

Exploring the Random Impulses Vehicle Trajectory Model for Dimensionality Reduction and Motion Extraction from Aerial Videos

Thinh Hoang Dinh, *Student Member, IEEE*, Vincent Martinez, Pierre Maréchal, and Daniel Delahaye

Abstract—Intelligent Transportation Systems (ITS) hold great potential for improving the safety, efficiency, and comfort of road users. These applications heavily rely on processing trajectory data, a significant component of which is the road vehicle trajectory model. This paper presents a kinematic statistical model for large-scale ITS that is more suitable for batch processing of data, especially in systems with limited hardware capabilities such as vehicular communication modems and embedded systems.

Our proposed model approximates slipping velocity and acceleration as random impulses originating from stationary distributions. After validation, we present two applications of this model.

First, our derived Functional Principal Component Analysis (FPCA) basis allows for effective dimensionality reduction, a key step for various machine learning tasks such as trajectory prediction and anomaly detection. Second, we apply the model to filter and smooth trajectories captured from aerial videos. We achieve this through a combination of object detection using a deep neural network, and a Two-Filter scheme, which includes a Gaussian Mixture Kalman Filter (GMKF) for forward filtering, and a Backward Information Filter for smoothing. Results on the NGSIM 101 dataset demonstrate that highway trajectories can be significantly reduced to vectors with very few dimensions for the first application. For the second application, the combination of filtering and smoothing yields a smoothed estimate of velocity with a confidence interval up to two times smaller than that achieved by filtering alone.

We believe that the proposed models are suitable for implementing stringent-constrained data processing at the edges.

Index Terms—ITS, road trajectories, Random Impulses model, FPCA, motion extraction, Kalman Filter, Information Filter.

I. INTRODUCTION

THE newest generation of vehicles and roadside units, equipped with sensors and communication systems, generate a substantial amount of data, which holds vast potential for improving vehicular safety and urban traffic infrastructure efficiency. Consequently, processing this data is crucial for realizing intelligent transportation, as its successful analysis could assist in traffic congestion control [1], promote autonomous driving [2], and enhance passenger comfort. Nu-

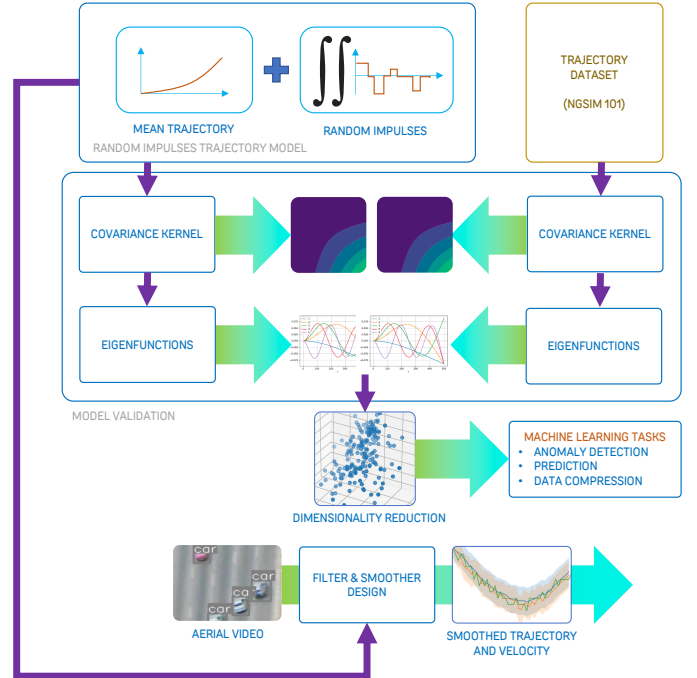


Fig. 1: Overview of the paper: we introduce novel signal models for road vehicle trajectories and validate these models through the computation of covariance kernels. The eigenfunctions of these kernels can be used for dimensionality reduction, a process that plays a critical role in several machine learning tasks. Lastly, we present a filtering and smoothing framework based on these models, which is applied to the task of trajectory extraction from aerial videos.

merous trajectory models have been proposed in the literature, including car-following and lane-changing models [3], [4], in addition to deep learning generative models [5]–[7]. Termed as *interaction-aware* [8], these models incorporate inter-vehicle interactions, often requiring substantial computational resources. They are ideally suited for tasks such as path optimization and collision avoidance, where performance evaluation is primarily assessed from the standpoint of each individual vehicle. Conversely, *kinematic models*, despite receiving less attention, offer an alternative approach. While these models are typically simpler due to their lack of explicit modeling of inter-vehicle interactions, their application to collective tasks, such as optimizing data transmission in vehicular network or batch trajectory smoothing at scale, often leads to

Thinh Hoang Dinh is with Artificial and Natural Intelligence Toulouse Institute (ANITI), Université de Toulouse, Toulouse, France.

