Second Order Sliding Mode Real-Time Networked Control of a Robotic Manipulator

Luca Capisani
Electrical and Automation Lab
Department of Computer Science Engineering
University of Pavia (Italy)
luca.capisani@unipv.it

Tullio Facchinetti
Robotic Lab
Department of Computer Science Engineering
University of Pavia (Italy)
tullio.facchinetti@unipv.it

Antonella Ferrara
Department of Computer Science Engineering
University of Pavia (Italy)
antonella.ferrara@unipv.it

Abstract

The distributed control of industrial plants, where the control algorithm runs on a machine that receives information from sensors and sends actuation commands over a communication network is a challenging task, due to the delays introduced by communication. This paper presents the networked control of a robotic anthropomorphic manipulator based on a second order sliding mode technique. The control objective is to track a desired trajectory for the manipulator. The adopted control scheme allows an easy and effective distribution of the control algorithm over two networked machines. While the predictability of real-time tasks execution is achieved by the S.Ha.R.K. real-time operating system, the communication is established via a standard Ethernet network. The performances of the control system are evaluated under different experimental system configurations using a COMAU SMART3-S2 industrial robot, and the results are analyzed to put in evidence the robustness of the proposed approach against the possible network delays.

1 Introduction

The remote control of industrial plants over communication networks, or *Network Control Systems (NCSs)*, is an important trend for the industrial automation. Traditional control techniques for robot manipulators are usually implemented so that the control loop is executed on the same machine at which sensors and actuators are directly connected through the I/O devices [22]. This allows one to consider negligible the delays between the three steps of a control procedure: sensor data acquisition, control algorithm execution, and actuators driving.

In contrast, in NCSs a network connects the machine where the control algorithm is executed and remote devices attached to sensors or actuators [24]. Data are sampled from sensors and control values are sent to actuators through the network. NCSs are receiving an increasing attention in control applications due to their positive impact on system costs, since off-the-shelf components can usually be employed, and centralized control machines can be used to control several devices/plants. Moreover, networked control offers advantages in terms of flexibility and modularity in system design [24]. Since networking devices become cheaper and cheaper, it is worth making the effort to move from using traditional control techniques implemented on machines directly connected to the system under control, to distributed networked control systems. Yet, in NCSs, network delays must be accounted in the controller design to achieve the desired performance.

In the past decades, many communication technologies have been proposed with the goal of achieving timeliness and real-time guarantees on the exchanged messages. Important examples are the CAN, PROFIBUS, TTP and many others (see [18] for a comprehensive overview). Meanwhile, the Ethernet communication standard, which was originally intended as a general-purpose networking technology, is grown as a major competitor for dedicated networks in the field of industrial control applications [15]. With its increasing speed, decreasing price, widespread usages and well-established infrastructure, Ethernet is now a concrete alternative of all the cited technologies.

In real-time networked control a crucial problem is the variable arrival rate, or *jitter*, of messages exchanged among the devices composing the control system. In other words, there is a non-constant difference between the arrival times of two consecutive messages. The presence of jitter on the scheduled control task is a relevant issue also in traditional (non networked) real-time control systems. Several approaches, like, for instance, [19], have been proposed to compensate such a jitter. Recently, the comparative assessment and evaluation of some possible approaches to reduce the jitter of real-time tasks have been provided in [7]. The communication jitter is usually limited using dedicated network protocols [12]. Since real-time communication protocols are often inspired to task scheduling techniques, such approaches could also be used to limit the impact of jitter on the message flow. However, no dedicated communication protocol is considered in this paper, but the standard Ethernet is used as underlying network layer.

The sliding mode control is a well known control technique which is especially appreciated for its simplicity and robustness against noise, which can be generated by model parameter variation, and bounded disturbances [11, 25]. However, it is a common opinion that one of the main drawback of sliding mode controllers is the so-called chattering [25, 26]. Especially in the robotic field, this phenomenon can disrupt the actuators, producing undesired vibrations. Moreover, the chattering phenomenon is significantly amplified if the control law is applied with delays, as happens, for instance, when the control frequency is reduced by the jitter introduced because of the use of a general purpose network to exchange the control values between the controller and the actuators. This phenomenon sometimes is called *discretization chatter effect* [26].

To circumvent the chattering problem various solutions have been proposed, but some of them lead to a loss of the robustness properties of the sliding mode control, as the so-called quasi sliding mode control approaches [4]. Other techniques consider the problem directly in the discrete time fashion [6, 8, 21, 26], but in this case it is often required an adaptive mechanism to cope with the disturbance effects [6, 20]. An alternative way to solve the problem of chattering is that of confining the discontinuities of the control law to its first derivative, see [3, 4, 5, 10, 17]. The idea underlying such an approach is that of enforcing a sliding mode on the manifold $s[x(t)] = 0$ in the system state space, with $\dot{s}[x(t)]$ identically equal to zero, by using a control signal depending on $s[x(t)]$, but directly acting on $\ddot{s}[x(t)]$. These approaches allow one to generate *second order sliding modes*.

