AI Technique for Online Non-Linear Feedrate Scheduling

FLORIN BOGDAN MARIN, MIHAELA BANU, VASILE MARINESCU, ALEXANDRU EPUREANU, IONUT CONSTANTIN
Manufacturing Science and Engineering Department
Dunarea de Jos University
111, Domneasca Street Galati - 800201
ROMANIA
alexandru.epureanu@ugal.ro http://www.tcm.ugal.ro/personal/Epureanu.html

Abstract: - The optimal feedrate scheduling is an important issue so far that concern machine tool performance. Moreover, taking into account the unavoidable variation of detached material thickness the process performance can be increased through online high-resolution optimal feedrate scheduling. Such an approach implies the need for considerable software processing resources. In order to solve this problem, the holonic concept, which is part of distributed artificial intelligence field, is applied for feedrate scheduling. In this paper a specific holonic architecture and algorithm is proposed. The basic structural entity of the holonic architecture is the holon, which is an operating system process acting autonomous and cooperative. By intermediary of the holons actions, according to the holonic algorithm, suitable positioning of cutter edge is obtained. On the other hand, each holon represents an interpolated cutter location. The process sequence between two successive interpolated cutter locations is the resolution of non-linear optimization. The optimization criterion and restrictions are applied independently to each process sequence. By intermediary of simulation the proposed holonic feedrate scheduling algorithm is tested. The results obtained in the case of turning processes confirmed its feasibility and efficiency.

Key-Words: - Holonic algorithms, High-resolution optimization, Feedrate scheduling, Software model, Reconfigurable machine tool, Distributed artificial intelligence.

1 Introduction

The optimal feedrate scheduling is an important issue so far that concern machine tool performance.

Moreover, taking into account the unavoidable variation of detached material thickness the process performance can be increased through online high-resolution non-linear optimal feedrate scheduling. Such an approach implies the need for considerable software processing resources.

The literature underlines in the last years the importance of introducing the distributed intelligence in machine tool control [1]. A relative new concept, called holon, which is an extension of multi-agent concept, is used to deal with different problems.

The traditional concept of holon developed by Arthur Koestler [3] is based on the idea of recursive or self-similar structures in biological systems. Holons, if they are sub-holons, merge into a new holon.

The paradigm of holonic control of a factory have been proposed to address the challenge of developing control systems capable of handling certain types of disturbances at the factory level [4][5][6][7][8]. One of these holonic architecture is ADACOR, which integrates a set of paradigms and technologies for distributed manufacturing systems, to achieve a flexible and adaptive control [2].

We need to develop the cognition-based holons, characterized by instinctual and social behavior associated with the capability of online creating knowledge, of reasoning and applying the results. Like multi-agent systems (MAS), the holonic systems are composed of self-reliant units that are capable of flexible behavior. More specifically, a holon can be thought of as a special type of agent that is characteristically autonomous, cooperative and recursive in the holarchy. This means that a holon can embodies other holons so there is a recursive structure, and this is the fundamental difference compared to an agent.

In order to solve the feederate scheduling problem, based on the holonic concept, we propose a specific modeling method to be applied. In this paper a specific holonic architecture and algorithm are proposed. The basic structural entity of the holonic architecture is the holon, which is an operating system process acting autonomous and cooperative. By intermediary of the holons actions, according to the holonic algorithm, suitable positioning of cutter edge is obtained. On the other hand, each holon is representing an interpolated cutter location.

The process sequence between two successive interpolated cutter locations is the resolution of non-linear optimization. The optimization criterion and restrictions are applied independently to each process sequence.
sequence. By intermediary of simulation the proposed holonic feedrate scheduling algorithm is tested. The results obtained in the case of turning processes confirmed its feasibility and efficiency.

The rest of the paper is organized as it follows. In section two it is defined the high resolution optimization problem, in section three it is presented the proposed holonic algorithm. Section four show simulations performed with the software developed and finally in section five conclusions are drawn.

2 High-resolution optimization

Current cutter location is described by \((i,j)\) pair, while \(i\) is defining the current cutter location and \(j\) is current generation point belonging to cutter profile. Another variable in our optimization program is the variable \(k\), which is the indexing variable for \((i,j)\) pair.

As described below, the reconfigurable machine is configured as lathe with an additional degree of freedom represented by rotation of the tool assembly, with an \(\varphi\) angle, variable that determines \(j\) point belonging to cutter profile.

Our search algorithm is using as reference calculation the previous cutter location \((k-1)\), as in Fig.2, representing the last selected cutter location. Consequently, every \(k\) position is associated with the \(\varphi(k), Z(k)\) and \(X(k)\) variable values.

In the proposed search optimization algorithm we use current \(L\) variable value defining the length of active cutting edge.

![Fig. 1 Optimisation problem](image)

Our objective is to determine \((i,j)\) pair under the criteria of maximizing locally chip area.

