http://www.diva-portal.org

Postprint

This is the accepted version of a paper presented at *European Control Conference*.

Citation for the original published paper:

Eriksson, L., Wigren, T., Zachariah, D., Teixeira, A. (2026)
Detecting Feedback-path Delay Injection Attacks Using InteractingMultiple Model Filtering
In: *24th European Control Conference (ECC), Reykjavík, July 7-10, 2026*

N.B. When citing this work, cite the original published paper.

Permanent link to this version:
http://urn.kb.se/resolve?urn=urn:nbn:se:uu:diva-595926

Detecting Feedback-path Delay Injection Attacks Using Interacting Multiple Model Filtering

Lovisa Eriksson, Torbjörn Wigren, Dave Zachariah and André M. H. Teixeira

***Abstract*— Time-delays are known to have a detrimental effect on feedback systems. In the context of networked cyber-physical systems, delays can be injected by malicious adversaries. Detecting them early is an important challenge. This paper proposes a novel variation of Interacting Multiple Model filtering to detect delay injection attacks in feedback control systems, when hidden in an open loop setting. The detection scheme is formalised by treating delay as alternative modes of the system, and theoretical analysis of the stationary distribution informs a reduction to a three parameter model as well as the choices of hyper parameter values. The method is evaluated on a cruise control application, and shows detection within a few seconds and a low false alarm probability.**

I. INTRODUCTION

Time delays can cause serious issues in feedback control systems [1], particularly in control over networks [2], leading to poor performance, lack of robustness, and possibly loss of stability. Therefore, delay injection attacks have been studied in multiple previous works [3], [4], [5], [6], [7]. These suggest various mitigation techniques, such as hypothesis testing [6] and Neural Network delay estimators [4], [7].

This paper follows [3] and considers delay attacks where an unknown delay is inserted in the feedback loop, so that it only affects the observations, as illustrated in Fig. 1. Both the amount of delay and time of attack are unknown, but the system dynamics are assumed to be known. By adapting the widely applied Interacting Multiple Model (IMM) filter [8], [9] to the problem of delay detection, a set of Kalman filters for various modes (i.e., delay hypotheses) is constructed, and propagated at each time step. The estimates from all filters at the previous time step are mixed according to the probabilities of the modes.

By determining how well the estimates of each mode fit the observations, the probability of each hypothesis is calculated in a Bayesian fashion. When the probability of 'no attack' falls below a threshold, an attack is detected, and action can be taken to prevent reduced performance or destabilisation of the feedback control loop.

A similar approach for delay detection was proposed in [6], where hypothesis testing is applied to check how well the data fit different assumptions on delay. This differs from the approach in the present paper by not allowing mixing of estimates; instead it is assumed at each time step that there was no delay in the previous time step and no

This work is supported by the Swedish Research Council under grants 2021-05022, 2023-05234, and 2024-03903, the Swedish Foundation for Strategic Research, and the Knut and Alice Wallenberg Foundation. The authors are with Department of Information Technology, Uppsala University, Uppsala, Sweden. {lovisa.eriksson, torbjorn.wigren, dave.zachariah, andre.teixeira}@it.uu.se

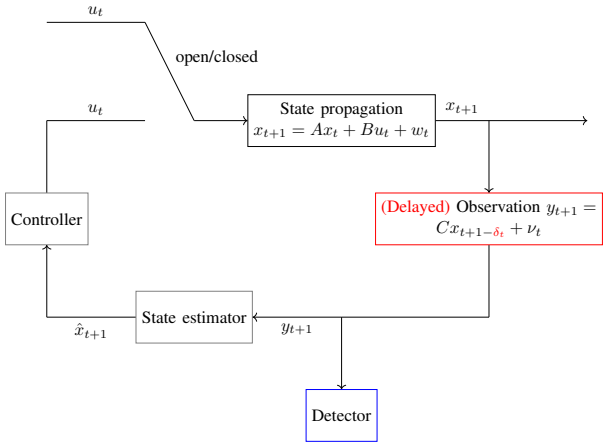

Fig. 1: Block diagram of the model. The attack only affects the observation.

Markov transitions are used. Further, [6] does not specifically consider delays in the feedback path. In [3], on the other hand, the attack is in the feedback path but the delay is estimated together with the full system dynamics, making it more flexible but less accurate than the proposed IMM based method. None of the previous papers develops a formal detection scheme, but rather plots estimated delays so that a human overseer could make manual decisions on whether actions are necessary. The method proposed in the current paper allows automatic detection, and is faster than both above mentioned schemes.

Inserting the attack only in the feedback path ensures it goes unnoticed in an open loop setting, but still causes problems when the loop is closed. For a more thorough motivation of the attack placement, see [3].

The contributions of this paper are as follows:

1) The IMM-based scheme for delay attack detection is formulated under a quickest change detection setting, leveraging the posterior probabilities as the test statistic for the detection rule;
2) Based on the underlying Hidden Markov Model (HMM), the theoretical analysis provides guidelines for tuning the hyperparameters of the HMM and the detection threshold;
3) The numerical example on a safety-critical application, cruise control, illustrates the applicability of the method and shows good detection performance.

The paper is organised as follows. Section II-A presents the formal problem, specifying the system model and the

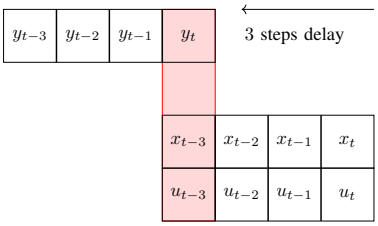

Fig. 2: Illustration of observations delayed by 3 time steps. The observed output at time t reflects the actual state at time $t-3$.

delay attacks. The evaluation metrics to assess the performance of the detection scheme are defined in Section II-B. Section III-A describes how IMM can be applied for delay detection in a general setting, Section III-B how the HMM is designed, and Section III-C how the remaining parameters are set. In Section IV, the model is applied to cruise control, and simulation results are presented and discussed. The paper is concluded in Section V with final remarks and suggestions for future work.

