Stabilization by encoded feedback with Markovian communication channels ⋆

R. Sailer and F. Wirth *

* *Institut für Mathematik, Universität Würzburg, Am Hubland, D-97074 Würzburg, Germany, (e-mail: {sailer,wirth}@mathematik.uni-wuerzburg.de).*

Abstract: We consider stabilization over communication channels with delays and packet loss. Using a variation of dynamic quantization an encoder/decoder scheme is presented which is able to achieve stabilization if a closed loop system exists which is ISS with respect to measurement errors. Conditions for stabilization require bounds on the long-term average throughput of the communication channel. In communication channels which can be modelled as Markov processes this can be guaranteed almost surely provided an ergodicity assumption is met.

Keywords: Control with limited information, quantization, ISS, nonlinear systems, Markov process, asymptotic stability

1. INTRODUCTION

In this paper we consider the problem of stabilizing a nonlinear finite-dimensional system over a digital communication channel. The situation may be represented as in Figure 1. One of the main problem in this area is that the amount of information that can be sent from the sensing device to the controller is limited and can be corrupted in several ways. This type of problem has received considerable attention in recent years, De Persis and Isidori [2004], De Persis [2007], Wu and Chen [2007], Hespanha et al. [2007], Nešić and Liberzon [2009], Sailer and Wirth [2009], De Persis [2009].
First steps in this area considered communication constraints such as limited bandwidth or data rate, but issues as delays and packet loss were not treated. Also the communication channel was treated as static, De Persis and Isidori [2004], Wu and Chen [2007], Hespanha et al. [2007], Nešić and Liberzon [2009], whereas many realizations of communication channels use protocols which define internal dynamics of the channel. Examples of this are given by TCP and certain wireless protocols. In this paper we continue our work in Sailer and Wirth [2009] by extending the idea of dynamic quantization to the case of delays and packet loss. We also consider communication channels which are modelled as Markov chains.
The main result of the paper states that if the classic concept of dynamic quantization is complemented by time stamp information, then a sufficiently frequent average communication rate results in stabilization. The conditions are quite similar to previous results in this area. As an intermediate step we prove this result in a deterministic framework. This provides the basis for the proof of almost sure convergence in a Markovian setting.

The approach of dynamic quantization we are using here, was introduced by Brockett and Liberzon [2000] and has been extended to the nonlinear case in Liberzon and Hespanha [2005]. However, the notion of non static quantization is not new. It was first mentioned within the control community in the PhD thesis of Tatikonda [2000] or even earlier within the communication community (c.f., Goodman and Gersho [1974]).

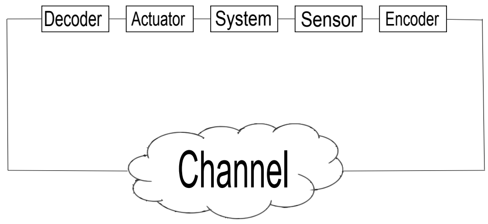

Fig. 1. The setup

We will explain the idea behind dynamic quantization with the help of Figure 2. At time t_1 we have a quantization region, where we know that the actual state of the system $x(t_1)$ lies within. This hypercube is centered at $\hat{x}_e(t_1)$ and its edges have the length $\ell(t_1)$. We divide this hypercube in N^n hypercubes, where n is the dimension of the state space and N is the number of partitions per dimension. Each of the smaller hypercubes has a length of ℓ/N. We will refer to those smaller hypercubes as subregions.
The sensor determines the actual subregion in which the state lies and calculates the center $x_e(t_1)$. We will refer to this process as encoding. The decoder on the other end of the channel has a copy of these values, i.e. the decoder knows the center and length of the region and the values n and N. Thus if we transmit the number of the subregion in which the state lies, the decoder is able to reconstruct the value $x_e(t_1)$.
If both encoder and decoder let the center x_e respectively x_d of the subregion follow the closed loop dynamics until time t_2, the error between the estimate x_e and the state x can grow by a certain factor. If we let the subregion grow by the same amount (the augmented region is the dashed box in Figure 2), we are sure that the state at time t_2 is

⋆ This work is supported by the German Science Foundation (DFG) within the priority programme 1305: Control Theory of Digitally Networked Dynamical Systems.

