Proceedings of the
46th IEEE Conference on Decision and Control
New Orleans, LA, USA, Dec. 12-14, 2007

Hybrid Model Predictive Control Based on Wireless Sensor Feedback: An Experimental Study

Alberto Bemporad, Stefano Di Cairano, Erik Henriksson and Karl Henrik Johansson

Abstract— **This paper presents the design and the experimental validation of model predictive control (MPC) of a hybrid dynamical process based on measurements collected by a wireless sensor network. The proposed setup is the prototype of an industrial application in which a remote station controls the process via wireless network links. The experimental platform is a laboratory process consisting of four infrared lamps, controlled in pairs by two on/off switches, and of a transport belt, where moving parts equipped with wireless sensors are heated by the lamps. By approximating the stationary heat spatial distribution as a piecewise affine function of the position along the belt, the resulting plant model is a hybrid dynamical system. The control architecture is based on the reference governor approach: the process is actuated by a local controller, while a hybrid MPC algorithm running on a remote base station sends optimal belt velocity set-points and lamp on/off commands over a network link exploiting the information received through the wireless network. A discrete-time hybrid model of the process is used for the hybrid MPC algorithm and for the state estimator.**

I. Introduction

The society of today is moving towards a wireless community where large numbers of mobile embedded systems interact with each other. Common examples are handheld communication devices exchanging information over cellular or wireless local area networks, and wireless appliances interacting in smart houses.

Wireless networks are also involved in a growing number of applications in industrial automation. There are several advantages by introducing a wireless medium between a process and a controller, such as reduced cost due to less wiring at system commissioning and no need for re-wiring at system upgrade. Another advantage is in increased flexibility in positioning sensors and actuators and in the possibility of moving them during operation. In general, the network technology enables intelligence to be embedded in more devices and information to be easily exchanged between them; hence, wireless technology makes it possible with more dynamic and adjustable control architectures.

This work was supported by the European Commission under the HYCON Network of Excellence, contract number FP6-IST-511368, and by the Italian Ministry for University and Research (MIUR) under project "Advanced control methodologies for hybrid dynamical systems" (PRIN'2005). The work by E. Henriksson and K. H. Johansson was also supported by the European project SOCRADES, the Swedish Research Council, and the Swedish Strategic Research Foundation through an Individual Grant for the Advancement of Research Leaders.

A. Bemporad and S. Di Cairano are with Department of Information Engineering, Faculty of Engineering, University of Siena, 53100 Siena, Italy bemporad,dicairano@dii.unisi.it

E. Henriksson and K. H. Johansson are with the Automatic Control Lab, School of Electrical Engineering, Royal Institute of Technology, 10044 Stockholm, Sweden erike02,kallej@ee.kth.se

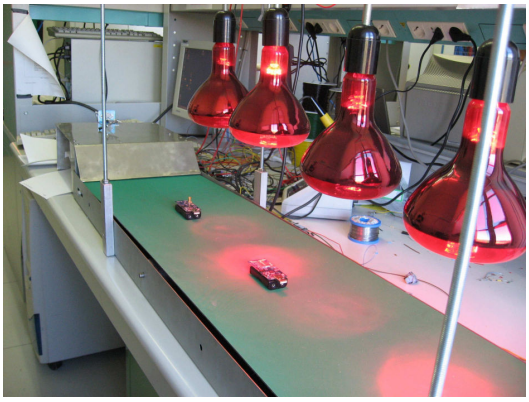


Fig. 1. Experiment at the Automatic Control Lab, University of Siena.

The usage of wireless technology within feedback control loops raises, however, new challenges. The network medium introduces uncertainties on packets loss and transmission delay, among the others. The impact of these uncertainties, and the uncertainties variation depend on the network technology; certain networks guarantee the delivery of a message, but with high delay variability, while others provide a less reliable communication, but ensure better delay characteristics. For wireless networks, the packet loss typically varies heavily with the radio conditions, so that if the environment is changing or the nodes are mobile, the control system needs to handle many different network conditions.

Control over wireless networks is a young research area without mature theory or tools, but with a lot of current activity. A general view on the need for interaction between control and communication in the design of wireless networks was recently introduced [1]. Open research problems in the area of control using wireless sensors networks include choice of architectures and modular design and implementation [2], [3]. A cross-layer framework for the joint design of wireless networks and distributed controllers is attempting [4], although care needs to be taken to avoid undesirable interactions [5].

