2024 European Control Conference (ECC)
June 25-28, 2024. Stockholm, Sweden

# Delay Attack and Detection in Feedback Linearized Control Systems

Torbjörn Wigren and André M. H. Teixeira

*Abstract*— **Delay injection attacks on nonlinear control systems may trigger instability mechanisms like finite escape time dynamics. The paper guards against such attacks by showing how a recursive algorithm for identification of nonlinear dynamics and delay can simultaneously provide parameter estimates for controller tuning and detection of delay injection in the feedback path. The attack methodology is illustrated using a simulated feedback linearized automotive cruise controller where the attack is disguised, but anyway rapidly detected.**

## I. Introduction

The paper addresses cyber attacks by injecting delay into nonlinear feedback control systems. Due to the nonlinear system dynamics and the nonlinear controller, destabilization mechanisms with delay become more intricate than for linear systems [10], [12], [23]. To counter such attacks, the paper proposes joint recursive identification of the delay and the nonlinear system dynamics. Identification of the nonlinear dynamics is needed for accurate estimation of the delay if model-free [3] or adaptive [4] controller design is applied.

The deployment of networked feedback control systems with wireless delay increases the need for improved methods for attack detection and mitigation, [1], [8]. Systems under delay injection attack have been studied previously in [5], [13], [29], [30], where the latter discussed delay attack strategies, and proposed recursive identification of delay and dynamics for defense of linear servo and regulator feedback systems. Robust controller design as a means for mitigation of delay attacks was studied in [20], [21]. However, the understanding on how to rapidly detect disguised delay injection attacks on nonlinear control systems with unknown or partially known dynamics remains limited.

Identification of delay in linear dynamic systems has been extensively studied, see e.g. [6], [11] and [14] for examples of frequency and time-domain methods. For nonlinear systems series expansions [7], or sequential Monte-Carlo methods, [24], may be applied for joint identification of delay and nonlinear dynamics. However, except for the bootstrap particle filter these methods are mostly based on batch processing, while a rapid identification of delay changes would be better served by recursive identification [15].

The main contribution of the paper demonstrates that joint recursive identification of delay and *nonlinear* dynamics can efficiently detect a delay injection attack in a nonlinear feedback regulator loop, despite the fact that the attack is almost perfectly disguised in open-loop. The algorithm applied is of output error type [26], which is a method known to perform well in linear cases [6]. Additional contributions isolate new destabilization mechanisms that may be used by a delay injection attacker, and provides numerical illustrations for nonlinear cruise control dynamics, [2], [9], [16], [31].

The organization is as follows. The control system is described in Section II. Delay attacks on nonlinear systems in general are discussed in Section III. The recursive algorithm used for delay attack detection is reviewed in Section IV, while Section V illustrates a disguised delay injection attack and its detection for a feedback linearized automotive cruise controller. Conclusions follow in Section VI.

## II. Control System Description

To study delay injection attacks the following nonlinear system is considered

$$\dot{\mathbf{x}}(t) = \mathbf{f}_{\boldsymbol{\theta}_S}(\mathbf{x}(t)) + \mathbf{g}_{\boldsymbol{\theta}_S}(\mathbf{x}(t)))\mathbf{u}(t), \tag{1}$$

$$\mathbf{y}(t) = \mathbf{h}(\mathbf{x}(t)). \tag{2}$$

Here the state vector $\mathbf{x}(t)$, the input signal vector $\mathbf{u}(t)$ and the output signal vector $\mathbf{y}(t)$ are given by

$$\mathbf{x}(t) = \left(x_1(t) \ \ ... \ \ x_n(t)\right)^T, \tag{3}$$

$$\mathbf{u}(t) = \left(u_1(t) \ \ ... \ \ u_K(t)\right)^T, \tag{4}$$

$$\mathbf{y}(t) = \left(y_1(t) \ \ ... \ \ y_L\right)^T. \tag{5}$$

The parameter vector of the dynamics, $\boldsymbol{\theta}_S$, is defined in detail in Section IV. Here it is assumed to be fixed and for example obtained after an initial identification run. The parameter dependence of $\mathbf{f}$ and $\mathbf{g}$ is therefore indicated by a subscript. In (1) and (2), $\mathbf{f}(\cdot)$, $\mathbf{g}(\cdot)$ and $\mathbf{h}(\cdot)$ are nonlinear vector functions and the vectors (3), (4) and (5) are indexed by the subscripts $i, k, l$, respectively. $^T$ denotes transpose.

In the present paper, the discussion is limited to the following standard input-state feedback linearization method:

*Lemma 1 ([12], Definition 12.1)*: Assume that the nonlinear system (1) with $\mathbf{f}_{\boldsymbol{\theta}_S} : D_{\mathbf{x}} \to R^n$ and $\mathbf{g}_{\boldsymbol{\theta}_S} : D_{\mathbf{x}} \to R^{n\times K}$ has continuous derivatives of sufficient order on $D_{\mathbf{x}} \subset R^n$. The system is then input-state linearizable if there exists a diffeomorphism $\mathbf{T}_{\boldsymbol{\theta}_S} : D_{\mathbf{x}} \to R^n$ such that $D_{\mathbf{z}} = \mathbf{T}_{\boldsymbol{\theta}_S}(D_{\mathbf{x}})$ contains the origin and the change of variables $\mathbf{z} = \mathbf{T}_{\boldsymbol{\theta}_S}(\mathbf{x})$ transforms (1) to

$$\dot{\mathbf{z}}(t) = \mathbf{A}_{\boldsymbol{\theta}_S}\mathbf{z}(t) + \mathbf{B}_{\boldsymbol{\theta}_S}\mathbf{b}_{\boldsymbol{\theta}_S}^{-1}(\mathbf{x}(t))\left(\mathbf{u}(t) - \mathbf{a}_{\boldsymbol{\theta}_S}(\mathbf{x}(t))\right),$$

$(\mathbf{A}_{\boldsymbol{\theta}_S}, \mathbf{B}_{\boldsymbol{\theta}_S})$ controllable and $\mathbf{b}_{\boldsymbol{\theta}_S}(\mathbf{x})$ nonsingular, for $\mathbf{x} \in D_{\mathbf{x}}$. □

Torbjörn Wigren and André Teixeira are with the Department of Information Technology, Uppsala University, Box 337, SE-75105 Uppsala, Sweden. { torbjorn.wigren,andre.teixeira } @it.uu.se. This work is supported by the Swedish Research Council under the grant 2021-06316 and by the Swedish Foundation for Strategic Research.

