's environment. Processes run over time and have a continuous effect on numeric values. They are initialised or terminated either by a direction action of the agent or by an event triggered in the environment. This three-part structure is referred to as the start-process-stop model. In PDDL+ it is possible to plan with continuous, non-linear effects. One example, is the automatic construction of battery usage policies using PDDL+ [56, 57] where the continuous, non-linear dynamics of battery usage is modelled in PDDL+, and battery usage policies are the process that governs the discharge and recharge of a battery. The example given in the paper is policies for laptop battery management.

3.7.2 System Definition

Using PDDL requires the use of two components: (1) a description of the domain, and (2) a description of the problem. The domain description contains a description of the objects, predicates and actions. Actions provide the means of changing the state of the world by applying effects when a set of preconditions are satisfied. A PDDL planning problem can be formally defined as a 4-tuple $P = (s, a, i, g)$ where s is the set of all possible predicates and objects, a is the set of all possible actions that can be applied, i is the set of instantiated predicates describing the initial state, and g is the set of instantiated predicates describing the goal. The produced plan is a sequence of instantiated actions that achieve the goal from the initial state.

3.7.3 HTN and Goal Achievement Planning

As described in this section, PDDL is a family of languages which are used to encode domain knowledge. The domain model along with an initial and goal state are then interpreted by a planning tool, where it applies actions to change the current state in order to achieve a goal. In comparison, a HTN is an approach to reduce a problem by creating a task network which encodes knowledge of how to decompose tasks, thus making them easier to solve. This is because heuristic information of how the problem should be solved is encoded into the decomposition tree, whereas in PDDL less information regarding the ordering of actions is encoded. However, HTN systems such as SHOP2 [44] can take as input PDDL domains and convert them to HTN models. Likewise, methods exist to convert from HTN to PDDL domains [58]

3.8 State-of-the-Art Planning Systems

State-of-the-art in domain-independent automated planning tools can be observed from the results of the IPCs, as well as publications regarding the planner's implementation and performance. Given the nature of the planning problem presented in this thesis (temporal and numeric), planners that meet the minimum requirement of PDDL2.2 allowing durative actions, numerics, and exogenous events will be discussed.

3.8.1 Planning with Predictable Exogenous Events

Local Search for Planning Graphs with TILs and derived predicates (LPG-td) [59] is a domain-independent planning tool and was a top performer in the third International Planning Competition (IPC) [60], solving 428 planning problems with a success of 87%. Additionally LPG-td was a top performer involving domains with predictable exogenous events (TILs) [61]. LPG-td implements an extended local search algorithm (Section 3.4.1) and action graph representation. This representation is a Numerical Action (NA) graph which extends the action graph [62] to contain propositional nodes and numerical nodes, labelled with propositions and numerical expressions, respectively [63]. Since the production of LPG-td, many other planners have been developed that can solve PDDL2.2 problems and beyond.

CRIKEY [64] is a PDDL2.1 planner developed in JAVA. CRIKEY requires compilation of TILs to support exogenous events (Section 3.7.1). The CRIKEY system was developed to improve co-ordination between planning and scheduling. CRIKEY was a competitive planner, competing in the fourth IPC [49]. The performance of CRIKEY was not exceptional in terms of plan generation speed and plan quality. Hansley [39] explains that this is because CRIKEY splits planning and scheduling to try and find a better solution in terms of scheduling, however, in reality this has reduced the performance of the system.

Linear Programming alongside Relaxed Planning Graph (LPRPG) [65] is a planner designed for use with domains that have numeric resource flows. LPRPG uses a hybrid heuristic comprising the propositional structure of the relaxed planning graph, based on the heuristic present in Metric-Fast Forward (Metric-FF) [66], with Linear Programming (LP) to enhance numeric reasoning. A linear programming is an optimisation method to achieve the best outcome in a mathematical model where requirements are represented by linear relationships. LPRPG was developed with the motivation in-mind to develop planners that can solve more complex, real-world problems. LPRPG-P [67] is an extension to LPRPG to add support for PDDL3.0 preferences. PDDL3.0 preferences are soft-constraints that are not required to be satisfied, but their effect can be incorporated into the plan metric [68]. Metric-FF is a non-durative planner that supports PDDL2.1 to level 2, and is therefore, not able to plan whilst considering time.

3.8.2 Linear Continuous Numeric Effects

COLIN [69] is a planner capable of planning with COntinuous LINear numeric change through the use of linear programming. COLIN in loosely based around CRIKEY, albeit it implemented in a different language (C++). However, COLIN does not support PDDL+ process and events, instead it is limited to continuous change as expressed through the durative action [70].

An extension of COLIN that uses Partial Order Planning using Forward-chaining POPF [71] to handle domains with linear numeric effects. The difference being that POPF incorporated ideas from partial-order planning. This implementation seeks to only introduce ordering constraints needed to resolve threats rather than insisting the new action occurs after all those already in the plan.

Optimising Preferences and Time-Dependent Costs (OPTIC) is a temporal planner that is an extension of POPF that implements the semantics of PDDL3.0 [72]. OPTIC was developed for use in problems where plan cost is determined by preferences of time-dependent goal-collection costs, such as scheduling the delivery of perishable goods. OPTIC has recently been applied to the problem of automated planning for liner shipping fleet repositioning [73].

The Temporal Metric - Linear Programming Satisfiability planner (TM-LPSAT) [74] uses satisfiability solving techniques and is an evolution of LPSAT [75]. TM-LPSAT is able to solve resource planning problems with real values, as well as PDDL+ features but is restricted to linear domains. Versatile Heuristic Partial Order Planner (VHPOP) [76] is a partial order causal link planner that has support for durative planner actions. The main limitation of VHPOP is that it is not able to solve planning problems with numeric preconditions and effects.

3.8.3 Non-Linear Continuous Numeric Effects

Recently, to improve the quality of modelling real-world applications, there has been a shift towards the study of continuous, non-linear effects. However, due to the complexity of the dynamics of such models, they are restricted to solving small scale problem instances or problems with domain-specific heuristics, such as the automatic construction of battery usage policies using the Universal Planner Murphi (UPMurphi) [56] planner.

UPMurphi [77] provides a "discretize and validate" approach to continuous planning and supports the full PDDL+ semantics. Although, this planner is both powerful and novel in its approach, it performs an exhaustive breadth-first search. This results in an exponential increase in the produced search space [78]. This restricts the use of the planner to solve real-world problems by implementing a strong, domain-specific heuristic function [56]. This is a useful tool, however the loss of domain independence departs from the aim of the planning community.

A version of the Simple Hierarchical Ordered Planner 2 has been developed that also supports the full systematics of PDDL+ ($SHOP2_{PDDL+}$) [79]. It is different in its approach because it contains a modified the SHOP2 algorithm. Even though the published work suggests that this planner could out-perform the rest in terms of plan generation time, it is difficult to evaluate the scalability of the planner because it is currently not in the public domain.

3.9 Encoding Domain Knowledge

Knowledge Engineering (KE) for automated planning is the process that deals with acquisition, formulation, validation and maintenance of planning knowledge, where the key product is the domain model. In recent years, knowledge engineering tools for domain-independent planning have progressed, helped by a series of competitions. Domain engineers will typically either develop domain models using (1) a traditional text editor, or (2) a Graphical User Interface. Traditionally, all domain models had to be developed in a text editor (e.g. Notepad), but recent improvements in Graphical User Interface (GUI) knowledge engineering tools are helping to make knowledge engineering a more efficient process.

3.9.1 State-of-the-art Domain Engineering Tools

The Extensible Universal Remote Operations Planning Architecture (EUROPA) [80] is an integrated platform for AI planning and scheduling, constraint programming and optimisation. The main goal of the application is to deal with complex real-world problems and has been used in various NASA missions. EUROPA provides modelling support, result visualisation and an interactive planning process. Europa uses the New Domain Definition Language (NDDL) [81, 82]. NDDL is different from PDDL in that it uses a timeline and activity representation, rather than the propositional representation in PDDL. NDDL is also different in that there is no concept of states or actions, only intervals (activities) and the constraints between them.

The Graphical Interface for Planning with Objects (GIPO) [83] is based on its own Object-Centred Languages OCL and OCL_h for hierarchical domains. GIPO also provides a method functionality to support interactive modelling.

itSIMPLE [84] provides an environment that enables knowledge engineers to model a planning domain using the Unified Modelling Language (UML) standard [85]. itSIMPLE focuses on the initial phases of a disciplined design cycle, facilitating the transition of requirements to formal specifications. Requirements are gathered and modelled using UML to specify, visualise, modify, construct and document domains in an object-oriented approach. A second representation is automatically generated from the UML model, and it is used to analyse dynamic aspects of the requirements such as deadlocks and invariants. Finally, a third representation in PDDL is generated in order to input the planning domain model and instance into an automated planner.

JABBAH [86] is an integrated domain-dependent tool that aims to develop process transformation to be represented in a corresponding HTN planning domain model. The system mainly deals with business processes and workflows. The processes are represented as Gantt charts or by using an open source workflow engine.

Mashup Automation with Runtime Invocation and Orchestration (MARIO) [87] is an integrated framework for composing workflow for multiple platforms, such as Web Services and Enterprise Service Bus. This tool provides a tag-based knowledge representation language for composition of planning problems and goals. It also provides a web-based GUI for AI planning system so that the user can provide software composition goals, views and generated flow with parameter to deploy them into other platforms.

PDDL Studio [88] is a recent PDDL editor that allows the user to write and edit PDDL domain and problem files. The main goal of the tool is to provide knowledge engineers the functionality to edit and inspect PDDL code, regardless of how the PDDL code was created. The tool supports the user by identifying syntactic errors, highlighting PDDL components and integrating planners. PDDL Studio does not require the user to draw any diagram, it is more like writing traditional programming language code by using an Integrated Development Environment (IDE). The current version of this tool can help editing basic PDDL and also provides error checking.

VIZ [89] is a knowledge engineering tool inspired by GIPO and itSIMPLE. It shares many characteristics of those systems (GIPO and itSIMPLE) with the addition of a simple, user friendly GUI by allowing inexperienced knowledge engineers to produce PDDL domain models. This tool uses an intuitive design process that makes use of transparent diagrams to produce a PDDL domain model. The tool does not support any third party planner integration. However, the tool is still being developed.

3.9.2 Limitation of State-of-the-art

A main issue of current KE approaches for encoding domain models is that they require specific expertise. Tools such as PDDL Studio require a PDDL expert, itSIMPLE requires some expertise in UML language, which is common in software engineering. GIPO requires some expertise in the OCL_h language, which is not a widely known language in the AI Planning community. This requirement might significantly reduce the number of potential users of the KE tools and slow their development. Since users with different research backgrounds usually do not have the required expertise, they are not able to exploit existing approaches for encoding domain models. They require an expert that, due to his limited knowledge of the real world domain, will introduce some noise in the encoding. Moreover, given the difficulty of generating domain models for planning, many users are not exploiting automated planning but use simpler approaches, even if they are less efficient. It is also worth considering that KE tools for encoding domain models are, usually, not very well known outside the planning community. This, again, reduces the number of potential users that could exploit them.

Current KE tools are designed for a single user. This is usually acceptable because the majority of the generated domain models are simplified encoding that only require one or few editors. KE tools and domain-independent planners are currently developing rapidly, and in the future it is likely that knowledge engineering of complex real-world applications will require engineering collaboration.

Users are not supported by existing KE tools in writing documentation related to the generated model. As a result, users do not usually maintain proper documentation. Given this, it is often quite difficult to change an existing domain model even only few months after its generation. Providing support for writing documentation would make changes easier and would also help the users while encoding the model. The process of describing what has been done is a first test for the model. Furthermore, some tools are not able to handle domain models that have been changed manually, or by using a different tool. This limits the support that such tools could give to the life cycle of domain models.

EUROPA provides an extensive range of graphical KE tools that significantly enhance the process of knowledge engineering. The tool overcomes many of the issues regarding KE tools that produce PDDL models. For example, the tool has been designed and created embedding functionality to facilitate collaboration, revision and documentation. The main limitation of EUROPA and NDDL is that it is proprietary technology produced by NASA and does not follow the principles of the automated planning community. Whereas, producing a model in PDDL can be used with any PDDL supporting planning tool within the community, NDDL models can only be used with EUROPA. Therefore, any advances in automated planning can only be exploited by a NDDL model once they are implemented in EUROPA.

Finally, existing KE tools for generating domain models for planning have a very limited support of the features of PDDL language. Most of them only support PDDL, while a few of them are also able to handle some structures of PDDL2.1 [47]. It is noticeable that the latest versions of PDDL have features (e.g. durative actions, actions costs, etc.)

that are fundamental for a correct encoding of real world domains. Furthermore, none of the existing tools support PDDL+ [55]. PDDL+ provides features for dealing with continuous planning, which is needed in systems working in real-time and that must be able to react to unexpected events.

3.10 Chapter Summary

To summarise, in this chapter a conceptual model of automated planning (classical) has been provided and discussed. In addition, a brief description of of the algorithms embedded in automated planning tools is provided. This is then expanded detailing searching with time and resources that is essential for modelling real-world problems. This leads to the description of Hierarchical Task Networks and their advantage for applying automated planning to processes that can easily be decomposed.

The development of PDDL is then discussed, showing the features and how they allow the use of automated planning technologies. An investigation into state-of-the-art in domain independent planning tools is the presented.

Finally, a survey of knowledge engineering tools is performed, discussing the advantages and disadvantages of available tools. This survey suggests that knowledge engineering tools are not currently at a sufficient level to model complex, real-world problems with strong temporal and numeric aspects.

Based on the information presented in this section, the following informed decisions can be made regarding the technology and techniques that are to be used in this project; The advantages of developing a domain model using PDDL make it a asset to the project. The expressiveness of PDDL allows for both temporal and numeric properties of a planning domain to be encoded and solved using the current state-of-the-art planning tools. Using PDDL will also allow any developed models to benefit from advancements in the state-of-the-art.

As identified in Section 3.9, there are many different KE tools that aim to make domain engineering more maintainable and reliable. However, because the tools are in their infancy and are yet to support features such as revision control and importing PDDL domains, making changes that are outside the tool's performance will result in a loss of compatibility. For these reasons, developing the domain model manually using a text editor will allow for better control and iterative development. However, this approach will require careful design to minimise costly model implementation and debugging times.

Planner	Encoding	PDDL Version
LPD-td	ADL	2.2
CRIKEY	STRIPS	2.1 level 3
LPRPG	STRIPS	2.1 level 1, 2 and 3
METRIC-FF	ADL	2.1 level 2
MIPS-XXL	ADL	2.1 level 1, 2 and 3
COLIN	STRIPS	2.2
POPF	STRIPS	2.2
OPTIC	STRIPS	2.2
TM-LPSAT	ADL	PDDL+
VHPOP	STRIPS	PDDL+ level 1 and 3
UPMurphi	ADL	PDDL+
SHOP2$_{PDDL+}$	ADL	PDDL+

TABLE 3.2: Comparison of current state-of-the-art planners

In Section 3.8 the current state-of-the-art in domain-independent planners is discussed. As shown in Table 3.2, even though each planner is designed to be domain-independent, there are few that support the full PDDL2.2 semantics and only LPG-td can handle PDDL2.2, ADL and numeric pre-conditions and effects. It is important to develop the domain model using a planner that can support as much of the PDDL2.2 language as possible. This will place fewer restrictions on the domain engineering process and help to improve the quality of the produced domain. For this reason, the LPG-td planner will be used for the development of the domain model in this thesis. Although, other domain-independent planners will be used for experimental analysis where possible.

Chapter 4

Temporal Optimisation

This chapter examines temporal aspects of calibration planning, discussing how individual measurement tasks interact. Through examining the temporal construct of measurements, a model is developed that can produce temporally optimal calibration plans when using LPG-td.

An initial feasibility investigation is performed to examine the potential of using automated planning and scheduling as a potential solution. Following this, state-of-the-art tools in automated planning and scheduling are used to solve a variety of different calibration instances. Comparison between automatically constructed calibration plans can then be drawn with those from industrial and academic experts. This empirical data can be used to validate the model's fitness for purpose.

4.1 Parametrisation

The process of machine tool calibration requires the consideration of many individual parameters. Figure 4.1 shows a breakdown of the individual parameters associated with the instrument (Section 2.2.2), measurement (Section 2.2.1) and machine tool (Section 2.1).

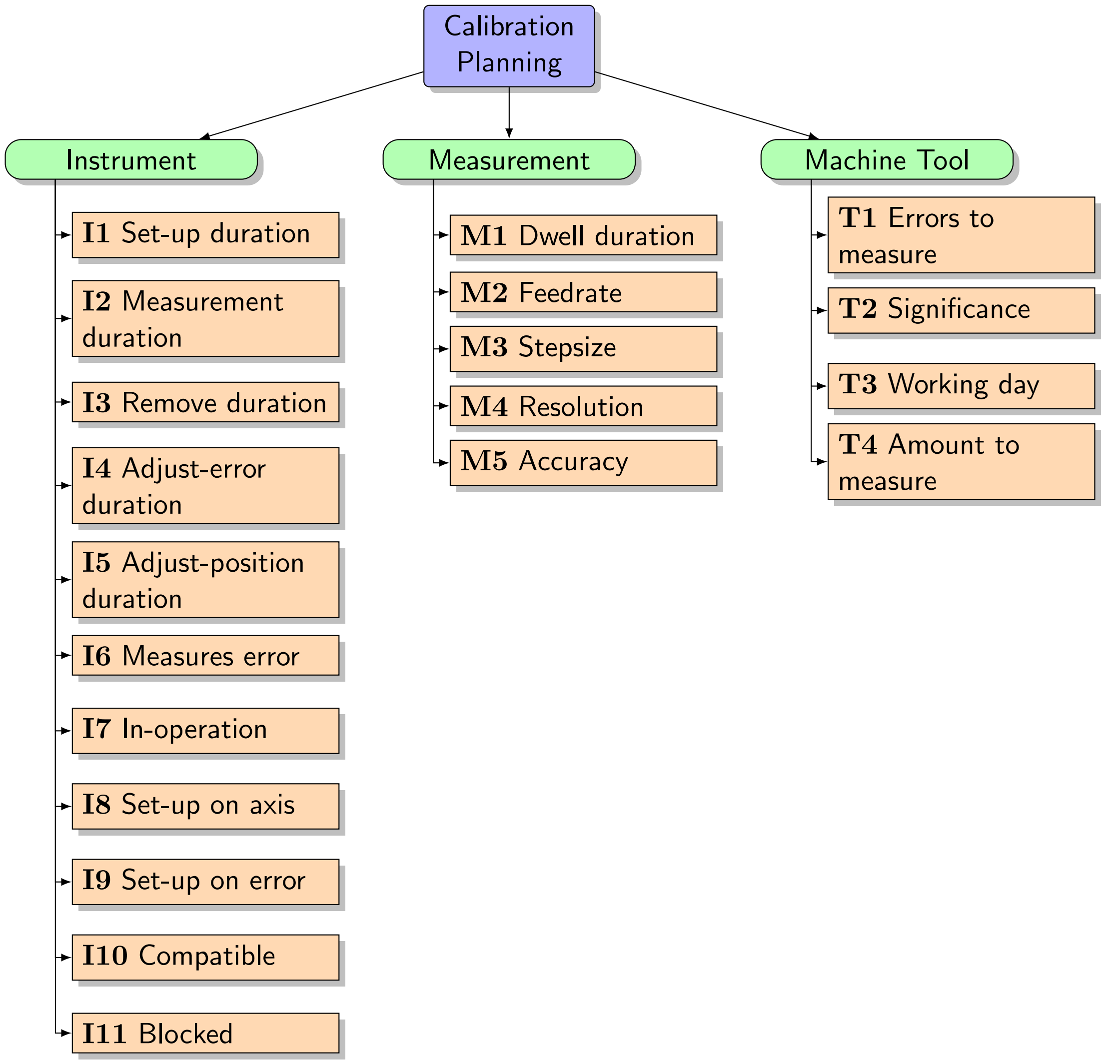


FIGURE 4.1: Calibration parameters

4.1.1 Instrumentation Parameters

In Figure 4.1, the individual parameters associated with the instrumentation are shown. The first five denote the duration required for setting-up the instrumentation (**I1**), performing the measurement (**I2**), removing the instrumentation (**I3**) or adjusting the instrumentation to for another error (**I4**) or to a different position (**I5**). The remaining six parameters are predicates used to express the state of the instrumentation throughout the measurement.

I6 Specifies whether the instrumentation can measure a specific error.

I7 Indicates whether the instrumentation is currently in operation.

I8 The instrumentation is set-up on a specified axis.

I9 The instrumentation is set-up to measure a specified error component.

I10 The instrumentation is compatible with other specified instrumentation.

I11 The instrumentation is blocked from use.

4.1.2 Measurement Parameters

Also displayed in Figure 4.1 are the measurement parameters. These are the measurement specified parameters within the model and are described in the following list:

M1 The duration that the machine will be stationary, allowing the measurement to take place.

M2 The velocity that the machine is required to move between the targets (feedrate).

M3 This is the distance between any two targets (stepsize).

M4 The measurement will require the instrumentation to be able to measure at a specified resolution.

M5 The measurement will also require the instrumentation to be able to measure to a specified accuracy.

4.1.3 Machine Tool Parameters

Finally, in Figure 4.1 the parameters related to the machine tool are discussed.

T1 Defines the errors that are to be measured.

T2 States the significance of individual errors on the manufacturing process.

T3 Defines the hours of the machine tool's working day.

T4 Specifies the proportion of the axis travel that requires measuring for each error.

4.1.4 Optimisation Criteria

As described in Section 3.3.2, using a heuristic when searching can help to find an optimal solution quicker. For reducing the duration of a machine tool calibration, the heuristic function is required to take into consideration the following ways to reduce the overall calibration plan's duration.

4.1.4.1 Smallest Accumulative Duration

This minimisation function is to return the most efficient selection of measurements where the objective is to reduce estimated time. Each measurement task comprises of several sub-tasks that have an associated duration.

$$f(mt) = min(\sum_{i=1}^{n} m(\sum_{i=1}^{n} d)) \tag{4.1}$$

Equation 4.1 shows an abstract minimisation function, $f(e)$, for measuring the machine tool mt, where m are individual measurements (error component) and is made up of the sum of durations, d. For example, the duration to setup a measurement and the duration to perform the measurement. min is the combination of d for measurement m where the accumulation of all the durations is as low lowest possible.

4.1.4.2 Concurrent Measurements

To reduce the overall temporal span for calibration, aspects of measurements should be performed concurrently where possible. This means that any number of task (measurement) intervals ($i_1 = [t_1, t_2]$) would have the temporal relationship of $\{=, o, o', d, d', s, s', f, f'\}$ if they do not interact with each other (notation described in Section 3.5.1, Table 3.1). An example of two measurement tasks that could be concurrent are measuring a horizontal linear axis' angular roll deviation using an electronic level while a laser interferometer has been switched on and is stabilising.

It is also likely during measurement that interacting tasks can happen concurrently, however they will have a coordination $\{o, o', d, d'\}$ relationship. A different variation of the previous example would require coordination. For example, if measuring both positional deviation using a laser interferometer and angular deviation using an electronic level are scheduled together, their interaction must be considered. This interaction requires that each parameter $p_1, p_2, \ldots, p_n$ for each coordinated task $t_1, t_2, \ldots, t_n$ is compatible (**I10**) $t_1(p_1) = t_2(p_1), t_1(p_2) = t_2(p_2), \ldots, t_n(p_n) = t_{n+1}(p_n)$. Following the previous example, the two measurements (laser interferometry and electronic level) can only be coordinated together if the feedrate (**M2**), dwell time (**M1**) and step size (**M3**) are compatible.

4.1.4.3 Instrument Adjustment

Another identified method of temporal reduction is by careful consideration of instrument set-up and adjustment. In some instances multiple error components can be measured using the same instrumentation, with small adjustments to the configuration and set-up. For example, using a laser interferometer, it is possible to measure many positional, angular and straightness deviations by changing the optics and realigning the laser beam. Once the device has been stabilised for one measurement no lengthy initialisation procedure will need to be performed for subsequent tasks with using the same device. In addition, if the measurement is taking place on the same axis, it is possible to use the same machine part-program. This decision can be made by comparing the estimated time to adjust $a(e)$ to measure an error e against the estimated time to set-up $s(e)$ to measure an error, $a(e) < s(e)$.

4.2 An Initial Hierarchical Task Network Solution

In the planning community, a well-established guideline is that the connection between the HTN paradigm of task decomposition can aid encoding domain knowledge [7]. Applying this paradigm to machine tool calibration will allow for the production of an HTN domain that represents task decomposition for machine tool calibration. The model developed for this initial HTN solution uses a relaxed, core set of parameters instead of the extensive list as seen in Figure 4.1.

4.2.1 Task Decomposition

Figure 4.2 shows machine tool calibration as an abstract task decomposition tree. From the figure, it is evident that an abstract version of machine tool calibration can be represented using a small number of tasks.

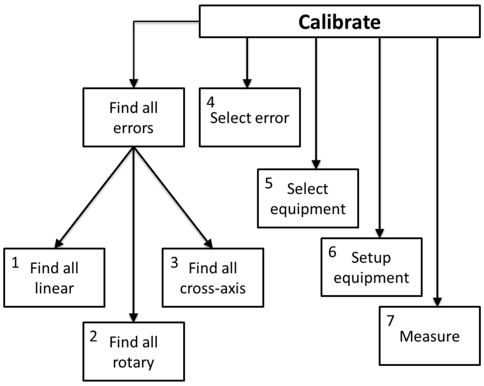


FIGURE 4.2: Machine tool calibration task decomposition tree

4.2.2 System Definition

An HTN planning problem is a 4-tuple $\sum = (s_0, w, o, m)$ where s_0 is the initial state, w is the initial task network, o is a set of operators, and m is the set of methods which perform the task decomposition based on a logical precondition.

The initial state s_0 consists of a set of first-order predicates. The six predicates shown in Table 4.1 provide the means for describing the basic kinematic chain of the machine tool, measurement instrumentation and measurement method. The variables used in each predicate are prefixed by a question mark.

Parameter	HTN Predicate
T1	`Axis(?a)`
T1	`Linear(?a)`
T1	`Geometric Error(?a, ?e, ?significance)`
I6	`Instrument(?i, ?cost`$_s$`, ?cost`$_a$`)`
I6	`Method(?i, ?m)`
I6	`Measures(?e, ?i, ?cost`$_m$`)`

TABLE 4.1: Calibration parameter to HTN predicate mapping

Where $?a$ is an axis object, $?e$ is an error object, $?significance$ is an assigned significance weight for the geometric error $?e$, $?i$ is the instrument, $?cost_s$ is the cost of setting up

Parameter	HTN Operators
T1	`(select error ?a ?e ?i)`
I6 & I7	`(select equipment ?a ?e )`
I1 & I8	(set-up equipment ?a ?e ?cost$_s$)
I4	(adjust equipment ?a ?e ?cost$_a$?e$_{-1}$)
I2 & I6	(measure ?a ?e ?cost$_m$)
n/a	`(assert ?g)`
n/a	`(remove ?g)`

TABLE 4.2: Calibration parameter to HTN operator mapping

Parameter	HTN Methods
T1	`(:method perform calibration)`
T1	`(:method find all required)`
I6	`(:method calibrate)`
I8, I9, I10 & I11	`(:method select equipment)`
I2 & I6	`(:method setup equipment)`
n/a	`(:method measure error)`
I3, I8, I9, I10 & I11	`(:method remove previous)`

TABLE 4.3: Calibration parameter to HTN method mapping

the instrument, $?cost_a$ is the cost of adjusting the instrument, $?m$ is the measurement method, and $?cost_m$ is the cost of performing the measurement.

The initial network w consists of the single high level task of $(perform - calibration)$. o is the set of 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 is the head, P is the precondition, D is the delete list, A is the add list, and c is the optional cost. Table 4.2 provides the seven operators that are used in the model.

Where $?e_{-1}$ is the previous error and $(assert\ ?g)$ and $(remove\ ?g)$ are two house-keeping tasks for listing tasks still to be executed.

The set of methods $?m$ which describe how non-primitive tasks can be decomposed.

$$\texttt{(:method h}[n_1]\ \texttt{C}_1\ \texttt{T}_1\ [\texttt{n}_2]\ \texttt{C}_2\ \texttt{T}_2\ \ldots[\texttt{n}_n]\ \texttt{C}_n\ \texttt{T}_n\texttt{)}$$

Where h is the head, n_i is the name for each succeeding C_i T_i pair, C_i is the precondition and T_i is the task list (tail).

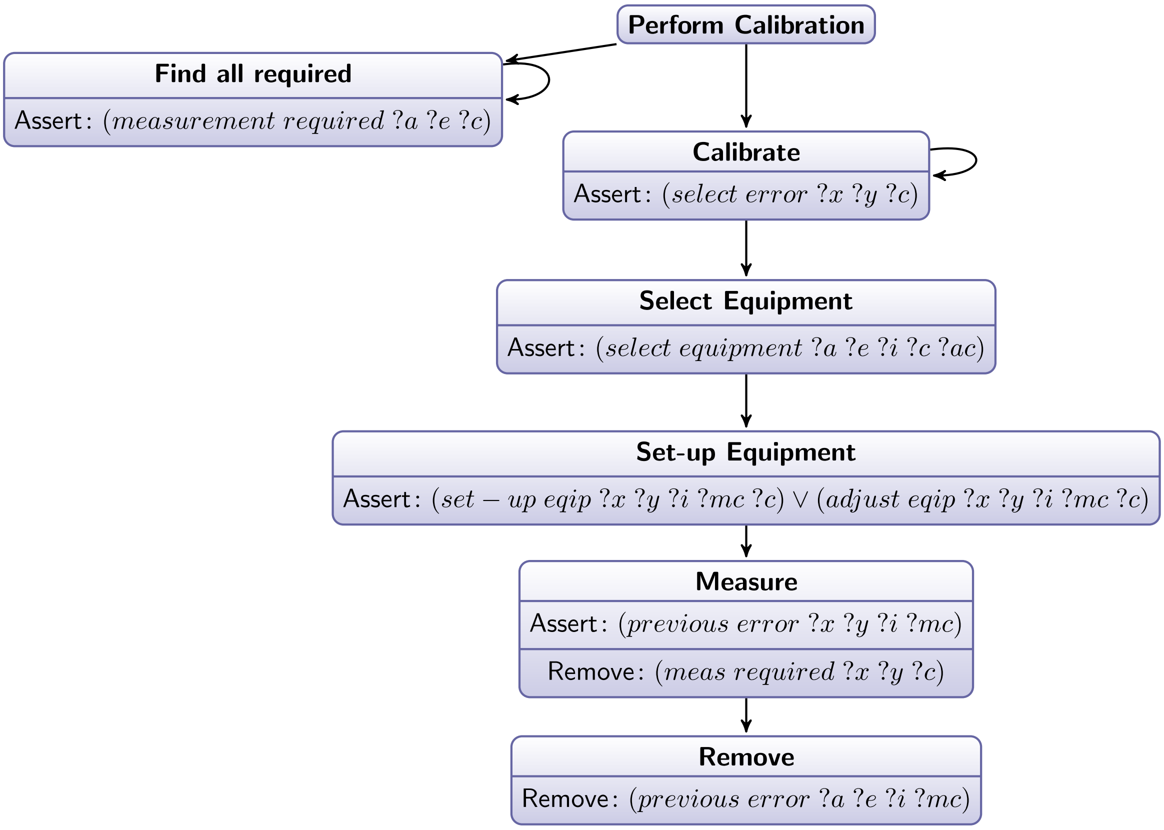


Figure 4.3: HTN model structure - methods and operators

In the model, the seven methods displayed in Table 4.3 are present. Figure 4.3 illustrates the domain model (Appendix A) by showing the flow of the methods and how task decomposition takes place. The operators are shown as assert and remove methods that control the execution of non-primitive tasks.