In this paper, the so-called sub-optimal second order sliding mode technique (SOSMC) [3] is considered, and the controller is designed by combining second order sliding mode control with the so-called *inverse dynamics control* [1, 16, 22]. Note that a possible combination of these controllers has been investigated and evaluated, from a theoretical point of view in [13]. In order to obtain a suitable model for the inverse dynamic control, a specific identification procedure, recently devised [9], is applied.

The goal of this work is to evaluate the performances of the combined inverse dynamics/second order sliding mode controller when it is used for the remote control of an industrial device in presence of jitter. The evaluation is based on experimental results obtained by applying the control algorithm to a COMAU SMART3-S2 industrial robot. The experimental setup includes the implementation of the control algorithm under the S.Ha.R.K. real-time operating system [14], a modular real-time kernel which allows the easy implementation and management of periodic activities with hard real-time constraints.

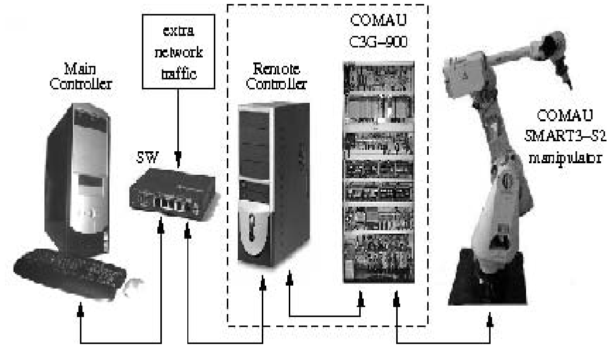

Figure 1. System configuration.

One of the main benefits of using the real-time features of S.Ha.R.K. is the high accuracy of the time handling.

The paper is organized as follows: Section 2 describes the architecture of the control system and the structure of the control application. Sections 3 and 4 illustrate the control algorithm. The control application and the features of the real-time operating system is described in Section 5. The experimental results are illustrated and compared in Section 6. Finally, Section 7 summarizes our conclusions.

2 System description

The control algorithm which will be described in Section 4 has been implemented on the system depicted in Figure 1.

The Remote Controller (RC), a 90MHz Intel Pentium, is directly interfaced with the COMAU C3G-900 unit through a dedicated interface board shipped with the manipulator. The same board is also used to read the values sampled from the resolvers mounted on the manipulator. The COMAU C3G-900 drives the COMAU SMART3-S2 manipulator, generating the correct power and voltage levels of the signal waveforms required to drive the motors of the manipulator.

The COMAU C3G-900 and the RC are grouped together in Figure 1 to indicate that they can be seen, from the viewpoint of the control system, as a single device.

Complex control algorithms, as the one adopted in this work, require heavy computations that cannot be executed by the RC. The networked control, on the other hand, allows to logically consider the RC as the I/O interface device of the manipulator, so as to circumvent the limitations in computational capability of the RC itself.

The Main Controller (MC) uses a 2GHz 64 bit AMD processor, on which the control algorithm can be executed together with many other tasks, like manipulator interaction control (using force sensors, proximity sensors and cameras), image processing, path planning, and a richer user interface.

The two machines, the MC and the RC, are linked through a 100Mbit Ethernet switch denoted with SW.

The control approach described in this paper has been tested on the SMART3-S2 industrial anthropomorphic rigid robot manipulator by COMAU, located at the Department of Electrical Engineering of the University of Pavia, shown in Figure 2. It consists of six links and six rotational joints driven by brushless electric motors. For

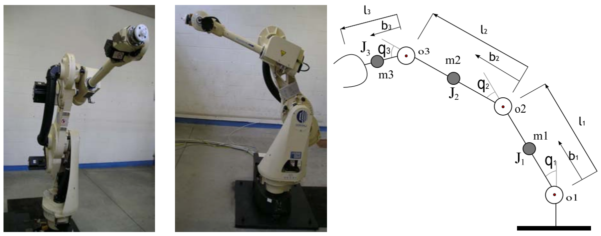

Figure 2. SMART3-S2 robot and the three link planar manipulator.

our purposes, joints 1, 4 and 6 are not used. As a result, it is possible to consider the robot, without loss of generality, as a three link-three joint in the sequel numbered as $\{1,2,3\}$, manipulator. The manipulator in the planar configuration is schematically represented in Figure 2. Yet, the proposed method can be easily extended to a n-joints robot.

3 The manipulator dynamic model

It is well-known that the dynamics of an n-joints robot manipulator can be written in the joint space, by using the Lagrangian approach, as

$$u = B(q)\ddot{q} + C(q,\dot{q})\dot{q} + g(q) + F_d sign(\dot{q}) + F_v\dot{q} \quad (1)$$

where $q \in \Re^n$ is the vector of joints displacements $q^T = [q_1, \dots, q_n]$, $u \in \Re^n$ is the vector of motor control torques, $B(q) \in \Re^{n\times n}$ is the inertia matrix, $C(q,\dot{q})\dot{q} \in \Re^n$ represents centripetal and Coriolis torques, $F_v \in \Re^{n\times n}$ and $F_d \in \Re^{n\times n}$ are the viscous and static friction diagonal matrices, and $g(q) \in \Re^n$ is the vector of gravitational torques.