Vincent Martinez is with NXP Semiconductors Toulouse, Toulouse, France.

Pierre Maréchal is with Toulouse Institute of Mathematics, Toulouse, France.

Daniel Delahaye is with École Nationale de l'Aviation Civile, Toulouse, France.

Manuscript received April 19, 2021; revised August 16, 2021.

converging *average* performance with *interaction-aware models*. For instance, when designing a vehicular communication modem, predicting all surrounding vehicle trajectories using an optimality solver that accounts for all relative distances and velocities to optimize transmission parameters may not be necessary. In this particular scenario, a simpler model performing well *on average* would be adequate, despite the possibility of its performance being less optimal for certain vehicles compared to others.

The latter model category is valuable for its ability to handle large-scale trajectory information. Distinguished by its simplicity, ease of use, and minimal hardware requirements, it is optimally suited for edge computing applications, such as in smart traffic lights or vehicular communication modems. In the emerging landscape of urban ITS, such as Vehicle-to-Everything (V2X) networks and smart road infrastructure, the accessibility of such a model becomes crucial. It can considerably enhance the design of safe and efficient applications, underlining its transformative potential in this rapidly evolving field.

This paper presents the “Random Impulses” trajectory model, tailored to meet the specific requirements of a kinematic statistical model in batch data processing contexts. At its core, this model perceives individual vehicle trajectories as variations of a “mean trajectory”. This mean trajectory represents the safety distance-keeping by drivers, coupled with a slow-varying random component that captures the influence of various factors considered to be Markovian. This approach simplifies the interactive effects produced by surrounding vehicles, while preserving the simplicity of the more basic kinematic models that assume constant velocity or acceleration [9]–[11]. The model’s simplicity and computational efficiency make it particularly attractive for deployment in large-scale ITS, especially those operating under hardware constraints.

The considerable value of this model becomes more apparent in the subsequent sections of this paper, where we demonstrate its applications through various application frameworks related to the Karhunen-Loève Transform (KLT) - also known as FPCA - and an approximated linear state-space model. The KLT has strong ties to Principal Component Analysis (PCA), a technique that has found widespread use in the machine learning community for data dimension reduction. The intricate nature of the lateral impulses distribution invites the use of a Gaussian Mixture implementation for the filter and smoother. Achieving these applications with car-following or constant velocity models would have posed substantial challenges, which likely contributes to their limited implementation in edge processing facilities. This stark contrast underscores the innovative contribution of the “Random Impulses” model to the field.

Furthermore, the field of high-resolution trajectory extraction from aerial videos has also witnessed tremendous developments, with a special mention of the NGSIM dataset [12]. This method has inspired other approaches, such as using video streams from Unmanned Aerial Vehicles (UAVs) for traffic monitoring [13]–[16]. Kalman and Particle Filters have been widely used in vehicle tracking models such as SORT [17] and its variants [18]–[20]. Nevertheless, the process and

measurement models implemented in these trackers often use random walk models without justification. Our paper also provides the reasoning behind, and the corresponding implementation reference for these feature trackers by using the GMKF for the lateral component. We go a step further and propose a smoother for offline processing of video data, leading to better estimates thanks to the assimilation of future measurements compared to filtering-based solutions alone.

Fig. 1 provides the overview of the paper. It unfolds as follows: Section II reviews various trajectory models and relevant literature on trajectory estimation from aerial videos. Section III introduces the “Random Impulses” Model and Section IV justifies its validity through the analytical derivation of the covariance kernel. Section V expounds on the model’s eigenfunctions for trajectory dimensionality reduction, a crucial component for anomaly detection in Advanced Driver Assistance Systems (ADAS). Section VI showcases the model’s secondary application in filtering and smoothing trajectory data from aerial videos. Section VII that follows will provide various discussions around the assumptions and the results, and the paper concludes with Section VIII.

II. RELATED WORK

Lefevre *et al.* [8] categorized trajectory models into three types: physics-based, maneuver-based, and interaction-aware. Interaction-aware models, including car-following and lane-changing models, define motion as functions of surrounding vehicles’ movement [21]–[24]. Studies also considered ego-motion due to multi-agent influences using Dynamic Bayesian Networks [25]. Conversely, maneuver-based models presume a finite number of “maneuvers” that drivers can perform, typically learned from a clustering algorithm or modeled by a Gaussian Process [26], [27].

Deep Generative models were extensively explored as well. Chandra *et al.* [28] utilized a hybrid LSTM-CNN to model road agents’ dynamics, shapes, and behaviors. Their model could predict multiple agent behaviors simultaneously. However, the inherent fading gradient problem of RNNs led to the introduction of the attention mechanism [29]. Messaoud *et al.* [30] incorporated global and partial attention to surrounding vehicles into their model. Despite their success, these models were often considered too complex for trajectory data processing in systems with limited computational power.

