Predictive Control of Teleoperated Constrained Systems with Unbounded Communication Delays

Alberto Bemporad

Institut für Automatik
ETH - Swiss Federal Institute of Technology
ETL K 22.1, CH 8092 Zürich, Switzerland
tel.+41-1-632 6679 fax +41-1-632 1211
bemporad@aut.ee.ethz.ch

Abstract

In this paper we present a control technique which allows the teleoperation of systems subject to input/state constraints through transmission channels with unbounded time-delays, such as Internet TCP/IP connections. The main idea is based on the fact that predictive controllers provide, as a by-product, command sequences which can be executed as emergency maneuvers whenever the communication channel is broken by excessive time-delays. We show how this idea can be exploited by equipping the predictive controller with some additional control logic which enables the synchronization between plant, predictive controller, and human operator.

Key Words— Teleoperation, unbounded time-delays, predictive control, constraints, Internet, TCP/IP

1 Introduction

Teleoperation consists of controlling a plant situated at a remote location through a communication channel. This is usually affected by signal transmission delays and failures. Delays can be imposed by: limits on the speed of light during radio transmission, e.g. 0.4 s for vehicles in low earth orbit [1]; limits on the speed of sound during acoustic telemetry, for example 2 s on a 1700 m round-trip; network traffic, such as on TCP/IP (Transmission Control Protocol/Internet Protocol) or UDP/IP (User Datagram Protocol) connections on the Internet. While the first two are exactly predictable or even constant (for example in space applications), the third kind of delay cannot be bounded a priori. The presence of delays highly disturbs the human operator's intuition and can lead to instability of the overall teleoperation. A widely adopted solution consists of using computer-graphic predictors which, on the basis of the last available state measurement from the remote location and a reliable dynamic model of the plant, generate virtual representations for the human operator. Therefore, the predictor's action compensates the delay in the back channel. By even generating predictions which go further in the future, the delay on the forward channel can be canceled as well. This requires that the operator generates commands according to a "future" situation, commands which can be executed in time when they arrive at the remote location. This scheme assumes that time-delays can be upper-bounded. On the other hand, when time-delays can be arbitrarily large, a possible lack of commands can take place at the plant's location. In such a circumstance, it is desirable to have (open-loop) emergency sequences of commands which allow to avoid unwanted behaviours of the plant.

Predictive controllers [2, 3, 4, 5, 6] naturally provide this kind of sequences. In predictive control, the main idea is to use a *model* of the plant to *predict* the future evolution of the system and, accordingly, select the command input. For this reason, in the literature it is also referred to as *model predictive control* (MPC). Prediction is handled according to the so called *receding horizon* philosophy: a 'virtual" sequence of future control actions is chosen by predicting the future evolution of the system and applied to the plant until new measurements are available. Then, a new sequence is evaluated so as to replace the previous one. Each selected sequence is the result of an optimization procedure which takes into account two objectives: (i) maximize the tracking performance, and (ii) guarantee that constraints on input and state variables are—and *will be*—fulfilled, i.e., no "blind-alley" is entered.

Therefore, during normal predictive control operation, typically only the first sample(s) of the optimal sequences is applied, and the remaining samples are simply "thrown away". However, when the loop is broken by a forward time-delay, the new sequence cannot arrive to the remote plant. In this case, the subsequent

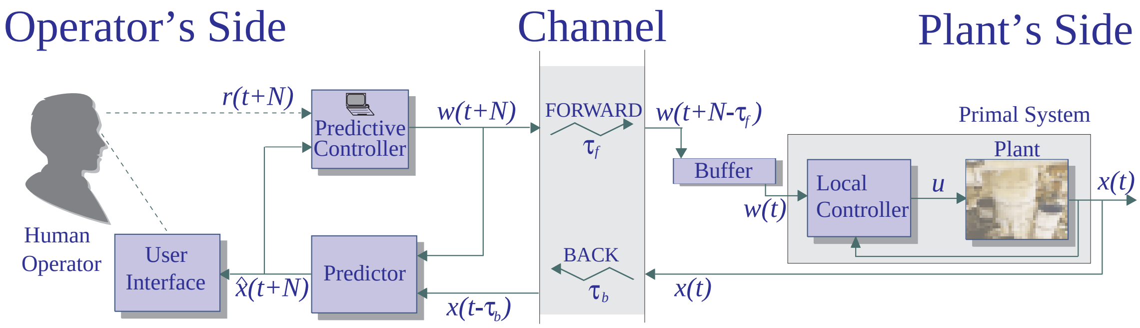


Figure 1: Teleoperation through a channel with unbounded communication delays

terms of the last available sequence can be then recovered from the "trash can", and *safely* applied to the plant.