4.2.3 The Planner

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 [90]. The domain model (Appendix A) is written in LISP using the syntax necessary for the SHOP2 architecture. SHOP2 uses the branch-and-bounds algorithm for finding lowest cost solution to an optimisation problem [90]. Cost calculation is performed by accumulating the individual costs associated with instrumentation set-up, adjustment, measurement and removal as described in Section 4.1.4.

4.2.4 Experimental Analysis

To evaluate the HTNs performance, empirical observations have been made using the following two problem instances:

1. A machine tool with three linear axes. As seen in Figure 2.1, each linear axis will have six geometric plus one non-orthogonal error component. 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 s_0 for this problem is fifty-three.

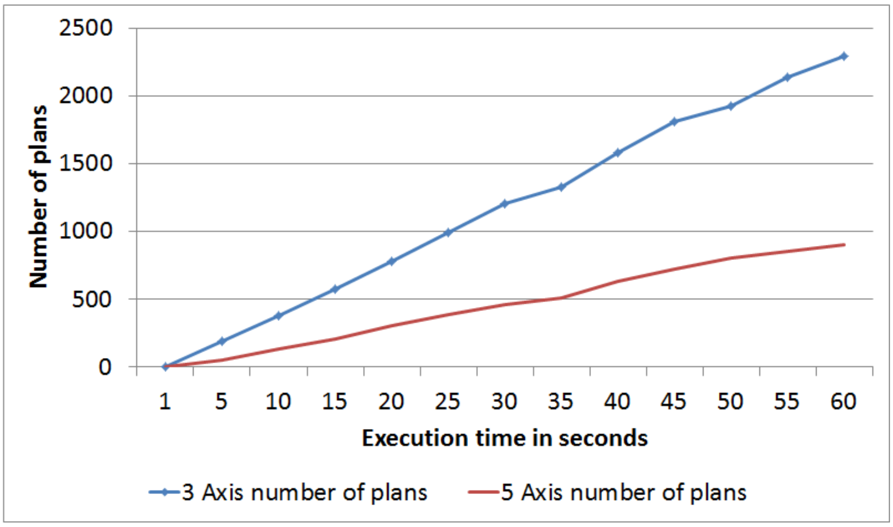
2. A five axis machine tool with three linear and two rotary axes. Each linear axis will have six geometric plus one non-orthogonal error components, and as seen in Figure 2.2, each rotary axis will have ten error components. There will also be a total of five different instruments available, and each error component can be measured by using at least two of the available instruments. The size of s_0 for this problem is ninety-nine.

4.2.4.1 Context

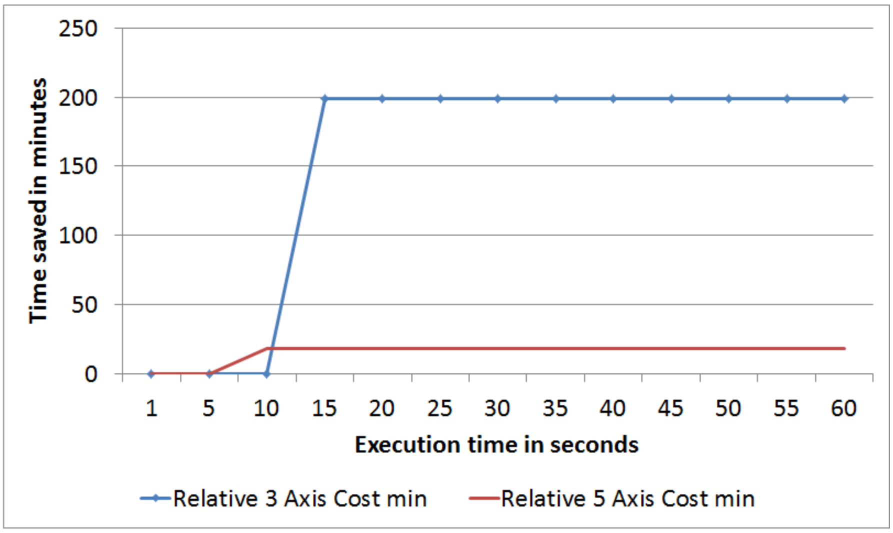
The purpose of this experimental analysis is to examine the performance of using the developed HTN model. The performance is measured in terms of the execution time required to find both valid and optimal calibration plans. The quality of the calibration plan is measured by the duration that the machine will be unavailable for normal manufacturing operation (downtime). Following this, the schedule of the calibration plans will be evaluated with expert knowledge to establish their fitness for purpose.

4.2.4.2 Plan Exploration

Executing the HTN with both the three- and five-axis planning problems will result in the generation of all the 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. The motivation behind



(a) HTN Plan Exploration



(b) HTN Plan Efficiency

FIGURE 4.4: HTN Graphs

this procedure is to get a better understanding of how problem complexity affects the required processing time and solution optimality.

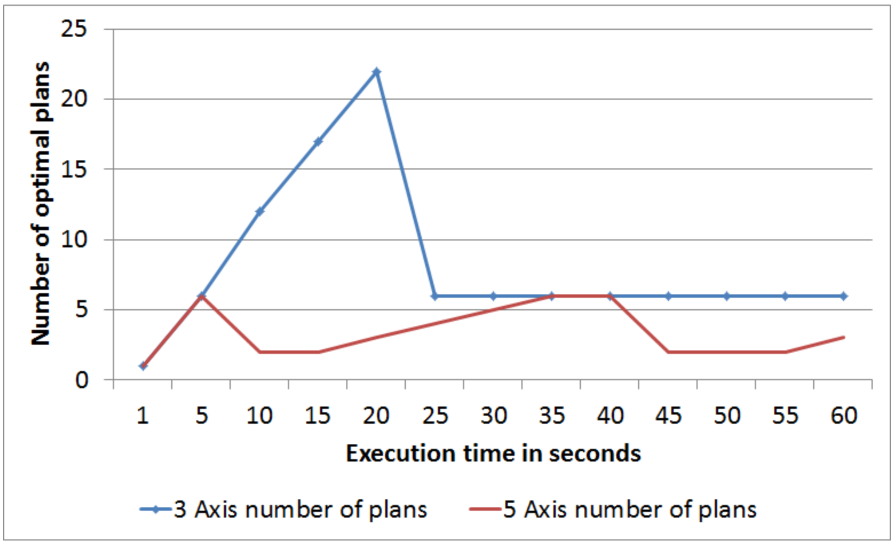
As seen in Figure 4.4(a), it is noticeable that the number of complete plans generated for the three-axis machine is more than twice that of the five-axis machine. This highlights the higher computational effort for larger problem instances. Figure 4.4(b) also shows the efficiency increase in terms of the time saved when comparing the first identified plan with the plan of the lowest cost discovered within the specified time-frame. For the tests that are executing in 5 second intervals, the plan with the lowest cost stabilise at a saving of 200 minutes for a three-axis machine (42:26 (hh:mm)) after exploring 574 plans, and a saving of 18 minutes for a five-axis machine (79:19) 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.

In the above experiment it is surprising to see that the benefit is larger for the three-axis problem instance. This is because more complex problems are more difficult to solve, thus requiring more time and processing power to find more solutions. For the five-axis problem, the optimum solution was found within 10 seconds of execution. Since the five-axis problem contains all the timings for the three-axis problem as well as the additional timings for the rotary axis, it should be possible to get an efficiency gain that is greater than what is seen for the three-axis (200 minutes). However, in this experiment the optimise-cost flag was not used. Therefore, the planner is looking for solutions, rather than optimal solutions, thus any advancement in the optimal is found by exploring more plans and not by searching using a better heuristics.

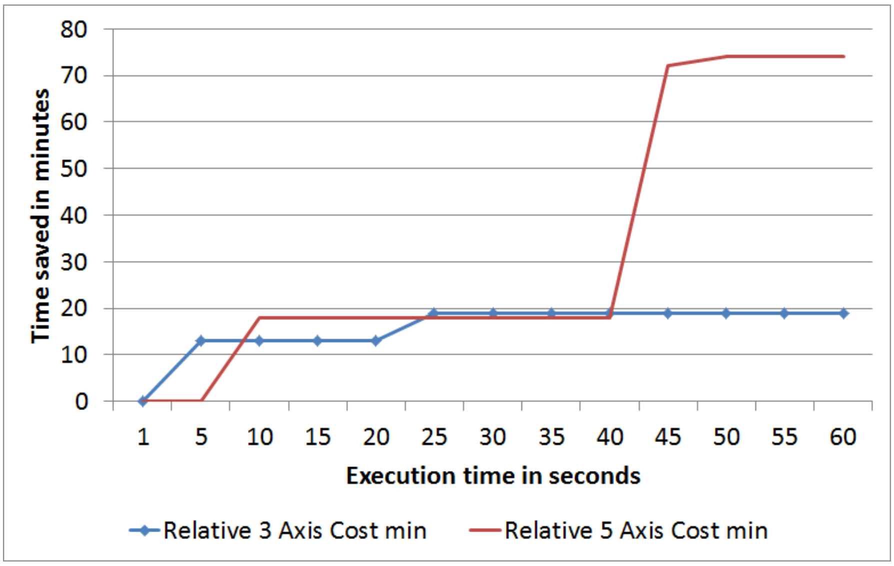
4.2.4.3 Plan Optimisation

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 4.5(a) that the number of complete plans generated in the allocated time frame is much lower with the use of the branch-and-bounds algorithm.

It is also noticeable in Figure 4.5(a) that the number of 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 behaviour is because the branch-and-bound optimization 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 4.5(b) 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 (42:20). The five-axis problem increases rapidly until it stabilises with an efficiency gain of 74 minutes (78:23) in 50 seconds of execution time.



(a) HTN Plan Exploration (optimised)



(b) HTN Plan Efficiency (optimised)

FIGURE 4.5: HTN Graphs (optimised)

4.2.4.4 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 irrelevant providing that the identified plans are the most efficient and the method is robust.

It is evident from Table 4.4 that the first identified plan for the three-axis machine when using the branch-and-bound algorithm has a cost of 42:39 which is 3:06 reduction over the plan where optimisation is not used. The initial cost for a five-axis machine has the same cost for both tests. As seen in Table 4.5 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. Further experimentation was undertaken and concluded that increasing the search time beyond sixty seconds did not result in the production of optimised plans with a lower cost.

Plan	First Plan Cost	First Optimised Plan Cost	Difference
3-axis	45:45	42:39	3:06
5-axis	79:37	79:37	0

Table 4.4: Comparison of the first identified HTN plan

Plan	Lowest Plan Cost	Lowest Optimised Plan Cost	Difference
3-axis	42:26	42:20	0:06
5-axis	79:19	78:23	0:56

Table 4.5: Comparison of optimised plan cost

Table 4.6 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 problems complexity also increases.

Plan	Non-optimised Time	Optimised Time
3-axis	0:15	0:50
5-axis	0:10	0:45

Table 4.6: Comparison of execution time to identify lowest cost plan

4.2.4.5 Industrial and Produced Plan Comparison

An industrial case-study conducted using the same machine tool, instrumentation and measurement techniques as the expert produced calibration plans in Section 2.4.2 was then performed.

Figure 4.6 shows the most efficient plan identified by the HTN algorithm within a ten minute period. It is immediately noticeable that the planner has grouped the measurements into axis order, much like that of the academic's calibration plan seen in Figure 2.12. The exception to this ordering is where the non-orthogonal measurements have been grouped together because the model has evaluated that it is more efficient for them to be performed directly after each other.

It is also noticeable that the model has selected the equipment which can perform the required measurement in the lowest time. It is evident that the model has selected equipment, and prioritised the measurements, based on instrumentation that can be adjusted to save time.

The result from using the HTN model show that calibration plans can be automatically constructed, whilst minimising machine tool downtime. However, the HTN is only a prototype system and does not pay full attention to the parameters listing in Section 4.1. Producing an HTN model was the quickest and most logical way to apply automated planning to machine tool calibration. However, to provide a better solution the created model should be expanded and written in PDDL, allowing for a range state-of-the-art domain-independent plans to be used to further improve performance. This is discussed in the following section and remainder of this chapter.

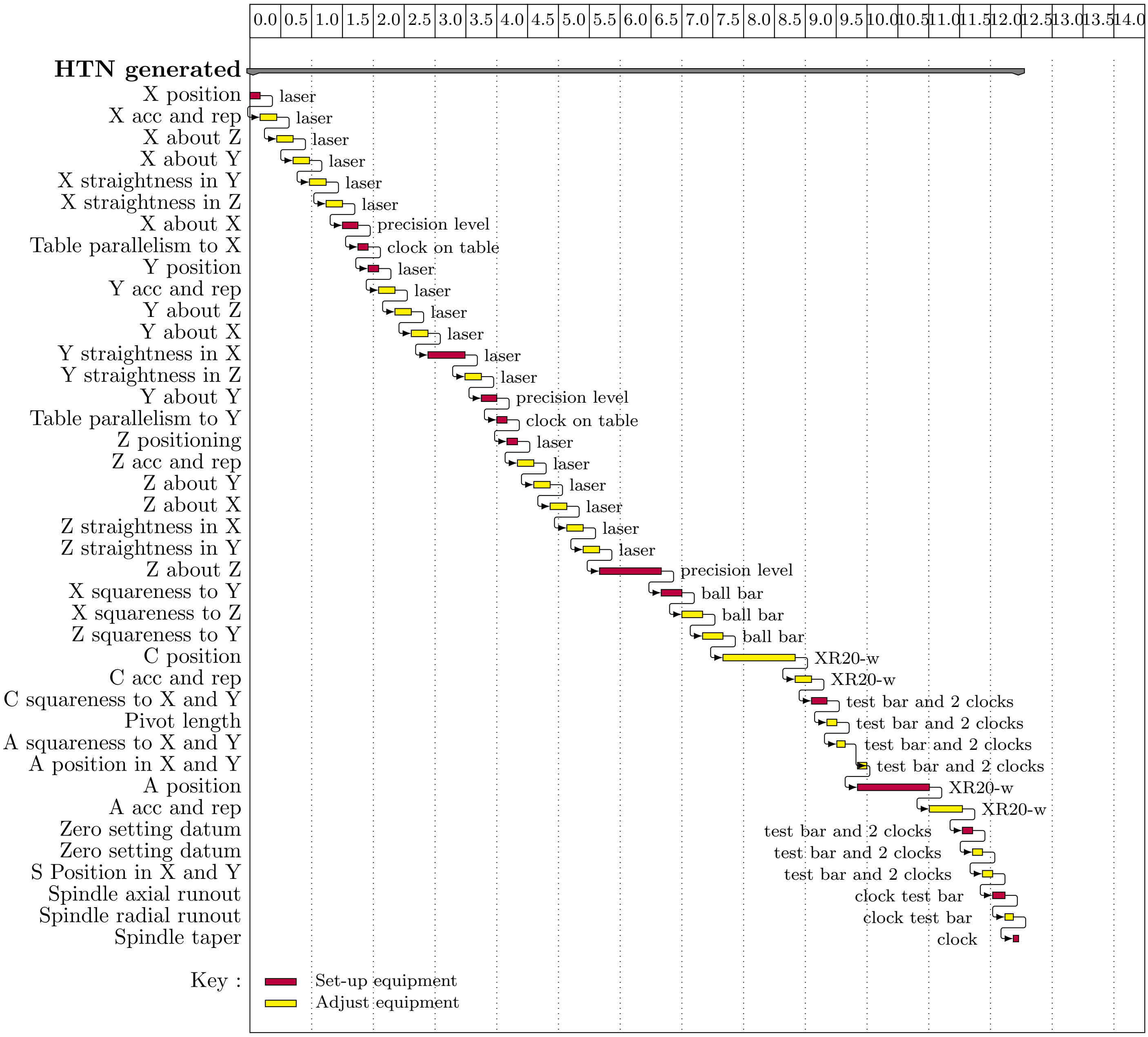
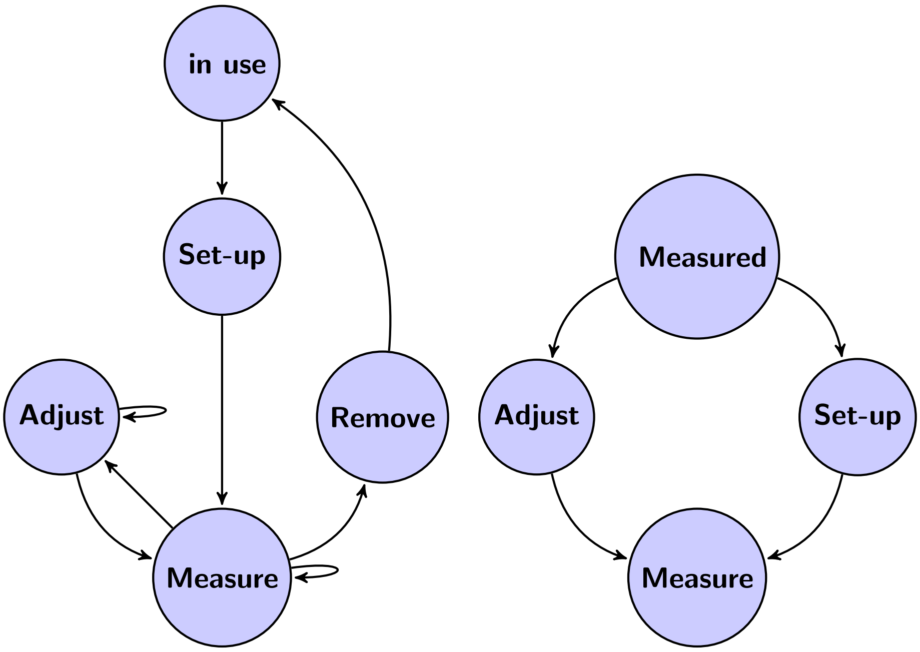


FIGURE 4.6: HTN calibration plan

4.3 PDDL Solution

As discussed in Section 3.7.1, PDDL has been through many revisions to extend its modelling capabilities. Given that producing machine tool calibration plans is a temporal reduction problem, PDDL2 is required to encode and allow reasoning of time.



(a) Instrument Object (b) Error Object

FIGURE 4.7: Diagrammatic illustration of the timeline of instrument and error objects in the PDDL model.

4.3.1 Objects, Predicates and Functions

The PDDL model contains three different objects that are manipulated during the planning process: (1) Axis, (2) Error, and (3) Instrument. This is different from the design of the SHOP2 HTN model because the PDDL model is object-oriented whereas the HTN model is task-oriented. This process is illustrated in Figure 4.7 where the interaction between the objects and the PDDL actions is shown. It is noticeable that the instrument object is involved in more interactions when compared to the error object. It is also noticeable that the logical flow in which these objects interact is different. For the instrument object, it is possible that the instrument will be adjusted to measure either another portion of the error component, or a different error component before it is removed.

Table 4.7 contains a set of predicates that are used in the PDDL model to describe the configuration of the machine tool, the measurement requirements and the instruments that are capable of performing the measurement. As a means of specifying which instruments can possibly operate simultaneously, the predicate `(compatible ?ins1 ?ins2 - Instrument)` is used. Additionally, the `(blocked ?in - Instrument ?ax - Axis)` predicated provides the means of specifying when a specific piece of equipment cannot be used on the machine tool in question. Additionally, a `(working-day)` predicate is used to determine when the factory is open and access can be gained to the machine tool. It is intended that the `(working-day)` will be used as a TILs to specify the working hours as predictable exogenous events.

Parameter	PDDL Predicates
T1	`(axis-error ?axi - Axis ?err - Error))`
I6	`(measures ?ins - Instrument ?err - Error)`
I7	`(in-operation ?ins - Instrument)`
I8	`(set-up-axis ?ins - Instrument ?axi - Axis))`
I9	`(set-up-error ?ins - Instrument ?err - Error)`
I10	`(compatible ?ins1 ?ins2 - Instrument)`
I11	`(blocked ?in - Instrument ?ax - axis)`
T3	`(working-day)`

Table 4.7: Calibration parameter to PDDL predicate mapping

4.3.2 Functions

In a PDDL model, functions (numeric fluents) provide the means to store and access numeric values in the initial state, goal state and during search. In the machine tool calibration domain, functions provide the necessary means of storing numerics to represent the instrument, measurement and machine tool aspects. Table 4.8 shows the mapping between the individual parameters shown in Section 4.1 and the PDDL functions in the implemented model.

Parameter	PDDL Function
I1	`(set-up-time ?in - Instrument ?er - Error ?ax - Axis)`
I2	`(measurement-time ?in - Instrument ?er - Error ?ax - Axis)`
I3	`(removal-time ?in - Instrument ?er - Error ?ax - Axis)`
I4	`(adjust-error-time ?in - Instrument ?er - Error ?ax - Axis)`
I5	`(adjust-position-time ?in - Instrument ?er - Error ?ax - Axis)`
M1	`(dwell ?ax - Axis ?er - Error ?in - Instrument)`
M2	`(feedrate ?ax - Axis ?er - Error ?in - Instrument)`
M3	`(targets ?ax - Axis ?er - Error ?in - Instrument)`
M4	`(resolution ?ax - Axis ?er - Error ?in - Instrument)`
M5	`(accuracy ?ax - Axis ?er - Error ?in - Instrument)`
T1	`(length-to-measure ?ax - Axis ?er - Error ?in - Instrument))`
T2	`(significance ?ax - axis ?er - Error)`
T4	`(amount-measured ?ax - Axis ?er - Error ?in - Instrument))`

Table 4.8: Calibration parameter to PDDL function mapping

4.3.3 Actions

In PDDL, an action is a way of changing the current state of the world and is made up of a preconditions list and an effect list. If the precondition list can be satisfied by the current state, then the effects are asserted. Because it is desired to minimising the total plan length, durative actions are being used. Durative actions return a cost (in time) for the action to take place. Durative actions differ from regular PDDL actions because they allow for `at start`, `over all` and `at end` semantics. This means that a precondition and effect can be required to have the timing satification or assertion of `at start`, `over all` and `at end`.

The following section describes the PDDL durative actions in table form showing the cross-reference to the parameters identified in Section 4.1. The full PDDL domain can be found in Appendix B.

The “set-up” action models the logical preconditions and effects that take place when an instrument is set-up on an axis to measure an error component. This action interacts with the instrument action and error objects to determine the error components that are still to be measured and which instruments are capable of performing the measurement. The actions duration is established by a numeric fluent present in the initial state that denotes the estimated time to set-up the equipment to measure that specific error component. The possibility of concurrency is handled by an ADL condition defining for all instruments that are compatible with the chosen instrumentation, and can measure an error component on the same axis, can be set up concurrently providing that their measurement parameters agree.

SET-UP	
parameters	?i - instrument ?a - axis ?e - error
duration	**I1** set-up time of ?i on ?a
preconditions	**I11**: ?i is not blocked on ?a **T1**: error ?e requires measuring on ?a **I6**: instrument ?i can measure error ?e on axis **I8** and **I9**: operating range of ?i is sufficient **I8** and **I9**: for all set-up instruments ?j, ?j is set-up on ?a **I10**: for all set-up instruments ?j, ?j is compatible with ?i **T3**: set-up occurs during the working day **M1**, **M2**, **M3**, **M3**, and **M4**: for all set-up instrument ?j for error ?k on axis ?l, the dwell time, feedrate and target count for ?a ?e ?i are compatible
effects	**I8** and **I9**: ?i is set-up to test ?e on ?a **I7**: increment the number of tests being performed by ?i

In some instances, it is possible that the instrumentation will need to be adjusted multiple times allowing for multiple readings to be taken at different locations. The "adjust position" durative action adjusts an instrument to measure the remainder of an error component. For example, measuring the straightness of a 1.2m linear axis with a 0.8m granite straight edge. This is done by analysing whether the `length-measured` numeric fluent is less than the `length-to-measure` numeric fluent. In the same way as for the set-up durative action, concurrency is handled to allow for measurement repositions to happen simultaneously where possible. During the "measure" action any overlap due to measurement stitching is deducted from the amount measured.

ADJUST POSITION	
parameters	?i - instrument ?a - axis ?e - error
duration	**I5** adjustment time of ?i on ?a
preconditions	**I8** and **I9**: ?i is set-up on ?a to measure error ?e **I8** and **I9**: for all set-up instruments ?j, ?j is set-up on ?a **I10**: for all set-up instruments ?j, ?j is compatible with ?i **I7**: instrument ?i is current in operation **T4**: amount measured for error ?e is less than the length to measure for ?e **T3**: adjustment occurs during the working day **M1**, **M2**, **M3**, **M3**, and **M4**: for all set-up instrument ?j for error ?k on axis ?l, the dwell time, feedrate and target count for ?a ?e ?i are compatible
effects	**I8** and **I9**: ?i is set-up to test ?e on ?a **I7**: increment the number of tests being performed by ?i

It is likely that the same piece of instrumentation can measure multiple error components. The "adjust error" durative action models the process of switching the current instrumentation set-up to measure a different error component. This method handles concurrency in the same way as the set-up action. Any currently set-up instrumentation can be adjusted simultaneously to measure another error component.

ADJUST ERROR	
parameters	?i - instrument ?a - axis ?e - error
duration	**I4** adjustment time of ?i on ?a
preconditions	**I8** and **I9**: ?i is set-up on ?a to measure error ?e **T1**: error ?e requires measuring on ?a **I6**: instrument ?i can measure error ?e **I8** and **I9**: operating range of ?i is sufficient **T3**: adjustment occurs during the working day
effects	**I8** and **I9**: ?i is set-up to test ?e on ?a **I7**: increment the number of tests being performed by ?i

The "measure" durative action models performing the measurement and acquiring the required data. The measurement action handles concurrency by assuming that any measurements that have been set-up or adjusted at the same time can be measured concurrently because they have previously been identified as concurrently compatible.

MEASURE	
parameters	?i - instrument ?a - axis ?e - error
duration	**I2** measurement time of ?i for ?e on ?a
preconditions	**I8** and **I9**: ?i is set-up on ?a to measure error ?e **T1**?e has not been measured on ?a **T3**: set-up occurs during the working day
effects	**T1**: ?e is measured on ?a **T2**:increase the global significance by the significance of ?e on ?a **I7**: decrease the number of tests being performed by ?i **T4**: increase the amount measured for ?e by the amount minus any overlap

The "remove" action removes an instrument from an axis, providing that the instrument is not currently set-up to measure any other error component on that axis.

REMOVE	
parameters	?i - instrument ?a - axis ?e - error
duration	**I3** remove time of ?i on axis ?e
preconditions	**I8** and **I9**: ?i is set-up on ?a to measure error ?e **I7**: the number of tests being performed by ?i is 0
effects	**I8** and **I9**: ?i is not set-up on ?a to measure error ?e

4.3.4 Initial and Goal State

The initial and goal state provided in the PDDL problem file is a set of objects, predicates and function values that are instantiated in the initial state, as well as a set of predicates that make up the goal state. Objects provide the architecture to model the physical aspects of calibration planning. In the calibration problem there are three different objects: (1) axis, (2) error components, and (3) instrument. Combining these objects with predicates in the initial state makes it possible to model the machine configuration and available instrumentation. For example, `x y z - Axis` specifies that the machine has three axes (X, Y and Z). Using the axis and error component objects it is possible to state their error components using the predicate: `(axis-error x position)`.

Functions can be specified in the initial and goal state to assign a value to the numeric fluent. For example, `(= (length-to-measure x position) 1500)` defines that the length to measure in the initial state is assigned the value 1500mm. This value can then be used during plan exploration within an actions precondition and effect.

Predicable exogenous events (TILS) are also defined in the initial state. For example, the TIL `(at 540 (not(working-day)))` defines that at nine hours in the plan, the working-day predicate becomes false and any action that requires working-day to be true in its precondition will not be satisfied.

4.3.5 Plan Metric

The PDDL model contains two different metrics.

1. **Time:** Each of the durative actions has an associated duration. During planning, these durations are accumulated to determine the total-time taken to reach the goal when using the produced plan. Planners are able to keep track of a 'total-time' fluent and the calibration plan can be optimised to reduce it.

2. **Significance:** Each of the different errors on a machine has a different significance value. Depending on the work-piece and its tolerances, different axes will also hold more significance. This is typically due to which axis holds the other axes. When there is insufficient time to fully calibrate a machine (E.g only one day is permitted for a calibration), it is still desirable to test the most significant errors in the time available. Therefore, we maintain a 'global significance' fluent that sums the significance of the errors measured in the plan. The significance of an error for an axis is taken as the product of the significance of the axis and the significance of the error independent of a particular configuration. For example, the roll error of a Z-axis on a three-axis machining centre is insignificant because it will only result in rotation of the cutting tool which will have no effect on the work-piece. Conversely, the Z-axis positional deviation would have a significant impact on the depth tolerance of a hole drilled in the work-piece.

4.4 Experimental Analysis

The benchmarks that are provided take timing information from the expert produced calibration plans seen in Section 2.4.2.2. Additionally, the benchmarks are close to reality; the configurations of the machines are common configurations and the timings are derived from similar real machines and were validated by experienced users performing the calibration.

The instances are based on four different machine configurations.The first two instances are based on machine configurations with three linear axes. Each linear axis will have six geometric error component. Additionally, there will be a non-orthogonal error between any nominally perpendicular axes. There are a total of eight different instruments available, and each error component can be measured by using at least two of the available instruments. Secondly, tests are performed on two five axis machine configuration with three linear and two rotary axes. Each linear axis will have six geometric and there will be non-orthogonal errors between each. Additionally, each rotary axis will have ten error components. There will also be a total of eight different instruments available, and each error component can be measured by using at least two of the available instruments. For each machine, there are three different instances (denoted A, B and C in the tables) which correspond to models with different timings for setting up and adjusting the instruments. Even for the same machine, depending on the experience of the engineer, setting up and adjusting instrumentation will take a variable amount of time.

Two sets of experiments have been conducted. The first is to compare the calibration plans produced by the HTN planner and the calibration plans produced by LPG. In the HTN, the **T4** constraints has not been encoded. This is because predictable exogenous events can not be handled by the SHOP2 architecture. The second contains concurrent actions both allowed and disallowed for LPG-td, showing whether or not any benefit is gained from this approach.

4.4.1 Context

The purpose of this experimental analysis is to examine and understand the structure and quality when using the developed PDDL model. The PDDL model is tested when using both simultaneous and concurrent measurements to establish their effect on machine tool downtime. In this analysis, the downtime of the HTN and PDDL produced calibration plans are compared and discussed. Additionally, experimental analysis is performed to examine the possibility of producing calibration plans that span multiple working days. In this analysis, the downtime, quality (in terms of summed significance), and the quantity of tests measured are compared for twelve different problem instances when imposing a one, two and five day limit. The structure of the calibration plans is then evaluated with expert knowledge to establish their fitness for purpose.