The aim of the control strategy is to make the robot track a desired trajectory q_d, q_d being the vector of the desired joint displacements, and we assume that q_d, $\dot{q}_d$, $\ddot{q}_d$, and $\frac{d^3 q_d}{dt^3}$ exist and are bounded. In Figure 2, the following symbols are used for each link i: m_i is the mass, J_i is the inertia, θ_i is the angular position, l_i represents the link length and b_i the center of mass position with respect to the i-th link.

4 The control algorithm

The control algorithm considered in this work consists of an inverse dynamics control which performs a non ideal feedback linearization combined with a robust SOSMC [3]. In order to implement a feedback linearization, the parameters of the model (1) must be known. By applying an identification process recently devised [9], a suitable

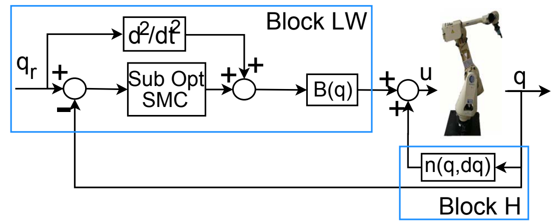

Figure 3. The manipulator control scheme with the decoupling-linearizing compensator. The block LW represents the lightweight part, while the block H represents the heavy part.

model of the form (1) can be determined for the considered robotic manipulator. Since the identified model is not perfect, the inversion of dynamics produces a non ideal cancellation of non linearities. This is the reason why the inverse dynamics approach is coupled with a robust control approach as SOSMC.

4.1 The inverse dynamics method

The inverse dynamics control consists in transforming the nonlinear system (1) into a linear and decoupled system by means of a suitable nonlinear feedback, see [2, 23]. More specifically, by choosing

$$u = B(q)y + n(q,\dot{q}) \quad (2)$$

with $n(q,\dot{q}) = C(q,\dot{q})\dot{q} + F_v\dot{q} + F_d sign(\dot{q}) + g(q)$, system (1) simply becomes $\ddot{q} = y$ (see Figure 3). Note that, even if the term $n(q,\dot{q})$ in (2) has been accurately identified, it can be quite different from the real one because of uncertainties and unmodelled dynamics, friction, elasticity and joint plays. Now assume that the term $\hat{n}(q,\dot{q})$ includes the identified centripetal, Coriolis, gravity and friction torques terms, while the inertia matrix $B(q)$ is known. So letting $u = B(q)y + \hat{n}(q,\dot{q})$, the compensated system becomes

$$\ddot{q} = y + B^{-1}\tilde{n} = y - \eta \quad (3)$$

where $\eta = -B^{-1}\tilde{n}$, being $\tilde{n} = \hat{n} - n$, represents an equivalent input noise.

4.2 The second order sliding mode control

The control objective can be reformulated as that of steering to zero the error state vector

$$x_i(t) = \begin{bmatrix} x_{i1}(t) \\ x_{i2}(t) \end{bmatrix} \quad (4)$$

where $x_{i1}(t) = q_{di}(t) - q_i(t)$, and $i = \{1,2,3\}$ represents the joint considered. In this case, a suitable sliding manifold can be chosen as

$$s_i = s(x_i) = x_{2i} + \beta x_{1i}, \ \ \beta > 0, \ \ \forall i \quad (5)$$

In the particular case of the second order sliding mode control, the controller aim is to steer to zero in a finite time not only the s sliding variable, but also its first order time derivative. Indeed, in this way, a chattering reduction is achieved because the control signal is continuous.

To design the controller here proposed, with the aim of testing it on the COMAU SMART3-S2 manipulator, the so-called *sub-optimal approach* introduced in [3] has been followed. Such an approach, based on the *bang-bang* control technique, enables to avoid the measurement of $\dot{s}$, while guaranteing robustness capabilities typical of sliding mode control. To design a control law of sub-optimal type to solve the problem in question, an auxiliary state vector can be introduced, by letting

$$\xi_i(t) = \begin{bmatrix} \xi_{i1}(t) \\ \xi_{i2}(t) \end{bmatrix} = \begin{bmatrix} s_i(t) \\ \dot{s}_i(t) \end{bmatrix} \qquad (6)$$

where s as in (5). Then, taking into account (3) and (4), the following auxiliary system can be formulated for each joint i

$$\begin{cases} \dot{\xi}_{i1}(t) = \xi_{i2}(t) \\ \dot{\xi}_{i2}(t) = F_i[q(t), \dot{q}(t), \ddot{q}(t), y_i(t)] - \dot{y}_i(t) + \\ \qquad + \frac{d^3 q_{di}(t)}{dt^3} + \beta \ddot{q}_{di}(t) \\ \dot{y}_i(t) = w_i(t) + \frac{d^3 q_{di}(t)}{dt^3} + \beta \ddot{q}_{di}(t) \end{cases} \qquad (7)$$