With this idea in mind, in this paper we show how a given predictive controller can be modifed, by adding some logic, in order to perform teleoperation of systems subject to input/state constraints through transmission channels with unbounded time-delays, such as Internet TCP/IP connections. The scenario is depicted in Fig. 1.

The motivation for placing the predictive controller at the operator's side is twofold. First, the controller requires dedicated hardware (the more powerful the more accurate is the model of the plant) which sometimes can be not worth to place remotely, for example in space applications or in highly noisy environments. Second, a single remote server can run several predictive controllers for as many plants positioned in different locations and wired through a network.

We assume that the plant to be controlled is eventually equipped by a local controller, which, *in the absence of constraints*, stabilizes the plant, and denote by *primal system* the resulting closed-loop system.

We also assume that a predictive controller has already been selected and tuned so as to obtain constraint fulfillment and desired tracking performance *when no time-delay is present.* The predictive controller receives the desired trajectory to be tracked by human operators (or, in alternative, higher level computer programs), and provides command inputs. In Section 2 we show how to keep synchronized plant, predictive controller, and human operator. Finally, in Section 3 we report a simulation example of a teleoperated position servomechanism.

2 Problem Formulation

Consider a primal system of the form

$$\left\{\begin{array}{rcl} x'(t) & = & \phi(x(t), w(t)) \\ y(t) & = & h(x(t), w(t)) \\ c(t) & = & \ell(x(t), w(t)) \end{array}\right. \tag{1}$$

where: $x'(t)$ denotes either $\dot{x}(t)$ (if equations are expressed in continuous time) or $x(t+1)$ (discrete time), and accordingly $t \in \mathbb{R}$ (continuous time) or $t \in \{0, 1, \ldots\}$ (sampling steps); $x(t) \in \mathbb{R}^n$ is the state vector; $w(t) \in \mathcal{W} \subseteq \mathbb{R}^m$ the command input to the primal system; $y(t) \in \mathbb{R}^p$ the output which is required to track the reference $r(t) \in \mathbb{R}^p$; and $c(t)$ the vector to be constrained within a given set $\mathcal{C} \subseteq \mathbb{R}^q$. We assume that system (1), the reference $r(t)$, and the constraint set $\mathcal{C}$ satisfy the assumptions required by the specific predictive control law which has been selected, and, therefore, that desired convergence properties are guaranteed in the absence of time-delays. Let ΔT denote the period of the predictive controller, where for discrete time systems ΔT is an integer number, usually $\Delta T = 1$. We consider the following general class of predictive controllers. Let the sequence of future control moves $\{v(0),\ v(1),\ \ldots\}$ be described by a vector $\theta \in \mathbb{R}^{n_\theta}$, for instance

$$\begin{array}{rcl} \theta & \triangleq & [v'(0)\ v'(1)\ \ldots\ v'(N_u-1)]' \\ v(j) & \triangleq & v(N_u-1),\ \forall j = N_u, N_u+1, \ldots \end{array} \tag{2}$$

and $n_\theta = N_u m$. At each time $t = k\Delta T$, $k \in \{0, 1, \ldots\}$, an optimal sequence of future control moves $\{v(0),\ v(1),\ \ldots\}$ is evaluated by solving the optimization problem

$$\theta^*(t) = \left\{\begin{array}{l} \arg\min J(t, x(t), r(t), \theta) \\ \text{subject to } c(t) \in \mathcal{C} \end{array}\right. \tag{3}$$

where J is a performance index which depends on the predicted evolution of (1) due to initial state $x(t)$ and input $w(\tau) = v(j)$, $\tau \in [t+j\Delta T, t+(j+1)\Delta T)$, where τ is integer for discrete time primal systems. Typically, J is obtained by summing/integrating the squares of the tracking errors $y - r$ and inputs v on the interval

$[t, t+N_y\Delta T)$. Eventually, additional constraints are taken into account in (3), for instance terminal state constraints $x(t+N_y\Delta T)=0$.

