A fault-tolerant real-time supervisory scheme for an interconnected four-tank system

Alessandro Casavola, Domenico Famularo, Giuseppe Franzè and Angelo Furfaro

Abstract—In this paper, the implementation of a Command Governor (CG) strategy on a real-time computing system is described for the supervision of a laboratory four-tank test-bed. In particular, the real-time architecture has been developed on the RTAI/Linux operating system kernel and the CG module has been implemented in C++ on a general purpose off-the-shelf computing unit. An accurate model of the four-tank process has been derived from both physical and experimental data and the applicability of the proposed method has been proved by means of real-time tests, which testified on the CG strategy ability to enforce the prescribed operative constraints even under unexpected adverse conditions, e.g. water pumps failures.

I. INTRODUCTION

The successful design and implementation of real-time control applications requires interactions between several technical disciplines including Control and Computer Engineering. This is especially important in view of current and future manufacturing, automotive, robotics, etc. applications where subsystems are integrated to increase their functionalities, reliabilities and reduce costs. Integration is done by the inclusion of coordinating control functions on the control side and by exploiting virtualization tools on the computer system side.

The aim of this paper is to describe a real-time implementation of a predictive constrained control strategy for the supervision of a laboratory test-bed. Constrained control strategies based on predictive ideas are now commonplace in process industry.

In spite of that, in the control system literature the real-time issue is not fully treated and the control community normally does not pay too much attention to the unavoidable real-time aspects beyond control algorithms.

While there exist in the literature several variants of the Constrained Control Paradigm, each one with its own special features, all control schemes based on predictive ideas rely on the concept of generating commands as solutions of an on-line constrained optimization problem. This problem is constructed on the basis of a process model and process measurements. Process measurements provide the feedback (and, optionally, feedforward) element in the control strategy. It should be noted that important issues, such as the efficiency and effectiveness of various numerical algorithms used to solve the on-line optimization problems are responsible for the usage of such sophisticated optimization-based control algorithms in real applications. Moreover, human factors, fault tolerance and programming environments may influence the choice as well. In fact, it is often hard to tell when these methods, and the corresponding hardware components are needed for improving the control performance in practice or when simpler control structures could be sufficient.

Moving from the previous discussion, we have considered a laboratory process composed of four interconnected water tanks and two supply pumps. The system is shown in Figs. 1, 2, where the inputs are the voltages to the pumps and the outputs the water levels in the lower two tanks. The quadruple-tank process can easily be built up by using two double-tanks, which are benchmarks in many control laboratories [2], [3]. The setup is simple, but the plant can be used as a test-bed to illustrate several interesting nonlinear phenomena.

The aim here is at developing an implementation of a real-time Reference Governor strategy on a PC equipped with a real-time operating system. It is important to emphasize that general-purpose computers are not suitable for hard real-time applications, despite their sufficient computational resources. The latter is mainly due to the fact that standard operating systems, e.g. Windows or Linux, exhibit phenomena deriving from Direct Memory Access (DMA), CPU cache mechanism, virtual-memory management, etc. that lead to nondeterministic latencies and jitter that cannot be allowed within a real-time environment [8]. Therefore, it is recognized that only real-time operating systems (RTOS) [8] are capable to avoid the above undesirable phenomena. Specifically, RTOS’s provide features such as real-time scheduling functions, inter-process synchronization mechanisms, memory locking flags allowing the usage of low-cost off-the-shelf computers in real-time control applications [9].

In view of this, a RTAI/Linux kernel based C++ implementation of the CG algorithm [19] has been considered. RTAI is a hard real-time extension for the Linux kernel, that allows one to intercept the Linux system calls and to emulate them via software. As a consequence, hard real-time activities can directly access hardware resources with full priority and preemption on standard non real-time Linux processes [9].

II. THE FOUR-TANK PROCESS PHYSICAL MODEL

In this section we derive a mathematical representation for the four-tank process from input/output measurements and geometrical/physical data. A schematic diagram of the
process is shown in Fig. 1 and the goal is to regulate the water levels $h_3(t)$ and $h_4(t)$ (plant outputs) at given set-points by acting on the incoming water flows via the supply pump voltages $V_1(t)$ and $V_2(t)$ (plant inputs).

Fig. 1. Schematic diagram of the four tanks process. The water levels in Tanks 3 and 4 are controlled by two pumps.