Kinematic models, like those assuming constant velocity or acceleration [9]–[11], [31], align closest with the Random Impulses model. These models are frequently chosen for applications involving Kalman filters but their utility for prediction is limited due to disregarding road curvature and distance-keeping tendencies.

While these models are useful for object tracking applications, they provide deterministic trajectories without a confidence interval. For instance, several trackers like SORT [17]–[20] extensively used Kalman Filter and its variants. Yet, these models are suboptimal for batch offline applications. Chen *et al.* [15] attempted to smooth trajectories using wavelets but without maintaining the same probabilistic framework for tracking. Furthermore, like other methods, they didn’t provide confidence intervals.

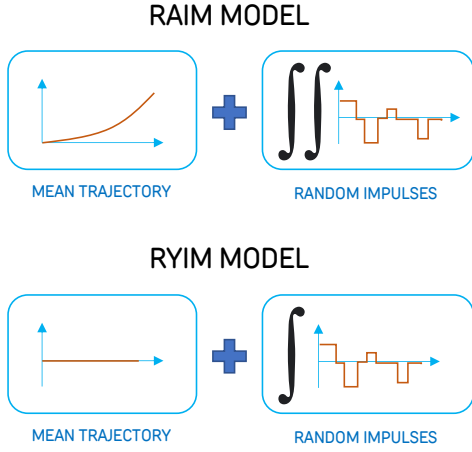


Fig. 2: An overview of the RAIM and RYIM Models for longitudinal and lateral components of the trajectory time series. The “mean” component represents the self-regulatory effect of distance-maintaining behavior for safety assurance. The *Random Impulses* model the effects of various stochastic factors.

Learning-based models, though sophisticated, are typically computationally intensive, and performance guarantees cannot be made due to the impossibility of performing thorough analysis. On the other hand, while kinematic models are well-understood, they often lack the predictive power required for advanced ADAS applications. The Random Impulses model addresses this gap by augmenting kinematic models with a component that mirrors self-regulatory distance-keeping behavior. The result are models that retain the simplicity inherent to kinematic models, while introducing an additional component (the mean trajectory) for greater adaptability. The careful balance of these trade-offs is confirmed through the validation process. These models pave the way for new applications, several of which are presented in the subsequent sections.

III. RANDOM IMPULSES MODEL

In this section, we formulate the Random Impulses Models for both the longitudinal and lateral components of the vehicle trajectory. The former represents the “fast-slow” characteristics of speeding, while the latter characterizes lane-changing behaviors. For reasons that will become apparent, we will refer to these models as the Random Acceleration Impulses Model (RAIM) and the Random Yawing Impulses Model (RYIM), respectively (Fig. 2). Consider the trajectory of vehicle i , denoted as $W_t^{(i)}, Y_t^{(i)}$. Here, $W_t^{(i)}$ specifies the vehicle’s relative position with respect to the left roadside, and $Y_t^{(i)}$ indicates the longitudinal distance the vehicle has traveled into the segment (Fig. 3a). We conceptualize each vehicle’s trajectory as an independent realization of two corresponding random processes, Z_t and X_t . These processes will be defined in Equations (1) and (5).

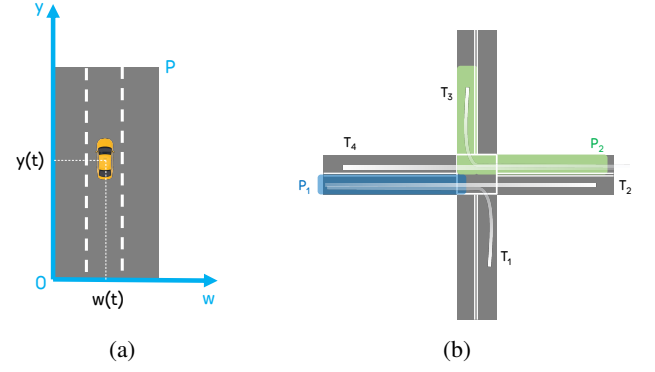


Fig. 3: (a) Definition of the P -segment coordinate system. W_t, Y_t are relative coordinates, and absolute coordinates require road topology be taken into account. (b) An example of two road segments P_1 - a rectangle, and P_2 - an L-shape road segments, which are shaded polygons at an intersection. Here, both trajectories T_1 and T_2 are considered valid for the road segment P_1 , since vehicles enter and leave at the same places. However, for road segment P_2 only trajectory T_3 is considered valid because T_4 leaves P_2 prematurely.



Fig. 4: All vehicles in the segment (blue) are considered to be following a virtual vehicle (green). If the car-following is exact, the separation (distance) between each vehicle is maintained, which is also the optimal scenario.