where

$$F_i(\cdot) = \dot{\eta}_i(t) + \beta\eta_i(t) - \beta y_i(t) \qquad (8)$$

includes all the uncertain terms and $w_i(t)$ can be regarded as an auxiliary control signal. By virtue of the robotic application here considered, the uncertain term can be assumed to be bounded. Let

$$| \; F_i[q(t), \dot{q}(t), \ddot{q}(t), y_i(t)] \; | < \overline{F}_i \qquad \forall i \qquad (9)$$

then, according to [4] the control signal $w_i(t)$ can be designed

$$w_i(t) = +\alpha_i W_{iMAX} sign\{\xi_{i1}(t) - \frac{1}{2}\xi_{i1MAX}\} \qquad (10)$$

where $W_{iMAX} > 0$, α_i is chosen as indicated in [4], and ξ_{i1MAX} is the last extremal value of ξ_{i1}, that can be found as specified in [3] or with the algorithm depicted in Figure 4, in which the δ term refers to the sampling time, and the k term refers to the sampling instant.

Note that, as in the original algorithm, no evaluations of the $\xi_{i2}(t)$ term are required.

The term $F_i(\cdot)$, cannot be determined but, since the uncertainty terms can be assumed bounded, the term W_{iMAX} can be determined through an experimental tuning procedure. Table 1, where W_{iMAX} refers to the i-th joint, shows the value of the parameters chosen to perform the experimental tests. From a theoretical point of view it can be proved that by choosing the amplitude of signal $w_i(t)$ as

$$W_{iMAX} > \left(\frac{\overline{F}_i}{\alpha^*}; \frac{4\overline{F}_i}{3 - \alpha^*} \right) \qquad \alpha^* \in (0, 1] \qquad (11)$$

if $\xi_{i1}(0) > 0$ **then**
  $fmax =$ **true**
else
  $fmax =$ **false**
end if
while $k\delta < T_{end}$ **do**
  if $fmax =$ **true** $\bigcap \xi_{i1}(\delta(k+1)) < \xi_{i1}(\delta k)$ **then**
    $\xi_{i1MAX} = \xi_{i1}(\delta k)$
    $fmax =$ **false**
  end if
  if $fmax =$ **false** $\bigcap \xi_{i1}(\delta(k+1)) > \xi_1(\delta k)$ **then**
    $\xi_{i1MAX} = \xi_{i1}(\delta k)$
    $fmax =$ **true**
  end if
end while

Figure 4. The algorithm to find the extremal values.

in accordance to [4], the state of system (7) is steered to zero in a finite time. This means that $s_i(t)$ and $\dot{s}_i(t)$ become zero in a finite time, i.e. a second order sliding mode is enforced, thus solving the control problem. Note that, as desired, the actual control signal acting on joint i, namely, $y_i(t)$, is continuous, so attaining the requirement of reducing the possible generation of vibrations.

Parameter	α^*	W_{1MAX}	W_{2MAX}	W_{3MAX}
Value	0.9	230	1330	10220

Table 1. Parameters used in the experiments on the COMAU SMART3-S2 manipulator.

The results obtained with this control algorithm, applied to the COMAU SMART3-S2 manipulator, will be discussed in Section 6.

4.3 The distributed control scheme

While the determination of the output of the SOSMC block of the control scheme in Figure 3 can be considered an easy task, the computation of the inverse dynamical part of the scheme may be quite heavy.

The proposed combined controller, produces a control action u which can be divided into two parts

$$u = \tau' + \tau'', \quad \tau' = B(q)y_{sm}, \quad \tau'' = \hat{n}(q, \dot{q}) \qquad (12)$$

in which $y_{sm} = [y_1, y_2, y_3]^T$ is the output signal of the sub-optimal algorithm (see Equation (7)). The computations of the terms τ' and $\hat{n}(q, \dot{q})$ can be conveniently seen as separated tasks, that can be executed on the two different machines (MC and RC) of our distributed architecture [24]. In our solution, the RC computes the reference signal and the τ' term, which requires few computational resources, while the MC, which has a more powerful processor, computes $\hat{n}(q, \dot{q})$ in real-time as depicted in Figure 6. Note that this architecture allows to easily control more than one robotic manipulator, by using a cheap remote controller for each robot and a centralized powerful computer to calculate, for each controlled system, the torque term $\hat{n}(q, \dot{q})$. As previously mentioned, this control scheme could also enable to perform, on top of MC, CPU-expensive tasks such as image processing and path planning.

4.4 Network control with random delays

In this section it is showed that, even if random delays and packet loss are present in the communication network, the robustness of the proposed distributed control scheme allows us to attain the control objective.