The geometrical parameters (length, height and width) of the four tanks, made out of aluminium and plexiglas (Fig. 2), are $12.15 \text{ cm}$, $23 \text{ cm}$, and $7.7 \text{ cm}$, respectively. The pumps are Rule™ 360 GPH type, having a 12 Volts supply voltage and the water tank levels are measured by using pressure transducers Cerabar™ T PMP 131 model. The tanks and the pumps are connected by flexible plastic pipes, whose diameters are $d_1 = d_2 = 2 \text{ cm}$, $d_{13} = d_{24} = d_3 = d_4 = 0.98 \text{ cm}$ and $d_{14} = d_{23} = 0.75 \text{ cm}$ (see Fig. 1).

Fig. 2. The laboratory four tanks process.

A nonlinear process model could be achieved by using mass balance and Bernoulli’s law

$$
\begin{align*}
\frac{dh_1}{dt} &= -\mu_1 A_1 \sqrt{2g h_1} - \mu_2 A_2 \sqrt{2g h_2} - \mu_3 A_3 \sqrt{2g h_3} + \frac{1}{A_1} q_1(V_1) \\
\frac{dh_2}{dt} &= -\mu_2 A_2 \sqrt{2g h_2} - \mu_3 A_3 \sqrt{2g h_3} + \frac{1}{A_2} q_1(V_2) \\
\frac{dh_3}{dt} &= -\mu_3 A_3 \sqrt{2g h_3} - \mu_4 A_4 \sqrt{2g h_4} + \frac{1}{A_3} q_2(V_2) \\
\frac{dh_4}{dt} &= -\mu_4 A_4 \sqrt{2g h_4} + \frac{1}{A_4} q_2(V_2)
\end{align*}
$$

(1)

where $A_i, i = 1, \ldots, 4$, are the tanks and outlet hole cross-section areas and $h_i, i = 1, \ldots, 4$ the water levels. The supply voltage to the $i$-th pump is $V_i$, the corresponding output water flow is given by the nonlinear law $q_i(V_i)$ and the gravity acceleration is denoted by $g$. The parameter values of the laboratory process are given in Table I.

The $\mu_i$ coefficients are used for taking into consideration the effects of water turbulence, real pipe water flows and other model uncertainties. Such factors cannot be avoided in an accurate system modelling and their numerical values are not so simple to be determined. In order to overcome such a drawback, the idea here developed is to resort to a gray-box identification process by considering the following mathematical model for the four-tank process

$$
\begin{align*}
\frac{dh_1}{dt} &= -\alpha_1 \sqrt{h_1} + \beta_1 V_1 + \beta_{1o} \\
\frac{dh_2}{dt} &= -\alpha_2 \sqrt{h_2} + \beta_2 V_1 + \beta_{2o} \\
\frac{dh_3}{dt} &= -\alpha_3 \sqrt{h_3} + \beta_3 V_2 + \beta_{3o} - \alpha_3 \sqrt{h_3} \\
\frac{dh_4}{dt} &= -\alpha_4 \sqrt{h_4} + \beta_4 V_2 + \beta_{4o} - \alpha_4 \sqrt{h_4}
\end{align*}
$$

(2)

where

$$
\begin{align*}
\alpha_1 &= \frac{1}{A_1} \sqrt{2g} (\mu_1 (a_{13} + a_{14})) , \quad \alpha_2 = \frac{1}{A_2} \sqrt{2g} (\mu_1 (a_{23} + a_{24})) \\
\alpha_3 &= \frac{1}{A_3} \sqrt{2g} (\mu_3 a_{34}) , \quad \alpha_4 = \frac{1}{A_4} \sqrt{2g} (\mu_4 a_{24}) \\
\beta_i &= \frac{k_{io}}{A_i}, \quad i = 1, 2
\end{align*}
$$

Based on experimental tests, it has been determined that the pumps operate in linear regimes when restricted within the voltage range $[6 \text{ V}, 8 \text{ V}]$. As a consequence, we will put those constraints on pump voltages and will consider the following linear relationship between the input voltages $V_i(t)$ and the incoming water flows $q_i(V_i(t))$, i.e. $q_i(t) = k_i V_i(t)$, $i = 1, 2$. Moreover, the parameters $\beta_{io} = \frac{k_{io}}{A_i}, i = 1, 2$, along with the coefficients $k_{io}$, will be treated as offsets and added to take into account such a simplified linear behavior.