II. PROBLEM FORMULATION

A. State Space and Attack Model

Consider the linear time-invariant system model

$$\begin{cases} x_{t+1} &= Ax_t + Bu_t + w_t \\ y_t &= Cx_{t-\delta_t} + \nu_{t-\delta_t}, \end{cases} \tag{1}$$

with independent white noise processes $w_t \sim \mathcal{N}(0, Q)$, $\nu_t \sim \mathcal{N}(0, R)$. The system model parameters A, B, C, as well as R and Q, are known. However, the *observed* output y_t is subject to a (possibly) time-varying delay δ_t. For ease of evaluation, the system is assumed to have no inherent delay. A delay injection attack adds an arbitrary delay into the observations. No further assumptions are made, and so the delay size can depend arbitrarily on time. This is formalised as Assumption 1.

Assumption 1. *The delay δ_t is a function $\delta : \mathbb{N} \to \mathbb{N}$, such that $\delta_t = 0$ for $t < t_*$ and $\delta_t > 0$ for $t \geq t_*$, where t_* denotes an* unknown *attack time. Otherwise, the function class of the delay as well as its current value are* unknown.

This assumption is weak and allows a wide range of attack schemes, e.g. constant delay, randomized delay, slowly increasing delay, and replay attacks.

An example of the resulting observations for 3 time steps of delay is shown in Fig. 2, where it can be seen how the observation at time t is not a function of the true state at that time, but rather the true state at time $t-3$.

This attack model, illustrated in Fig. 1, is parsimonious, in that it only affects the observations via the feedback path, yet effective: it does not affect the system's states in open loop, and thus remains hidden, but is often able to cause destabilisation of the closed-loop system. For a more thorough motivation of placing attacks in the feedback path, see [3].

Remark 1. *It should be noted that the delay is only inserted in the feedback path. As detection is performed in the open loop setting, the state estimator and controller in Fig. 1, do not affect the detection performance, but is still included to highlight the potential destabilizing effect of the attack if the loop is closed.*

The potential to switch between open and closed loop settings is common in semi-automated tasks, where a human can switch between determining the input u_t and using a feedback controller. Two safety-critical examples are cruise control and autopilot systems in cars [10], [11] and aircraft [12].

B. Attack Detector and its Performance

Attack detection is performed by a detector $\text{ALARM}(t)$, detecting the presence of a delay attack, i.e. $\delta_t > 0$, in open loop. The detector is formally defined in Definition 1.

Definition 1. *A delay detector is a function*

$$\text{ALARM}(t) : U^t \times Y^t \to \{0, 1\}, \tag{2}$$

where U is the input domain and Y the measurement domain. The detector is

$$\text{ALARM}(t) \equiv \begin{cases} 0, & \pi((u_{k-1}, y_k)_{k=1}^t) > \xi \\ 1, & \pi((u_{k-1}, y_k)_{k=1}^t) \leq \xi \end{cases} \tag{3}$$

where π is a function of the data up to time t, ξ a threshold, and $\text{ALARM}(t) = 1$ denotes that a delay has been detected.

With this formalisation of the detection scheme, two quantities are needed: the function π and the threshold ξ. This paper focuses mainly on how to set π through IMM estimation, but a short discussion on setting the threshold is also provided in Section III-C.

The evaluation of the detector is based on when the first detection occurs, from here on denoted the time of detection, formally

$$\tau = \{\min t : \text{ALARM}(t) = 1\}. \tag{4}$$

It is well-established that there is a trade-off between accuracy and false alarm in detection problems [13], [14], and thus such a trade-off is also expected in this particular case. However, in time-series detection the standard accuracy measure is not well defined, and therefore the alternative performance metric of average detection delay (ADD) [13] is considered. Intuitively, this is the amount of time that detection takes, on average, after the event of interest occurs. In the current application, this is the time between a delay injection and the time of detection.

Definition 2. *The average detection delay (ADD) is*

$$\text{ADD}(\tau) = \mathbb{E}[\max(0, \tau - t_*)], \tag{5}$$

where $\tau = \{\min t : \text{ALARM}(t) = 1\}$ and t_* is the time of delay injection.

A false alarm on time series data is a detection before the event of interest occur, and with this notion the probability of false alarm (PFA) [13] is defined as follows.

Definition 3. *The probability of false alarm (*PFA*) is*

$$\text{PFA}(\tau) = \mathbb{P}(\tau < t_*), \tag{6}$$

where $\tau = \{\min t : \text{ALARM}(t) = 1\}$ *and* t_* *is the time of delay injection.*

The problem can then be summarised as finding π and ξ in Definition 1 such that PFA and ADD are kept small enough for the specific problem at hand.

III. Delay detection by Estimation

In this paper, the detection is performed by estimating the current delay in the system, and from this determining the *posterior probability that there is no delay* ($\delta_t = 0$) present. This probability constitutes the function π in Definition 1. Going forward, let $\pi_t^i \equiv \mathbb{P}(\delta_t = i | \{y_k, u_{k-1}\}_{k=1}^t)$, to simplify notation. Then the proposed detector is

$$\text{ALARM}(t) = \begin{cases} 0, & \pi_t^0 > \xi \\ 1, & \pi_t^0 \leq \xi \end{cases}. \tag{7}$$

To estimate the probability π_t^0 online, IMM filtering is applied.

