Available online at www.sciencedirect.com

ScienceDirect

Procedia CIRP 00 (2019) 000–000

www.elsevier.com/locate/procedia

9th CIRP Conference on High Performance Cutting

OpenCN: an open-source CNC with new Trajectory Optimization for high Performance Milling

Raoul Herzog[a,*], Alain Schorderet[a], Daniel Rossier[a]

[a]*School of Management and Engineering Vaud, HES-SO // University of Applied Sciences and Arts Western Switzerland, HEIG-VD, CH-1401 Yverdon-les-Bains*

Abstract

Todays commercial CNC's are black boxes with almost no access to trajectory planning. LinuxCNC is open-source, but not suitable for high performance machining because of the lack of jerk limitation. OpenCN overcomes the limitations of LinuxCNC with the following features: efficient geometric G^2 rounding between blocks, jerk control using LP optimization with receding horizon, automatic C code generation for trajectory planning algorithms, and a 10 kHz EtherCAT transmission to the drives using distributed clock. A new framework with real-time Xenomai asymmetric multiprocessing was developed using a recent Linux kernel. A functional validation of OpenCN is carried out on a high performance 3 axis mini-milling machine for work pieces in brass up to F9000 feedrate.

Keywords: Computer Numerical Control; CNC; Optimization; Trajectory Planning; Open Source

1. Introduction

Commercial CNC's are closed with no possibility to change trajectory planning algorithms. There are cases where the performance of high end CNC's are poor compared to optimized trajectory planning as shown by Schorderet et al. [5]. This motivates the development of an open source CNC.

In the open source community related to machine controller, LinuxCNC remains a well-known software solution that fits a wide range of applications. LinuxCNC stems from a long history of various developments around the EMC project in early nineties published by Staroveški et al. [1]. LinuxCNC is not suitable for high performance applications for several reasons, the most important being the lack of jerk limitation in trajectory planning.

This paper describes a new open source software "OpenCN", originally based on LinuxCNC, but featuring a completely reworked trajectory planning and a fast EtherCAT transmission to the drives, enabling it to work properly in high end applications. The ultimate goal of OpenCN consists in providing a powerful and flexible open source CNC framework for future research (different kinematics, new algorithms for trajectory planning incorporating the vibrational behaviour of the machine, real-time process monitoring, Industry 4.0, etc.) OpenCN runs on standard x86 and ARM without need of special hardware.

The paper is organized as follows: Section 2 explains the trajectory planning; Section 3 shows the real time Xenomai AMP framework, and Section 4 describes the hardware of a high performance 3 axis mini-milling machine. Section 5 highlights the experimental validation, followed by conclusions in Section 6.

2. Trajectory planning

Trajectory planning concerns the generation of time optimal setpoint values subject to constraints regarding speed, acceleration, jerk, and geometrical tolerances w.r.t. the programmed path. Literature about trajectory planning is abundant, and the following discussion is non exhaustive. Existing solutions can be divided in 2 groups: *simultaneous* optimization of geometry and feedrate, and

* Corresponding author. Tel.: ++41 24 557 61 93
E-mail address: raoul.herzog@heig-vd.ch (Raoul Herzog).

subsequent, i.e. separate optimization of geometry followed by feedrate planning. Simultaneous optimization was proposed by Mercy et al. [7], where the trajectory is parametrized by B-splines with time as independent variable. Spline relaxation avoids gridding of the constraints. The resulting nonlinear program is non convex, leading to a potential lack in robustness of numerical solving. Therefore, *subsequent* optimization of geometry and feedrate was preferred, with a special emphasis to the utilization of analytical solutions, convex formulations of optimization problems, and proven numerical algorithms for standard mathematical problems.

2.1. Geometric G^2 rounding, splitting and compressing

The interpretation of G code (RS274) yields a list of parametric curve pieces described by $\mathbf{r}_k(u_k)$, $k = 1, \ldots, N$, where each curve abscissa u_k is normalized $0 \le u_k \le 1$. In order to avoid acceleration jumps, the transitions must be rounded to be G^2 continuous. An optimal G^2 Hermite interpolation was proposed by Herzog and Blanc [6], based on the minimization of $\int_0^1 ||\boldsymbol{r}'''(u)||^2\, du$, subject to G^2 continuity at lift-off points. The analytic determination of the transition curve turns out to be a parametric curve given by quintic polynomials. Determination of the coefficients leads to finding the positive roots of a polynomial of degree at most 9. This approach leads to the global solution with an extremely fast (12 μs/transition on 1 Intel i7 core) and reliable algorithm with established geometrical tolerance analysis.