### III. System Identification

In this section, the identification of the model (2) is described. Standard system identification techniques are used

---

**Table I**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$A_1$</td>
<td>93.555 cm$^2$</td>
</tr>
<tr>
<td>$A_2$</td>
<td>0.754 cm$^2$</td>
</tr>
<tr>
<td>$A_3$</td>
<td>0.442 cm$^2$</td>
</tr>
<tr>
<td>$A_4$</td>
<td>981 cm$^2$</td>
</tr>
</tbody>
</table>
TABLE II
ESTIMATED PARAMETERS OF THE MODEL (2)

<table>
<thead>
<tr>
<th>α</th>
<th>0.0713</th>
<th>β1</th>
<th>0.1769</th>
<th>β2o</th>
<th>-1.2102</th>
</tr>
</thead>
<tbody>
<tr>
<td>α2</td>
<td>0.1232</td>
<td>β3</td>
<td>0.4870</td>
<td>β2o</td>
<td>0.2429</td>
</tr>
<tr>
<td>α4</td>
<td>0.2348</td>
<td>β4</td>
<td>0.2127</td>
<td>β2o</td>
<td>0.3146</td>
</tr>
</tbody>
</table>

for that purpose [17]. It has resulted that the identified model is satisfactorily accurate and fits well with the previously described physical model structure. Identification experiments were performed with Pseudo Random Binary (PRBS) voltage signals as inputs. The PRBS levels were chosen so that the pumps remained in their linear regimes. The models (eq. (2)) obtained from the identification process are listed in Table II. Finally, in Fig. 3 the validation curves for the proposed model are depicted.

Fig. 3. Validation curves of the identified state-space model. Experimental data (solid-line) and estimated trajectories (dashed-line).

IV. COMMAND GOVERNOR (CG) DESIGN

A CG control scheme, with plant, primal controller and CG, is depicted in Fig. 4. A state-space description of the closed-loop plant regulated by the primal controller is given by

\[
\begin{align*}
\Phi x(t+1) & = Gg(t) + G_d d(t) \\
y(t) & = H_g x(t) \\
c(t) & = H_p x(t) + L g(t) + L_d d(t)
\end{align*}
\]

where: \( t \in \mathbb{Z}_+ \), \( x(t) \in \mathbb{R}^n \) is the overall state which includes the plant and compensator states (if a dynamic compensator is used); \( g(t) \in \mathbb{R}^{m_g} \), which would be typically \( g(t) = r(t) \) if no CG were present, is the CG action, viz. a suitably modified version of the reference signal \( r(t) \in \mathbb{R}^m \); \( d(t) \in \mathbb{D} \) an exogenous disturbance satisfying \( d(t) \in \mathbb{D} \), \( \forall t \in \mathbb{Z}_+ \), with \( \mathbb{D} \) a specified convex and compact set such that \( 0_{n_d} \in \mathbb{D} \); \( y(t) \in \mathbb{R}^{m_y} \) is the output, viz. a performance related signal which is required to track \( r(t) \); \( c(t) \in \mathbb{R}^n_c \) the vector to be constrained

\[
c(t) \in \mathcal{C}, \ \forall t \in \mathbb{Z}_+ \quad (4)
\]

with \( \mathcal{C} \) a specified convex and compact set. It is also assumed that

\[
(A1) \left\{ \begin{array}{l}
\Phi \text{ is a stability matrix, i.e. all eigenvalues are in the open unit disk;} \\
\text{System (3) is offset-free, i.e. } H(I_n - \Phi)^{-1} G = I_p
\end{array} \right.
\]

One important instance of (3) satisfying (A1) consists of linear plants under stabilizing feedback control. The offset-free property happens to be satisfied under a suitable feedback configuration which provides infinity dc loop gain. The CG design problem is that of generating, at each time instant \( t \), the set-point \( g(t) \) as a function of the current state \( x(t) \) and reference \( r(t) \)

\[
g(t) := \gamma(x(t), r(t)) \quad (5)
\]

in such a way that, under suitable conditions and regardless of disturbances, the constraints (4) are always fulfilled along the system trajectories generated by the application of the modified set-points \( g(t) \) and possibly \( y(t) \approx r(t) \). Moreover, it is required that: 1) \( g(t) \to \hat{r} \) whenever \( r(t) \to \hat{r} \), where \( \hat{r} \) is either \( r \) or its best feasible approximation; and 2) the CG has a finite settling time, viz. \( g(t) = \hat{r} \) for a possibly large but finite \( t \) whenever the reference stays constant after a finite time. Theoretical studies along these lines on CGs appeared in [1], [4], [5], [12], [13], [14], [18].