A. IMM Filtering

IMM filtering is an extension of Kalman filtering (for details on the Kalman Filter, see [15]) to systems that switch between different *modes* over time. A common application of IMM is air-target tracking, where the modes can represent turns, constant velocity flight, or hovering [16]. This was also the motivation for its development [17]. Inspired by the great success of IMM for air target tracking, this paper adapts IMM to work for delay detection. This is done by tracking the system state by considering switches between delay sizes i, with two main differences to standard IMM: (i) the state estimation is performed in a delayed manner to capture current delays, and (ii) the modes use the same state space model with different inputs, as opposed to standard IMM where the inputs are the same and state space models change between modes. The main idea of IMM, interaction between multiple models through estimate mixing, remains.

In many applications the aim of IMM is to accurately estimate the states, but in the current application the goal is rather to calculate the evolving mode posterior probabilities π_t^i, to be used in the detector. However, these probabilities and the state estimates are closely related in IMM filtering. In standard IMM filtering the posterior probabilities are calculated as a means to receive state estimates, whereas in the application to delay detection the state estimates are only calculated as needed to update the posterior probabilities. This distinction has no impact on the IMM filtering method in itself.

The delay mode at time t is here denoted as $\delta_t = i$, $i \in \mathbb{N}$. The transitions between modes are described by a Markov model with *transition probabilities*

$$\mathbb{P}(\delta_t = i | \delta_{t-1} = j). \tag{8}$$

The conditional probability of the system being in mode i at time t is exactly the probability used by the detector (7), and updated dynamically by the IMM filter.

The notion of delay size as modes is believed to be novel and requires careful consideration of time indices, which goes beyond that of standard IMM. For a certain delay $\delta(t) = i$, the observation at time t depends on the state at time $t - i$, *not* on the state at time t. Thus, under $\delta_t = i$ time steps of delay, the state of the system is not observable until time $t + \delta_t$. To observe the full system, the observer needs to be delayed by the maximum considered delay N. This is formalised in Proposition 1.

Proposition 1. *Consider a system of the form* (1)*, with* $\delta_t = N$*. Then, the state at time* t *is not observable before time* $t + N$*.*

Proof: See appendix A.

By Proposition 1, it is clear that standard IMM can not be directly applied to the problem. Motivated by this, the state estimation is delayed by N time steps, , so that at time t the IMM filters estimate x_{t-N}.

Another necessary divergence from conventional IMM can be found by studying the state space model in (1). Namely, due to the possible unknown delay, y_t does not necessarily depend on x_t, see (1). This contradicts the assumptions for applying Kalman filters, a crucial part of the IMM filtering. The issue is resolved by considering a buffer of measurements $\{y_k\}_{k=t-N}^t$, and feeding the appropriate measurement to each mode. Specifically, the measurement y_{t-N+i} is fed to the filter corresponding to mode i. Let $\tilde{t} \equiv t - N$ denote the time for which state estimation is performed, and let $\tilde{y}_{\tilde{t}}$ denote the observations that *would* be obtained at time $\tilde{t}$ if no delay were present, i.e., $\tilde{y}_{\tilde{t}} = Cx_{\tilde{t}} + v_{\tilde{t}}$. Note that, under the hypothesis assigned to mode i that the delay is $\delta_t = i$, $\tilde{y}_{\tilde{t}} = y_{\tilde{t}+i}$.

Remark 2. *Going forward, both* t *and* $\tilde{t} = t - N$ *will be used, to denote the actual time and the time of estimation respectively.*

With this reformulation and routing of measurements from the buffer, the standard state space model for each mode is recovered as

$$\begin{cases} x_{\tilde{t}+1} &= Ax_{\tilde{t}} + Bu_{\tilde{t}} + w_{\tilde{t}} \\ \tilde{y}_{\tilde{t}} &= Cx_{\tilde{t}} + \nu_{\tilde{t}}. \end{cases} \tag{9}$$

By this, the difference between modes in the proposed version of IMM is not in the system dynamics A, B, C, but in the measurements used. Therefore, the stationary Kalman gain for all different modes is the same.

The overall structure of the IMM filter can be seen in Fig. 3. At each time step t, the observations $y_{\tilde{t}+i}$ and control actions $u_{\tilde{t}}$ are given as input to the Kalman filter of mode i, and the output is an updated state estimate $\hat{x}_{\tilde{t}}^i$ for each potential delay $\delta_t = i$, as well as updated probabilities of the different modes, π_t^i.

The final IMM algorithm can now be described in detail, and the full description is provided as Algorithm 1. At time

t, the first step of IMM is to *calculate mixing probabilities*. For each possible delay δ_t, this is the probability of each previous potential delay δ_{t-1} given the previous estimates and the system dynamics, e.g. for $\delta_t = i$ and $\delta_{t-1} = j$, the mixing probability is

$$\mathbb{P}(\delta_{t-1} = j | \delta_t = i, \mathcal{I}_{t-1}) = \frac{\pi_{t-1}^j \mathbb{P}(\delta_t = i | \delta_{t-1} = j)}{\mathbb{P}(\delta_t = i)}, \quad (10)$$

where $\mathcal{I}_t$ denotes the available information $\{u_{0:t-1}, y_{1:t}\}$; $\mathbb{P}(\delta_t = i | \delta_{t-1} = j)$ is the Markov model transition probabilities, taking on the role of prior; and π_{t-1}^j is given as output of the IMM filter at time $t-1$, thus implicitly depending on $\mathcal{I}_{t-1}$, and taking the role of likelihood. Intuitively, the mixing probabilities describe the *probability of the system being in a certain mode at the previous time step, given a hypothesis on the current mode*. The current state is of course not known, so these probabilities are calculated for all possible hypotheses on δ_t.