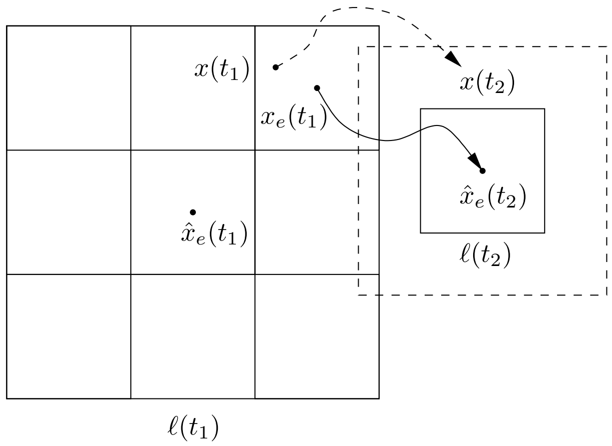

Fig. 2. Schematic representation of the dynamic quantizer

still within the subregion. This subregion becomes the new quantization region with the new center $\hat{x}_e(t_2) = x_e(t_2^-)$. Now we are in the same situation as we started, namely to know a hypercube in which the state lies and are able to repeat the same steps.
If the quotient between the growth of the quantization region and the reduction of the error due to N is smaller than 1, the quantization error converges to 0.

As sketched above it is important that encoder and decoder agree on certain values. This is easily achieved, if there is no delay in the channel, because the decoder can copy the behavior of the encoder exactly.
The case of fixed delay has been examined in De Persis [2007]. In order to cope with time varying delays we propose to send time information along with the encoded state. The encoder has to send the time information so the decoder knows the time when the state was encoded. As soon as the new encoded state is available to the decoder, it changes the control action. This time information has to be known by the encoder to copy the behavior of the state dynamics of the decoder.
We proceed as follows. In the ensuing Section 2.1 we collect the necessary notations and definitions. The communication channel is introduced in Section 2.2. In Section 2.3 we give a detailed description of the quantization scheme and of the corresponding dynamics of encoder and decoder. It is important to note that both encoder and decoder have identical internal models of the system. The important idea is to ensure that at certain time instances encoder and decoder are certain to have the same information about the state of their respective internal models.
In Section 3 we prove that with the encoding-decoding-scheme introduced in Section 2.3 it is possible to achieve asymptotic stability. We conclude our paper with some remarks in Section 4.

2. PRELIMINARIES

2.1 Notations

We use the following definitions. The symbol $|x| = \max\{|x_i| | 1 \le i \le n\}$ denotes the maximum norm on $\mathbb{R}^n$. The floor function $\lfloor\cdot\rfloor : \mathbb{R}^n \to \mathbb{R}^n,\ \ x \mapsto \lfloor x\rfloor$ is componentwise the biggest integer smaller or equal to x_i, $1 \le i \le n$. Similarly the ceiling function $\lceil\cdot\rceil : \mathbb{R}^n \to \mathbb{R}^n,\ \ x \mapsto \lceil x\rceil,\ \lceil x\rceil = -\lfloor -x\rfloor$. We introduce $r(t^-) := \lim_{t\nearrow t^-} r(t)$, if the limit exists. If a continuous function $\alpha : [0,\infty) \to [0,\infty)$ is strictly increasing and $\alpha(0) = 0$ then it is said to be of class $\mathcal{K}$. If α is also unbounded, we say it is of class $\mathcal{K}_\infty$. A function $\beta : [0,\infty)\times[0,\infty) \to [0,\infty)$ is said to be of class $\mathcal{KL}$ if $\beta(\cdot,t) \in \mathcal{K}$ for each fixed $t \ge 0$ and $\beta(r,t)$ decreases to 0 as $t \to \infty$ for each fixed $r \ge 0$.
We consider systems of the form

$$\dot{x} = f(x,u), \quad x \in \mathbb{R}^n, \quad u \in U \subset \mathbb{R}^m\,, \tag{1}$$

where $f : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$ is continuous and Lipschitz in the first component uniformly with respect to u, i.e.,

$$|f(x,u) - f(y,u)| \le L|x-y|,\ \forall x,y \in \mathbb{R}^n\,, \forall u \in U\,. \tag{2}$$

Remark 2.1. Because of the stochastic nature of Theorem 3.6 we cannot bound the region in which the state will remain *a priori*. Therefore we assume global Lipschitz continuity.
We expect that an extension to the case of local Lipschitz continuity is possible by using packet drop information, which determines bounds on the trajectories.
However, for our deterministic result (Theorem 3.1) it would be sufficient to know a local Lipschitz constant as discussed in Sailer and Wirth [2009].

