

# On the Design and Control of Wireless Networked Embedded Systems

Karl-Erik Årzén, Antonio Bicchi, Stephen Hailes, Karl H. Johansson, John Lygeros

**Abstract**—Wireless networked embedded systems are becoming increasingly important in a wide area of technical fields. In this tutorial paper we present recent results on the design of these systems and their use in control applications, that have been developed within the project Reconfigurable Ubiquitous Networked Embedded Systems (RUNES). RUNES is a European Integrated Project with the aim to control complexity in networked embedded systems by developing robust and scalable middleware systems. New components for control under varying network conditions are discussed for the RUNES architecture. The paper highlights how the complexity of the closed-loop system is increased, due to additional disturbances introduced by the communication system: additional delays, jitter, data rate limitations, packet losses etc. Experimental work on integration test beds that demonstrates these results is presented, together with motivating links to the RUNES disaster relief tunnel scenario.

## I. INTRODUCTION

Quality metrics for control applications that are executed over wireless networks are different from the corresponding notion of quality of service for personal communication and multimedia applications. Control performance is related in a complex and dynamical way to the interaction between plant, sensors and actuators with one or more controller nodes. For networked control systems, the complexity is further increased due to the coupling with the variations and disturbances introduced by the communication channels. A network in a closed-loop control system may introduce additional delays, jitter, data rate limitations, packet losses etc.

The vision of the European project Reconfigurable Ubiquitous Networked Embedded Systems (RUNES) is to enable the creation of large scale, widely distributed, heterogeneous networked embedded systems that inter-operate and adapt to their environments. The inherent complexity of such systems must be simplified if the full potential for networked embedded systems is to be realized. The RUNES project aims to develop technologies (system architecture, middleware, etc.) to assist in this direction, primarily from a software and

communications standpoint. Control applications, however, impose additional requirements on the RUNES platform that arise from the need to manipulate the environment in which the networked systems are embedded. The purpose of this paper is to highlight these requirements and present recent results that have been developed within the RUNES project to address the needs of networked control systems.

The paper presents details on three key problems that arise in networked control systems: control under variable communication rate, latency and packet loss. A survey of results developed in the RUNES project in these three areas is given, together with references to publications with in-depth analysis. The research is motivated by a disaster relief tunnel scenario, which illustrates the potential of advanced control in future applications of networked embedded systems. Motivated by this application, a test bed on control of mobile robots using an ad hoc wireless network has been developed and can be used to illustrate and evaluate the results.

Section II presents the motivating scenario for networked embedded systems. Section III presents results on the three fundamental constraints imposed on networked control systems by the communication infrastructure: quantization, latency and packet loss. Finally, the mobile robot test bed motivated by the scenario is described in Section IV.

## II. MOTIVATING SCENARIO: TUNNEL DISASTER RELIEF

The potential applications of the RUNES platform are highly varied, given the rich set of sensing modalities and environmental conditions that can be considered. To illustrate one potential application in greater detail, the project currently focuses on a disaster relief tunnel scenario. The scenario acts as a source of architectural requirements and reference points for technology trials and integration.

The scenario deals with disaster relief activities in response to an emergency, in particular a fire in a road tunnel caused by an accident. The scenario comes with a story line that sets out a sequence of events and the desired response of the system, part of which is as follows. Initially, traffic flows normally through the road tunnel; then an accident results in a fire. This is detected by a wired system that is part of the tunnel infrastructure and is reported back to the Tunnel Control Room. The emergency services are summoned by Tunnel Control Room personnel. As a result of the fire, the wired infrastructure is damaged and the link is lost between fire detection nodes. However, using wireless communication as a backup, information from the fire detection nodes continues to be delivered to the Tunnel Control Room seamlessly. The first response team arrive from the fire brigade. Several robots and a number of firemen are sent into

This work was supported by the European Commission through the Integrated Project RUNES, FP6-IST-004536. The authors gratefully acknowledge the contribution by their collaborators within the RUNES project.

K.-E. Årzén is with Department of Automatic Control, Lund University, Sweden [karlerik@control.lth.se](mailto:karlerik@control.lth.se)

A. Bicchi is with Centro E. Piaggio, University of Pisa, Italy [bicchi@ing.unipi.it](mailto:bicchi@ing.unipi.it)

S. Hailes is with Department of Computer Science, University College London, United Kingdom [s.hailes@cs.ucl.ac.uk](mailto:s.hailes@cs.ucl.ac.uk)

K. H. Johansson is with School of Electrical Engineering, Royal Institute of Technology, Stockholm, Sweden [kallej@ee.kth.se](mailto:kallej@ee.kth.se). Corresponding author.