This paper advocates the use of model predictive control (MPC) [6] as a tool to tackle control problems in such uncertain environments arising from wireless sensor and actuator loops. MPC is widely spread in industry for control of complex multivariable processes [7]. Given a model of the process dynamics, constraint specifications on system variables (input saturations, state bounds, etc.), and desired performance specifications, at each time step the MPC control algorithm solves an optimal control problem, which

1-4244-1498-9/07/$25.00 ©2007 IEEE.

depends on the current state as initial condition and on the current reference signals, over a future prediction horizon. The result of the optimization is a sequence of future control moves. Only the first element of the sequence is applied to the process, the remaining moves are discarded, and the optimization is repeated at the next time step.

MPC based on *hybrid* dynamical models [8] has emerged as a very promising approach to handle switching linear dynamics, on/off inputs, logic states, as well as logic constraints on input and state variables. The associated finite-horizon optimal control problem can be formulated as a mixed-integer program (MIP) for which efficient solvers are available. Extensions of the hybrid MPC formulation introduced above have been recently proposed for stochastic hybrid systems [9] that appears to be suitable for application within a hybrid networked control architecture.

To handle the unreliability of the communication links between the base station (where the MPC computations are performed) and the process, we use the reference governor approach [10], [11]. Here, the process stability is granted by a local controller at the plant. The local controller receives its reference from the remotely executed MPC algorithm, which aims at obtaining the desired performance. Thus, the computational power required to solve the optimization problem is moved away from the plant. The reference governor approach was first studied in the context of unreliable network links in [12], where the command sequences computed by the predictive controller are used to possibly overcome packet loss and large (possibly unbounded) delays, together with a synchronization algorithm.

The main contribution of this paper is to present the design and the experimental validation of hybrid MPC over wireless links on a new laboratory process built for this purpose at the University of Siena, see Figure 1. The process consists of a transport belt where moving parts equipped with wireless sensors are heated by four infrared lamps. The latter are commanded in pairs by two on/off switches. The process is actuated by a local controller, while a hybrid MPC algorithm running on a remote base station sends optimal belt velocity set-points and lamp on/off commands. The remote controller receives information from the sensors through a wireless network formed by Telos motes [13]. A discrete-time hybrid model of the process is used for the design of the hybrid MPC algorithm and a state estimator.

The outline of the paper is as follows. Section II presents the laboratory process, the communication network, and the corresponding mathematical models. Section III deals with the hybrid control system design, and Section IV with the control architecture implementation. Simulations and experimental results are reported in Section V, and conclusions and a discussion on future research directions are given in Section VI.

II. Process Description, Modeling, and Architecture

This section describes the laboratory process, derives a hybrid dynamical model, and discusses how the communication network is modeled in the feedback control system.

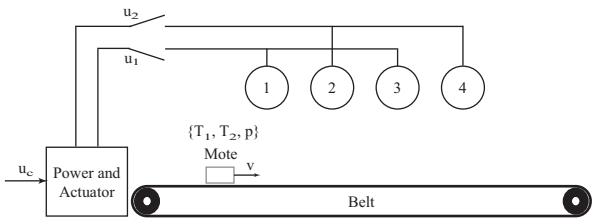


Fig. 2. Schematics of the process.

A. Physical Plant

The physical plant is shown in Figure 1. The main components of the plant, whose schematics are shown in Figure 2, are a belt and four infrared lamps. The belt is actuated by an electric servo motor. The heating lamps are placed in a row over the belt, and two on/off switches are available to actuate them. The first switch controls lamps 1 and 3, the second switch, lamps 2 and 4. The lamps are grouped to reduce the complexity of the model and of the control algorithm.

To derive a dynamic model of the process we define as states the sensor casing temperature $T_1 \in \mathbb{R}$, the sensor temperature $T_2 \in \mathbb{R}$, and the position $p \in \mathbb{R}$ of the part that moves along the belt. The system evolution is governed by the differential equations