The next step is to use these mixing probabilities to weigh the previous estimates. Under the hypothesis $\delta_t = i$, an estimate on the previous state of the system is given as

$$\tilde{x}_{\tilde{t}-1|\tilde{t}-1}^i = \sum_{j=0}^{N} \mathbb{P}(\delta_{t-1} = j | \delta_t = i) \hat{x}_{\tilde{t}-1|\tilde{t}-1}^j, \quad (11)$$

where $\hat{x}_{\tilde{t}-1|\tilde{t}-1}^j$ denotes the estimate of the state at time $\tilde{t}-1$ given that the system is in mode j at time $t-1$, and $\tilde{x}_{\tilde{t}-1|\tilde{t}-1}^i$ denotes the estimate of the state at time $\tilde{t}-1$ given the assumption that the system is in mode i at time t. This definition of mixing is a key to the high performance observed in air target tracking applications [8].

Third, a Kalman filter is applied to each of the mixed estimates $\{\tilde{x}_{\tilde{t}-1|\tilde{t}-1}^i\}_{i=0}^N$, using the inputs $y_{\tilde{t}+i}$ and $u_{\tilde{t}-1}$, to return new estimates $\{\hat{x}_{\tilde{t}|\tilde{t}}^i\}_{i=0}^N$.

Independently to the state estimate update, the mode probabilities are updated based on the new data as

$$\pi_t^i = \frac{\mathbb{P}(y_{\tilde{t}} | \delta_t = i, \tilde{x}_{\tilde{t}-1|\tilde{t}-1}^i) \mathbb{P}(\delta_t = i | \pi_{t-1}^i)}{\mathbb{P}(y_{\tilde{t}})}, \quad (12)$$

where

$$\mathbb{P}(\delta_t = i | \pi_{t-1}^i) = \sum_{j=1}^{N} \mathbb{P}(\delta_t = i | \delta_{t-1} = j) \pi_{t-1}^j, \quad (13)$$

and $\mathbb{P}(y_{\tilde{t}} | \delta_t = i, \tilde{x}_{\tilde{t}-1|\tilde{t}-1}^i)$ calculated from observations.

B. Formulation of a Parsimonious Markov Model

To apply IMM to delay detection, a Markov model describing the potential transitions between delay modes is needed. This describes both which transitions are possible, and how probable they are. Known or assumed information about this behaviour can be encoded in the transition probabilities. For example, there could be physical limitations or other safeguards that restrict the ways delay can be injected, and this information could be encoded in the Markov Model by changing probabilities or disabling some transitions.

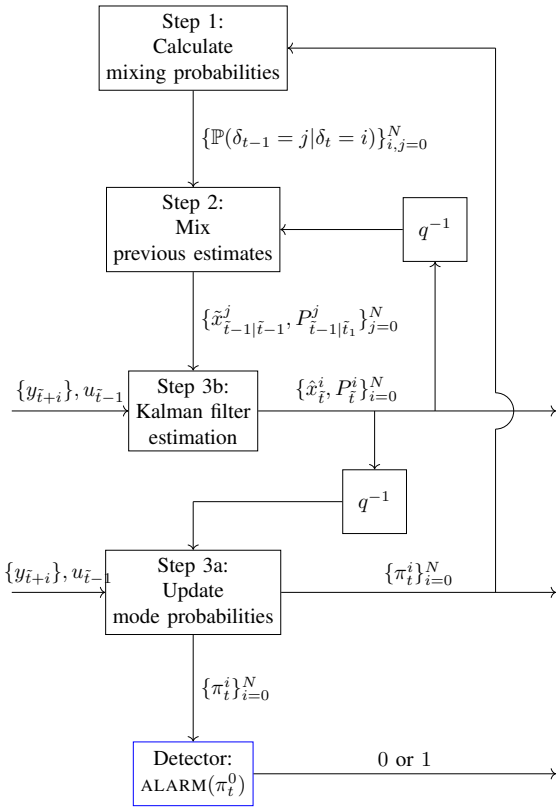

Fig. 3: Block diagram over IMM for delay detection. The blue box is the actual detection step. The operator q^{-1} denotes a unit delay shift operator.

However, when there are no physical limitations to the attack behaviour, a parsimonious model is more robust. Even if there were historical patterns on how delay attacks are implemented that could be encoded in the Markov model, detection methods optimised for these would encourage deviations from these patterns, thus invalidating the imposed assumptions. Therefore, this paper follows Assumption 1 and does not incorporate any detailed information into the Markov model.

When designing the Markov model, it is important to note that without any imposed structure, $N(N-1)$ parameters need to be set when considering N potential delays, by the law of total probability. To allow for informed parameter tuning, a smaller set of parameters is needed. Thus, a general Markov model is considered where the system without observations changes between delay modes $\delta_t > 0$ with uniform probability. This should not be viewed as a narrow assumption on system or attack behaviour, but rather as the most general prior that can be put on the transition probabilities, allowing the observed data to steer the estimation as much as possible. By the same reasoning, and standard IMM practice [8], transition probabilities are considered constant over time and independent of observations. With these simplifications,

Algorithm 1 IMM

Input: $(y_k, u_{k-1})_{k=\tilde{t}}^{t}$

Output: $(\pi_t^i, \hat{x}_{\tilde{t}}^i)_{i=0}^{N-1}$

Step 1: The mixing probabilities are calculated using Bayes' rule as

$$\mathbb{P}(\delta_{t-1}|\delta_t, \mathcal{I}_{t-1}) = \frac{\mathbb{P}(\delta_t|\delta_{t-1})\pi_{t-1}}{\mathbb{P}(\delta_t)}, \tag{14}$$

where $\mathbb{P}(\delta_t|\delta_{t-1})$ are the transition probabilities, and the denominator can be treated as a normalization.