Assumption 2.2. There exists a smooth $k : \mathbb{R}^n \to \mathbb{R}^m$, $x \mapsto k(x)$ with $k(0) = 0$ such that

$$\dot{x} = f(x, k(x + e_d(t))) \tag{3}$$

is ISS with respect to the measurement error e_d. Note that this is equivalent to the existence of functions $\beta \in \mathcal{KL}$ and $\gamma \in \mathcal{K}_\infty$ so that the solutions of (3) satisfies

$$|x(t)| \le \beta(|x(t_0)|, t-t_0) + \gamma\left(\sup_{s\in[t_0,t]} |e_d(s)|\right)\ \forall t \ge t_0. \tag{4}$$

For a comprehensive introduction and the precise definition of ISS see e.g., Sontag [2008].
The next section is concerned with the properties of the communication channel.

2.2 The Communication Between Encoder and Decoder

We consider TCP like packet based transmissions over a noiseless, errorfree channel with delay and packet loss.
The encoder encodes the state and sends a symbol from a finite alphabet to the decoder together with the time when the state was encoded (time stamping). As soon as a packet arrives at the decoder, it reconstructs the encoded state and sends an acknowledgment (ack) back to the encoder. If this ack arrives at the encoder or a predefined time elapses without receiving one, it repeats the encoding. Denote by t_k the kth time instance the encoder received an ack. The time when the kth information sent by the encoder is received by the decoder is denoted by t_k^*. Note that we assume that there is no time delay between the arrival of an information and the sending of the next packet i.e. t_k and t_k^* are also the time instances when the encoder sends information and the decoder sends an ack respectively.

Assumption 2.3. There exists a long time average of the difference between $t_k - t_{k-1}$. This average is given by

$$\tau^* = \limsup_{k\to\infty} \frac{1}{k}\sum_{j=2}^{k}(t_j - t_{j-1}) = \limsup_{k\to\infty}\frac{1}{k}t_k. \tag{5}$$

Remark 2.4. Assumption 2.3 states that on average on an infinite time horizon, every τ^* units of time a packet will be successfully acknowledged.
From the relation between t_k and t_k^* it can be seen that the long time average of the difference $t_{k+1}^* - t_k^*$ is also equal to τ^*.

Assumption 2.5. For the communication channel the following should hold:

(1) All packets are time stamped with the current time they are sent
(2) Only packets sent from encoder to decoder are lost
(3) There exists a minimal delay from encoder to decoder, given by τ_{min}, i.e., $t_k^* - t_k \geq \tau_{min}$ and $t_k - t_{k-1}^* \geq \tau_{min}$
(4) The channel is able to transmit packets containing a value from a set of N^n (N odd) discrete values within τ_{min} units of time
(5) If τ_{max} time elapses without receiving an ack, the packet sent last time is considered lost and a new packet will be sent

Remark 2.6. By (1) we have to send the actual time together with the encoded state information. It is not reasonable to be able to transmit the state information quantized and the time information not. For the sake of simplicity we omit details on time quantization, see Sailer and Wirth [2009] for a discussion.
Assumption 2.5 (2) is a major restriction on the channel used. But because the ack's are much smaller than the state information the decoder could send many ack's to ensure that at least one arrives at the encoder. Without this assumption we could not guarantee that the encoder and the decoder agree on their states.
Assumption 2.5 (3) is in general not a restrictive one. In every real communication channel such a minimal delay exists.
Assumption 2.5 (4) states that the bandwith of the channel B must be large enough to transmit the state information within τ_{min} units of time. For instance, if binary encoding is used we require

$$B \geq \frac{n \log_2 N}{\tau_{min}} \,. \tag{6}$$

If this condition is not met, the decoder could introduce an artificial delay by waiting to ensure that τ_{min} is large enough to fulfill the bandwith constraint.
The choice for N to be odd guarantees that the center of the quantization region lies in the interior of one of the subregions. Although this is not needed in general, it eases the presentation.

Remark 2.7. By item (5) of Assumption 2.5 and the remarks on τ_{min}, the values τ_{min} and τ_{max} may be regarded as design parameters. While choosing τ_{min} too small, can violate the bandwith constraint, a larger value can degrade the performance of the overall system. Similar statements hold for τ_{max}. If τ_{max} is chosen too small, no ack will arrive at the encoder before a new packet will be sent. And again choosing τ_{max} too large may have a negative effect on the performance.

Assumption 2.8. Clocks of encoder and decoder are synchronized and the time $t_1 = 0$ when the encoder sends the first packet is known by the encoder and the decoder.

Assumption 2.9. Both the encoder and the decoder know the same bound of the initial state of the system (i.e., encoder and decoder know the same $X \in \mathbb{R}$ s.t. $|x(0)| \leq X$).