Instance	SHOP2	LPG_S	LPG_C
3AX-01A	30:17	33:04	12:40
3AX-01B	26:43	27:08	12:42
3AX-01C	27:15	29:08	11:56
5AX-01A	54:35	53:55	30:09
5AX-01B	45:59	49:39	29:53
5AX-01C	45:59	51:45	28:26
3AX-02A	29:34	30:28	13:14
3AX-02B	26:01	27:41	11:20
3AX-02C	19:40	18:12	8:14
5AX-02A	50:40	46:33	25:16
5AX-02B	47:00	36:56	25:06
5AX-02C	37:00	37:52	20:49

TABLE 4.9: Comparison Between SHOP2 and LPG-td on 12 Machine Tool Calibration Instances

4.4.2 HTN and PDDL Planner Comparison

Table 4.9 shows the results of comparing SHOP2 with LPG-td on 12 machine tool calibration instances. Six of the instances are from three-axis machines, six from five-axis machines. The results show the length of the plans in minutes. LPG_S and LPG_C are result from LPG-td when finding sequential and concurrent plans, respectively. This work is not intended to show the relative merits of planning using HTN and PDDL encodings. The results show, that in the sequential case, the HTN typically provides plans with a shorter duration. However, the differences are typically quite small, and it is clearly possible to find good solutions with either planning technique. Once concurrency is allowed, the PDDL model makes it possible to find much shorter plans, typically halving the plan length.

The second set of experiments show the effect of introducing the working day constraints (**T4**) in the model. The results shown minimising the timespan of the plan when the number of days available exceeds the minimum days required to calibrate the machine. The result of maximising the significance of the tests carried out in the case when there is a limited time to carry out the calibration are also shown. Table 4.10 shows the result of introducing the working day constraint. The first 'Time' column shows the makespan (as *days*,*hours*:*minutes*) of the best quality found global solution. The remainder of the table shows the overall quality (in terms significance of errors measured) of the plans

		1 Day		2 Days		5 Days	
Instance	Time	Quality	Tests	Quality	Tests	Quality	Tests
3AX01A	2,2:35	5654	13	6689	19	6857	21
3AX01B	2,2:03	5206	11	5902	15	6604	21
3AX01C	1,5:50	6358	14	7416	21	7916	21
5AX01A	3,1:42	4374	10	6013	18	9808	41
5AX01B	3,1:28	5222	12	5286	15	9847	41
5AX01C	3,0:39	4838	12	5517	18	9514	41
3AX02A	2,4:30	5877	15	6285	17	6856	21
3AX02B	1,7:57	5585	13	6492	21	6604	21
3AX02C	1,1:56	6837	17	7416	21	7417	21
5AX02A	2,23:28	4372	15	4594	20	9633	41
5AX02B	2,16:53	4548	10	4905	15	9372	41
5AX02C	2,18:16	5097	17	3568	15	9031	41

TABLE 4.10: The results of solving the test instances with the working day constraints enabled

found within a restricted makespan, and also the number of errors that were measured in those plans. The significance is calculated by taking the significance value for each geometric error and multiplying it by the time when it is measured within the plan, thus minimising the summed significance will result in error components with a higher significance being measured earlier on in the plan. The makespan of these plans is shown in the first column ('Time'). When the working day constraints are set to five days, all the error components in each problem instance are measured. However, when the working day constraint is reduced to two and one day, only those with the highest significance are measured.

When solving the problems with limited makespan, no goals are enforced, but a metric is set to maximize the global significance. As can be seen in the results, LPG does solve these problems whilst taking into account the metric function. When given extra time, it solves the problem with a higher metric value. In some cases, for the three-axis machines, there is sufficient time to satisfy all of the goals, these are the cases when 21 errors are measured.

4.4.3 Encoding Numerics Propositionally

Numeric preconditions and effects significantly reduce the range of planners that can solve the problem. An alternative, yet not obvious, method is to re-encode the domain encoding the numerics propositionally using predicates and objects. For example, encoding the `length-to-measure` and `amount-measured` functions propositionally can be achieved by introducing the following components:

- `Distance` object use to represent the distance numeric.
- `(length-to-measure ?ax - Axis ?er - Error ?d - Distance)` predicate to represent the length to measure by using a distance object.
- `(amount-measured ?ax - Axis ?er - Error ?d - Distance)` predicate to represent the length that has currently been measured using a distance object.
- `(working-range ?in - Instrument ?d1 - Distance ?d2 - Distance)` predicate to represent the working-range of an instrument. This predicate is also used to order the set of distance objects.

The modification of the `length-to-measure` and `amount-measured` predicates can be performed during the execution of an action. For example, during the measurement action it is possible to check that the amount measured is not the length to measure by using the precondition:

```
(at start (and(amount-measured ?ax ?er ?d) (not(length-to-measure ?ax ?er ?d))))
```

If this precondition is satisfied, another precondition must also be satisfied determining the next length in the set:

```
(at end (working-range ?in ?d ?d1))
```

This would allow for the amount measured predicated to be updated to the new distance:

```
(at end (amount-measured ?ax ?er ?d1))
(at end (not(amount-measured ?ax ?er ?d)))
```

significantly more objects and predicates are required in the initial state to allow the propositional encoding to work. Firstly, the Distance objects must be defined, and secondly, the `working-range` set ordering encoding must be added. An example is shown in the following:

```
(working-range laser-interferometer zero oneh)
(working-range laser-interferometer oneh twoh)
(working-range laser-interferometer twoh threeh)
(working-range laser-interferometer threeh fourh)
```

This method allows for removing of fluents, conditions and effects. However, it complicates the domain model by adding many more objects and predicates. Additionally, it also reduces the numeric granularity. For example, if the an instrument requires the stepsize of 25mm, then an object to represent every 25mm throughout the travel would be required and a large set of predicates would be required to define their relationship.

4.4.3.1 Experimental Data

Using the LPG-td planner, a comparison can be made between two identical domains, one encoding the values using numeric fluents and one where numerics are encoded propositionally. The results shown in Table 4.11 show the difference in search time and duration of the solution for both a three- and five-axis calibration within a 10 minute period. The table shows the LPG-td is able to find the same optimal plan using using either method. However, the number of optimal plans found with the lowest cost is higher. This indicates that LPG-td finds the numeric fluent domain more computationally complex in comparison with the propositional encoding.

Instance	Fluent Encoding		Propositional Encoding	
	Number of Plans	Lowest Plan Duration	Number of Plans	Lowest Plan Duration
3-axis	5	35:30	8	35:30
5-axis	2	58:00	3	58:00

TABLE 4.11: Comparison between fluent and propositional numeric encoding

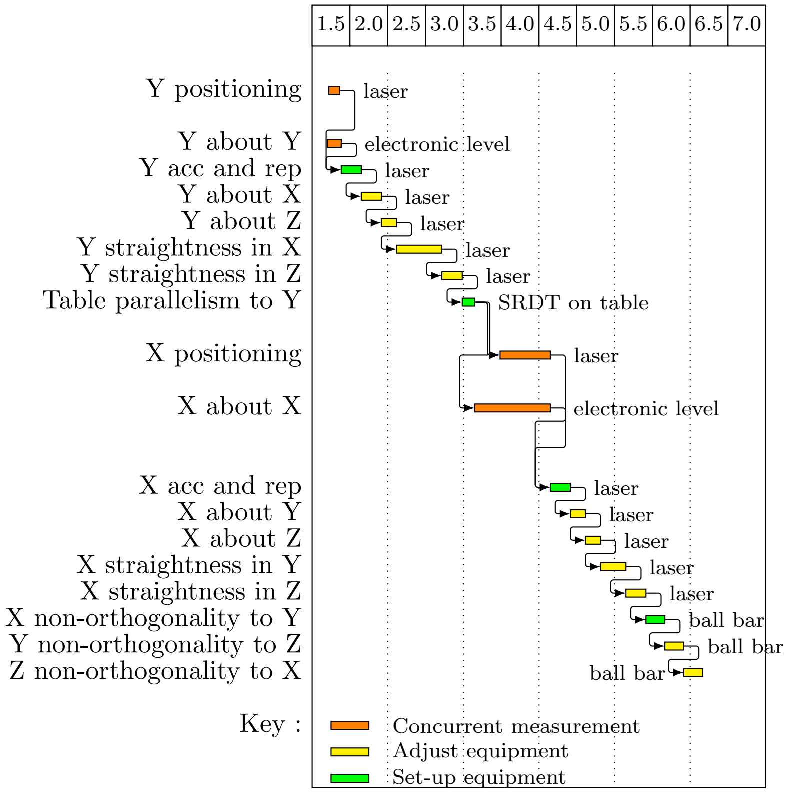


Figure 4.8: PDDL Calibration plan

4.4.4 Industrial and Produced Plan Comparison

Differently from the expert's plans and HTN calibration plans, the PDDL model has produced a plan that contains measurements that can be performed simultaneously.

The PDDL-produced plan contains the same ordering as the HTN-produced plan, but there are differences in terms of test instrumentation selection. For this reason, only an excerpt of the PDDL produced plan is shown. Figure 4.8 shows that the first two measurements E_{YY} and E_{BY} can be performed simultaneously. This is possible because both tests involve moving the axis by the predefined amount, over the same range but with different equipment. This agreement of parameters, and the absence of physical obstruction or interferences, means that both the measurements can be performed simultaneously.

4.4.4.1 Plan Duration

Table 4.12 shows the estimated time for the four calibration plans. It can be seen that the industrial expert's plan is one hour shorter than the academic expert's plan, which results from different calibration objective, experience and different equipment as described in Section 2.4.2.2. It is also evident that the HTN produced calibration plan is forty five minutes more efficient that the academics plan in terms of time, but does not give any time-saving over the industrial expert's plan, even though the HTN planner has optimised the plan to cluster the use of instrumentation together so that it only has to be adjusted, rather than set-up. The reason that the HTN produced plan is longer is because the timings used in the problem definition were the highest taken from the expert and academic plan to ensure that the planner did not under estimate. Taking this into consideration, planning definitions were created to contain the best-case timings. The results can be seen in the lower section of Table 4.12. It is evident that if the best-case timings are taken, the HTN produced calibration plan is reduced by thirty five minutes to twelve hours and ten minutes, which is twenty minutes shorter than the industrial expert's plan.

On the other hand, as seen in Table 4.12, the plan produced by the PDDL model has an estimated execution time of eleven hours and fifty two minutes. This is fifty three minutes shorter than the HTN produced plan, and thirty eight minutes shorter than the industrial expert's calibration plan. The reasoning for that reduction in estimated time is the simultaneous measurements that have been identified and incorporated into the produced calibration plan. Producing a version of the PDDL problem definition using the best-case timings resulted in an additional reduction in plan length of thirty four minutes, making the new total for the lowest cost plan eleven hours and eighteen minutes, which is one hour and twelve minutes shorter than the best expert plan.

Generation method	Time in hours
Industrial expert	12:30
Academic expert	13:30
HTN worst-case	12:45
PDDL worst-case	11:52
HTN best-case	12:10
PDDL best-case	11:18

TABLE 4.12: Comparison of estimated calibration time for different plans.

From these results it is possible to generalise that when using the developed model and encoded knowledge, optimised calibration plans can be found for a given machine configuration, instrumentation and known test methods. However, comparing these results against calibration plans from experts with different opinions could potential result in the production of plans that are not as optimised as those constructed by experts. This is because although the considered academic and industrial experts have extensive and well informed knowledge of machine tool calibration, it is possible that experts elsewhere have new time-saving knowledge that has not been encoded in this PDDL model. In this case, the new expert knowledge would need to be encoded, so that the PDDL generated calibration plans can achieve comparable results.

4.4.4.2 Plan Quality

Both the automated plans follow the same structure of measuring the X, Y and Z linear axis pseudo-static errors followed by measuring the non-orthogonality between each. Next, the C and A rotary axis errors are measured, followed lastly by the measurement of the spindle errors (S axis).

The pseudo static geometric errors of a linear axis are measured in what the model has determined to be the most convenient and efficient order. Taking the X-axis for example, the positional (E_{XX}) error is measured using the Renishaw XL-80 followed by the accuracy and repeatability (X acc and rep) test. Sequencing these two measurements is logical because they both use the same equipment set-up, only the accuracy and repeatability test is repeated a set amount of five times. Next, the pitch (E_{AX}) and yaw (E_{CX}) angular errors are measured using laser interferometry. Both these measurements use the same equipment and machine parameters, making it logical for them to be clustered together, even if the angular optics are aligned differently. Similarly, laser interferometry is then used to measure the two (E_{YX} and E_{ZX}) straightness errors because the only difference is the orientation of the optics'. Finally, the roll error (E_{BX}) is measured using a precision level. This measurement is scheduled as the last for the pseudo static geometric error because it requires the use of different instrumentation, which in this case is a precision level. Once all the six-degree-of-freedom errors have been measured for each linear axes, the non-orthogonal errors between each are then measured (X non-orthogonality to Y, X to Z and Z to Y). These three measurements are sequenced together because they all make use of the ball bar equipment as well as the movement of two linear axis, making it not only more time efficient to group them together, but also an altogether more repeatable metrological process. When measuring the linear component errors, it would be bad practice to measure the non-orthogonal error half way through because it would involve changing the position of the machine's other axes, reducing the measurements repeatability.

Next, the rotary axes are measured starting with the C axis. Firstly, the positioning error of the C axis is using the Renishaw XR20-w rotary axis calibrator. This is then followed by the accuracy and repeatability measurement because, like for the linear axes, the same instrumentation and test set-up is used. The XR20-w is then no longer required for the C axis, so the C axis non-orthogonality to the X and Y axis is then measured using a test bar and two SRDTs. This is then followed by measurement of the pivot length using the same equipment. Next the plan focuses its attention to the measurement of the A axis errors that can be measured using the instrumentation that is already set-up on the C axis. The planner has identified that measuring the non-orthogonality in X and Y, followed by the measurement of the position in X and Y is the most efficient choice in terms of time. Once the planner has accounted for the measurements that can be performed using the test bar and two SRDTs, it then finds the suitable way of measuring the A position and A accuracy and repeatability sequentially using the XR20-w. The final two component errors that require planning are the two zero settings errors, which are measured using the test bar and two SRDTs.

The remainder of the calibration plan contains the spindle component errors. The first spindle component error to be measured is the spindle position in X and Y because the instrumentation that was last used on the rotary A-axis, which is the test bar and two SRDTs. Following this, the spindle's axial and radial runout are measured using a SRDTs and a test bar, which are subcomponents of the previous instrumentation. The final component error left to measure is the spindle taper which is performed using a single SRDT.

This same ordering is evident in the plan produced from the PDDL model. The difference being that some of the measurements are scheduled concurrently rather than sequentially. Taking the concurrent planning of the E_{YY} and E_{BY} (roll) that can be seen in Figure 4.8 as an example, it is possible to evaluate the effect this scheduling has on the plan's quality. Based on the machine's configuration and available instrumentation it is a viable choice to set-up the instrumentation and then measure both simultaneously, and does not, therefore reduce the plan's quality.

4.5 Chapter Summary

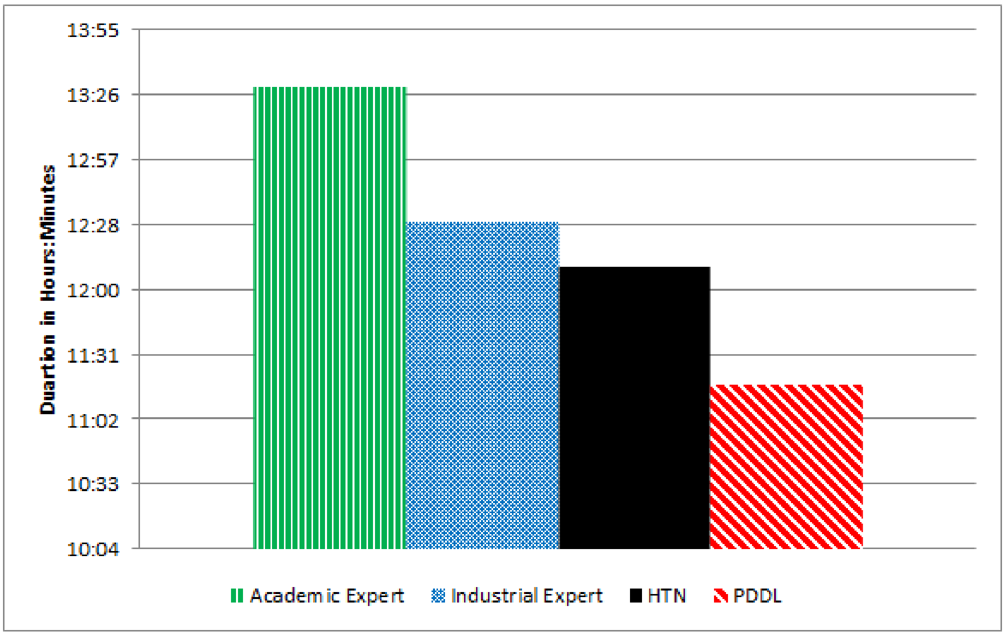


Figure 4.9: Comparison between expert and automated plans

In this chapter, machine tool calibration planning has been broken down into a logical process parameters that describe the parameters for each calibration. Using this process, an HTN method of automated planning using the SHOP2 architecture was developed and tested to evaluate the feasibility of using automated planning and scheduling for machine tool calibration. The HTN model produced calibration plans that intelligently order the measurements to reduce instrumentation configuration time.

Following this achievement, a PDDL model was produced that could be used with more powerful, state-of-the-art planning tools to further reduce the duration of the produced calibration plan. However, the complexity of the produced model resulted in few planning tools that can support all the required PDDL language features. This resulted in the production of an alternative domain where numerics were encoded propositionally to increase the range of planners that can process the domain. However, due to the other requirements of the domain (e.g ADL), only LPG-td can be used. The results indicated the LPG-td finds the domain with numeric fluents more computationally complex.

The calibration plans produced by the HTN and PDDL model have been compared with those produced by industrial and academic experts (Section 2.4.2). In summary, this has resulted in the observation that automatically generating calibration plans can produce valid calibration plans. In addition, it has also been identified that the calibration downtime when using automated planning can be reduced when compared to expert generated plans. Figure 4.9 illustrates the downtime for the calibration plans produced by an academic expert, industrial expert, the HTN, and PDDL model. In this figure it is noticeable that the PDDL model produced a 10.6% reduction in machine tool downtime when compared to the expert generated plan. This 10.6% reduction can result in a potential £120 reduction for a single calibration.

Chapter 5

Uncertainty of Measurement Optimisation

In the previous chapter, a temporal model has been developed to reduce machine tool downtime. However, as identified in Chapter 2.3 the uncertainty of measurement is also a key criterion. In this section, a method of extending the temporal optimisation model (Chapter 5) to also include an optimisation function for the uncertainty of measurement is investigated.

5.1 Temporal Model Extension

Firstly, the extension of the PDDL2.2 model (Section 4.3) is investigated for a single, linear laser measurement to identify the feasibility of the approach. This leads to the development of a universal and extensible method suitable for reducing the uncertainty of measurement due to the ordering of the plan.

5.1.1 Uncertainty of Linear Laser Measurement

ISO 230 part 1 [12] defines linear deviation as "...the straightness of the trajectory of the functional point or the representative point of a moving component." In this work, the measurement of linear deviation (positioning) using a laser interferometer is considered.

The following section provides the equations for estimating uncertainty of measurement as found in ISO 230 part 9 [23].

Firstly, the device's calibration certificate is used to calculate its uncertainty ($u_{DEVICE\,LASER}$) using Equation 5.1 where the calibration uncertainty has been provided in micrometers per metre (µm/m). If the calibration uncertainty is provided in micrometers (µm) the length L should be removed.

$$u_{DEVICE\,LASER} = \frac{U_{CALIBRATION} \times L}{k} \tag{5.1}$$

Measuring positioning error using laser interferometry requires the alignment of the laser beam parallel to the axis under test. A misalignment between the laser and axis can be observed and can be reduced by manual adjustment of the laser. Misalignment of the laser has a second order effect on the measurement and the difference in length ($\Delta L_{MISALIGNMENT}$) as a result of the misalignment can be calculated using Equation 5.2.

$$\Delta L_{MISALIGNMENT} = L \times (1 - \cos \gamma) \times 1000 \tag{5.2}$$

The influence of the misalignment can be significant on short travel axes. From the difference in length, the uncertainty contribution ($u_{MISALIGNMENT}$) can then be calculated using Equation 5.3.

$$u_{MISALIGNMENT} = \frac{\Delta L_{MISALIGNMENT}}{2\sqrt{3}} \tag{5.3}$$

As stated in ISO 230 part 2 [3] in Section 3.1, the "measuring instrument and the measured object are soaked in an environment at a temperature of 20°C" before any measurement takes place. Therefore, any deviation from this temperature should result in compensation of the machine tool. This compensation introduces the uncertainty of the temperature measurement, and the uncertainty of the coefficient of thermal expansion of the machine tool. Equation 5.4 provides the method for calculating the uncertainty due to the temperature change of the machine tool ($u_{M,MACHINETOOL}$). Equation 5.5 describes how to calculate the uncertainty due to the coefficient of thermal expansion of the machine tool ($u_{E,MACHINETOOL}$).

$$u_{M,MACHINE\,TOOL} = \alpha \times L \times R(\theta) \tag{5.4}$$

$$u_{E,MACHINE\,TOOL} = T\Delta \times L \times R(\alpha) \tag{5.5}$$

Similarly to the machine tool, the measurement device uncertainty ($u_{M,DEVICE}$) will also need to be compensated due to the temperature, as well as the uncertainty due to the coefficient of thermal expansion ($u_{E,DEVICE}$). Equation 5.6 describes how to calculate the device uncertainty due to the temperature measurement, and Equation 5.7 shows how to calculate the uncertainty due to the coefficient of thermal expansion of the device. In some cases, such as when using a laser interferometer, the device will automatically compensate for temperature change of the device and machine tool using environmental monitoring. In this particular example it is not necessary to calculate $u_{M,DEVICE\,LASER}$ and $u_{E,DEVICE\,LASER}$ as they are automatically compensated for by the device. However, it is worth noting that the compensation will have uncertainty, and consideration should be taken to include this uncertainty, no matter how small.

$$u_{M,DEVICE\,LASER} = \alpha \times L \times R(\theta) \tag{5.6}$$

$$u_{E,DEVICE\,LASER} = T\Delta \times L \times R(\alpha) \tag{5.7}$$

During measurement, the temperature of the environment might change resulting in the possibility of instrument and machine tool drift that influences the measurement result. An experiment can be performed to monitor drift by leaving the instrument active for a period of time equal to the length of time for the test to identify any change in the value relative to temperature. From this, Equation 5.8 can be used to determine the uncertainty due to environmental variation u_{EVE}. A downside of this approach is that it doubles the length of time to perform the measurement.

$$u_{EVE} = \frac{E_{VE}}{2\sqrt{3}} \tag{5.8}$$

Equation 5.9 shows the calculation necessary to compute the estimated uncertainty for one measurement. However, many measurements will be made when calibrating a machine tool, making the combined uncertainty, u, the sum of all $u(c)$.

$$u_c = \sqrt{\begin{array}{l} u^2_{DEVICE\ LASER} + u^2_{MISALIGNMENT} + u^2_{M,MACHINETOOL} \\ + u^2_{M,DEVICE} + u^2_{E,MACHINETOOL} + u^2_{E,DEVICE} + u^2_{EVE} \end{array}} \tag{5.9}$$

The expanded uncertainty U can then be calculated by using Equation 5.10 where the combined standard uncertainty is multiplied by the coverage factor k.

$$U = k \times u_c \tag{5.10}$$

5.2 Increasing Numerical Expressiveness of PDDL

PDDL provides access to four arithmetic operators (`+, -, /, *`) [47]. These operators can be used to implement numeric preconditions, effects and the duration statement in PDDL actions. The expressive power of PDDL allows for the modelling of many complex real-world problems that have significantly motivated planner development. However, planning problems that have strong numeric requirements are difficult to implement using the standard set of arithmetic operators, which is true when attempting to implement the estimated uncertainty of measurement equations. In the section we consider a possible solution of implementing the square root function in PDDL2.1 by using the Babylonian method.

5.2.1 Initial Babylonian Encoding Solution

In this section a method that provides an approximation to the square root function in PDDL2.1 is provided. The technique commonly known as the Babylonian Method [91] is used to calculate approximate square roots. The Babylonian method is an iterative approach to calculating the square root, and is therefore suitable for implementation where access to a square root function is not available but some method of recursion is. This method can be described by the following equations:

$$
\begin{aligned}
x_0 &\approx \sqrt{S} \\
x_{n+1} &= \frac{1}{2}(x_n + \frac{S}{x_n}) \\
\sqrt{S} &= \lim_{n\to\infty} x_n
\end{aligned}
\tag{5.11}
$$

The Babylonian method is iterative and for any value of n, that can represent x_n in terms of S and x_0 which are both known. For example:

$$
\begin{aligned}
x_1 &= \frac{1}{2}(x_0 + \frac{S}{x_0}) \\
x_2 &= \frac{1}{2}(\frac{1}{2}(x_0 + \frac{S}{x_0}) + \frac{S}{\frac{1}{2}(x_0 + \frac{S}{x_0})})
\end{aligned}
\tag{5.12}
$$

```
(:durative-action calculate-sqrt
     :duration(= ?duration 1)
     :condition
     (and
         (at start (<=(current-step)(number)))
         (at start (start))
     )
     :effect
     (and
         (at start(update(calculated-sqrt)
         (*(+(calculated-sqrt)(/(number)
            (calculated-sqrt)))0.5)))
            (at end(increase(current-step)1))
     )
)
```

FIGURE 5.1: Partial PDDL encoding to calculate the square root using the Babylonian method

In the absence of the square root function when using a PDDL2.1 capable planner, it is possible to encode a method which can enumerate the square root for a given value. In Figure 5.1 a PDDL encoding is shown that uses the Babylonian method that is shown in Equation 5.11 to calculate the square root. In the encoding the `calculated-sqrt` fluent as the current x_n value and the `number` is the S value that the square root is required calculating. This method will calculate the square root for a given number S, however, the number of iterations i required is equal to S. This is a large limitation because it is computationally exhaustive and not optimal. It is often the case that a very close approximation will be produced within only a few iterations. However, imposing an iteration limit will depend upon the application and the desired level of accuracy.

5.2.2 Experimental Analysis of Initial Encoding

Iteration(i)	Result (x_n)	Difference ($x_n - x_{n+1}$)
1	5.50000000	4.50000000
2	3.65909091	1.84090909
3	3.19600508	0.46308583
4	3.16245562	0.03354946
5	3.16227767	0.00017796
6	**3.16227766**	0.00000001
7	3.16227766	0.00000000
8	3.16227766	0.00000000
9	3.16227766	0.00000000
10	3.16227766	0.00000000

TABLE 5.1: Results of using the Babylonian method in PDDL 2.1 to calculate the square root of 10.

An example can be seen in Table 5.1 where the square root for the number 10 is calculated. In the table it is noticeable that after six iterations, the Babylonian method correctly calculates the square root of 3.16227766 to eight decimal places. In the first iterations $x_0 = 10$. It is also noticeable that the difference between the current calculation x_{n+1} and the result with the previous calculation x_n converges to zero to eight decimal places after six iterations. This shows that the correct square root was calculated in the sixth iteration. However, if accuracy to only two decimal places was required, it would be possible to stop after four iterations.

Figure 5.2 shows the difference between x_n and x_{n+1} per iteration i when calculating the square root for the value 1000 using the Babylonian method. The experiment was performed using the PDDL encoding as seen in Figure 5.1. Although it is noticeable that it only took eleven iterations to converge, the LPG-td planner performed 1000 iterations, the majority of which are not required. Solving this problem alone took 4.66 seconds, but could be significantly reduced if not all iterations were executed.

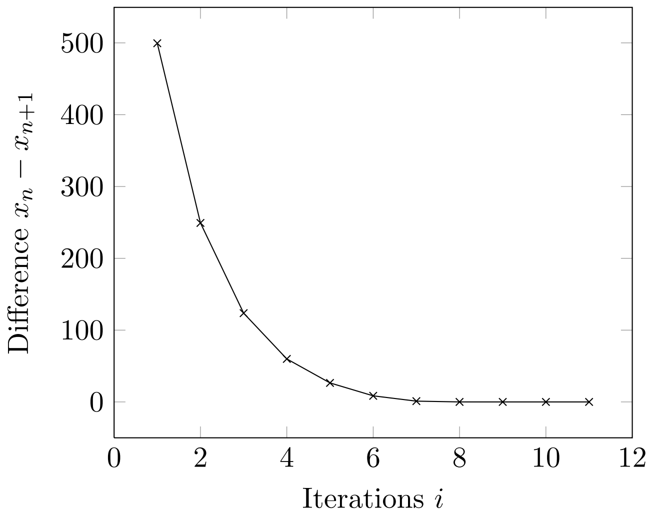


FIGURE 5.2: Results from calculating the square root for the number 1000 using the Babylonian method

Combining this implementation with other models would have an adverse affect on plan generation and quality. This shows that even if it is always possible to encode an algorithm that could perform the desired mathematical function in PDDL, it could be regarded as excessive modelling effort and planning computation.

5.2.3 Pre-processor Solution

An alternative solution is to use a preprocessor to generate PDDL approximations of the square root function as one equation, reducing the requirement to iteratively perform the Babylonian algorithm using a recursive PDDL action. Therefore, keeping the completeness of being able to calculate the square root but removing the redundant computation overheads as a result of excess planning.

Algorithm 1 The Approximate Square Root Function

Require: a_0 : initial guess
Require: S : formula input
Require: i : required depth

function SqrtGenerate(a_0,S,i)
  if $i = 0$ **then**
    return a_0
  else
    $s_{i-1} \leftarrow$ SqrtGenerate($a_0, S, i-1$)
    return '(/(+' s_{i-1}'(/'S s_{i-1}'))2)'
  end if
end function

Algorithm 1 shows the method used inside the preprocessor to generate PDDL approximations of the square root function. Line 6 performs the main computation, using string concatenation to construct the output formula. The algorithm depends on three parameters: S, the input formula; i the required depth (i.e. generating a formula equivalent to x_i in the Babylonian Method sequence) and a_0, the initial guess. The accuracy of the Babylonian Method is sensitive to the selected values of i and a_0. The Babylonian Method converges quadratically, roughly providing a result twice as accurate for each increment of i. Therefore, an ideal situation is one in which a_0 is as close to $\sqrt{S}$ as possible and i is as high as possible.

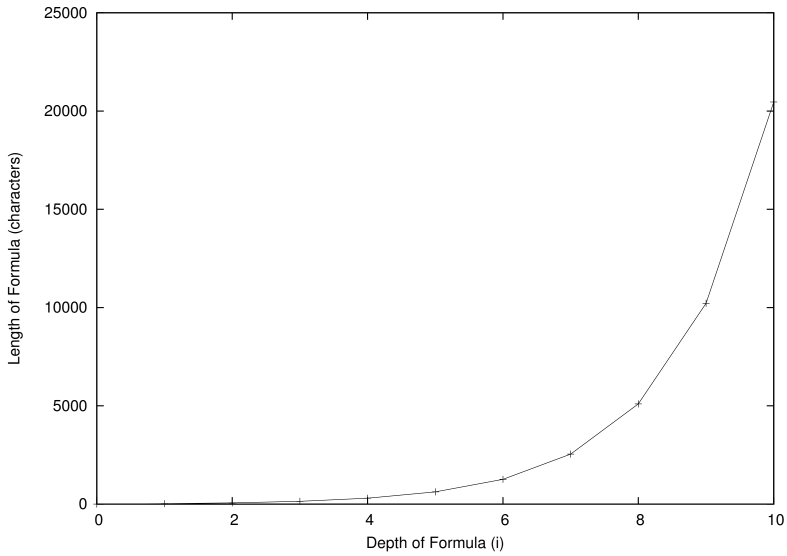


FIGURE 5.3: Rate of growth of the size of the output function as i increases

It is a challenge to satisfy these constraints using a preprocessor method. Despite the fact that the accuracy of the approximation increases quadratically, the size of the formula increases quadratically as i is increased (see Figure 5.3). There is, therefore, a trade-off between the accuracy and the size of formula. As the preprocessor stage is static (i.e. occurs before planning) the input to the formula is likely to be unknown. Therefore, selecting the best value for a_0 is also challenging. However, given a range of expected input values, it is possible to calculate an average square root over the expected range. This value can then be used as a_0. It is not always possible to gain the desired accuracy by choosing a single x_0 value.