$$\dot{T}_1 = -\alpha\big(T_1 - T_{\rm ss}(p, u_1, u_2)\big), \tag{1a}$$

$$\dot{T}_2 = -\beta\big(T_2 - T_1\big), \tag{1b}$$

$$\dot{p} = \gamma(u_c), \tag{1c}$$

where $u_c \in \mathbb{R}$ and $u_1, u_2 \in \{0, 1\}$ are control inputs, and $T_{\rm ss} : \mathbb{R}^3 \to \mathbb{R}$ is a static nonlinearity. The parameters $\alpha, \beta > 0$ are physical constants. The continuous signal $\gamma(u_c)$ corresponds to the part velocity, which is obtained through a static nonlinear mapping $\gamma(\cdot)$ of the control command. As regards the discrete input signals, $u_1 = 0$ when lamps 1 and 3 are off, $u_1 = 1$ when they are on, and similarly for u_2 relatively to lamps 2 and 4. The steady-state temperature of the sensor casing at position p, when the lamps switches are (u_1, u_2), is $T_{\rm ss}(p, u_1, u_2)$,

$$T_{\rm ss}(p, u_1, u_2) = f_1(p)u_1 + f_2(p)u_2 + T_{\rm amb}, \tag{2}$$

where $T_{\rm amb} \in \mathbb{R}$ is the ambient temperature, $f_i(p) : \mathbb{R} \to \mathbb{R}$, $i \in \{1, 2\}$ describes the increase in steady-state temperature at position p obtained by turning on the i-th switch.

B. Hybrid Model

We want to approximate the continuous-time model (1) and the nonlinearity $T_{\rm ss}$ in (2) by a hybrid model.

First, we consider a piecewise affine approximation χ of $(T_{\rm ss} - T_{\rm amb})$: we partition $\mathbb{R}$ into ℓ intervals $\{I_1, I_2, \ldots, I_\ell\}$, and approximate f_i, $i = 1, 2$, with the piecewise affine

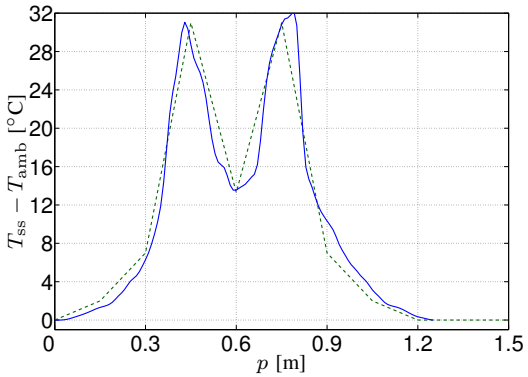


Fig. 3. $T_{\rm ss} - T_{\rm amb}$ and its piecewise affine approximation.

functions

$$\chi_i(p) = \begin{cases} K_j^i p + h_j^i & \text{if } u_i = 1, p \in I_j, j = 1, \dots \ell \\ 0 & \text{otherwise,} \end{cases} \quad i = 1, 2, \tag{3a}$$

$$\chi(p) = \chi_1(p) + \chi_2(p). \tag{3b}$$

The effect of $T_{\rm amb}$ will be introduced later as a measured disturbance. The nonlinear function and its piecewise affine approximation are shown in Figure 3.

The continuous-time model of the physical plant is sampled with sampling period $T_s = 250$ ms using a zero-order hold. We obtain the following discrete-time system

$$x(t+1) = \underbrace{\begin{pmatrix} a_{11} & 0 & 0 \\ a_{21} & a_{22} & 0 \\ 0 & 0 & 1 \end{pmatrix}}_{\Phi} x(t) + \underbrace{\begin{pmatrix} b_{11} & 0 \\ b_{21} & 0 \\ 0 & b_{32} \end{pmatrix}}_{\Gamma} \begin{pmatrix} \chi(t) \\ v_c(t) \end{pmatrix}, \tag{4a}$$

$$y(t) = \underbrace{\begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}}_{C} x(t), \tag{4b}$$

where $x = (T_1, T_2, p)^T$, and the belt velocity $v_c = \gamma(u_c)$ is used as system input. The motor command can be recovered by the inversion $u_c(t) = \gamma^{-1}(v_c(t))$.

In order to apply hybrid model predictive control, the system is formulated as a mixed logical dynamical (MLD) system [8]

$$x(t+1) = Ax(t) + B_1 u(t) + B_2 \delta(t) + B_3 z(t), \tag{5a}$$

$$y(t) = Cx(t), \tag{5b}$$

$$E_2 \delta(t) + E_3 z(k) \leqslant E_1 u(k) + E_4 x(k) + E_5, \tag{5c}$$

where $u = (v_c, u_1, u_2)^T$ is the input vector, and $z(t) \in \mathbb{R}^{22}$ and $\delta(t) \in \{0, 1\}^{10}$ are the continuous and binary auxiliary variables, respectively. The auxiliary variables describe the piecewise affine dynamics (3).