While the search criteria in the MRR (Material Removal Rate) and force model are the MRR and the force respectively, in our approach the search criterion is the locally maximized chip area.

In the above two approaches the variable for controlling the process is the feedrate, whereas we are using two distinct variables, cutter location \(CL\) (or feedrate) and cutter profile \(CP\). The cutter profile variable is the additional freedom degree determined by rotary tool assembly.

Concerning the restrictions, in the case of MRR model, is the chip area which is set to a minimum allowable \((A<A_a)\), and in the case of force model, the restriction is the force \((F<F_a)\). Our approach defines four restrictions: \((A_a>A(i,j), Ra>R(i,j), aa>a(i,j), La>L(i,j))\), where \(Ra\) means roughness and \(aa\) means chip thinness.

Online control such as used in force model, needs a monitoring hardware, the proposed optimizing method needs a measurement system, which is exploring every blank.

3 The proposed holonic algorithm

In our approach, holon thinking are defined on the basis of three different features: operations, instinct-laws and knowledge.

Operations are the fundamental actions that are universal, these always runs in the same way, regardless of the task to be performed.

Instinct-laws are the rules governing the holon and indicating the situation in which the various operations are executed.

Knowledge is summarized in several search functions, which is defined by the user and forwarded to holon through a specific language learning.

Primary holon is the first holon existing at the beginning of the search task, from which all holons of the holarchy will inherit all properties. This will create new holons to meet the task undertaken.

We must specify that the new holons created inherit all the knowledge and properties of the primary holon, which loses its identity as primary holon.

Firstly the human defines the search space as the space in which the holons are searching for a solution. For example, the search space can be an image, which is a 2D search space, or may be a search space of states defined by 5 variables.

Further we define the typology of the three constituent features of holon thinking.

Operations typology

1) Operation of unification is the action by which two or more holons are grouped in a holarchy, forming a new holon. It should be noted that the sense of holon is lost, as long as there is not at least another holon as a partner in solving the task.

2) Operation of isolation is the action whereby a holon comes from a holarchy to be partner with the holon of “former members” of holarchy.
3) **Operation of birth** is the action of one holon that creates new holon (the software process will create one or more other processes).

4) **Operation of destruction** is the destruction of its own. Software process is self-killed on its own initiative. This operation occurs when a holon notes that no longer can move in the search space, so reason of existence disappears.

5) **Change of holarchy** is the action of holon moving from a holarchy to another.

6) **Proposal** is a holon action of informing other holon as regard to a proposal.

7) **Accept** is an action of accepting a proposal.

8) **Coordination deployment** is the holon action to occupy a position in the search space, using the functions defined by the user through our proposed specific language.

**Instinct laws typology**

Instinct laws are those laws universally applied, regardless of the search function, which governs holarchy.

We define the following instinct laws:

1) Any open space must be filled.

2) If it is noticed inactivity of itself the holon will execute self-destruction, calling the destruction operation.

3) Where a holon notes that the search space is vacant, and while he can not occupy this space alone, it will born another holon, calling the birth operation.

4) Space occupied by other holon cannot be filled. In this case the need for more space is satisfied only through the unification operation.

5) Operations are running only between neighbors in the search space.

Further we define some of the learning functions, by which the human teaches the holon how to act for reach the solution.

**Knowledge typology**

Knowledge is represented by those functions that are provided by the user. These define the knowledge that informs the user on the task of finding the solution in the search space.

These functions are most of the time an ambiguous description of the reality. This information is passed through a language learning, based on a logic of heuristics. The following functions define some features of the learning language.

1) Function defining the search space.
   Format: **Space.indication of dimension of the space. Function to read space.** where size indication field indicates the size and space to read function.

   For example, Space.2d.GetImage - 2d space is defined which can be read with the read image, the space defined by an image.

   2) Function of reading the current state position.
   Format: **ReadPoz (size 1, size 2 ... size n)** where size 1, size 2 ... size n is the size indication of position in search space.

   For example, ReadPoz (X, Y, Z, K, M) - read the current value of the 5D space.

   3) Function defining the secondary functions of assessment - to define secondary functions that allow analyzing the current position.
   Format: **SecondaryFunction.1 = CalcSurface (P1, P2, .. Pn)** where P1, P2, .. Pn are points.

   For example, calculating the area determined by the m1, m2, m3, m4, m5 or line length determined by the X2, X5.

   SecondaryFunction.2 = CalcLenLine (X2, X5)

   By intermediary of secondary functions it is defined primary function, which is defining the solutions in search space.

