Mixed Hard and Soft Techniques in Nonlinear Control

Béla Lantos

Budapest University of Technology and Economics

Department of Control Engineering and Information Technology

H-1117 Budapest, Magyar Tudósok krt. 2

lantos@iit.bme.hu

Abstract: *The paper presents control algorithms for a two degree of freedom robot, the small size model of a planar crane and a weeled mobile robot which are based on nonlinear model based predictive control. The continuous time dynamic model has been discretized and the finite dimensional optimization problem is solved by conjugate gradient technique in every horizon. Extended Kalman filter is used for state and disturbance estimation. For the crane the initial approximation of the control sequence within the actual horizon is determined by using the flatness properties. Friction effects can be modeled by neuro-fuzzy techniques and integrated into the nonlinear predictive control algorithm. For development purposes the concept of a multiprocessor system has been elaborated where the control algorithm is running under QNX real-time operating system. The paper also presents experimental results that demonstrate the applicability of the proposed algorithms under real time conditions.*

Keywords: *Predictive control, Nonlinear systems, Robotics, Disturbance rejection, Real-time systems.*

1. Introduction[1]

Model predictive control is a popular method especially in the process industry where relatively slow process models allow online optimization. Linear predictive control is well elaborated both in frequency (operator) domain (Camacho, E. F. and C. Bordons, 2000; Lantos, 2003) and state space (Rossiter et al., 1998; Lantos,

[1] Support for the research was provided by the Hungarian National Research Programs under grant No. OTKA T042634 and NKFP 2/016/2001, and by the Control System Research Group of the Hungarian Academy of Sciences.

2003). Depending on the type of constraints optimal prediction leads to Quadratic Programming (QP) or Nonlinear Programming (NP) based on Sequential Optimization wich are well supported by existing softwares (e.g. Optimization Toolbox in MATLAB).

For nonlinear systems recent methods are usually based on new optimum seeking methods suited for the predictive control problem or traditional analytical optimum conditions and gradient based optimization techniques (Allgöver and Zheng, 2000; Kim and Shin, 2003). Basis for the later ones is a general form of the Lagrange multiplicator rule which is also valid in Banach spaces (Lantos, 2003).

Typical finite horizon nonlinear predictive control problems in discrete time lead to optimization in finite dimensional space where the variables are $x = \{x_i\}_{i=0}^{N}$ and $u = \{u_i\}_{i=0}^{N-1}$, the optimality criterion is

$$\begin{aligned} F_0(x,u) &= \frac{1}{2}\sum_{i=0}^{N-1}[< Q_i x_i, x_i > + < R_i u_i, u_i >] + \frac{1}{2} < Q_N x_N, x_N > \\ &=: \sum_{i=0}^{N-1} L_i(x_i, u_i) + \Phi(x_N), \end{aligned} \tag{1}$$

the constraints are the state equation

$$\varphi(x_i, u_i) - x_{i+1} = 0, \tag{2}$$

control set $u_i \in M$ and initial condition $a - x_0 = 0$.

If (x^*, u^*) is the optimal solution then

$$f(x,u) = J'_x(x^*, u^*)x + J'_u(x^*, u^*)u$$

is the derivative of

$$\begin{aligned} J(x,u) = F_0(x,u) &+ < \lambda_0, a - x_0 > + < \lambda_1, \varphi(x_0, u_0) - x_1 > + \cdots \\ &+ < \lambda_N, \varphi(x_{N-1}, u_{N-1}) - x_N > . \end{aligned}$$

By introducing the Hamiltonian functions as

$$H_i = < \lambda_{i+1}, \varphi(x_i, u_i) > + L_i(x_i, u_i),$$

the necessary condition of the optimality results in

$$\begin{aligned} \lambda_N &= Q_N x_N, \\ \lambda_i = \partial H_i / \partial x_i &= Q_i x_i + \left(\partial\varphi / \partial x_i\right)^T \lambda_{i+1}, \\ \partial H_i / \partial u_i &= R_i u_i + \left(\partial\varphi / \partial u_i\right)^T \lambda_{i+1}, \\ dJ &= \sum_{i=0}^{N-1} < \partial H_i / \partial u_i, u_i^* - u_i > \le 0. \end{aligned} \tag{3}$$

For the control design within the actual horizon first the initial condition x_0 and the approximation of u are needed (the latter may be the solution in the previous horizon shifted to the left).