Geometrical operations also include detection of cusps in the programmed path, or curve pieces being already G^2 continuous. In both cases there is no need for inserting a transition curve. Furthermore, a homogenization of the length of curve pieces is necessary. Too long curve pieces may be split in smaller ones, and a sequence of tiny straight line segments must be compressed into an analytic representation, e.g. as a B-spline.

2.2. Feedrate planning with receding horizon

The time derivatives $\frac{d^n}{dt^n}\,\mathbf{r}(u(t))$ can be calculated by applying chain and multiplication rule

$$\dot{\mathbf{r}} = \mathbf{r}'\,\dot{u} \tag{1}$$

$$\ddot{\mathbf{r}} = \mathbf{r}''\,\dot{u}^2 \;+\; \mathbf{r}'\,\ddot{u} \tag{2}$$

$$\dddot{\mathbf{r}} = \mathbf{r}'''\,\dot{u}^3 \;+\; 3\,\mathbf{r}''\,\dot{u}\,\ddot{u} \;+\; \mathbf{r}'\,\dddot{u}. \tag{3}$$

It already was observed by Verscheure et al. [4] that by applying a nonlinear change in variable $q(u) := \dot{u}^2$, $\ddot{u} = \frac{1}{2}q'$, acceleration $\ddot{\mathbf{r}}$ becomes a linear function of the new unknown $q(u)$, and the time optimal control problem *without* jerk constraint becomes convex

$$q(u) \;<\; v_{\max}^2 \,/\, ||\mathbf{r}'(u)||^2 \tag{4}$$

$$-\mathbf{a}_{\max} \;<\; q(u)\,\mathbf{r}''(u) + \frac{1}{2}q'(u)\,\mathbf{r}'(u) \;<\; \mathbf{a}_{\max}, \tag{5}$$

where inequality (5) is understood componentwise. Erkokmaz et al. [2] showed that the resulting problem *without* jerk constraint can be cast as a linear program (LP), observing that minimizing the travel time T is equivalent to maximizing $\int_0^1 q(u)\,du$ which is linear in the decision variables. For this purpose, the unknown function $q(u)$ is approximated by a B-spline of degree d and a given knot vector. The new unknowns are the coefficients $\mathbf{x}$ of the B-spline. The jerk constraint is taken into account by

$$\left| q(u)\,\mathbf{r}'''(u) + \frac{3}{2}q'(u)\,\mathbf{r}''(u) + \frac{1}{2}q''(u)\,\mathbf{r}'(u) \right| \sqrt{q(u)} \;<\; \mathbf{j}_{\max}.$$

Erkokmaz et al. [2] proposed to replace the term $\sqrt{q(u)}$ in Eq. (2.2) by a precomputed upper bound $\sqrt{q^\star(u)}$ being the solution of the feedrate planning without jerk constraint. This gives a conservative approximation of the jerk constraint compatible with the LP setting taking the standard form

$$\max_{\mathbf{x}} \; \mathbf{c}^T\,\mathbf{x} \quad \text{subject to} \quad \begin{cases} \mathbf{A}\,\mathbf{x} \;<\; \mathbf{b} \\ \mathbf{A}_{\text{eq}}\,\mathbf{x} \;=\; \mathbf{b}_{\text{eq}}. \end{cases} \tag{6}$$

The Simplex solver CLP (COIN-OR) is used for LP solving. The feedrate planning is broken up to small chunks using a standard receding horizon technique. For the choice of the receding horizon depth N_h, several tests indicated that values above 5 give no significant gain in time optimality.

