TECHNISCHE UNIVERSITÄT
ILMENAU

50. Internationales
Wissenschaftliches Kolloquium

September, 19-23, 2005

Maschinenbau von Makro bis Nano / Mechanical Engineering from Macro to Nano

Proceedings

Fakultät für Maschinenbau /
Faculty of Mechanical Engineering

Impressum

Herausgeber:	Der Rektor der Technischen Universität Ilmenau Univ.-Prof. Dr. rer. nat. habil. Peter Scharff
Redaktion:	Referat Marketing und Studentische Angelegenheiten Andrea Schneider Fakultät für Maschinenbau Univ.-Prof. Dr.-Ing. habil. Peter Kurtz, Univ.-Prof. Dipl.-Ing. Dr. med. (habil.) Hartmut Witte, Univ.-Prof. Dr.-Ing. habil. Gerhard Linß, Dr.-Ing. Beate Schlütter, Dipl.-Biol. Danja Voges, Dipl.-Ing. Jörg Mämpel, Dipl.-Ing. Susanne Töpfer, Dipl.-Ing. Silke Stauche
Redaktionsschluss: (CD-Rom-Ausgabe)	31. August 2005
Technische Realisierung: (CD-Rom-Ausgabe)	Institut für Medientechnik an der TU Ilmenau Dipl.-Ing. Christian Weigel Dipl.-Ing. Helge Drumm Dipl.-Ing. Marco Albrecht
Technische Realisierung: (Online-Ausgabe)	Universitätsbibliothek Ilmenau ilmedia Postfach 10 05 65 98684 Ilmenau
Verlag:	isle Verlag ISLE, Betriebsstätte des ISLE e.V. Werner-von-Siemens-Str. 16 98693 Ilmenau

ISBN (Druckausgabe):	3-932633-98-9	(978-3-932633-98-0)
ISBN (CD-Rom-Ausgabe):	3-932633-99-7	(978-3-932633-99-7)

Startseite / Index:
http://www.db-thueringen.de/servlets/DocumentServlet?id=15745

50. Internationales Wissenschaftliches Kolloquium
Technische Universität Ilmenau
19.-23. September 2005

T. Schröder / M. Krabbes / R. Neugebauer

Optimization of a Trajectory Splitting Algorithm for Machine Tools with Hierarchical Drive Structures

ABSTRACT

One of the important features of machine tools is its dynamics. Fast machines mean less time for the working process and lower running costs. A machine tool, however, which should work on large work pieces, has to be big too. Because of Newton's 2nd axiom (F=m·a), heavy structures imply low accelerations when keeping the force constant.
This article refers to an approach using more drives than degrees of freedom are applied. This way, it is possible to build up hierarchical drive structures with different dynamics and specialized tasks. Small, lightweight 5DOF structures can be supported by additional large, heavy machine structures. Long ways are done by the portal and high acceleration and jerk is provided by the small structure. In 2004, a reactive algorithm has been developed to split a trajectory for a machine tool with hierarchical drive structures. Now it has been optimized and its effectiveness is nearly doubled.

1. INTRODUCTION

One of the most important features of machine tools is working speed. When the dynamics of a machine is good enough, modern processing technologies like High-Speed-Cutting can be used. But large work pieces require huge machine tools and this causes heavy moved masses. Because of Newton's 2nd axiom [7], heavy structures imply low accelerations when keeping the driving force constant. There are ways to lower these masses like changing the used materials or applying different design principles (i.e. parallel kinematics).

Furthermore, the eigenfrequency of long beam structures lowers quadratically in relation to its length L. The equation for the first eigenfrequency $\omega_0 = \sqrt{\frac{EI \cdot \pi^4}{\mu \cdot L^4}}$ [8] shows this relation. Mass increases disproportionately if we make the opposite requirement of maintaining constant eigenfrequency. Parallel kinematics also cannot solve these problems very well since the relatively thin strut structures have a detrimental effect on the dynamic behaviour in large working spaces. The relatively low eigenfrequencies cause slow dynamics to keep the received energy for vibration initiation on the machine structure at its minimum.