The optimization repeats the following steps:

1. Solution of the state equations in $x = \{x_i\}_{i=0}^{N}$.
2. Computation of the Lagrange multiplicators λ_i.
3. Computation of the derivatives $\partial H_i / \partial u_i$.
4. Numerical optimization based on gradient type methods (gradient, conjugate gradient, Davidon-Fletcher-Powell etc.) to find $u = \{u_i\}_{i=0}^{N-1}$.

Non-predictive design method should be used to find the initial approximation for the first horizon. If the original sytem is a continuous time one then first it can be approximated by a discrete time one, e.g.

$$\dot{x} = f_c(x,u) \Rightarrow x_{i+1} = x_i + Tf_c(x_i, u_i) =: \varphi(x_i, u_i) \tag{4}$$

where T is the sampling time. If the full state can not be measured then x_0 can be approximated by using extended Kalman filter.

If $y = Cx$ is the system output and $\tilde{y} = y_d - y$ is the error then the cost function can be modified as

$$\tilde{y} = y_d - Cx,$$
$$2L_i(x_i, u_i) = < \tilde{Q}_i \tilde{y}_i, \tilde{y}_i > + < S_i x_i, x_i > + < R_i u_i, u_i >,$$
$$2\Phi(x_N) = < \tilde{Q}_N \tilde{y}_N, \tilde{y}_N >,$$

and the derivatives can be computed by

$$\partial L_i / \partial x_i = -C^T \tilde{Q}_i \tilde{y}_i + S_i x_i,$$
$$\partial \Phi / \partial x_N = -C^T \tilde{Q}_N \tilde{y}_N.$$

Input constraints are enforced by projecting u_i into the constraints set. State constraints can be taken into consideration as an additional penalty added to $L(x_i, u_i)$ in the cost function. It is known that the weighting in the term $\Phi(x_N)$ has great influence on the stability and dynamic behaviour of the sytem under predictive control (Allgöver and Zheng, 2000). To improve the stability properties the techniques of Frozen Riccati Equation (FRE) and Control Lyapunov Function (CLF) can be suggested (Yu et al., 2001). Unfortunately their real time realization is time consuming. In the paper experimentally chosen weighting terms have been applied.

The control design strategy can be summarized in the following steps:

1. Development of the nonlinear dynamic model of the system.
2. Optimal (suboptimal, flatness-based etc.) open loop control signal design used for initial approximation of the control sequence in the horizon.
3. Identification of the nonlinear friction model.
4. Elaboration of the disturbance model reduced on the system input.
5. Development of the model based nonlinear predictive controller and its use in closed loop control. The first element of the control sequence in the actual horizon is completed by the feedforward compensations of the friction and the disturbance.

2. Nonlinear Predictive Control of a 2-Dof Robot Arm

A two degree of freedom (DOF) open chain rigid robot having nonlinear friction effect $h_f(q,\dot{q})$ and dynamic model $H(q)\ddot{q}+h(q,\dot{q})+h_f(q,\dot{q})=\tau$ is considered first. Low level torque (current) control is assumed hence the control input is the torque τ . Inertia and friction parameters in the model are the results of previously performed identification and assumed known. The dynamic model of the robot arm (without the friction effect) is

$$\begin{aligned} D_{11}\ddot{q}_1+D_{12}\ddot{q}_2+2D_{121}\dot{q}_1\dot{q}_2+D_{122}\dot{q}_2^2+D_1&=\tau_1 \\ D_{12}\ddot{q}_1+D_{22}\ddot{q}_2-D_{112}\dot{q}_1^2+D_2&=\tau_2, \end{aligned} \tag{5}$$

where the parameters and functions are

$$\begin{array}{ll} \alpha_1=m_1l_{c1}^2+m_2(l_1^2+l_{c2}^2)+I_1+I_2 & D_{11}=\alpha_1+2C_2\alpha_2 \\ \alpha_2=m_2l_1l_{c2} & D_{12}=\alpha_3+C_2\alpha_2 \\ \alpha_3=m_2l_{c2}^2+I_2 & D_{22}=\alpha_3 \\ \alpha_4=g(m_1l_{c1}+m_2l_1) & D_{112}=-S_2\alpha_2=D_{122} \\ \alpha_5=gm_2l_{c2} & D_1=C_1\alpha_4+C_{12}\alpha_5 \\ & D_2=C_{12}\alpha_5 \end{array} \tag{6}$$

and $C_{12}=\cos(q_1+q_2)$ etc. as usual in robotics. In the experiments $m_1=m_2=5$, $I_1=I_2=1$, $l_1=l_2=1$, $l_{c1}=l_{c2}=0.5$ have been used (all in SI units). If $x=(q^T,\dot{q}^T)^T$ denotes the state then the nontrivial part of the state equation is $\ddot{q}=-H^{-1}h+H^{-1}\tau$.