J. Lygeros is with Department of Electrical and Computer Engineering, University of Patras, Greece [lygeros@ee.upatras.gr](mailto:lygeros@ee.upatras.gr)

the tunnel. Each carries a wireless communication gateway node, sensors for environmental temperature, chemical and smoke monitoring, and the robots carry light detectors that help them identify the seat of the blaze. The role of the robots is both to help identify hazards and people that need rescuing without exposing the firefighters to danger, and to augment the communications infrastructure to ensure that both tunnel sensor nodes and those on firefighters remain in contact with the Tunnel Control Room. To accomplish this, the robots are controlled remotely over wireless links, the control algorithm taking into account both information from tunnel sensors about the state of the environment and from a human controller about overall mission objectives. A local backup controller allows the robots to behave reasonably in the event that communication is lost.

### III. CONTROL UNDER VARYING NETWORK CONDITIONS

The unreliability of wireless communication has important implications for networked control applications. If a control loop is closed over a wireless link the application must tolerate a high number of lost packets and be able to run in open-loop over considerable periods of time. If this is not the case, a local backup controller must be implemented. In this section, we describe a few aspects on control under varying network conditions studied within RUNES. We provide details on control under variable communication rate, latency and packet loss, and references to publications where an in-depth analysis can be found.

#### A. Control under variable information rate

Traditional control theory is usually based on ideal assumptions on the information flow across the control loop. However, as soon as real implementations are considered, these assumptions prove to be inadequate and the system's closed-loop performance turns out to be severely affected. These problems arise with particular impact in the framework of control over wireless networks, where multiple feedback loops share a limited pool of computation and communication resources. One of the most direct effects is quantization. Measurements used as outputs for monitoring and controlling a plant over the network have to be encoded in a finite number of symbols before being sent to the processors at the available communication rate. The controller, typically implemented as a set of concurrent processes in one or more computing units, provides feedback actions to properly steer the system. These actions also have to be codified in symbols, which must be sent over the network to reach the plant, where they can be decoded in suitable continuous control functions to be fed to the plant.

In order to clarify the relations between information rate, quantization error and system performance, let us consider the case of an unstable scalar linear system under quantized control:

$$\dot{\tilde{x}}(t) = a\tilde{x}(t) + \tilde{u}(t) + \tilde{w}(t), \quad a > 0, \quad \tilde{x}(0) = \tilde{x}^0, \quad (1)$$

where  $\tilde{w}(t) \in I(w) := [-w/2; w/2]$ ,  $w \geq 0$ , represents a bounded exogenous noise term. The control  $u(t)$  takes values

in a finite set  $\mathcal{U} \subset \mathbb{R}$ . Such a system is denoted by the pair  $(a, w)$ . We assume that the state  $\tilde{x}$  is sampled periodically at time  $0, T, 2T, \dots$ . Based on the sample, the control value is selected and transmitted over a finite capacity communication bus to a zero-order hold device in the corresponding actuator node. The sampled-data control system corresponding to (1) is

$$x(k+1) = \Phi x(k) + \Gamma u(k) + w(k), \quad x(0) = \tilde{x}^0, \quad (2)$$

where  $x(k) = \tilde{x}(kT)$ ,  $\Phi = \exp(aT)$ ,  $\Gamma = \int_0^T \exp(as) ds$  and  $w(k) = \int_{kT}^{(k+1)T} \exp(a((k+1)T - s)) \tilde{w}(s) ds$ . Such a system is denoted by the triple  $(a, w, T)$ . Notice that the discrete-time disturbance  $w(k)$  takes values in  $I(\Gamma w)$ . In this setting, the continuous-time control law is the piecewise constant function. The practical stability problem consists in guaranteeing the invariance of a sufficiently small neighborhood of the equilibrium irrespective to any noise affecting the systems. There is no loss of generality in formulating the problem for the discrete time model [1]. For a system  $(a, w, T)$  with control set  $\mathcal{U}$ , the interval  $I(\Delta)$  is said to be controlled invariant if  $\forall x^0 \in I(\Delta)$  there exists  $u \in \mathcal{U}$  such that  $\forall w \in I(\Gamma w)$ ,  $x^{0+} = \Phi x^0 + \Gamma u + w \in I(\Delta)$ . We assume that a limited bandwidth communication bus of capacity  $\mathcal{R}$  is connecting the plant and the controller, that is a device capable of transmitting  $\mathcal{R}$  bits per unit of time. In particular, the number of symbols  $\sigma$  that can be transmitted during the time interval  $T$  satisfies  $\sigma \leq 2^{\mathcal{R}T}$ . Since the number of bits to be transmitted at each sampling instant is integer, we require that  $\sigma \leq 2^{\lfloor \mathcal{R}T \rfloor}$ . Consider system  $(a, w)$ , suppose that a communication bus of capacity  $\mathcal{R}$  is connecting the controller to the plant: the triple  $(\mathcal{R}, T, \Delta)$  is said to be feasible for the invariance problem if there exists a control set  $\mathcal{U} \subset \mathbb{R}$  rendering  $I(\Delta)$  controlled invariant for system  $(a, w, T)$  and satisfying  $\#\mathcal{U} \leq 2^{\lfloor \mathcal{R}T \rfloor}$ .