Then, only the command input

$$w(t) = v^*(0) \tag{4}$$

is applied to the primal system (1), where for continuous time primal systems $w(t)$ is held constantly at $v(0)$ during the time interval $[t, t+\Delta T)$.

Without loss of generality, we assume asymptotical stability properties of (1). In fact, as mentioned above, (1) represents a model of a system which has been precompensated, typically via standard control techniques. Note that in this case, because of feedback loops, possible input saturations become state-dependent constraints. However, in (1) the constrained vector $c(t)$ can be any combination of inputs and states, and therefore possible saturating actuators can be still tackled.

Consider the control configuration depicted in Fig. 1, where there exists a delayed channel between the *operator's side*, where reference commands $r(t)$ are imposed and command inputs $w(t)$ are generated by the predictive controller (MPC), and the *plant's* or *remote side*, where the plant operates. Hereafter, for the sake of notational simplicity, we shall consider discrete time primal systems with $\Delta T = 1$.

Let τ_f and τ_b denote respectively the forward and backward communication delays. At time t, from the remote side the measurement of the current state $x(t)$ is sent to the operator's side. Here, at the same time t, $x(t-\tau_b)$ is received. This value is used to predict an estimate $\hat{x}(t+N-1)$ of the future state, which is visualized through a user interface to the operator. Accordingly, the operator generates a new reference command $r(t+N-1)$, which is processed by the MPC to obtain the command input $w(t+N-1)$. Simultaneously, at the plant's side $w(t+N-1-\tau_f)$ is received and buffered. Provided that $0 \le \tau_f \le N-1$, this scheme is synchronized, i.e. is able to provide the correct $w(t)$ to the primal system at each time t.

In principle, any controller could be used in the above scheme instead of an MPC. However, when time-delays cannot be a-priori upperbounded, for example on an Internet connection, the above scheme is not able to mantain synchronization between the operator's and the plant's side. As remarked above, an interesting feature of predictive controllers is that they provide a each time-step, as a by-product, seminfinite command sequences that can be "safely " applied to the plant— i.e. produce an evolution which satisfies the given constraints— no matter how large is the forward time-delay. Therefore, predictive controllers can be used to still ensure constraint fulfillment and avoid loss of synchronization between the teleoperator and the plant. This requires additional logic, which is depicted in the flow-charts in Figs. 2-3. Consider the plant's side algorithm (Fig. 2). As soon as the buffer becomes empty because of a forward delay greater than $N-1$, say this happens at time t_r, an alarm is sent to the MPC, and the plant enters a *recovery-mode*, where it is supplied by the last available virtual command sequence until acknowledged by the MPC. The alarm is labeled with the key $t^p_{old} \triangleq t_r - 1$. The recovery-mode is terminated as soon as new commands arrive from the MPC, in time to be executed, which are labeled as t^p_{old}.