This article refers to a different approach. When using more drives than degrees of freedom are applied, it is possible to build up hierarchical drive structures [6]. The redundant axes have different dynamics and are used for special tasks. For example a big (and slow) portal can move throughout the complete working area (Fig. 1). It transports a small drive structure, which works along the same axes but with much higher acceleration and jerk. When using this machine, the trajectory of the tool can be split into drive-specific parts. The portal achieves long travels, whereas short and high-dynamic paths are executed by the small structure.

Figure 1: Hierarchical drive structures

Different from other approaches [1], a reactive algorithm has been developed [4; 5] to do this trajectory splitting. In the further development, this algorithm has been enhanced to solve the problem nearly optimal. For realistic machine dynamics, the time for executing a NC-program can be reduced to 50-70%. The machine uses the whole working area by moving with high dynamics.

2. MOTION OFFSET

The advantage of a reactive algorithm is its computing speed. It is able to work in the position-control-time and it can be easily implemented since no outside information is needed. All the intelligence behaviour of the hierarchical drive structures can be integrated into the numerical control. So, existing functionalities in the numerical control as well as NC-programs, written by hand or CAM systems, can be used unchanged. To the outside, the machine tool acts as a simple and fast one with a big working area. Therefore, the user does not need any additional knowledge.

In Figure 2 the concept of a standard reactive algorithm is shown. The slower machine part follows the trajectory of the tool centre position in a master-slave-link. The trajectories of each redundant processing axis are split into two parts. For the slower part $s_S(t)$, a path following the original movement is calculated under consideration of the limited dynamics as described in section 3. The remaining displacement gives the path $s_A(t)$ for the agile component. The sum of both positions is the tool centre point, which is equal to the original position $s_O(t)$. It is not necessary for the algorithm to know the complete machine structure or any kinematics. Each system of two re-

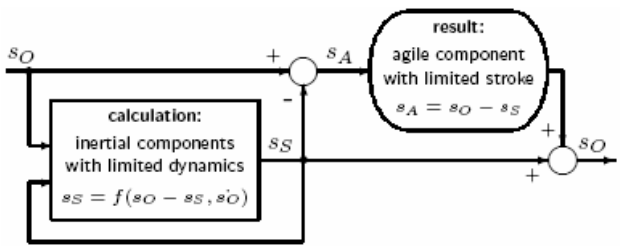

Figure 2: Structure of a reactive algorithm

dundant axes is processed separately.

When a_S is the maximum acceleration of the slower machine part, $\dot{s}_S(t_0)$ is the speed before a velocity change and $\dot{s}_S(t_n)$ the speed afterwards, the upper boundary for the maximum excursion of the smaller machine part can be calculated by equation 2. In this worst case formula, s'_S is the initial offset before the velocity change because only the slower part is moving in the case of constant velocities.

$$\Delta s_{A\max} = s'_S + \frac{\left(\dot{s}_S(t_n) - \dot{s}_S(t_0)\right)^2}{2 \cdot a_S} \cdot sign\left(\dot{s}_S(t_n) - \dot{s}_S(t_0)\right) \tag{2}$$

s'_S has to be controlled to minimize the maximum excursion, which is in fact the required work area of the small machine part. In figure 3, the function space for $s'_S = 0$ and $s'_S = \dot{s}_S(t_0) \cdot \lambda$, is shown. As seen in the figure, the second one needs significantly less space for the small machine part.