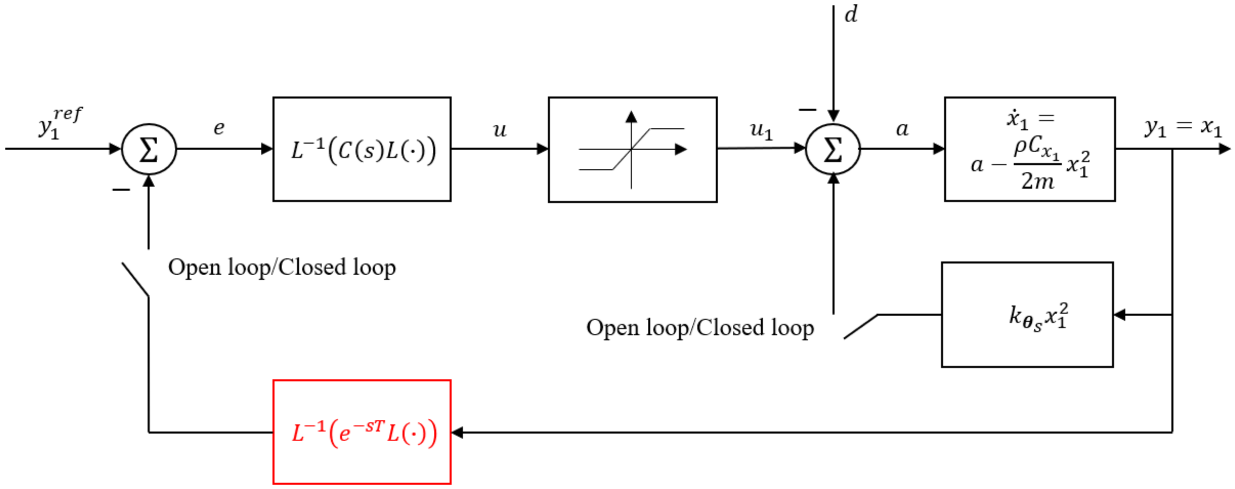

Fig. 1. Block diagram of the input-state feedback linearization of the automotive cruise control feedback loop treated in Section V. All variables are defined in Section V. Note that the outer controller handles the additive disturbance $d(t)$, and the control signal limitation, and that it may or may not depend on $\boldsymbol{\theta}_S$. The operator $L(\cdot)$ denotes Laplace-transformation. The block diagram appears in continuous time, with the delay injection attack red.

Lemma 1 handles multi-dimensional additive and multiplicative nonlinearities and provides a method of cancellation by combined multi-dimensional subtraction and division. As an example, the block diagram of Fig. 1 applies subtraction of the quadratic air resistance nonlinearity in the automotive cruise control example of Section V. As can be seen, a special feature of the treated control problem of the paper is the *selection* between open-loop and closed-loop control that is assumed to be either under manual control, or to be controlled by external logic. Note that two switches with *simultaneous* operation are needed in Fig. 1, one for the feedback linearization and one for the outer feedback loop.

## III. Delay Attack

### A. Attack Objectives

The objectives when attacking a regulator loop include response time violations, for example disturbing cascaded production stations. An immediate attack mechanism works best in the forward path from controller to system dynamics. A delayed and hidden attack can be implemented with a feedback path delay injection attack as shown in Fig. 1. That attack would not have any effect on the response time until the loop is closed.

Destabilization is the ultimate attack objective. For linear systems the destabilization is obtained for large enough delay, provided that the gain of the feedback loop is large enough as implied by the Nyquist criterion [23]. This follows since the delayed loop gain $\hat{g}_T(j\omega)$ suffers from an associated phase loss, as in

$$\hat{g}_T(j\omega) = e^{-j\omega T}\hat{g}(j\omega), \tag{6}$$

where $\omega$ is the angular frequency, $T$ is the injected delay and $\hat{g}(j\omega)$ is the loop gain of the linear system.

### B. Nonlinear Destabilization

For nonlinear systems the destabilizing mechanism may be more involved. For example, assume that the input-state linearization of Fig. 1 is not perfectly tuned, so that the forward path contains a term proportional to the square of the velocity, with $k_{\boldsymbol{\theta}_S} - \frac{\rho C_{x_1}}{2m} > 0$. When the driver switches to closed loop cruise control and if the injected delay is large enough, the input-state linearization loop has finite escape time dynamics during a time $T$ until the outer loop feedback takes effect. In the example of Section V, the parameters and attack delay are such that this does not occur, but the possibility may be important for other systems with very fast dynamics. This effect motivates why further research on destabilization mechanisms for non-linear systems caused by injected delay is of central importance to maintain integrity of critical feedback control loops.

### C. Delay Injection

Manipulation of time stamps can be used to tamper with control and/or feedback signals. For example, the attacker could decrease the values of the time stamps of the control signal so that they appear to arrive earlier than at the nominal arrival time for which the controller has been designed. If the manipulation passes undetected, the system may delay control actions, thereby reducing the stability margin. Similar ideas could be applied in the feedback path.

A physical delay may be more difficult to implement since queues may need to be created in the attacked software.

### D. Disguised Attack

In case of a physical delay injection attack, e.g. by addition of queuing delay, time stamps may reveal the attack. Therefore, physical delay injection needs to be accompanied by time stamp manipulation that has the potential to disguise the attack, both in the forward path and in the feedback path.

Another approach could be to hide the injected average delay by jitter, provided that jitter is normally present in the system as in [29].

For systems that can be operated both in open and closed loop it is advantageous to perform the delay injection in the feedback path, as close to the controller as possible [30]. The reason is that there will be no effect in the forward path that could be sensed by a human operator or driver, as a perceived slow system response. The selection of the feedback path for delay injection therefore disguises the attack perfectly from being perceived in the forward path. This situation persists until the feedback loop is closed at which point the attack takes immediate and full effect.

## IV. Nonlinear Recursive Delay Detection

This section reviews the algorithm of [26] that is applied for joint recursive identification of delay and nonlinear dynamics in the paper. The convergence is analysed in [28], where it is shown that the correct parameter vector is in the set towards which the algorithm converges globally.

### A. Nonlinear model with Delay

The nonlinear model is an ordinary differential equation (ODE) in state space form, with a single delay. Since the ODE is selected to be time invariant, it does not matter if the delay appears at the input or at the output. The input and state vectors of [26] are more general than in (1)-(4) which implies that (1)-(4) are in the model set of [26]. The input and state vectors $\mathbf{u}(t)$ and $\hat{\mathbf{x}}(t,\boldsymbol{\theta}_S)$ of [26] are

$$\mathbf{u}(t) = \left(\mathbf{u}_1^T(t) \;\; ... \;\; \mathbf{u}_K^T(t)\right)^T \tag{7}$$

$$\mathbf{u}_k(t) = \left(u_k(t) \;\; ... \;\; u_k^{(n_k)}(t)\right)^T, \; k = 1, ..., K, \tag{8}$$

$$\hat{\mathbf{x}}(t,\boldsymbol{\theta}_S) = \left(\hat{x}_1(t,\boldsymbol{\theta}_S) \;\; ... \;\; \hat{x}_n(t,\boldsymbol{\theta}_S)\right)^T. \tag{9}$$