2.3 The Equations for Encoder and Decoder

The initial states for the encoder and the decoder are:
$k = 1,\ t_0 = 0,\ t_0^* = 0,\ t_1 = 0\ t_1^* = 0$ and $\hat{x}_d(0) = \hat{x}_e(0) = 0$
$x_e(0^-) = x_d(0^-) = 0$ and $\ell_e(0^-) = \ell_d(0^-) = 2X$.
The Encoder equations read:
(i) Every time an ack arrives at the encoder ($t = t_k$)

$$t_s = t \tag{7}$$

$$\ell_e(t_k) = \ell_e(t_{k-1}) e^{L(t_k - t_{k-1})}/N \tag{8}$$

$$x_e(t_k^-) = x_e(t_{k-1}) + \int_{t_{k-1}}^{t_{k-1}^*} f(x_e(s), k(\hat{x}_e(s)))ds + \int_{t_{k-1}^*}^{t_k} f(x_e(s), k(x_e(s)))ds \tag{9}$$

$$s(t_s) = \varphi(x_e(t_k^-), x(t_k), \ell_e(t_k)) \tag{10}$$

$$x_e(t_k) = x_e(t_k^-) + s(t_s)\frac{\ell_e(t_k)}{N} \tag{11}$$

$$\hat{x}_e(t_k) = x_e(t_k^-) \tag{12}$$

Every time the encoder receives an ack ($t = t_k$) it updates the length of the quantization region according to the growth of the error on the last interval (8). The center of the quantization region is updated via (9). Both integrals are needed to account for the change in the control action on the decoder side. The subregion in which the state lies is calculated by (10). This information will be sent to the decoder together with the actual time (7). The jump from the center to the subregion is done by equation (11). The value of the old quantization region is copied by (12).
(ii) If τ_{max} time without an ack elapse

$$t_s = t \tag{13}$$

$$\ell_e(t) = \ell_e(t_k) e^{L(t - t_k)} \tag{14}$$

$$x_e(t^-) = \hat{x}_e(t) \tag{15}$$

$$s(t_s) = \varphi(x_e(t^-), x(t), \ell_e(t)) \tag{16}$$

$$x_e(t) = x_e(t^-) + s(t_s)\frac{\ell_e(t)}{N} \tag{17}$$

If τ_{max} units of time elapse without receiving an ack, the packet sent last time is considered lost and an new one will be sent. Similar to the case of no loss, the encoder updates the length of the quantization region (14). Note that there is no division by N. Equation (15) cancels the jump from the center to the subregion made in the last encoding step. The equations (16) and (17) follow the same reason as in the case of no loss. In both cases ((i) and (ii)) t_s and $s(t_s)$ are auxiliary variables, describing the data payload of the packets sent from encoder to decoder.
(iii) Otherwise:

$$\dot{\hat{x}}_e(t) = f(\hat{x}_e(t), k(\hat{x}_e(t))) \tag{18}$$

We need (18) to know the trajectory which will be used to close the loop on the decoder side as can be seen from Lemma 3.3 and 3.4. It is also needed to treat the case of packet loss (15).
The decoder equations read:
(i) Every time a packet arrives at the decoder ($t = t_k^*$)

$$\ell_d(t_s) = \ell_d(t_{k-1})e^{L(t_s - t_{k-1})}/N \quad (19)$$

$$x_d(t_s^-) = \hat{x}_d(t_{k-1}^*) + \int_{t_{k-1}^*}^{t_s} f(x_d(s), k(x_d(s)))ds \quad (20)$$

$$x_d(t_s) = x_d(t_s^-) + s(t_s)\frac{\ell_d(t_s)}{N} \quad (21)$$

$$\hat{x}_d(t_k^*) = x_d(t_s) + \int_{t_s}^{t_k^*} f(x_d(s), k(\hat{x}_d(s)))ds \quad (22)$$

(ii) Otherwise

$$\dot{\hat{x}}_d(t) = f(\hat{x}_d(t), k(\hat{x}_d(t))) \quad (23)$$

The decoder copies the behavior of the encoder with the help of (19)-(21). Equation (22) compensates for the delay between encoder and decoder.
A sketch of the evolution of the different trajectories is depicted in Figure 3.
The function φ calculates the subregion in which the state

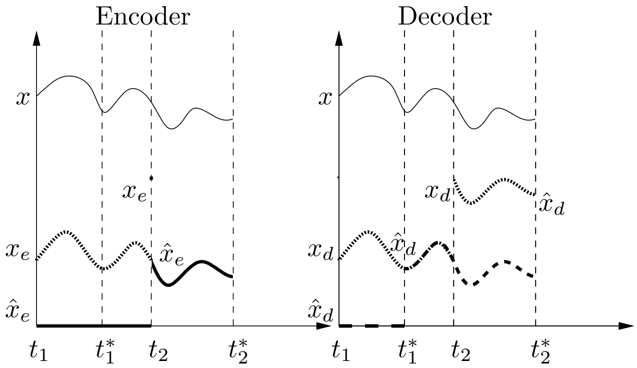

Fig. 3. Snapshot of the different trajectories at time t_2^*

lies at each encoding step.