If we consider an unknown delay T affecting the packets exchanged between the MC and the RC, the real torque term $\hat{n}(q, \dot{q})$ becomes a delayed signal that can be expressed as

$$\hat{n}(q(t-T), \dot{q}(t-T)) \tag{13}$$

Therefore, the effect of delay can be modelled as an unknown input disturbance

$$d(q, \dot{q}, T) = \hat{n}(q(t-T), \dot{q}(t-T)) - \hat{n}(q(t), \dot{q}(t)) \tag{14}$$

Note that, since $\hat{n}(\cdot)$ can be considered as bounded in practical situations, $d(\cdot)$ results also bounded. With these assumptions, the dynamics of the system (3) become

$$\ddot{q}(t) = y(t) + B^{-1}(q)(\tilde{n}(q, \dot{q}) + d(q, \dot{q}, T)) \tag{15}$$

The packet loss problem can also be modelled as a delay effect. In fact, the RC only considers the last packet received from the MC. If packets are lost in the time interval $[t, t+T]$, the system behaves like having an additional, time-dependent unknown (but still bounded) input disturbance $d(q, \dot{q}, T)$.

In both cases, Equation (15) represents the controlled system with the additional disturbance, and can be rewritten as

$$\ddot{q}(t) = y(t) - \eta^{'}(q, \dot{q}, T) \tag{16}$$

Then, it is possible to apply the sub-optimal strategy, by choosing the auxiliary state vector as in (6) and having, for each joint i,

$$| F_i^{'}(\cdot) | = | \dot{\eta}_i^{'}(t) + \beta \eta_i^{'}(t) - \beta y_i(t) | < \overline{F}_i^{'} \tag{17}$$

as a consequence, by selecting the auxiliary control amplitude to respect the constraint

$$W_{iMAX} > \left(\frac{\overline{F}_i^{'}}{\alpha^*}; \frac{4\overline{F}_i^{'}}{3-\alpha^*} \right) \tag{18}$$

convergence to zero of $s_i(t)$, and $\dot{s}_i(t)$ is still guaranteed, in spite of the presence of jitter and packet loss.

In Section 6 the experimental results obtained with this control architecture are compared with the ones obtained by implementing the entire algorithm on the MC in case of random delays.

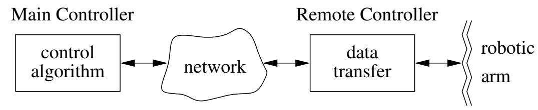

Figure 5. System configuration with the whole controller executed by the MC.

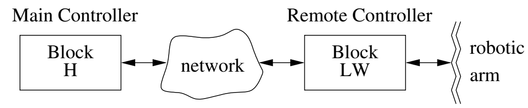

Figure 6. System configuration using the proposed distributed control scheme; see Figure 3 for blocks definition.

5 The control application

To assess the performance of the sliding mode control scheme, the controller has been implemented on top of the system architecture described in Section 2. Two different system configurations have been evaluated for testing the considered control scheme.

The first system configuration, depicted in Figure 5, refers to the case in which is implemented such as the whole control algorithm is executed on the MC. In this case, the RC is used only to forward the control values received from the MC to the power control device, and thus to the manipulator. The RC also reads the sensor samples from the manipulator and sends them to the MC.

The second system configuration is illustrated in Figure 6. With this configuration the control algorithm is splitted into two separated sub-processes: the linearization process is accomplished by the MC, while the sliding mode control algorithm is executed on the RC. The feedback linearization phase, as stated in Section 4.3, represents the heaviest part of the control algorithm. Thus executed by the MC, which is the fastest and easily to upgrade machine. On the other hand, after the linearization, the sliding mode control is a lightweight process, which can be computed by the slowest machine RC.

5.1 The real-time kernel

The real-time applications which run on the two computers depicted in Figure 1 have been developed using the implementation facilities provided by the S.Ha.R.K. real-time operating system.

S.Ha.R.K. is a free real-time kernel developed at the Scuola Superiore S. Anna of Pisa [14]. The main feature that distinguishes S.Ha.R.K. with respect to other real-time kernels is its high configurability. The user can combine different scheduling policies to comply with specific application requirements. Such a flexibility also applies to resource access protocols and aperiodic service algorithms. Each policy is implemented as a scheduling module that can be selected at system initialization to work in combination with the others modules according to a multi-level scheduling architecture. The kernel supports several I/O peripheral devices, including network cards, frame grabbers, and data acquisition boards.

The application implemented to perform the experiments described in Section 2 exploits some features of the S.Ha.R.K. operating system, such as the advanced time handling, the automatic scheduling of periodic time-constrained tasks, the resource sharing between task, and the network I/O devices.

6 The experimental results

In this section the experimental results obtained applying the described control strategy to the COMAU SMART3-S2 industrial manipulator are presented. In particular, three different control implementations have been tested:

1. the whole control algorithm, consisting of the parts LW and H in Figure 3, is executed on the MC without artificial packet delays (see Figure 5);

2. as before, the control algorithm is entirely executed by the MC, but an artificial delay is introduced in the messages exchanged between the MC and the RC (see Figure 5);

3. the control algorithm is distributed between the MC, which performs the LW block of Figure 3, and the RC, which executes the H block (see Figure 6). As in the previous case, the random network delay is introduced.