The predictive control algorithm needs the derivative of the right side of the state equation by x and u . Meanwhile it can be utilized that if $f(x)=A^{-1}(x)b(x)$ is any smooth function then

$$\frac{\partial f}{\partial x_i} = -A^{-1} \frac{\partial A}{\partial x_i} A^{-1} b + A^{-1} \frac{\partial b}{\partial x_i} \tag{7}$$

The robot repeats a predefined path which allows that the desired $q_d(t+k)$ and $\dot{q}_d(t+k)$ are known in the prediction horizon. The initial approximation for the first horizon is based on the gravity compensation in the initial state and the compensation of the friction. Unknown disturbace (load etc.) consisting of first order deterministic polynomial and Gaussian noise was considered on the input of the robot (output of the controller). Experiments with estimated $\dot{q}$ and estimated disturbances d_1, d_2 reduced on the system inputs are performed. For estimation of the augmented state extended Kalman filter was used (Chui and Chen, 1999; Lantos, 2003).

Optimization is based on conjugate gradient technique. For horizon length $N=10$ satisfactory accuracy of the optimization can be reached within $T=25$ ms sampling time on standard processors. Fig. 1 shows the results. If needed, fine interpolation in closed loop is also possible based on $u(t)$ and $u(t+1)$ determined in the actual horizon. For large N basis functions (splines etc.) can be suggested.

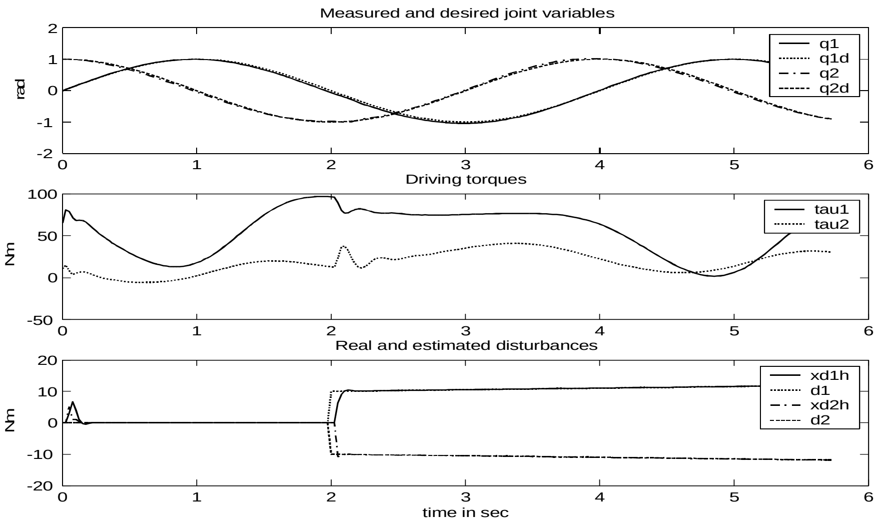

Fig. 1. Behaviour of the 2-DOF robot arm during predictive control.

3. Nonlinear Predictive Control of a 2D Crane

The second problem is the control of the small size model of a real cable-driven crane used by US Navy (Kiss et al., 2000). For simplicity only the two dimensional (2D) planar version is considered. The goal is the precise and quick motion of the load without overshoot. The scheme of the 2D crane is shown in Fig. 2. (Since the pulley's mass in point B has been neglected hence L_s and its actuator motor is omitted in the discussion). The planar crane is a 3-DOF system which is underactuated because the only inputs are the two motor torques moving the ropes L_1 and $L_2 + L_3$. The model of the planar crane is differencially flat (Kiss, 2001). In simplified formulation it means that if $\dot{x} = f(x,u)$ is the dynamic model of the system then there exist a new variable y (the flat output) and finite integers q and p such that $x = \phi_1(y, \dot{y}, \ddot{y}, \ldots, y^{(q)})$, $u = \phi_2(y, \dot{y}, \ddot{y}, \ldots, y^{(q+1)})$, $y = \psi_1(x, u, \dot{u}, \ldots, u^{(p)})$. For the crane $y := (x,z)^T$ where x, z are the coordinates of the load in the plane of the crane. The flatness relations for the 2D crane are summarized as follows:

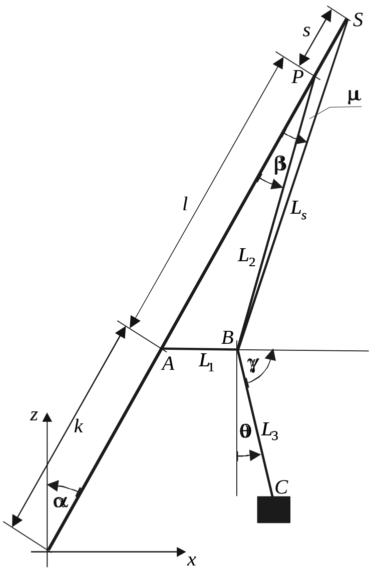

Fig. 2. Scheme of the 2D crane.

$$
\begin{aligned}
6 &= \arctan(-\ddot{x}/(\ddot{z}+g)) \\
d &= \frac{\ddot{x}(z-(k+l)C_\alpha) - (\ddot{z}+g)(x-(k+l)S_\alpha)}{(\ddot{z}+g)S_a - \ddot{x}C_a} \\
\beta &= \arcsin((1-d/l)S_{\alpha+\theta}) \\
\gamma &= (\pi + \beta - (\alpha+\theta))/2 \\
L_1 &= lS_\beta / S_\gamma \\
L_2 &= lS_{\gamma-\beta} / S_\gamma \\
x_B &= (k+l)S_\alpha - L_2 S_{\alpha-\beta} \\
z_B &= (k+l)C_\alpha - L_2 C_{\alpha-\beta} \\
L_3 &= \sqrt{(x-x_B)^2 + (z-z_B)^2} \\
T_3 &= m_{load}((\ddot{z}+g)C_\theta - \ddot{x}S_\theta) \\
u_1 &= T_1\rho_1 - \frac{J_1}{\rho_1}\ddot{L}_1 \\
u_2 &= T_3\rho_2 - \frac{J_2}{\rho_2}(\ddot{L}_2 + \ddot{L}_3)
\end{aligned}
\tag{8}
$$

If $y, \dot{y}, \ldots, y^{(4)}$ are known then all system variables can be computed from them except singularities. For prescribed initial y_0 and final y_1 and transient time T_F five times differenciable $y(t)$ polynomials can be found connecting the points with straight line from which all the system states $x(t)$ and inputs $u(t)$ can be reconstructed. Results u_1, u_2 of the flatness based design have been used for the initial approximation of the sequence u in all the horizons.

It follows from the equilibrium of the internal forces that the line AB in the direction of the horizontal rope L_1 is the bisectrix of the angle 2γ between PB and BC. Hence the magnitudes of the internal forces T_3 for L_2 and T_1 for L_1 are related by $T_1 = 2C_\gamma T_3$. By using the flat output all the variables discribing the 2D crane can be determined by (8). In order to avoid the first and second order symbolic differentiation of the nonlinear expressions for L_1, L_2, L_3, numerical differentiation was applied (see *spline*, *unmkpp*, *mkpp* in MATLAB). Finally the flatness based input torques u_1, u_2 for the motors can be determined, where J_i and ρ_i denote the inertia and the radius of the winch, respectively.