5.2.3.1 Algorithmic Properties

Using an approximation method to calculate square roots has a negative impact on algorithmic properties. With respect to the definition of the domain with the square root to a certain precision, using an approximation renders the planner incomplete, unsound and removes optimality guarantees. This arises because of the error introduced by the numerical approximation. The significant for this applications is that any error due to approximation will give an incorrect estimate of the uncertainty of measurement and could result in the production of an non-optimal plan. It should be noted that all numbers in computers are represented using floating point number representation [92]. Floating points are designed to encode a wide range of numbers using a finite representation. However, this leads to approximation which ultimately results in error. However, this section is to describe how the error from using the Babylonian approximation method may affect a planning problem. This is done by demonstrating the relative error that arises from the approximation.

	S				
i	50	100	500	1000	5000
1	0.0607	0.2500	1.3479	2.2413	6.1064
2	0.0017	0.0250	0.3869	0.7749	2.6236
3	0.0000	0.0003	0.0540	0.1692	0.9498
4	0.0000	0.0000	0.0014	0.0122	0.2313
5	0.0000	0.0000	0.0000	0.0001	0.0217
6	0.0000	0.0000	0.0000	0.0000	0.0002
7	0.0000	0.0000	0.0000	0.0000	0.0000

TABLE 5.2: The relative error of the Babylonian Method with initial guess x_0 of 5.00.

For this demonstration, an arbitrary value for x_0 of 5.0 has been selected. Table 5.2 shows the relative error for various combinations of values of i and S where $x_0 = 5.0$. It is clear that without selecting appropriate values for x_0 and i the approximation could lead to inaccuracies.

Soundness Figure 5.4 a) shows a problem that has a solution if the approximation with $x_0 = 5$ and $i = 2$ is used. This approximates $\sqrt{5000} \approx 256.23$ (thus satisfying the goal) whereas $\sqrt{5000} = 70.71$ and there is no solution. Therefore the approximation is unsound.

```
;; a) Soundness
(:init (= (x) 5000))
(:goal (and (< (x) 1000) (> (x) 100)))

;; b) Completeness
(:init (= (x) 5000))
(:goal (and (< (x) 75) (> (x) 70)))

;; c) Optimality
(:init (= (x) 5000))
(:goal (< (x) 100))
```

FIGURE 5.4: Three PDDL problems that demonstrate how the approximation leads to unsoundness, lack of completeness and sub-optimality.

Completeness Repeated application of the `apply-square-root` action provides the sequence of values of x:$(5000, 256.23, 18.62, 4.31)$. This never generates the goal state of Figure 5.4 b) that is possible in the true domain ($\sqrt{x} = 70.71$), hence the approximation leads to incomplete models.

Optimality Figure 5.4 c) shows a problem that generates sub-optimal solutions with respect to the true domain. As mentioned previously, a single application of the `apply-square-root` action should leave $x = 70.71$, thus satisfying the goal. Using the approximation in order to satisfy the goal using the approximation requires more than the single action.

Despite these theoretical problems, there are values of i and x_0 for which the approximation is clearly useful. In the following section, ways in which it is possible to reliably produce good approximations within a given set of numeric bounds are discussed.

	S				
i	50	100	500	1000	5000
1	1.1296	0.5940	0.2680	0.0060	0.4475
2	0.2996	0.1107	0.0283	0.0000	0.0692
3	0.0345	0.0055	0.0004	0.0000	0.0022
4	0.0006	0.0000	0.0000	0.0000	0.0000
5	0.0000	0.0000	0.0000	0.0000	0.0000
6	0.0000	0.0000	0.0000	0.0000	0.0000
7	0.0000	0.0000	0.0000	0.0000	0.0000

TABLE 5.3: The relative error of the Babylonian Method with initial guess x_0 of 28.53

5.2.3.2 Selecting x_0 Values

With prior knowledge about the distribution of S values that for which computing the square root is required, it is possible to pre-compute a useful value of x_0. This is demonstrated by finding a more appropriate value of x_0 for the distribution of S values from Table 5.3. To do this, the mean root value from the S values is acquired, which in this case is $x_0 = 28.53$. With this selection of x_0, the depth of computation required in order to gain accuracy of two decimal places is reduced from $i = 6$ to $i = 4$. In other words, from a formula length of about 1500 characters down to 300. For many applications, even the single digit accuracy gained when $i = 3$ may be sufficient.

It may be that even using this method for finding a good value for x_0 is insufficient. In this case, another strategy based on partitioning the target range into several sub-ranges, finding several values for x_0 that can then be used to guarantee a certain level of accuracy is used. In order to do this, either conditional effects can be used or actions can be split into many actions. We use Algorithm 2 to do discover these ranges. The function returns a set of tuples with the structure $(low, high, x_0)$ where low and $high$ represent the lower and upper bounds of the interval; x_0 represents an initial guess for this interval guaranteeing relative error lower than a specified level for a specified formula depth. The algorithm works by growing a candidate range until it is too large to satisfy the relative error constraint, at which point the previous range is accepted and a new search begins for the next range.

Algorithm 2 The Generate Ranges Function

Require: S : formula input
Require: i : required depth
Require: e : maximum allowable relative error
Require: $[lb, ub]$: input range

function GENERATERANGES(S,i,e,$[lb, ub]$)
  $ranges = \emptyset$
  $low = lb$
  $high = lb + 1$
  while $high < ub$ **do**
    $x_0 \leftarrow (\sqrt{low} + \sqrt{high})/2$
    if RelError $(S, i, x_0, low, high) \geq e$ **then**
      $ranges \leftarrow ranges \cup \{(low, high - 1, x_0)\}$
      $low = high = high - 1$
    else
      $high = high + 1$
    end if
  end while
  $ranges \leftarrow ranges \cup \{(low, high - 1, x_0)\}$
  return $ranges$
end function

5.3 PDDL Implementation

The developed PDDL domain is an extension to the temporal optimisation version presented in Section 4.3. In the temporal optimisation model, machine tool downtime is reduced by using durative actions where their durations are determined by the time taken to set-up, measure, remove, adjust-position and adjust-error using the specific piece of instrumentation. To encode the measurement uncertainty problem, attention has been focused on the "measure action". In the extended mode, the equations necessary to estimate the uncertainty for linear positioning deviation when measuring using a laser interferometer have been encoded. Table 5.4 shows a cross-reference between the values required in the formula and the PDDL numeric fluents.

Symbol	Equation	Fluent
L	5.1, 5.2, 5.3, 5.4, 5.5, 5.6, 5.7	`(length-to-measure ?ax-axis ?er-error)`
$misalignment$	5.2, 5.3	`(M-A ?in-instrument)`
α	5.4, 5.6	`(T-E-C ?ax-axis)`
$R(\theta)$	5.4, 5.6	`(T-D-M ?ax-axis)`
$T\Delta$	5.5, 5.7	`(D-20-C ?ax-axis)`
$R(\alpha)$	5.5, 5.7	`(D-E-C ?in-instrument)`

TABLE 5.4: The numeric fluents required to implement the uncertainty calculations.

To implement the uncertainty calculations, the "measure" action in the domain requires modification to include the formula. This is achieved by using the standard set of arithmetic operators and assigning the result to a fluent. The equation that requires calculation is the square root and is denoted by (`sqrt(xx ?in ?er ?ax)`) (as seen in Figure 5.5) which will be replaced by its approximation, generated by the Babylonian Method. In the model, part of the calculation is performed at the start of the action, with the square root being calculated at the end of the action to keep the expanded function as small as possible.

```
(:durative-action measure
     :parameters (?in - instrument ?er - error ?ax - axis)
     :duration(= ?duration (measurement-time ?in ?er ?ax))
     :condition
     (and
         (over all (set-up-at ?ax ?er ?in))
         (at start (ready-to-measure ?ax ?er ?in))
         (at start (<=(length-to-measure ?ax ?er)(working-range ?in)))
         (at start (<=(amount-measured ?ax ?er)(length-to-measure ?ax ?er)))
         (over all (working-day))
     )
     :effect
     (and
         (at end (not(ready-to-measure ?ax ?er ?in)))
         (at end (repos-available ?ax ?er ?in))
         (at end (increase(amount-measured ?ax ?er) (*(working-range ?in)1)))
         (at start (assign (xx ?in ?er ?ax)
         ;;calculate u_device (Equation 5.1)
         (+(*(/(*(U-C ?in)(length-to-measure ?ax ?er))(k))
             (/(*(U-C ?in)(length-to-measure ?ax ?er))(k)))
         ;;calculate u_misalignment (Equation 5.3)
         (+(*(/(M-A ?in)0.6)
             (/(M-A ?in)0.6))
         ;;calculate u_m_machine tool (Equation 5.4)
         (+(*(*(*(T-E-C ?ax)(/(length-to-measure ?ax ?er)1000))(T-D-M ?ax))
             (*(*(T-E-C ?ax)(/(length-to-measure ?ax ?er)1000))(T-D-M ?ax)))
         ;;calculate u_e_machine tool (Equation 5.5)
         (+(*(*(*(D-20-C ?ax)(/(length-to-measure ?ax ?er)1000))(D-E-C ?in))
             (*(*(D-20-C ?ax)(/(length-to-measure ?ax ?er)1000))(D-E-C ?in)))
         ;;calculate u_eve (Equation 5.8)
         (*(*(E-V)0.6)
             (*(E-V)0.6))))))))
     (at end (increase (u-m) (sqrt (xx ?in ?er ?ax)))))
```

FIGURE 5.5: PDDL measure action for estimating positional deviation measurement uncertainty

5.3.1 Experimental Analysis

The current implementation of the uncertainty estimation calculation allow for estimation to take place during the planning process. This makes it possible for the planner to search for an optimal plan that reduces uncertainty.

The studied problem instances comprise two different three-axis machine tools with twenty one geometric error components that require calibrating (six-degrees-of-freedom plus one non-orthogonal error per axis pair). The machine tools are different in terms of axis travel lengths and kinematic configuration. There are three problem instances for each machine tool. The first representing a baseline instance, whereas the second one models calibration by a more experienced engineer. The third instance represents an experienced engineer with a wider range of measurement equipment.

The equations in the domain are currently for estimating the measurement uncertainty for positional deviation when measuring using a laser interferometer. However, the problem instances contain many other different errors to measure using different techniques. In this initial experiment, all measurements use the equations as seen in Figure 5.5. Although this implementation will produce incorrect results in terms of estimated uncertainty, it allows for the scalability and reliability of the planner to be tested as well as the feasibility of the approach.

In order to evaluate the models, the LPG-td planner is used on the preprocessed PDDL domain. LPG-td is used for the experimental analysis because it was identified as being the best planner to use for the temporal domain, and the experimental domain is an extension of said domain. Table 5.5 shows the results of the experiments. Three distinct experiments with LPG are performed, each set having a different metric, selected from the following metrics:

1. U - `(:metric minimize (u-m))`
2. $T(min)$ - `(:metric minimize (total-time))`
3. $\sqrt{U.T}$ - `(:metric minimize (sqrt(*(total-time)(u-m)))`

	Metric: Uncertainty		$Metric: Time$		Metric: $\sqrt{U.T}$	
Instance	U(µm)	T	U(µm)	T	U(µm)	T
3A1A	463	34:00	545	33:00	509	33:30
3A1B	469	28:41	558	27:30	539	28:27
3A1C	498	33:52	515	27:39	561	29:48
3A2A	417	34:00	475	33:30	451	33:20
3A2B	422	28:41	484	28:27	441	28:23
3A2C	436	33:33	444	39:48	453	33:10

Table 5.5: Results of empirical analysis

5.3.1.1 Context

The purpose of this experimental analysis is to examine and understand the performance of LPG-td when searching for solutions with the developed PDDL model. The developed model includes the necessary numerics, equations and Babylonian method of calculating the square root when estimating the uncertainty of measurement. In the analysis, three different metrics are used with the problem instances. The first is the accumulative estimated uncertainty of measurement, the second is the downtime of the machine tool, and the third is the average of both estimated uncertainty of measurement and downtime. These results allow for conclusions to be made regarding the automatic construction of calibration plans when using single and multi-objective optimisation.

5.3.1.2 Results

In Table 5.5, the summed uncertainties for each measurement and the time in hours and minutes is shown for the three different metrics. The first two of these are self-explanatory. The third specifies a metric that attempts to compromise between the two objectives by taking the arithmetic mean. Note the requirement to use the square root function in this metric function, demonstrating wider applicability of the preprocessor. A modified version of LPG-td was used in the experiments to allow longer formulae. This modification is performed easily by adjusting the parameters in the configuration file before recompiling the source code. For the measure action and the metric function, the square root preprocessor provides relative error ≤ 0.0001.

LPG-td behaves consistently: uncertainty is lowest when time is not taken into account, and vice versa. LPG-td solves the multi-objective optimisation problem that have been set satisfactorily. In almost all cases, the $\sqrt{U.T}$ plan reduces the time taken in the U plan whilst also reducing the uncertainty in the T plan, thus providing a compromise. It is important to consider the trade-off between minimising the plan duration and uncertainty estimation as in reality this is a pragmatic decision that both a machine tool owner and calibration engineer have to take.

5.3.2 Critique of Model

In this section, the extension of the temporal model has demonstrated that it is possible to extend the machine tool calibration domain to reduce estimated measurement uncertainty. The fact that the estimated uncertainty can be reduced by careful planning is well established throughout the metrology community, although less known by the machine tool maintenance community. However, the novel method of using automated planning to reduce the estimated uncertainty will be a welcomed addition as it can be applied to a whole range of complex measurement planning problems. Previous approaches aim to estimate and minimise uncertainty for each test, whereas by using planning technology the uncertainty can be reduced for the whole calibration plan. In creating the model, a robust preprocessor that provides a square root function that can be used in a generic PDDL domain to specified levels of relative error has been developed.

This initial solution highlighted that it is not possible to implement the square root function using PDDL2.2 arithmetic operators. The Babylonian method was implemented in PDDL as one formula that is produced using a preprocessor. However, the preprocessor has limitations regarding estimating the correct formula depth: too small and the correct square root will not be calculated because the formula will end before convergence; too large and the formula will result in excessive computation as convergence is reached before the formula finishes.

The implemented equations currently only estimate the uncertainty of measurement for linear deviation when using a laser interferometer. There is a vast range of potential measurements that can be used when calibrating a machine tool, and these equations are not suitable for a generic approach. This highlights the requirement for a generic, extensible method that can estimate the uncertainty of measurement for many different measurement techniques and instrumentation. It is possible that additional PDDL actions can be implemented to suit each required measurement, however, this goes away from the fundamental philosophy of producing a generic, extensible method of automatically constructing calibration plans.

As identified in Section 3.8, some planners are unable to handle non-linear effects. This means that calculations like $u^2_{DEVICE\ LASER}$ can not be encoded. However, LPG-td that is being used can handle non-linear effects. Additionally, PDDL does not have any support for the square root function. The absence of both these functions requires either modifying the planning tool to support these features or implement a post-processor. The post-processor is the better of the two solutions as it still allows for domain-independence, increasing the range of planners than can solve the problem. Post-processing provides a solution to complete the estimated uncertainty equations and determine the actual estimated uncertainty, but it will result in a cumbersome solution. This is because the contribution from each measurement towards the metric value m needs to be identified. If the planner could handle non-linear calculations like $u^2_{DEVICE\ LASER}$ it would only be necessary to calculate the square root of m. This would not affect the planner's ability to find an optimal solution as the square root function is a monotonic function, meaning that minimizing m would have the same affect as minimising $\sqrt{m}$.

This approach to calculating the square root using PDDL2.1 provides a novel contribution, allowing for the inclusion of a square root calculating on a PDDL domain. This implementation shows that is is possibility to use PDDL to closely model real-world domains with strong numeric properties, and provide a useful result for the end user. However, if the end user does not need to know the numeric result, then studying the aspects of the domain to model how the metric can be minimised could result in a less complex domain model.

5.4 Measurement Uncertainty Due to Plan Order

The philosophy behind the investigation performed is that, rather than calculating the estimated uncertainty for each individual measurement, it might be more efficient to consider only the contributors that affect the estimated uncertainty due to scheduling. This means that it is only necessary to model aspects that cause the estimated uncertainty of measurement to change, thus simplifying the domain model.

5.4.1 Factors that Affect the Uncertainty of Measurement

There are many potential contributors that affect the uncertainty of measurement. However, when automatically constructing a calibration plan, the aim is to select the most suitable instrumentation and measurement technique that has the lowest estimated uncertainty. In addition, the estimated uncertainty should take into consideration the changing environmental data, and where possible, schedule the measurements to take place where the effect of temperature on the estimated uncertainty is at its lowest.

The following list provides the factors that affect the estimated uncertainty of the calibration plan, and suggests how they can be optimised.

- Measurement instrumentation having the lowest estimated uncertainty of measurement. Where possible, selecting instrumentation with the lowest uncertainty will reduce the overall estimated uncertainty of measurement.
- The change in environmental temperature throughout the duration of a measurement can significantly increase the uncertainty of measurement. When possible, the measurement should be scheduled to take place where the temperature is stable.
- When considering inter-related measurements, the change in environmental temperature between their measurement can significantly increase the uncertainty. During planning, it is important to schedule interrelated measurements where the change in environmental temperature is at its lowest.
- Allowing the equipment to correctly stabilise in the environment before the measurement can reduce the uncertainty due to coefficient of thermal expansion and self-learning.

5.4.2 Domain Modelling

The previously developed temporal model as described in Section 4.3 and Section 5.1 are extended to make it applicable to a wider range of measurements. Figure 5.6 shows the functional flow between the PDDL actions within the newly extended temporal model. In the figure, durative actions are represented using a circle with a solid line, whereas

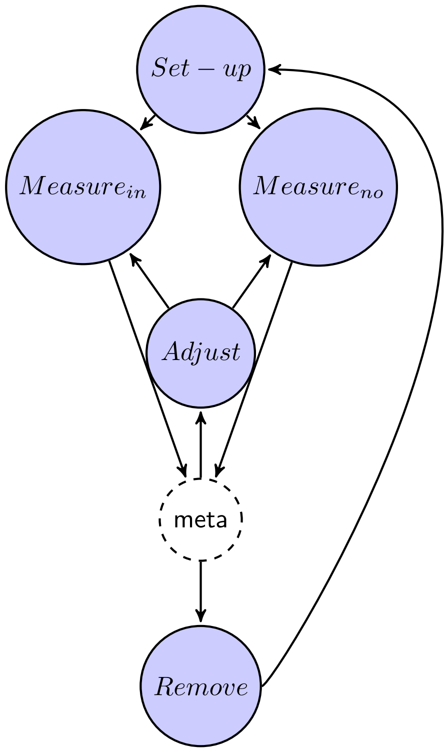


FIGURE 5.6: Illustration showing the PDDL actions and their functional flow.

the meta, non-durative action is represented with a dashed line. From Figure 5.6 it is noticeable that the measurement action has been split up into two different actions and a non-durative `meta` action has been added to implement parts of the equation. The following list details the extension of the measurement action into two actions and the addition of a meta action:

Measure$_{no}$: The measurement action represents a measurement where no consideration is taken for any influencing errors.

Measure$_{in}$: Conversely, this measurement action represents a measurement where consideration is taken for any influencing errors.

Meta : Zero cost meta action required to encode temperature information and uncertainty equations.

In the temporal model, the cost of each action is the time taken to perform that specific task. Using this model will produce a calibration plan, indicating the ordering of the measurements and the time taken to perform each test. In addition to these actions, one instantaneous, zero-cost meta-action has been added after the measurement action to implement the temperature dependent uncertainty equations. The motivation behind the addition of the meta action is because in the model the temperature change throughout the measurement procedure is required. Using `at start`, `over all` and `at end` semantics of a PDDL durative action, it is possible to acquire the temperature at the start and at the end of the measurement action. From these two temperatures, the deviation throughout the measurement action can be calculated. However, implementing the uncertainty estimation as an `at end` effect will not allow the use of the other fluents updated as `at end` effect. Therefore, not allowing access to the change in environment temperature. The chosen solution is to implement a instantaneous meta action.

5.4.2.1 Uncertainty Contributors

The developed model is encoded in PDDL 2.1. This is because of the use of numbers, time, and durative actions [47]. Numeric fluents are especially important for modelling uncertainty of measurement as they provide the contributors. For example, a device's uncertainty ($U_{DEVICE\ LASER}$) can be represented in PDDL as `(=(device-u ?i - instrument)0.001)` where the instrument object `?i` has the value of 0.001.

5.4.2.2 Temperature Profile

In PDDL2.1 it is not possible simply to represent predictable continuous, non-linear numeric change. More specifically, it is not possible to represent the continuous temperature change throughout the calibration process. This presents the challenge of how to optimise the sequence of measurements while considering temperature. The solution implemented in the model involves discretizing the continuous temperature change into sub-profiles of linear continuous change.

This can be achieved by using Algorithm 3 which iterates over the temperature data looking for a difference in temperature greater than a given sensitivity. This allows the temperature profile to be discretized into a set of sub-profiles. An example can be seen in Figure 5.7 where the environmental temperature profile (difference from 20°C) for a forty-eight hour period is shown (Monday and Tuesday). The reason that the second twenty-four hour cycle is greater than the first is due to the cooling effect of the weekend where no production is taking place still being evident throughout the Monday period.

Algorithm 3 The Discretized Temperature Data Function. This converts a continuous temperature profile into discrete, sub-profiles

Require: Initial sensitivity s

Require: Ordered pair of timestamps and temperature data $T = (t_1, d_1), (t_2, d_2), (\ldots), (t_n, d_n)$ where t_n is the time stamp and d_n is the temperature data

Require: Set of ordered temperature profiles $P = (t_1, p_1), (t_2, p_2), (\ldots), (t_n, p_n)$ where p_n is the rate-of-change in °C per minute

function DiscretizeTemp(s,T,P)
  $i \leftarrow 0$
  $td_p \leftarrow 0$
  while $i <= Size(T)$ **do**
    $td = |T(d_i) - T(d_{i+1})|$
    if $(td > td_p)\&(td >= s)$ **then**
      $md = T(t_i) - T(t_{i+1})$
      $rate = \frac{td}{md}$
      $P \leftarrow (t_{i+1}, rate)$
      $td_p = td$
      $i++$
    end if
  end while
return P
end function

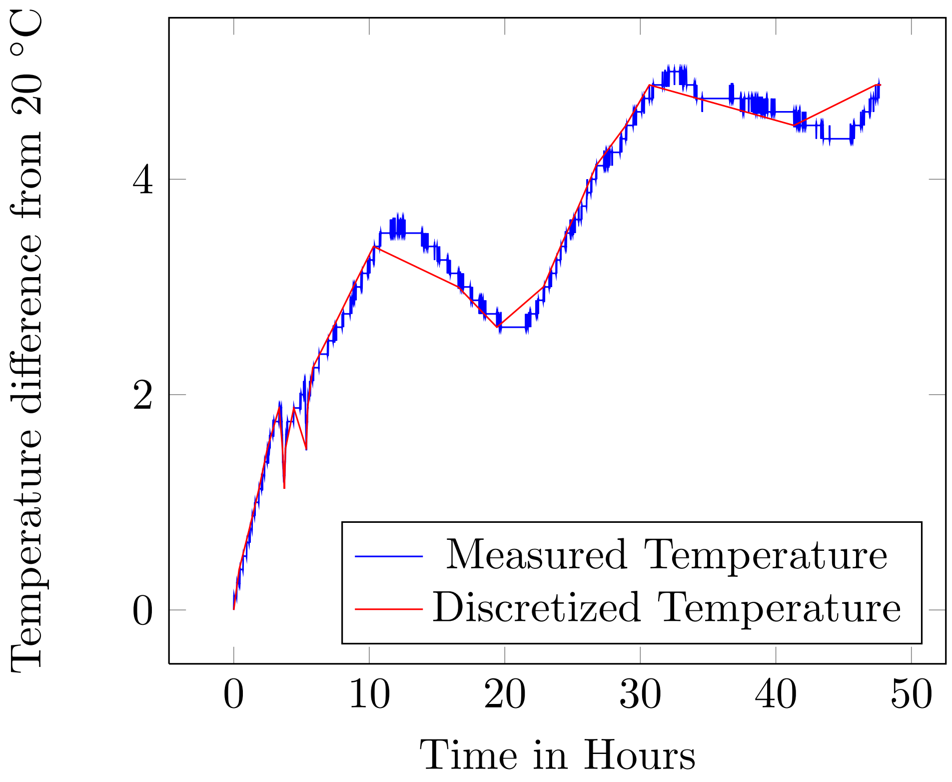


Figure 5.7: Graph showing both the original and discretized temperature profile.

```
(:durative-action temp-profile1
         :duration(= ?duration 42.0)
         :condition
            (and (at start (start1)))
         :effect
            (and (at start (assign (rate)0.00595))
            (at end (not(start1)))
            (at end (start2))
            (at start (clip-started)))
     )
)
```

FIGURE 5.8: Durative actions that represents the temperature sub-profile, p_1, where the duration is $t_1 = 42$.

To model these sub-profiles in the PDDL model, they are represented as predetermined exogenous effects. To increase the range of planners that can be used to solve the problem, the model produced in this work is encoded in PDDL2.2 where TILs are introduced [49], providing a mechanism to represent predetermined exogenous effects. However, representing the temperature sub profiles using TILs complicates the plan, making it unsolvable by the current state-of-the-art. The solution is to represent predetermined exogenous effects is by clipping durative actions together (Section 3.7.1).Therefore, keeping the domain encoded in PDDL2.1. An example durative action, d_1, that represents a sub-profile, p_1, can be seen in Figure 5.8 where the duration $t_1 = 42$. This durative action shows how the update of the temperature is performed as an `at start` effect and will be the current rate for the duration of the durative action.

Figure 5.9 illustrates how the greatest deviation from 20°C ($T\Delta$) throughout the measurement and meta action is encoded. All temperature dependent aspects of the equation are contained within the meta action. This is to allow access to the temperature rate at the start of the measurement action, r_1, and in the proceeding instantaneous meta action, r_2. Therefore, in the meta action it is possible to can calculate the rate of change based on two rates of change, r_1 and r_2, and the time, Δt, that the measurement requires.

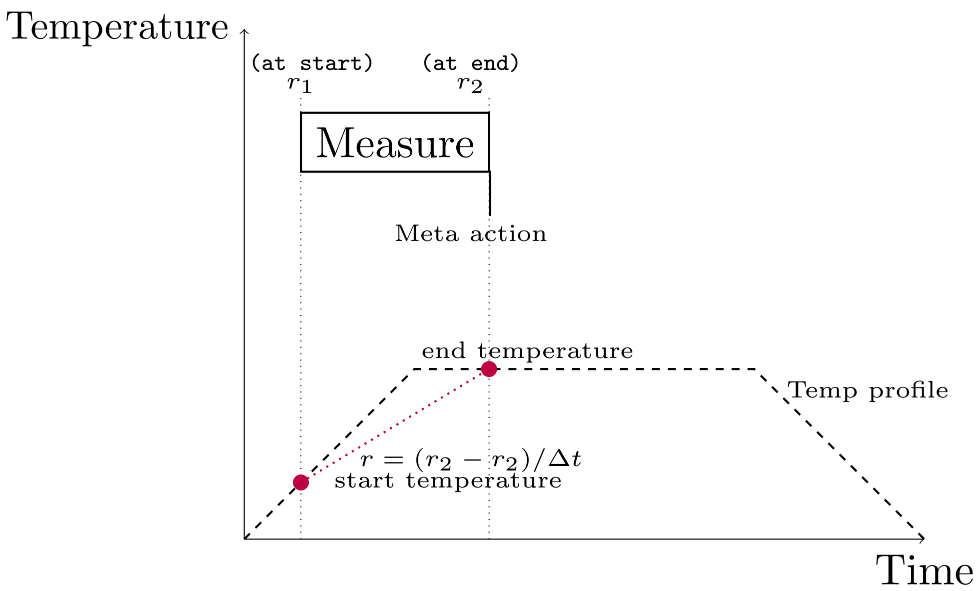


FIGURE 5.9: Illustrating how the meta action and the measure durative action interact to calculate the current environmental temperature.

5.4.2.3 Uncertainty Equations

As described earlier in Section 5.4.2.1, numeric fluents will be used to represent the uncertainty contributors. It is then possible, using the binary operators provided in the PDDL language (`+,-,/,*`) and the update functions for numeric fluents (`assign, increase, decrease, scale-up, scale-down`) to implement the uncertainty equations.

Implementing equations where the result is influenced by other measurements is also encoded in the PDDL using fluents. For example, Figure 5.10 shows the calculation for the non-orthogonal error measurement using a granite square and a short range displacement transducer described in Section 2.3.4 where the uncertainty is influenced by the two straightness errors. In the model, this is encoded by assigning two fluents `(error-val ?ax ?e1))` and `(error-val ?ax ?e2))` the maximum permissible straightness error in the PDDL initial state description. This fluent will then be updated once the measurement estimation has been performed. The planner will then schedule the measurements to reduce the effect of the contributing uncertainty. Therefore, this shows how the uncertainty can be reduced due to the ordering of the plan.

Figure 5.11 shows the partial PDDL encoding for the estimation calculations used when

```
(at start(assign(temp-u)
        ;calculate u_device using the length to measure.  Equation 5.1
             (+(*(/(k_value ?in)(*(u_calib ?in)(length-to-measure ?ax ?er)))
                  (/(k_value ?in)(*(u_calib ?in)(length-to-measure ?ax ?er))))
        ;calculate u_misalignment.  Equation 5.2
             (+(*(/(+(u_misalignment ?in)(u_misalignment ?in))(2sqr3))
                  (/(+(u_misalignment ?in)(u_misalignment ?in))(2sqr3)))
        ;calculate u_error contributors.
             (+(*(/(+(error-val ?ax ?e1)(error-val ?ax ?e2))(2sqr3))
                  (/(+(error-val ?ax ?e1)(error-val ?ax ?e2))(2sqr3)))
        ;calculate u_m_machine tool.  Equation 5.4
             (+(*(*(u_t-m-d)(*(length-to-measure ?ax ?er)(u_m-d)))
                  (*(u_t-m-d)(*(length-to-measure ?ax ?er)(u_m-d))))
        ;calculate u_m_device.  Equation 5.6
             (+(*(*(u_t-m-d)(*(length-to-measure ?ax ?er)(u_m-d)))
                  (*(u_t-m-d)(*(length-to-measure ?ax ?er)(u_m-d))))
        ;calculate u_eve.  Equation 5.8
             (*(/(u_eve)(2sqr3))(/(u_eve)(2sqr3)))))))
             )
        )
```

FIGURE 5.10: PDDL code showing part of the `measure-influence` action.

```
(decrease (error-val ?ax ?er)(-(error-val ?ax ?er)
        (+(temp-u)
        ;calculate u_e_machine tool.  Equation 5.5
             (+(*(*(u_t-e-c)(*(length-to-measure ?ax ?er)(rate)))
                  (*(u_t-e-c)(*(length-to-measure ?ax ?er)(rate))))
        ;calculate u_e_device.  Equation 5.7
             (+(*(*(u_d-t-e-c ?in)(*(length-to-measure ?ax ?er)(rate)))
                  (*(u_d-t-e-c ?in)(*(length-to-measure ?ax ?er)(rate))))))
        )
    )
```

FIGURE 5.11: PDDL code showing the `meta` action.

performing a non-orthogonal measurement using a mechanical square and a SRDT (Section 2.3.4). From this PDDL encoding, it is noticeable that when performing the equation to estimated $u_{E,MACHINE\,TOOL}$, the temperature deviation from 20 °C ($T\Delta$) supplied to the equation is the maximum deviation calculated in the PDDL action (`rate`).