It can be proved that if  $\Delta$  is such that the triple  $(\mathcal{R}, T, \Delta)$  is feasible, then there exists a feasible triple  $(\mathcal{R}, T', \Delta)$  such that the corresponding control set  $\mathcal{U}$  ensuring the invariance of  $I(\Delta)$  is of the type  $\mathcal{U} = \{-\rho/2, \rho/2\}$ . Actually, in that case, invariance can always be ensured by a binary controller [2], [1]. The parameter  $\rho > 0$  is called the dispersion of  $\mathcal{U}$  and is equal to twice the quantization error. A well known fact is that a necessary condition for the triple  $(\mathcal{R}, T, \Delta)$  to be feasible is  $\mathcal{R} \geq a/\log 2$ , see [3], [4], [5], [1]). We are now interested in the following problems, see [1] for details:

*Problem 1:* For a given  $\mathcal{R} > a/\log 2$ , find the smallest value of  $\Delta$ , say  $\Delta_m(\mathcal{R})$ , such that the triple  $(\mathcal{R}, T, \Delta)$  is feasible for some  $T > 0$ . Determine the corresponding dispersion  $\rho(\mathcal{R})$  of  $\mathcal{U}$ .

It can be shown that the solution to this problem is

$$\Delta_m(\mathcal{R}) = \frac{2w(e^{a/\mathcal{R}} - 1)}{a(2 - e^{a/\mathcal{R}})} \quad (3)$$

$$\rho(\mathcal{R}) = \frac{we^{a/\mathcal{R}}}{2 - e^{a/\mathcal{R}}}. \quad (4)$$

*Problem 2:* For a given  $\mathcal{R} > a/\log 2$  and an assigned dispersion  $\rho > 0$ , find the smallest value of  $\Delta$ , say  $\Delta_m(\mathcal{R}, \rho)$ ,

such that the triple  $(\mathcal{R}, T, \Delta)$  is feasible for some  $T > 0$  with  $\mathcal{U} = \{-\rho/2, \rho/2\}$ .

It can be shown that the solution to this problem is

$$\Delta_m(\mathcal{R}, \rho) = \frac{\rho + w}{a} (e^{a/\mathcal{R}} - 1), \quad \mathcal{R} \geq \frac{a}{\log\left(\frac{2\rho}{\rho+w}\right)}, \quad \rho > w.$$

First of all we notice that Problem 1 is trivial if a noise term is not considered. Indeed, if  $w = 0$ , then  $\Delta_m(\mathcal{R}) \equiv 0$ . This means that for any fixed rate  $\mathcal{R} \geq a/\log 2$ , the invariance of an arbitrarily small interval  $I(\Delta)$  can be guaranteed. It is a consequence of the fact that the control set  $\mathcal{U}$  can be chosen with an arbitrarily small dispersion, entailing an arbitrarily small quantization error. Problem 2 is meaningful also for  $w = 0$  because of the constraint imposed by the fixed control dispersion.

As it is expected, by increasing the communication rate better performance can be obtained, namely trajectories can be confined within a smaller invariant region. Equation (3) quantifies the decrease of  $\Delta_m(\mathcal{R})$  as the bandwidth  $\mathcal{R}$  increases: in particular,  $\lim_{\mathcal{R} \rightarrow +\infty} \Delta_m(\mathcal{R}) = 0$  and

$$\Delta_m(\mathcal{R}) \sim \frac{2w}{\mathcal{R}} \quad \text{for } \mathcal{R} \rightarrow +\infty.$$

Also the dispersion can be diminished by increasing the communication rate, nevertheless it is lower bounded by the amplitude of the noise, namely

$$\rho(\mathcal{R}) > w \quad \forall \mathcal{R} \in ]\frac{a}{\log 2}; +\infty[ \quad \text{and} \quad \lim_{\mathcal{R} \rightarrow +\infty} \rho(\mathcal{R}) = w.$$

In Problem 2 the supplementary constraint

$$\mathcal{R} \geq a/\log\left(\frac{2\rho}{\rho+w}\right)$$

is required. Notice that

$$a/\log\left(\frac{2\rho}{\rho+w}\right) \geq a/\log 2$$

and that equality is achieved if and only if  $w = 0$ : therefore the reason for the need of a larger bandwidth relies on the presence of both the noise term and the fixed control dispersion.