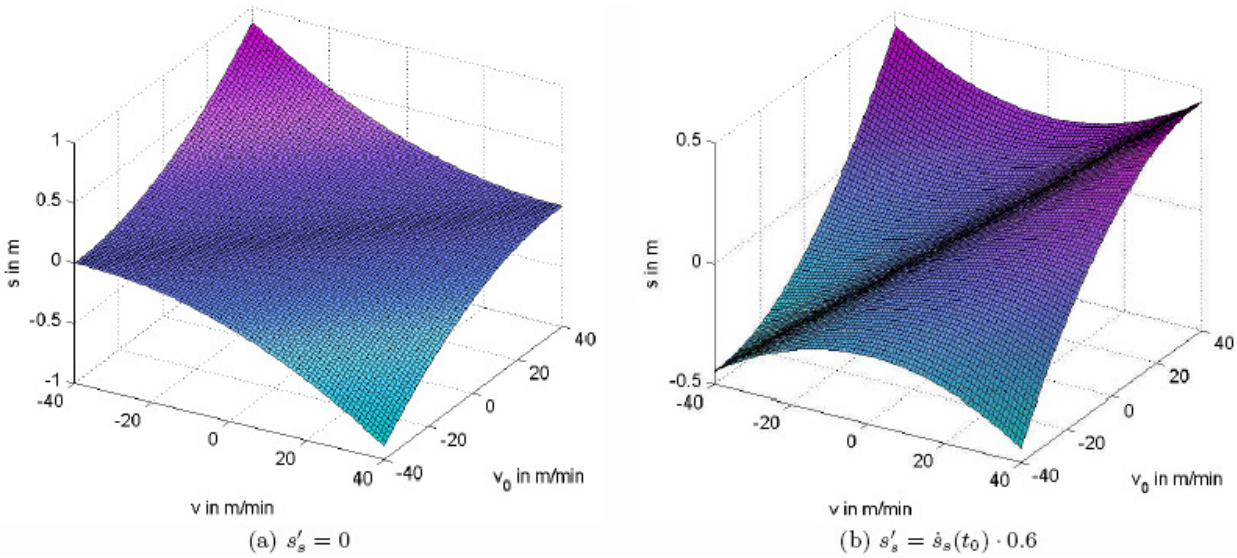

Figure 3: Work area of the small machine part when changing velocity

At this point, the knowledge of the dynamics of the master trajectory is used. When moving at a high velocity, the master cannot accelerate much more. Its maximum velocity is limited. Therefore the required workspace of the agile machine part in this direction cannot be much larger. So, it is useful to shift the base of the agile system s'_S in relation to the velocity of the slower machine part $\dot{s}_S(t_0)$. This consideration leads to the equation mentioned above.

The optimum value of the constant λ depends on the dynamics of both machine parts. By using this motion offset, the required work area for the small machine part can be reduced to half. Figure 4 shows the new structure of the algorithm.

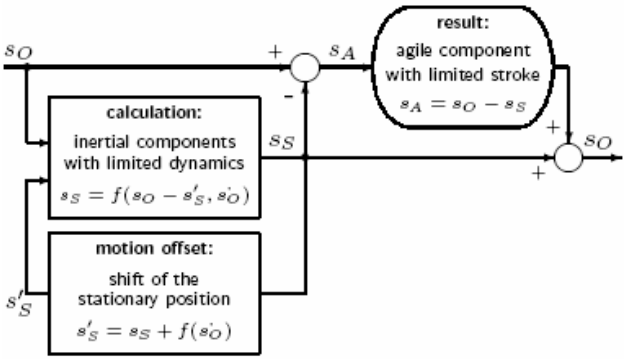

Figure 4: Scheme of a reactive trajectory splitting with motion offset

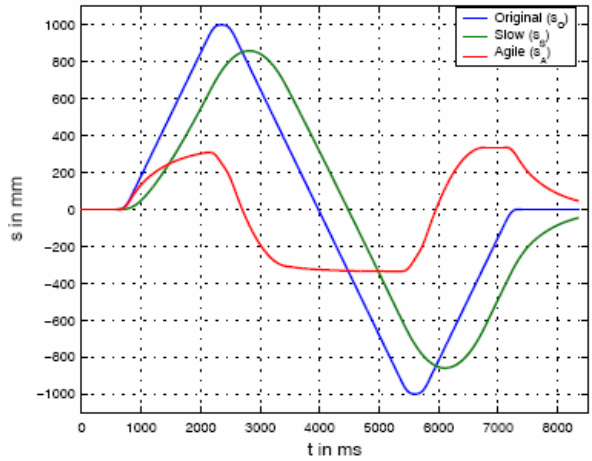

Figure 5: Trajectory splitting by the reactive algorithm

The result of the algorithm is shown in figure 5. The acceleration of the base machine part is significantly lower than the acceleration of the small one. There is a phase shift between master and base because the base reacts on the master's movement.