C. Communication Network

We introduce a simple model for the communication links between the remote controller and the plant, see Figure 4. Let u and y denote the plant input and output, respectively, and $\hat{u}$ and $\hat{y}$ denote the signals after transmission. We have that $\hat{u}(t) = u(t)$ and $\hat{y}(t) = y(t)$ if the corresponding packets at time t are received. Hence, we suppose that the communication delays are negligible compared to the plant dynamics. Moreover, we suppose that the receivers can detect if a packet is lost, and that the detection is instantaneous. We denote a lost packet by ϵ, so that if the command packet was lost at time t, $\hat{u}(t) = \epsilon$.

III. Control System Design

The overall control strategy uses a hybrid MPC algorithm as a reference governor, as described in [10], [11]. We use the Hybrid MPC implemented in the Hybrid Toolbox for MATLAB [14], where the optimization problem is solved on-line in real time using the mixed-integer quadratic programming solver CPLEX [15].

A. Wireless Control Architecture

The control architecture we consider is a cascade controller. The inner controller is placed at the plant location, hence it is referred as *local controller*. The outer controller is placed remotely with respect to the plant site, hence it is referred as *remote controller*. The remote controller exploits network links to exchange information with the plant and the local controller. While the terms "local" and "remote" characterize the controllers basing on their locations, the terms "inner" and "outer" characterize them from a hierarchical point of view. The inner (local) controller receives high level commands (setpoints) from the outer (remote) controller, and regulates the process accordingly, ensuring stability and tracking. The outer controller generates the setpoints based on the simplified system model which results from the plant in closed-loop with the local controller. A simplified model makes easier the long term planning, which is computationally intensive, while stability and tracking need to be enforced by a local controller to avoid delays and data losses. From an implementation point of view the local controller must be fast, but does not need large computational effort. A simple linear controller can in general accomplish these tasks. On the other hand the remote controller can run slower, but needs larger computational effort to generate optimal plans. We implement the remote controller by hybrid MPC.

The purpose of the network medium is to delocalize the outer controller from the plant site. In this way it can be placed remotely, in a powerful computer, which could not be located in the proximity of the plant. Two network channels are used to transmit the process measurements from the plant to the remote controller and to transmit the setpoints from the remote controller to the local one, respectively. The physical channels can be the same, hence the behavior of the packets sent in one direction is related to the behavior of the packets sent in the opposite direction, or different, hence the communications in different directions do not interfere with each other. The communications can also run different protocol enabling a large set of possible design choices.

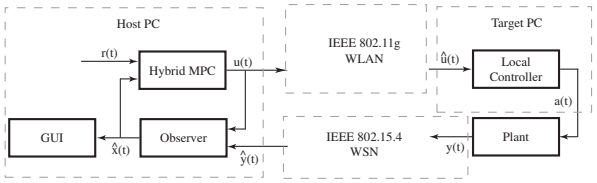


Fig. 4. Wireless control architecture.

The controller architecture implementation is described in Section IV.

B. The Hybrid MPC algorithm

The hybrid MPC algorithm [8] is based on the solution at each time step t of the optimization problem

$$\min_{\{u(k),\delta(k|t),z(k|t)\}_{k=0}^{N-1}} J(\{u(k),\delta(k|t),z(k|t)\}_0^{N-1},x(t)) \triangleq$$

$$q(Q_\rho,\rho)+q(Q_{xN},x(N|t)-x_r)+\sum_{k=1}^{N-1} q(Q_x,x(k)-x_r)+$$

$$+\sum_{k=0}^{N-1}\Big(q(Q_u,u(k)-u_r)+q(Q_z,z(k|t)-z_r)+$$

$$+q(Q_y,y(k|t)-y_r)\Big) \tag{6a}$$

s.t.

$$\begin{cases} x(0|t) &= x(t)\\ x(k+1|t) &= Ax(k|t)+B_1u(k)+B_2\delta(k|t)+B_3z(k|t)\\ y(k|t) &= Cx(k|t)+D_1u(k)+D_2\delta(k|t)+D_3z(k|t)\\ E_2\delta(k|t) &+ E_3z(k|t)\leqslant E_1u(k)+E_4x(k|t)+E_5\\ u_{\min}-\mathbf{1_u}\rho &\leqslant u(k)\leqslant u_{\max}+\mathbf{1_u}\rho,\quad k=0,1,...,N-1\\ x_{\min}-\mathbf{1_x}\rho &\leqslant x(k|t)\leqslant x_{\max}+\mathbf{1_x}\rho,\quad k=1,...,N\\ y_{\min}-\mathbf{1_y}\rho &\leqslant y(k|t)\leqslant y_{\max}+\mathbf{1_y}\rho,\quad k=0,...,N-1 \end{cases} \tag{6b}$$