Step 2: The previous estimates and covariances are mixed as

$$\tilde{x}_{\tilde{t}-1|\tilde{t}-1}^i = \sum_{\mathrm{j}=1}^{N} \hat{x}_{\tilde{t}-1|\tilde{t}-1}\mathbb{P}(\delta_{t-1} = j|\delta_t = i), \tag{15}$$

and

$$P_{\tilde{t}-1}^i = \sum_{j=1}^{N} P_{\tilde{t}-1}^j \mathbb{P}(\delta_{t-1} = j|\delta_t = i), \tag{16}$$

for each mode i, using the mixing probabilities from (14).

Step 3a: The new mode probabilities are calculated as

$$\pi_t^i = \frac{\mathbb{P}(y_{\tilde{t}}|\delta_t = i, \tilde{x}_{\tilde{t}-1|\tilde{t}-1}^i)\mathbb{P}(\delta_t = i)}{\mathbb{P}(y_{\tilde{t}})}, \tag{17}$$

where

$$\mathbb{P}(\delta_t = i) = \sum_{j=1}^{N} \mathbb{P}(\delta_t = i|\delta_{t-1} = j)\pi_{t-1}^j, \tag{18}$$

and $\mathbb{P}(y_{\tilde{t}}|\delta_t = i, \tilde{x}_{\tilde{t}-1|\tilde{t}-1}^i)$ is calculated from observations.

Step 3b: The new estimates are calculated, for each $i = 0, 1, 2, ...N$, by applying the Kalman filter to the mixed estimate distributions $\{\tilde{x}_{\tilde{t}-1}^i, P_{\tilde{t}-1}^i\}_{i=0}^N$ from step 2 and the observation matching the mode hypothesis, $y_{\tilde{t}+i}$.

the only transition probabilities that need to be set are

$$p_0 \equiv \mathbb{P}(\delta_t > 0|\delta_{t-1} = 0) = \sum_{j=1}^{N-1} \mathbb{P}(\delta_t = j|\delta_{t-1} = 0), \tag{19}$$

and

$$p_1 \equiv \mathbb{P}(\delta_t = 0|\delta_{t-1} > 0) = \sum_{j=1}^{N-1} \mathbb{P}(\delta_t = 0|\delta_{t-1} = j), \tag{20}$$

where p_0 is the probability of transitioning from zero delay to some delay, and p_1 the probability to transition from some delay to no delay. By Assumption 1, it is tempting to set $p_1 = 0$ to capture the knowledge that the delay injection attack is persistent. However, this is detrimental to the performance of the method as it leads to asymptotic detection when no attack is present, without observations. This behaviour of the IMM detector can be understood by analysing the stationary distribution of the Markov Model.

C. Setting Parameters via the Stationary Distribution

Independent of the initialisation, a Markov model always converges to some distribution. This distribution is the *stationary distribution* of the model.

The stationary distribution is a valuable tool in the analysis of the IMM detection scheme. As the aim is only to differentiate between there being a delay or not, the interesting part is the stationary distribution over these two (groups of) modes, see (21). This is fully determined by the stationary probability that no attack is present, from here on denoted $\overline{\pi}^0$.

To analyse the stationary distribution, consider the simplified Markov model resulting from (19) and (20), whose transition matrix can be represented as

$$\begin{bmatrix} 1-p_0 & p_1 \\ p_0 & 1-p_1 \end{bmatrix}. \tag{21}$$

As an example of how the stationary distribution can provide new insights, such analysis shows why setting $p_1 = 0$ is always a bad idea, no matter if this is the true nature of an attack. Specifically, this results in a stationary distribution of certain delay, which means no choice of ξ can save the detector from guaranteed false positives. This is formalised as Proposition 2.

Proposition 2. *Consider the Markov model* (21). *If* $p_0 > 0$ *and* $p_1 = 0$*, the stationary distribution is* $\overline{\pi}^0 = 0$.

Proof: It is easily verified that

$$\begin{bmatrix} 1-p_0 & 0 \\ p_0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \tag{22}$$

and by definition of a stationary distribution this directly gives $\overline{\pi}^0 = 0$.

The same principles can be used as a tool to design the Markov model parameters. The case of uninformative (or equivalently, missing) observations is considered. Then, the stationary distribution gives the probability of nominal mode $\overline{\pi}^0$ by:

$$\begin{bmatrix} 1-p_0 & p_1 \\ p_0 & 1-p_1 \end{bmatrix} \begin{bmatrix} \overline{\pi}^0 \\ 1-\overline{\pi}^0 \end{bmatrix} = \begin{bmatrix} \overline{\pi}^0 \\ 1-\overline{\pi}^0 \end{bmatrix}, \tag{23}$$

and solving for p_0 yields:

$$p_0 = p_1 \frac{1-\overline{\pi}^0}{\overline{\pi}^0}. \tag{24}$$

Using this relation between p_0 and p_1, the full Markov model can be determined by a desired stationary distribution and some domain knowledge of p_0. However, this does not guarantee a p_1 that is a probability, i.e. ≤ 1, and p_0 may not be known. Instead, it can be noted that if $\overline{\pi}^0 \geq 0.5$, which is a reasonable design choice in most applications, $\frac{1-\overline{\pi}^0}{\overline{\pi}^0} \leq 1$ and thus p_1 can be chosen freely in $[0, 1]$. Further, this allows interpreting p_1 as the sensitivity of the model to new observations. A low p_1, and thus a low p_0, implies

slow transitions between modes and thus more observations are needed to change the estimated probabilities. By instead using a p_1 close to 1, the system adapts quickly and ADD is minimised.