As shown in equation 3, it is possible to implement also the current velocity of the master trajectory to lower the required workspace for the small unit even further. The predictability of the algorithm, however, gets lower when raising η because the function (eq. 2) is no longer strictly monotonic. The local extrema of it cause an unpredictable behaviour of the algorithm.

$$s'_S = \left(\dot{s}_S(t_0) + \eta \cdot \dot{s}_O(t_0)\right) \cdot \lambda \tag{3}$$

Useful values for η depend on the dynamics of both machine parts. It does not seem to be expedient to set η greater than 10%. This way, the needed workspace for the small unit can be reduced by about 10%. On the other hand, when calculating the motion offset this way, the optimization of λ gets very difficult (see section 5).

3. Smooth Moon Landing

The smooth-moon-landing problem existed long before computer and automation [2]. But it also has been used in robotics. The problem is described by a target, which moves at constant speed $\dot{s}_O$, the distance to the target Δs, the own speed $\dot{s}_S$ and the own possible acceleration $a_S \in (-a_{\max}; a_{\max})$. It has to be computed at which point of time the direction of the acceleration has to be changed to land smoothly, which is equal to $\dot{s}_S(t_n) = \dot{s}_O$ and $\Delta s = 0$. It can be shown that by using only the maximum acceleration ($a_{\max}$ and $-a_{\max}$) the landing process is time optimal. The result of this calculation is shown in equation 4.

$$a_S = \begin{cases} +a_{\max} \\ 0 \\ -a_{\max} \end{cases} \quad \text{when} \quad \dot{s}_O + \sqrt{2 \cdot |\Delta s| \cdot a_{\max}} \cdot sign(\Delta s) \quad \begin{matrix} < \dot{s}_S(t_n) \\ = \dot{s}_S(t_n) \\ > \dot{s}_S(t_n) \end{matrix} \tag{4}$$

In the presented work, the acceleration of the slower machine part is calculated this way. The target is the master trajectory modified by the above-mentioned motion offset (see Sec. 2).

There is a small error, however, because the real machine moves with limited jerk (the acceleration function is continuous) in order to reduce vibrations. So, the algorithm based on formula 4 changes the acceleration too late. To correct this, equation 4 has to be modified to integrate the limited jerk $r_{\max}$ of the real machine. The modified value should not be the acceleration a_S any longer but the jerk $r_S = \dot{a}_S$.

The result is a system of equations with varies between cases. To simplify this calculation, the limit of the acceleration has been removed (eq. 5). This is sufficient for low ratio values of velocities and accelerations (ca. $\dot{s}_S / \ddot{s}_S \leq 1s$). Higher quotients are very rare at machine tools.

$$r_S = \begin{cases} +r_{\max} \\ 0 \\ -r_{\max} \end{cases} \quad \text{when} \quad \dot{s}_O + r_{\max} \cdot \left(\frac{|\Delta s|}{r_{\max}} \right)^{\frac{2}{3}} \cdot sign(\Delta s) \quad \begin{matrix} < \dot{s}_S(t_n) \\ = \dot{s}_S(t_n) \\ > \dot{s}_S(t_n) \end{matrix} \tag{5}$$

By using formula 5, vibrations could be reduced to a minimum and the algorithm works more precisely for real machine tools. Also, the needed work area of the small machine part could be reduced by a small amount.

3. Phase – Time shift

As shown in figure 5, there is a phase shift between master and slave trajectory. Because the slave only reacts on the master's movement, it does not behave in an optimal way. To change this, a time shift has been implemented. The structure of the algorithm has been changed as shown in figure 6.