where N is the prediction horizon, $\{x(k|t)\}_{k=1}^{N}$ is the sequence of predicted states at time $t+k$, obtained when the input sequence $\{u(k)\}_{k=0}^{N-1}$ is applied at time instants $t+k$, $k=0,\ldots,N-1$ from initial state $x(0|t)=x(t)$; $\{y(k|t)\}_{k=0}^{N-1}$ is the corresponding output sequence. The matrices Q_x, Q_u, Q_y, Q_z, Q_N, are weights on state, input, output, auxiliary variable and terminal state vectors, respectively, and $q(Q,x)=x^TQx$. The vectors x_r, y_r, u_r, z_r, are given references on state, output, input and auxiliary variable vectors, respectively, while $u_{\min},u_{\max},x_{\min},x_{\max},y_{\min},y_{\max}$ are lower and upper bounds on inputs, states and outputs, respectively. The variable ρ is a scalar non-negative slack variable and $\mathbf{1_u},\mathbf{1_x},\mathbf{1_y},\mathbf{1_h}$ are vectors of proper dimensions composed of ones. The use of ρ allows violation of the constraints by "softening" them, thus guaranteeing feasibility of the optimization problem. The violation of the constraints is penalized by the weight matrix Q_ρ, that is usually at least two orders of magnitude larger than the other weights.

The reference choice and the weight function tuning are performed by the control objectives. The acceleration and the velocity of the belt should be low, in order to reduce power consumption and to avoid wild dynamics, that can cause excessive wear. We want to track a position reference and a temperature reference and we also want the state to remain in a predefined "safe" set, that excludes high and low temperatures and excessive velocities. In order to enforce these, we apply references on the output variables, on the continuous input variables, and we impose constraints on states. The continuous input reference is set to 0, favoring light actuation of the belt. The output reference profile y_r defines the desired behavior of the system. The length of the horizon N affects the performance of the controller. A longer horizon gives a smoother behavior, a shorter one gives a more aggressive controller. Furthermore, the longer the horizon the more complex the optimization problem, hence the prediction horizon N is chosen by trading off between the performance and the available computational power.

The original sampled model needs to be extended by two additional states. The first one is the ambient temperature T_{amb} in (2). Such a state remains constant in the prediction model and represents a measured disturbance. The second additional state is the "input memory" state x_u, which is used to constrain the acceleration of the belt, not explicitly modeled in (4). The dynamics of x_u are defined by

$$x_u(t+1)=v_c(t). \tag{7}$$

The acceleration at time t for a given input $v_c(t)$ can be computed by backward Euler approximation from $x_u(t)$ and $v_c(t)$ as $\big(v_c(t)-x_u(t)\big)/T_s$, so that constraints on the acceleration can be expressed as state constraints. The system model becomes

$$x(t+1)=\begin{pmatrix} a_{11} & 0 & 0 & 1 & 0\\ a_{21} & a_{22} & 0 & 1 & 0\\ 0 & 0 & 1 & 0 & 0\\ 0 & 0 & 0 & 1 & 0\\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} x(t)+\begin{pmatrix} b_{11}\chi(t)\\ b_{11}\chi(t)\\ b_{32}v_c(t)\\ 0\\ v_c(t) \end{pmatrix}, \tag{8a}$$

$$y(t)=\begin{pmatrix} 0 & 1 & 0 & 0 & 0\\ 0 & 0 & 1 & 0 & 0 \end{pmatrix} x(t), \tag{8b}$$

where $x=(T_1,T_2,p,T_{\text{amb}},x_u)^T$.