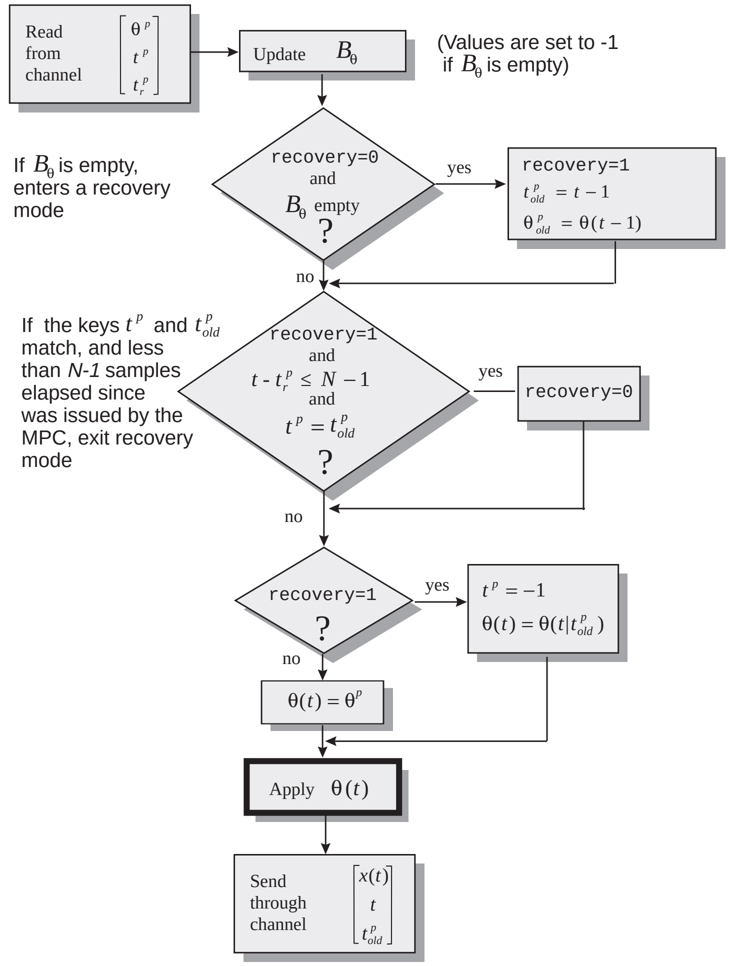


Figure 2: Plant's side algorithm

On the MPC's side (Fig. 3), the predictor starts providing wrong estimates as soon as the plant enters a recovery-mode, since the MPC commands issued for times $t > t_r$ have no longer been executed by the plant. As a new key t^p_{old} is received at time t^r_{rec}, the predicted state estimate is corrected and the issued commands are labeled with the new key t^p_{old}. This allows the plant to disregard the commands received through the channel until those ones labeled as t^p_{old} are received. The MPC assumes that the plant will end the recovery state later on at time $t_c \triangleq t^r_{rec} + N - 1$. As soon as the MPC receives from the plant data referring to the time $t - \tau_b = t_c$, the MPC actually verifies that the recovery mode was terminated properly, i.e. at time t_c the plant was not sending alarms ($t^r = -1$). If this is not the case, again the predicted state estimate is corrected and t^r_{rec} is set to the current time t.

We assume that initially the plant is at an equilibrium state x_0 corresponding to $w(t) \equiv w_0$. Accordingly, we set the vector θ_0 such that $v(k) \equiv w_0$, $\forall k \in \{0, 1, \ldots\}$. The execution of the algorithm at the MPC's side is started as soon as x_0 is received through the channel at time $t = 0$. The two algorithms are intialized as follows: `status`=0, `recovery`=0, $t^p_{old} = t^r_{old} = -1$, $\theta^p_{old} = \theta_0$, $t^r_{rec} = -1$, $t_c = -1$.

Remark 1 During the recovery state, the commands $w(t)$ are executed in open loop. While this is not a worry within a deterministic framework, it is fair to ask what "open-loop" leads in the presence of model or measurement errors and input disturbances. On the other hand, it should be noted that robustness is lost only with respect to the constraint fulfillment problem, because the plant is always in closed-loop with the local controller. Moreover, constraint violation problems due to non-deterministic situations might be overcome by adopting robust predictive controller, e.g. [3, 7].