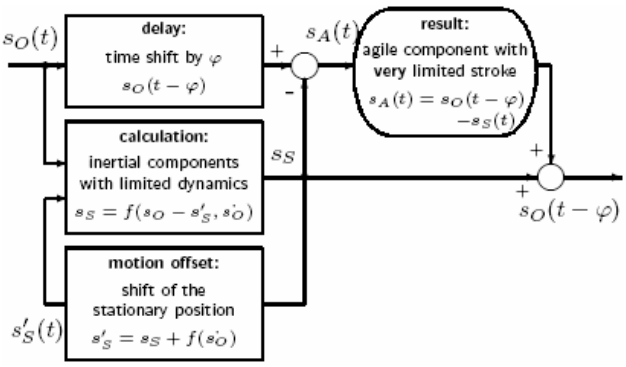

Figure 6: Scheme of a reactive trajectory splitting with time shift

In this new approach, the calculation of the slower machine part is based on the actual master trajectory $s_O(t)$ while the calculation of the small machine part is bases on a delayed master trajectory $s_O(t-\varphi)$. The constant φ defines the time shift and, therefore, also the phase shift. Figure 7 illustrates this relationship, where only the trajec-

tory of the small unit has been changed. The finding of a suitable value φ for different machine dynamics is discussed in section 5.

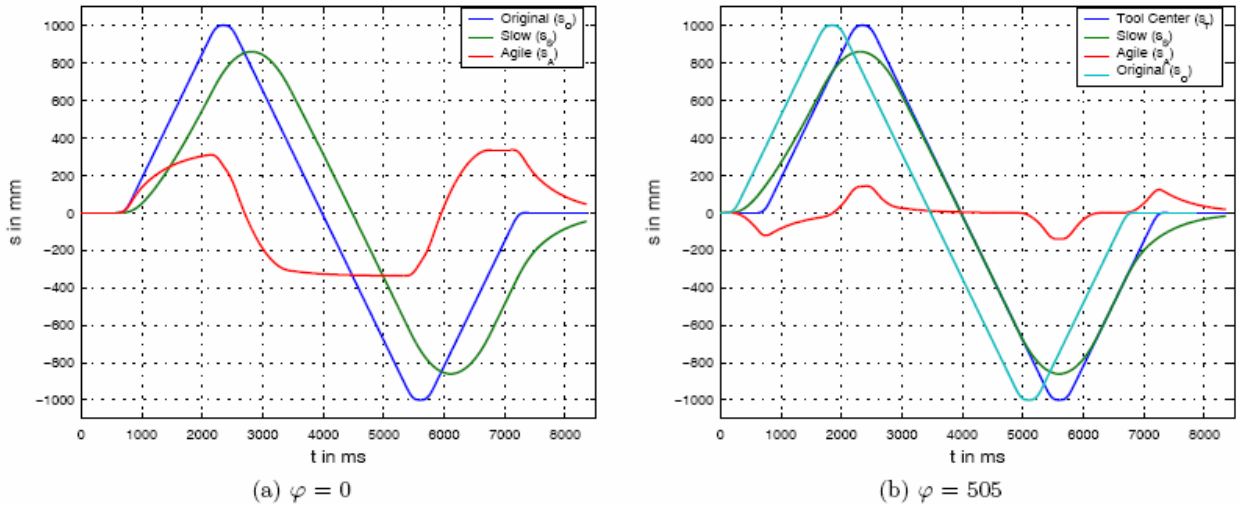

Figure 7: Trajectory splitting with different time shifts φ

By modifying the algorithm this way, the necessary work area of the small machine part can be reduced significantly. It can be shown that the algorithm works optimally for long travels. However, in case of processing short trajectories, only minor changes towards an optimal separation could be detected.

It is obvious that there must be a memory to save the older master positions. The size of this memory is determined by φ. Also the movement at the tool centre point is delayed for φ seconds. So, when the control starts to execute a program, the slower machine part begins to move while the tool remains at its position for φ seconds. After that, the tool starts to move with high speed, acceleration and jerk. At the end of the program the tool reaches the final position and stops while the slow machine part is still moving. This is important to know when interacting with the user and designing security functions.

By using this little extra time, the usage of the work area of the small machine part is much more efficient as demonstrated in Figure 8. The path is a set of two nooks in x-y-plane. Because all master movements are in the first quadrant (right or up), the algorithm without time shift only uses right or up moves for the small machine part to realize the desired dynamics. This way only 1/4 of its work area is used. On the right side, the algorithm with time shift uses the whole work area. As shown in the figures, the paths of the small machine part for one step have the same length in both experiments.