$$\varphi(x_e, x, \ell) = \left\lfloor \frac{N}{\ell}(x - x_e) + \frac{1}{2} \right\rfloor . \quad (24)$$

If at time t the state lies within the quantization region, the error between the state and the estimate shrinks by N because of the jump from the center of the region to the center of a subregion. Hence

$$|x(t) - x_e(t^-)| \leq \frac{\ell}{2} \Rightarrow |x(t) - x_e(t)| \leq \frac{\ell}{2N} \quad (25)$$

holds, which can be seen from (24) and (11). Overall, the evolution of the closed loop system is given by

$$\dot{x}(t) = f(x(t), k(\hat{x}_d(t))) . \quad (26)$$

In the ensuing section we will give conditions under which this scheme results in stabilization.

3. MAIN RESULT

3.1 Deterministic Result

The proofs given in this section are similar to those from Sailer and Wirth [2009]. The main difference is that we consider here a long time average for the delay as opposed to a worst case scenario as in Sailer and Wirth [2009]. To let this paper stand for its own, the whole proofs are given.

Theorem 3.1. Consider system (1) with encoder/decoder scheme described in (7)-(23) and let Assumptions 2.2, 2.3, 2.5, 2.8 and 2.9 hold. If

$$N > e^{L\tau^*} \quad (27)$$

then $u = k(\hat{x}_d)$, where $\hat{x}_d$ is generated by the decoder (23), asymptotically stabilizes the equilibrium $x^* = 0$ of (1).

Remark 3.2. By putting together condition (6) and (27), we get

$$B \geq \frac{n \log_2(e^{L\tau^*} + 2)}{\tau_{min}} .$$

This condition on the bandwidth (with binary encoding) ensures the existence of an odd number N which fullfils both (6) and (27).

To achieve asymptotic stability of the closed loop system we have to make sure that the error between the state of the system and the value which will be used to close the loop converges to zero (Corollary 3.5). To this end we have to ensure the convergence of the encoder state to the state of the system (Lemma 3.4). To bound the growth of the error between encoder and system, we have to make sure that they close their loops with the same signal (Lemma 3.3).
For the ensuing lemma it is important to recall that all the states of encoder and decoder (with the exception of $\hat{x}_d$) are only calculated at discrete time instances. Nevertheless we need a continuous version of the evolution of the encoder state given by

$$\dot{x}_e(t) = \begin{cases} f(x_e(t), k(\hat{x}_e(t))) & t \in [t_k, t_k^*) \\ f(x_e(t), k(x_e(t))) & t \in [t_k^*, t_{k+1}) \end{cases} . \quad (28)$$

Note that even if packet loss occurred, (28) is valid, because every time a loss happened the encoder undoes the jump made in the last encoding step via equations (15) and (14).

Lemma 3.3. Consider encoder/decoder scheme described in (7)-(23). If Assumptions 2.2, 2.3, 2.5, 2.8 and 2.9 hold, then $\hat{x}_d(t) = \hat{x}_e(t)$ for all $t \in [t_k, t_k^*)$ and for all $t \in [t_k^*, t_{k+1})$ we have $\hat{x}_d(t) = x_e(t)$.

Proof. As soon as a packet arrives at the decoder, it knows the time when the state was encoded due to the time stamping. Hence the decoder can use (19) to reconstruct the length of the quantization region used to encode the state (8) respectively (14). Hence it holds

$$\ell_e(t_s) = \ell_d(t_s). \quad (29)$$

Because of the initial condition of the encoder and the decoder and (12) it holds that $\hat{x}_d(0) = \hat{x}_e(0) = 0$. Using $t_1 = 0$ and equations (18) and (23) we obtain

$$\hat{x}_d(t) = \hat{x}_e(t) \quad \forall t \in [t_1, t_1^*). \quad (30)$$

At time t_1^* the value $s(t_s)$ as well as the time t_s becomes available to the decoder. Because of the initialization of encoder and decoder and (11) respectively (21) and (29) it holds that $x_e(t_1) = x_d(t_1)$. By (22), (28) and (30) we have $\hat{x}_d(t_1^*) = x_e(t_1^*)$.
Since both trajectories follow the same dynamics on the interval $[t_k^*, t_{k+1})$ by (23) and (28) we get $\hat{x}_d(t) = x_e(t), \quad \forall t \in [t_1^*, t_2)$.
Due to the continuity of $\hat{x}_d$ at t_k and (12) $\hat{x}_d(t_2) = \hat{x}_e(t_2)$ holds. From (9), (20) and (22) as well as (30) we can deduce

$$x_d(t_2^-) = x_e(t_2^-).$$

Now we can use (11) respectively (21) and (29) to get $x_d(t_2) = x_e(t_2)$. To conclude the proof, repeat the arguments inductively.