6.1 Random delay generation

When a standard switched Ethernet network is used to exchange the control values between the controllers, every packet may experience a random delay or it can be dropped (i.e., due to queue overflow). This can happen quite often when many hosts are simultaneously trying to send their messages.

In this paper, we consider that the network delay and the time interval in which packets are lost are bounded. This condition has been obtained by delaying the transmission of each message by a random value, which is upper bounded by $\overline{T}$. Moreover, if the delays cause a bursty arrival of packets to RC only the last message is considered, thus using the latest control values sent by MC.

Several experiments have been carried out on the COMAU manipulator, with different values of $\overline{T}$, showing the robustness of the proposed control scheme in presence of network delays.

6.2 Experiments definition and results

In order to test the control architecture on the COMAU SMART3-S2 manipulator, three different reference signals have been used, such as a step angular reference, with an amplitude of 2 degrees, for each considered joint; a sinusoidal angular reference, with an amplitude of respectively (30,30,40) degrees; a spline reference specified by

$$\dot{q}_{di}(t) = \begin{cases} A_i'' & \text{if } t - \mathcal{N}P \leq 1 \\ 0 & \text{if } 1 < t - \mathcal{N}P \leq 3 \\ -A_i'' & \text{if } 3 < t - \mathcal{N}P \leq 4 \\ 0 & \text{if } 4 < t - \mathcal{N}P < 6 \end{cases} \tag{19}$$

where t represents the time instant, $\mathcal{N} = \lfloor(\frac{t}{P})\rfloor$, $q_d(0) = (-20, 0, 0)^T$ represents the initial reference signal, $P =$ 6sec is the period of the signal, and A_i'' is the derivative amplitude, respectively $(70, 40, 60)$ degrees/sec for each joint. It is apparent that the selected reference signals violate the assumptions of $\dot{q}_{di}$, $\ddot{q}_{di}$, $\frac{d^3}{dt^3}q_{di}$ bounded. In spite of this, the controlled system behaves satisfactorily, as will be shown in the sequel. The reason for this, is due to the fact that the loss of boundedness takes place in an infinitesimal time interval, but the controlled system does not have an infinitesimal escape time.

In Table 2, the good performances of the proposed control approach are shown, by evaluating the Root Mean Square (RMS) of the tracking error for each experiment, i.e.

$$e_{RMSi} = \sqrt{\sum_{j=1}^{N} \frac{(q_{dij} - q_{ij})^2}{N-1}} \tag{20}$$

where N is the number of sampled data.

In each column of Table 2 there are the three RMS values associated with the experiment, one for each joint. In the first column the results obtained by running the whole control algorithm on the MC without jitter are reported. The second column displays the results obtained implementing the whole control algorithm on the MC and introducing the delay. Finally, in the third column the good results achieved with the distributed approach are shown.

6.3 Case 1: No artificial jitter

When no artificial jitter is introduced in the network, the quality of the control obtained with the network structure and the one achieved by the standard control structure results practically equivalent (see Table 2), even though the former has twice the sampling frequency of the latter (1000 Hz against 500 Hz). In the standard control structure the full control algorithm runs on the RC machine and, due to its scarce computation power, the sampling frequency has to be reduced. Using the proposed networked control allows to move the heavy calculations on a faster machine (MC). However, as depicted in Table 2, the chattering effect is low in both cases. This is due to the good control properties of the adopted second order sliding approach [4].

6.4 Case 2: Artificial jitter

In this case, many tests have been done by varying the maximum delay of each message between the expected and the actual arrival time on RC. The sliding mode controller, which in the ideal case is applied with infinite

Ref sig	e_{RMSi}[deg] *LW + H on MC* W/o Jitter	e_{RMSi}[deg] *LW + H on MC* jitter Max 12ms	e_{RMSi}[deg] *LW on RC / H on MC* jitter Max 30ms
step	0.0179 0.0528 0.0794	0.0585 0.7718 0.2128	0.0069 0.0470 0.0833
sin	0.0200 0.0906 0.2201	0.1558 0.8339 0.6524	0.0232 0.1280 0.2490
spline	1.529 0.2911 0.1772	1.9012 0.5625 0.7612	1.4858 0.2725 0.1948

Table 2. Comparison of the Root Mean Square Errors for the three joints using the traditional control approach and the proposed control approach (experimental results).

switching frequency, when an unpredictable delay can affect the switcher decision, exhibits discretization chatter effect [26]. This implies bad tracking of the reference signal and high vibrations of the robot manipulator, as can be deduced by the Root Mean Square values. In the left three plots of Figure 7 it can be seen that the reference tracking capability results non satisfactory, for instance, for a step reference signal.

6.5 Case 3: Distributed controller with artificial jitter

With the proposed control scheme, it is possible to execute a very cpu intensive computation, by using a data network commonly used in industrial and personal applications. In fact, the jitter acts only on a slow-varying part of the controller, enabling to construct a control architecture very robust to delay noise, by applying the core of the sliding mode controller with the maximum allowed frequency, and letting part of the inverse dynamics computation on a centralized network controller. In Table 2, it can be seen that, in this case, the results obtained introducing an artificial jitter bounded to 30 milli seconds, are practically comparable, and sometimes better than the ones obtained without the introduction of the jitter. In Figure 7, 8, and 9 it can be seen that with this controller a very good reference tracking is achieved, while Figure 10 shows a comparison of the error signal obtained with the proposed scheme against the ones obtained with the basic control scheme in which all the computations are on top of MC, with delays.

