Model-Based Energy Recuperation of Multi-Axis Machines

Tamás Juhász, Matthias Kennel, Marco Franke, Ulrich Schmucker
Fraunhofer Institute for Factory Operation and Automation IFF
Sandtorstr. 22, 39106 Magdeburg, Germany
{tamas.juhasz, matthias.kennel, marco.franke, ulrich.schmucker}@iff.fraunhofer.de

Abstract

The peak power consumption of multi-axis production machinery (e.g. industrial robots) is determined by the forces during acceleration phases of continuous movements. The required high electric currents represent a cost factor in terms of the mains power supply. In this paper a new Modelica-based method is presented to save the mechanical braking energy of production machinery into a local flywheel-based energy recuperation system (ERS) for later utilization. An ERS has twofold advantages: on the one side it reduces the apparent power peaks from the mains power system. On the other side the overall energy consumption can also be reduced therewith. However, a mechanical ERS with a flywheel needs to be controlled in advance, as its internal inertia and the switched magnetic field introduce some dead time in the process. Therefore, the here presented approach uses an interdisciplinary Modelica model of the machinery to compute future power requirements prior execution of new movements. It is assumed that the machine program is known upfront in a textual form. The movement commands must be carried out with the virtual machine model first. The simulation computes the energy demand, according to which the stored amount of energy within the ERS (i.e. the angular velocity of its flywheel) must be controlled. The here computed reference angular velocity signal is put later also to the real controller, where the physical ERS is attached to as an extra motor. This paper presents the methodology along an example of a 3-axis robotic manipulator that is installed in the VDTC building of the Fraunhofer IFF, Magdeburg. This specific example has been used to validate the concept: 12% less power-consumption and 10% less power peaks were achieved during the operation.

Keywords: production machine, power efficiency, energy recuperation, flywheel, Modelica simulation, virtual NC

1 Introduction

Energy efficiency is nowadays a crucial competitive factor within the production industry. Additionally, in order to reduce production time, cycle times must always become shorter. The power requirement of production machinery is mostly dominated by the acceleration and braking of mechanical loads (e.g. heavy chassis parts in a car production line). As a common power supply for their induction motors, multi-axis machines are utilizing a single intermediate DC circuit. The motors of the machine are usually able to generate electric current upon external driving forces. This generated current can be recovered by the frequency inverters into the capacitors of the intermediate circuit.

The aim of this research was to develop a new methodology to integrate an electromechanical energy storage unit here, thus larger amounts of recovered energy can be utilized later in the production process again.

1.1 Energy Recuperation System - ERS

Modern high-speed flywheel-based energy recuperation systems (ERS) have high durability and can store a large amount of energy in a relatively compact space. A mechanical ERS itself is equipped with an AC induction machine that can also accelerate and decelerate the flywheel, thus convert between electrical and mechanical energy.

There are multiple industrial application fields where electromechanical ERS are being used with utmost efficiency. In case of production machinery they can be used as an uninterruptible power supply to overcome blackouts enabling the machinery to enter a safe state without causing any damage of the product. A most recent application is to compensate power fluctuations in electric networks of city trams [5]. This very application suggested the idea of this study: the integration of an ERS into a production machine.

1.2 Storing Energy in a Rotating Flywheel

The following formula can be used to compute the stored mechanical energy of a rotating flywheel:

$$E = \frac{1}{2}\theta\omega^2 \tag{1}$$

where θ is the rotational inertia and ω is the angular velocity of the flywheel rotor. The bearings of the flywheel are utmost optimized, thus the idling short-term friction losses are negligible. At ultra-high speeds (>25000 rpm, carbon-fiber flywheel) there is a need of a vacuum chamber to reduce drag losses. Nevertheless, the power consumption of this extra vacuum pump lessens the efficiency of the ERS.

There is a hurdle at using mechanical energy storage units in multi-axis machines due to large differences in mechanical and electrical time constants. The speed of an inert flywheel must be controlled in advance to be able to supply energy to and from the process effectively.

As production machines are usually program-driven, the next movement command can be known a priori. Therefore, the proper flywheel speed can be pre-computed by utilizing a mechatronic model of the whole system. The flywheel speed must be decreased a short time earlier than the axes begin to accelerate, and vice versa: it must be accelerated upon recuperating mechanical energy via the machine axes.