The objective function is

$$J(\{u(k),\delta(k|t),z(k|t)\}_0^{N-1},x(t)) \triangleq$$

$$\sum_{k=0}^{N-1}\Big(q_{v_c}v_c(k)^2+q_z\left(\frac{1}{T_s}\right)^2\big(v_c(k)-x_u(k|t)\big)^2+$$

$$+\|Q_y(y(k|t)-y_r)\|_2\Big)+q_\rho\rho^2 \tag{9a}$$

$$N=4, q_\rho=10^3,\ q_{v_c}=2,\ q_z=1,$$

$$Q_y=\begin{pmatrix} 0.01 & 0\\ 0 & 0.6 \end{pmatrix},$$

and the state and input constraints are

$$\begin{pmatrix} 15\\ 15\\ 0 \end{pmatrix}\leqslant\begin{pmatrix} T_1(k|t)\\ T_2(k|t)\\ p(k|t) \end{pmatrix}\leqslant\begin{pmatrix} 80\\ 80\\ 1.2 \end{pmatrix}, \tag{9b}$$

$$-0.1\leqslant v_c(k|t)\leqslant 0.1, \tag{9c}$$

$$u_1(k|t),\ u_2(k|t)\in\{0,1\}. \tag{9d}$$

C. Observer

In the considered process, not all of the states are measurable. Thus, we design an observer to estimate the unmeasured states. The simplest applicable observer is the reduced order nonlinear Luenberger observer

$$\begin{aligned}\hat{x}(t+1|t+1) =& \Phi\hat{x}(t|t) + \xi(t) \\ &+ K\big[\hat{y}(t+1) - C\big(\Phi\hat{x}(t \mid t) + \xi(t)\big)\big], \qquad (10a)\end{aligned}$$

$$\xi(t) = \begin{pmatrix} b_{11}\chi(t|t) \\ b_{21}\chi(t|t) \\ T_s v_c(t) \end{pmatrix}, \; K = \begin{pmatrix} k_{11} & k_{12} \\ 1 & 0 \\ 0 & 1 \end{pmatrix}, \qquad (10b)$$

where $T_s = 250$ ms is the sampling period, and $k_{11} = 5$, $k_{22} = 0$.

As referred in Section II-C, the wireless network links introduce packet losses. In fact, the observer uses the received signal $\hat{y}$, instead of the locally measured signal y. When packets containing measurements are lost a simple estimation method is to let the estimation evolve in open loop [16], which means that if no measurements are received, the system is assumed to evolve according to the hybrid prediction model. Note that in our control problem, the packets can be lost also in the link that connects the controller to the plant, even if, being a TCP/IP wireless link, this is supposed to be more reliable. When packets containing commands are lost, the local controller keeps the previous reference, which is considered to be the best guess of what would have been received, and it is safe. This induces additional estimation errors, which are currently under investigation.

IV. Control Architecture Implementation

The control architecture can be divided into five independent blocks, according to Figure 4. The entire control loop runs with 4 Hz sampling frequency.

The controller infrastructure is implemented through Mathworks xPC-Target [17]. A remote computer, called *Host* runs the MPC algorithm and solves problem (6). The computed optimal action is sent via a WLAN 802.11g wireless link to the *Target* computer, which is physically connected to the belt motor and to the lamps switches by a DAQ board. The *Target*, together with a servo controller that reduces the nonlinearities of the belt motor, works as a local controller and actuates the commands, providing the software/hardware interface. The actions taken to overcome packet losses and to estimate unmeasurable system states (the position of the part) also take places at the *Host*.

The loop is closed by the wireless sensor network. We use the Berkeley motes *Tmote Sky* from Moteiv corporation [13] for simulating the parts to be processed on the belt. These motes are equipped with temperature, humidity, and light sensors, a low-power 8 MHz 16 bit microprocessor, and 2.4 GHz IEEE 802.15.4 radio communication capabilities. The on-board temperature sensor is used for feedback, via a wireless sensor network (WSN) which is composed by a varying number of motes. The sensor on the belt takes temperature measurements and transmits them, possibly crossing other motes that act as intermediate nodes, to a mote acting as base station. The base station mote is connected to the *Host* computer via a USB port, hence providing the required measurements. The information obtained from the temperature sensor is used to update the system state, for the next MPC iteration.

The Host runs on a 1.2 GHz Pentium-M laptop, equipped with 632 MB RAM, MATLAB 7.1, the Hybrid Toolbox v1.1.0, and CPLEX 9.0. The Target is implemented on a Pentium 133 MHz, running the Mathworks xPC-Target the real time kernel, programmable using SIMULINK. A National Instruments PCI-6024E DAQ-board interfaces the *Target* with the process.