From this results it is clear that splitting the control algorithm in two controllers enables to perform all the calculations within the deadline of 1 ms if no jitter are introduced, then, if random delays of the order of tenth of milliseconds are present in the network, the good properties of the control algorithm are preserved.

7 Conclusions

In this paper the possibility of exploiting the robustness feature of second order sliding mode control to overcome the practical criticalities of the the networked control of a robotic manipulator is investigated.

The performances of different possible implementations have been compared to highlight their limits and benefits. The proposed real time robust control architecture, has been tested on a COMAU SMART3-S2 robotic manipulator showing that the chattering effect is very low

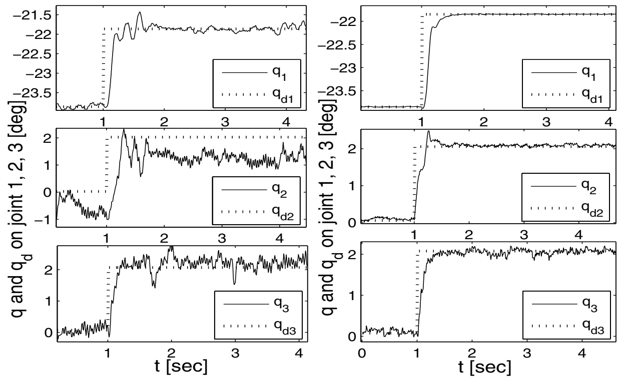

Figure 7. Comparison between the performances obtained with a step reference when all the computations are on top of MC (on the left) and the distributed control configuration (on the right).

despite of noise and mechanical non linearities, since the designed control action is continuous. Moreover, very good tracking capabilities can be obtained even if the controller is distributed over a network with delays.

References

[1] C. Abdallah, D. Dawson, P. Dorato, and M. Jamshidi. Survey of robust control for rigids robots. *IEEE Control System Magazine*, 11(2):24–30, 1991.

[2] H. Asada and J. E. Slotine. *Robot Analysis and Control*. Wiley, NJ, 1986.

[3] G. Bartolini, A. Ferrara, and E. Usai. Output tracking control of uncertain nonlinear second-order systems. *Automatica*, 33(12):2203–2212, 1997.

[4] G. Bartolini, A. Ferrara, and E. Usai. Chattering avoidance by second order sliding mode control. *IEEE Transactions on Automatic Control*, 43(2):241–246, 1998.

[5] G. Bartolini, A. Ferrara, E. Usai, and V. I. Utkin. On multi-input chattering-free second-order sliding mode control. *IEEE Transactions on Automatic Control*, 45(9):1711–1717, 2000.

[6] G. Bartolini, A. Ferrara, and V. I. Utkin. Adaptive sliding mode control in discrete-time systems. *Automatica*, 31(5):763–769, 1995.

[7] G. Buttazzo and A. Cervin. Comparative assessment and evaluation of jitter control methods. *Proc. of the 15th International Conference on Real-Time and Network Systems*, pages 29–30, March 2007.

[8] A. Calanca, L. M. Capisani, A. Ferrara, and L. Magnani. An inverse dynamics-based discrete-time sliding

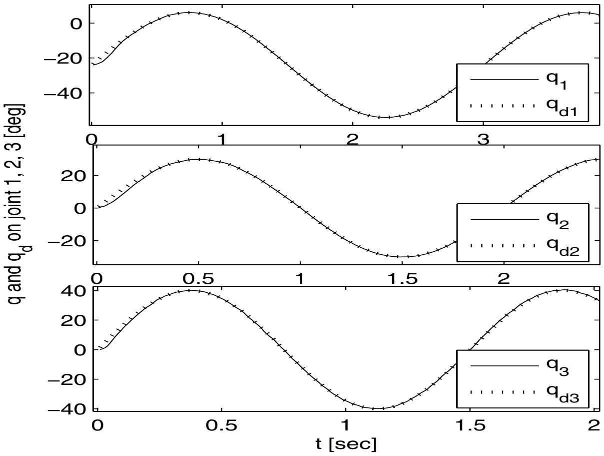

Figure 8. Good tracking of a sinusoidal reference signal with the proposed distributed control architecture.