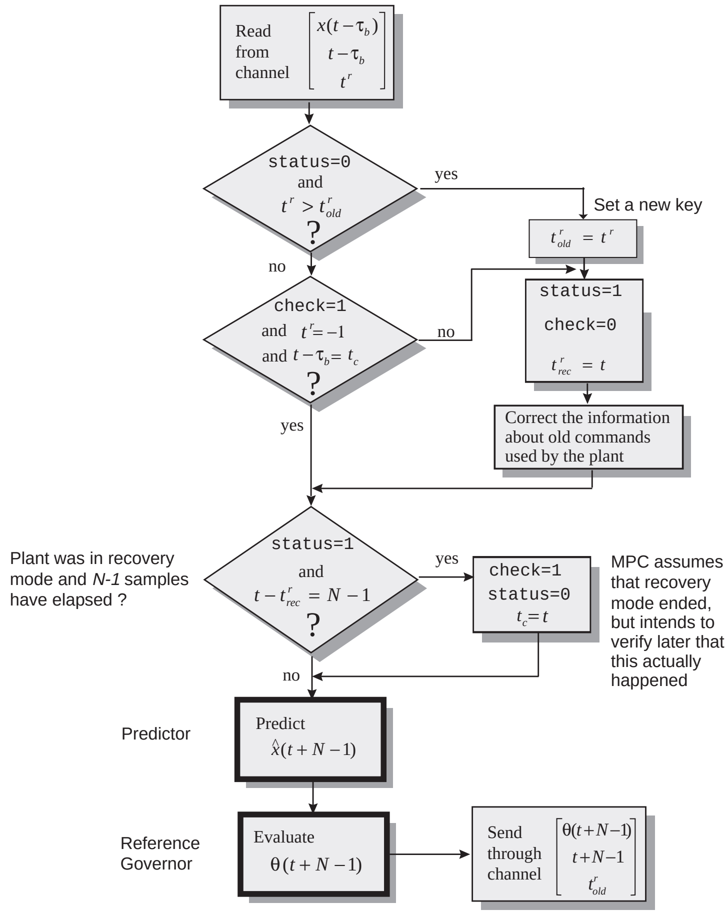


Figure 3: Predictive controller's side algorithm

3 A Simulative Example

The teleoperation scheme described in the previous section is applied to control the servomechanism depicted in Fig. 4. This consists of a DC-motor, a gear-box, an elastic shaft and a mechanical load. Technical specifications involve bounds on the shaft torsional torque T as well as on the input voltage V. The values of the parameters of the system are reported in [7]. Denoting by θ_M, θ_L respectively the motor and the load angle, and by setting $x_p \triangleq [\theta_L\ \dot{\theta}_L\ \theta_M\ \dot{\theta}_M]'$, the model can be described by the following state-space form

$$\dot{x}_p = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -\frac{k_\theta}{J_L} & -\frac{\beta_L}{J_L} & \frac{k_\theta}{\rho J_L} & 0 \\ 0 & 0 & 0 & 1 \\ \frac{k_\theta}{\rho J_M} & 0 & -\frac{k_\theta}{\rho^2 J_M} & -\frac{\beta_M + k_T^2/R}{J_M} \end{bmatrix} x_p + \begin{bmatrix} 0 \\ 0 \\ 0 \\ \frac{k_T}{R J_M} \end{bmatrix} V \tag{5}$$

$$\theta_L = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix} x_p \tag{6}$$

$$T = \begin{bmatrix} k_\theta & 0 & -\frac{k_\theta}{\rho} & 0 \end{bmatrix} x_p \tag{7}$$

where k_θ is the torsional rigidity, J_L and J_M respectively load and motor inertia, β_L and β_M load and motor friction coefficients, ρ the gear ratio, R the resistance of armature, and K_T the motor constant. Since the steel shaft has finite shear strength, determined by a maximum admissible $\tau_{adm} = 50 N/mm^2$, the torsional torque T must satisfy the constraint

$$|T| \leq 78.5398\ Nm \tag{8}$$

Moreover, the input DC voltage V has to be constrained within the range

$$|V| \leq 220\ V \tag{9}$$

The model is transformed in discrete time by sampling every $T_s = 0.1$ s and using a zero-order holder on the input voltage. The local digital controller has the following transfer function from $e = (r - \theta_L)$ to V

$$G_c(z) = 1000 \frac{9.7929z^3 - 2.1860z^2 - 7.2663z + 2.5556}{10z^4 - 2.7282z^3 - 3.5585z^2 - 1.3029z - 0.0853}, \tag{10}$$

and provides a very fast response but inadmissible voltage inputs and torsional torques for the references of interest, as shown in Fig. 5 for a set-point $r = 60$ deg.

As a predictive controller, the reference governor (RG) developed in [8] has been selected. This is a predic-

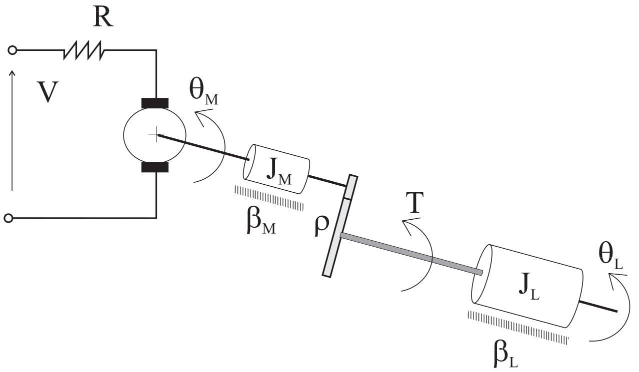


Figure 4: Servomechanism model.