2.3. Trajectory sampling and zero speed handling

Denoting the sampling period by Δt, $u_k := u(t_k)$ and $u_{k+1} := u(t_k + \Delta t)$, the truncated Taylor expansion of $u(t)$ yields

$$u_{k+1} = u_k + \sqrt{q(u_k)}\,\Delta t \;+\; \frac{1}{4}q'(u_k)\,\Delta t^2. \tag{7}$$

If $u_{k+1} > 1$, a transition to the next curve piece must take place. To this end, we first calculate the elapsed time T_r from the previous point u_k to the end point $u = 1$. If $q(u)$ is approximated by a piecewise linear function, analytic integration [4] gives

$$T_r = \frac{2\,(1 - u_k)}{\sqrt{q(1)} + \sqrt{q(u_k)}}. \tag{8}$$

The sampling of the next curve piece must be started with a shortened value $\Delta t_0 = \Delta t \;-\; T_r$ using Eq. (7).

Zero speed is a singularity to be treated separately by imposing a constant pseudo jerk j_{ps} during short pieces of curve around standstill.

$$u_k = \frac{1}{6}j_{ps}\,(k\,\Delta t)^3 \tag{9}$$

Continuity of speed and tangential acceleration are imposed by equality constraints to the adjacent standard feedrate planning.

3. Xenomai–AMP framework

LinuxCNC supports several hard realtime extensions such as RTAI, Xenomai and RT-Preempt which can guarantee a high degree of determinism as required by the EtherCAT fieldbus for example. However, over the years, the software architecture of LinuxCNC became very complex; upgrading to a recent kernel, adding new path planning algorithms, keeping a distribution as lightweight and customizable as possible, and having a bootup time less than 30 seconds for the complete environment is difficult to achieve. For these reasons, the development of OpenCN with asymmetric multiprocessing (AMP) started from a vanilla Linux (LTS) kernel 4.19 in which a minimal set of pieces of LinuxCNC like HAL, RTAPI and components-driven interfaces have been integrated from scratch with distinctive mechanisms in order to overcome the problems mentioned before.

Currently, OpenCN has been developed on a quadcore x86 platform. Hyperthreading is disabled to simplify CPU interactions and to avoid cache latency issues. The hard real-time (RT) extension is based on the Cobalt core of Xenomai providing RT services at the lowest (kernel) level. By design choice, OpenCN runs all real-time (RTDM) tasks within the kernel space on a dedicated CPU core (CPU #1) also known as RT domain. Hence, the Linux kernel has been patched to configure the task affinity to specific cores according to their role, and to prevent the Linux scheduler to interfere in any way with the RT domain. With such an approach, the low-level I-pipe layer used by Xenomai to manage syscall interrupts and interrupt requests (IRQ) has been removed in order to reduce some useless processing overhead. The overall architecture is depicted on Fig 1.

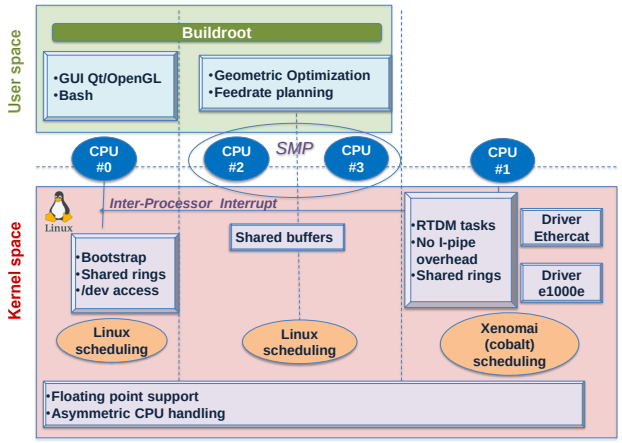

Fig. 1. OpenCN (AMP) general architecture.

While CPU #0 is normally used to bootstrap the kernel, to set up initial tasks and to run normal GUI applications, CPU #2 and #3 are used in SMP mode to run multi-threaded path planning algorithms.

The OpenCN-AMP framework uses the widely used open source EtherCAT master stack from IgH. The driver can be enhanced with Xenomai (RTDM) tasks to provide real-time capabilities. Therefore, it was straightforward to integrate it in the OpenCN real-time domain. Further adaptations have been carried out to get a stable distributed clock and to enable a reliable and jitter-free 10 kHz transmission.

Furthermore, the e1000e network card driver has been patched to run in the RT domain using a polling mechanism instead of IRQs. Warfield et al. [8] proposed shared rings for efficient data sharing between RT and non-RT domains. Inter-Processor Interrupts (IPIs) between non-RT and RT domains are used to ensure tasks and buffering synchronization.