Even if the dispersion is fixed, arbitrarily small intervals can be rendered invariant by increasing the bandwidth. Nevertheless, the decrease with  $\mathcal{R}$  is slower: for a given  $\rho > w$ ,  $\lim_{\mathcal{R} \rightarrow +\infty} \Delta_m(\mathcal{R}, \rho) = 0$  and

$$\Delta_m(\mathcal{R}, \rho) \sim \frac{w + \rho}{\mathcal{R}} \quad \text{for } \mathcal{R} \rightarrow +\infty.$$

It is easy to verify that for all  $\mathcal{R} \geq a/\log\left(\frac{2\rho}{\rho+w}\right)$ ,  $\Delta_m(\mathcal{R}, \rho) \geq \Delta_m(\mathcal{R})$ , where equality is achieved for  $\rho = \rho(\mathcal{R})$ .

### B. Control under variable latency

Controlling a system connected to a server over a mobile wireless network is the focus of this section. In particular, we consider the issues of variable latency times for such a control system. We again consider a time invariant, discrete time, linear system

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) \\ y(k) &= Cx(k), \end{aligned}$$

but this time allow the state input and output dimension to be arbitrary. With zero-latency communication the utilized controller corresponds static output feedback [6], or  $\tilde{u}(k) = Ky(k)$ . With non-zero delays, however, the measurement used to compute the control signal is delayed to  $y(k-d_2)$  and the applied control signal is further delayed to  $u(k) = \tilde{u}(k-d_1)$ . Let  $r_s(k) = d_1 + d_2$  be the round trip delay at time  $k$ . Note that since the delay is due to the communication network used to carry sensor readings and control commands, the round trip delay will in general be time varying (hence the dependence on  $k$  of  $r_s(k)$ ). We assume that the round trip delay can be measured (e.g. by time stamping the data packets) and use it to ‘‘gain schedule’’ the controller  $K$ . The resulting control law is then given by

$$u(k) = K(r_s(k))Cx(k-r_s(k)),$$

where  $r_s(k)$  is a bounded sequence of integers  $r_s(k) \in \{0, 1, \dots, D\}$ , and  $D$  is the upper bound of the delay term.

The closed-loop system can be represented as a switched system by augmenting the state vector to include all the delayed terms

$$\tilde{x} = [x(k)^T x(k-1)^T \dots x(k-D)^T]^T.$$

The dynamics of the open-loop system, at time  $k$ , with the augmented state vector then take the following form

$$\begin{aligned} \tilde{x}(k+1) &= \tilde{A}\tilde{x}(k) + \tilde{B}u(k) \\ y(k) &= \tilde{C}(r_s(k))\tilde{x}(k) \end{aligned}$$

where

$$\tilde{A} = \begin{bmatrix} A & 0 & \dots & 0 \\ I & 0 & \dots & 0 & 0 \\ 0 & I & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & I & 0 \end{bmatrix}, \quad \tilde{B} = \begin{bmatrix} B \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

$$\tilde{C}(r_s(k)) = [0 \quad \dots \quad 0 \quad C \quad 0 \quad \dots \quad 0],$$

where the vector  $\tilde{C}(r_s(k))$  has zero elements, except from element number  $r_s(k)$ , whose value is  $C$ .