5.4.2.4 Search Metric

In the PDDL problem definition it is necessary to provide a search metric. The search metric is used by the planning tool's heuristic function to find the optimal solution. In the temporal model, the search metric `(:metric minimize (total-time))` was used to find the solution that took the least amount of time. The metric used to reduced the uncertainty of measurement is `(:metric minimize (u-c))`, where `u-c` is the fluent used to accumulate the result of estimated uncertainty for each measurement. Therefore, this measurement is the uncertainty due to the order of the plan.

5.4.3 Experimental Analysis

Initial validation of the PDDL model was performed by creating a test-case problem to solve using the developed model. The produced solution can then be analysed to evaluate whether the use of this model along with state-of-the-art domain-independent automated planning can result in intelligent behaviour being exhibited that results in calibration plans with a reduced estimated uncertainty of measurement.

5.4.3.1 Three-axis Case-study

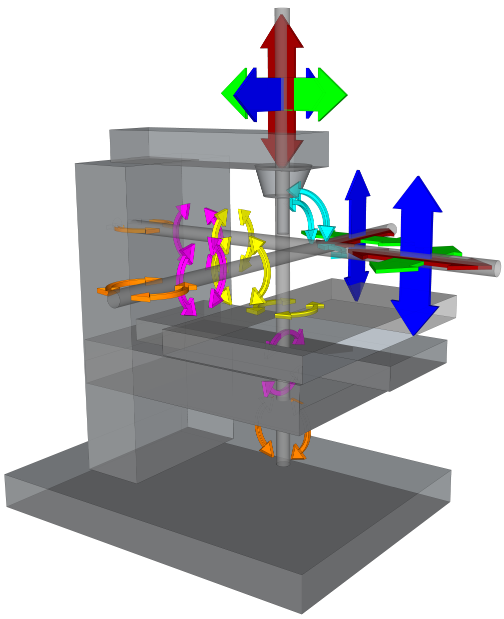


FIGURE 5.12: Three axis machine tool with twenty-one pseudo static geometric errors.

In this experiment, we consider the calibration of a three-axis machine tool. The three-axis machine tool is a cross-table design with two horizontal linear axes (X- and Y-Axis) and a vertical axis (Z-axis). This machine tool has a total of twenty-four pseudo-static geometric test that require measurement during calibration. This total is made up of six-degrees-of-freedom per linear axis, an accuracy and repeatability test, as well as a non-orthogonal error with the nominally perpendicular axes. Figure 5.12 illustrates both the configuration of the machine tool in this example and its twenty-one pseudo-static geometric errors. The discretized temperature profile discussed in Section 5.4.2.2 is used in this case-study.

5.4.3.2 Context

The purpose of this experimental analysis is to examine the structure of an automatically produced three-axis calibration plan. This analysis is to examine the calibration plan and determine whether measurements have been scheduled to reduce the estimated uncertainty of measurement, as well as their fitness for purpose. The analysis will involve analysing the scheduling of measurements in the produce three-axis plan and comparing the optimised uncertainty metric (minimised) with the maximised uncertainty metric. This will highlight the difference in estimated uncertainty of measurement between the best- and worst-case calibration plan.

5.4.3.3 Produced Plan

The produced calibration plan was found in 6 minutes 58 seconds with an uncertainty due to plan order metric of 28 µm. In the remainder of this section, an excerpt taken from the three-axis calibration plan, showing the plan for calibrating the errors in the X-axis direction is discussed. Both the LPG-td planner and the PDDL syntax make it possible to maximise a search metric as well as minimising. Modifying the PDDL problem definition to include `(:metric maximize (u-c))` returns a plan with the metric of 47µm. This shows that there is a significant difference of nearly 20µm between the maximum and minimum uncertainty due to the plan order.

It is expected that the planning tool will schedule the measurement where the temperature difference will have the smallest effect on a measurement. Figure 5.13 show ordering

$\mathbf{E_{AX}}$: angular error around A-axis
$\mathbf{E_{BX}}$: angular error around B-axis
$\mathbf{E_{CX}}$: angular error around C-axis
$\mathbf{E_{YX}}$: straightness error in Z-axis
$\mathbf{E_{ZX}}$: straightness error in Z-axis
$\mathbf{E_{XX}}$: linear positioning error
$\mathbf{E_{C0Y}}$: non-orthogonality to X-axis

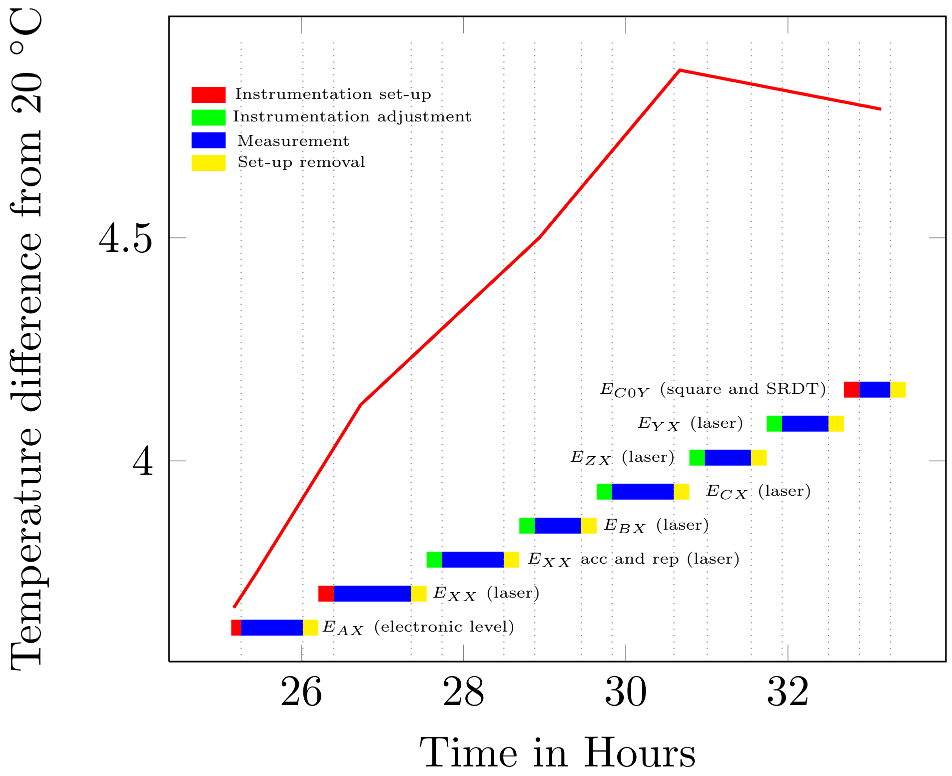


FIGURE 5.13: Graph showing an extract from the discretized temperature profile and an excerpt (errors in X-axis direction) from the produced calibration plan.

of the measurements against time and with respect to the discretized temperature profiles. The red boxes indicate measurement setting up, green boxes for measurements where the instrumentation is adjusted, blue boxes represent the measurement, and yellow boxes represent removal of equipment. It is evident in this figure that interrelated measurements (E_{ZX}, E_{YX}, and E_{C0Y}) are scheduled where the temperature deviation throughout their measurement is at the lowest. Considering the non-orthogonality measurement seen in Section 2.3.4 where two straightness measurements influence the uncertainty of the non-orthogonal measurement. In the plan excerpt it can be seen that the two straightness measurements and the non-orthogonal measurement are scheduled adjacently at a point where the maximum temperature deviation is minimal. This has been achieved by encoding the method so that initially the maximum permissible error is included. The value is subsequently replaced by the actual measurement value plus any change as a result from a change in temperature. This illustrates that the planner can intelligently produced calibration plans to reduce the uncertainty of the calibration plan. The plan excerpt also shows another example of where interrelated measurements are scheduled to reduced the estimated uncertainty. This is the scheduling of the position (E_{XX}), pitch (E_{BX}) and yaw (E_{CX}) measurements. The position error is measured first because positional deviation can affect the measurement of the pitch and yaw errors.

5.4.3.4 Critique of Model

The proposed method overcomes the earlier identified problem that there is insufficient consideration for the uncertainty of measurement due to scheduling of the calibration plan for machine tool calibration. This is achieved by implementing a novel approach to minimising the estimated uncertainty of measurement in automatically produced calibration plans is presented

The challenges of implementing the model have been discussed in detail. Experimental analysis has confirmed that automated planning is a justifiable choice for producing calibration plans that are optimised based on the effect of temperature on the uncertainty of measurement. The significance being the ability to encode this expert knowlsedge in such a way that intelligent algorithms can reason with it and find an optimal solution to the presented problem.

The provided three-axis machine tool case-study has shown that using this novel technique can produce calibration plans where the uncertainty of measurement is minimised. The presented three-axis case-study shows a 58% reduction in the uncertainty of measurement due to the scheduling of the calibration plan where the maximum is 48µm and the minimum is 20µm. Since machining tolerances are often in the order of 20µm, this experiment has proved the importance of the plan order. Although the planner has found a sequence of measurements that are believed to be optimal to reduce the uncertainty of measurement, some consideration of temporal optimisation is lost. Pressures of production mean that machine tool maintenance would often go for the quickest calibration plan and compare the uncertainty of measurement after.

5.5 Temporal and Uncertainty Optimisation

In the previous section, a case study has been performed to examine the uncertainty optimisation model's ability to produce calibration plans that exhibit intelligent planning and scheduling that results in a reduced estimated uncertainty of measurement due to the ordering of the plan. Additionally, in Chapter 4 the precursor to this model was produced and examined to investigate its ability to produce temporally optimised plans. A further extension of the model is to code it to optimise the multi-objective requirement of both time and the uncertainty of measurement.

To examine this relationship between optimisation of temporal and the uncertainty of measurement, twelve different problem instances are used and optimised for following three different metrics:

1. U - `(:metric minimize (u-m))`
2. T - `(:metric minimize (total-time))`
3. $\frac{(U+T)}{2}$ - `(:metric minimize (/(+(u_c)(totaltime))2))`

The experiments were performed on a AMD Phenom II 3.50 GHZ processor with 4 GB of RAM. The results show the most efficient plan produced within a 10 minute CPU time limit. All the produced plans are then validated using VAL [93]. VAL is the automatic validation tool for PDDL that is capable of validating PDDL solutions against PDDL problems and domains. These experiments were carried out without the ability to schedule measurements concurrently. This is because in this current model, the effect that concurrent measurements will have on the uncertainty of measurement has not been accounted for. It is likely that uncertainties could improve due to lower change in ambient conditions during relative measurements, but this could be counteracted by any need to use instrumentation with a higher uncertainty in order to achiever concurrent measurement.

Table 5.6 shows the empirical data from performing these experiments. From these results, it is evident that when optimising for time, no consideration is taken for the uncertainty due to the plan order. Similarly, it is evident that when optimising for the uncertainty due to the plan order, no consideration for temporal implications is taken.

Instance	Metric: Time T	U(µm)	Metric: Uncertainty T	U(µm)	Metric: $\overline{T+U}$ T	U(µm)
3A1A	33:12	99	34:12	52	33:38	53
3A1B	29:42	76	28:03	52	30:12	72
3A1C	29:21	66	31:45	59	29:21	70
3A2A	31:14	142	33:00	92	31:19	115
3A2B	28:27	135	30:34	94	28:57	112
3A2C	26:04	212	27:05	142	26:05	168
5A1A	52:05	120	56:56	18	55:11	27
5A1B	52:28	150	55:11	138	52:55	138
5A1C	50:18	199	51:29	193	50:54	193
5A2A	47:46	93	50:58	27	50:28	33
5A2B	45:17	90	47:46	82	46:05	82
5A2C	47:46	152	49:11	93	48:27	116

TABLE 5.6: Temporal & uncertainty optimisation results (PC).

However, when optimising the plan for both the uncertainty due to the order of the plan and reducing the overall timespan, it is evident that the planner (LPG-td) can establish a good compromise.

From Table 5.6 it is noticeable that a solution to each problem instance is found within the 10 minute time limit. In addition Table D.1 located in Appendix D shows exactly how many plans were produced during this time-limit and at what time the optimal plan was discovered. This information shows that the optimal plans were discovered on average after 8 minute 29 seconds of execution. This highlights that it is possible that the optimal plans are not being found within the 10 minute period. It is worth reiterating here that the results, much like those in Section 4.4.4.1, demonstrate the potential advantage of using automated planning based on the developed model. However, it is possible that experts with different opinions and knowledge might produce calibration plans that have a lower estimated uncertainty of measurement. Encoding this new knowledge in the model would then allow for comparable optimised calibration plans to be produced.

5.5.1 High Performance Computing

To investigate this further, without imposing a strict computation restriction, experiments were performed on a hardware platform with larger resource availabilities. The

	Metric: Time		Metric: Uncertainty		Metric: $\overline{T+U}$	
Instance	T(%)	U(%)	T(%)	U(%))	T(%)	U(%))
3A1A	1.5	-21.1	-1.0	0.4	2.3	1.9
3A1B	1.6	-61.6	-2.2	12.8	2.2	14.2
3A1C	0.3	-16.7	-0.6	-11.8	0.3	4.7
3A2A	1.2	0	0.5	2.2	0.9	21.4
3A2B	0.9	0	3.8	6.6	0	0
3A2C	0.5	-27.6	1.5	2.1	0	18.9
5A1A	0.4	-12.8	-10.1	0	4.4	50
5A1B	2.9	-44.8	-3.7	0	0.6	0
5A1C	2.5	-31.8	-4.0	0	1.4	0
5A2A	0	-18.6	2.5	0	5.7	20.4
5A2B	0.6	1.1	0	0	0.54	0
5A2C	0.8	-6.2	1.3	5.0	1.4	24.7
Average	1.1	-20.6	-1.0	1.4	1.6	13

TABLE 5.7: Percentage improvement between QQG and PC

chosen platform is the Huddersfield University Queensgate Grid (QGG) High Performance Computing (HPC) architecture. The dedicated hardware has 37 cores with a clock speed of 2.53GHz with 8GB of RAM allocated to each core. The same experiments as for the PC were performed on the QGG with a CPU execution time-limit of 24 hours. Table D.2 located in Appendix D shows the results from these experiments. From these results, it is evident that in almost all instances plans have been found with a lower metric. This highlights that providing significantly more computation time can result in plans that are better optimised. However, it is important to consider the gain in optimality to evaluate whether the extra computational resources are necessary. Table 5.7 shows the percentage improvement for each metric when comparing the experiments performed on the QGG and those on the PC. It is noticeable that while in most cases there is an improvement in the optimised metric, there is also often deterioration for the non-optimised metric. Additionally, there is an improvement for both metrics for the multi-object experiments.

The use of the QQG has shown that improvements over the optimal solutions identified on a PC can be achieved by using greater computation power. However, determining whether this is necessary is down to the end user. For example, for a calibration engineer wishing to perform a quick and effective calibration on an old machine tool operating with large tolerances, the use of a PC architecture is sufficient. Conversely, a calibration engineering calibration a state-of-the-art machine tool that operates to sub-micron tolerances within the aerospace sector will want to perform both the quickest and most effective calibration that can minimise the uncertainty of measurement, making the use of HPC for this engineer is justified.

5.5.2 Plan Excerpts

The following three plan excerpts (Figure 5.14, Figure 5.15 and Figure 5.16) illustrate the produced plans for the three different metrics and the differences between the order of measurement.

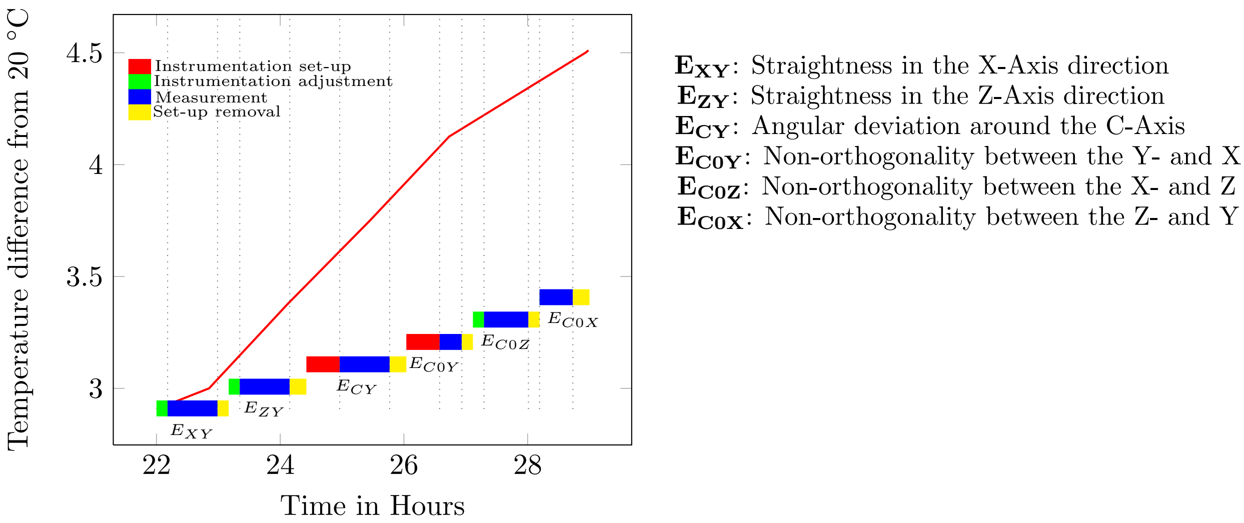
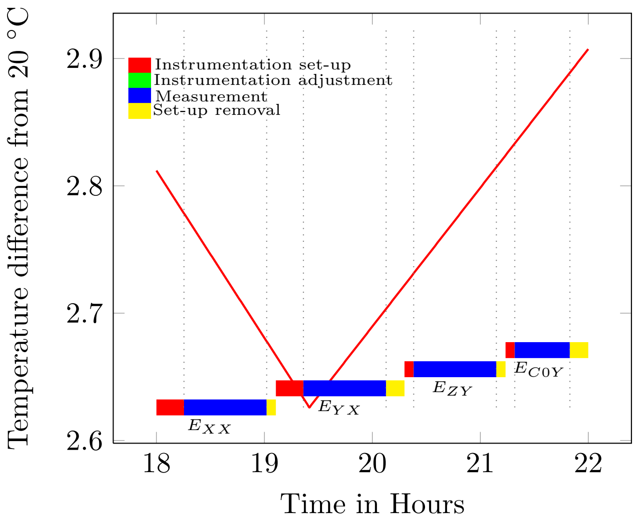


FIGURE 5.14: Temporal optimisation.

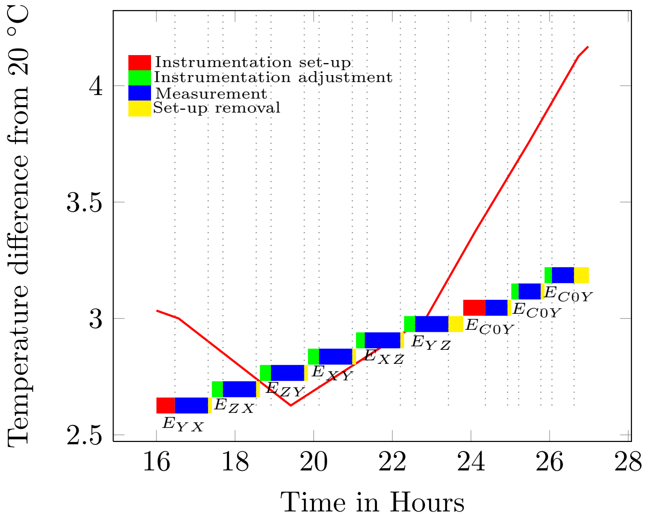
Figure 5.14 shows an excerpt from a temporally optimised plan produced from the 3A1A problem instance. The motivation for showing this particular excerpt is to investigate how the measurement of interrelated measurements is scheduled in the produced plan. Firstly, it is noticeable in the plan that the measurements that can use the same instrumentation are cluster together so the instruments can be adjusted from a previous measurement to save time, rather than set-up from a packaged state. It is also noticeable that the measurement order is not optimum for reducing the estimate uncertainty of measurement because of the measurement of the Y-axis about the Y-axis angular deviation (E_{CY}). This adds a time increase of around one hour between the interrelated straightness and non-orthogonal errors. The significance of this time period on uncertainty is that the continuing temperature increase will have a negative impact on the estimated uncertainty of measurement. From Table 5.6 it can be seen that the total machine downtime when using this calibration plan would be 33 hours and 12 minutes with an uncertainty of measurement due to the plan order metric of 99 µm.



$\mathbf{E_{XX}}$: Angular deviation around the C-Axis
$\mathbf{E_{XY}}$: Straightness in the X-Axis direction
$\mathbf{E_{ZY}}$: Straightness in the Z-Axis direction
$\mathbf{E_{C0Y}}$: Non-orthogonality between the Y- and X

FIGURE 5.15: Unceratinty optimisation.

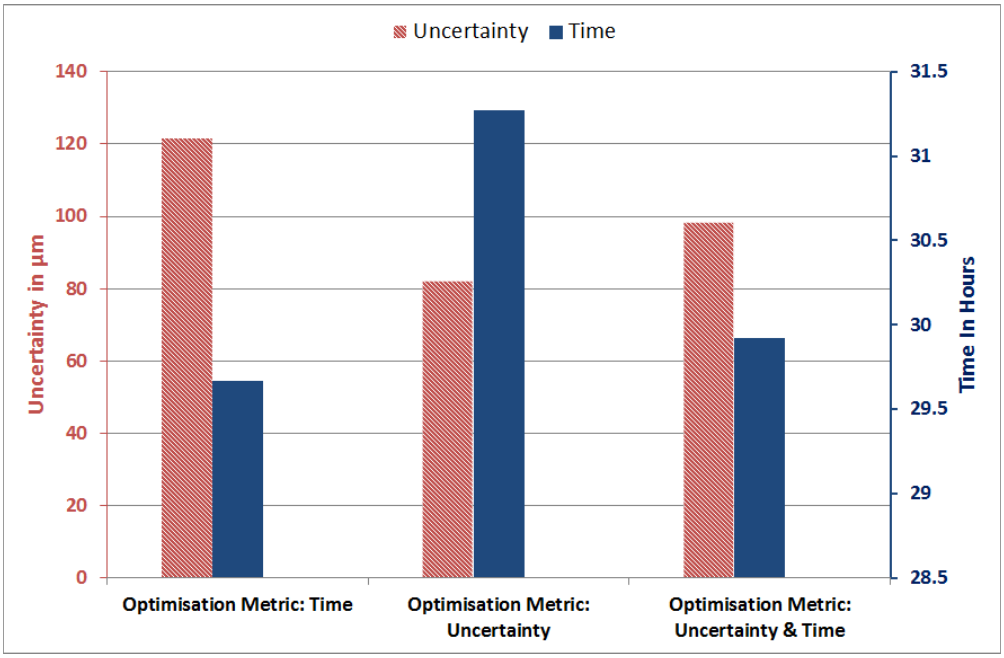
Figure 5.15 illustrates an excerpt from the produced plan for the same 3A1A. However this time optimising for the uncertainty of measure due to the ordering of the plan. Similarly to the plan excerpt shown in Figure 5.14, the plan excerpt shown in Figure 5.15 also displays the section of the plan that details the scheduling of interrelated measurements. From the plan, it is noticeable that temporal aspects have not been considered because even though measurements using the same instrumentation are grouped together, the planner has scheduled for the instrumentation to be removed and set-up, rather than adjusted. It is also noticeable that the plan is optimised to reduce the estimated uncertainty of measurement due to the plan order. This can be seen by the fact that the two interrelated straightness errors (E_{YX} and E_{ZY}) are scheduled sequentially followed by the measurement of non-orthogonality between the Y- and X-axis (E_{C0Y}). Scheduling these errors sequentially means that any effect due to changing temperature over time can be minimised. It can also be seen in the produced plan that the temperature variation over the course of the three interrelated measurements is only 0.3°C. The machine downtime when using this calibration plan would be 34 hours and 12 minutes with a plan order uncertainty of measurement metric of 52 µm. This plan results in an increased downtime of 1 hour over the temporally optimised plan, but reduces the uncertainty of measurement metric by 47 µm.



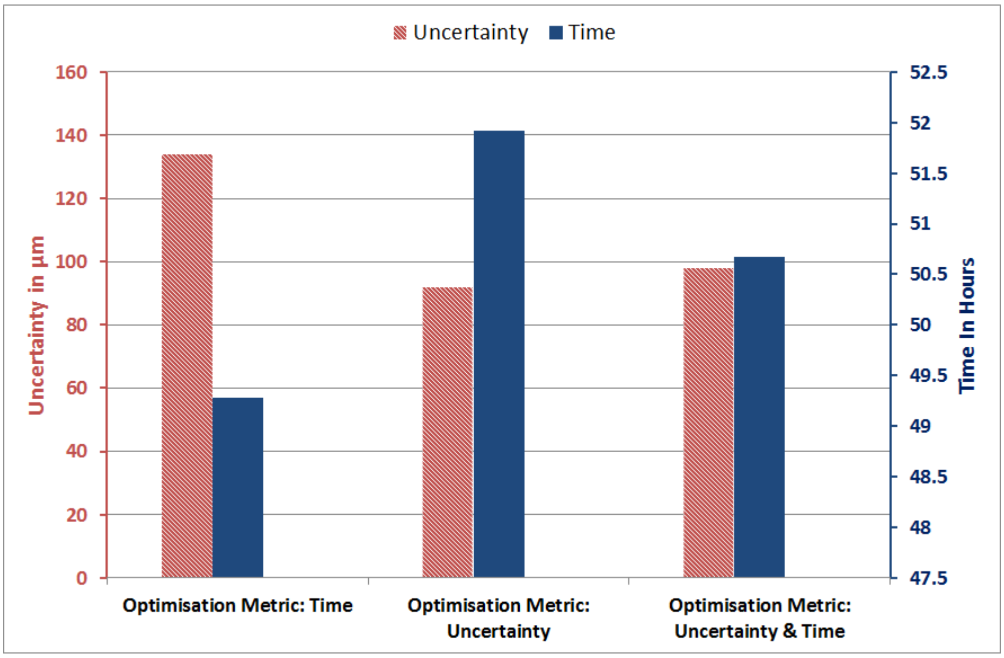
$\mathbf{E_{YX}}$: Straightness in the Y-Axis direction
$\mathbf{E_{ZX}}$: Straightness in the Z-Axis direction
$\mathbf{E_{ZY}}$: Straightness in the Z-Axis direction
$\mathbf{E_{XY}}$: Straightness in the X-Axis direction
$\mathbf{E_{XZ}}$: Straightness in the X-Axis direction
$\mathbf{E_{YZ}}$: Straightness in the Y-Axis direction
$\mathbf{E_{C0Y}}$: Squareness between the Y- and X
$\mathbf{E_{C0Z}}$: Squareness between the Z- and Y
$\mathbf{E_{C0Y}}$: Squareness between the X- and Z

FIGURE 5.16: Uncertainty and temporal optimisation.

The third plan excerpt shown in Figure 5.16 shows the plan order when optimising for both machine tool downtime and the uncertainty of measurement due to the plan order for problem instance 3A1A. Firstly, it is evident that temporal optimisation has been achieved by scheduling measurements that use the same instrumentation sequentially so that the instrumentation only needs to be adjusted, not removed and set-up once again. Secondly, it can be seen that the uncertainty of measurement due to the plan order has been reduced by scheduling interrelated measurements together as well as scheduling them where the temperature difference is at its lowest. From examining the temperature profile seen in Figure 5.7 it is evident that there are areas where the temperature difference is lower. However, when solving multi-objective optimisation planning problems, a trade-off between both metrics is going to take place. In Table 5.6 this trade-off can be seen where the calibration plan duration is 33 hours 38 minutes and the uncertainty of measurement metric is 53 µm. It is evident that both metrics are not as low as when optimising for them individually, but it is clear that the plan is a suitable compromise, showing significant reduction in both machine tool downtime and the uncertainty of measurement due to the plan order. In comparison between the single-objective optimum plans, the metrics in the multi-objective plans are on average 2.1% worse for time and 8.7% worse for the uncertainty of measurement than when they are optimised individually. However, the multi-objective search plans are on average have a 3.1% reduction in the time metric when compared to the downtime of the uncertainty optimised plan and a 23.2% improvement in estimated uncertainty of measurement metric when compared to the uncertainty of the temporally optimised plan.



(a) Average metrics for the three-axis problems



(b) Average metrics for the five-axis problems

FIGURE 5.17: Graph showing the average metrics for optimising time, uncertainty and time & uncertainty

The graph presented in Figure 5.17(a) shows the average metrics for the six different three-axis calibration instances, and Figure 5.17(b) shows the six different five-axis calibration instances. In these two figures, the effect on both metrics when performing a single-object optimisation can be visualised. Additionally, the trade-off between time and uncertainty when performing the multi-objective optimisation and the compromise in the final solution can easily be visualised. From these two graphs, it can be concluded that performing the multi-objective optimisation is beneficial as it produces plans that are close to the optimum.

5.6 Chapter Summary

In this chapter, an initial encoding for estimating the uncertainty of measuring linear deviation using a laser interferometer was produced by extending the previously developed temporal model (Section 4.3). This highlighted that using current PDDL arithmetic operators, it is not possible to implement the square root function. The Babylonian method was then implemented in form of iterative PDDL actions to solve the square root. However, this highlighted that computing the square root using this method will artificially increase the plan's complexity, therefore having adverse effects on generating plans for machine tool calibration. This resulted in the production of an alternative method where the Babylonian method was implemented in a single PDDL2.1 equation. This method does not require additional actions to be completed. However, generating an PDDL2.1 equation that is too short could result in the wrong value being calculated, whereas an PDDL2.1 equation that is too long may result in excessive computation as convergence is reached before the end of the equation.

The success of using the Babylonian method and embedding the equation in PDDL2.1 is dependent on the pre-processors ability to generate a formula of the correct depth. The solution was developed to calculate initially the uncertainty of measurement for positional deviation when using a laser interferometer. However, the equations cannot be used for different measurements, so a more generic and extensible solution was required. The extension of the model for many different measurement techniques and methods would be exhaustive and become too complex and go against the philosophy of being easily extensible.