Assuming zero derivatives until order five in the initial and final positions, the 11th order polynomial connecting (x_0, z_0) and (x_1, z_1) has the form

$$x(\tau) = x_0 + (x_1 - x_0)\tau^6 \times [a_6 + a_7\tau + a_8\tau^2 + a_9\tau^3 + a_{10}\tau^4 + a_{11}\tau^5] \tag{9}$$

where $\tau = t / T_F$ is the normalized time and the coefficients for both $x(\tau)$ and $z(\tau)$ are

$$\begin{aligned} &a_6 = 0.462, \quad &a_7 = -1.980, \quad &a_8 = 3.465, \\ &a_9 = -3.080, \quad &a_{10} = 1.386, \quad &a_{11} = -0.252. \end{aligned} \tag{10}$$

Neglecting the feedforward compensated friction and the centripetal and Coriolis effects (because of the slow angular velocities of typical cranes) the basic equations of the underactuated system are

$$\begin{aligned} \frac{J_1}{\rho_1}\ddot{L}_1 &= T_1\rho_1 - u_1, \\ \frac{J_2}{\rho_2}(\ddot{L}_2 + \ddot{L}_3) &= T_3\rho_2 - u_2. \end{aligned} \tag{11}$$

The system is submitted to constraints with respect to L_1 and L_2 hence Lagrange multiplicators could have been used. Fortunately, instead of this relatively complicated method, the equilibrium between internal forces can be exploited, see (Kiss, 2001), who developed a form of the dynamic model for the choice $x = (\gamma, \alpha - \beta, L_3, \dot{\gamma}, (\dot{\alpha} - \dot{\beta}), \dot{L}_3)^T$ where $\alpha = \text{const}$. However the measured outputs are $L_1, L_2 + L_3, \dot{L}_1, \dot{L}_2 + \dot{L}_3$ which depend on these state variables in a nonlinear way making necessary the use of nonlinear state estimation and some modifications of the predictive control algorithm of (Kim and Shim, 2003). Hence a new model has been developed using $x = (L_1, L_2, L_3, \dot{L}_1, \dot{L}_2, \dot{L}_3)^T$. The idea is the use of the cosine theorem by which

$$C_\gamma = \frac{l^2 - L_1^2 - L_2^2}{2L_1L_2}, \; C_\beta = \frac{l^2 + L_2^2 - L_1^2}{2lL_2}. \tag{12}$$

For L_1 and L_2 the expressions in (8) can be used. By elementary geometry and using

$$\varphi = (\pi / 2) - (\gamma + (\alpha - \beta))$$

$$\theta = \pi - (2\gamma + (\alpha - \beta))$$

it yields

$$\begin{aligned} x &= kS_\alpha + L_1 C_\varphi + L_3 S_\theta \\ &= kS_\alpha + L_1 S_{\gamma+(\alpha-\beta)} + LS_{2\gamma+(\alpha-\beta)}, \\ z &= kC_\alpha + L_1 S_\varphi - L_3 C_\theta \\ &= kC_\alpha + L_1 C_{\gamma+(\alpha-\beta)} + LC_{2\gamma+(\alpha-\beta)}. \end{aligned} \tag{13}$$

The main problem is the computation of T_3 from the state variables which is needed in (11). This can be performed using $m\ddot{x} = -T_3 S_\theta$, $m(\ddot{z} + g) = T_3 C_\theta$ for the load, from which it follows

$$C_\theta \ddot{x} + S_\theta (\ddot{z} + g) = 0, \quad m(-S_\theta \ddot{x} + C_\theta (\ddot{z} + g) = T_3. \tag{14}$$

The remaining part is based on carefully performed symbolic differentiation to find the Jacobian (dx) and Hessian $(D2x)$ of the necessary functions, for example in the case of $x(t)$

$$\begin{aligned} \dot{x} &= dx \cdot (\dot{L}_1\ \dot{L}_2\ \dot{L}_3)^T, \\ \ddot{x} &= dx \cdot (\ddot{L}_1\ \ddot{L}_2\ \ddot{L}_3)^T + (\dot{L}_1\ \dot{L}_2\ \dot{L}_3) \cdot D2x \cdot (\dot{L}_1\ \dot{L}_2\ \dot{L}_3)^T. \end{aligned} \tag{15}$$

The dynamic model in MATLAB notation is the following:

$$\begin{aligned} &A(1,:) = C_\theta dx + S_\theta dz, \\ &b(1) = (\dot{L}_1\ \dot{L}_2\ \dot{L}_3)(-C_\theta D2x - S_\theta D2z)(\dot{L}_1\ \dot{L}_2\ \dot{L}_3)^T - S_\theta g, \\ &A(2,:) = 2C_\gamma m(S_\theta dx - C_\theta dz)\rho_1, \\ &A(2,1) = A(2,1) + J_1 / \rho_1, \\ &b(2) = (\dot{L}_1\ \dot{L}_2\ \dot{L}_3)(2C_\gamma m(-S_\theta D2x + C_\theta D2z)\rho_1)(\dot{L}_1\ \dot{L}_2\ \dot{L}_3)^T + 2C_\gamma m C_\theta g \rho_1, \\ &A(3,:) = m(S_\theta dx - C_\theta dz)\rho_2, \\ &A(3,2:3) = A(3,2:3) + [J_2 / \rho_2 \ \ J_2 / \rho_2], \\ &b(3) = (\dot{L}_1\ \dot{L}_2\ \dot{L}_3)(m(-S_\theta D2x + C_\theta D2z)\rho_2)(\dot{L}_1\ \dot{L}_2\ \dot{L}_3)^T + mC_\theta g \rho_2, \\ &(\ddot{L}_1\ \ddot{L}_2\ \ddot{L}_3)^T = A^{-1}b - A^{-1}(0\ u_1\ u_2)^T. \end{aligned} \tag{16}$$

$$(\ddot{L}_1\ \ddot{L}_2\ \ddot{L}_3)^T = A^{-1}b - A^{-1}(0\ u_1\ u_2)^T.$$

Unfortunately, for predictive control A and b have to be differentiated once more, see (3).

An important question is the appropriate choice of the sampling time T so that the approximation of the continous time system by the discrete time one in (4) is accurate enough. For this purpose a stright line motion of the load is designed by using the above flatness technique resulting in the open loop control signals $u_1(t)$ and $u_2(t)$ and the flatness based rope length and load position. The parameters of the small size 2D crane model were $\alpha = 0.4454\,\text{rad}$, $k = 0.1662\text{m}$, $l = 0.351\,\text{m}$, $J_1 = J_2 = 10^{-4}\,\text{kg/m}^2$ and $\rho_1 = \rho_2 = 0.025\,\text{m}$. It was experimentally proved that for $u_1(t)$, $u_2(t)$ and $T = 0.002\text{s}$ the discrete time approximation reproduces the results $L_1(t), L_2(t), L_3(t)$ determined by the flatness technique. It is clear that the small sampling time is the consequence of the small size model (1:80 reduction) and for real crane much larger sampling time is possible.

The optimization has been performed by using conjugate gradient technique. Satisfactory accuracy of the optimisation has been reached within reasonable time, see Fig. 3, however it is far away of the real time expectations in the case of the small size model.

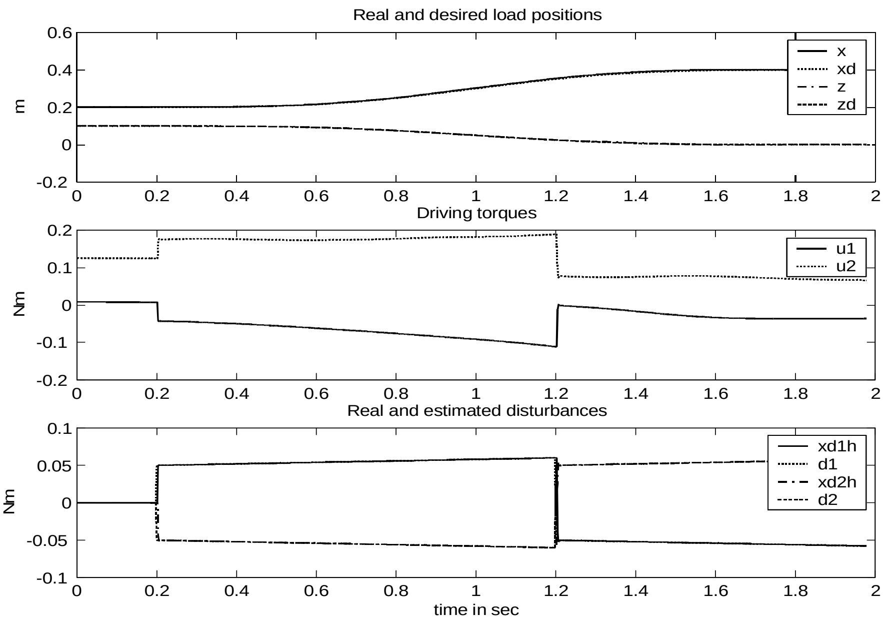

Fig. 3. Behaviour of the small size model of the 2D crane during predictive control.