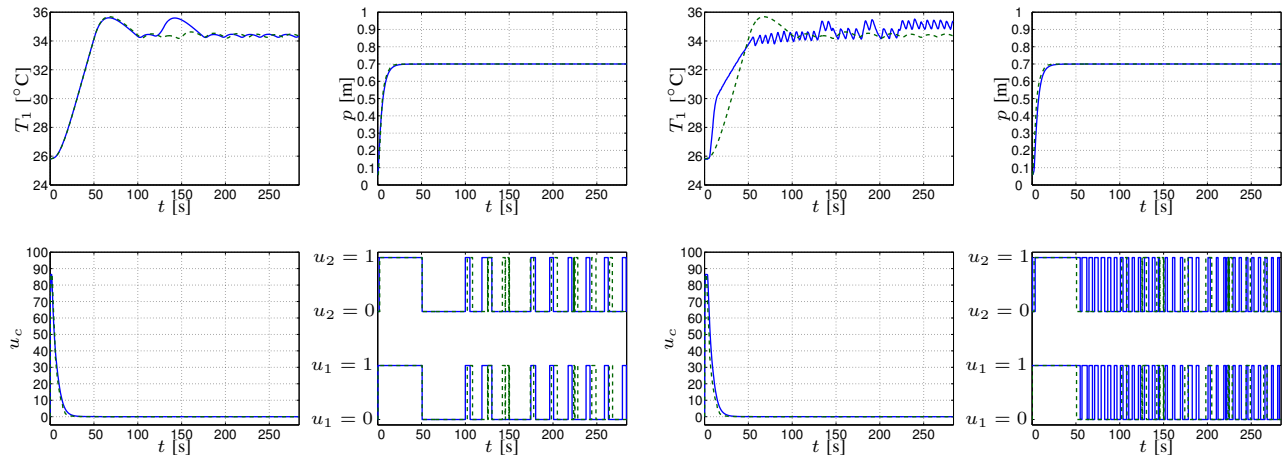
The Hybrid MPC algorithm is implemented within the Hybrid Toolbox for MATLAB [14]. System model (8) is written in HYSDEL [18] and automatically converted by the associated compiler into the MLD system (5). The optimal control problem (6) is formulated using the Hybrid Toolbox [14], and included into a SIMULINK model, as an S-function. The resulting optimization problem consists of 141 optimization variables, 93 continuous and 48 binary, respectively, and 585 mixed-integer linear inequalities. The average time required to solve the optimization problem using CPLEX is 17ms, with a worst-case computation time of around 125ms, which further motivates the choice of the sampling time.

After the control action has been computed, it is sent to the *Target* computer via the wireless TCP/IP link. The local controller implemented on the *Target* provides a simple interface towards the DAQ and the process. The operations performed by the *Target* include computing the motor command $\hat{u}_c(t)$ from the commanded belt velocity received through the network $\hat{v}_c(t)$ by performing the inversion $\hat{u}_c(t) = \gamma^{-1}\big(\hat{v}_c(t)\big)$, and turning on and off the lamps. Another important function of the local controller is to hold the last received input. This functionality handles the recovery action for the packet losses that occur in the link from the *Host* to the *Target*.

V. Experimental Results

The hybrid MPC controller (9) is first tuned in simulation, also taking into account the packet loss model of Section II-C. We set $y_r = (35, 0.7)^T$, giving temperature and position references. First, the nominal step response (no packet loss, perfect modeling) is simulated. The results show a small steady state error, due to the input quantization and to the absence of integral action. Then, a lossy feedback channel is simulated, with packet loss profile obtained from a real sensor network. The simulation results are shown in Figure 5(a), where the dashed lines indicate the nominal response and the solid line indicate the response of the model with packet losses simulated from real data. The deviation from the nominal behavior that occurs around $t = 150$ s is due to a massive packet drop burst. During such an interval, the lamps are commanded off, but the temperature keeps increasing. Hence, the packets containing the current commands are being dropped and an older command is being applied.

The experimental results are shown in Figure 5(b). Other than by the packet losses, the errors are now introduced



(a) Simulations. Nominal behavior (dashed line) and behavior with feedback packet losses simulated from real packet loss profile (solid line).

(b) Experimental behavior (solid line) and nominal simulated behavior (dashed line).

Fig. 5. Simulations and experiments on the process shown in Figure 2.

by external noise and modeling imperfections. In particular, due to the piecewise affine approximation of (2) the input behavior is more aggressive. However, the experimental results (solid lines) are still close to the simulation of the nominal model (reported as dashed lines for comparison).