Even if the overall architecture leads to very stable and robust long time behaviour, some additional jitters in the EtherCAT driver running on CPU #1 may occur due to direct memory access (DMA) requests triggered by the on-board GPU of the x86 motherboard. Indeed, the GPU apparently has a very high priority on the data bus during GUI operations and can block the RT CPU #1 during significant delays, leading to some loss of EtherCAT frames. In release 2 the GUI is moved to another PC using a remote connection and a lightweight protocol.

4. Hardware description of the m3 mini-milling machine

Fig. 2 shows the three-axis mini-milling machine used for the experimental validation of OpenCN. The aspect ratio between machine and workpiece is small (5:1), and overall power consumption is less than 780 W. The machine design is based on a stacked serial Cartesian axis configuration, driven by ball screws, creating a working space of 50×50×30 mm. The sum of moving masses is less than 10 kg, static stiffness of the spindle holder is $5 \cdot 10^6$ N/m, and the first structural mode is at about 180 Hz. Maximum velocity and acceleration are 30 m/s and 20 m/s^2 respectively. The machine is equipped with a high speed Meyrat spindle of 240 W, rotating at max 80'000 rpm.

Two high-end EtherCAT drives TSD80E from the company Triamec are used for driving axes and spindle. They use a dual loop feedback from rotation encoders and linear encoders. Internal sampling frequency is 100 kHz for all control loops and feedforward compensations. Reference setpoint values at an EtherCAT frame rate of 10 kHz are internally fine interpolated and upsampled to 100 kHz.

OpenCN runs on a x86 PC, i7-860 @2.8GHz, equipped with a standard Ethernet e1000e controller used for EtherCAT communication.

5. Experimental validation of OpenCN

As a first step, position setpoint values calculated from the OpenCN path planning are checked to ensure that ve-

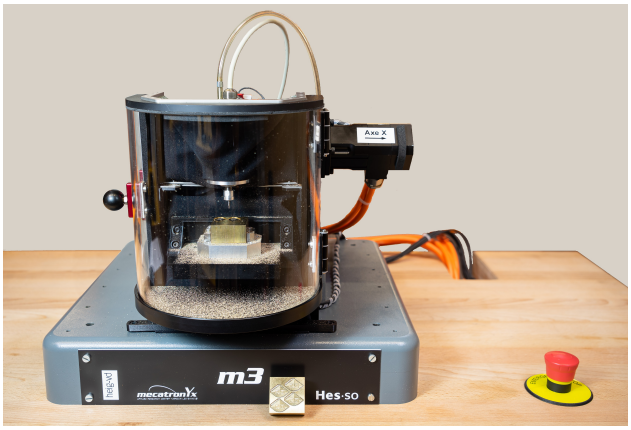

Fig. 2. Mini-milling machine m3.

locity, acceleration and jerk constraints hold at any time. Pontryagin's maximum principle for time optimality compels the solution to have at any time at least one *active*, i.e. saturated constraint (bang-bang behaviour). The trajectory starts from standstill with the maximum programmed jerk in one axis until reaching the maximum programmed acceleration in one axis and later the maximum programmed feedrate. Realtime behaviour of OpenCN was validated using a drive variable which increments in case of a lost EtherCAT frame. The experimental validation on the mini-milling machine has been carried out using the parameters in Table 1.

Table 1. OpenCN parameters used for the experimental validation.

Parameter	Symbol	Value	Unit
Cutoff length for smoothing	L_{cut}	0.1	mm
Threshold for splitting	L_{split}	2	mm
Max speed	$v_{\max}$	see fig. 3	mm/min
Max acceleration per axis	$a_{\max}$	$20{\cdot}10^3$	mm/s^2
Max jerk per axis	$j_{\max}$	$1.5{\cdot}10^6$	mm/s^3
EtherCAT period	Δt	0.1	ms
Receding horizon depth	N_h	3	-
B-spline degree for $q(u)$	d	3	-
# of knots for B-spline $q(u)$	N_k	10	-
# of grid points for inequalities	N_g	20	-

The experimental validation aimed at a functional validation of OpenCN using a G-code generated by Autodesk HSMWorks. A milling tool with a diameter of 1 mm running at 45'000 rpm was used for the brass workpiece shown in fig. 3 and in video [9].