Here the superscript $^{(n)}$ denotes differentiation $n$ times. $\boldsymbol{\theta}_S$ is the parameter vector of the ODE. The ODE is selected with one parameterized nonlinear right hand side state component that is integrated by a chain of integrators. This gives

$$\dot{\hat{\mathbf{x}}}(t,\boldsymbol{\theta}_S) = \begin{pmatrix} \dot{\hat{x}}_1(t,\boldsymbol{\theta}_S) \\ \vdots \\ \dot{\hat{x}}_{n-1}(t,\boldsymbol{\theta}_S) \\ \dot{\hat{x}}_n(t,\boldsymbol{\theta}_S) \end{pmatrix} = \begin{pmatrix} \hat{x}_2(t,\boldsymbol{\theta}_S) \\ \vdots \\ \hat{x}_n(t,\boldsymbol{\theta}_S) \\ f\left(\hat{\mathbf{x}}(t,\boldsymbol{\theta}_S), \mathbf{u}(t), \boldsymbol{\theta}_S\right) \end{pmatrix} \tag{10}$$

$$\hat{\mathbf{y}}(t,\theta_T,\boldsymbol{\theta}_S) = \mathbf{C}\hat{\mathbf{x}}(t-\theta_T,\boldsymbol{\theta}_S), \tag{11}$$

where $\hat{\mathbf{y}}(t,\theta_T,\boldsymbol{\theta}_S)$ is the model output, $\mathbf{C}$ is the output matrix, and

$$\theta_T = T \tag{12}$$

is the delay parameter. The total parameter vector becomes

$$\boldsymbol{\theta} = \left(\theta_T \;\; \boldsymbol{\theta}_S^T\right)^T. \tag{13}$$

Motivated by [22], (10) is parameterized by the polynomial

$$f(\hat{\mathbf{x}}(t,\boldsymbol{\theta}_S), \mathbf{u}(t), \boldsymbol{\theta}_S) = \boldsymbol{\varphi}^T(\hat{\mathbf{x}}(t,\boldsymbol{\theta}_S), \mathbf{u}(t))\boldsymbol{\theta}_S, \tag{14}$$

$$\begin{aligned} &\boldsymbol{\varphi}^T(\hat{\mathbf{x}}(t,\boldsymbol{\theta}_S), \mathbf{u}(t)) \\ &= \Bigg(1 \;\; ... \;\; \left(u_K^{(n_K)}(t)\right)^{I_{u_K^{(n_K)}}} \;\; ... \\ &\left(u_K^{(n_K-1)}(t)\right)^{I_{u_K^{(n_K-1)}}} \left(u_K^{(n_K)}(t)\right)^{I_{u_K^{(n_K)}}} \;\; ... \\ &\Bigg(\left(\hat{x}_1(t,\boldsymbol{\theta}_S)\right)^{I_{x_1}} \;\; ... \;\; \left(\hat{x}_n(t,\boldsymbol{\theta}_S)\right)^{I_{x_n}} \left(u_1(t)\right)^{I_{u_1}} \\ &\;\; ... \;\; \left(u_K^{(n_K)}(t)\right)^{I_{u_K^{(n_K)}}}\Bigg)\Bigg), \end{aligned} \tag{15}$$

where $I_m$ denotes a maximum degree, cf. [26], [27]. The regression vector component 1 corresponds to $\theta_{S,0...0}$, where

$$\begin{aligned} \boldsymbol{\theta}_S^T = \Bigg(&\theta_{S,0...0} \;\; ... \;\; \theta_{S,0...I_{u_K^{(n_K)}}} \\ &... \;\; \theta_{S,0...I_{u_K^{(n_K-1)}} I_{u_K^{(n_K)}}} \;\; ... \;\; \theta_{S,I_{x_1}...I_{u_K^{(n_K)}}}\Bigg). \end{aligned} \tag{16}$$

The parameter and regression vectors are filled with terms from left to right when the indices of the parameter vector vary. The rightmost index varies the fastest. The leftmost index represents the first state component of (9), while the rightmost index represents the last input signal component. Examples that provide further clarification of the notation appear in [25], [27] and [28].

The delay model is based on interpolation between multiple ODE models, each time shifted backwards with one sampling period $T_S$. The fractional delay is obtained by interpolation between adjacent ODE models as

$$\theta_T = T = mT_S + T_f, \quad m \in [0, M-1]. \tag{17}$$

Here $mT_S$ is the integer part of the delay where $m$ is the number of sampling periods. The maximum delay is $MT_S$ and $T_f$ denotes the fractional delay

$$0 \leq T_f < T_S. \tag{18}$$

Define $M+1$ models in terms of $\mathbf{u}_m(t)$ and $\hat{\mathbf{x}}_m(t,\boldsymbol{\theta}_S)$ by

$$\mathbf{u}_m(t) = \mathbf{u}(t - mT_S), \quad m = 0, ..., M, \tag{19}$$

$$\hat{\mathbf{x}}_m(t,\boldsymbol{\theta}_S) = \hat{\mathbf{x}}(t - mT_S, \boldsymbol{\theta}_S), \quad m = 0, ..., M, \tag{20}$$

Because of time invariance, (20) can be generated by using the $m$:th delayed input of (19) to solve the ODE.

The recursive identification algorithm performs interpolation of inputs, state vectors and gradients, towards the running estimate of the delay

$$\hat{\theta}_T(t) = \hat{m}(t)T_S + \hat{T}_f(t), \quad \hat{m}(t) \in [0, M-1]. \tag{21}$$

Linear interpolation is applied to get

$$\begin{aligned} &\hat{\mathbf{x}}(t - \hat{\theta}_T(t), \hat{\boldsymbol{\theta}}_S(t)) \\ &= \left(1 - \frac{\hat{T}_f(t)}{T_S}\right)\hat{\mathbf{x}}_{\hat{m}}(t, \hat{\boldsymbol{\theta}}_S(t)) + \frac{\hat{T}_f(t)}{T_S}\hat{\mathbf{x}}_{\hat{m}+1}(t, \hat{\boldsymbol{\theta}}_S(t)). \end{aligned} \tag{22}$$

$M+1$ integer delay models are defined by (19)-(22). The restriction (17) will keep the estimate interior to the delay range of the multiple models.

The output of the model is obtained from (11)

$$\begin{aligned} &\hat{\mathbf{y}}(t, \hat{\theta}_T(t), \hat{\boldsymbol{\theta}}_S(t)) \\ &= \hat{\mathbf{y}}(t - \hat{\theta}_T(t), \hat{\boldsymbol{\theta}}_S(t)) = \mathbf{C}\hat{\mathbf{x}}(t - \hat{\theta}_T(t), \hat{\boldsymbol{\theta}}_S(t)). \end{aligned} \tag{23}$$

The same interpolation is applied for the gradients, cf. [26].

### B. Discretization and Scaling

The ODE model and the gradient matrix ODE then need to be discretized. The reader is referred to [26] for a description of the discretization of the gradients of (10) and (11).

In the present paper, the discretization is augmented with scaling of $T_S$ as proposed by [25]. Briefly, the idea is to apply a scaled value of $T_S$ when the ODE and the associated gradient is discretized. The scaled sampling period is denoted

$$T_S^s = \alpha T_S, \tag{24}$$

where the superscript $^s$ denotes scaling. The analysis of [25] shows why this scaling can improve the convergence properties significantly, in particular when it comes to avoiding convergence to false local minima. The reason is believed to be the improved conditioning of the optimization problem that results from the changed scaling of the states, see Theorems 1 and 3 of [25]. A consequence of the scaling is that the identified parameter vector $\boldsymbol{\theta}_S$ is changed to $\boldsymbol{\theta}_S^s$, while $\theta_T = \theta_T^s$. However, as shown by Theorem 2 of [25] there is a linear relation between $\boldsymbol{\theta}_S$ and $\boldsymbol{\theta}_S^s$, which recovers $\boldsymbol{\theta}_S$.