This resulted in the production of a domain where instead of calculating the actual uncertainty for each measurement, the model takes into consideration factors that are known to reduce the uncertainty of the plan (For example, environment temperature). This method is tested and is able to produce optimal plans. However, to obtain the actual estimated uncertainty of measurement value, a post processor would be required. The overall outcome of the implementation shows that automated planning can successfully reduce the estimated uncertainty of measurement due to the plan order. In addition, the possibility to reduce both machine tool downtime and the estimated uncertainty of measurement using this method has been investigated. Results have suggested that it is possible to optimise for two different metrics and reach a good compromise that is close to optimal for both individual metrics. Additional experimentation has been performed on a HPC architecture to evaluate the effect of more computation time on the production of optimal plans. This concluded that using HPC can produce a greater increase in optimality and would be beneficial for calibrating machine tools manufacturing to sub-micron tolerances.

Chapter 6

Summary and Conclusions

In Chapter 2 of this thesis, it was identified that there are no published intelligent methods of producing calibration plans aimed at reducing machine tool downtime or the uncertainty of measurement. Even though the literature suggests there has been little research into calibration planning, the state-of-the-art in calibration planning is identified and discussed. This motivated research into automated planning technology for constructing human plans (Chapter 3) and the potential of it being applied to machine tool calibration planning. In Chapter 3, an evaluation of the state-of-the-art in terms of domain-independent planning technology is investigated, describing how problems are engineered, expressed and solved in the planning community.

The investigation to test the feasibility of automated planning for machine tool calibration resulted in parametrisation of the calibration process. This process involved logically identifying the parameters behind the decision criteria when performing a machine tool calibration. Initially, an HTN model was developed using the SHOP2 architecture. The main emphasis when developing the model was to reduce machine tool downtime. This resulted in a technique that can intelligently reason about instrumentation selection, set-up, adjustment and error measurement. Experimental analysis concluded that automated planning can be used to minimise machine tool downtime. However, the HTN model was a simplified representation of machine tool calibration and does not consider an extensive range of the parameters, most of which are concerned with taking concurrent measurements. Extending the HTN model to provide a fully defined domain model would have been feasible, however encoding the domain in this way would restrict the analysis to the SHOP2 architecture. It was identified that the best approach was to follow the philosophy of the automated planning community, where domain specific knowledge should be encoded using a domain-independent language and solved using domain independent planners. This allows for the use of an increased range of planners, resulting in a better utilisation and simpler adoption of state-of-the-art planning technology.

This resulted in the development of a PDDL2.2 domain. The domain, much like the HTN domain, is concerned with the temporal optimisation of the produced plan. However, in this extension the full set of parameters identified in Section 4.1 were implemented. Comparisons were then made between plans produced from the HTN and PDDL model, and it was noticed that both models produce similar temporally optimal plans when planning measurements sequentially. Once concurrent actions were enabled, the PDDL2.2 model demonstrated a significant reduction in machine tool downtime. In some cases the reduction was almost 50%, equating to around £1300. Plans for a five-axis machine tool with concurrent measurements were evaluated by human experts and then compared with the experts calibration plan (Chapter 2), where a reduced machine tool downtime of 11% is observed, equating to around £134. This clearly demonstrates the planners ability to produce valid, optimal calibration plans that are shorter in duration than those generated by an expert. This is a significant achievement because automatically producing a calibration plan, without the need of an expert, is both quicker in generation and cheaper in financial cost.

The development of the PDDL model aimed to produce a planning domain that could be solved by all domain independent planners that support the required level and expressiveness of PDDL. The produced temporal domain is implemented in PDDL2.2 and requires STRIPS and ADL, which many state-of-the art planning tools support (Chapter 3). Of the few that do solve the domain, problems regarding specific encoding techniques (negative preconditions, numeric conditions) restrict the domain to the LPG-td planner. Fortunately, LPG-td is one of the most powerful domain-independent planners, with a recognised ability to solve problems of a large variety with different requirements (temporal, numeric, non-linear effects, etc.). Research into domain-independent planners is a fast moving discipline with new, state-of-the-art planners being developed rapidly. It is envisaged that in the future there will be an increased range of planners able to solve the produced PDDL2.2 domain

The parametrisation of the planning process made in the temporal domain allows for the developed model to be easily used with developments in the state-of-the-art. For example, new machines with different kinematic chains, new instrumentation and measurement techniques can easily be included. This extensible philosophy is fundamental because in the future the range of planners able to solve the calibration problem will increase, potentially allowing for shorter plan generation and higher quality solutions. If the model was not able to handle advancements in the state-of-the-art, there is a possibility that it will become redundant as automated planning technology becomes more powerful.

The other optimisation function, identified in Chapter 2 is minimising the estimated uncertainty of measurement for the entire calibration plan. The first attempt to implement this function was by extending the temporal PDDL2.2 domain. However, it was soon observed that PDDL does not provide the square root function that is essential for estimating the uncertainty of measurement. An initial work-around was developed which implements the Babylonian method using a recursive PDDL action. Although this method can calculate the correct square root and provide it to the end user, it results in excess planning; in addition to planning a solution to the presented problem, the planner is also planning for a solution to the square root problem. This led to a less exhaustive method, where the Babylonian method is implemented as a nested equation. Although this solution is better in terms of computational effort, selecting the correct nesting depth, and initial estimate for the Babylonian method can result in an incorrect square root value.

Experiments were conducted, combining the nested equation with the temporal model to validate the feasibility of the model and the planner's ability to solve it. The experimental data demonstrated that this implementation could still be solved using the LPG-td planner. In addition, it was demonstrated that the planner could intelligently select measurement instrumentation and methods while reducing the uncertainty of measurement. However, the model was designed for estimating the uncertainty of measurement for measuring linear deviation using a laser interferometer, making the equations inapplicable for other measurements. Implementing the estimation equations for many different measurements would be exhaustive and go against the fundamental philosophy of producing a generic, extensible model that can take advantage of advancements in the future state-of-the-art.

After careful consideration, it was decided that rather than implementing an extensive set of uncertainty estimation formulae, which would without doubt make the domain too difficult for state-of-the-art planners to solve, only the necessary equations to describe the effect of environmental temperature on scheduling inter-related measurements are implemented. The difficulty of implementing the predictable, continuous temperature was achieved by using predictable, discretized exogenous events compiled down to PDDL2.1 durative actions. The model is constructed to consider the prevailing environmental temperature at different ages of a measurement (set-up, adjust, measure, etc.) to determine the temperature change throughout the measurement. This temperature change can then be used to estimate the uncertainty of measurement. Inter-related measurements and the effect of temperature change between their measurement are also considered.

Research illustrated that there was a potential 58% difference between the maximum and minimum uncertainty due to the ordering of the plan, and within a 10 minute cut-off, LPG-td found the optimal plan in 6 minutes 58 seconds. This shows that the model is capable of reasoning with environmental temperature and the effect it has on measurements and inter-related measurements to reduce the estimated uncertainty of measurement, which to the best of the author's knowledge is novel. Typically measurements are processed and the estimated uncertainty of measurement is reduced individually. However, using the developed model, the entire calibration plan is considered to find the best overall schedule. Additional experiments were then performed to investigate the possibility of optimising a multi-objective calibration plan for both time and uncertainty of measurement. In comparison between the single-objective optimum plans, the metrics in the multi-objective plans are on average 2.1% worse for time and 8.7% worse for the uncertainty of measurement than when they are optimised individually. However, the multi-objective search plans are on average have a 3.1% reduction in the time metric when compared to the downtime of the uncertainty optimised plan and a 23.2% improvement in estimated uncertainty of measurement metric when compared to the uncertainty of the temporally optimised plan.

Knowledge regarding the discovery of optimal plans when performing the experiments on a PC architecture (Table D.1) highlighted that the experimental analysis should be performed on a HPC architecture. These experiments displayed that there is on average a 3.7% improvement in optimality when compared with the experiments performed on the PC architecture. This warrants the use of the HPC resources for calibration engineers working to sub-micron tolerances and also suggests that a standard PC architecture is enough for most applications. As the state-of-the-art in both AI autonomous planners and PC computation power improve, the requirement for HPC resources should potentially reduce.

6.1 Limitations

The produced work in this thesis is novel and demonstrates an alternative approach to machine tool calibration planning. However, this approach is in its infancy and has the following limitations:

1. In the temporal model, the combination of PDDL2.2, numeric preconditions and effects as well as ADL, results in only LPG-td being able to solve the problem. LPG-td is still a state-of-the-art planner, but to what extent other planners can outperform LPG-td on the temporal domain has not been established. It is anticipated that the availability of the temporal and uncertainty domain will motivate planner development and result in a increased set of planners able to solve the domain in the future.

2. The duration data used in the problem definitions is based on historic information, and in the first instance acquired from expert-generated calibration plans. The correctness of the estimated durations is essential for generating optimal calibration plans. This reliance means that if the data is incorrect in the first instance, the produced calibration plan will also be incorrect. Additionally, a broader quantitative survey to determine the required durative data could help to improve the performance of the technology.

3. The implementation of repeatable temperature data can potential result in the loss of an important change if the measurement spans more than two discretized subprofiles. This is a big limitation for environments where the temperature fluctuates rapidly due to some predictable activity, like the factory door opening at exactly 10:00 hours each day.

4. The complexity of the uncertainty optimisation model restricts the use of domain-independent planners to LPG-td. To increase the range of planners that could solve it, the model could be relaxed to simplify it, but this would result in plans that are not realistic and do not closely represent the real-world planning problem.

5. The uncertainty model currently does not consider any other factors that can effect the estimated uncertainty of measurement other than temperature deviation. It should also be noted that this multi-objective function is a simple arithmetic mean of the timespan and the estimated uncertainty of measurement. Further extension of the model to allow for weighting between timespan and estimated uncertainty of measurement, depending upon the industry, situation and requirements should be considered in future work.

6.2 Summary of Novel Contributions

To summarise, the author believes that the work undertaken in this thesis has resulted in several contributions to knowledge for both the machine tool and automated planning communities. The following list provides the novel contributions within the machine tool community:

1. A generic and extensible method of automatically producing calibration plans that can reduce machine tool downtime. This method is implemented in both an HTN and PDDL2.2 model and can find an optimal measurement plan based on available temporal information. The literature suggests that previously little consideration to optimising a calibration plan by careful consideration to its construction. Therefore, the provided method is a novel contribution showing how optimisation can be performed.

2. A generic and extensible method of automatically producing calibration plans, reducing the uncertainty of measurement due to the order of the plan. This method takes into consideration predictable changed in environmental temperature and the effect it has on estimated uncertainty. The literature suggests that estimated measurement uncertainty is typically optimised on a per-measurement basis with little consideration of the effect of interrelated measurements and the dynamics of the

environment. The provided method is a novel contribution where the uncertainty of measurement for the entire calibration plan is reduced by careful construction.

3. The possibility to automatically produce calibration plans that are optimised in terms of machine tool downtime and the uncertainty of measurement due to the plan schedule. This provides a novel contribution to the machine tool maintenance community where calibration plans are not easily optimised for one criteria. However, using this model can allow for them to be be optimised for both criteria without the requirement of multiple experts.

4. Comparison have been made between automatically constructed, academic expert, and industrial expert calibration plans. This highlights the different philosophies behind machine tool calibration as well as showing the performance of the automatically constructed plans.

In addition to the novel contributions to the machine tool community, the following list provides the novel contributions to the automated planning community:

1. A Hierarchical Task Network model and a series of problem instances to represent the process of machine tool calibration. This domain and problem instances can be used as benchmarks in HTN research.

2. PDDL2.2 Temporal model representing the process of machine tool calibration that includes concurrent measurements. This domain and problem instances can be used in future IPC to motivated planner development. The domain and problem instances are showcased as a 'Real and Realistic Planning Domain' through the Special Interest Group for Applications of AI Planning and Scheduling (SIGAPS)[1].

3. Method of encoding the square root function in PDDL2.2 by using the Babylonian method implemented in a preprocessor which is useful technique for PDDL domain-engineering.

4. PDDL2.1 Numeric model implementing the process of planning for a machine tool calibration, optimising based on the estimated uncertainty of measurement due to the ordering of the plan.

[1]SIGAPS Real and Realistic Planning Domain webpage: `http://users.cecs.anu.edu.au/~patrik/sigaps/index.php?n=Main.RealDomains`

6.3 Suggested Future Work

This work has highlighted many potential areas for research. Some are provided below in order of importance:

1. Currently, only the effect of environmental temperature on the scheduling of measurements is considered. However, there are considerable more factors that effect the estimated uncertainty of measurement. For example, the effect of performing simultaneous measurements on the estimated uncertainty of measurement needs to be investigated and modelled. This would allow for concurrent measurements to be scheduled to minimise the uncertainty of measurement. Therefore, further reducing machine tool downtime and the uncertainty of measurement.

2. To implement these models in an industrial setting, a knowledge engineering tool will need to be provided since interacting with the PDDL temporal and uncertainty domains and problem instances is challenging. As highlighted in Section 3.9.2, the current state-of-the-art in knowledge engineering tools is not adequate for this purpose. Although, considering that the tools are developed without any specific domain in mind, their current achievement is powerful. This motivates research in to producing a suitable knowledge engineering tool that makes it both easier to encode machine tool calibration knowledge and to use state-of-the-art planning technology.

3. Extend the technology for online planning as well as offline planning. It is often the case that when an engineer arrives on site there will be unexpected constraints. For example, the machine might not have sufficient space to setup the laser interferometer. Therefore, the calibration problem instance and possibly the domain model will need to be modified and the plan regenerated. This would be a beneficial advancement and would allow for calibration plans to be regenerated online, taking into consideration any new constraints.

4. This work has highlighted that because of the complex requirements of the produced model, only LPG-td can be used to solve the planning problems. This motivates research into extending state-of-the-art domain-independent planners to support the full syntax of PDDL and domains with strong numeric properties.

This would be beneficial because any advancement in state-of-the-art domain-independent planning tools can be exploited by using the PDDL domain. This could possible result in faster plan generation time and higher quality solutions.

6.4 Published Papers

6.4.1 Refereed Journal Papers

1. Parkinson, Simon, Longstaff, Andrew P., Fletcher, Simon, Crampton, Andrew and Gregory, Peter (2012)**Automatic Planning for Machine Tool Calibration: A Case Study**. Expert Systems with Applications, 39 (13). pp. 11367-11377. ISSN 09574174

2. Parkinson, Simon, Longstaff, Andrew P and Fletcher, Simon (2013) **Automated Planning to Minimise Uncertainty of Machine Tool Calibration**. Accepted for publication in: International Journal of Engineering Applications of Artificial Intelligence.

3. Longstaff, Andrew P, Fletcher, Simon, Parkinson, Simon, Myers Alan (2013) **The Role of Measurement and Modelling of Machine Tools in Improving Product Quality**. Accepted for publication in: International Journal of Metrology and Quality Engineering.

6.4.2 Refereed Conference Papers

1. Parkinson, Simon, Longstaff, Andrew P., Crampton, Andrew and Gregory, Peter **Automated Planning for Multi-Objective Machine Tool Calibration: Optimising Makespan and Measurement Uncertainty**. Accepted for the Twenty-Fourth International Conference on Automated Planning and Scheduling

2. Longstaff, Andrew P, Fletcher, Simon, Parkinson, Simon, Myers, Alan (2013) **The Role of Measurement and Modelling of Machine Tools in Improving Product Quality**. Measurement Systems and Process Improvement (MIPS).

3. MMS Shah, L Chrpa, F Jimoh, D Kitchin, TL McCluskey, S Parkinson, M Vallati. (2013) **Knowledge Engineering Tools in Planning: State-of-the-art and**

Future Challenges. Proceedings of Knowledge Engineering for Planning and Scheduling (KEPS13).

4. Parkinson, Simon and Longstaff, Andrew P. (2012) **Increasing the Numeric Expressiveness of the Planning Domain Definition Language**. Proceedings of The 30th Workshop of the UK Planning and Scheduling Special Interest Group (PlanSIG2012).

5. Parkinson, Simon, Longstaff, Andrew P., Crampton, Andrew and Gregory, Peter (2012) **The Application of Automated Planning to Machine Tool Calibration**. In: Proceedings of the Twenty-Second International Conference on Automated Planning and Scheduling. ICAPS 2012 . AAAI Press, California, USA, pp. 216-224. ISBN 9781577355625

6. 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 13685708

7. Parkinson, Simon, Longstaff, Andrew P., Crampton, Andrew, Fletcher, Simon, Allen, Gary and Myers, Alan (2011) **Representing the Process of Machine Tool Calibration in First-order Logic**. In: Proceedings of the 17th International Conference on Automation & Computing. Chinese Automation and Computing Society, Huddersfield, UK. ISBN 9781862180987

6.4.3 Internally Refereed Conference

1. Parkinson, S., Longstaff, Andrew P., Crampton, Andrew, Fletcher, Simon, Allen, Gary and Myers, Alan (2012) **Automation as a Solution for Machine Tool Calibration Planning**. In: Proceedings of The Queens Diamond Jubilee Computing and Engineering Annual Researchers Conference 2012: CEARC12. University of Huddersfield, Huddersfield, pp. 57-62. ISBN 9781862181069

2. Parkinson, Simon, Longstaff, Andrew P., Fletcher, Simon, Allen, Gary, Crampton, Andrew and Myers, Alan (2010) **A Novel Framework for Establishing a Machine Tool Quality Metric**. In: Future Technologies in Computing and

Engineering: Proceedings of Computing and Engineering Annual Researchers' Conference 2010: CEARC10. University of Huddersfield, Huddersfield. ISBN 9781862180932

References

[1] Samir Mekid. *Introduction to Precision Machine Design and Error Assessment*. Number v. 10 in Mechanical Engineering Series. Taylor & Francis Group, 2009. ISBN 9780849378867.

[2] H. Schwenke, W. Knapp, H. Haitjema, A. Weckenmann, R. Schmitt, and F. Delbressine. Geometric error measurement and compensation of machines an update. *CIRP Annals - Manufacturing Technology*, 57:660 – 675, 2008. ISSN 0007-8506.

[3] ISO230. Part 2: Determination of accuracy and repeatabiliy of positioning numerically controlled axes, 2006.

[4] LaserTRACER - measuring sub-micron in space. http://www.etalon-ag.com/index.php/en/products/lasertracer. Accessed: 2013-06-03.

[5] B Bringmann and W Knapp. Machine tool calibration: Geometric test uncertainty depends on machine tool performance. *Precision Engineering*, 33(4):524 – 529, 2009. ISSN 0141-6359.

[6] Scott R. Postlethwaite. *Electronic based accuracy enhancement of CNC machine tools*. PhD thesis, Huddersfield Polytechnic, June 1992.

[7] Malik Ghallab, Dana Nau, and Paolo Traverso. *Automated planning: theory & practice*. Morgan Kaufmann, 2004.

[8] Kutluhan Erol, Dana S. Nau, and V.S. Subrahmanian. Complexity, decidability and undecidability results for domain-independent planning. *Artificial Intelligence*, 76(12):75 – 88, 1995. ISSN 0004-3702.

[9] Drew M McDermott. The 1998 AI planning systems competition. *AI magazine*, 21 (2):35, 2000.

[10] Drew McDermott, Malik Ghallab, Adele Howe, Craig Knoblock, Ashwin Ram, Manuela Veloso, Daniel Weld, and David Wilkins. PDDL-the planning domain definition language. 1998.

[11] Y Kakino, Y Ihara, and A Shinohara. *Accuracy inspection of NC machine tools by double ball bar method*. Hanser Gardner Publications, 1993.

[12] ISO230. Part 1: Geometric accuracy of machines operating under no-load or quasi-static conditions, 2012.

[13] T Moriwaki. Multi-functional machine tool. *CIRP Annals - Manufacturing Technology*, 57(2):736 – 749, 2008. ISSN 0007-8506.

[14] Naeem S. Mian, Simon Fletcher, Andrew P. Longstaff, and Alan Myers. Efficient thermal error prediction in a machine tool using finite element analysis. volume 22, page 085107. IOPP, August 2011.

[15] Naeem S. Mian, Simon Fletcher, Andrew P. Longstaff, and Alan Myers. An efficient offline method for determining the thermally sensitive points of a machine tool structure. *37th MATADOR Conference*, September 2012.

[16] ISO230. Part 7: Geometric accuracy of axes of rotation, 2006.

[17] Simon Fletcher, Andrew P Longstaff, and Alan Myers. Investigation into the accuracy of a proposed laser diode based multilateration machine tool calibration system. In *Journal of Physics: Conference Series*, volume 13, page 398. IOP Publishing, 2005.

[18] 2-dimensional precision inclination sensor with automatic reversale measurement. http://www.wylerag.com/pages_eng/e4_421a.html. Accessed: 2013-06-03.

[19] BIPM. Evaluation of measurement data Guide to the expression of Uncertainty in Measurement, 2008.

[20] Stephanie Bell. *A beginner's guide to uncertainty of measurement*. National Physical Laboratory, 1999.

[21] Alistair Forbes. Measurement uncertainty and optimized conformance assessment. *Measurement*, 39(9):808 – 814, 2006. ISSN 0263-2241.

[22] Keith Birch. Estimating uncertainties in testing. *Measurement Good Practice Guide*, (36), 2003.

[23] ISO230. Part 9: Estimation of measurement uncertainty for machine tool tests according to series iso 230, basic equations, 2005.

[24] M Alkan Donmez, DS Blomquist, RJ Hocken, CR Liu, and Moshe M Barash. A general methodology for machine tool accuracy enhancement by error compensation. *Precision Engineering*, 8(4):187–196, 1986.

[25] Abubaker Shagluf, Andrew Longstaff, and Simon Fletcher. Towards a downtime cost function to optimise machine tool calibration schedules. In *Proceedings of the Interational Conference on Advanced Manufacturing Engineering and Technologies NEWTECH 2013*, volume 16, pages 231–240. KTH Royal Institute of Technology, Stockholm, Sweden, 2013.

[26] B Bringmann, JP Besuchet, and L Rohr. Systematic evaluation of calibration methods. *CIRP Annals - Manufacturing Technology*, 57(1"):529 – 532", 2008". ISSN 0007-8506.

[27] ISO230. Part 4: Circular tests for numerically controlled machine tools, 2005.

[28] Jody E Muelaner, Bin Cai, and Paul G Maropoulos. Large-volume metrology instrument selection and measurability analysis. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, 224(6):853–868, 2010.

[29] XD laser - precision laser measurement system. http://www.apisensor.com/index.php/products-en/machine-tool-calibration-en/xd-laser-en. Accessed: 2013-06-11.

[30] System software for QC20-W ballbar. http://www.renishaw.com/en/system-software-for-qc20-w-ballbar--11076. Accessed: 2013-09-16.

[31] Beamex CMX Professional Calibration Software. http://www.beamex.com/beamex_products/CMX-Professional-Calibration-Software/na15ghgl/6245a6b8-0ace-495e-bd6e-504381bbd080#Features. Accessed: 2013-06-03.

[32] S. G. Tzafestas. *Advances in manufacturing: decision, control, and information technology*. Springer, 1999.

[33] HB Marri, A Gunasekaran, and RJ Grieve. Computer-aided process planning: a state of art. *The International Journal of Advanced Manufacturing Technology*, 14 (4):261–268, 1998.

[34] Hsin-Chuan Chen, Hong-Tzong Yau, and Chien-Chung Lin. Computer-aided process planning for nc tool path generation of complex shoe molds. *The International Journal of Advanced Manufacturing Technology*, 58(5-8):607–619, 2012.

[35] Stuart Jonathan Russell, Peter Norvig, John F Canny, Jitendra M Malik, and Douglas D Edwards. *Artificial intelligence: a modern approach*, volume 74. Prentice hall Englewood Cliffs, 1995.

[36] Xinyu Li, Chaoyong Zhang, Liang Gao, Weidong Li, and Xinyu Shao. An agent-based approach for integrated process planning and scheduling. *Expert Systems with Applications*, 37(2):1256 – 1264, 2010. ISSN 0957-4174.

[37] Behrokh Khoshnevis and QingMei Chen. Integration of process planning and scheduling functions. *Journal of Intelligent Manufacturing*, 2(3):165–175, 1991. ISSN 0956-5515.

[38] Peter E Hart, Nils J Nilsson, and Bertram Raphael. Correction to a formal basis for the heuristic determination of minimum cost paths. *ACM SIGART Bulletin*, (37):28–29, 1972.

[39] Keith Halsey. *CRIKEY! - It's co-ordination in temporal planning.* PhD thesis, Durham University, 2004.

[40] James F Allen and Patrick J Hayes. A common-sense theory of time. *International Joint Conferences on Artificial Intelligence (IJCAI)*, 85:528–531, 1985.

[41] James F. Allen. Towards a general theory of action and time. *International Journal of Artificial Intelligence*, 23(2):123 – 154, 1984. ISSN 0004-3702.

[42] Satyandra K Gupta, Dana S Nau, and William C Regli. IMACS: A case study in real-world planning. *Intelligent Systems and their Applications, IEEE*, 13(3):49–60, 1998.

[43] Ferenc Deák, András Kovács, József Váncza, and Tadeusz P Dobrowiecki. Hierarchical knowledge-based process planning in manufacturing. In *Proceedings of the*

IFIP 11 international PROLAMAT conference on digital enterprise, pages 428–439. Citeseer, 2001.

[44] Dana Nau, Yue Cao, Amnon Lotem, and Hector Muñoz-Avila. SHOP: Simple hierarchical ordered planner. In *Proceedings of the 16th international joint conference on Artificial intelligence-Volume 2*, pages 968–973. Morgan Kaufmann Publishers Inc., 1999.

[45] Richard E Fikes and Nils J Nilsson. Strips: A new approach to the application of theorem proving to problem solving. *Artificial intelligence*, 2(3):189–208, 1972.

[46] Edwin P. D. Pednault. Formulating Multi-Agent Dynamic-World Problems in the Classical Planning Framework. In Michael P. Georgeff and Amy L. Lansky, editors, *Reasoning About Actions and Plans: Proceedings of the 1986 Workshop*, pages 47–82, San Mateo, CA, 1987. Morgan Kaufmann Publishers.

[47] Maria Fox and Derek Long. PDDL2.1: An extension to PDDL for expressing temporal planning domains. *Journal of Artificial Intelligence Research (JAIR)*, 20: 61–124, 2003.

[48] Derek Long and Maria Fox. The 3rd international planning competition: Results and analysis. *Journal of Artificial Intelligence Research (JAIR)*, 20:1–59, 2003.

[49] Stefan Edelkamp and Jörg Hoffmann. PDDL2. 2: The language for the classical part of the 4th international planning competition. *Proceedings of the International Conference on Automated Planning and Scheduling (ICAPS)*, 2004.

[50] Jörg Hoffmann, Stefan Edelkamp, Sylvie Thiébaux, Roman Englert, Frederico Liporace, and Sebastian Trüg. Engineering benchmarks for planning: the domains used in the deterministic part of IPC-4. *Journal of Artificial Intelligence Research*, 26(1):453–541, 2006.

[51] Maria Fox, Derek Long, and Keith Halsey. An investigation into the expressive power of pddl2. 1. In *ECAI*, volume 16, page 338, 2004.

[52] Alfonso Gerevini and Derek Long. Plan constraints and preferences in PDDL3. *The Language of the Fifth International Planning Competition. Tech. Rep. Technical Report, Department of Electronics for Automation, University of Brescia, Italy*, 2005.

[53] M Helmert. Changes in PDDL 3.1. *Unpublished summary from the IPC-2008 website*, 2008.

[54] Maria Fox and Derek Long. Modelling mixed discrete-continuous domains for planning. *Journal of Artificial Intelligence Research*, 27(1):235–297, 2006.

[55] Maria Fox and Derek Long. PDDL+: Modelling continuous time-dependent effects. In *Proc. 3rd International NASA Workshop on Planning and Scheduling for Space*. Citeseer, 2002.

[56] Maria Fox, Derek Long, and Daniele Magazzeni. Automatic construction of efficient multiple battery usage policies. In *Proceedings of the Twenty-Second international joint conference on Artificial Intelligence-Volume Volume Three*, pages 2620–2625. AAAI Press, 2011.

[57] Maria Fox, Derek Long, and Daniele Magazzeni. Automatic Construction of Efficient Multiple Battery Usage Policies. In *Proceedings of the International Conference on Automated Planning and Scheduling (ICAPS)*, pages 74–81, 2011.

[58] Ronald Alford, Ugur Kuter, and Dana S Nau. Translating htns to pddl: A small amount of domain knowledge can go a long way. In *International Joint Conferences on Artificial Intelligence (IJCAI)*, pages 1629–1634, 2009.

[59] Alfonso E Gerevini, Alessandro Saetti, and Ivan Serina. An approach to efficient planning with numerical fluents and multi-criteria plan quality. *Artificial Intelligence*, 172(8):899–944, 2008.

[60] Derek Long and Maria Fox. The 3rd international planning competition: Results and analysis. *Journal of Artificial Intelligence Research (JAIR)*, 20:1–59, 2003.

[61] Alfonso Gerevini, Alessandro Saetti, and Ivan Serina. An Approach to Temporal Planning and Scheduling in Domains with Predictable Exogenous Events. *Journal of Artificial Intelligence Research (JAIR)*, 25:187–231, 2006.

[62] Alfonso Gerevini and Ivan Serina. Lpg: A planner based on local search for planning graphs with action costs. In *AIPS*, volume 2, pages 281–290, 2002.

[63] Alfonso Gerevini, Alessandro Saetti, and Ivan Serina. Planning with numerical expressions in lpg. 2004.

[64] Andrew Coles, Maria Fox, Keith Halsey, Derek Long, and Amanda Smith. Managing concurrency in temporal planning using planner-scheduler interaction. *Artificial Intelligence*, 173(1):1 – 44, 2009. ISSN 0004-3702.

[65] AI Coles, Maria Fox, Derek Long, and AJ Smith. A hybrid relaxed planning graph-lp heuristic for numeric planning domains. In *Proceedings of the 18th International Conference on Automated Planning and Scheduling*, 2008.

[66] Jörg Hoffmann and Bernhard Nebel. The FF planning system: Fast plan generation through heuristic search. *Journal of Artificial Intelligence Research (JAIR)*, 14: 253–302, 2001.

[67] A. J. Coles and A. I. Coles. LPRPG-P: Relaxed plan heuristics for planning with preferences. In *Proceedings of the International Conference on Automated Planning and Scheduling (ICAPS)*, June 2011.

[68] Alfonso Gerevini and Derek Long. Preferences and soft constraints in pddl3. In *Proceedings of the International Conference on Automated Planning and Scheduling (ICAPS)*, pages 46–53, 2006.

[69] A. J. Coles, A. I. Coles, M. Fox, and D. Long. Temporal planning in domains with linear processes. In *International Joint Conferences on Artificial Intelligence (IJCAI)*. AAAI Press, July 2009.

[70] Amanda Jane Coles, Andrew Coles, Maria Fox, and Derek Long. Colin: Planning with continuous linear numeric change. *Journal of Artificial Intelligence Research (JAIR)*, 44:1–96, 2012.

[71] Amanda Jane Coles, Andrew Coles, Maria Fox, and Derek Long. Forward-chaining partial-order planning. In *Proceedings of the International Conference on Automated Planning and Scheduling (ICAPS)*, pages 42–49, 2010.

[72] J Benton, Amanda Coles, and Andrew Coles. Temporal planning with preferences and time-dependent continuous costs. In *Twenty-Second International Conference on Automated Planning and Scheduling*, 2012.

[73] Kevin Tierney, Amanda Jane Coles, Andrew Coles, Christian Kroer, Adam M Britt, and Rune Møller Jensen. Automated planning for liner shipping fleet repositioning.

In *Proceedings of the 22nd International Conference on Automated Planning and Scheduling*, 2012.