Define $e_e(t) := x_e(t) - x(t)$ as the error between the encoded state and the state of the system. Now that we know when certain signals on the encoder side coincide with the corresponding signal on the decoder side, we can bound the error $e_e(t)$ with the help of the next lemma.

Lemma 3.4. Consider system (1) with encoder/decoder scheme described in (7)-(23). If Assumptions 2.2, 2.3, 2.5, 2.8, 2.9 and condition (27) hold, then $\lim_{k\to\infty} |e_e(t_k)| = 0$.

Proof. Because of Assumption 2.9 and initialization of the encoder, the initial state $x(t_1)$ is within the quantization region.

$$|0 - x(t_1)| = |e_e(t_1^-)| \leq X = \frac{\ell(t_1^-)}{2}.$$

Hence we can use (25) to obtain

$$|e_e(t_1)| \leq \frac{\ell(t_1^-)}{2N} \ . \tag{31}$$

Let $a \vee b := \max\{a, b\}$ and $a \wedge b := \min\{a, b\}$. The encoder error e_e satisfies for $t \in [t_{k-1}, t_k)$ according to (28) and (26):

$$|e_e(t)| = |x_e(t) - x(t)| = |x_e(t_{k-1}) +$$
$$\int_{t_{k-1}}^{t_{k-1}^* \wedge t} f(x_e(s), k(\hat{x}_e(s)))ds + \int_{t_{k-1}^*}^{t \vee t_{k-1}^*} f(x_e(s), k(x_e(s)))ds$$
$$-x(t_{k-1}) - \int_{t_{k-1}}^{t} f(x(s), k(\hat{x}_d(s)))ds|.$$

We can split the last integral and collect the corresponding terms to get:

$$|e_e(t)| = |x_e(t_{k-1}) - x(t_{k-1}) +$$
$$\int_{t_{k-1}}^{t_{k-1}^* \wedge t} (f(x_e(s), k(\hat{x}_e(s))) - f(x(s), k(\hat{x}_d(s))))ds$$
$$+ \int_{t_{k-1}^*}^{t \vee t_{k-1}^*} (f(x_e(s), k(x_e(s))) - f(x(s), k(\hat{x}_d(s))))ds|.$$

Now we can use (2) and Lemma 3.3 to arrive at:

$$|e_e(t)| \leq |e_e(t_{k-1})| + L \int_{t_{k-1}}^{t} |e_e(s)|ds.$$

Because of the continuity of $e_e(t)$ on the interval $t \in [t_{k-1}, t_k)$ the Gronwall-lemma yields

$$|e_e(t_k^-)| \leq |e_e(t_{k-1})| e^{L(t_k - t_{k-1})}.$$

Using (25) again gives

$$|e_e(t_k)| \leq |e_e(t_{k-1})| e^{L(t_k - t_{k-1})}/N.$$

Starting with (31) and repeating the previous arguments yields

$$\lim_{k\to\infty} |e_e(t_k)| \leq \lim_{k\to\infty} |e_e(t_1)| e^{L(t_k - t_1)}/N^k. \tag{32}$$

With the help of (5) we get

$$\lim_{k\to\infty} |e_e(t_k)| \leq X \lim_{k\to\infty} \left(\frac{e^{L\tau^*}}{N}\right)^k = 0 \ ,$$

where the convergence is guaranteed by condition (27) and the proof is complete.

With the help of the bound on the error on the encoder side, we can bound the error on the decoder side as well. To this end define $e_d(t) := \hat{x}_d(t) - x(t)$ as the error between the trajectory used to close the loop and the state of the system.

Corollary 3.5. Consider system (1) with encoder/decoder scheme described in (7)-(23). Let the premise of Lemma 3.4 hold, then $\lim_{k\to\infty} |e_d(t_k^*)| = 0$.