To describe the discretization, (20) is first generated as

$$\begin{aligned} &\hat{\mathbf{x}}_m^s(t+T_s, \hat{\boldsymbol{\theta}}_S^s(t)) \\ &= \hat{\mathbf{x}}_{m-1}^s(t, \hat{\boldsymbol{\theta}}_S^s(t-T_S)), \quad m = 1, ..., M, \end{aligned} \tag{25}$$

Following this step, $\hat{\mathbf{x}}_0^s(t+T_s, \hat{\boldsymbol{\theta}}_S^s(t))$ is generated by the Euler forward integration method [17]. This results in

$$\begin{aligned} \begin{pmatrix} \hat{x}_{0,1}^s(t+T_S, \hat{\boldsymbol{\theta}}_S^s(t)) \\ \vdots \\ \hat{x}_{0,n-1}^s(t+T_S, \hat{\boldsymbol{\theta}}_S^s(t)) \\ \hat{x}_{0,n}^s(t+T_S, \hat{\boldsymbol{\theta}}_S^s(t)) \end{pmatrix} &= \begin{pmatrix} \hat{x}_{0,1}^s(t, \hat{\boldsymbol{\theta}}_S^s(t)) \\ \vdots \\ \hat{x}_{0,n-1}^s(t, \hat{\boldsymbol{\theta}}_S^s(t)) \\ \hat{x}_{0,n}^s(t, \hat{\boldsymbol{\theta}}_S^s(t)) \end{pmatrix} \\ &+ \alpha T_S \begin{pmatrix} \hat{x}_{0,2}^s(t, \hat{\boldsymbol{\theta}}_S^s(t)) \\ \vdots \\ \hat{x}_{0,n}^s(t, \hat{\boldsymbol{\theta}}_S^s(t)) \\ \boldsymbol{\varphi}^T(\hat{\mathbf{x}}_0^s(t, \hat{\boldsymbol{\theta}}_S^s(t)), \mathbf{u}(t)) \hat{\boldsymbol{\theta}}_S^s(t) \end{pmatrix}. \end{aligned} \tag{26}$$

Here the first index of the subscript of the state components refers to $m = 0$, while the second index denotes the state component number.

### C. The recursive identification algorithm

To ensure that the parameter estimate remains in the model set, a projection algorithm is needed. The model set $D_{\mathcal{M}}$ underpinning the projection algorithm is approximated with the linearized asymptotically stable models that have a delay in the set of (17), i.e.

$$\begin{aligned} &D_{\mathcal{M}}^s \\ &= \left\{ \left( \theta_T^s \ (\boldsymbol{\theta}_S^s)^T \right)^T \mid |eig(\mathbf{S}^s(\boldsymbol{\theta}^s))| < 1 - \kappa, m \in [0, M-1] \right\} \end{aligned} \tag{27}$$

where $\kappa > 0$ is a small number. Furthermore

$$\begin{aligned} &\mathbf{S}^s(\boldsymbol{\theta}^s) \\ &= \mathbf{I}_n + \alpha T_S \begin{pmatrix} 0 & 1 & 0 & \ldots & 0 \\ 0 & 0 & 1 & \ddots & 0 \\ \vdots & \vdots & \ddots & \ddots & 0 \\ 0 & 0 & \ldots & 0 & 1 \\ & & (\boldsymbol{\theta}^s)^T \ \frac{d\boldsymbol{\varphi}^s(\mathbf{x}^s(t,\boldsymbol{\theta}^s))}{d\mathbf{x}^s(t,\boldsymbol{\theta}^s)} & & \end{pmatrix}. \end{aligned} \tag{28}$$

The scaled algorithm of [26] now follows by minimization of the criterion

$$\begin{aligned} V(\boldsymbol{\theta}^s, \boldsymbol{\Lambda}^s) = \frac{1}{2} \lim_{t\to\infty} E[(\boldsymbol{\varepsilon}^s(t, \boldsymbol{\theta}^s)^T (\boldsymbol{\Lambda}^s(t, \boldsymbol{\theta}^s))^{-1} \boldsymbol{\varepsilon}^s(t, \boldsymbol{\theta}^s) \\ + \ln \det (\boldsymbol{\Lambda}^s(t, \boldsymbol{\theta}^s))], \end{aligned} \tag{29}$$

using the Gauss Newton method of [15], where $\boldsymbol{\Lambda}^s(t)$ is the covariance matrix of the prediction error

$$\boldsymbol{\varepsilon}^s(t, \boldsymbol{\theta}^s) = \mathbf{y}(t) - \hat{\mathbf{y}}^s(t, \hat{\theta}_T(t), \hat{\boldsymbol{\theta}}_S^s(t)). \tag{30}$$

The resulting algorithm updates $\hat{\boldsymbol{\theta}}^s(t)$, $\boldsymbol{\Lambda}^s(t)$ and the Hessian $\mathbf{R}^s(t)$, given scaled model and gradient predictions. At each time step, it is checked if the estimates are in the model set (27). If not, no update is performed. All further algorithmic details are available in [26] and [28].

### D. Tuning

The running estimates $\hat{\boldsymbol{\theta}}^s(t)$, $\boldsymbol{\Lambda}^s(t)$ and $\mathbf{R}^s(t)$ need to be initialized. Typically the matrices can be initialized as diagonal matrices, with diagonal elements of the same order of magnitude as the squared expected initial prediction errors and the expected initial squared parameter errors. Often a single parameter can be used for each matrix, as shown by the free software package [27]. In the present tracking application the exponential forgetting factor should be 1.

The scale factor $\alpha$ should be set so that the magnitude of the state components are equalized. Experimentation may be needed using plots of the time evolution of the eigenvalues of $\mathbf{R}^s(t)$ obtained from [27].

Another relevant aspect concerns model selection. For [26], this amounts to selection of what polynomial terms that are included in the identified model. The software package [27] provides selection flexibility. It is particularly important to make selections that strengthens *observability*.

### E. Potential Detection Mechanisms

The proposed detection mechanism is based on the delay parameter $\hat{\theta}_T(t)$ that is identified by the algorithm of [26]. The idea is to define an allowed range for $\hat{\theta}_T(t)$ during normal operation, and to define a detection threshold $\theta_{T,max}$ such that

$$\hat{\theta}_T(t) > \theta_{T,max} \tag{31}$$

triggers an alarm that, for example, disables closed loop operation. The details of optimal threshold setting could be based on an analysis of the false alarm rate. The details are left for further research.