4. Predictive Control of a Wheeled Mobile Robot

The third problem, which will be investigated in the paper, is the control of a wheeled mobile robot. It is known (Warren et al., 2001) that both position control and tracking can be performed by using nonlinear time variable dynamic state feedback. Nonlinear predictive control is an alternative method for both type of problems.

The control is divided into 3 levels. High level control designs the motion of the reference mobile robot in the form $\dot{x}_r = v_r \cos\vartheta_r$, $\dot{y}_r = v_r \sin\vartheta_r$ and $\dot{\vartheta}_r = \omega_r$. Low level control realizes the control of the mobile robot according to the control outputs of the middle level based on the kinematic model. Only the high and middle levels have been investigated.

State, output and input are $x := (x, y, \vartheta)^T$, $y := x$ and $u := (v, \omega)^T$, respectively. The system model is $f_c(x,u) = (u_1 \cos x_3, u_1 \sin x_3, u_2)^T$.

The derivatives $\partial f_c / \partial x_i, \partial f_c / \partial u_i$ needed for NPC can easily be determined. $T = 0.02\,s$ is suitable for discrete time approximation of the continuous time system if the speed is in the order of $0.5\,m/s$. Horizon length $N = 10$ has been chosen. The reference robot's $u := (v_r, \omega_r)^T$ is used in NPC as initial approximation of the control sequence in the first horizon. Constant plus linear disturbances reduced on the mobile robot inputs have been assumed and estimated by using extended Kalman filter. Optimization has been performed by using conjugate gradient technique. Satisfactory accuracy of the optimization has been reached within reasonable time fulfilling real time expectations, see Fig. 4.

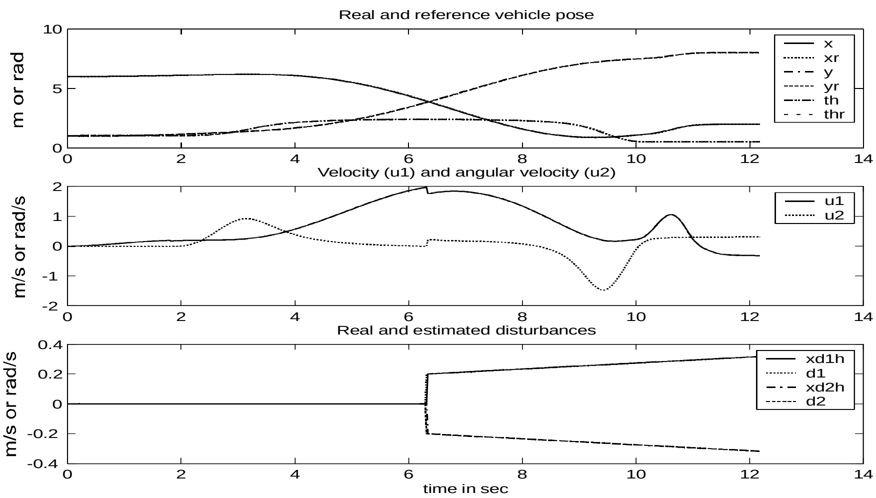

Fig. 4. Behaviour of the mobile robot during predictive control.

6. Friction Modeling

Friction phenomena can not be neglected in high precision mechatronic systems which are actuated by rotating axes. Typical friction models contain Coulomb and Striebeck effects. Based on these effects the friction force can linearly be parametrized:

$$h_{Fi}(\dot{q}_i) = \mu(\dot{q}_i)\theta_{FPi}^T \xi_{FPi}(\dot{q}_i) + (1-\mu(\dot{q}_i))\theta_{FNi}^T \xi_{FNi}(\dot{q}_i) \qquad (17)$$

where $\mu(\dot{q}_i) = 1$ if $\dot{q}_i \geq 0$ and 0 otherwise. The vectorrs θ_{Fi} contain the unknown parameters of the friction model and $\xi_{Fi}(\dot{q}_i)$ are the regressor vectors. This relation means that the friction can be be described by two separately linearly parametrized models, one for the positive (P) velocity regime and another for the negative (N) one. The switching between them occures at zero velocity. The switching function μ can be shifted into the regressor functions hence the friction model can be written in the compact form

$$\begin{aligned} &\theta_{Fi} = (\theta_{NFi}^T \;\; \theta_{PFi}^T)^T, \\ &\xi_{Fi}(\dot{q}_i) = ((1-\mu(\dot{q}_i))\xi_{FNi}^T(\dot{q}_i) \;\; \mu(\dot{q}_i)\xi_{FPi}^T(\dot{q}_i))^T, \\ &h_{Fi}(\dot{q}_i) = \theta_{Fi}^T \xi_{Fi}(\dot{q}_i). \end{aligned} \qquad (18)$$

Marton and Lantos (2004) developed a sliding mode robot control algorithm for payload and friction estimation where the friction was modeled by $h_{Fi}(\dot{q}_i)$.