1.3 The ERMA Project

The here presented work has been done in the *ERMA* project (Energy Recuperation for Multi-Axis Machines) funded by the German Federal Ministry of Education and Research under the project number 02PK2192. The consortium agreed to demonstrate the methodology with a simple, 3-axis robotic manipulator that can be used to transport alloy wheels.

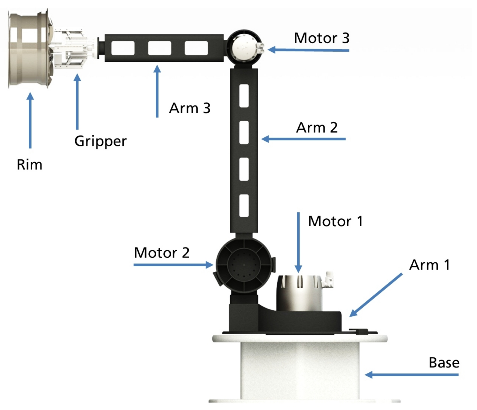


Figure 1: CAD model of the ERMA demonstrator

As production machinery in general is targeted by the here presented concept, a Siemens *Sinumerik 840D sl* NC controller was chosen to control the demonstrator. Numeric control is very common in the production industry and basically does not differ much from other interpreted robot programming languages. However, conventional NC controllers give no access to their internal axis interpolator states prior execution. Therefore, a look-ahead in the interpreted program is not directly possible.

Fortunately, Siemens offers a virtual NC kernel (VNCK), which is a software service that covers the whole functionality of a real Sinumerik NC interpreter offline, under Windows. An NC program (G-code) can be interpreted with the VNCK the same way as if it would run on the real controller, thus the reference trajectories can be extracted for each axis in advance. Therefore, it is known how a given movement command will be executed. The commissioning of VNCK is quite straightforward: once the configuration of the real Sinumerik NC has been set up with the real machine axes and parameters, the exported settings can be used later on demand to simply boot the VNCK. The identical configurations ensure a consistency between the simulated trajectories and the real executed ones.

The *ERMA* demonstrator system has been equipped with a prototype, steel flywheel-based ERS from the partner company *rosseta Technik GmbH* [5]. The ERS is shown on Figure 2 and has the following main properties:

- Mass: 52 kg
- Dimensions: 410 x 190 mm
- Induction machine: 20A peak current
- Rated power: 10 kW
- Rated energy: 22.8 kWs
- Max. speed 25000 rpm

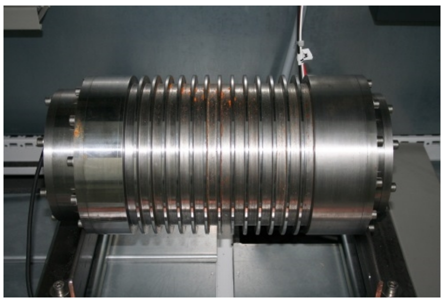


Figure 2: The flywheel-based ERS in the cabinet

The electrical cabinet including the four motor controllers (3 synchronous permanent-magnet drives of the manipulator + 1 asynchronous induction machine of the flywheel) have been developed by the partner company *aradex AG* [6].

2 Modeling in Modelica

In this chapter the modeling workflow is described along the example of the *ERMA* demonstrator system. It can be used to model a production machine conveniently in Modelica environment.

2.1 Mechanical Subsystem

In the recent years in the Fraunhofer IFF there has been a research focusing on automated translation of CAD information (geometry, physical properties and kinematic structure) into a parameterized Modelica multi-body description [1]. A plug-in was developed to generate MKS models directly from the Pro/Engineer CAD system. However, a lot of 3rd party Pro/Engineer assemblies could never be translated to a valid model without manual work. In a CAD environment the effort of defining the right constraints of a complex articulated mechanical system can be a troublesome work. In order to ease this task, designers usually circumvent the right definition of degrees-of-freedom between adjoined CAD assemblies and parts. Unfixed bodies or overdetermined constraints render the automated translation process erroneous. During the model conversion the constraints are normally mapped to Modelica joints automatically. Although the original constraints can be named, no meaningful joint names can be specified in Pro/Engineer. Therefore, the direct usability of a fully auto-generated model is questionable in model-in-the-loop scenarios.

Virtual commissioning of digital machine models with real controllers in the loop represents a new research focus in the Fraunhofer IFF. There must be a clear naming convention of joints to define the communication channels to and from simulated axes. Therefore, the solution of automated model translation had to be developed further. Using the *VINCENT* framework [2][7] the multi-body structure can be configured separately. The rigid bodies can be assembled together intuitively by using drag & drop from imported CAD assemblies and parts. The STEP interchange format is supported to let designers use various CAD systems that they have expertise in. Unfortunately there exists no widely-spread data exchange format yet, in which many CAD system developers would agree to include the constraints between the objects being designed. Therefore, the joints must be placed and connected with body elements in *VINCENT* manually, as well. However it seems to be a tedious work, after some practice a complete machine configuration can be defined with 40 bodies and 30 joints within a few hours. Once the structure has been configured, it can easily be reused upon changing geometries or some other parameters. The Modelica model is directly generated out of the information in *VINCENT* using the same workflow as in [1]. Figure 3 shows the generated model according to the simple *ERMA* 3-axis robot. As long as the internal details are also modeled, the physical properties (mass, inertia, center of gravity) of rigid body elements with known uniform density can be simply derived from the fine-triangulated geometry models [3]. In case of simplified 3rd party CAD parts with no modeled interior the mass properties from datasheets of the manufacturer must be used instead.

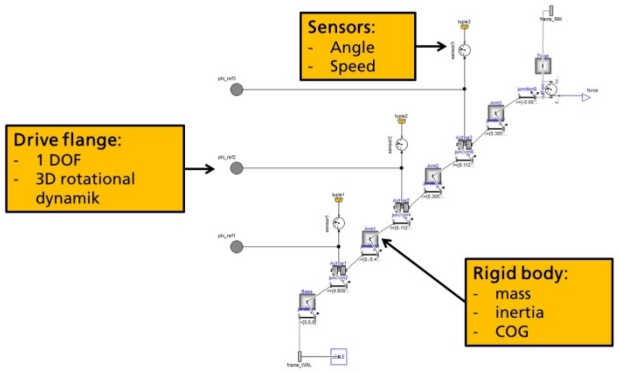


Figure 3: The generated multi-body model of the ERMA demonstrator

2.2 Usage of the Modelica Model during Sizing of the Drives

During its planned pick-and-place job of transporting alloy wheels the *ERMA* demonstrator must reach various points in space as fast as possible. At the early phase of the project the aforementioned multi-body model was used to compute acceleration limits. For this reason a new Modelica block was developed supporting point-to-point path interpolation of arbitrary goal positions within the workspace of a robot. Similar to the *KinematicPTP2* model in the standard Modelica library, this path interpolator allows the specification of kinematic constraints (maximum velocity and acceleration) of each output. Besides the start and end positions any number of intermediate locations can be given, too. As a result, time-optimal trajectories between adjacent goal positions are being computed.

By using ideal position sources the simulator could compute the required torques during the sizing of the demonstrator. The simulated torque curves were taken to select the proper drive characteristics to be used. After multiple simulations involving iterative model updates using manufacturer data, a proper drive selection could be found. Table 1 summarizes the chosen drives including the gear ratios:

Table 1: Selection of Wittenstein-alpha permanent-magnet synchronous drives of the ERMA demonstrator

	Axis1	Axis2	Axis3
	TPM+ power 110	TPM+ power 110	TPM+ high torque 025
Gear ratio (stages):	1:7 (1x)	1:35 (2x)	1:55 (2x)
Stall torque [Nm]:	(vertical)	878	214
Max. torque [Nm]:	600	1600	530

2.3 Mechatronic Model of the Intermediate DC Circuit and the Electric Drive Subsystem

The prediction of future power requirements needs more than analyzing a pure mechanical model. Only an interdisciplinary model can compute the correct energy balance including the losses in other domains. The *Smart Electric Drives* Modelica Library has been developed at the *Austrian Institute of Technology* [8]. The SED library allows convenient modeling and simulation of an entire electric drive subsystem. It provides components of various electrical devices including motor models with power electronics and also energy storage modules. As both thermal and transient magnetic effects are taken into consideration, these models can fulfill the purpose of computing realistic power requirements. Figure 4 shows the diagram of a synchronous permanent magnet (SMPM) machine including field-oriented control and voltage and current limitations. A very similar control strategy is used in the real motor modules of the *ERMA* demonstrator.

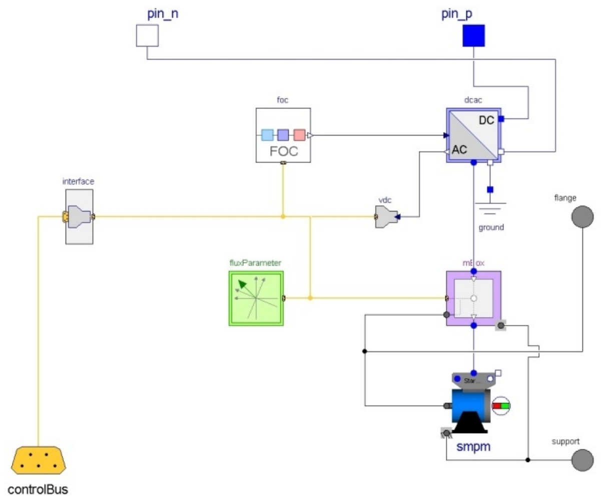


Figure 4: Field-oriented controlled permanent magnet synchronous induction machine

This torque-controlled SMPM machine model from the SED library is to be connected to the intermediate DC supply circuit. It utilizes an internal DC/AC frequency converter and a three-phase synchronous induction machine model with permanent magnet excitation. The input torque is to be computed externally: a position controller can retrieve the required electrical state variables over a bus connector. The mechanical flange is connected to the respective revolute joint in the mechanical model.

Based upon the specifications from the manufacturer the drive models were parameterized with real data. Measurements were carried out at the *aradex AG* to validate some model parameters such as winding resistances and inductances. The electric parameters of the AC/DC rectifier module were taken into account in a conventional three-phase diode bridge rectifier model from the SED library. It is extended with buffering capacitors to model the intermediate DC circuit that is built into the real electric cabinet. Figure 5 shows the diagram of the drive subsystem in Modelica:

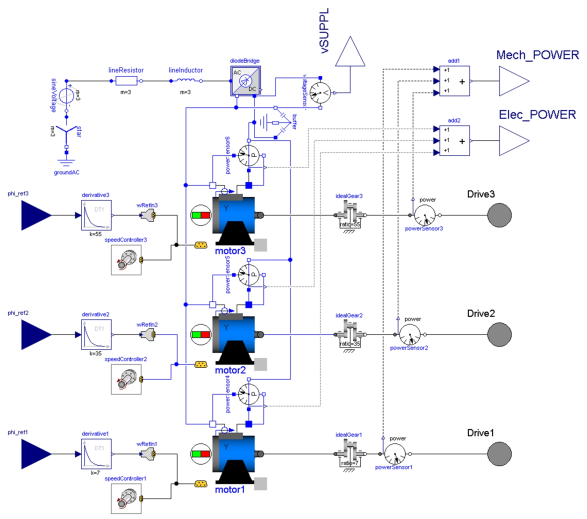


Figure 5: Mechatronic model of the electric drive subsystem including the DC intermediate supply circuit

The drive subsystem model of the demonstrator includes a 400V three-phase AC power source, the DC circuit model and three synchronous permanent magnet motor blocks with an ideal gear and a position controller per each instance. The three-phase motor model uses space phasor formulation with d and q axes [4] and it considers thermal losses in the rotor bushing (mechanical friction) and electrical losses (eddy current, ohmic winding resistance), as well. Permanent magnet excitation is modeled by a constant equivalent excitation current feeding the d-axis:

$$I_e = \frac{1}{\sqrt{2}\pi}\frac{U_{OC}}{L_{md}f_{nom}} \qquad (2)$$

where U_{OC} is the RMS open circuit voltage at nominal speed, L_{md} is the main field inductance in d-axis and f_{nom} is the nominal frequency. These parameters could be found on the datasheet of the real motors. The here implemented flux-weakening control strategy weakens the magnetic rotor field so that the stator voltage is restrained to given voltage limitations even though the shaft speed rises.

The derivatives of the input axis position signals are fed to the proportional integral speed controllers. The internal gear ratio is taken here into consideration as a gain factor. The reference torque of each motor is computed according to the aforementioned field-oriented control. Feedback signals are available over the bus connector (see Figure 4).

The included standard Modelica power sensors allow measuring the momentary mechanical and electrical power consumption. The sums of those are the *POWER* outputs of this subsystem model. This way, after mathematical integration of the electric power signal, the energy consumption of arbitrary axis movements can be determined.

2.4 The Concept of the "Energy" Axis

In this work the energy demand of a production machine needs to be pre-computed with the presented mechatronic model according to original machine control programs. The energy demand within a small amount of time frame determines the next reference angular velocity of the flywheel in the ERS. For this reason an *ENERGY* spindle axis was defined in the Sinumerik controller of the *ERMA* demonstrator. This fourth axis represents the reference flywheel speed. Due to its inertia, the flywheel must accelerate earlier, in order to be able to recuperate the generated electric currents during a braking phase. However, the axes within the same channel of an NC controller are always controlled together as a bundle. There is no direct "predictive" way to let one of them move earlier than the others.

There is a solution to this problem by means of NC curve tables. A curve table is a way to look up and interpolate values of a *follower* machine axis based on the position of a *leader* axis. For this purpose there can be arbitrary pairs of leader-follower axis positions stored in the rows of a curve table. The only rule is that the leader positions must be increasing or decreasing monotonously. Upon moving the leader the NC controller interpolates a smooth movement of the follower. In the *ERMA* case the aforementioned *ENERGY* axis takes the role of the follower in each curve table definition. Upon each NC movement command there is at least one machine axis involved. As the axis interpolation during movements is always monotonous, a leader machine axis can always be chosen respectively.

2.5 Coupling with the VNCK Service

As briefly introduced in chapter 1.2, there is a Windows service that emulates the core functionality of a real Sinumerik NC controller. The Virtual NC Kernel (VNCK) service registers a COM (Component-Object-Model) interface for easy integration into other Windows-based applications. Using this interface a Microsoft C# .NET-based GUI application called *IFF VNCK Manager* was developed. It can attach the Modelica simulation to the VCNK service and perform the transformation of an original NC program into a new form that includes the usage of curve tables to control the flywheel-based ERS.

Figure 6 shows the top-level view of the *ERMA* demonstrator model with VNCK coupling:

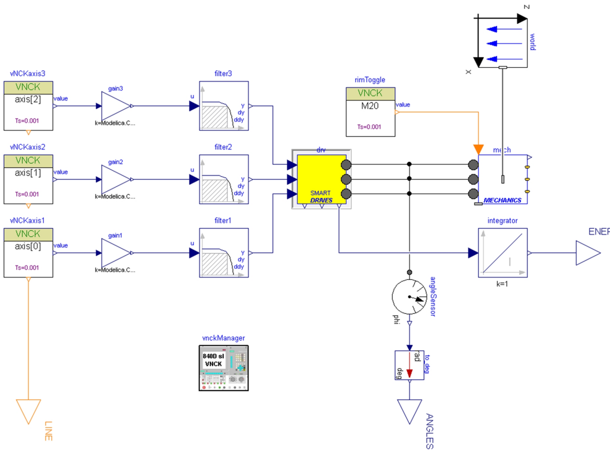
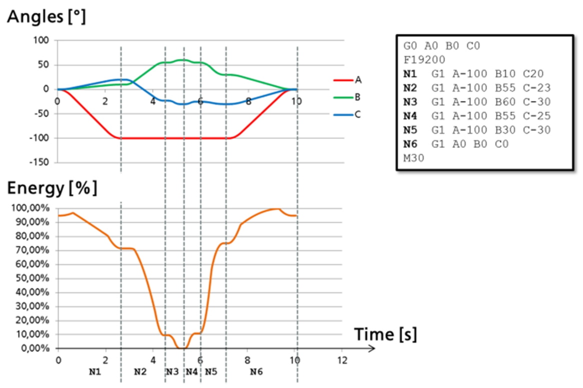


Figure 6: Top-level model of the ERMA demonstrator with VNCK binding

On the part of Modelica, a new *VNCK Library* has been created in the Fraunhofer IFF. It includes external C functions for binding, communicating with and finally stopping the VNCK service over a .NET/Managed C++ wrapper within the *ERMA* .NET solution. Modelica blocks were also developed for retrieving interpolated axis values and machine states at discrete time events. In the scenario of the demonstrator the latter way is used to detect commands for gripper control influencing the mass being transported. As the real gripper can programmatically get hold of or release a car wheel during the process, the load can vary over time. However, the virtual wheel body must always remain attached to the simulated robot's gripper, because the model structure cannot be changed in Modelica during the simulation. Therefore a mass override input variable has been added to the standard Modelica rigid body model. After a gripper release command is decoded by the VNCK, the mass of the wheel is set to zero immediately.

There is a single *VCNKManager* instance placed in this topmost Modelica model: the VNCK service is started during its initialization (in the initial equation section). The VNCK axis reader blocks are referencing this singleton instance over inner / outer Modelica constructs. During the processing of the original (textual) NC program, the VNCK is configured to report the source line positions, as well. This *LINE* integer output of the model is used to split the computed signals into multiple curve table definitions, as described later. According to an example NC program Figure 7 illustrates the splitting positions of the output signals (yielding six sections for six *G1* commands):



```
G0 A0 B0 C0
F19200
N1  G1 A-100 B10 C20
N2  G1 A-100 B55 C-23
N3  G1 A-100 B60 C-30
N4  G1 A-100 B55 C-25
N5  G1 A-100 B30 C-30
N6  G1 A0 B0 C0
M30
```

Figure 7: Splitting simulation signals according to NC statement blocks

The beginning of a new line with a movement command (in the example of Figure 7 those lines are labeled with **N1...N6**) is decoded by the VNCK and reported to the VNCK Manager.

The smallest step size of the VNCK service for axis-interpolation is 12 milliseconds. As only relatively coarse axis angles are being output therewith, filters must be applied in Modelica to smooth and reconstruct the continuous position reference signals. Each *VNCKaxis* block has a third-order Butterworth filter (10Hz cut-off frequency), and an additional first-order low-pass filter is applied in sequence, too. According to the smooth input signals the drive subsystem (Figure 5) can now compute the total power requirement at each simulation step (1ms). The smooth machine angles together with the integrated energy signal and the current NC source *LINE* signal are being output to the common simulation result file at equidistant intervals of 1ms.

The model of the machine being simulated can be arbitrary complex. There is no requirement of real-time capability, because buffering mechanisms are used with semaphores within the .NET VNCK Manager application to synchronize the VNCK service and the Modelica simulation. The complete mechatronic model of the three-axis *ERMA* demonstrator is approximately 2 times slower than real-time. If the simulation would be faster and could overtake the VNCK process, it is forced to wait for the VNCK to yield the goal machine angles. For this reason the external C function call in each *VNCKaxis* can block the calling simulator thread until the buffer receives the next VNCK output (@ 83Hz / every 12msec) greater than or equal to the time step.

However, in case of *ERMA* the VNCK can compute three interpolated angles much faster than real-time. Therefore, the internal buffer of target angles is filled beforehand, thus buffer synchronization occurs only upon the beginning of the simulation.

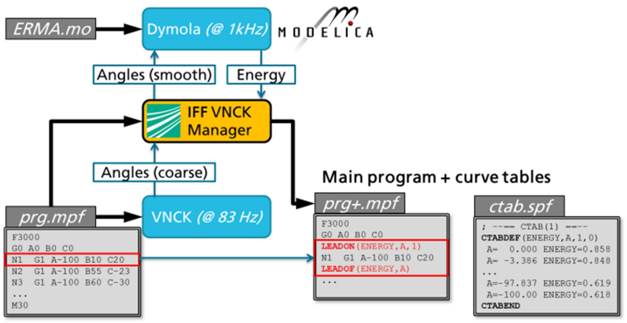
The *VNCKManager* block shuts down the VNCK service and the simulation after receiving a *terminate* notification at the end of the interpreted NC program (*M30* command). After this event, the .NET application can start to process the simulation results.

2.6 Flywheel Control Strategy

Within the *ERMA* project a new strategy had to be developed to transform the computed energy levels to appropriate flywheel speeds, thus allow the ERS to reduce the power consumption of a production machine. There are multiple permanent consumers in the electric cabinet that cannot be eliminated: such as frequency inverters, decentralized peripherals (switched magnetic valves, limit switches, etc.) or the Sinumerik controller itself. Besides further thermal losses in the system their static power consumption could neither be modeled nor compensated with the flywheel strategy.

According to the *ERMA* concept the total sum of the flywheel energy in the ERS together with the potential and kinetic energy of the machine must be kept at a constant energy level. The range of the computed *ENERGY* signal can be determined after the simulation. This has to be transformed into a range of flywheel speed according the inverse of the formula (1) in chapter 1.1. Right at the moment where the energy signal has its maximum, the flywheel must rotate at its lowest (idle) speed. Similarly, the upper flywheel speed-limit must correspond to the lowest energy state of the machine during the process. Using higher flywheel speed as needed would increase the friction losses, thus an optimum had to be found. The ideal flywheel speed range is also influenced by the maximum amount of energy that can be recovered during the whole process. In case of typical fast pick-and-place tasks in a production line the flywheel of the ERS never stops, because it has relatively high power consumption upon startup accelerations. In case of the *ERMA* demonstrator the range of 5400-6800 rpm has been determined experimentally. Within this range of 1400 rpm 1300J of process energy can be stored theoretically.

As mentioned earlier, the simulation result file contains data about the original NC program lines, too. The splitting of the result file occurs at each moment the *LINE* signal changes. The example of Figure 7 yields six individual sections after the simulation. A new curve table definition is then created out of each section. It is not to forget that the flywheel must be controlled in advance. Therefore, prior writing the corresponding leader-follower axis values into the output curve table, the section of the energy signal is shifted 54 milliseconds forwards in time (towards prediction). This amount of time was determined empirically, after exhaustive testing with the real energy recuperation system. Figure 8 shows the software tool chain that is used in this work. A .NET solution "*IFF VNCK Manager*" has been created to modify an existing main NC program (*.mpf*) with the incorporation of curve tables, thus the NC controller is able to control the energy axis of the ERS transparently during the real executed process.



```
F3000
G0 A0 B0 C0
N1  G1 A-100 B10 C20
N2  G1 A-100 B55 C-23
N3  G1 A-100 B60 C-30
...
M30
```

```
F3000
G0 A0 B0 C0
LEADON(ENERGY,A,1)
N1  G1 A-100 B10 C20
LEADOF(ENERGY,A)
...
```

```
; ---- CTAB(1) ----
CTABDEF(ENERGY,A,1,0)
 A=  0.000 ENERGY=0.858
 A= -3.386 ENERGY=0.848
...
 A=-97.837 ENERGY=0.619
 A=-100.00 ENERGY=0.618
CTABEND
```

Figure 8: Software chain to create modified NC programs with ERS control capability

As there are usually multiple axes moving upon a single NC command, the leader axis of a curve table definition is chosen according to the biggest position difference being travelled. In the example of Figure 7 the role of the leader axis in the first three curve tables is A, B and C, respectively.

3 Summary

In this paper a new concept of including a flywheel-based energy recuperation system into production machinery is presented. Based on a semi-automated workflow of creating a detailed, realistic mechatronic Modelica model of the system and coupling with a virtual controller, the movements to be executed can be assessed energetically in advance. This information can be used to control a mechanical energy storage unit earlier, overcoming its intrinsic flywheel inertia, which otherwise would hinder effective energy recuperation.

The demonstrator of the here presented *ERMA* project is located in the Virtual Development and Training Centre of the Fraunhofer IFF in Magdeburg, Germany. Using this methodology a 12% reduction of the machine's energy consumption was achieved.

4 Acknowledgement

The here presented work has been funded by the German Federal Ministry of Education and Research within the Resource and Energy Efficiency Program under the project named "*ERMA* - Energy Recuperation for Multi-Axis Machines" (02PK2192).

References

[1] Juhasz, T.; Schmucker, U.: "Automatic Model Conversion to Modelica for Dymola-based Mechatronic Simulation"; Proc. of 6th International Modelica Conference, 3rd-4th March, 2008, Bielefeld, Germany, Vol. 2, pp. 719-726

[2] Thielicke, R. ; Adam, T. ; Böhme, T. ; Kennel, M.: "Entwicklung eines Werkzeuges zur automatisierten NC-Code-Generierung mit maschinennaher NC-Simulation" In: IFF-Wissenschaftstage, 2011 — ISBN 978-3-8396-0281-2

[3] Mirtich, B.: "Fast and Accurate Computation of Polyhedral Mass Properties" Journal of Graphics Tools, Volume 1, Number 2, 1996

[4] Schröder, D.: "Elektrische Antriebe - Regelung von Antriebssystemen"; Springer Verlag 2009; ISBN 978-3-540-89612-8

[5] rosseta Technik GmbH - http://www.rosseta.de/

[6] aradex AG - http://www.aradex.de/en

[7] VINCENT - http://www.iff.fraunhofer.de/de/geschaeftsbereiche/ virtual-engineering/vincent.html

[8] AIT - www.ait.ac.at