## V. Numerical Example - Automotive Cruise Control

### A. Feedback Linearized Controller

A vehicle traveling with a velocity $x_1(t)$ is subject to a number of forces, including engine thrust, friction, air resistance and gravitational forces in hilly terrain, see [2], [16], [31]. In the present work, the friction and gravitational forces are treated as a lumped system disturbance $d(t)$. The forces listed and Newton's second law give

$$\dot{x}_1(t) = a(t) - \frac{\rho F C_{x_1}}{2m} x_1^2(t) - d(t). \qquad (32)$$

Here $a(t)$ is the accelerator command, $m$ is the mass of the vehicle, $\rho$ is the density of the air, $F$ is the frontal area and $C_{x_1}$ is the air resistance coefficient.

The dynamics of (32) is strongly nonlinear. Noting that the structure of (32) equals that of (1) with

$$\mathbf{u}(t) = a(t) - d(t), \qquad (33)$$

$$\mathbf{g}_{\boldsymbol{\theta}_S^0}(\mathbf{x}(t)) = 1, \qquad (34)$$

$$\mathbf{f}_{\boldsymbol{\theta}_S^0}(\mathbf{x}(t)) = -\frac{\rho F C_{x_1}}{2m} x_1^2(t), \qquad (35)$$

it follows that feedback linearization according to Lemma 1 can be applied. Here $\boldsymbol{\theta}_S^0$ indicates the true parameter vector. The additive input-state linearizing transformation

$$u_1(t) = a(t) - k_{\boldsymbol{\theta}_S} x_1^2(t) + \delta x_1(t) \qquad (36)$$

transforms (32) to

$$\begin{aligned}\dot{x}_1(t) &= u_1(t) + \left(k_{\boldsymbol{\theta}_S} - \frac{\rho F C_{x_1}}{2m}\right) x_1^2(t) - \delta x_1(t) - d(t) \\ &= -\delta x_1(t) + u_1(t) - d(t), \qquad (37)\end{aligned}$$

provided that

$$k_{\boldsymbol{\theta}_S} = k_{\boldsymbol{\theta}_S^0} = \frac{\rho F C_{x_1}}{2m}. \qquad (38)$$

Here $\delta > 0$ is a small constant selected to avoid pure integration in the stability analysis of Section V.C. Elsewhere, $\delta = 0$ is used.

The next step is to design an outer regulator to generate $u_1(t)$. Equation (37) contains an integrator, therefore it may seem that integrating outer loop control is not needed. However, (38) cannot be expected to hold exactly. In such cases a small signal analysis shows that the integrator pole is shifted to a pole beside the imaginary axis, which means that integrating outer loop control is anyway advisable to regulate away static disturbance components.

In [30] it was shown how integrating LQG control could be applied to a conventionally linearized version of (32), with $(\gamma/m)x_1(t)$ replacing the quadratic function. The resulting controller turned out to be the PI-controller

$$C(s) = K_P + \frac{1}{T_I}\frac{1}{s} \qquad (39)$$

where $s$ is the Laplace transform variable, $K_P$ is the proportional gain and $T_I$ is the integration time. To allow a comparison to [30], the same design methodology as in [30] is applied here. The controller $C(s)$ filters the control error between the reference signal $y_1^{ref}(s)$ and the potentially delayed attacked output signal $e^{-sT}y_1(s)$.

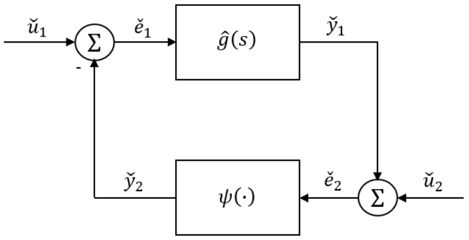

Fig. 2. Block diagram for which the Popov criterion holds. Note that carets are used to denote the signals of the figure.

The acceleration command is limited by the maximum thrust, and by the maximum braking/electrical re-generation that is allowed. Noting that $d(t)$ is bounded and small in comparison to the acceleration command and air resistance, the acceleration command limitation can be moved to after $u_1$ of Fig. 1 by a slight reduction of the range of $u_1(t)$. The limitation is then given by

$$\begin{aligned} u_1(t) &= f_u(u(t)) \\ &= \begin{cases} u_{1,max}, & u_{1,max} < u(t) \\ u(t) & u_{1,min} \leq u(t) \leq u_{1,max} \\ u_{1,min} & u(t) < u_{1,min} \end{cases}. \qquad (40)\end{aligned}$$

where $u(t)$ is defined by Fig. 1, and where $u_{1,min}$ and $u_{1,max}$ are the lower and upper limits of $u(t)$, respectively.

### B. Feedback Signaling Delay Injection

As motivated in section III.D, a disguised delay injection attack is assumed, marked with red in Fig. 1, and given by

$$y_1(t-T) = L^{-1}\left(e^{-sT}L(y_1(t))\right). \qquad (41)$$

### C. Stability

Assume that (38) holds exactly, i.e. that

A1) $k_{\boldsymbol{\theta}_S} = k_{\boldsymbol{\theta}_S^0} = \frac{\rho F C_{x_1}}{2m}$.

Then the regulator problem of (37) is linear except for the static nonlinear limitation and no dependence on $\boldsymbol{\theta}_S$ appears. Because of the delay injection attack, the global $\mathcal{L}_2$-stability of the system can be analysed by application of the input-output version of the Popov criterion [18], [19], [23], [33]. The analysis builds on the definitions D1-D5 of the appendix, that can be used to prove

*Lemma 2 (Popov Criterion, [23] Theorem 6.7.63):* Consider the system of Fig. 2. Assume that the inverse Laplace transform of the transfer function $\hat{g}(s)$ fulfils

$$g(\cdot) \in \mathcal{A}, \quad \dot{g}(\cdot) \in \mathcal{A},$$

that the time invariant continuous static nonlinearity $\psi(\cdot)$ fulfils

$$0 \leq \sigma\psi(\sigma) \leq \beta\sigma^2,$$

and that $\check{u}_1 \in \mathcal{L}_2$, $\check{u}_2 \in \mathcal{L}_2$, $\dot{\check{u}}_2 \in \mathcal{L}_2$. Under these conditions the system is $\mathcal{L}_2$-stable if there exist constants $\check{q}$, $\delta_P$, such that the Popov plot

$$\omega \in [0, \infty) \rightarrow Re[\hat{g}(j\omega)] + j\omega Im[\hat{g}(j\omega)] \in \mathcal{C}$$

is entirely to the right of a line through $-1/\beta + \delta_P + j0$ with slope $1/\check{q}$, for some $\check{q} \geq 0$ and some $\delta_P > 0$. □

*Proof:* See [23], Section 6.7.

To apply the Popov criterion, the signals, transfer functions and static nonlinearities of Fig. 2 that appear in Lemma 2 need to be computed in terms of quantities of Fig. 1. Assumptions then need to be imposed on the signals, such that the conditions of Lemma 2 are fulfilled. This will prove that Lemma 2 holds and enable its application.