Proof. Using Lemma 3.3 we are able to conclude

$$|e_d(t_k^*)| = |e_e(t_k^*)|.$$

Using the Gronwall Lemma as in the proof of Lemma 3.4, the error evolves according to:

$$|e_d(t_k^*)| = |e_e(t_k^*)| \leq |e_e(t_k)| e^{L(t_k^* - t_k)}.$$

Applying limits on both sides and considering (32), we get

$$\lim_{k\to\infty} |e_d(t_k^*)| \leq |e_e(t_1)| \lim_{k\to\infty} e^{L(t_k^* - t_k)} e^{L(t_k - t_1)}/N^k \ .$$

From (5) we can deduce

$$\lim_{k\to\infty} |e_d(t_k^*)| \leq |e_e(t_1)| \lim_{k\to\infty} e^{L(t_k^* - t_1)}/N^k \tag{33}$$
$$\leq |e_e(t_1)| \lim_{k\to\infty} \left(\frac{e^{L\tau^*}}{N}\right)^k e^{L(t_1^* - t_1)} = 0 \ ,$$

where the last equality follows from condition (27).

The proof of our main Theorem 3.1 is now an easy consequence of the ISS Assumption 2.2.

Proof. (of Theorem 3.1) By Corollary 3.5 and the continuity of $e_d(t)$ on the interval $[t_k^*, t_{k+1}^*)$ the bound $W := \sup_{t\geq 0} |e_d(t)| < \infty$ exists. From the bound of the initial value X and the maximal error on the decoder side W we obtain using (4) that

$$|x(t)| \leq \beta(X, 0) + \gamma(W) =: E \quad \forall t \geq 0.$$

Using (4) again we get

$$|x(t)| \leq \beta(E, t - t_0) + \gamma\left(\sup_{s\in[t_0,t]} |e_d(s)|\right) \quad \forall t \geq t_0.$$

As t_0 goes to infinity the right hand side converges to zero which shows the attractivity of $x^* = 0$. On the other hand we can use (33) to interpret (4) as:

$$|x(t)| \leq \beta(|x(0)|, 0) + \gamma(\sup_{k\in\mathbb{N}} |x_e(0) - x(0)| e^{L(t_k^* - t_1)}/N^k).$$

The existence of the supremum is guaranteed by Corollary 3.5. Hence the right hand side can be chosen arbitrarily small by choosing $|x(0)|$ small, which together with the attractivity concludes the proof.

3.2 Markovian Communication Models

In this section we derive conditions for the stabilization of the system given in (1) under the condition that the communication channel can be described by a (time-homogeneous) Markov process. The model of system, encoding and decoding will remain the same. Only the process of delays and packet loss will have additional assumptions. The approach of modeling the communication channel in a Markovian way is justified by the fact that in many cases this yields appropriate models for the dynamical behavior of the channel, as can be seen for example in Wirth et al. [2006] and Shorten et al. [2007] for TCP and in Bianchi [2000] for the wireless case.

In the following we assume as given a communication channel in which external perturbations such as average load of other users is stationary. We consider a Markov chain $\{X(k)\}_{k\in\mathbb{N}}$, where the state X lies in a state space of the chain S. This state space would be specified by concrete situations. As we are considering TCP like transmissions over a digital channel, we can only send information at discrete time instances, justifying a Markov process which is discrete in time.

We assume as given two continuous maps describing the communication, namely

$$T : S \to [T_{min}, \infty)\,, \quad g : S \to \mathbb{N}\,, \tag{34}$$

where $T(X)$ denotes the length of the next communication interval depending on the state X of the channel and $g(X)$ denotes the number of bits that can be sent in that interval. Thus if communication starts at a time $t_1 \in \mathbb{R}$ and $l := n \log_2 N + b$ bits have to be sent, where b is the acknowledgment size in bits, we define a stopping time of the Markov chain by

$$k_1^* = \min \left\{ k \;\middle|\; \sum_{j=0}^{k} g(X(j)) \geq l \text{ or } \sum_{j=0}^{k} T(X(j)) \geq \tau_{max} \right\}$$

and until k_1^*

$$\sum_{j=0}^{k_1^*} T(X(j)) =: \tau(1)$$

units of time elapse. If $\tau(1) \geq \tau_{max}$, we consider the information to be lost. To ensure the Markovian property of our description, we assume that $X(k)$ has the strong Markov property, i.e. the evolution of the process only depends on the state of the chain at the stopping time k^*. The next time we want to send information, i.e. at time $t_1 + \tau(1)$ we define the next stopping time k_2^* by

$$k_2^* = \min \left\{ k \;\middle|\; \sum_{j=k_1^*}^{k} g(X(j)) \geq l \text{ or } \sum_{j=k_1^*}^{k} T(X(j)) \geq \tau_{max} \right\}.$$

And the duration from k_1^* until k_2^* by $\sum_{j=k_1^*}^{k_2^*} T(X(j)) =: \tau(2)$ If we repeat this procedure a sequence of time instances is given by $\{\tau(j)\}_{j\in\mathbb{N}}$.