The specification of transition probabilities in (21) is now parametrized by the stationary probability $\overline{\pi}^0 \in (0,1)$ and the tuning parameter $p_1 \in (0,1)$.

By using this simplified Markov Model and the above reparametrisation, *the full Markov model of the IMM filter is parametrised by two variables, no matter how many delay modes are considered.* Without this simplification, $N(N-1)$ parameters would be needed to describe the model, all of which are hard to get from domain knowledge.

The choice of detection threshold of detectors is an open question in general, and this paper does not claim to solve it. Some guiding principles for setting the threshold is that the stationary distribution without attack, i.e. $\overline{\pi}_0$, should *not* trigger detection, while the stationary distribution under attack *should* yield detection. The latter can be estimated by simulation, and in the experiments of this paper this is estimated as the stationary distribution under an average delay attack, according to the Markov model. Tuning can also be done directly by visualisation of simulated data. More sophisticated methods to pick the threshold is left as future work.

IV. DELAY DETECTION IN CRUISE CONTROL

A. Cruise Control Model

A cruise control system is an engineering example of a dynamic system that is switched between open and closed loop settings. Due to the delay being injected in the feedback loop, the attack is in general not noticeable until the loop is closed. However, when the loop is closed, the attack is effective and may destabilise the system; see [3]. The importance of the example is motivated by the increased usage of cruise control systems, and the potentially disastrous outcomes of destabilisation. Therefore, such attacks need to be detected before the loop is closed. A linearised state space model for the car dynamics is

$$\begin{cases} x_t = \Delta x_{t-1} + u_{t-1} + w_t \\ y_t = x_t + \nu_t \end{cases}, \tag{25}$$

where Δ describes the expected amount of speed lost in one time step. This model can be obtained from a non-linear physics-based model. Such a derivation can be found in [3].

In all experiments, to mimic manual control by a human driver, a simple proportional controller is used, which directly feeds the error signal, without delay, as its control action, with an imposed saturation of $|u| < 5$.

The attack is implemented by storing both the true observations (the observations that would have been seen if there where no attack), and the actual observations. These are equal until the time of attack, but when a delay of δ_t is injected the actual observation at time t instead equals the true observation at time $t-\delta_t$, similar to the previous intuition from Fig. 2. Depending on the experiment, the attack time t_* is either fixed or uniformly random on the time scale used.

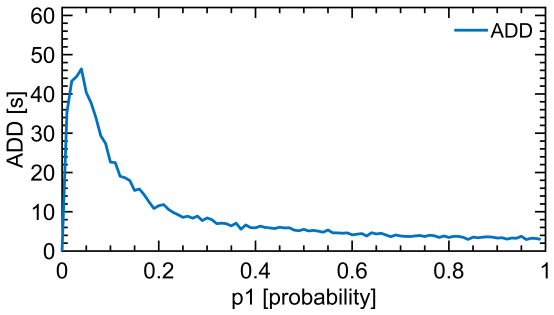

Fig. 4: ADD depending on the tuning parameter p_1. The plot shows the average over 50 runs, with the detection threshold fixed at 0.4, and the delay of 0.2 seconds always being added after 250 seconds.

As model parameters, $\Delta = 0.99$ and independent $w_t \sim \mathcal{N}(0,1)$ and $\nu_t \sim \mathcal{N}(0,0.01)$ are used. The variance of the observations needs to be lower than that of the model to obtain good performance, since detection based on observations can only happen if the observations are significantly more trustworthy. This is a realistic assumption in the cruise control example, as the velocity state can be measured with the Global Positioning System (GPS) with high accuracy, while e.g. steep hilly terrain may significantly deviate from the general car model. Uncertainty in the model specification is also captured by the system noise.

The system works over time increments corresponding to 0.1 seconds, and there are IMM filters for delays of $0, 1, 2, ..., 10$ time steps, i.e. between 0.0 and 1.0 seconds, where the 0.0 second delay filter corresponds to the nominal mode.

The initial estimates are the true state, and the simulations are started with a speed of 60km/h and a reference speed of 60km/h. The Kalman filters are designed using the stationary Kalman gain [15]. The ADD and PFA are estimated from simulations by Monte Carlo estimates of the probabilities and expectations, with some fixed probability distribution of the attack point t_*.

B. Tuning of Hyper Parameters

In this paper, a strong prior belief of 90% chance of the nominal mode and equal chance for the different delays is used as initialisation. The transition probabilities are chosen to guarantee that the stationary distribution with no (or equivalently, uninformative) observations is the same as the initial prior (i.e., $\mathbb{P}(\delta_0 = 0) = \pi^0 = 0.9$), so that any change in the distribution can be interpreted as an actual change in probability based on the observations.

By applying (19) to the desired stationary distribution (90% probability of nominal mode), the transition p_0 is given by

$$p_0 = p_1 \frac{1-\overline{\pi}^0}{\overline{\pi}^0} = p_1 \frac{0.1}{0.9} \approx 0.11 p_1 \tag{26}$$

with $0 \leq p_1 \leq 1$. The tuning of p_1 is done experimentally. The threshold is fixed to $\xi = 0.4$, and the attack is always

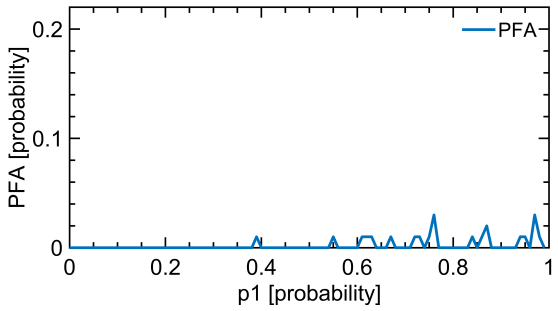

Fig. 5: PFA depending on p_1. The plot shows the average over 50 runs, with the detection threshold fixed at 0.4, and the delay of 0.2 seconds always being added after 250 seconds.