Noting that delays can be freely moved through a static nonlinearity and that linear blocks can be re-ordered, an analysis of Fig. 1 and Fig. 2 immediately shows that

$$\hat{g}(s) = e^{-sT} C(s) \frac{1}{s + \delta}, \tag{42}$$

$$\psi(u) = f_u(u), \tag{43}$$

$$\check{u}_1(s) = d(s), \tag{44}$$

$$\check{u}_2(s) = C(s) y_1^{ref}(s). \tag{45}$$

The requirements on $g(t)$ and $\dot{g}(t)$ means that $\hat{g}(s)$ needs to be asymptotically stable and strictly proper. To ensure that these requirements hold, introduce the assumption

A2) C(s) is asymptotically stable and proper.

A2 is true provided that the PI controller is replaced by the leaky PI controller $K_P + \frac{1}{T_I} \frac{1}{(s+\delta)}$. A1 and A2 then imply that $g(\cdot) \in \mathcal{A}$ and $\dot{g}(\cdot) \in \mathcal{A}$. Furthermore, the static nonlinearity meets the continuity condition. The requirements on $\check{u}_1$ and $\check{u}_2$ are secured by the following assumptions

A3) $d(s)$ is the Laplace transform of a signal generated by asymptotically stable and proper filtering.

A4) $y_1^{ref}(s)$ is the Laplace transform of a signal generated by asymptotically stable and strictly proper filtering with static gain $y_1^{ref}$.

The assumptions A3 and A4 are not restrictive in practice. A3 immediately implies that $\check{u}_1 \in \mathcal{L}_2$ , while A2 and A4 imply that $\check{u}_2 \in \mathcal{L}_2$ and $\dot{\check{u}}_2 \in \mathcal{L}_2$. This gives

*Theorem 1:* Consider the control system of Fig. 1 and assume that A1-A4 hold. Then Lemma 2 holds. □

It is stressed that a small $\delta > 0$ is needed for the stringent validity of the Popov criterion. It is, however, conjectured that the result holds unaltered for Fig. 1 when $\delta \rightarrow 0$.

### *D. Destabilization at Activation*

To illustrate the effect of a delay injection attack in the feedback path, the closed loop cruise control dynamics was simulated. The parameters of (32) were adjusted to give a maximum velocity of 60 $m/s$ in stationary state when a maximum acceleration of $u_{1,max} = 3.0\ m/s^2$ was applied with $d = 0.0\ m/s^2$. The maximum retardation was $u_{1,min} = -3.0\ m/s^2$. The mass of the vehicle was $m = 1500\ kg$. The state penalties of [30] were selected as $q_{11} = 3.00$ and

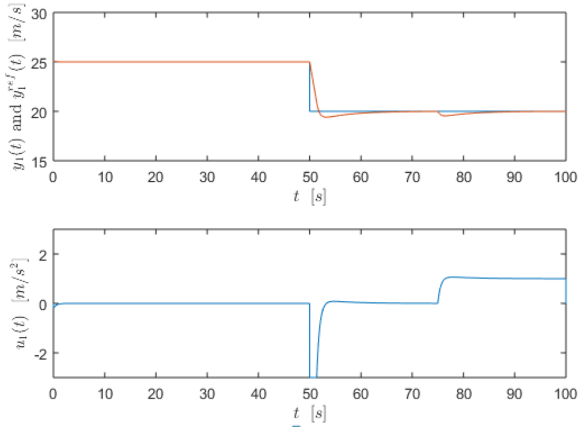

Fig. 3. Closed loop performance for perfect input-state linearization.

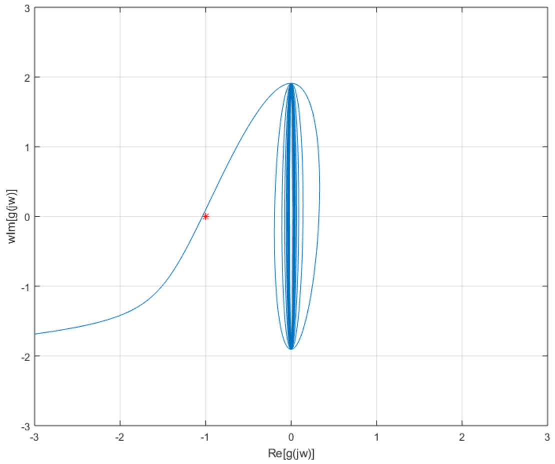

Fig. 4. The Popov curve with a delay injection attack of $T = 0.80\ s$ for $\delta = 0.1$. The critical pivot point $-1 + 0j$ of the Popov line is marked red.

$q_{22} = 0.10$, while the control signal penalty was $q_2 = 1.00$. This resulted in $K_P = 1.91$ and $T_I = 3.16$ s. A disturbance $d(t) = -1.0\ m/s^2$ starts affecting the system at $t = 75\ s$. Tustin's approximation, [17], was used to discretize the outer loop with a sampling period of $T_s = 0.001\ s$. The nonlinear input-state linearization loop was discretized with the Euler method.

Without any attack, and with perfect linearization as stated by A1, the performance of Fig. 3 was obtained.

A delay attack occurring after 65 $s$, injecting a delay of $T = 0.80\ s$ was then simulated. As seen by Fig. 4, there is no longer any possibility to draw a Popov line through $-1 + 0j$ so that the entire Popov curve is to the right of the line, hence the Popov criterion does not imply stability. As shown by Fig. 5 the conclusion of the Popov criterion is correct. The instability quickly becomes disturbing.

The performance of the feedback linearization was robust, with very similar behaviour as in Fig. 3 and Fig. 5, when $k_{\boldsymbol{\theta}_S}$ varied and was overestimated up to a factor of 4.

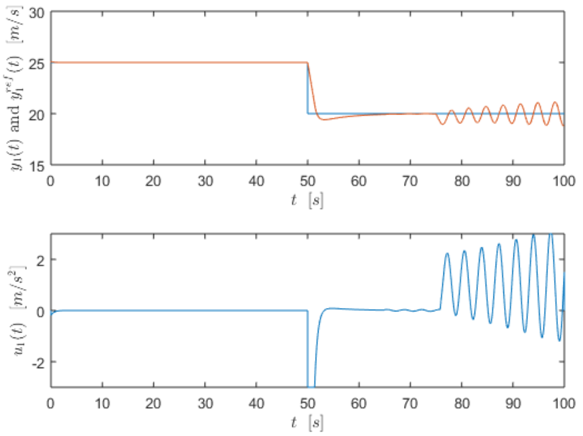

Fig. 5. The effect of the delay attack, injecting $T = 0.80$ $s$ at 65 $s$.

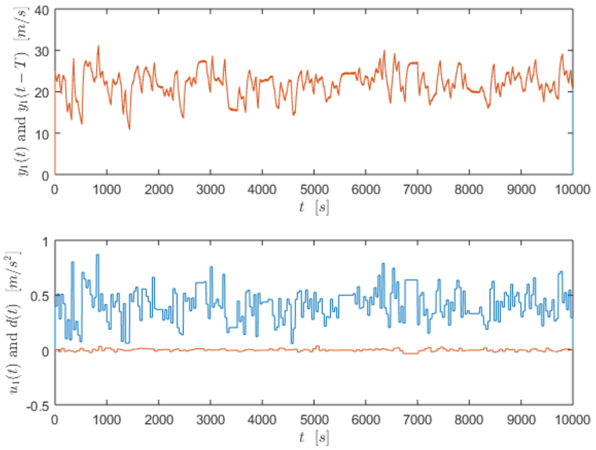

Fig. 6. Open loop delay attack at $t = 6000$ $s$. The bottom figure shows the manual control blue and the disturbance red.