V. THE EXPERIMENTS

A. Computer controlled system

The objective of this section is to outline the implementation details of the proposed CG real-time based architecture. In Figure 5, the proposed hw/sw scheme is depicted. It can be noted that the computing hardware resources are directly managed by the Real Time Application Interface (RTAI). This sw layer is in charge to properly filter the Linux Kernel system calls and give suitable APIs for the real-time applications, while all non-real-time applications are running in a transparent way w.r.t. the RTAI layer action. Two different design approaches can be undertaken in developing a RTAI application. The first consists of an “ad hoc” module which runs and exchanges data within the kernel space.
The second way is more flexible and will be adopted here. It makes use of the LXRT interface modules, which allow hard real-time programs to run in the user space. The CG task for the four-tanks plant is depicted on the upper-right part of Figure 5. A simply customizable C++ framework has been developed to allow the simple switching between different plant families. In particular, the four-tank control task relies on the RTAISerial APIs, which support real-time data exchange over the serial port.

![Computer running RTA/Linux](image)

Fig. 5. Overview of the computer controlled system

### B. Numerical Results

The supervisory CG strategy has been implemented on a real-time computing unit. For CG design purposes, the model (2) has been linearized around the following equilibrium point $x_{eq} = [0.6065 1.305 5.5]^T$, $u_{eq} = [7.1550 6.9424]^T$, where $x(t) = [h_1(t) h_2(t) h_3(t) h_4(t)]^T$ and $u(t) = [V_1(t) V_2(t)]^T$. Then, the linearized model has been discretized using forward Euler differences with a sampling time $T_c = 0.1 \text{sec}$. and the following physical constraints have been considered

\begin{align*}
    h_i(t) &\leq 16, \, [\text{cm}], \quad i = 1, \ldots, 4, \quad (6) \\
    6 &\leq V_i(t) \leq 8, \, [\text{Volt}], \quad i = 1, 2, \quad (7)
\end{align*}

Specifically, constraints (6) are prescribing maximum water bound levels for each tank whereas in (7) maximum pump supply voltage saturation bounds are instead considered.

The following CG parameters, $\delta = 10^{-6}$, and $\Psi = I_2$ have been chosen and the constraint horizon $k_0 = 130$ was computed via the numerical procedure proposed in [14].

Further, in order to properly characterize the set of admissible disturbances acting on the four tanks process and the output measurement errors, the following convex and compact region is considered and used in the CG design:

\[ D := \{d \in \mathbb{R}^4 \mid Ud \leq h \} \]

where $U = \begin{bmatrix} I_4 & -I_4 \end{bmatrix}$ and $h = 0.5 \cdot \{1 1 1 1 1 1 1 \}^T \, [\text{cm}]$.

Then, a two-degree of freedom controller (the primal linear compensator)

\[ u(t) = \begin{bmatrix} 1.1190 & 0.0857 & 0.6696 & 0.1902 \\
    1.0857 & 0.9771 & 0.0793 & 0.5355 \\
    1.0118 & 0.0116 & -0.2175 & 1.1118 \\
\end{bmatrix} x(t) + \begin{bmatrix} 0.0116 \\
    -0.2175 \end{bmatrix} g(t) \]

was designed without taking care of the constraints, so as to asymptotically stabilize the closed-loop system and to regulate its steady state behavior with zero constant reference input tracking error.

Some details are now given regarding the CG real-time implementation. In Figure 6, the UML diagram of the proposed scheme is presented. The algorithm has been embedded into a customizable C++ framework, where abstract classes corresponding to the state measurements unit, reference signals and actuator devices are defined for matching purposes w.r.t. the real physical devices. Note that in such classes, the interface code between the data acquisition board and the pressure sensors/pumps devices is hidden. Algebraic computations have been performed by means of uBLAS, which is a C++ implementation of BLAS routines [6] and is included into the BOOST libraries [7]. Moreover, the CG output $g(t)$ is achieved by formulating a QP optimization problem (see [5]), which is the numerical CG engine, and here is solved by resorting to a C++ implementation of the Goldfarb-Idnani algorithm [15]. Finally, the control loop consists of a single RTAI hard real-time periodic task running at the highest priority. The real-time framework exploited in the implementation of the proposed constrained control strategy is the RTAI v. 3.5 hosted on a i386 Linux Kernel v. 2.6.19 architecture. In the next experiments we have contrasted the dynamical behaviors of the system with $e$ without the use of the CG device.