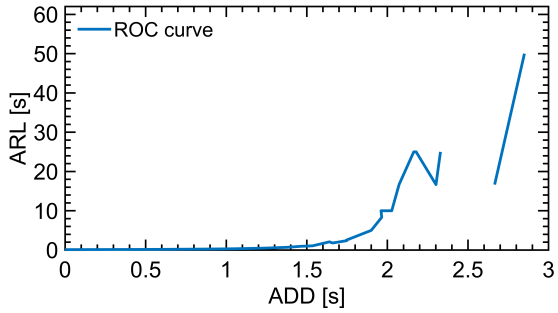

Fig. 6: Empirical ROC curve of ADD and ARL $\equiv 1/$PFA in seconds, with the threshold for detection varied between 0.01 and 0.99. The delay is of 0.2 seconds and introduced after 20s.

applied after 250 seconds, introducing a delay of 0.2 seconds. In Figs. 4 and 5, the ADD and PFA are plotted as functions of p_1, which is varied between 0.0 and 0.99. The metrics are calculated by averaging over 50 runs for each value of p_1. It can be seen that choosing p_1 is strongly related to finding a suitable trade-off between false alarm and detection delay. However, with the current threshold PFA is less than 0.5% for all p_1. Thus, for the rest of the experiments a high $p_1 = 0.99$ is implemented to minimise the ADD, although it is noted that a smaller p_1 is useful in cases where one would like to further decrease the PFA.

Experimentally, the mean likelihood after an attack is introduced is estimated over 50 runs to be 0.20, with a 99th quantile of 0.28. Using this to calculate the stationary distribution under attack and the intuition that the threshold ξ should be no smaller than this, it is concluded that the detection threshold should be in the interval $(0.28, 0.9)$. Since the probabilities vary around these stationary distributions, some additional safety-margin should be included. Prioritising a lower false alarm probability, a threshold of 0.4 is chosen.

In Fig. 6, an empirical ROC curve for the detector is shown. As suspected, a low average detection delay corresponds to a lower average run length (ARL $\equiv \frac{1}{\text{PFA}} \cdot \frac{\text{time steps}}{s}$), and the trade-off is relatively beneficial.

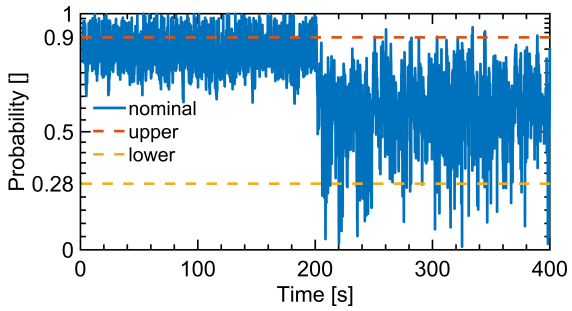

Fig. 7: A single trajectory of how the nominal probability change over time, using $p_1 = 0.99$ and a delay attack of 0.2 seconds being introduced after 200 seconds. The upper and lower bound for reasonable threshold from the estimated range $(0.28, 0.9)$ are shown for reference.

C. Simulation Results

All simulation results consider a delay attack defined as a step function, with $\delta_t = 0$ until some attack time t_*, when a fixed delay is inserted.

In Figure 7, a single trajectory of the IMM filter estimate is shown. It is clear from the figure that the filter is quick to adjust the probabilities when the attack is introduced. A threshold of 0.4 is supported by this trajectory as well, as it is clear that it quickly detects the attack without any false alarms, in this instance.

In Table I, the estimated ADD and PFA are shown after running the model for 5000 runs for injected delays of $0.1, 0.2, ..., 1.2$ seconds. The attack time is sampled uniformly random between 2 and 302 seconds. It should be noted that the two largest delays in this simulation, 1.1 and 1.2 seconds, are higher than any of the hypotheses used by the model, and the detector performs slightly worse for these. The performance is still good though, which suggests that detection is possible even for attack sizes that were not anticipated. The main outlier is the 0.1s delay, which takes on average over 90 seconds to detect. It is hypothesised that this delay has a very minor effect on the observations, which makes it hard to detect. Understanding this better and improving the method for small delays is left as future work. Provided that the cruise controller is designed to be robust against such low delays, for example as in [3], a high detection time for low delays is not a problem in practical applications.

It can be noted that the implemented attack behaviour (i.e., as a persistent delay injected from time t_*) differ significantly from the implemented Markov model. Therefore, the good performance further supports the usage of a robust parsimonious Markov model.

D. Comparison with Existing Detection Schemes

Previous work on delay detection builds on identification of the system parameters. In [3], an identification scheme is applied to the same vehicle model as in this paper, but assuming unknown system dynamics. This gave a detection time

TABLE I: Estimated ADD and PFA over 500 runs, for varied delays attacks inserted at a uniformly random time.