## E. Early Detection with Recursive Identification

The control system of Fig. 1 was then simulated in open loop. The zero mean Gaussian disturbance $d(t)$ had a standard deviation of $0.01$ $m/s^2$. A delay injection attack was executed at $t = 6000$ $s$, during open loop operation. The simulated velocity with and without delay, the manual control and the disturbance are depicted in Fig. 6. The attack remains very well disguised. The curves of the top figure coincide and it is not possible to notice any effect of the attack.

The algorithm of [26], set for tracking, was used for delay injection attack detection. The algorithm used the control signal $u_1(t)$ and the potentially delayed velocity $y(t-T)$, measured at the left open loop/closed loop switch of Fig. 1.

Guided by the physical model behind the control loop the following model structure was selected for recursive identification, cf. (13)-(16)

$$f(\hat{\mathbf{x}}(t,\hat{\boldsymbol{\theta}}_S(t)),\mathbf{u}(t),\hat{\boldsymbol{\theta}}_S(t))$$

$$= \hat{\theta}_{S,00}(t)1 + \hat{\theta}_{S,01}(t)u_1(t) + \hat{\theta}_{S,20}(t)\hat{x}_1^2(t,\hat{\boldsymbol{\theta}}_S(t)), \quad (46)$$

$$\hat{y}(t,\hat{\theta}_T(t),\hat{\boldsymbol{\theta}}_S(t)) = \hat{x}_1(t-\hat{\theta}_T(t),\hat{\boldsymbol{\theta}}_S(t)). \quad (47)$$

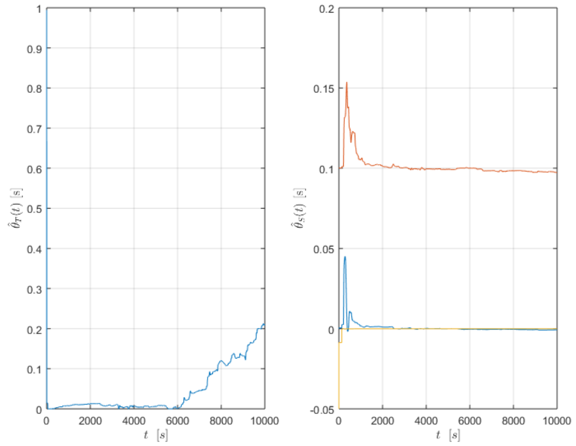

Fig. 7. The parameter estimates of the algorithm of [26].

The parameter $\hat{\theta}_{S,00}(t)$ compensates for any biases. Note that no term linear in $\hat{x}_1(t,\hat{\boldsymbol{\theta}}_S(t))$ was included. The reason is that for small signal variations, the quadratic term would behave as linear, thereby compromising observability. The SW implementation of the algorithm, [27], used the tracking gain $\mu_0 = 1.0$, the scale factor $\alpha = 10$, the projection radius $1-\kappa = 0.9995$, the delay range $[0.0, 1.0]$ $s$, and the sampling period $T_S = 0.10$ $s$. The algorithm was initialized with $\Lambda^s(0) =$ 10 to handle large initial prediction errors. The Hessian and the parameter vector were initialized by

$$\mathbf{R}^s(0) = \begin{pmatrix} 0.10 & \mathbf{0} \\ \mathbf{0} & 1.00\mathbf{I}_3 \end{pmatrix}, \quad (48)$$

$$\hat{\boldsymbol{\theta}}^s(0) = (0.2000 \quad 0.0000 \quad 0.1000 \quad -0.0500)^T. \quad (49)$$

The time evolution of the parameter estimates of the recursive algorithm are depicted in Fig. 7. The performance is excellent. The algorithm immediately converges to a setting where $0.00 < \hat{\theta}_T(t) < 0.02$ $s$. This estimate is retained until the delay attack, when $\hat{\theta}_T(t)$ starts to increase, allowing for delay change detection when $t > 6300$ $s$, as judged by manual inspection of Fig. 7. Considering the variation of the velocity of Fig. 6, the response is rapid. The parameters describing the dynamics also converge rapidly, to the final re-scaled estimate

$$\hat{\boldsymbol{\theta}}_S(10000) = (-0.00730 \quad 0.97290 \quad -0.00080)^T. \quad (50)$$

This can be compared to the true parameter vector

$$\hat{\boldsymbol{\theta}}_S^0 = (0.00000 \quad 1.00000 \quad -0.00083)^T. \quad (51)$$

## VI. Conclusions

The paper developed a method to detect delay injection into nonlinear control systems, subject to input-state feedback linearization. The paper also analysed the $\mathcal{L}_2$-stability of the system, thereby addressing the robustness of the linearized feedback loop against delay attacks. The detection of the delay attack was performed by a recursive identification algorithm, based on a nonlinear state space model with output delay. A distinctive advantage is that the method is applicable also in cases where the system dynamics is not completely known. The identified dynamics can also be used for controller tuning. The numerical evaluation performed for a nonlinear automotive cruise controller showed that the proposed recursive identification algorithm is able to detect a well disguised delay attack, and do so rapidly.

## VII. Appendix

### $\mathcal{L}_p$ Stability Definitions

D1) For all $p \in [1, \infty)$, $\mathcal{L}_p[0, \infty)$ denotes the set of all measurable functions $f(\cdot) : [0, \infty) \to \mathcal{R}$, such that

$$\|f(\cdot)\|_p^p = \int_0^\infty |f(t)|^p dt < \infty.$$

D2) The set of all measurable functions $f(\cdot) : [0, \infty) \to \mathcal{R}$, such that their truncations

$$f_{\check{T}}(t) = \left\{ \begin{array}{ll} f(t), & 0 \leq t \leq \check{T} \\ 0, & t > \check{T} \end{array} \right. \in \mathcal{L}_p[0, \infty), \forall \check{T},$$

is denoted the extension $\mathcal{L}_{pe}[0, \infty)$ of $\mathcal{L}_p[0, \infty)$.

D3) The mapping $A : \mathcal{L}_{pe} \to \mathcal{L}_{pe}$ is $\mathcal{L}_p$-stable if i) $Af \in \mathcal{L}_p$ whenever $f \in \mathcal{L}_p$, and ii) there exist finite constants $l, c$, such that

$$\|Af\|_p \leq l\|f\|_p + c, \quad \forall f \in \mathcal{L}_p.$$

D4) $\mathcal{A}$ denotes the set of generalized functions of the form

$$f(t) = \left\{ \begin{array}{cc} 0, & t < 0 \\ \sum_{i=0}^\infty f_i \delta(t - t_i) + f_a(t), & t \geq 0 \end{array} \right. ,$$

where $\delta(\cdot)$ is the unit delta distribution, $t_i$ are non-negative constant delays, $f_a(t)$ is measurable and

$$\textstyle\sum_{i=0}^\infty |f_i| < \infty, \qquad \int_0^\infty |f_a(t)| dt < \infty.$$

D5) $\hat{A}$ denotes the set of all function $\hat{f} : \mathcal{C}_+ \to \mathcal{C}$ that are Laplace transforms of elements of $\mathcal{A}$.