The closed-loop system is switched [7], since  $r_s(k)$  is of time-varying nature. The overall closed loop system is

$$\begin{aligned} \tilde{x}(k+1) &= \tilde{A} + \tilde{B}K(r_s(k))\tilde{C}(r_s(k))\tilde{x}(k) \\ y(k) &= \tilde{C}(r_s(k))\tilde{x}(k) \end{aligned}$$

The closed-loop matrix  $\tilde{A} + \tilde{B}K(r_s(k))\tilde{C}(r_s(k))$  can switch in any of the vertices  $A_i$  obtained by substituting the  $D+1$  possible values of  $r_s(k)$  in the above expression. The stability problem under variable network induced delays then reduces to the stability problem for the switched linear system

$$\tilde{x}(k+1) = A_i \tilde{x}(k), \quad i = 0, \dots, D.$$

Under the assumption that at every time instance  $k$  the round trip latency time  $r_s(k)$  can be measured, and therefore the index of the mode  $i$  above is known, the system can be described as:

$$x(k+1) = \sum_{i=0}^D \xi_i(k) A_i x(k), \quad (5)$$

where  $\xi(k) = [\xi_0(k), \dots, \xi_D(k)]^T$  and

$$\xi = \begin{cases} 0 & \text{if mode} \neq A_i \\ 1 & \text{if mode} = A_i. \end{cases}$$

The work of [6] has shown that the stability of this switched system is ensured if  $D+1$  positive definite matrices  $P_i$ ,  $i = 0, \dots, D$  can be found that satisfy the following LMI:

$$\begin{bmatrix} P_i & A_i^T P_j \\ P_j A_i & P_j \end{bmatrix} > 0, \forall (i, j) \in I \times I, \quad (6)$$

$$P_i > 0, \forall i \in I = \{0, 1, \dots, D\}. \quad (7)$$

Based on these  $P_i$ -matrices, it is feasible to calculate a positive Lyapunov function of the form:

$$V(k, x(k)) = x(k)^T \left( \sum_{i=0}^D \xi_i(k) P_i \right) x(k)$$

whose difference  $\Delta V(k, x(k)) = V(k+1, x(k+1)) - V(k, x(k))$  is a negative function along all the solutions of the switched system, thus ensuring asymptotic stability.

In [8], this condition was used to establish the largest sets  $I = \{I_{\min}, I_{\min} + 1, \dots, I_{\max}\}$ , where  $0 \leq I_{\min} < I_{\max} \leq D$  for which a preset controller gain can stabilize the system against any switching of the delay within members of this set. In [9] this approach was then extended to deal with multi-hop networks, where the variation of delays is considerably higher.

An upper limit of the maximum (theoretical) bound of  $D$  can be obtained from the continuous-time solutions of the time delayed system. Assume that the transfer function  $G(s)$  description of the system can be cast in the state-space format

$$\dot{x}_c(t) = A_c x(t) + B_c u(t), \quad y(t) = C_c x(t).$$

Given the controller's action  $u(t) = Ky(t - \Delta_L^1 - \Delta_L^2)$  the closed-loop system takes the form delay  $\tau^{\max}$  can be computed from the solution of the following optimization problem (with a set of LMIs) [10], [11]:

$$\begin{aligned} \tau^{\max} = \max \tau, & \quad \text{subject to} \\ \begin{bmatrix} (A_c + A_d)Q_1 + Q_1(A_c + A_d)^T & \tau Q_1 A_c^T & \tau Q_1 A_d^T \\ + \tau A_d(Q_2 + Q_3)A_d^T & & \\ \tau A_c Q_1 & -\tau Q_2 & 0 \\ \tau A_d Q_1 & 0 & -\tau Q_3 \end{bmatrix} < 0 \\ Q_i > 0, \quad i = 1, 2, 3. \end{aligned} \quad (8)$$

Given  $\tau^{\max}$  the maximum delay  $D$  that preserves stability can be computed as  $D = \lceil \frac{\tau^{\max}}{T_s}, 1 \rceil$ .

### C. Control under variable packet loss

As discussed in Section IV, another major source of difficulty with networked embedded systems is that data that must be communicated within the control system might be lost. The data drop can cause severe performance degradation of the control system. A common way to handle packet drops between two end nodes in a computer network is to re-send data. For control systems, this is in most cases not a suitable approach, because detecting and communicating the drop information often takes a considerable period of time. There are at least three approaches to cope with variable packet

loss in control systems closed over networks: (1) add error correction to the transmitted sensor and control data packets to counteract the losses, (2) modify the state estimation in the control node, and (3) modify the control law explicitly.