We now assume ergodicity of the Markov chain, which ensures that almost surely

$$\tau_M^* := \lim_{k\to\infty} \frac{1}{k} \sum_{j=0}^{k} \tau(j) \mathbb{1}\{\tau(j) < \tau_{max}\} \tag{35}$$

exists, where $\mathbb{1}$ denotes the indicator function.

With the above considerations we are able to state the next theorem.

Theorem 3.6. Consider system (1) with encoder/decoder scheme described in (7)-(23) and let Assumptions 2.2, 2.5, 2.8 and 2.9 hold. Let an ergodic Markov process X with stopping times as in (3.2) exist with T and g given in (34). If

$$N > e^{L\tau_M^*}\,, \tag{36}$$

where τ_M^* is denoted by (35), then $u = k(\hat{x}_d)$, where $\hat{x}_d$ is generated by the decoder (23), asymptotically stabilizes the equilibrium $x^* = 0$ of (1) almost surely.

Proof. The proof goes along the lines of the proof of Theorem 3.1. Only the condition (27) has to be exchanged with condition (36).

4. CONCLUSION

In this note we have presented an extension to Sailer and Wirth [2009] to a more sophisticated model for delay and packet loss. By considering a long time average rather than a maximal delay, we could give in general better conditions for the communication channel. In the case of deterministic delays we had to assume the existence of such a long time average. For the case that the communication channel is modeled by a Markov process, we have given conditions under which such an average almost surely exists.

REFERENCES

G. Bianchi. Performance analysis of the IEEE 802.11 distributed coordination function. *IEEE Journal on selected areas in communication*, 18(3):535–547, 2000.

R. W. Brockett and D. Liberzon. Quantized feedback stabilization of linear systems. *IEEE Trans. Autom. Control*, 45(7):1279–1289, 2000.

C. De Persis. Minimal data rate stabilization of nonlinear systems over networks with large delays. In *Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks and Workshops, 2007. WiOpt 2007. 5th International Symposium on*, pages 1–9, 2007.

C. De Persis. Robustness of quantized continuous-time nonlinear systems to encoder/decoder mismatch. In *Proc. 48th IEEE Conf. Decision and Control, CDC 2009*, pages 13–18, Shanghai, China, 2009.

C. De Persis and A. Isidori. Stabilizability by state feedback implies stabilizability by encoded state feedback. *Systems & Control Letters*, 53(3-4):249–258, 2004.

D. Goodman and A. Gersho. Theory of an adaptive quantizer. *IEEE Trans. on Comm.*, 22(8):1037–1045, 1974.

J. P. Hespanha, P. Naghshtabrizi, and Y. Xu. A survey of recent results in networked control systems. *Proc. of IEEE* Special Issue on Technology of Networked Control Systems, 95(1):138–162, Jan. 2007.

D. Liberzon and J.P. Hespanha. Stabilization of nonlinear systems with limited information feedback. *IEEE Trans. Autom. Control*, 50(6):910–915, 2005.

D. Nešić and D. Liberzon. A unified framework for design and analysis of networked and quantized control systems. *IEEE Trans. Autom. Control*, 54(4):732–747, 2009.

R. Sailer and F. Wirth. Stabilization of nonlinear systems with delayed data-rate-limited feedback. In *Proc. 10th European Control Conf. 2009, Budapest, Hungary*, pages 1734–1739, 2009.

R. Shorten, C. King, F. Wirth, and D. Leith. Modelling TCP congestion control dynamics in drop-tail environments. *Automatica*, 43(3):441–449, 2007.

E. D. Sontag. Input to state stability: Basic concepts and results. In *Nonlinear and Optimal Control Theory*, volume 1932 of *Lecture Notes in Mathematics*, pages 163–220. Springer Berlin/Heidelberg, 2008.

S. C. Tatikonda. *Control Under Communication Constraints.* PhD thesis, Department of EECS, Masachussets Institute of Technology, Cambridge, MA, 2000.

F. Wirth, R. Stanojevic, R. Shorten, and D. Leith. Stochastic equilibria of AIMD communication networks. *SIAM Journal on Matrix Analysis and Applications*, 28 (3):703–723, 2006.

J. Wu and T. Chen. Design of networked control systems with packet dropouts. *IEEE Trans. Autom. Control*, 52 (7):1314–1319, 2007.