Delay [s]	ADD [s]	PFA [probability]
0.1	92.9	0.0055
0.2	3.16	0.0052
0.3	3.13	0.0052
0.4	3.17	0.0050
0.5	3.15	0.0040
0.6	3.21	0.0046
0.7	3.15	0.0052
0.8	3.15	0.0054
0.9	3.12	0.0052
1.0	3.15	0.0072
1.1	3.67	0.0048
1.2	4.37	0.0060
Average	10.78	0.0053
Average excl. 0.1	3.31	0.0053

exceeding 100 seconds on an example trajectory, which is significantly worse than the results for the IMM framework. They are not fully comparable, since the IMM approach assumes known model dynamics, but it shows that much better performance is possible when the dynamics are known. With small uncertainties in the model, the IMM model is believed to be able to handle them as long as the threshold is chosen with this uncertainty in mind.

It is harder to compare the current work to that of [6], since those experiments are not performed on the same application and do not use a fixed-threshold detector. However, using that method, it is possible to note a difference in the distribution after around 10 seconds, slightly worse than the IMM method, but only if the delay is inserted momentarily. A gradually deployed attack is not noticeable since the method assumes a nominal mode for the previous time step so any uncertainty of previous modes are discarded. This issue should not be present in IMM, since the uncertainties of the mode from past time steps are carried over to future steps.

V. CONCLUSIONS

The paper described how Interacting Multiple Model filtering can be used to detect delay attacks in the feedback path. By derivation of stationary distributions, both the transition probabilities and threshold could be tuned in an informed manner. The tuned model detected delay attacks in a few seconds on a simulation of a cruise control application. Interesting directions for future work is to find theoretical bounds on the false alarm probability given the threshold, or to detect delay attacks in closed loop settings using the IMM approach.

APPENDIX

A. Proof of Proposition 1

Proof: Let

$$\mathbf{x}_t = \begin{bmatrix} x_t & x_{t-1} & ...x_{t-N} \end{bmatrix}^T.$$

Using this, (1) can be expressed as a proper state space model with y_t depending on $\mathbf{x}_t$, with new system dynamics

$$\mathbf{A} = \begin{bmatrix} A & 0 & ... & 0 & 0 \\ I & 0 & ... & 0 & 0 \\ & \cdot & & & \\ & & \cdot & & \\ & & & \cdot & \\ 0 & 0 & ... & I & 0 \end{bmatrix} \text{ and } \mathbf{C} = \begin{bmatrix} 0 & ... & C \end{bmatrix}.$$

Then, the observability matrix is

$$\begin{bmatrix} 0 & ... & 0 & C \\ 0 & ... & 0 & 0 \\ & \vdots & & \end{bmatrix},$$

so only state x_{t-N} is observable.

REFERENCES

[1] E. Fridman, *Introduction to Time-delay Systems: Analysis and Control*. Basel, Switzerland: Birkhäuser, 2014.

[2] J. P. Hespanha, P. Naghshtabrizi and Y. Xu, "A survey of recent results in networked control systems", *Proceedings of the IEEE*, vol. 95, no. 1, pp. 138-162, Jan. 2007.

[3] T. Wigren and A. Teixeira, "Feedback path delay attacks and detection" in Proc. *IEEE Conf. on Decision and Control*, Bucharest, Romania, 2023, pp. 3864-3871.

[4] A. Abbasspour, A. Sargolzaei, M. Victorio, and N. Khoshavi, "A neural network-based approach for detection of time delay switch attack on networked control systems", *Procedia Computer Science*, vol.168, pp. 279-288, Nov. 2019.

[5] G. Bianchin and F. Pasqualetti, "Time-Delay Attacks in Network Systems" in *Cyber-Physical Systems Security*, Cham.: Springer, 2018, pp. 157-174.

[6] E. Korkmaz, M. Davis, A. Dolgikh, and V. Skormin, "Detection and mitigation of time delay injection attacks on industrial control systems with PLCs" in *Computer Network Security*, Cham.: Springer, 2017, pp. 62-74.

[7] X. Lou, C. Tran, R. Tan, D. K. Y. Yau, Z. T. Kalbarczyk, A. K. Banerjee, and P. Ganesh, "Assessing and mitigating impact of time delay attack: Case studies for power grid controls", *IEEE Select. Areas Commun.*, vol. 38, no. 1, pp.141-155, Nov. 2020.

[8] Y. Bar-Shalom, X.R. Li, and T. Kirubarajan, *Estimation, Tracking and Navigation: Theory, Algorithms and Software*. Hoboken, NJ: Wiley, 2001.

[9] Y. Bar-Shalom, X.-R. Li, and T. Kirubarajan, "IMM estimator versus optimal estimator for hybrid systems", *IEEE Trans. Aerosp. Electron. Syst.*, vol. 41, no. 3, pp. 986-991, Jul. 2003.

[10] 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.

[11] 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*, Yantai, China, 2011, pp. 2384-2389.

[12] A. E. Bryson, Jr. and Y.-C. Ho, *Applied Optimal Control—Optimization, Estimation and Control*. New York, NY: Hemisphere, 1975.

[13] V. V. Veeravalli and T. Banerjee, "Quickest change detection" in *Academic Press Library in Signal Processing: Volume 3*, Elsevier, 2014, pp. 209-255.

[14] M. Basseville and I. V. Nikiforov, *Detection of Abrupt Changes: Theory and Application*, Hoboken, NJ: Prentice-Hall, 1993.

[15] T. Kailath, A. H. Sayed, and B. Hassibi, *Linear Estimation*, Pearson Education, England, 2000.

[16] T. Kirubarajan and Y. Bar-Shalom, "Kalman filter versus IMM estimator: when do we need the latter?", *IEEE Trans. Aerosp. Electron. Syst.*, vol. 39, no. 4, pp. 1452-1457, 2003.

[17] H. A. P. Blom, "An efficient filter for abruptly changing systems", in *Proc. IEEE Conf. on Decision and Control*, Las Vegas, NV, pp. 656–658, 1984.