The position of them, however, has changed. While the algorithm without time shift uses a work area of about [0; 200] millimetres, the algorithm with time shift uses a work area of about [−100; 100] millimetres.

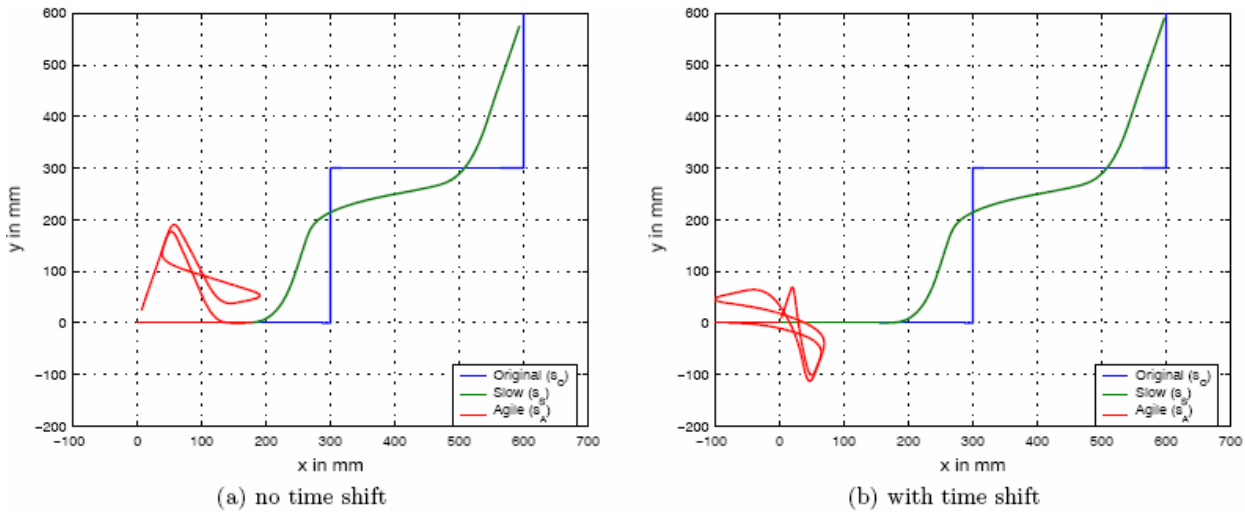

Figure 8: Trajectory splitting of two nooks in x-y-plane with and without time shift

5. OPTIMIZATION OF THE PARAMETERS

The parameters of the algorithm λ, φ and η control its behaviour. The optimal value of these parameters depends on the dynamics of both machine parts. The mathematical representation of the algorithm is a complex set of differential equations so it cannot be optimized analytically.

The parameter η has to be set to 10% permanently to avoid problems caused by local extrema and minimizing the required workspace of the agile unit.

It can be assumed, that the optimal phase shift minimizes the offset between original trajectory and the slow machine part (see fig. 7). That means, the agile unit should have no deviation from its centre ($s_A(t_x) = 0$). By using this approach, a good approximation of φ can be done by $\varphi = \lambda \cdot (1+\eta)$ (see eq. 3). For longer phases of constant velocity of the master trajectory $\dot{s}_S = \dot{s}_O$ can be set.

The optimization of λ is much more complicated. If it gets too small, the behaviour of the algorithm is no longer predictable and will require an additional look-ahead level. When the value of λ is too high, the algorithm works inefficiently and needs too much space for the small unit.

To optimize λ, the worst cases of velocity and acceleration that causes the largest required working space for the agile machine part for the given dynamics and the tested λ have to be found. The worst case of the jerk can be left out because the jerk causes only a minor effect on the path. To generate a worst case for velocity in a one-dimensional workspace, which means one axis, a trajectory of three points is required. The worst cases for acceleration and jerk can be generated by trajectories of four respectively five points.

Therefore it is sufficient to generate trajectories of 4 points to simulate all velocity and acceleration worst cases. Figure 9 shows a part of the function space of the required work area of the small ma-

chine part. The original trajectory runs through the points P1, P2, P3 and P4. For figure 9, P1 and P4 are kept constant.

Obviously, simple gradient-climbing algorithms cannot be used because of the very planar function space. In the latest development, an approach based on genetic algorithms [3] has been tested successfully. Each genetic agent represents a single trajectory of four points. These points are encoded in the genetic codes of the agent. The required workspace for the small unit is used as fitness function. By controlling the composition of the population, the amount of chance can be set very precisely. This way, the worst cases can be detected relatively quickly and using these results, λ can be modified towards its optimum.

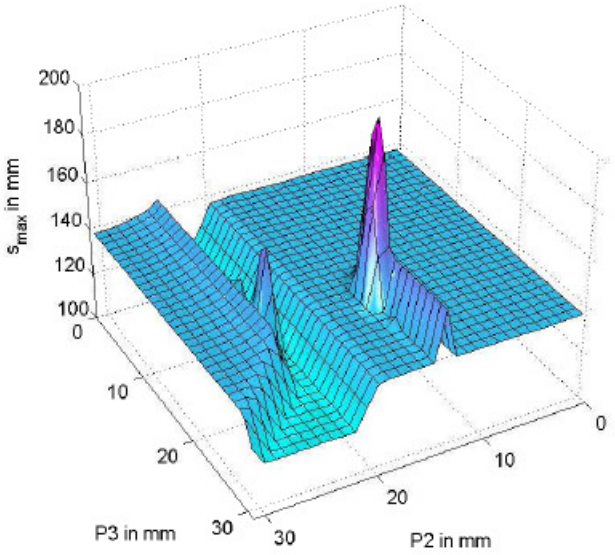

Figure 9: the required work area of the agile machine part

By using genetic algorithms, the time to find the global extremum of a given machine is about 10 times shorter than testing the whole function space. As the downside of this approach it cannot be ensured that the last found extremum is the global extremum. The calculation has to be carried out for a defined time (genetic generations) to get a sufficient approximation of the worst case.

Once the worst cases of velocity and acceleration that causes the largest required working space for the agile machine part for the given dynamics and the tested λ are known, the optimization of λ is simple. Because of the structure of the algorithm, the function space of λ has only one extremum as shown in figure 10.

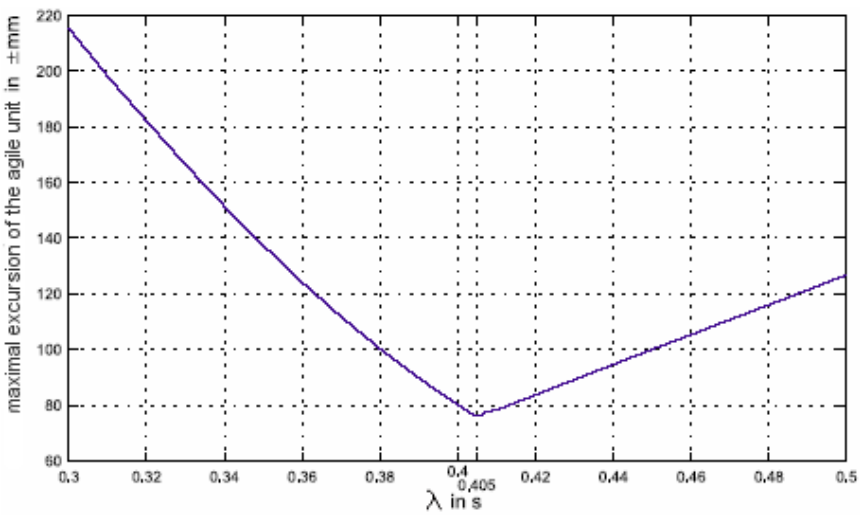

Figure 10: Maximal excursion of the agile unit in relation to λ

6. CONCLUSION AND FUTURE RESEARCH

The processing speed of large work pieces on machine tools is limited by the heavy structures of the machine tool itself. Because the forces of the drives cannot be increased arbitrarily, other approaches have to be used. One of them is to build a system of hierarchical drive structures for specialized tasks.

This article presents a reactive algorithm, which splits one master trajectory (position of the tool centre) into two separate trajectories. Each of them has defined limits for velocity, acceleration and jerk. The smaller one also has a very tight limit for its working area. Because the algorithm works reactively, it can be implemented directly in an open numerical control. This way, all features of the control can be used further. Additionally, the user does not need any special knowledge about the machine. Since each axis is split separately there is no limitation of the complexity of the machine tool.

The current version of the shown algorithm works optimally for long travels and acceptably for shorter ones. The dynamics of the whole machine is the difference between the dynamics of the agile and the slow machine part. The algorithm can also work in the whole work area of the slow part.

In the future work, the algorithm will be implemented as a standard function in a real numerical control. The development should end in a prototype machine tool with the described characteristics, which is optimized to best benefit from hierarchical drive separation (fig. 11).

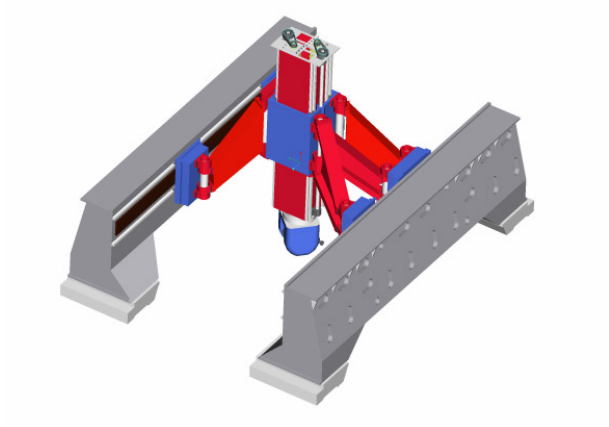

Figure 11: Scissor kinematics with hierarchical drive structures

References:

[1] Heimpel W., Schwaar M., Wagner R. (2004) Industrial experience in application of hybrid kinematics machines with overlayed axes. In: Neugebauer R (ed) Parallel Kinematic Machines in Research and Practice, Verlag Wissenschaftliche Scripten, pp 313–328

[2] Hohmann W (1925) Die Erreichbarkeit der Himmelskörper. R. Oldenbourg Verlag

[3] Holland John H. (1986). Escaping Brittleness: The Possibilities of General-Purpose Learning Algorithms Applied to Parallel Rule-Based Systems. In: Michalski R. S., Carbonell J. G.and Mitchell T. M., Ed.: Machine Learning II. Morgan Kaufmann.

[4] Krabbes M., Neugebauer R., Ihlenfeld S., Schroeder T. (2004) Reactive splitting into dynamic components for overlayed drive structures. In: Neugebauer R (ed) Parallel Kinematic Machines in Research and Practice, Verlag Wissenschaftliche Scripten, pp 247–256

[5] Krabbes M., Neugebauer R., Ihlenfeld S., Schroeder T. (2004) Reaktive Dynamikzerlegung für überlagerte Antriebsstrukturen. In: Elektrisch-mechanische Antriebssysteme, VDE Verlag GmbH, pp 835–846

[6] Kuhfuss B., Schenk C. (2002) Optimized hybrid machine structures - a new development approach. In: Neugebauer R (ed) 3. Chemnitzer Parallelkinematik Seminar 2002, pp 213–225

[7] Sir Isaac Newton (1687) Philosophiae Naturalis Principia Mathematica

[8] Smith Stuart T. (2000) Flexures: elements of elastic mechanisms. Gordon and Breach Science Publishers

Autorenangabe(n):
Dipl.-Inf. Thomas Schröder
TU-Chemnitz, Professur Werkzeugmaschinen, Reichenhainer Str. 70
09126 Chemnitz
Tel.: 0371/5397-405
Fax: 0371/5397-488
E-mail: thomas.schroeder.tu-chemnitz@iwu.fhg.de

Prof. Markus Krabbes
Leipzig University of Applied Sciences

Prof. Dr.-Ing. habil. Prof. E.h. Dr.-Ing. E.h. Reimund Neugebauer
Fraunhofer Institute for Machine Tools and Forming Technology IWU