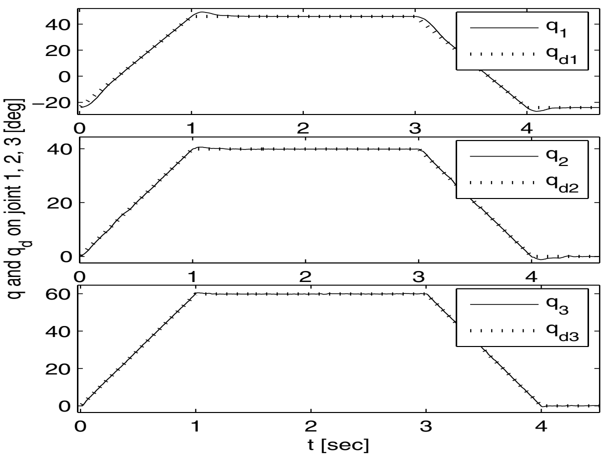

Figure 9. Good tracking of a spline reference signal with the proposed distributed control architecture.

mode controller for robot manipulators. *Lecture Notes in Control and Information Sciences, Robot Motion and Control 2007,Springer-Verlag, London*, 360:137–146, 2007.

[9] L. Capisani, A. Ferrara, and L. Magnani. Mimo identification with optimal experiment design for rigid robot manipulators. In *IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM2007)*, 2007.

[10] J. Davila, L. Fridman, and A. Levant. Second-order sliding-mode observer for mechanical systems. *IEEE Transactions on Automatic Control*, 50(11):1785–1789, 2005.

[11] C. Edwards and S. K. Spurgeon. *Sliding Mode Control: Theory and Applications*. Taylor & Francis, 1998.

[12] T. Facchinetti, G. Franchino, P. Pedreiras, and R. Marau. An ethernet layer for supporting enhanced real-time communication services. *Proceedings of the 12th IFAC Symposium on Information Control Problems in Manufacturing (INCOM 2006)*, pages 119–124, May 2006.

[13] A. Ferrara and L. Magnani. Motion control of rigid robot manipulators via first and second order sliding modes. *Journal of Intelligient and Robotic Systems*, 48:23–36, 2007.

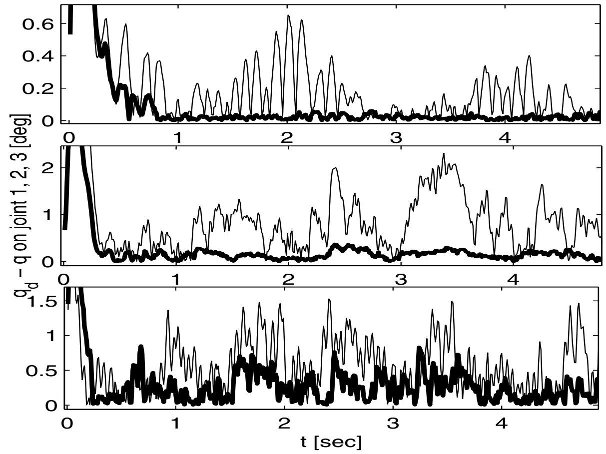

Figure 10. Error reduction using the proposed scheme (bold line) against the error obtained with the whole controller on MC (thin line) when a sinusoidal reference input is imposed and network delays are present.

[14] P. Gai, L. Abeni, M. Giorgi, and G. Buttazzo. A new kernel approach for modular real-time systems development. *Proceedings of the 13th IEEE Euromicro Conference on Real-Time Systems*, June 2001.

[15] G. Kaplan. Ethernet's winning ways. *IEEE Spectrum*, 38(1):113–115, 2001.

[16] C. Y. Kuo and S. P. T. Wang. Nonlinear robust industrial robot control. *ASME Journal of Dynamic Systems Measurements and Control*, 11:24–30, 1989.

[17] A. Levant. Homogeneity approach to high-order sliding mode design. *Automatica*, 41:823–830, 2005.

[18] N. P. Mahalik. *Fieldbus Technology: Industrial Nerwork Standard for Real-time Distributed Control*. Springer, 2003.

[19] P. Marti, G. Fohler, K. Ramamritham, and J. M. Fuertes. Jitter compensation in real-time control systems. *Proceedings of the IEEE Real-Time Systems Symposium*, December 2001.

[20] J. S. Perk, G. S. Han, H. S. Ahn, and D. Kim. Adaptive approaches on the sliding mode control of robot manipulators. *Transactions on Control Automation and System Engineering*, 3(2):15–20, 2001.

[21] S. Z. Sarpturk, Y. Istefanopulos, and O. Kaynak. On the stability of discrete-time sliding mode control system. *IEEE Transactions on Automatic Control*, 32(10):930–932, 1987.

[22] L. Sciavicco and B. Siciliano. *Modelling and Control of Robot Manipulators*. 2nd ed., Springer-Verlag Advanced Textbook in Control and Signal Processing, London, UK, 2000.

[23] M. W. Spong, F. Lewis, and C. Abdallah. *Robot Control: Dynamics, Motion Planning, and Analysis*. IEEE Press, NJ, 1993.

[24] Y. Tipsuwan and M. Y. Chow. Control methodologies in networked control systems. *Control Engineering Practice*, 11:1099–1111, 2003.

[25] V. I. Utkin. *Sliding modes in control and optimization*. Berlin, Springer, 1992.

[26] V. I. Utkin, J. Guldner, and J. Shi. *Sliding Mode Control in Electromechanical Systems*. Taylor & Francis, 1999.