6. Conclusions

OpenCN is a large scale open source CNC software, featuring jerk control and fast EtherCAT communication based on a Xenomai AMP framework. The generation of setpoint values is calculated on-line based on proven numerical algorithms as polynomial root solving and linear programming (LP) solvers. The path planning algorithms are coded in Matlab followed by automatic C code generation for better maintainability and coding efficiency. The first release of OpenCN is available under [9]. Five axis kinematic support, further improvements in reliability, and stronger parallelization are underway. Support from other research groups are highly appreciated. This work was funded by HEIG-VD and realized within the mecatronYx platform.

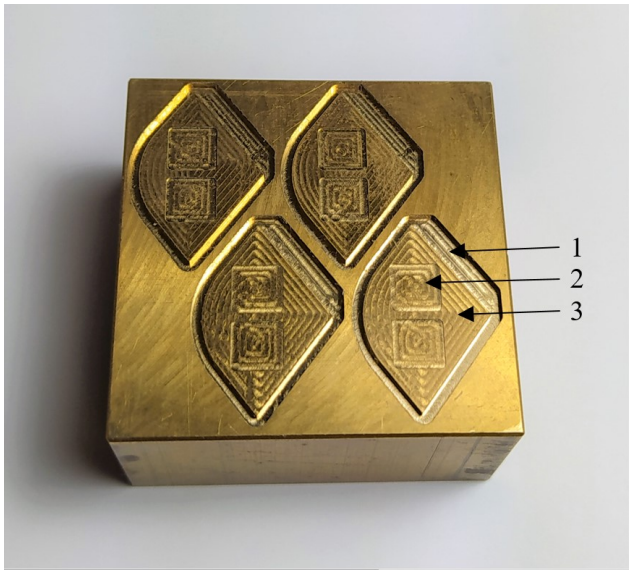

Fig. 3. Validation work piece (40 × 40 mm). Contouring (1): roughing 0.2mm/pass @ F5000. Finishing 0.1mm/pass @ F1000. Pockets (2): F272. Facing (3): F2000.

References

[1] Staroveški, T., Brezak, D., Udiljak, T., Majetić, D. "Implementation of a Linux-based CNC open control system", 12th int. scientific conference on production engineering, 2009.
[2] Erkorkmaz, K., Chen, Q.-G., Zhao, M.-Y., Beudaert, X., 2017. "Linear Programming and Windowing based Feedrate Optimization for Spline Toolpaths", CIRP Annals - Manufacturing Technology 66, Elsevier, pp. 393-396.
[3] Beudaert, X., Pechard, P.-Y., Tournier, Ch., "5-Axis Tool Path Smoothing based on Drive Constraints", International Journal of Machine Tools & Manufacture 51, 2011, pp. 958-965.
[4] Verscheure, D., Demeulenaere, B., Swevers, J., De Schutter, J., Diehl, M., "Time-Optimal Path Tracking for Robots: A Convex Optimization Approach", IEEE Trans. on Automatic Control, Vol. 54, No. 10, Oct. 2009, pp. 2318-2327.
[5] Schorderet, A., Herzog, R., Jacquod, N., Marchand, Y., Prongue, Ch., 2019. "Productivity Increase of High Precision Micro-Milling by Trajectory Optimization", Modern Machinery (MM) Science Journal, pp. 3179-3186.
[6] Herzog, R., Blanc, Ph., 2019. "Optimal G^2 Hermite Interpolation for 3D Curves", Computer-Aided Design, Elsevier, Vol. 117, Dec. 2019.
[7] Mercy, T., Jacquod, N., Herzog, R., Pipeleers, G., "Spline-Based Trajectory Generation for CNC Machines", IEEE Transactions on Industrial Electronics, Vol. 66, Issue 8, Aug. 2019.
[8] Warfield, A., Fraser, K., Hand, S., Deegan, T., "Facilitating the development of soft devices", USENIX annual technical conference, 2005.
[9] OpenCN: https://gitlab.com/mecatronyx/opencnc/opencn, Documentation: https://mecatronyx.gitlab.io/opencnc/doc/, https://www.youtube.com/channel/UC8FQCu_fKYfK7QRDN0j_dBw