   4) Function defining the primary function
   Format:

   **PrimaryFunction=Search.SeconderyFunction.1,Search.SeconderyFunction.2,.. SearchFunction n)**

   For example the primary function is defined as that position for which function 2 result is bigger than the function 5, and also the result of function 1 (suppose that is calculating an area) is maximum. **Search SecondaryFunction.2 > SecondaryFunction.5, SecondaryFunction.1 = MAX;**

**General algorithm of modeling**

The holonic modeling algorithm is defined by the following main steps (Fig.2):

- Deployment is the random placement in the search space of holons and marking positions as part of the territory of themself.
- Further, each holon is calculating position search functions values;
- Calculate the current positions for the functions;
- Operations in the holarchy are executed to occupy positions until all search space was divided between holons. It should be noted that this does not necessarily means that all positions in space were evaluated singular.
Comparison between proposed holonic modeling and genetic algorithm

We further formulate a brief comparison between genetic algorithm and holonic modeling. Though our development it is in early stage, we noted several similarities between the two algorithms.

In case of genetic algorithms, the problem is given as follow: the search space is a space with \( n \) dimensions, and we need to find the point of this space that satisfy a given condition. A solution of the problem is represented by any of the search space. The result is the set of values of the \( n \) coordinates of the point that every condition is best satisfied.

In the case of holonic modeling, we formulate the problem as: giving it a field of action it is required to structure the search space, so that the structural components is satisfying a set of given conditions. The problem is to structure that space, satisfied within the limits of certain conditions. Unlike genetic algorithms, problem solutions can be organized into a rating.

Engine of evolution

In case of genetic algorithms, the engine of evolution is the sequence of operations which constitute loop algorithm. Loop is either covered by a given number of times, after which the algorithm is stopped. The stopping is determined by the compliance of performance conditions. The execution of the loop always is the same, regardless of the state in which the evolutionary process lay. Evolution of the calculation to obtain the problem solution is based on the similarity between the population evolutions of a given species, with the changing environment on the one hand, and on the other hand, the artificial evolution of populations, based on a similar genetic algorithm.

- In the case of holonic modeling the engine of evolution is the instinct of the holon to meet its own best interest by exploring the neighborhood. Also to this is added the operations that ensure the expansion process in the field of action (search space).
- The reason for the evolution is always to find the solution (and not vice versa) is that each decision of the holon, in every moment of the development, is to meet best interests (not worse).

Complementarities of application fields of genetic algorithms and holonic modeling

The comparative analysis of the two techniques currently showed that the proposed holonic modeling proposed in this paper shows specific problems that is suited.

Genetic algorithms can be used for search problems where the search criterion is expressed by the value function and the target is a certain value of it. The most frequently encountered problems is to find the optimum, called optimization problems, where the search is used as objective function and constraints are used to delimit the search space.

Genetic algorithms cannot apply when search condition is expressed by several functions and the target is the compliance with conditions on the values of these functions. Also, genetic algorithms may not be used when the number of components of the solution is not previously known.

Finally, so far that concern the problems on which the search target is not unique (as is the target objective function, for genetic algorithms), but varies depending on the position of search space in which the assessment can not be solved with genetic algorithms. This is because the genetic algorithm use is not feasible because of the extremely large number of solutions that could form the total population.

Holon modeling can be considered as a technique to solve the problems of search, in which the solution is
unknown, the requirements are multiple search, and the search can be limited. Thus, the two techniques of evolutionary computation are complementary, allowing the extension of the range of problems that can be solved. At this time, we cannot make a comparative assessment, even provisional, of the performance of the two techniques.

We can appreciate that, in the modeling complex problems the holonic modeling is a better technique, and for simple problems, genetic algorithms could be superior.

Implementation
Using our proposed holonic modeling algorithm we succeeded to solve the feedrate scheduling problem. As shown in Fig.3 the input for holonic modeling are the restrictions variable, as well as several conditions (C1,C2,C3,C4,C5,C6). For instance one condition is that the cutter must be tangent to part. The output is the (i, j) pair. The issue so far that concern the holonic modeling is the way of expressing conditions and this is done by our proposed ambiguous language.

![Fig. 3 The input and the output of the holonic modeling](image)

4 Simulations and discussions
Software was developed in Visual C, to implement holonic modeling. Compared to exhaustive search the holonic modeling provides fast processing. However, the exhaustive search is more precise than the holonic modeling.

Due to the fact the computing of the algorithm must be performed online, the holonic modeling proposed in this paper is suitable so far concern processor resource requirements. Below is shown a fragment of the tool path cutter location obtained using holonic modeling.

![Fig. 4 Successive cutter location computed by holonic modeling](image)

5 Conclusions
The proposed algorithm for holonic modeling has all the chances to be a solution for many complex multi-variable problems. We also conclude that the genetic algorithm and the proposed holonic modeling are complementary algorithms.

Acknowledgement
The authors gratefully acknowledge the financial support of the Romanian Ministry of Education and Research through grant PN-II-ID_653/2007.

References: