POTENTIAL OF STATE-FEEDBACK CONTROL FOR MACHINE TOOLS DRIVES

L. Novotny[1], P. Strakos[1], J. Vesely[1], A. Dietmair[2]
[1] Research Center of Manufacturing Technology, CTU in Prague, Czech Republic
[2] SW, Universität Stuttgart, Germany
e-mail: l.novotny@rcmt.cvut.cz

This paper describes basic principle of the state feedback control technique and its potential for the feed drive control in the machine tool domain. It evaluates the quality of the state control and compares it with the traditional cascade control. The emphasis is put on an estimation of the state variables using an FEM model of the feed drive mechanical structure and machine frame.

Keywords
Machine tools, drives and control, state – feedback control, state estimation

1. Introduction

With the increasing demands for speed and dynamics of cutting processes go hand in hand also demands on dynamics of the control. We can see now that increasing control parameters reach their limitations from the mechanical frequencies of machine tools point of view. These frequencies influence the control loop through a measuring system that is a part of the machine tool. This paper refers to the possible solutions of this problem using the state-feedback control.

2. Comparison of the state-feedback cascade control approach

Cascade control cannot directly influence the characteristic frequencies of the machine. It only damps their influence by reducing the control parameters. Doing this also reduces the dynamic abilities of the drive, but it is undesirable in view of the global technological requirements.

To eliminate machine vibrations, there is the need to get some additional data (i.e.: data in real time) using an accelerometer and then to process these data in the control loop. This problem should be solved by state-feedback regulation. Possibilities of the state-feedback regulation will be demonstrated on the simplified model at first. This will be the theoretical basis for further expansion, which will enable the application in more complex systems of machine tools feed drives.

2.1 Simple example of machine tool drive

The problem will be illustrated using a very simplified case. Let us assume the oscillation problem of rigid machine tool base with linear drive positioning axis (see fig.1).

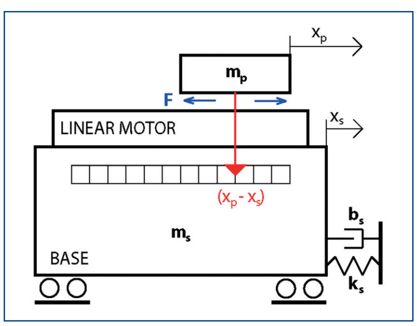

Figure 1. *Model of feed drive with linear motor*

Mechanical properties of machine tool are for this case simplified on one-mass system with 40 Hz resonance frequency. We also assume that all states are measurable (next chapters deal with estimation problem of measurable states). This mechanical system is then described by the system of state equations

$$\begin{aligned} \dot{q}_{p1} &= q_{p2} \\ \dot{q}_{p2} &= \frac{1}{m_p} F \\ \dot{q}_{S1} &= q_{p2} \\ \dot{q}_{S2} &= -\frac{k_S}{m_S} \cdot q_{S1} - \frac{b_S}{m_S} \cdot q_{S2} - \frac{1}{m_p} F \end{aligned} \tag{1}$$

where the state variables are:

$$\begin{aligned} q_{p1} &= x_p - \textit{position of primary} \\ q_{p2} &= \dot{x}_p - \textit{velocity of primary} \\ q_{S1} &= x_S - \textit{position of based} \\ q_{S2} &= \dot{x}_S - \textit{velocity of based}. \end{aligned}$$

Matrix of state-space then gets following form

$$\mathbf{A} = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -\frac{k_S}{m_S} & -\frac{b_S}{m_S} \end{pmatrix} \qquad \mathbf{B} = \begin{pmatrix} 0 \\ \frac{1}{m_p} \\ 0 \\ -\frac{1}{m_S} \end{pmatrix} \tag{2}$$

$$\mathbf{C} = \begin{pmatrix} 1 & 0 & -1 & 0 \end{pmatrix} \qquad \mathbf{D} = 0$$

The basis of state-feedback controller is introducing of the state variables vector multiplied by vector of feedback gains *K* (see figure 2).

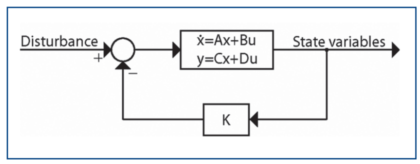

Figure 2. *Fundamental state-feedback control scheme*

Poles that are the solution of the characteristic equation determine the dynamics of this machine tool.

$$t|\mathbf{A} - \mathbf{BK}| = 0. \tag{3}$$

It is possible to prove that *K* affects all poles of the feedback system. In other words it means that the system is controllable. We can directly influence dynamics of the machine by choosing the gain vector *K*.

Feed axes of real machine tools are better described by common multi-mass models. Technically these models are more precise, but still analogical to one-mass system model. The difference is in dynamics. Dynamics of multi-mass system is constituted by more poles. System is also controllable nevertheless with longer vector of the state-feedback gain (it is represented by more states).

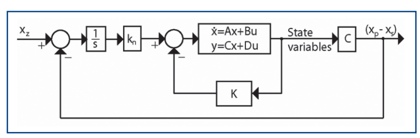

Figure 3. *State-feedback control scheme used for feed drives*

State-feedback scheme, shown in the figure 2, is not applicable for drive position control. It introduces a constant steady state position deviation in reaction to the disturbance step change. It is not possible to set precisely the required output value. This is the reason why we have to add an integrator and the position control input (according to the scheme in the figure 3). It enables the precise control of the drive position and the disturbance influence suppression at the same time. The system will be extended by the state variable $\boldsymbol{q}_n$. This state variable characterizes the position setpoint. The complete system is now expressed as follows

$$\begin{pmatrix} \dot{q}_{p1} \\ \dot{q}_{p2} \\ \dot{q}_{S1} \\ \dot{q}_{S2} \\ \dot{q}_n \end{pmatrix} = \left\{ \begin{pmatrix} \mathbf{A} & \mathbf{0} \\ -\mathbf{C} & \mathbf{0} \end{pmatrix} - \begin{pmatrix} \mathbf{B} \\ \mathbf{0} \end{pmatrix} \cdot \mathbf{K_n} \right\} \cdot \begin{pmatrix} q_{p1} \\ q_{p2} \\ q_{S1} \\ q_{S2} \\ q_n \end{pmatrix} + \begin{pmatrix} \mathbf{0} \\ 1 \end{pmatrix} \cdot x_z \tag{4}$$

where from the characteristic equation

$$\det \left| s\mathbf{I} - \begin{pmatrix} \mathbf{A} & \mathbf{0} \\ -\mathbf{C} & \mathbf{0} \end{pmatrix} - \begin{pmatrix} \mathbf{B} \\ \mathbf{0} \end{pmatrix} \cdot \mathbf{K_n} \right| = 0 \tag{5}$$

It is clear that system is directly controllable by the state-feedback gain vector.

Gains for state-feedback regulation were selected with respect to the limitation of maximal possible current consumption of the linear motor.

The same mechanical structural model was controlled by using the cascade controller. Maximal possible parameters of velocity and position control loop have been set according to control principles of machine tools drives. In the picture we can see the comparison of position controls pole-placements between state-feedback and cascade control. It is obvious that in the case of state-feedback we can significantly shift the poles to the left by choosing the gain vector. It is important to mention that in this case of the state-feedback control the overlapping of poles has been chosen. This is the reason why it might look like there is lower amount of poles in comparison with cascade control.

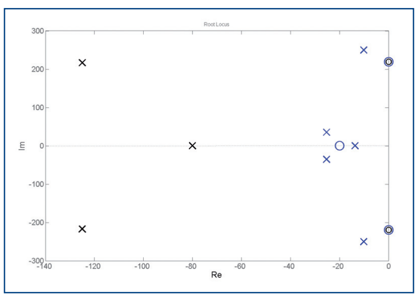

Figure 4. *Position control loop root-locus comparison. Blue color – cascade controller; Black - state-feedback control*

In the following pictures we have introduced other important tests that present significantly better properties of state-feedback control in comparison to cascade control.

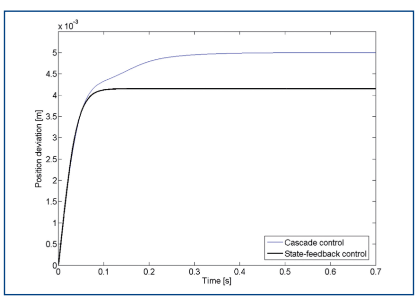

Figure 5. *Position deviation for cascade and state-feedback control*

It is necessary to notice that the above-mentioned example only simulates possible advantages of state-feedback application. Following considerable simplifications were adopted – measurability of all state variables, arbitrary placement of poles etc. These factors are discussed in the next chapters.

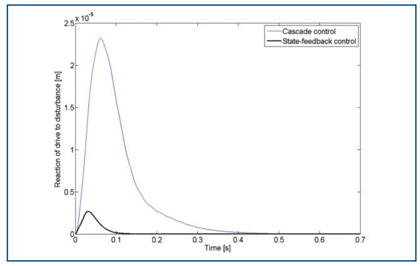

Figure 6. *Drive reaction to disturbance for cascade and state-feedback control*

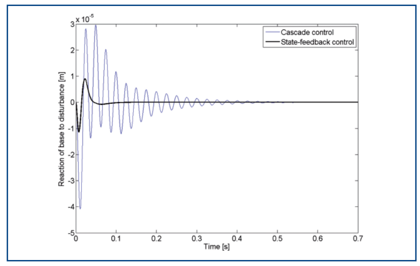

Figure 7. *Reaction of base to disturbance for cascade and state-feedback control*

3. State-feedback via LQR technique tuned by genetic algorithm

Model of the real mechanical system can be very advantageously described by (6)

$$\ddot{\boldsymbol{q}} + \mathbf{C_q} \cdot \dot{\boldsymbol{q}} + \boldsymbol{\Lambda} \cdot \boldsymbol{q} = \mathbf{V^T} \cdot \boldsymbol{f} \tag{6}$$

where $\boldsymbol{q}$ is the vector of modal coordinates, $\mathbf{C}_q$ is the matrix of modal damping, Λ is the spectral matrix, $\boldsymbol{V}$ is the modal matrix and $\boldsymbol{f}$ is the vector of external forces. The model expressed by (6) offers to reduce the original full-size model coming from modal analysis by choosing only the significant eigenmodes which play the key role in the system dynamics. This can be done without reducing the number of geometrical degrees of freedom of the original model.

State-feedback for such system is then designed by means of the LQR technique. Unfortunately since the state vector acting in the feedback law is expressed in modal coordinates, it is hard to effectively tune the state-feedback. This problem is overcome by using the genetic algorithms as a tool for tuning the state-feedback.

3.1 LQR as a technique for computation of the system state-feedback gain

The LQR as a control technique from the modern control theory group offers to find the state-feedback gains for the controlled system in the way they minimize the quadratic cost function [1]. For the purpose of explanation, the continuous time system and the corresponding continuous variant of the LQR technique are presented.

Let's suppose we have a dynamical system described by (6). This equation can be rewritten to state-space form (7) suitable for the application of any state-feedback control technique.

$$\dot{\boldsymbol{q}}_S = \mathbf{A} \cdot \boldsymbol{q}_S + \mathbf{B} \cdot \boldsymbol{u} \tag{7}$$

In equation (7) ***qs, A, B*** and ***u*** are expressed as follows

$$\boldsymbol{q}_S = \begin{bmatrix} \boldsymbol{q} \\ \dot{\boldsymbol{q}} \end{bmatrix}, \quad \mathbf{A} = \begin{bmatrix} \mathbf{0} & \mathbf{E} \\ -\boldsymbol{\Lambda} & \mathbf{C_q} \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} \mathbf{0} \\ \mathbf{V^T} \end{bmatrix}, \quad \boldsymbol{u} = \boldsymbol{f} \tag{8}$$

where ***E*** stands for unity matrix and ***0*** for zero matrix.

For the continuous time system (7), the state-feedback law (9) minimizes the quadratic cost function (10) by solving the associated Riccati equation (11)

$$\boldsymbol{u} = -\mathbf{K} \cdot \boldsymbol{q_s} \tag{9}$$

$$\mathrm{J}(\boldsymbol{u}) = \int_0^\infty \left(\boldsymbol{q_s^T} \cdot \mathbf{Q} \cdot \boldsymbol{q_s} + \boldsymbol{u^T} \cdot \mathbf{R} \cdot \boldsymbol{u} \right) dt \tag{10}$$

$$\mathbf{A^T} \cdot \mathbf{S} + \mathbf{S} \cdot \mathbf{A} - \mathbf{S} \cdot \mathbf{B} \cdot \mathbf{R^{-1}} \cdot \mathbf{B^T} \cdot \mathbf{S} + \mathbf{Q} = \mathbf{0} \tag{11}$$

Matrix Q in the equation (10) and (11) is the weight matrix and it is typically a diagonal matrix where the elements on the diagonal reflect the importance of each state in the designed feedback. Meanwhile the matrix R is the matrix of control corresponding to the effect of control inputs.
The *K* vector, which is the vector of the feedback gains, is then computed as follows

$$\mathbf{K} = \mathbf{R^{-1}} \cdot \mathbf{B^T} \cdot \mathbf{S} \tag{12}$$

For the purpose of computation the LQR procedure from MATLAB software has been used.

3.2 Genetic algorithm as a technique for tuning the state-feedback

When the system described by (7) is rather large while also the state vector is expressed in modal coordinates, it is very difficult to effectively tune the state-feedback while also meeting the user demands. This is mainly caused by the non-physical meaning of the state vector coordinates. This fact causes a problem in LQR design technique. By means of LQR the state-feedback is tuned via coefficients in **Q** matrix, but since the state vector is expressed in modal coordinates we have no idea how to set the **Q** matrix to directly influence the system outputs having a physical meaning to meet user demands. The so far used procedure was a trial and error method. In fact this is still the case but in much more sophisticated manner. The genetic algorithm technique to design the **Q** matrix has been used.

As a type of the genetic algorithm technique the already preprogrammed procedure from MATLAB toolboxes has been used [Houck].

Example

As an example the flexible mechanical structure driven by a ball screw feed drive has been chosen. Its character can be described by the frequency transfer function between the Tool Center Point (TCP) and the driving torque as depicted in fig. 8.

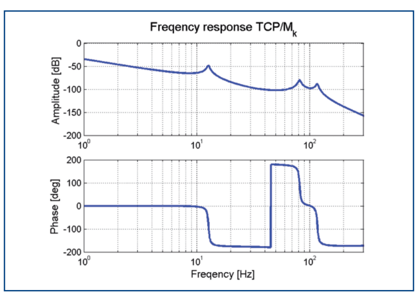

Figure 8. *Frequency transfer plot between TCP/Mk.*

The square **Q** matrix of the size 9x9 in the LQR state-feedback method had only diagonal elements while the non-diagonal elements were zero.

For the purpose of the genetic algorithm run the following setting that optimizes the **Q** matrix has been chosen:

- Size of the initial population was 100;
- Number of the generated populations was 400.

The following function as the fitness function has been used:

$$\mathrm{CF} = -\left[\mathrm{k}_1 \cdot t^7 \cdot (w - x_{TCP})^2 + \mathrm{k}_2 \cdot u^2 \right] \tag{13}$$

where coefficients $\boldsymbol{k_1}$ and $\boldsymbol{k_2}$ stands for the weight factors of each subcriterion in the function. First subcriterion in the equation (13) expresses the amount of position deviation between the command value *w* and the TCP position $\boldsymbol{x_{TCP}}$ while time ***t*** and its power in this subcriterion helps to suppress the oscillations of the TCP position. The second subcriterion in the equation (13) expresses the energy demands on the actuating torque ***u***.

The equation (13) can be rewritten into a simpler form

$$\mathrm{CF} = -\left[p_x + p_y \right] \tag{14}$$

where $\boldsymbol{p_x}$ represents the x value of the point in the pareto set and $\boldsymbol{p_y}$ represents the y value of the same point.

Results

Based on the above mentioned, the genetic algorithm procedure had been started and has shown the following results.

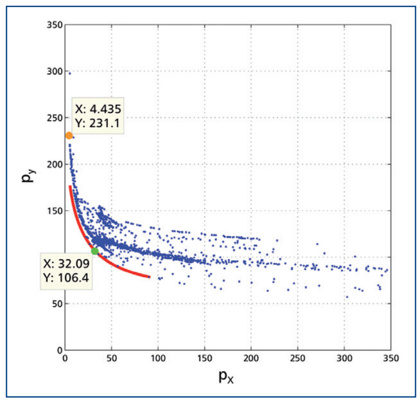

Figure 9. *Pareto set for the Q matrix optimization problem.*

In fig. 9 the pareto set there is a red border curve marking the best solutions. There are also two solutions marked by green and orange point signifying the lowest value of p_x factor and the lowest value of the fitness function. For these solutions the system reaction to the position ramp has been checked. In fig. 10 the time behavior of TCP position is visible while in fig. 11 the demanded torque command is shown.

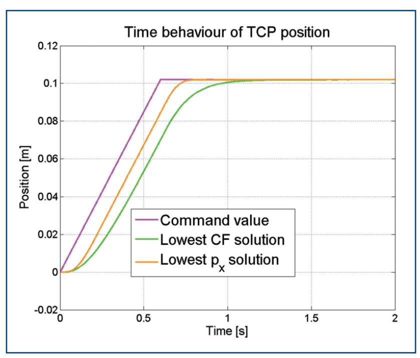

Figure 10. *Time behavior of TCP position.*

The lowest ***CF*** solution shows the optimum in both, lowest torque demand and the highest dynamical response at the same time while the lowest $\boldsymbol{p_x}$ solution represents the best dynamical response without dramatically restricted torque demand. Based on the above

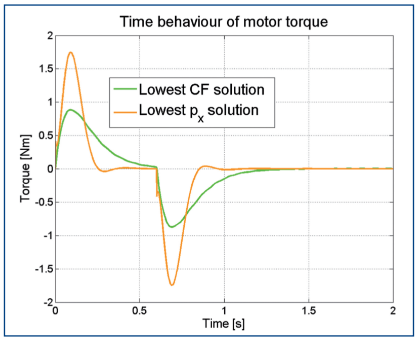

Figure 11. Time behavior of motor torque.

shown results the genetic algorithms proved good ability to be used as a tuning tool for LQR state-feedback technique.

4. Measurement of variables for state-feedback

4.1 Necessity of states estimation

Until now we have considered that all states necessary for the feedback can be measured. This case is almost always practically unachievable, because it requires too many sensors placed on the machine. In general, there are only one or two essential position measurement systems for each drive axis. The other signals must be reconstructed by using a state-observer.

There are many approaches for immeasurable states estimation. All of them can be tested in the simulations and it seems it works, but only few of them are robust enough to meet the needs of the real industrial application. Some estimators do not respect that their inputs signals contain noise and it causes a serious problems. The results of simple estimation task are presented on fig. 13. The aim was to estimate the position of the end-point of the rotor with known linear dynamical model. The efficiency of predictive and Kalman observer was studied. Both observers had the same input – the ideal torque command, but the real torque setpoint was distorted by noise (fig. 12). It is obvious that Kalman filter gives more precise estimation. It is more robust than simple predictive observer.

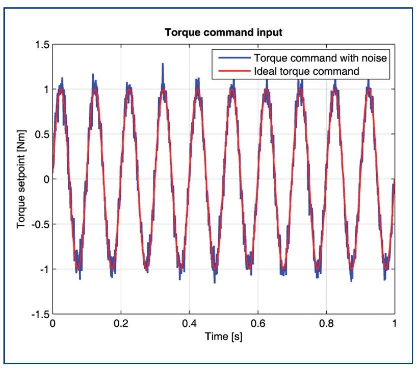

Figure 12. Available (red) value of torque setpoint is the ideal one. The real torque command can contain undefined noise.

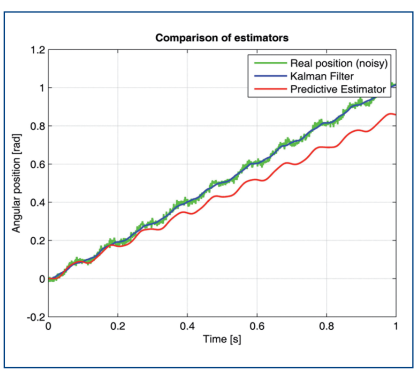

Figure 13. State estimation based on the ideal torque command. The predictive observer does not match the real measurement.

4.2 Discrete Kalman filter as a state estimator

The Kalman filter is a very interesting tool because it minimizes the variance of estimation error, so we can obtain a reliable estimate. We assume that control input u and the measurement vector y are affected by noises. They are assumed to be white, with normal probability distributions and independent of each other. We assume the control input noise covariance matrix Q and measurement noise covariance matrix R to be constant and in the form:

$$Q = (\sigma_u \cdot B)\cdot(\sigma_u \cdot B)^T \quad (15)$$

$$R = \sigma_y I \sigma_y^T \quad (16)$$

where σ_u and σ_y are the input noise standard deviation and measurement noise standard deviation respectively. These standard deviations must be suitably assessed.

The discrete Kalman filter is then running in the cycle. The first step is to predict the state and covariance estimates k from the step k-1. The second step is to correct the predicted estimates using available measurements and to calculate the Kalman gain.

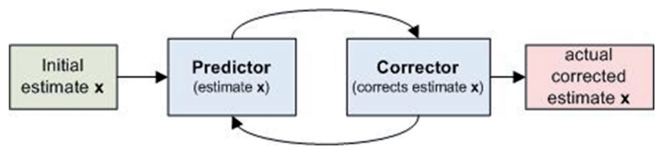

Figure 14. Kalman filter functional scheme

Predictor equations:

state estimate

$$\hat{\mathbf{x}}_k^- = \mathbf{A}\hat{\mathbf{x}}_{k-1} + \mathbf{B}\mathbf{u}_k \quad (17)$$

estimate error covariance matrix

$$P_k^- = AP_{k-1}A^T + Q \quad (18)$$

In our case we consider for the initial estimate

$$\hat{\mathbf{x}}_{k-1} = \hat{\mathbf{x}}_0 = 0 \quad (19)$$

$$P_{k-1} = P_0 = Q = (\sigma_u \cdot B)\cdot(\sigma_u \cdot B)^T \quad (20)$$

Corrector equations:

gain factor

$$K_k = P_k^- C^T \left(CP_k^- C^T + R\right)^{-1} \quad (21)$$

corrected state estimate

$$\hat{x}_k = \hat{x}_k^- + K_k\left(y_k - C\hat{x}_k^-\right) \quad (22)$$

corrected estimate error covariance matrix

$$P_k = (I - K_k C)P_k^- \quad (23)$$

4.3 Kalman filter implementation

The equations must be computed in real-time, with the same sampling time as the state-feedback control loop. This requires very fast computing hardware and a well reduced mathematical model of the driven structure. With increasing model sizes also the computational demands of matrix operations increase. In our laboratory conditions, the Kalman filter has been implemented by using National Instruments PXI control hardware and LabVIEW software (Figure 15).

Real industrial control applications require the implementation on cheap DSP/FPGA hardware or the machine control. In the EcoFIT Project, the Kalman Filter therefore is implemented into a commercial PC-based Numerical Control based on the Prosecco platform.

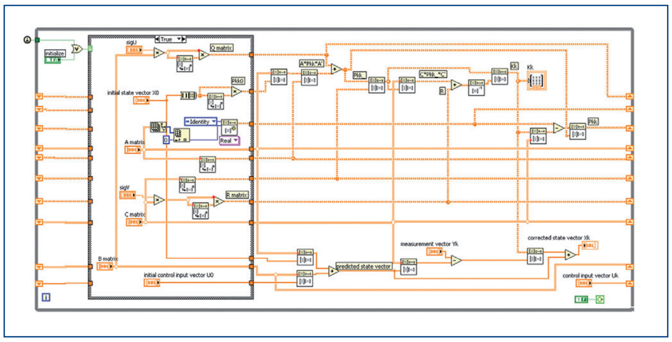

Figure 15. *Kalman filter realization in NI LabVIEW*

4.4 State Estimation and Real Time Simulation

Further measures have to be taken for machines whose properties vary over time and with position. In the ECOFIT project, state variables and resonances are thus computed using real-time simulation and used as input to the controllers and setpoint generation.

Three general solutions are possible for simulation based state estimation. Firstly, it would be possible to compute a Kalman filter with pre-selected matrices based on a linear approximation. For better results, an extended Kalman filter with a dedicated nonlinear model mapping the parameter variability can be used. This algorithm, in turn, can be improved by computing the linearization from the simulation model in real time.

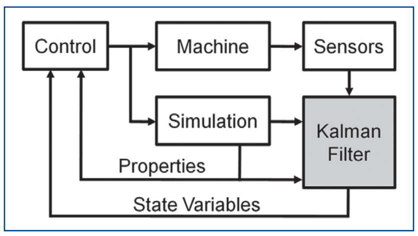

Figure 16. *Fusing sensor and simulation data for state vector reconstruction using a Kalman filter.*

All approaches can be based on a full system model. Another solution uses the Kalman filter as a means of fusing sensor data with simulation output to reconstruct immeasurable states for feedback (Figure 16).

5. Conclusions

It has been shown that the state-feedback control has a potential to improve the regulation. It has the ability to influence the dynamical behaviour of the mechanical structure of the machine tool. In practice it still encounters problems with state estimation, feedback signals influenced by noise, calculation speed, etc. Some solutions have been discussed in this paper. For industrial application it still requires numerous tests and modifications.

Acknowledgement

The project 1M0507 was financially supported by the Ministry of Education, Youth and Sports of Czech Republic. The authors appreciate kind support of the European Union 6th framework programme project NMP2-CT-2003-013989 "ECOFIT - Eco-efficient machine tools by means of radical mass and energy needs reduction" as well.

References

[Houck 1995] Houck, Ch., Joines, J., Kay, M.: A Genetic Algorithm for Function Optimization: A Matlab Implementation, MATLAB Help.

[Welch 2008] Welch, G., Bishop, G.: An Introduction to the Kalman Filter [online], [cited 2008-08-05] available on *http://www.cs.unc.edu/~welch/media /pdf/kalman_intro.pdf* .

[Zitek 2007] Zitek, P.: Basic issues of using models in control system design, 2007, Lectures, CTU Prague

[The MathWorks] The MathWorks, Inc., Linear-quadratic (LQ) State-feedback Regulator for State-space System, MATLAB Help.

Contacts:

Dipl. Ing. Lukas Novotny, CTU in Prague
Research Center of Manufacturing Technology
Horska 3, 128 00 Praha 2, Czech Republic
tel.: +420 221 990 0916, fax: +420 2 2199 0999
e-mail: L.Novotny@rcmt.cvut.cz