By adding redundancy to the original data before sending it over the network, the receiver can recover the original data through decoding of the received packets if a sufficient number of the total packets arrive. Adding redundancy in this way is called forward error correction. The reliability in transferring a message can be increased at the cost of increased network load. Hence, there will be a trade-off between the quality perceived by the control application and the resources used. The amount of redundancy that best handles this tradeoff will vary with the network conditions and possibly also application demands. Consequently, there is a need to adapt the amount of redundancy online, which is called adaptive forward error correction. A new feedback control algorithm for adapting redundancy is developed and evaluated in [12]. The data packets are assumed to be collected into blocks of fixed length  $N$  where in each block  $b$ ,  $N - u_b$  packets contain application data and  $u_b$  packets contain redundancy. This corresponds to the application having a fixed share of bandwidth. With such a coding scheme, it is possible to recover the original data if at most  $u_b$  packets are lost. The receiver will thus, in each block, get  $y_b$  packets of application data where  $y_b = N - u_b$  if at most  $u_b$  packets are lost and  $y_b = 0$  otherwise. Replacing application data by redundancy decreases the throughput, since fewer data are transmitted. But using too little redundancy will also decrease the throughput, since it might not be possible to recover the information sent in case of error. The objective for adapting the forward error correction is hence to maximize the throughput. From a control point of view, this can be done in two ways. The first is to use a model of the packet loss process and using online estimation of the parameters of this model to determine the current network state. The optimal redundancy can then be found by optimizing the throughput given the loss model. The major drawback with this approach, however, is the dependence on the accuracy of both the loss model and the parameter estimation. If the model or the estimated parameters, or possibly both, are not correct, the applied redundancy might differ from the optimal. We therefore propose a second approach, which is based on extremum seeking control. The redundancy controller uses feedback information of the effective throughput and gradient estimation to find the maximal throughput. When evaluated in simulations, this feedback controller tracks the optimum for different loss models and changes in network conditions, and performs better than the feed-forward controller in presence of model errors.

Estimation when data is lost can be done by modifying traditional estimation algorithms. Sinopoli et al.[13] consider estimation in the presence of independent losses of the measurement. They derive a Kalman filter, which is quite similar to the traditional Kalman filter, but the measurement update is just a propagation of the time update states when

a measurement is lost. They also show that there exists a critical arrival probability,  $\lambda_c$ , for the expected value of the state covariance to be bounded, i.e., for arrival probabilities below this value the mean covariance will be unbounded for some initial conditions, while arrival probabilities above this value will give bounded covariance for all initial conditions. Liu and Goldsmith [14] extend this work to the case when each measurement is sent in several packets, where some or all packets may be lost. Micheli and Jordan [15] consider continuous-time systems, where measurements occur at random time instances.

There are several proposals for handling packet loss explicitly in the control algorithm. Seiler and Sengupta [16] consider a case when losses for a linear plant are given by a Bernoulli process. By augmenting the state vector with the most recent output value, a time varying plant model containing the network model is derived. It results in a Markovian jump linear system with two modes. An LMI condition for mean-square stability given a certain packet loss probability is presented. Ling and Lemmon [17], [18], determine how the power spectral density of the output is affected by feedback drops. They use this to derive a compensator. Sinopoli et al. [19] extend their work on estimation discussed above to also include control. The control objective is to minimize a quadratic cost function of the states and control input, given old measurements and loss indicators. From this, they show that the optimal controller is a linear state feedback from observed states. Moreover, the separation principle holds in presence of communication losses, i.e., the estimator does not depend on the control input. In the case of a infinite horizon LQG, a lower bound on the arrival probability depending only on the maximum eigenvalue of the process is derived.

#### IV. INTEGRATION TEST BED

The tunnel disaster relief scenario requires control actions at several levels. Most of these actions involve a human in the loop component and are difficult to test in practice. To illustrate some of these networked control aspects in a lab environment, a test bed was developed to explore control of mobile robots over an ad hoc wireless network. The test bed involves one or more mobile robots moving around in an environment in which a sensor network is deployed, see Figure 1. The mobile robots are themselves nodes in the network, i.e., the network contains both stationary and mobile nodes, and they need remote feedback control action in order to accomplish their tasks. In the test bed these tasks are abstracted in the form of a one-dimensional inverted pendulum on a mobile robot. The open-loop instability of this process makes it especially challenging in a wireless control context.