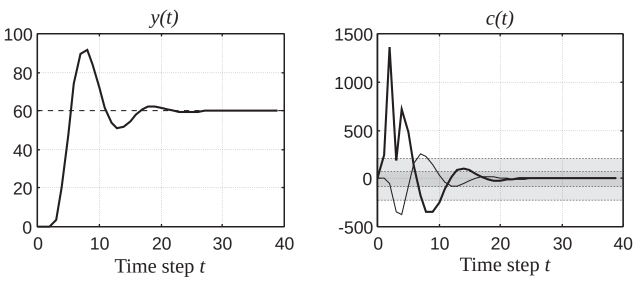


Figure 5: Unconstrained linear response of the primal system. The shadowed areas represent the admissible range for the voltage $V(t)$ (thick line, light gray), and the torque $T(t)$ (thin line, dark gray).

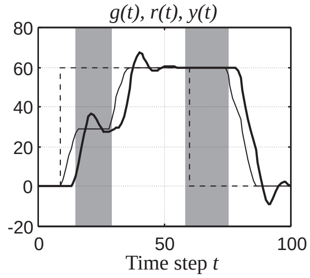
tive controller whose output is a reference trajectory rather than a control input to actuators, and it is added to a precompensated control system, which has already been designed to stabilize the plant, provide zero-offset set-point tracking in steady state, and provide nice tracking and disturbance attenuation properties, *in the absence of constraints*. Whenever necessary, the RG modifies the reference supplied to the precompensated system so as to enforce the fulfillment of the constraints, which otherwise would be violated. The RG operates in accordance with the receding horizon strategy mentioned above, by selecting *on-line* the optimal reference input sequence $\{v(k)\}$. This, in order to reduce the computational complexity, is parameterized by only two scalar quantities μ, $w \in \mathbb{R}$ in the following form

$$v(k) = \gamma^k \mu + w \tag{11}$$

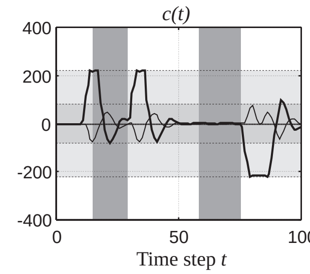
where $\gamma \in [0, 1)$ and the prime denotes transposition. Hence, in this case $\theta = [\mu\ w]'$. At each time t, $\theta^*(t)$ is computed by minimizing the performance index

$$J(x(t), r(t), \theta) = \|\mu\|^2_{\Psi_\mu} + \|w - r(t)\|^2_{\Psi_w} + \sum_{k=0}^{\infty} \|y(k, x(t), \theta) - w\|^2_{\Psi_y} \tag{12}$$

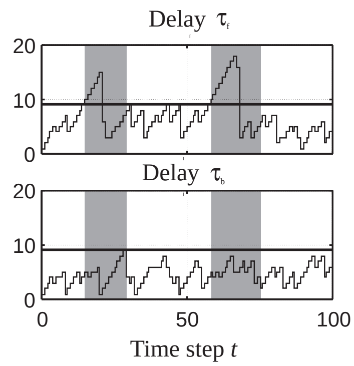
subject to the constraints $c(k, x(t), \theta) \in \mathcal{C}$, where $\|x\|^2_\Psi \triangleq x'\Psi x$, $\Psi_\mu = \Psi'_\mu > 0$, $\Psi_w = \Psi'_w > 0$, $\Psi_y = \Psi'_y \geq 0$, and $y(k, x(t), \theta)$ is the output response at time k due to the command (11) from initial state $x(t)$, the same notation being used for c. The task of the RG is to bound both the voltage $V(t)$ and the torque $T(t)$, according to constraints (8)–(9). In this example, the parameters are selected as $\gamma = 0.3$, $\Psi_\mu = 1$, $\Psi_w = 10$, and $\Psi_y = 0$.



(a) Output $y(t)$ (thick line), desired reference $r(t)$ (dashed line), and generated command $w(t)$ (thin line).