## References

[1] "Service requirements for next generation new services and markets, rev. 16.4.0", 3GPP, TS 22.261, 2018.

[2] B. van Arem, C. J. G van Driel and R. Visser, "The impact of cooperative adaptive cruise control on traffic-flow scenarios", *IEEE Trans. Intelligent Transportation Systems*, vol 7, no. 4, pp. 429-436, 2006.

[3] K. J. Åström and T. Hägglund, *Advanced PID Control.* Reasearch Triangle Park, NC: The Instrumentation, Systems and Automation Society, 2006.

[4] K. J. Åström and B. Wittenmark, *Adaptive Control.* New York, NY:Dover, 2013.

[5] G. Bianchin and F. Pasqualetti, "Time-Delay Attacks in Networked Systems", *in Ç. K. Koç (ed.), Cyber-Physical Systems Security*, pp. 157-174, Cham. :Springer, 2018.

[6] S. Björklund and L. Ljung, "A review of time-delay estimation techniques", In *Proc. 42nd IEEE Conference on Decision and Control*, pp. 2502-2507, Maui, Hawaii, USA, 2003.

[7] V. Bro and A. Medvedev, "Identification of continuous Volterra models with explicit time delay through series of Laguerre functions", In *Proc. 58th IEEE Conference on Decision and Control*, pp. 5641-5646, Nice, France, 2019.

[8] M. S. Chong, H. Sandberg and A. M. H. Teixeira, "A tutorial introduction to security and privacy for cyber-physical systems", In *Proc. European Control Conference*, Naples, Italy, 2019.

[9] D. Corona and B. DeSchutter, "Adaptive cruise control for a SMART car: a comparison benchmark for MPC-PWA control methods", *IEEE Trans. Contr. Systems Tech.*, vol. 16, no. 2, 2008.

[10] E. Fridman, *Introduction to Time-Delay Systems: Analysis and Control.* Cham.: Springer, 2014.

[11] A. J. Isaksson, A. Horch and G. A. Dumondt, "Event-triggered deadtime estimation from closed-loop data", In *Proc. American Control Conference*, pp. 3280-3285, Arlington, Virginia, USA, 2001.

[12] H. K. Khalil, *Nonlinear Systems, 2:nd Ed.*. Upper Saddle River, NJ: Prentice Hall, 2002.

[13] E. Korkmaz, A. Dolgikh, M. Davis and V. Skormin, "ICS security testbed with delay attack case study", In *Military Communications Conference*, pp. 283-288, 2016.

[14] H. Kurz and W. Goedecke, "Digital parameter-adaptive control of processes with unknown deadtime", *Automatica*, vol. 17, pp. 245-252, 1981.

[15] L. Ljung and T. Söderström, *Theory and Practice of Recursive Identification*. Cambridge, MA: MIT Press, 1983.

[16] P. Nilsson, O. Hussien, A. Balkan, Y. Chen, A. D. Ames, J. W. Grizzle, N. Ozay, H. Peng and P. Tabuada, "Correct-by-construction adaptive cruise control: two approaches", *IEEE Trans. Contr. Systems Tech.*, vol. 24, no. 4, pp. 1294-1307, 2016.

[17] A. V. Oppenheim and R. W. Schafer, *Digital Signal Processing*. Englewood Cliffs, NJ: Mc Graw Hill, 1975.

[18] V. M. Popov, "Nouveaux criteriums de stabilité pour les systemés automatiques non-linèaries", *Revue d´Electrotechnique et d´Energetique, Acad. de la Rep. Populaire Romaine*, vol. 5, no. 1, 1960.

[19] I. W. Sandberg, "A frequency-domain condition for the stability of feedback systems containing a single time-varying element", *Bell Sys. Tech. J.*, vol. 43, pp. 1601-1608, 1964.

[20] A. Sargolzaei, F. M. Segers, A. Abbaspour, C. D. Crane and W. E. Dixon, "Secure control design for networked control systems with nonlinear dynamics under time-delay-switch attacks", *IEEE Trans. Automat. Contr.*, vol. 68, no. 2, pp. 798-811, 2023.

[21] X.-C. Shanguan, Y. He, C.-K. Zhang, W. Yao, J. Liang and M. Wu, "Resilient load frequency control of power systems to compensate random time-delay attacks". *IEEE Trans. Ind. Elec.*, vol. 70, no. 5, pp. 5115-5128, 2023.

[22] M. H. Stone, "The generalized Weierstrass approximation theorem", *Math. Mag.*, vol. 21, pp. 148-157, 1948.

[23] M. Vidyasagar, *Nonlinear Systems Analysis*. Englewood Cliffs, NJ, USA: Prentice-Hall, 1978.

[24] A. Wigren, M. Wågberg, F. Lindsten, A. G. Wills and T. B. Schön, "Nonlinear system identification: learning while respecting physical models using a sequential Monte Carlo method", *IEEE Contr. Systems Mag.*, vol. 42, pp. 75-102, 2022.

[25] T. Wigren, "Scaling of the sampling period in nonlinear system identification", In *Proc. American Control Conference*, pp. 5058-5065, Portland, Oregon, U.S.A, 2005.

[26] T. Wigren, "Networked and delayed recursive identification of nonlinear systems", In *Proc 56th IEEE Conference on Decision and Control*, pp. 5851-5858, Melbourne, Victoria, Australia, 2017.

[27] T. Wigren, "MATLAB software for nonlinear and delayed recursive identification - revision 2", Uppsala University, Uppsala, Sweden, Technical Reports from the Department of Information Technology, 2022-002, 2022.

[28] T. Wigren, "Convergence in delayed recursive identification of nonlinear systems", In Proc. European Control Conference, Stockholm, Sweden, June 25-28, 2024.

[29] T. Wigren and A. Teixeira, "On-line identification of delay attacks in networked servo control", In *Prep. IFAC World Congress*, pp. 1041-1047, Yokohama, Japan, 2023.

[30] T. Wigren and A. Teixeira, "Feedback path delay attacks and detection", In *Proc. 62:nd IEEE Conference on Deccision and Control*, Singapore, 2023.

[31] H. Yueming and L. Zhiyuan, "An $\mathcal{H}_2/\mathcal{H}_\infty$ robust control approach to electric vehicle constant speed cruise", In *Proc. Chinese Control Conference*, pp. 2384-2389, Yantai, China, 2011.

[32] K. S. Xiahou, Y. Liu and Q. H. Wu, "Robust load frequency control of power systems against random time-delay attacks", *IEEE Trans. Smart Grid*, vol. 12, no. 1, pp. 909-911, 2021.

[33] G. Zames, "On the input-output stability of time-varying nonlinear feedback systems, Part1: Conditions derived using concepts of loop-gain, conicity and positivity", *IEEE Trans. Automat. Contr.*, vol. AC-11, pp. 228-238, 1966.