The control problem is to stabilize the pendulum in the upright position by moving the robot in the forward or backward direction. It is assumed that the node on the robot can measure the pendulum angle and that it commands the motors to drive forward or backward, and to change the direction of the robot. It is further assumed that the robot

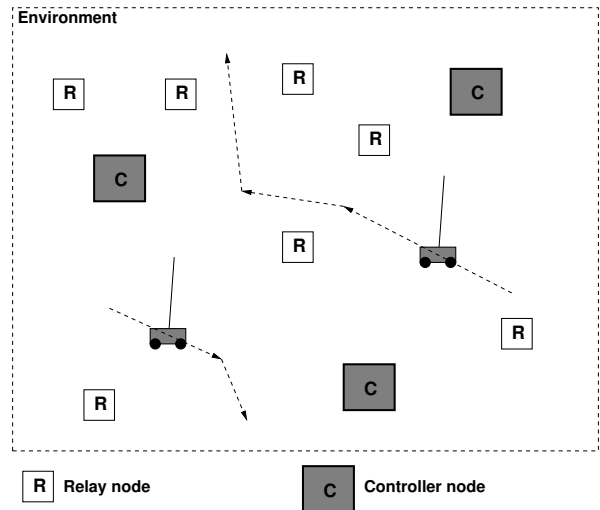


Fig. 1. The integration test bed consists of mobile robots in a sensor network environment. The nodes marked “R” are relay nodes and the nodes marked “C” are controller nodes.

knows its own position, e.g., through the use of GPS, UWB or ultrasonic position, or vision. In the tunnel scenario this information would be provided by the localization services of the wireless network.

Different control modes can be defined. In the local control mode it is the local sensor network node on the robot that performs the stabilization control. Alternatively, the control can be done by one or several of the stationary nodes in the network. One approach is to let a subset of the stationary nodes be designated controller nodes, with the capacity to realize the feedback control. The remaining stationary nodes would then simply be relay nodes that forward the sensor packets from the mobile node to the most suitable, e.g., closest, controller node, using static or dynamic routing with one or several hops.

The stabilization control for the inverted pendulum robot is implemented on a Moteiv Telos mote, either that which is local or one that is remote. A state feedback structure (equivalent to a PD-controller) is used. Finally, an additional PD-controller is used for position control of the robot. A vision system is used as a sensor. The camera is connected using Firewire to a PC where the image processing is performed. The position of the robot is then broadcast to the sensor nodes from a gateway sensor node connected to the PC. The overall control scheme is shown in Figure 2.

Several experiments were performed with both static and dynamic routing schemes. The communication delay per hop for the typical packets being sent were around 10 ms. Thus, a sensor packet from the local Telos mote directly to a controller node, followed by a control signal packet back to the local Telos node would imply a communication latency of around 20 ms. This should be compared with the sampling period of the stabilization controller of 50 ms. Hence, in order to have a communication delay less than the sampling period, a maximum of two hops can be allowed from the sensor/actuator node to the controller node and

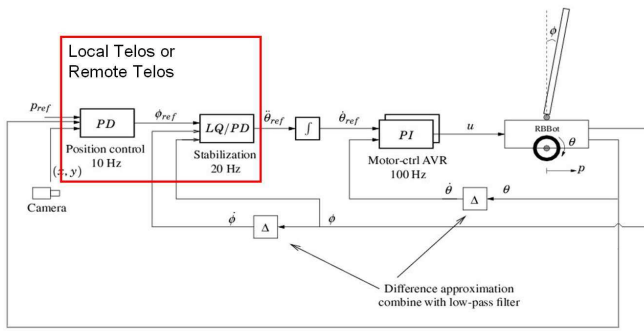


Fig. 2. Mobile robot control scheme.

from the controller node back to the sensor/actuator node. The unstable pendulum dynamics also necessitate the use of a local backup controller, to be used in the case one or several packets are lost.

After emitting the sensor packet, the local sensor node waits for control packets with an index tag that corresponds to the sensor packet recently sent out. The first control packet received within the sampling period of 50 ms is applied to the actuators; later incoming control packets are discarded. If no control packet is received within 50 ms the local backup controller is executed and a red lamp is activated on the robot. The on-time of the lamp provides a rough estimate of the reliability of the communication.

The experiences from the inverted pendulum scenario are mixed. The control scheme outlined above is rather simple, but it highlights some of the key difficulties of networked control systems. The results obtained on this test-bed have motivated more advanced control schemes described in the next section. The Telos mote technology together with the TinyOS environment worked without problems. However, a surprisingly large number of packets were lost in the wireless communication. Unless the internal communication in the network is scheduled to avoid collisions, e.g., by trying to enforce time-division, a large number of collisions leading to resends or lost packets occur. Scheduling the communication in a sensor network is in principle impossible. For example, in this scenario the broadcast packets from the camera node and the packets needed to maintain the tables in the dynamic routing scheme would interfere with the sensor and control signal packets in a non-predictable way. However, even if the internal communication was scheduled, by turning off the camera and using static routing, packet losses still occur. One explanation for this can be the “contaminated” radio environment found in typical in-door university locations, with numerous WLAN networks, laptops, and Bluetooth-enabled mobile phones. Another problem with this scenario is the relatively low speed of the IEEE 802.15.4 radio protocol. The per hop delay for a packet with 20 Byte payload was approximately 10 ms. This limited the number of multi-hops that could be allowed.