1) No Faulty case: The aim of this scenario is to test the effectiveness of the proposed scheme when it is requested that the lower tanks (Tank 3 and Tank 4) are holding a water level close to the ‘flood’ limit. To this end, we have considered the following situation:

Starting from the initial state $x(0) = [1.2 \, 1.4 \, 5 \, 6.8]^T$, the following set-points $h_{3, ref} = h_{4, ref} = 15.5 \, \text{cm}$ have been considered. The objective is that the water level is first reached and then maintained without constraint violation.

All the experimental results are reported in Figs. 7-9. As it clearly results, the controller (8) is not capable to deal with the critical scenario here proposed. In fact, constraint
violations arise in both water levels (Fig. 7) and in the pumps voltages (Fig. 8). In particular, an undesirable phenomenon can be observed in Tanks 3 and 4 where, at around 30 sec., the water is going to overcome the ‘flood’ limit equal to 16 cm.

Conversely, in the presence of the CG device, such an undesirable behavior is avoided and all variables are always constrained inside their bounds. This is obtained thanks to the CG variation of the nominal set-points $h_{3_{\text{ref}}}$, $h_{4_{\text{ref}}}$ into their best feasible approximation $g(t)$, as depicted in Fig. 9.

It is interesting to spend few words on the voltage behaviors due to the action of the primal controller, as depicted in Fig. 8 (dot-line). The effective applied voltages do not overcome the imposed physical limits due to the device characteristic of the Rule 360 GPH pump, therefore this enforces the fact that without the CG action is not possible to automatically impose hard constraints on the relevant system variables.

![Fig. 7. Water levels behaviors: solid-line with CG and dot-line without CG. The dashed line represents the boundary of the prescribed constraint.](image)

![Fig. 8. Voltages provided by the pumps: solid-line with CG and dot-line without CG. The dashed lines represent the boundaries of the prescribed constraint.](image)

2) Faulty case: In this second experiment the initial state has been chosen as $x(0) = [1.6 \ 0.8 \ 7.1 \ 6.6]^T$, while the nominal tank level set-points have been set to $h_{3_{\text{ref}}} = h_{4_{\text{ref}}} = 15$ cm. Moreover, in addition the following faulty scenario is considered:

A voltage drop of 20% lasting 50 seconds occurs on the Pump 1 at $t = 50$ sec. Such an unfavourable situation could be caused by several reasons, e.g. the electronics pump normal use and aging. This implies that the Pump 1 is capable to supply the Tank 1 at only 80% of the voltage ($V_1$) w.r.t. the no faulty condition.

![Fig. 9. Outputs of CG unit](image)

![Fig. 10. Water level behaviors: solid-line with CG and dot-line without CG. The dashed line represents the boundary of the prescribed constraint.](image)

Figs. 10, 11 show the dynamics of the compensated linear system with and without the CG unit. Under the proposed scenario, violations of the prescribed tank levels and voltage constraints result. In fact, just after the faulty time interval, starting at time $t = 52$ sec., the water levels in the Tanks 3 and 4 exceed the flood limits. This is mainly due to the fact that without the CG action the controller is no capable to adequately manage the fault occurrence by means of the...
supplied pumps voltage.

On the contrary when the responses of the system under the CG action are considered (solid-lines), no constraints violation is observed. This means that the system viability during the pre- and post-fault transient phases is ensured despite of the fault occurrence. This has been achieved by modifying the water level references from their nominal values into those reported in Fig. 12. In fact when the fault event occurs (grey zone) the water level references are suitably adjusted for dealing with the new operating scenario: essentially, the CG reconfigures the desired set-points on Tanks 3 and 4.

VI. CONCLUSIONS

The focus of this paper is to design and to prove fault tolerance capabilities of a CG control strategy for a laboratory interconnected four-tank system.

Special efforts have been devoted at investigating how this class of strategies behave under critical events such as faults on the electronic components. Pointwise-in-time constraints on maximum water levels and pump voltages have been considered. Constraints fulfillment is achieved by adjusting and reconfiguring the water level set-points.

Real-time experiments results demonstrate the applicability and the effectiveness of the public domain RTAI operating system in supporting the implementation of a quite sophisticated MPC method on low cost computing hardware.

REFERENCES