VI. Conclusions and Future Research

This paper has presented a hybrid MPC design and an experimental demonstration of remote control over wireless networks. Data packets dropped in both forward and feedback communication links can be handled with very good results using standard techniques. The hybrid MPC design has several advantages in comparison with traditional controllers. The most obvious advantage is that it offers the possibility to handle process nonlinearities and on/off inputs, and to enforce constraints on states, inputs and outputs in a simple and direct way. The setup has been also proven easy to tune. The only drawback is that hybrid MPC can be computationally intense, although it is fast enough for the application at hand, and the worst-case computation period can be bounded a priori by imposing time constraints on the optimization solver.

The authors are currently working on an implementation of a stochastic hybrid MPC controller [9] where the statistical properties of the communication channel will be estimated and used to adaptively constrain the working set of the controller to ensure robustness.

Acknowledgments

The authors gratefully acknowledge Francesco Molendi for his assistance in programming the motes, Davide Barcelli and Luca Lucherini for building most of the laboratory experiment, and Pan Gun Park for providing the wireless sensor network the packet drop sequences used in simulation.

References

[1] P. R. Kumar, "New technological vistas for systems and control: The example of wireless networks," *IEEE Control Systems Magazine*, vol. 21, pp. 24–37, 2001.

[2] B. Sinopoli, C. Sharp, L. Schenato, S. Schaffert, and S.S.Sastry, "Distributed control applications within sensor networks," *Proceedings of the IEEE*, vol. 91, pp. 1235–1246, Aug 2003.

[3] K.-E. Årzén, A. Bicchi, G. Dini, S. Hailes, K. H. Johansson, J. Lygeros, and A. Tzes, "A component-based approach to the design of networked control systems," *European Journal of Control*, 2007.

[4] X. Liu and A. Goldsmith, "Wireless network design for distributed control," in *Proc. 43th IEEE Conf. on Decision and Control*, Paradise Island, Bahamas, 2004, pp. 2823–2829.

[5] V. Kawadia and P. R. Kumar, "A cautionary perspective on cross layer design," *IEEE Wireless Communication Magazine*, vol. 12, pp. 3–11, 2005.

[6] J. Rawlings, "Tutorial overview of model predictive control," *IEEE Control Systems Magazine*, pp. 38–52, June 2000.

[7] S. Qin and T. Badgwell, "A survey of industrial model predictive control technology," *Control Engineering Practice*, vol. 93, no. 316, pp. 733–764, 2003.

[8] A. Bemporad and M. Morari, "Control of systems integrating logic, dynamics, and constraints," *Automatica*, vol. 35, no. 3, pp. 407–427, 1999.

[9] A. Bemporad and S. D. Cairano, "Optimal control of discrete hybrid stochastic automata," in *Hybrid Systems: Computation and Control*, ser. Lecture Notes in Computer Science, M. Morari and L. Thiele, Eds. Springer-Verlag, 2005, no. 3414, pp. 151–167.

[10] A. Bemporad, A. Casavola, and E. Mosca, "Nonlinear control of constrained linear systems via predictive reference management," *IEEE Trans. Automatic Control*, vol. AC-42, no. 3, pp. 340–349, 1997.

[11] E. Gilbert and I. Kolmanovsky, "Fast reference governors for systems with state and control constraints and disturbance inputs," *Int. J. Robust Nonlinear Control*, vol. 9, no. 15, pp. 1117–1141, Dec. 1999.

[12] A. Bemporad, "Predictive control of teleoperated constrained systems with unbounded communication delays," in *Proc. 37th IEEE Conf. on Decision and Control*, Tampa, FL, 1998, pp. 2133–2138.

[13] Moteiv Corporation, *Tmote sky*, 2007, http://www.moteiv.com/products/tmotesky.php.

[14] A. Bemporad, *Hybrid Toolbox – User's Guide*, Dec. 2003, http://www.dii.unisi.it/hybrid/toolbox.

[15] ILOG, Inc., *CPLEX 9.0 User Manual*, Gentilly Cedex, France, 2004.

[16] J. P. Hespanha, P. Naghshtabrizi, and Y. Xu, "Networked control systems: Analysis and design," 2006, to appear in the *Proc. of IEEE*, Special Issue on Networked Control Systems.

[17] The Mathworks Inc., *xPC Target - For Use with Real-Time Workshop*, 2000, User's Guide – Version 1.1.

[18] F. Torrisi and A. Bemporad, "HYSDEL — A tool for generating computational hybrid models," *IEEE Trans. Contr. Systems Technology*, vol. 12, no. 2, pp. 235–249, Mar. 2004.