(b) Constrained quantities. Voltage $V(t)$ (thick line), torque $T(t)$ (thin line).



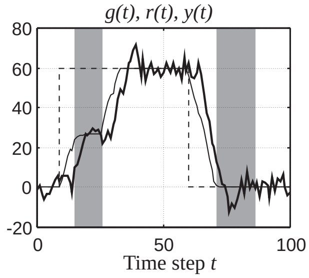
(c) Forward and backward delays $\tau_f(t)$, $\tau_b(t)$.

Figure 6: Teleoperation through a delayed channel. The intervals of recovery-state are depicted as gray areas.

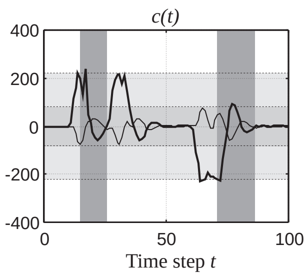
Teleoperation is performed through a channel where the forward delay $\tau_f(t)$ and the backward delay $\tau_b(t)$ are not known in advance, and cannot even be upperbounded a priori. The commands are generated $N-1 = 9$ steps in advance, and the delays are randomly generated between 1 and $2N-1$, with the obvious constraints $\tau_f(t) - \tau_f(t-1) \leq 1$, $\tau_b(t) - \tau_b(t-1) \leq 1$. The resulting trajectories are depicted in Figs. 6(a), 6(b), and 6(c).

In order to consider the effects of noise, a similar experiment is repeated by adding measurement noise

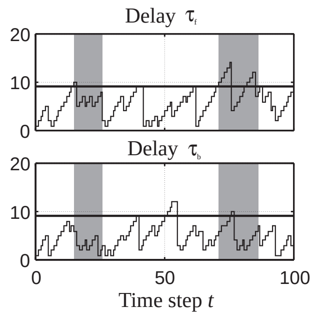
$$x(t - \tau_b) = x_{real}(t - \tau_b) + \xi$$



(a) Output $y(t)$ (thick line), desired reference $r(t)$ (dashed line), and generated command $w(t)$ (thin line).



(b) Constrained quantities. Voltage $V(t)$ (thick line), torque $T(t)$ (thin line).



(c) Forward and backward delays $\tau_f(t)$, $\tau_b(t)$.

Figure 7: Teleoperation through a delayed channel, with state measurement noise. The intervals of recovery-state are depicted as gray areas.

Each component ξ_i of the noise vector is uniformly distributed and has a maximum intensity equal to about the 10% of the full range of the corresponding component $x_i(t)$. Figs. 7(a), 7(b), and 7(c) show the related responses. Note that the prescribed bounds are only slightly violated.

4 Conclusions

This paper has presented a scheme for constrained systems teleoperated through communication channels with unbounded delays. This is achieved by using predictive controllers, because of their unique feature of automatically supplying command sequences which can be safely executed, despite arbitrarily large time delays. We wish that the proposed approach will be experimented for teleoperation through the Internet.

References

[1] T. Sheridan, "Space teleoperation through time delay: Review and prognosis," *IEEE Trans. Robotics Automat.*, vol. 9, no. 5, pp. 592–606, 1993.

[2] E. Mosca, *Optimal, Predictive, and Adaptive Control*. New York: Prentice Hall, Englewood Cliffs, 1995.

[3] M. Kothare, V. Balakrishnan, and M. Morari, "Robust constrained model predictive control using linear matrix inequalities," *Automatica*, vol. 32, no. 10, pp. 1361–1379, 1996.

[4] J. Rawlings and K. Muske, "The stability of constrained receding-horizon control," *IEEE Trans. Automat. Control*, vol. 38, pp. 1512–1516, 1993.

[5] D. Clarke and R. Scattolini, "Constrained receding-horizon predictive control," *Proc. IEE*, vol. 140, pp. 247–354, 1991.

[6] S. Keerthi and E. Gilbert, "Optimal infinite-horizon feedback control laws for a general class of constrained discrete-time systems: stability and moving-horizon approximations," *J. Opt. Theory and Applications*, vol. 57, pp. 265–293, 1988.

[7] A. Bemporad and E. Mosca, "Fulfilling hard constraints in uncertain linear systems by reference managing," *Automatica*, vol. 34, no. 4, pp. 451–461, 1998.

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