Having identified the friction model in the above form in separate experiments, the friction model can be implemented in nonlinear predictive control algorithms either immediately by using the original form $h_{Fi}(\dot{q}_i)$ or by modeling the function $h_{Fi}(\dot{q}_i)$ using neuro-fuzzy techniques. A robust fuzzy modeling can be based on singular value decomposition, see Yam (1997). It is also possible to immediately use Sugeno fuzzy systems for modeling the friction effects where the antecedent parts of the relations define the regressor and the consequence parts contain the parameters.

Conclusions

Finite horizon nonlinear predictive control algorithms have been developed and tested for a 2-DOF robot arm, a 2D crane and a weeled mobile robot. For further developments a multiprocessor system can be suggested which consists of three computers. On the first computer run the NPC control and the extended Kalman filter under QNX real-time operating system. Another process manages the receipt of the sensory information and the sending of the controller outputs to the system. The physical system (robot arm, crane or mobile robot) can be simulated as separate QNX process on the first computer. This process communicates with the other parts by using messages. The simulated system can easily be removed in future experiments with real systems saving the whole controller architecture. On the second computer (running also under QNX) can be performed the off-line design of the reference signals for the robot arm, the flatness based path and driving torque design for the crane and the design of the path of the reference mobile robot, respectively. The third computer runs under Windows2000 and documents the real control, state and output information in graphical form.

The results in the three examples show that nonlinear predictive control can be used not only in the process control but also in the control of moving systems including robots, cranes and vehicles. Nonlinear predictive control is able to improve the control performances and can take into consideration the system constraints. The time needed for the convergence of the online numerical optimization can be guaranteed for $T = 25\,\mathrm{ms}$ sampling time and $N = 10T$ horizon length in case of standard processors and QNX real time operating system. If needed, further fine interpolation can be applied. These give real chance also for applications in more complicated fields like robot cooperation, formation motion of aircraft systems and cooperation between underground and aerial vehicles.

References

[1] Camacho, E. F. and C. Bordons (2000). *Model predictive control*. Springer, London.

[2] Rossiter, J.A. et al. (1998). A numerically robust state-space approach to stable-predictive control strategies. *Automatica*, **34**, 65-73.

[3] Lantos, B. (2003). *Control systems theory and design. Vol. II. Advanced control systems* (in Hungarian). Hungarian Academic Press, Budapest.

[4] Allgöwer,F. and A. Zheng ed. (2000). *Nonlinear predictive control.* Birkhäuser, Basel.

[5] Kim, H.J. and D. H. Shim (2003). A flight control system for aerial robots. *Control Engineering Practice*, **11**, 1389-1400.

[6] Yu, J. et al. (2001). Comparison of nonlinear control design techniques on a model of the Caltech ducted fan. *Automatica*, **37**, 1971-1978.

[7] Chui, C.K. and G. Chen (1999). *Kalman filtering with real-time applications.* Springer, Berlin.

[8] Kiss, B., J. Levine and Ph. Mullhaupt (2000). *Control of a reduced size model of US Navy crane*. In Isidori: Nonlinear control of the year 2000, Springer, London.

[9] Kiss, B. (2001). Motion planning and control of a class of flat and Liouvillian mechanical systems. PhD Thesis, L'Ecole Nationale Superieure des Mines de Paris.

[10] Warren, E.D. et al. (2001). *Nonlinear control of wheeled mobile robots*. Springer, London.

[11] Márton, L., Lantos, B. (2004). Sliding mode robot control with friction and payload estimation. *Journal of Advanved Computational Intelligence and Intelligent Informatics*, **8**, 553-561.

[12] Yam, Y. (1997). Fuzzy approximation via grid point sampling and singular value decomposition. *Systems, Man and Cybernetics*, **27**, 933-951.