[74] Ji Shin and Ernest Davis. Processes and continuous change in a sat-based planner. *Artificial Intelligence*, 166(1):194–253, 2005.

[75] Steven A Wolfman and Daniel S Weld. The LPSAT engine & its application to resource planning. In *International Joint Conference on Artificial Intelligence*, volume 16, pages 310–317. Lawrence Erlbaum Associates Ltd, 1999.

[76] Håkan LS Younes and Reid G Simmons. VHPOP: Versatile heuristic partial order planner. *Journal of Artificial Intelligence Research (JAIR)*, 20:405–430, 2003.

[77] Giuseppe Della Penna, B Intrigila, D Magazzeni, and Fabio Mercorio. Upmurphi: a tool for universal planning on pddl+ problems. In *Proceedings of the International Conference on Automated Planning and Scheduling (ICAPS)*, pages 106–113, 2009.

[78] Giuseppe Della Penna, Daniele Magazzeni, and Fabio Mercorio. A universal planning system for hybrid domains. *Applied intelligence*, 36(4):932–959, 2012.

[79] Matthew Molineaux, Matthew Klenk, and David Aha. Planning in dynamic environments: Extending HTNs with nonlinear continuous effects. In *Twenty-Fourth AAAI Conference on Artificial Intelligence*, 2010.

[80] Javier Barreiro, Matthew Boyce, Minh Do, Michael Iatauro Jeremy Frank, Tatiana Kichkayloz, Paul Morrisy, James Ong, Emilio Remolina, Tristan Smith, and David Smithy. EUROPA: A Platform for AI Planning, Scheduling, Constraint Programming, and Optimization. In *Proceedings of the International Conference on Automated Planning and Scheduling (ICAPS)*, 2012.

[81] Jeremy Frank and Ari Jónsson. Constraint-based attribute and interval planning. *Constraints*, 8(4):339–364, 2003. ISSN 1383-7133.

[82] Sara Bernardini and David E Smith. Developing domain-independent search control for europa2. In *Proceedings of the International Conference on Automated Planning and Scheduling (ICAPS)*, 2007.

[83] Ron M Simpson, Diane E Kitchin, and TL McCluskey. Planning domain definition using gipo. *Knowledge Engineering Review*, 22(2):117–134, 2007.

[84] Tiago Stegun Vaquero, Victor Romero, Flavio Tonidandel, and José Reinaldo Silva. itSIMPLE 2.0: An integrated tool for designing planning domains. In *Proceedings of the International Conference on Automated Planning & Scheduling*, pages 336–343, 2007.

[85] OMG Uml. 2.0 superstructure specification. *OMG, Needham*, 2004.

[86] Arturo González-Ferrer, Juan Fernández-Olivares, Luis Castillo, et al. JABBAH: a JAVA application framework for the translation between business process models and HTN. *Proceedings of the 3rd International Competition on Knowledge Engineering for Planning and Scheduling (ICKEPS09)*, 2009.

[87] Eric Bouillet, Mark Feblowitz, Hanhua Feng, Anand Ranganathan, Anton Riabov, Octavian Udrea, and Zhen Liu. Mario: middleware for assembly and deployment of multi-platform flow-based applications. In *Proceedings of the 10th International Conference on Middleware*, Middleware '09, pages 26:1–26:7, New York, NY, USA, 2009. Springer-Verlag New York, Inc.

[88] Tomas Plch, Miroslav Chomut, Cyril Brom, and Roman Barták. Inspect, edit and debug pddl documents: Simply and efficiently with pddl studio. *Proceedings of the International Conference on Automated Planning and Scheduling (ICAPS)*, page 4, 2012.

[89] J. Vodrážka and L. Chrpa. Visual design of planning domains. In *KEPS 2010: Workshop on Knowledge Engineering for Planning and Scheduling*, 2010.

[90] Dana Nau, Tsu-Chiu Au, Okhtay Ilghami, Uger Kuter, J William Murdock, Dan Wu, and Fusun Yaman. SHOP2 : An HTN Planning System. *Journal of Artificial Intelligence Research (JAIR)*, 20:379–404, 2003.

[91] R. Burden and J.D. Faires. *Numerical Analysis*. Brookes/Cole, 2010.

[92] David Goldberg. What every computer scientist should know about floating-point arithmetic. *ACM Computing Surveys*, 23(1):5–48, March 1991. ISSN 0360-0300.

[93] Richard Howey, Derek Long, and Maria Fox. Val: Automatic plan validation, continuous effects and mixed initiative planning using pddl. In *Tools with Artificial Intelligence, 2004. ICTAI 2004. 16th IEEE International Conference on*, pages 294–301. IEEE, 2004.

Appendix A

HTN Domain Model

A.1 HTN Domain

```
(in-package :shop2-user)

;;operators
(defdomain mtc (
   (:operator (!select-error ?a ?e ?c)
   ((meas_required ?x ?y ?c ))
   ()
   ((meas_selected ?a ?e )))

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

   (:operator (!set-up-eqip ?a ?e ?i ?mc ?c )
   ((equip_selected ?a ?e ?i ?mc ?c ?ac))
   ()
   ((equip_setup ?a ?e ?i ?mc))
   (* 1 ?c));; equipment use cost

   (: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));; equipment use cost

   (: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));;measurement cost

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

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

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

;;find all the components that need measuring
(:method (find-all-required)
((linear ?axis)(linear-geometric-error ?err ?c)
(not(meas_required ?axis ?err ?c)))
;Decomposition
((!!assert ((meas_required ?axis ?err ?c )))(find-all-required))

((linear ?axis)(cross-axis-error ?err ?c)(not(meas_required ?axis ?err ?c)))
;Decomposition
((!!assert ((meas_required ?axis ?err ?c )))(find-all-required))

((rotary ?axis)(rotary-geometric-error ?err ?c)
(not(meas_required ?axis ?err ?c)))
;Decomposition
((!!assert ((meas_required ?axis ?err ?c )))(find-all-required))
nil
nil)

;;perform the measurement by first selecting the error
(:method(calibrate)
((meas_required ?x ?y ?c) (not(meas_selected ?x ?y ))(not(measured ?x ?y )))
;Decomposition
((!select-error ?x ?y ?c )(select-equipment)(calibrate))
nil
nil)

;;select equipment
(:method(select-equipment)
((meas_selected ?axis ?err )(equipment ?i ?c ?ac)(measures ?err ?i ?mc)
(not(equip_selected ?axis ?err ?i ?mc ?c ?ac)))
;Decomposition
```

```
   ((!select-equip ?axis ?err ?i ?mc ?c ?ac)(set-up-equipment)(select-equipment))
   nil
   nil)

   ;;set up equipment
   (:method(set-up-equipment)
   ((equip_selected ?x ?y ?i ?mc ?c ?ac)(not(previous_error ?x ?pe ?i ?pmc))
   (not((equip_setup ?x ?y ?i ?mc))))
   ;Decomposition
   ((!set-up-eqip ?x ?y ?i ?mc ?c)(measure-error) (set-up-equipment))

   ((equip_selected ?x ?y ?i ?mc ?c ?ac)(previous_error ?x ?pe ?i ?pmc)
   (not((equip_setup ?x ?y ?i ?mc))))
   ;Decomposition
   ((!adjust-equip ?x ?y ?i ?mc ?c ?pe ?pmc ?ac)(measure-error) (set-up-equipment))
   nil
   nil)

   ;;remove previous error
   (: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)

   ;;measure
   (:method(measure-error)
   ((equip_setup ?x ?y ?i ?mc)(meas_required ?x ?y ?c))
   ;;decompose
   ((!measure ?x ?y ?i ?mc )(remove-previous)
   (!!assert((previous_error ?x ?y ?i ?mc)))
   (!!remove ((meas_required ?x ?y ?c))))
   nil
   nil)))
```

A.2 Example Five-Axis HTN Problem

```
;;problem definition
(defproblem 5axis11 mtc
   ;;facts
   ;;Axes
   (axis X)
   (axis Y)
   (axis Z)
   (axis C)
   (axis A)
   (axis S)
```

```
;;Axis type
(linear X)
(linear Y)
(linear Z)
(rotary C)
(rotary A)
(spindle S)

;;Linear geometric error + cost in importance
(linear-geometric-error POS 2)
(linear-geometric-error ACC_AND_REP 2)
(linear-geometric-error PITCH 2)
(linear-geometric-error ROLL 2)
(linear-geometric-error YAW 2)
(linear-geometric-error HORZSTRAI 3)
(linear-geometric-error VERTZSTRAI 3)
(linear-geometric-error TABLE_PARALLELISM 3)
(cross-axis-error SQUARENESS 1)

(rotary-geometric-error ZERO_SETTING 2)
(rotary-geometric-error POSITIONAL_ACC  3)
(rotary-geometric-error ACC_AND_REP 3)
(rotary-geometric-error PARALLISM_TO_PLANE 3)
(rotary-geometric-error PIVOT_LENGTH 2)
(rotary-geometric-error CONINCIDENCE 2)

(spindle-geometric-error SPINDLE_AXIAL_RUNOUT 2)
(spindle-geometric-error RADIAL_RUNOUT 2)
(spindle-geometric-error SPINDLE_INTERNAL_TAPER 2)
(spindle-geometric-error CONINCIDENCE_SPINDLE 2)

;;Equipment + setup and adjust time (mins)
;;(equipment LASER_2  10 44)
(equipment LASER  10 7)
(equipment DIGITAL_LEVEL  10 5)
(equipment DTI_GRANIE_SQUARE  10 10)
(equipment DTI_STRAIGHT_EDGE  10 10)
(equipment BALLBAR  10 5)
(equipment PRECISION_BALL  10 5)
(equipment RENISHAW_XR20W 10 5)
(equipment DISPLACEMENT 10 2)
(equipment TEST_BAR_2_CLOCKS 10 5)
(equipment CLOCK_TEST_BAR 10 5)
(equipment CLOCK 10 5)
(equipment CLOCK_ON_TABLE 10 5)

;;Measurement + cost of performing (mins)
;;linear
(measures POS LASER  2)
(measures ACC_AND_REP  LASER  10)
```

```
    (measures PITCH LASER  2)
    (measures ROLL DIGITAL_LEVEL  2)
    (measures YAW LASER  2)
    (measures HORZSTRAI LASER  2)
    (measures VERTZSTRAI LASER  2)
    (measures SQUARENESS BALLBAR  2)
    (measures TABLE_PARALLELISM CLOCK_ON_TABLE  2)

    ;;rotoary
    (measures ZERO_SETTING TEST_BAR_2_CLOCKS 2)
    (measures POSITIONAL_ACC RENISHAW_XR20W 2)
    (measures ACC_AND_REP RENISHAW_XR20W 2)
    (measures PARALLISM_TO_PLANE TEST_BAR_2_CLOCKS 2)
    (measures PIVOT_LENGTH TEST_BAR_2_CLOCKS 2)
    (measures CONINCIDENCE TEST_BAR_2_CLOCKS 2)

    ;;spindle
    (measures SPINDLE_AXIAL_RUNOUT CLOCK_TEST_BAR 2)
    (measures RADIAL_RUNOUT CLOCK_TEST_BAR   2)
    (measures SPINDLE_INTERNAL_TAPER CLOCK   2)
    (measures CONINCIDENCE_SPINDLE TEST_BAR_2_CLOCKS   2)

   )
   (;;goal
    (perform-calibration)
))
;;Execution command including metric
(find-plans '5axis11 :verbose :plans :which :first :optimize-cost t :time-limit 60)
```

Appendix B

PDDL2.2 Temporal Optimisation Model

B.1 PDDL Domain

```
(define (domain calibration_domain)
    (:requirements :strips :fluents :typing :timed-initial-literals
:negative-preconditions)
    (:types
        Error - object
        Instrument - object
        Axis - object
    )

    (:predicates
        (axis-error ?axi - Axis ?err - Error)
        (measures ?ins - Instrument ?err - Error)
        (measured ?ax - Axis ?err - Error)
        (in-operation ?ins - Instrument)
        (set-up-axis ?ins - Instrument ?axi - Axis)
        (set-up-error ?ins - Instrument ?err - Error)
        (compatible ?ins1 ?ins2 - Instrument)
        (blocked    ?in - instrument ?ax - axis)
        (working-day)
    )

    (:functions
        (working-range ?ins - Instrument)
        (length-to-measure ?axi - Axis ?er - Error)
        (amount-measured ?ax - Axis ?er - Error ?in - Instrument)
        (measurement-time ?in - Instrument ?er - Error ?ax - Axis)
        (setup-time ?in - Instrument ?er - Error ?ax - Axis)
        (adjust-error-time ?in - Instrument ?er - Error ?ax - Axis)
        (adjust-repos-time ?in - Instrument ?er - Error ?ax - Axis)
        (removal-time ?in - Instrument ?er - Error ?ax - Axis)
```

```
    (measurement-overlap ?in - Instrument)
    (using      ?in - instrument)
    (dof  ?in - instrument)
    (importance ?ax - axis ?er - error)
    (total-importance)
    (feedrate ?ax - Axis ?er - Error ?in - Instrument)
    (targets ?ax - Axis ?er - Error ?in - Instrument)
    (dwell ?ax - Axis ?er - Error ?in - Instrument)
)

(:durative-action setup
:parameters (?in - Instrument ?er - Error ?ax - Axis)
:duration(= ?duration (setup-time ?in ?er ?ax))
:condition
    (and (over all (not (blocked ?in ?ax)))
      (over all (axis-error ?ax ?er))
      (over all (measures ?in ?er))
      (at start (not (measured ?ax ?er)))
      (over all (forall (?a - Axis ?e - Error ?i - Instrument)
      (imply (set-up-axis ?i ?a)
      (and(= ?a ?ax)
            (=(feedrate ?ax ?er ?in)(feedrate ?a ?e ?i))
            (=(targets ?ax ?er ?in)(feedrate ?a ?e ?i))
            (=(dwell ?ax ?er ?in)(feedrate ?a ?e ?i))))))
      (over all (forall (?i - Instrument) (imply (in-operation ?i)
      (compatible ?i ?in))))
      (over all (<=(amount-measured ?ax ?er ?in)(length-to-measure ?ax ?er)))
      (at start (not (set-up-error ?in ?er)))
      (at start (not (set-up-axis ?in ?ax)))
      (at start (not (in-operation ?in)))
      (at start (< (using ?in) (dof ?in)))
      (over all (working-day))
    )
:effect
    (and
      (at end (set-up-error ?in ?er))
      (at end (set-up-axis ?in ?ax))
      (at end (in-operation ?in))
      (at start (increase (using ?in) 1))
    )
)

(:durative-action adjust-postition
:parameters (?in - Instrument ?er - Error ?e - Error ?ax - Axis)
:duration(= ?duration (adjust-repos-time ?in ?er ?ax))
:condition
    (and
      (over all (not (blocked ?in ?ax)))
      (over all (set-up-error ?in ?er))
      (over all (set-up-axis ?in ?ax))
```

```
          (over all (not (measured ?ax ?er)))
          (at start (not (in-operation ?in)))
          (over all (forall (?i - Instrument) (imply (in-operation ?i)
          (compatible ?i ?in))))
          (at start (<=(amount-measured ?ax ?er ?in)(length-to-measure ?ax ?er)))
          (over all (working-day))
        )
    :effect
        (and
          (at end (decrease(amount-measured ?ax ?er ?in)(measurement-overlap ?in)))
        )
    )

    (:durative-action adjust-error
    :parameters (?in - Instrument ?er - Error ?e - Error ?ax - Axis)
    :duration(= ?duration (adjust-error-time ?in ?er ?ax))
    :condition
        (and
          (over all (not (blocked ?in ?ax)))
          (at start (set-up-error ?in ?er))
          (at start (set-up-axis ?in ?ax))
          (at start (not(in-operation ?in)))
          (over all (forall (?i - Instrument) (imply (in-operation ?i)
          (compatible ?i ?in))))
          (over all (>=(amount-measured ?ax ?er ?in)(length-to-measure ?ax ?er)))
          (over all (axis-error ?ax ?e))
          (at start (not (measured ?ax ?e)))
          (over all (measures ?in ?e))
          (over all (working-day))
        )
    :effect
        (and
          (at end (measured ?ax ?er))
          (at end (not(set-up-error ?in ?er)))
          (at end (set-up-error ?in ?e))
          (at end (in-operation ?in))
        )
    )

    (:durative-action remove
    :parameters (?in - Instrument ?er - Error ?ax - Axis)
    :duration(= ?duration (removal-time ?in ?er ?ax))
    :condition
        (and
          (at start (set-up-error ?in ?er))
          (at start (set-up-axis ?in ?ax))
          (at start (not(in-operation ?in)))
          (at start (>=(amount-measured ?ax ?er ?in)(length-to-measure ?ax ?er)))
          (over all (working-day))
        )
```

```
    :effect
        (and
          (at end (measured ?ax ?er))
          (at end (not(set-up-error ?in ?er)))
          (at end (not(set-up-axis ?in ?ax)))
          (at start (decrease (using ?in) 1))
        )
    )


    (:durative-action measure
    :parameters (?er - Error ?in - Instrument ?ax - Axis )
    :duration(= ?duration (measurement-time ?in ?er ?ax))
    :condition
        (and
          (at start (in-operation ?in))
          (over all (not (measured ?ax ?er)))
          (over all (set-up-error ?in ?er))
          (over all (set-up-axis ?in ?ax))
          (over all (working-day))
        )
    :effect
        (and
          (at end (increase(amount-measured ?ax ?er ?in)(working-range ?in)))
          (at end (not(in-operation ?in)))
          (at start (increase (total-importance) (importance ?ax ?er)))
        )
)
)
```

B.2 Example PDDL Five-Axis Problem

```
(define (problem calibration_time)
    (:domain calibration_domain)
    (:objects
        position pitch roll yaw hztl-s vtcl-s squareness acc-rep - Error
        x y z - Axis
        laser-interferometer electronic-level ballbar - Instrument
    )
    (:init
        (axis-error x position  )
        (axis-error x pitch     )
        (axis-error x roll      )
        (axis-error x yaw       )
        (axis-error x hztl-s    )
        (axis-error x vtcl-s    )
        (axis-error x squareness)
```

```
(axis-error x acc-rep)

(=(importance x position  )200)
(=(importance x pitch     )190)
(=(importance x roll      )180)
(=(importance x yaw       )170)
(=(importance x hztl-s    )160)
(=(importance x vtcl-s    )150)
(=(importance x squareness)140)
(=(importance x acc-rep   )195)

(axis-error y position  )
(axis-error y pitch     )
(axis-error y roll      )
(axis-error y yaw       )
(axis-error y hztl-s    )
(axis-error y vtcl-s    )
(axis-error y squareness)
(axis-error y acc-rep)

(=(importance y position  )200)
(=(importance y pitch     )190)
(=(importance y roll      )180)
(=(importance y yaw       )170)
(=(importance y hztl-s    )160)
(=(importance y vtcl-s    )150)
(=(importance y squareness)140)
(=(importance y acc-rep   )195)

(axis-error z position  )
(axis-error z pitch     )
(axis-error z roll      )
(axis-error z yaw       )
(axis-error z hztl-s    )
(axis-error z vtcl-s    )
(axis-error z squareness)
(axis-error z acc-rep)

(=(importance z position  )200)
(=(importance z pitch     )190)
(=(importance z roll      )180)
(=(importance z yaw       )170)
(=(importance z hztl-s    )160)
(=(importance z vtcl-s    )150)
(=(importance z squareness)140)
(=(importance z acc-rep   )195)

(measures laser-interferometer position)
(measures laser-interferometer pitch)
(measures electronic-level roll)
```

```
(measures laser-interferometer yaw)
(measures laser-interferometer hztl-s )
(measures laser-interferometer vtcl-s )
(measures ballbar squareness)
(measures laser-interferometer acc-rep)

(=(setup-time laser-interferometer position x )30)
(=(measurement-time laser-interferometer position x )60)
(=(adjust-error-time laser-interferometer position x )40)
(=(adjust-repos-time laser-interferometer position x )10)
(=(removal-time laser-interferometer position x )10)
(=(feedrate x position laser-interferometer )100)
(=(targets x position laser-interferometer )100)
(=(dwell x position laser-interferometer )2)

(=(setup-time laser-interferometer pitch x )30)
(=(measurement-time laser-interferometer pitch x )60)
(=(adjust-error-time laser-interferometer pitch x )40)
(=(adjust-repos-time laser-interferometer pitch x )10)
(=(removal-time laser-interferometer pitch x )10)
(=(feedrate x pitch laser-interferometer )100)
(=(targets x pitch laser-interferometer )100)
(=(dwell x pitch laser-interferometer )2)

(=(setup-time laser-interferometer roll x )30)
(=(measurement-time electronic-level roll x )60)
(=(adjust-error-time electronic-level roll x )40)
(=(adjust-repos-time electronic-level roll x )10)
(=(removal-time electronic-level roll x )10)
(=(feedrate x roll electronic-level )100)
(=(targets x roll electronic-level )100)
(=(dwell x roll electronic-level )2)

(=(setup-time electronic-level yaw x )30)
(=(measurement-time laser-interferometer yaw x )60)
(=(adjust-error-time laser-interferometer yaw x )40)
(=(adjust-repos-time laser-interferometer yaw x )10)
(=(removal-time laser-interferometer yaw x )10)
(=(feedrate x yaw electronic-level )100)
(=(targets x yaw electronic-level )100)
(=(dwell x yaw electronic-level )2)

(=(setup-time laser-interferometer hztl-s x )30)
(=(measurement-time laser-interferometer hztl-s  x )60)
(=(adjust-error-time laser-interferometer hztl-s  x )40)
(=(adjust-repos-time laser-interferometer hztl-s  x )10)
(=(removal-time laser-interferometer hztl-s  x )10)
(=(feedrate x hztl-s  laser-interferometer )100)
(=(targets x hztl-s  laser-interferometer )100)
(=(dwell x hztl-s laser-interferometer )2)
```

```
(=(setup-time laser-interferometer vtcl-s x )30)
(=(measurement-time laser-interferometer vtcl-s  x )60)
(=(adjust-error-time laser-interferometer vtcl-s  x )40)
(=(adjust-repos-time laser-interferometer vtcl-s  x )10)
(=(removal-time laser-interferometer vtcl-s  x )10)
(=(feedrate x vtcl-s  laser-interferometer )100)
(=(targets x vtcl-s  laser-interferometer )100)
(=(dwell x vtcl-s laser-interferometer )2)

(=(setup-time laser-interferometer squareness  x )30)
(=(measurement-time ballbar squareness x )60)
(=(adjust-error-time ballbar squareness x )40)
(=(adjust-repos-time ballbar squareness x )10)
(=(removal-time ballbar squareness x )10)
(=(feedrate x squareness  ballbar )100)
(=(targets x squareness  ballbar )100)
(=(dwell x squareness ballbar )2)

(=(setup-time ballbar acc-rep x )30)
(=(measurement-time laser-interferometer acc-rep x )60)
(=(adjust-error-time laser-interferometer acc-rep x )40)
(=(adjust-repos-time laser-interferometer acc-rep x )10)
(=(removal-time laser-interferometer acc-rep x )10)
(=(feedrate x acc-rep  laser-interferometer )100)
(=(targets x acc-rep  laser-interferometer )100)
(=(dwell x acc-rep laser-interferometer )2)

(=(setup-time laser-interferometer position y )30)
(=(measurement-time laser-interferometer position y )60)
(=(adjust-error-time laser-interferometer position y )40)
(=(adjust-repos-time laser-interferometer position y )10)
(=(removal-time laser-interferometer position y )10)
(=(feedrate y position laser-interferometer )100)
(=(targets y position laser-interferometer )100)
(=(dwell y position laser-interferometer )2)

(=(setup-time laser-interferometer pitch y )30)
(=(measurement-time laser-interferometer pitch y )60)
(=(adjust-error-time laser-interferometer pitch y )40)
(=(adjust-repos-time laser-interferometer pitch y )10)
(=(removal-time laser-interferometer pitch y )10)
(=(feedrate y pitch laser-interferometer )100)
(=(targets y pitch laser-interferometer )100)
(=(dwell y pitch laser-interferometer )2)

(=(setup-time laser-interferometer roll y )30)
(=(measurement-time electronic-level roll y )60)
(=(adjust-error-time electronic-level roll y )40)
(=(adjust-repos-time electronic-level roll y )10)
```

```
(=(removal-time electronic-level roll y )10)
(=(feedrate y roll electronic-level )100)
(=(targets y roll electronic-level )100)
(=(dwell y roll electronic-level )2)

(=(setup-time electronic-level yaw y )30)
(=(measurement-time laser-interferometer yaw y )60)
(=(adjust-error-time laser-interferometer yaw y )40)
(=(adjust-repos-time laser-interferometer yaw y )10)
(=(removal-time laser-interferometer yaw y )10)
(=(feedrate y yaw electronic-level )100)
(=(targets y yaw electronic-level )100)
(=(dwell y yaw electronic-level )2)

(=(setup-time laser-interferometer hztl-s y )30)
(=(measurement-time laser-interferometer hztl-s  y )60)
(=(adjust-error-time laser-interferometer hztl-s  y )40)
(=(adjust-repos-time laser-interferometer hztl-s  y )10)
(=(removal-time laser-interferometer hztl-s  y )10)
(=(feedrate y hztl-s  laser-interferometer )100)
(=(targets y hztl-s  laser-interferometer )100)
(=(dwell y hztl-s laser-interferometer )2)

(=(setup-time laser-interferometer vtcl-s y )30)
(=(measurement-time laser-interferometer vtcl-s  y )60)
(=(adjust-error-time laser-interferometer vtcl-s  y )40)
(=(adjust-repos-time laser-interferometer vtcl-s  y )10)
(=(removal-time laser-interferometer vtcl-s  y )10)
(=(feedrate y vtcl-s  laser-interferometer )100)
(=(targets y vtcl-s  laser-interferometer )100)
(=(dwell y vtcl-s laser-interferometer )2)

(=(setup-time laser-interferometer squareness  y )30)
(=(measurement-time ballbar squareness y )60)
(=(adjust-error-time ballbar squareness y )40)
(=(adjust-repos-time ballbar squareness y )10)
(=(removal-time ballbar squareness y )10)
(=(feedrate y squareness  ballbar )100)
(=(targets y squareness  ballbar )100)
(=(dwell y squareness ballbar )2)

(=(setup-time ballbar acc-rep y )30)
(=(measurement-time laser-interferometer acc-rep y )60)
(=(adjust-error-time laser-interferometer acc-rep y )40)
(=(adjust-repos-time laser-interferometer acc-rep y )10)
(=(removal-time laser-interferometer acc-rep y )10)
(=(feedrate y acc-rep  laser-interferometer )100)
(=(targets y acc-rep  laser-interferometer )100)
(=(dwell y acc-rep laser-interferometer )2)
```

```
(=(setup-time laser-interferometer position z )30)
(=(measurement-time laser-interferometer position z )60)
(=(adjust-error-time laser-interferometer position z )40)
(=(adjust-repos-time laser-interferometer position z )10)
(=(removal-time laser-interferometer position z )10)
(=(feedrate z position laser-interferometer )100)
(=(targets z position laser-interferometer )100)
(=(dwell z position laser-interferometer )2)

(=(setup-time laser-interferometer pitch z )30)
(=(measurement-time laser-interferometer pitch z )60)
(=(adjust-error-time laser-interferometer pitch z )40)
(=(adjust-repos-time laser-interferometer pitch z )10)
(=(removal-time laser-interferometer pitch z )10)
(=(feedrate z pitch laser-interferometer )100)
(=(targets z pitch laser-interferometer )100)
(=(dwell z pitch laser-interferometer )2)

(=(setup-time laser-interferometer roll z )30)
(=(measurement-time electronic-level roll z )60)
(=(adjust-error-time electronic-level roll z )40)
(=(adjust-repos-time electronic-level roll z )10)
(=(removal-time electronic-level roll z )10)
(=(feedrate z roll electronic-level )100)
(=(targets z roll electronic-level )100)
(=(dwell z roll electronic-level )2)

(=(setup-time electronic-level yaw z )30)
(=(measurement-time laser-interferometer yaw z )60)
(=(adjust-error-time laser-interferometer yaw z )40)
(=(adjust-repos-time laser-interferometer yaw z )10)
(=(removal-time laser-interferometer yaw z )10)
(=(feedrate z yaw electronic-level )100)
(=(targets z yaw electronic-level )100)
(=(dwell z yaw electronic-level )2)

(=(setup-time laser-interferometer hztl-s z )30)
(=(measurement-time laser-interferometer hztl-s  z )60)
(=(adjust-error-time laser-interferometer hztl-s  z )40)
(=(adjust-repos-time laser-interferometer hztl-s  z )10)
(=(removal-time laser-interferometer hztl-s  z  )10)
(=(feedrate z hztl-s  laser-interferometer )100)
(=(targets z hztl-s  laser-interferometer )100)
(=(dwell z hztl-s laser-interferometer )2)

(=(setup-time laser-interferometer vtcl-s z )30)
(=(measurement-time laser-interferometer vtcl-s  z )60)
(=(adjust-error-time laser-interferometer vtcl-s  z )40)
(=(adjust-repos-time laser-interferometer vtcl-s  z )10)
(=(removal-time laser-interferometer vtcl-s  z )10)
```

```
(=(feedrate z vtcl-s  laser-interferometer )100)
(=(targets z vtcl-s  laser-interferometer )100)
(=(dwell z vtcl-s laser-interferometer )2)

(=(setup-time laser-interferometer squareness  z )30)
(=(measurement-time ballbar squareness z )60)
(=(adjust-error-time ballbar squareness z )40)
(=(adjust-repos-time ballbar squareness z )10)
(=(removal-time ballbar squareness z )10)
(=(feedrate z squareness  ballbar )100)
(=(targets z squareness  ballbar )100)
(=(dwell z squareness ballbar )2)

(=(setup-time ballbar acc-rep z )30)
(=(measurement-time laser-interferometer acc-rep z )60)
(=(adjust-error-time laser-interferometer acc-rep z )40)
(=(adjust-repos-time laser-interferometer acc-rep z )10)
(=(removal-time laser-interferometer acc-rep z )10)
(=(feedrate z acc-rep  laser-interferometer )100)
(=(targets z acc-rep  laser-interferometer )100)
(=(dwell z acc-rep laser-interferometer )2)

(compatible laser-interferometer electronic-level)
(compatible electronic-level laser-interferometer)

;;workin range
(= (working-range laser-interferometer) 100)
(= (working-range ballbar) 100)
(= (working-range electronic-level) 100)

(=(measurement-overlap laser-interferometer) 5)
(=(measurement-overlap ballbar) 5)
(=(measurement-overlap electronic-level) 5)

(= (length-to-measure x position)  100)
(= (length-to-measure x pitch)  100)
(= (length-to-measure x roll)  100)
(= (length-to-measure x yaw)  100)
(= (length-to-measure x hztl-s)  100)
(= (length-to-measure x vtcl-s)  100)
(= (length-to-measure x squareness)  100)
(= (length-to-measure x acc-rep)  100)

(= (length-to-measure y position)  100)
(= (length-to-measure y pitch)  100)
(= (length-to-measure y roll)  100)
(= (length-to-measure y yaw)  100)
(= (length-to-measure y hztl-s)  100)
(= (length-to-measure y vtcl-s)  100)
(= (length-to-measure y squareness)  100)
```

```
        (= (length-to-measure y acc-rep)  100)

        (= (length-to-measure z position)  100)
        (= (length-to-measure z pitch)  100)
        (= (length-to-measure z roll)  100)
        (= (length-to-measure z yaw)  100)
        (= (length-to-measure z hztl-s)  100)
        (= (length-to-measure z vtcl-s)  100)
        (= (length-to-measure z squareness)  100)
        (= (length-to-measure z acc-rep)  100)

        (= (using laser-interferometer) 0)
        (= (using electronic-level)     0)
        (= (using ballbar)              0)

        (= (dof laser-interferometer)  1)
        (= (dof electronic-level)      1)
        (= (dof ballbar)               1)

        (= (total-importance) 0)

        (working-day)
        (at 480 (not (working-day)))
        (at 1000 (working-day))
        (at 1480 (not (working-day)))
        (at 2000 (working-day))
        (at 2480 (not (working-day)))
        (at 3000 (working-day))
    )
    (:goal
        (and
        (measured x position  )
        (measured x pitch     )
        (measured x roll      )
        (measured x yaw       )
        (measured x hztl-s    )
        (measured x vtcl-s    )
        (measured x squareness)
        (measured x acc-rep)

        (measured y position  )
        (measured y pitch     )
        (measured y roll      )
        (measured y yaw       )
        (measured y hztl-s    )
        (measured y vtcl-s    )
        (measured y squareness)
        (measured y acc-rep)

        (measured z position  )
```

```
        (measured z pitch     )
        (measured z roll      )
        (measured z yaw       )
        (measured z hztl-s    )
        (measured z vtcl-s    )
        (measured z squareness)
        (measured z acc-rep)
        )
    )
(:metric minimize total-time)
)
```