## V. CONCLUSIONS

In this paper, we have reviewed some of the main problems encountered in realistic applications of networked embedded systems. To focus attention, we have described a motivating scenario, the RUNES disaster relief tunnel case study. We have highlighted how the complexity of the closed-loop system is increased, due to additional disturbances introduced by the communication system. Experimental work on integration test beds that demonstrates these results was presented, with reference to a particularly challenging problem of stabilization over a wireless network.

## REFERENCES

- [1] B. Picasso, L. Palopoli, A. Bicchi, and K. H. Johansson, “Control of distributed embedded systems in the presence of unknown—but-bounded noise,” in *Proc. of IEEE Conference on Decision and Control*, 2004.
- [2] K. Li and J. Baillieul, “The appropriate quantization for digital finite communication bandwidth (dfbc) control,” in *Proc. of IEEE Conference on Decision and Control*, 2003.
- [3] S. Tatikonda, “Control under communication constraints,” Ph.D. dissertation, LIDS, MIT, USA, 2000.
- [4] J. Baillieul, “Feedback designs in information-based control,” in *Proc. of the Workshop on Stochastic Theory and Control*. Kansas: Springer-Verlag, 2001, pp. 35–57.
- [5] G. N. Nair and R. J. Evans, “Stabilizability of stochastic linear systems with finite feedback data rates,” *SIAM Journal on Contr. Optim.*, 2004.
- [6] J. Daafouz, P. Riedinger, and C. Iung, “Stability analysis and control synthesis for switched systems: A switched lyapunov function approach,” *IEEE Trans. on Automatic Control*, vol. 47, no. 11, pp. 1883–1887, 2002.
- [7] S. Ge, Z. Sun, and T. Lee, “Reachability and controllability of switched linear discrete-time systems,” *IEEE Transactions on Automatic Control*, vol. 46, no. 9, pp. 1437–1441, Sept. 2001.
- [8] G. Nikolakopoulos, A. Tzes, and I. Koutroulis, “Development and experimental verification of a mobile client-centric networked controlled system,” *European Journal of Control*, vol. 11, 2005.
- [9] G. Nikolakopoulos, A. Panousopoulou, A. Tzes, and J. Lygeros, “Multi-hopping induced gain scheduling for wireless networked control systems,” in *Proceedings of the IEEE Conference on Decision and Control*, Seville, Spain, December 12-15 2005.
- [10] M. Mahmoud, “Robust control and filtering for time delay systems,” *Marcel Dekker Inc.*, Jan 2000.
- [11] J. Zhang, C. Knopse, and P. Tsiotras, “Stability of Time-Delay Systems: Equivalence between Lyapunov and scaled Small Gain Conditions,” *IEEE Transactions on Automatic Control*, vol. 46, pp. 482–486, March 2001.
- [12] O. Flårdh, K. H. Johansson, and M. Johansson, “A new feedback control mechanism for error correction in packet-switched networks,” in *44th IEEE Conference on Decision and Control and European Control Conference*, 2005.
- [13] B. Sinopoli, L. Schenato, M. Franceschetti, K. Poolla, M. Jordan, and S. Sastry, “Kalman filtering with intermittent observations,” 2004.
- [14] X. Liu and A. Goldsmith, “Kalman filtering with partial observation losses,” in *43rd IEEE Conference on Decision and Control*, 2004.
- [15] M. Micheli and M. I. Jordan, “Random sampling of a continuous-time stochastic dynamical system,” in *Proceedings of the Fifteenth International Symposium on Mathematical Theory of Networks and Systems*, 2002.
- [16] P. Seiler and R. Sengupta, “Analysis of communication losses in vehicle control problems,” in *American Control Conference*, vol. 2, 2001, pp. 1491 – 1496.
- [17] Q. Ling and M. Lemmon, “Power spectral analysis of networked control systems with data dropouts,” *IEEE Transactions on Automatic Control*, vol. 49, no. 6, pp. 955–959, 2004.
- [18] —, “Optimal dropout compensation in networked control systems,” in *IEEE Conference on Decision and Control*, 2003.
- [19] B. Sinopoli, L. Schenato, M. Franceschetti, K. Poolla, and S. Sastry, “Time varying optimal control with packet losses,” in *43th IEEE Conference on Decision and Control*, 2004.