Appendix C

PDDL2.1 Uncertainty of Measurement Optimisation Model

C.1 PDDL Domain

```
(define (domain calibration_domain)
 (:requirements :strips :fluents :typing :timed-initial-literals
  :negative-preconditions :durative-actions)
   (:types
       Error - object
       Instrument - object
       Axis - object
   )

   (:predicates
       (axis-error ?axi - Axis ?err - Error)
       (measures ?in - Instrument ?err - Error)
       (measured ?ax - Axis ?err - Error)
       (in-operation)
       (set-up-axis ?in - Instrument ?axi - Axis)
       (set-up-error ?in - Instrument ?err - Error)
       (compatible ?ins1 ?ins2 - Instrument)
       (blocked    ?in - Instrument ?ax - Axis)
       (curr_profile ?p - Profile)
       (not_curr_profile ?p - Profile)
       (influencing-error ?ax - Axis ?er - Error ?err - Error)
       (no-influencing ?ax - Axis ?er - Error)
       (ready-to-meta ?in - Instrument ?ax - Axis ?er - Error)
       (ready-to-remove ?in - Instrument ?ax - Axis ?er - Error)
       (start0)
       (start1)
       (start2)
       (start3)
```

```
        (start4)
        (start5)
        (start6)
        (start7)
        (start8)
        (start9)
        (start10)
        (start11)
        (start12)
        (start13)
        (start14)
        (start15)
        (start16)
        (clip-started)
    )

    (:functions
        (using ?in - Instrument)
        (length-to-measure  ?ax - Axis ?er - Error)
        (measurement-time ?in - Instrument ?er - Error ?ax - Axis)
        (setup-time ?in - Instrument ?er - Error ?ax - Axis)
        (adjust-error-time ?in - Instrument ?er - Error ?ax - Axis)
        (adjust-repos-time ?in - Instrument ?er - Error ?ax - Axis)
        (removal-time ?in - Instrument ?er - Error ?ax - Axis)
        (importance ?ax - axis ?er - error)
        (total-importance)
        (max-concurrent)
        (number-meas)

        ;temperature
        (start-temp)
        (temp)
        (biggest-temp)
        (rate)
        (caled-rate)

        (test-u)
        (max-val ?ax - Axis ?er - Error)      ;;Max permissable error when unknown
        (error-val ?ax - Axis ?er - Error)    ;;error when unknown
        (contribution)
        ;uncertainty fluents
        (k_value ?in - Instrument)     ;Instrument K value
        (u_calib ?in - Instrument)     ;Instrument calibration value
        (u_missal ?in - Instrument)     ;Instrument misalignment
        (twosrtthree)     ;2 squareroot 3 (precomputed)
        (u_t-m-d)     ;Uncertainty of the temperature measurement device
        (u_m-d)     ;Uncertaity of temperature measurement device
        (u_d-c)     ;Difference to 20 degrees C
        (u_t-e-c)    ;Thermal expansion coefficient
        (u_d-t-e-c ?in - Instrument)    ;Device expansion coefficient
```

```
        (u_eve)    ;Enviromental variation

        (temp-u)
        (u_c)
 )

    (:durative-action setup
    :parameters (?in - Instrument ?er - Error ?ax - Axis)
    :duration(= ?duration (setup-time ?in ?er ?ax))
    :condition
        (and (at start (not (blocked ?in ?ax)))
            (at start (axis-error ?ax ?er))
            (at start (measures ?in ?er))
            (at start (not (measured ?ax ?er)))
            (over all (not (set-up-error ?in ?er)))
            (over all (not (set-up-axis ?in ?ax)))
            (over all (not (ready-to-remove ?in ?ax ?er)))
            (over all (not (in-operation)))
            (over all (<=(number-meas)(max-concurrent)))
            (over all(clip-started))
        )
        :effect
        (and
            (at end (set-up-error ?in ?er))
            (at end (set-up-axis ?in ?ax))
            (at end (in-operation))
            (at start (increase (number-meas) 1))
        )
    )


    (:durative-action adjust-error
    :parameters (?in - Instrument ?er - Error ?er2 - Error ?ax - Axis)
    :duration(= ?duration (adjust-error-time ?in ?er ?ax))
    :condition
        (and
            (over all (not (ready-to-meta ?in ?ax ?er)))
            (at start(ready-to-remove ?in ?ax ?er))
            (at start (set-up-error ?in ?er))
            (at start (set-up-axis ?in ?ax))
            (over all (not (measured ?ax ?er)))
            (over all(clip-started))
            (over all (axis-error ?ax ?er2))
            (at start (measures ?in ?er2))
            (at start (not (measured ?ax ?er2)))
            (over all (not (blocked ?in ?ax)))
        )
    :effect
        (and
            (at end (not(ready-to-remove ?in ?ax ?er)))
```

```
        (at end (measured ?ax ?er))
        (at end (not(set-up-error ?in ?er)))
        (at end (set-up-error ?in ?er2))
    )
)


(:durative-action measure-influcence
:parameters (?er - Error ?in - Instrument ?ax - Axis ?e1 - Error ?e2 - Error)
:duration(= ?duration (measurement-time ?in ?er ?ax))
:condition
    (and
        (over all (in-operation))
        (over all (not (measured ?ax ?er)))
        (over all (set-up-error ?in ?er))
        (over all (set-up-axis ?in ?ax))
        (over all (influencing-error ?ax ?er ?e1))
        (over all (influencing-error ?ax ?er ?e2))
        (over all (measured ?ax ?e1))
        (over all(clip-started))
        (at start (not (ready-to-meta ?in ?ax ?er)))
    )
:effect
    (and
        (at start (assign(start-temp)(temp)))
        (at end(increase(u_c)(temp-u)))(at end(increase(u_c)(temp-u)))
        (at start(assign(temp-u)
                ;calculate u_device using the length to measure
                    (+(*(/(k_value ?in)(*(u_calib ?in)
                    (length-to-measure ?ax ?er)))(/(k_value ?in)
                    (*(u_calib ?in)(length-to-measure ?ax ?er))))
                ;calculate u_misalignment
                    (+(*(/(+(error-val ?ax ?e1)(error-val ?ax ?e2))
                    (twosrtthree))(/(+(error-val ?ax ?e1)(error-val ?ax ?e2))
                    (twosrtthree)))
                ;calculate u_m_machine tool
                    (+(*(*(u_t-m-d)(*(length-to-measure ?ax ?er)(u_m-d)))
                    (*(u_t-m-d)(*(length-to-measure ?ax ?er)(u_m-d))))
                ;calculate u_e_machine tool
                    (+(*(*(u_t-e-c)(*(length-to-measure ?ax ?er)(rate)))
                    (*(u_t-e-c)(*(length-to-measure ?ax ?er)(rate))))
                ;calculate u_m_device
                    (+(*(*(u_t-m-d)(*(length-to-measure ?ax ?er)(u_m-d)))
                    (*(u_t-m-d)(*(length-to-measure ?ax ?er)(u_m-d))))
                ;calculate u_e_device
                    (+(*(*(u_d-t-e-c ?in)(*(length-to-measure ?ax ?er)(u_d-c)))
                    (*(u_d-t-e-c ?in)(*(length-to-measure ?ax ?er)(u_d-c))))
                    (u_eve)))))))
        (at start (ready-to-meta ?in ?ax ?er))
        (at start (ready-to-remove ?in ?ax ?er))
```

```
        )
    )

    (:durative-action measure-no-influence
    :parameters (?er - Error ?in - Instrument ?ax - Axis)
    :duration(= ?duration (measurement-time ?in ?er ?ax))
    :condition
        (and
            (over all (in-operation))
            (over all (not (measured ?ax ?er)))
            (over all (set-up-error ?in ?er))
            (over all (set-up-axis ?in ?ax))
            (over all (no-influencing ?ax ?er))
            (over all (clip-started))
            (at start (not (ready-to-meta ?in ?ax ?er)))
        )
    :effect
        (and
            (at start (assign(start-temp)(temp)))
            (at end (increase(u_c)(temp-u)))(at end(increase(u_c)(temp-u)))
            (at start(assign(temp-u)
                    ;calculate u_device using the length to measure
                        (+(*(/(k_value ?in)(*(u_calib ?in)
                        (length-to-measure ?ax ?er)))
                        (/(k_value ?in)(*(u_calib ?in)(length-to-measure ?ax ?er))))
                    ;calculate u_misalignment
                        (+(*(/(u_missal ?in)(twosrtthree))(/(u_missal ?in)
                        (twosrtthree)))
                    ;calculate u_m_machine tool
                        (+(*(*(u_t-m-d)(*(length-to-measure ?ax ?er)(u_m-d)))
                        (*(u_t-m-d)(*(length-to-measure ?ax ?er)(u_m-d))))
                    ;calculate u_e_machine tool
                        (+(*(*(u_t-e-c)(*(length-to-measure ?ax ?er)(rate)))
                        (*(u_t-e-c) (*(length-to-measure ?ax ?er)(rate))))
                    ;calculate u_m_device
                        (+(*(*(u_t-m-d)(*(length-to-measure ?ax ?er)(u_m-d)))
                        (*(u_t-m-d)(*(length-to-measure ?ax ?er)(u_m-d))))
                    ;calculate u_e_device
                        (+(*(*(u_d-t-e-c ?in)(*(length-to-measure ?ax ?er)(u_d-c)))
                        (*(u_d-t-e-c ?in)(*(length-to-measure ?ax ?er)(u_d-c))))
                        (u_eve)))))))
            (at start (ready-to-meta ?in ?ax ?er))
            (at start (ready-to-remove ?in ?ax ?er))
        )
    )

    ;;Temperature dependent effects must take place here
    (:durative-action meta
    :parameters (?in - Instrument ?er - Error ?ax - Axis)
    :duration(= ?duration 10)
```

```
    :condition  (and (at start (ready-to-meta ?in ?ax ?er)))
    :effect
        (and
            (at start (not (ready-to-meta ?in ?ax ?er)))
            (at start (increase(u_c)(+(temp-u)
                    (*(*(u_t-e-c)(*(length-to-measure ?ax ?er)
                        (+(start-temp)(temp))))(*(u_t-e-c)
                        (*(length-to-measure ?ax ?er)(+(start-temp)(temp))))))))
            (at start (decrease (error-val ?ax ?er)(-(error-val ?ax ?er)(+(temp-u)
                    (*(*(u_t-e-c)(*(length-to-measure ?ax ?er)
                        (+(start-temp)(temp))))(*(u_t-e-c)
                        (*(length-to-measure ?ax ?er)(+(start-temp)(temp)))))))))

            )
    )

    (:durative-action remove
    :parameters (?in - Instrument ?er - Error ?ax - Axis)
    :duration(= ?duration (removal-time ?in ?er ?ax))
    :condition
        (and
            (over all (not (ready-to-meta ?in ?ax ?er)))
            (at start(ready-to-remove ?in ?ax ?er))
            (at start (set-up-error ?in ?er))
            (at start (set-up-axis ?in ?ax))
            (over all (not (measured ?ax ?er)))
            (over all(clip-started))
        )
    :effect
        and
            (at end (not(ready-to-remove ?in ?ax ?er)))
            (at end (measured ?ax ?er))
            (at end (not(set-up-error ?in ?er)))
            (at end (not(set-up-axis ?in ?ax)))
            (at end (decrease (number-meas) 1))
            (at end (not (in-operation)))
        )
  )

    (:durative-action temp-profile0
    :parameters ()
    :duration(= ?duration 42.0)
    :condition
       (and (at start (start0)))
    :effect
       (and (at start (assign (rate)0.00595))
       (at end (not(start0)))
       (at end (start1))
       (at start (clip-started) )))
```

```
(durative-action temp-profile1
:parameters ()
:duration(= ?duration 51.0)
:condition
   (and (over all (not(start0)))(over all (start1)))
:effect
   (and (at start (assign (rate)0.0049))
   (at end (not(start1)))
   (at end (start2))))


(:durative-action temp-profile2
:parameters ()
:duration(= ?duration 56.0)
:condition
   (and (over all (not(start1)))(over all(start2)))
:effect
   (and (at start (assign (rate)0.00446))
   (at end (not(start2)))
   (at end (start3))))


(:durative-action temp-profile3
:parameters ()
:duration(= ?duration 145.0)
:condition
   (and (over all (not(start2)))(over all (start3)))
:effect
   (and (at start (assign (rate)0.00172))
   (at end (not(start3)))
   (at end (start4))))


(:durative-action temp-profile4
:parameters ()
:duration(= ?duration 27.0)
:condition
   (and (over all (not(start3)))(over all (start4)))
:effect
   (and (at start (assign (rate)-0.00926))
   (at end (not(start4)))
   (at end (start5))))


(:durative-action temp-profile5
:parameters ()
:duration(= ?duration 8.0)
:condition
   (and (over all (not(start4)))(over all (start5)))
```

```
:effect
   (and (at start (assign (rate)0.03125))
   (at end (not(start5)))
   (at end (start6))))


(:durative-action temp-profile6
:parameters ()
:duration(= ?duration 87.0)
:condition
   (and (over all (not(start5)))(over all(start6)))
:effect
   (and (at start (assign (rate)0.00287))
   (at end (not(start6)))
   (at end (start7))))


(:durative-action temp-profile7
:parameters ()
:duration(= ?duration 113.0)
:condition
   (and (over all (not(start6)))(over all (start7)))
:effect
   (and (at start (assign (rate)0.00221))
   (at end (not(start7)))
   (at end (start8))))


(:durative-action temp-profile8
:parameters ()
:duration(= ?duration 117.0)
:condition
   (and (over all(not(start7)))(over all(start8)))
:effect
   (and (at start (assign (rate)0.00214))
   (at end (not(start8)))
   (at end (start9))))


(:durative-action temp-profile9
:parameters ()
:duration(= ?duration 348.0)
:condition
   (and (over all (not(start8)))(over all (start9)))
:effect
   (and (at start (assign (rate)-0.00072))
   (at end (not(start9)))
   (at end (start10))))
```

```
(:durative-action temp-profile10
:parameters ()
:duration(= ?duration 474.0)
:condition
   (and (over all(not(start9)))(over all(start10)))
:effect
   (and (at start (assign (rate)0.00053))
   (at end (not(start10)))
   (at end (start11))))


(:durative-action temp-profile11
:parameters ()
:duration(= ?duration 97.0)
:condition
   (and (over all (not(start10)))(over all (start11)))
:effect
   (and (at start (assign (rate)0.00258))
   (at end (not(start11)))
   (at end (start12))))


(:durative-action temp-profile12
:parameters ()
:duration(= ?duration 171.0)
:condition
   (and (over all (not(start11)))(over all (start12)))
:effect
   (and (at start (assign (rate)0.00146))
   (at end (not(start12)))
   (at end (start13))))


(:durative-action temp-profile13
:parameters ()
:duration(= ?duration 162.0)
:condition
   (and (over all (not(start12)))(over all (start13)))
:effect
   (and (at start (assign (rate)0.00154))
   (at end (not(start13)))
   (at end (start14))))


(:durative-action temp-profile14
:parameters ()
:duration(= ?duration 582.0)
:condition
   (and (over all (not(start13)))(over all (start14)))
:effect
```

```
      (and (at start (assign (rate)-0.00043))
      (at end (not(start14)))
      (at end (start15))))


   (:durative-action temp-profile15
   :parameters ()
   :duration(= ?duration 386.0)
   :condition
      (and (over all (not(start14)))(over all (start15)))
   :effect
      (and (at start (assign (rate)0.00049))
      (at end (not(start15)))
      (at end (start16))))
)
```

C.2 Example Three-Axis Problem

```
(define (problem one)
   (:domain calibration_domain)
   (:objects
   position pitch roll yaw hztl-s vtcl-s squareness - Error
   x y z - Axis
   laser-interferometer electronic-level ballbar - Instrument
   p0 p1 p2 p3 p4 p5 p6 p7 p8 p9 p10 p11 p12 p13 p14 p15 p16 p17 p18 - Profile
   )
   (:init
      (= (length-to-measure x position) 1500)
      (= (length-to-measure x pitch) 1500)
      (= (length-to-measure x roll) 1500)
      (= (length-to-measure x yaw) 1500)
      (= (length-to-measure x hztl-s) 1500)
      (= (length-to-measure x vtcl-s) 1500)
      (= (length-to-measure x squareness) 1500)

      (= (length-to-measure y position) 1700)
      (= (length-to-measure y pitch) 1700)
      (= (length-to-measure y roll) 1700)
      (= (length-to-measure y yaw) 1700)
      (= (length-to-measure y hztl-s) 1700)
      (= (length-to-measure y vtcl-s) 1700)
      (= (length-to-measure y squareness) 1700)

      (= (length-to-measure z position) 1000)
      (= (length-to-measure z pitch) 1000)
      (= (length-to-measure z roll) 1000)
      (= (length-to-measure z yaw) 1000)
      (= (length-to-measure z hztl-s) 1000)
```

```
(= (length-to-measure z vtcl-s) 1000)
(= (length-to-measure z squareness) 1000)

(axis-error x position  )
(axis-error x pitch     )
(axis-error x roll      )
(axis-error x yaw       )
(axis-error x hztl-s    )
(axis-error x vtcl-s    )
(axis-error x squareness)

(=(importance x position  )23)
(=(importance x pitch     )20)
(=(importance x roll      )40)
(=(importance x yaw       )30)
(=(importance x hztl-s    )28)
(=(importance x vtcl-s    )90)
(=(importance x squareness)14)

(axis-error y position  )
(axis-error y pitch     )
(axis-error y roll      )
(axis-error y yaw       )
(axis-error y hztl-s    )
(axis-error y vtcl-s    )
(axis-error y squareness)

(=(importance y position  )23)
(=(importance y pitch     )20)
(=(importance y roll      )40)
(=(importance y yaw       )30)
(=(importance y hztl-s    )28)
(=(importance y vtcl-s    )90)
(=(importance y squareness)14)

(axis-error z position  )
(axis-error z pitch     )
(axis-error z roll      )
(axis-error z yaw       )
(axis-error z hztl-s    )
(axis-error z vtcl-s    )
(axis-error z squareness)

(=(importance z position  )23)
(=(importance z pitch     )20)
(=(importance z roll      )40)
(=(importance z yaw       )30)
(=(importance z hztl-s    )28)
(=(importance z vtcl-s    )90)
(=(importance z squareness)14)
```

```
(measures laser-interferometer position)
(measures laser-interferometer pitch)
(measures electronic-level roll)
(measures laser-interferometer yaw)
(measures laser-interferometer hztl-s )
(measures laser-interferometer vtcl-s )
(measures ballbar squareness)

(=(setup-time laser-interferometer position x )54)
(=(measurement-time laser-interferometer position x )45)
(=(removal-time laser-interferometer position x )0.01)
(=(adjust-error-time  laser-interferometer position x )44)

(=(setup-time laser-interferometer pitch x )54)
(=(measurement-time laser-interferometer pitch x )40)
(=(removal-time laser-interferometer pitch x )0.01)
(=(adjust-error-time  laser-interferometer pitch x )44)

(=(setup-time electronic-level roll x        )54)
(=(measurement-time electronic-level roll x )45)
(=(removal-time electronic-level roll x )0.01)
(=(adjust-error-time  electronic-level roll x )30)

(=(setup-time laser-interferometer yaw x )54)
(=(measurement-time laser-interferometer yaw x )30)
(=(removal-time laser-interferometer yaw x )0.01)
(=(adjust-error-time laser-interferometer yaw x )44)

(=(setup-time laser-interferometer hztl-s x )54)
(=(measurement-time laser-interferometer hztl-s  x )25)
(=(removal-time laser-interferometer hztl-s  x )0.01)
(=(adjust-error-time laser-interferometer hztl-s x )44)

(=(setup-time laser-interferometer vtcl-s x )54)
(=(measurement-time laser-interferometer vtcl-s  x )25)
(=(removal-time laser-interferometer vtcl-s  x )0.01)
(=(adjust-error-time laser-interferometer vtcl-s  x )44)

(=(setup-time ballbar squareness  x )60)
(=(measurement-time ballbar squareness x )30)
(=(removal-time ballbar squareness x )0.01)
(=(adjust-error-time ballbar squareness x  )30)

(=(setup-time laser-interferometer position y )54)
(=(measurement-time laser-interferometer position y )45)
(=(removal-time laser-interferometer position y )0.01)
(=(adjust-error-time  laser-interferometer position y )44)

(=(setup-time laser-interferometer pitch y )54)
```

```
(=(measurement-time laser-interferometer pitch y )40)
(=(removal-time laser-interferometer pitch y )0.01)
(=(adjust-error-time  laser-interferometer pitch y )44)

(=(setup-time electronic-level roll y         )54)
(=(measurement-time electronic-level roll y )45)
(=(removal-time electronic-level roll y )0.01)
(=(adjust-error-time  electronic-level roll y )30)

(=(setup-time laser-interferometer yaw y )54)
(=(measurement-time laser-interferometer yaw y )30)
(=(removal-time laser-interferometer yaw y )0.01)
(=(adjust-error-time laser-interferometer yaw y )44)

(=(setup-time laser-interferometer hztl-s y )54)
(=(measurement-time laser-interferometer hztl-s  y )25)
(=(removal-time laser-interferometer hztl-s  y )0.01)
(=(adjust-error-time laser-interferometer hztl-s y )44)

(=(setup-time laser-interferometer vtcl-s y )54)
(=(measurement-time laser-interferometer vtcl-s y )25)
(=(removal-time laser-interferometer vtcl-s  y    )0.01)
(=(adjust-error-time laser-interferometer vtcl-s  y )44)

(=(setup-time ballbar squareness  y )60)
(=(measurement-time ballbar squareness y )30)
(=(removal-time ballbar squareness y )0.01)
(=(adjust-error-time ballbar squareness y  )30)

(=(setup-time laser-interferometer position z )54)
(=(measurement-time laser-interferometer position z )45)
(=(removal-time laser-interferometer position z )0.01)
(=(adjust-error-time  laser-interferometer position z )44)

(=(setup-time laser-interferometer pitch z )54)
(=(measurement-time laser-interferometer pitch z )40)
(=(removal-time laser-interferometer pitch z )0.01)
(=(adjust-error-time  laser-interferometer pitch z )44)

(=(setup-time electronic-level roll z         )54)
(=(measurement-time electronic-level roll z )45)
(=(removal-time electronic-level roll z )0.01)
(=(adjust-error-time  electronic-level roll z )30)

(=(setup-time laser-interferometer yaw z )54)
(=(measurement-time laser-interferometer yaw z )30)
(=(removal-time laser-interferometer yaw z )0.01)
(=(adjust-error-time laser-interferometer yaw z )44)

(=(setup-time laser-interferometer hztl-s z )54)
```

```
(=(measurement-time laser-interferometer hztl-s  z )25)
(=(removal-time laser-interferometer hztl-s  z )0.01)
(=(adjust-error-time laser-interferometer hztl-s z )44)

(=(setup-time laser-interferometer vtcl-s z )54)
(=(measurement-time laser-interferometer vtcl-s z )25)
(=(removal-time laser-interferometer vtcl-s  z    )0.01)
(=(adjust-error-time laser-interferometer vtcl-s  z )44)

(=(setup-time ballbar squareness  z )60)
(=(measurement-time ballbar squareness z )30)
(=(removal-time ballbar squareness z )0.01)
(=(adjust-error-time ballbar squareness z  )30)

(compatible laser-interferometer electronic-level)
(compatible electronic-level laser-interferometer)

(= (using laser-interferometer) 0)
(= (using electronic-level) 0)
(= (using ballbar) 0)

(= (total-importance) 0)

(=(temp)0.0)
(curr_profile p0)

(influencing-error x squareness hztl-s)
(influencing-error x squareness vtcl-s)

(no-influencing x position)
(no-influencing x pitch)
(no-influencing x yaw)
(no-influencing x hztl-s)
(no-influencing x vtcl-s)
(no-influencing x roll)

(influencing-error y squareness hztl-s)
(influencing-error y squareness vtcl-s)

(no-influencing y position)
(no-influencing y pitch)
(no-influencing y yaw)
(no-influencing y hztl-s)
(no-influencing y vtcl-s)
(no-influencing y roll)

(influencing-error z squareness hztl-s)
(influencing-error z squareness vtcl-s)

(no-influencing z position)
```

```
(no-influencing z pitch)
(no-influencing z yaw)
(no-influencing z hztl-s)
(no-influencing z vtcl-s)
(no-influencing z roll)

(=(error-val x pitch) 3)
(=(error-val x position) 3)
(=(error-val x roll) 3)
(=(error-val x yaw) 3)
(=(error-val x hztl-s) 3)
(=(error-val x vtcl-s ) 3)
(=(error-val x squareness ) 3)

(=(max-val x pitch) 3)
(=(max-val x position) 3)
(=(max-val x roll) 3)
(=(max-val x yaw) 3)
(=(max-val x hztl-s) 3)
(=(max-val x vtcl-s ) 3)
(=(max-val x squareness ) 3)

(=(error-val y pitch) 3)
(=(error-val y position) 3)
(=(error-val y roll) 3)
(=(error-val y yaw) 3)
(=(error-val y hztl-s) 3)
(=(error-val y vtcl-s ) 3)
(=(error-val y squareness ) 3)

(=(max-val y pitch) 3)
(=(max-val y position) 3)
(=(max-val y roll) 3)
(=(max-val y yaw) 3)
(=(max-val y hztl-s) 3)
(=(max-val y vtcl-s ) 3)
(=(max-val y squareness ) 3)

(=(error-val z pitch) 3)
(=(error-val z position) 3)
(=(error-val z roll) 3)
(=(error-val z yaw) 3)
(=(error-val z hztl-s) 3)
(=(error-val z vtcl-s ) 3)
(=(error-val z squareness ) 3)

(=(max-val z pitch) 3)
(=(max-val z position) 3)
(=(max-val z roll) 3)
(=(max-val z yaw) 3)
```

```
(=(max-val z hztl-s) 3)
(=(max-val z vtcl-s ) 3)
(=(max-val z squareness ) 3)

(= (u_calib laser-interferometer) 1.5)
(= (u_calib electronic-level) 1.5)
(= (u_calib ballbar) 1.5)

(=(twosrtthree)0.6)
(=(u_t-m-d)0.008)
(=(u_m-d)0.008)
(=(u_d-c)0.008)
(=(u_t-e-c)0.008)

;;linear positioning
(=(k_value laser-interferometer)2)
(=(u_calib laser-interferometer)0.003)
(=(u_missal laser-interferometer)0.003)
(=(u_d-t-e-c laser-interferometer)0.003)

(=(k_value electronic-level)2)
(=(u_calib electronic-level)0.003)
(=(u_missal electronic-level)0.003)
(=(u_d-t-e-c electronic-level)0.003)

(=(k_value ballbar)2)
(=(u_calib ballbar)0.003)
(=(u_missal ballbar)0.003)
(=(u_d-t-e-c ballbar)0.003)

(start0)
(start1)
(start2)
(start3)
(start4)
(start5)
(start6)
(start7)
(start8)
(start9)
(start10)
(start11)
(start12)
(start13)
(start14)
(start15)
(start16)

(=(max-concurrent)1)
(=(number-meas)0)
```

```
        (=(start-temp)0)
        (=(rate)0)
        (=(temp-u)0)
        (=(u_c)0)
        (=(u_eve)0)
        (not(in-operation))
    )
    (:goal
        (and
             (measured x position  )
             (measured x pitch     )
             (measured x roll      )
             (measured x yaw       )
             (measured x hztl-s    )
             (measured x vtcl-s    )
             (measured x squareness)

             (measured y position  )
             (measured y pitch     )
             (measured y roll      )
             (measured y yaw       )
             (measured y hztl-s    )
             (measured y vtcl-s    )
             (measured y squareness)

             (measured z position  )
             (measured z pitch     )
             (measured z roll      )
             (measured z yaw       )
             (measured z hztl-s    )
             (measured z vtcl-s    )
             (measured z squareness)
             (not(start15))
        )
    )

(:metric maximize (-(+(total-time)(u_c))(+(+(total-time)(u_c))
(+(total-time)(u_c)))))
;(:metric minimize (total-time))
;(:metric minimize (u_c))
)
```

Appendix D

Uncertainty of Measurement Optimisation Results

Instance	Metric: Time		Metric: Uncertainty		Metric: $\overline{T+U}$	
	Number of plans	Optimal discover time	Number of plans	Optimal discover time	Number of plans	optimal discover time
3A1A	6	8:58	1	7:48	6	9:43
3A1B	5	7:45	1	9:21	5	9:02
3A1C	6	5:10	2	8:08	3	7:26
3A2A	8	9:20	2	8:40	4	8:40
3A2B	5	8:42	2	9:26	5	9:08
3A2C	7	9:47	1	7:14	2	8:19
5A1A	3	9:55	1	8:37	3	9:37
5A1B	2	9:36	1	8:33	1	9:50
5A1C	3	8:23	1	8:48	4	8:09
5A2A	4	7:39	2	9:20	5	9:41
5A2B	2	5:42	2	7:25	2	7.01
5A2C	4	9:00	2	8:08	1	7:51

Table D.1: The number of indentified plans and the discovery time of the optimal

	Metric: Time		Metric: Uncertainty		Metric: $\overline{T+U}$	
Instance	T	U(µm)	T	U(µm)	T	U(µm)
3A1A	32:42	137	34:32	52	32:52	18
3A1B	29:14	197	31:45	49	29:34	138
3A1C	29:15	80	31:57	67	29:15	193
3A2A	30:52	142	32:50	90	31:03	27
3A2B	28:12	135	29:27	89	28:57	82
3A2C	25:56	293	26:41	139	26:05	93
5A1A	51:52	137	63:20	18	52:50	52
5A1B	51:00	271	57:17	138	52:35	63
5A1C	49:05	291	53:38	193	50:11	67
5A2A	47:46	114	49:44	27	47:46	95
5A2B	45:02	89	47:46	82	45:52	112
5A2C	47:22	162	48:32	88	47:46	114

TABLE D.2: Temporal & uncertainty optimisation results (Cluster).