A. Random Acceleration Impulses Model (RAIM)

Consider a road segment with several lanes, all directing vehicles in the same direction. While vehicles might temporarily stop within the segment due to, for example, a traffic light, we simplify our model by assuming all vehicles enter and exit the segment at the same location (Figure 3b).

To avoid a collision, the motion of any vehicle entering a road segment is influenced by nearby cars. Similarly, the movement of these proximate vehicles also impacts the motion of other neighboring cars. This interaction results in a complex set of interlocking equations describing vehicle relationships without an easy solution. However, for simplicity, let’s consider all vehicles in a given road segment as following a “virtual vehicle” in a “leisurely” manner. By “leisurely,” we mean that the process of following is not exact. Nevertheless, any vehicle whose motion deviates substantially from the trajectory of the virtual vehicle runs an increased risk of colliding with other cars, as illustrated in Fig. 4. Formally, let $\{Y_t^{(i)}\}_{i=1}^{N_v}$ be the set of N_v longitudinal trajectories of vehicles traveling on the road segment P , $Y_t^{(i)} \in \mathbb{R}^+$. Let \bar{Y}_t be the trajectory of the virtual vehicle. Statistically, we consider $\{Y_t^{(i)}\}_{i=1}^{N_v}$ to be independent sample paths of a stochastic process:

$$\{X_t \triangleq X(t, \omega)\}, \quad (1)$$

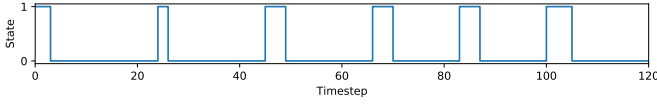


Fig. 5: One sample path of \mathcal{L}_t with renewal intervals $I_{0,k}$ given by $\Gamma(\alpha = 5, \beta = 0.5)$ and $I_{1,k}$ given by $\Gamma(\alpha = 30, \beta = 0.5)$. $\mathcal{L}_t = 1$ marks the period where there is acceleration activity, and $\mathcal{L}_t = 0$ otherwise. There are 11 regeneration points in this exemplary sample path located at $t \in \{3, 22, 24, 46, 48, 72, 75, 85, 88, 100, 105\}$.

where the sample function is defined as:

$$X(\cdot, \omega) : \mathbb{R}_+ \rightarrow \mathbb{R},$$

over the probability space (Ω, \mathcal{A}, P) indexed by $t \in \mathbb{R}_+$. The RAIM model assumes:

$$\mathbb{E}[X_t] = \bar{Y}_t,$$

and the “residual deviation” of the trajectory is generated from doubly integrating path-wisely the random acceleration impulses model A_t :

$$X_t = \mathbb{E}[X_t] + \int_{s=0}^t \int_{\tau=0}^s A_\tau d\tau ds. \quad (2)$$

We now describe the process A_t . For each $k \in \{0, 1\}$, let $\{I_{k,i}\}_{i \in \mathbb{N}^*}$ be a sequence of independent random variables (with respect to i), called renewal intervals associated with the state k , given by distribution functions:

$$F_k(u) = P(I_{k,i} \leq u), \quad k \in \{0, 1\}, \quad i \in \mathbb{N}^*.$$

Equipped with two sequences of renewal intervals corresponding to state $k = 0$ and $k = 1$, we define a sequence of regeneration points corresponding to moments where there is a jump from state 0 to state 1 and vice-versa:

$$W_I = \{\omega_i\}, \quad i \in \mathbb{N}^*,$$

such that:

$$\omega_i = \begin{cases} \omega_{i-1} + I_{1, \frac{i+1}{2}} & \text{if } i \text{ is odd,} \\ \omega_{i-1} + I_{0, \frac{i}{2}} & \text{if } i \text{ is even.} \end{cases}$$

This allows us to formally define a Markov renewal process \mathcal{L}_t , which is the *acceleration activity process* \mathcal{L}_t as follows:

$$\mathcal{L}_t = \sum_{i=1}^{+\infty} (\mathbf{1}_{\omega_i \leq t \leq \omega_{i+1}} \times \delta_{\{i \equiv 0[2]\}}). \quad (3)$$

Informally, the acceleration activity process is composed of alternating periods of idleness and acceleration. The lengths of these periods are determined by the renewal intervals of $I_{0,\cdot}$ and $I_{1,\cdot}$ respectively (Fig. 5). The RAIM represents this physical reality, suggesting that drivers following a virtual vehicle but intermittently apply and release the gas or brake pedals over random periods of time, at random times. This process results in the vehicle’s motion deviating from that of the virtual vehicle.

Building upon (3), the RAIM model further assumes that acceleration impulses are *identically independent* samples from

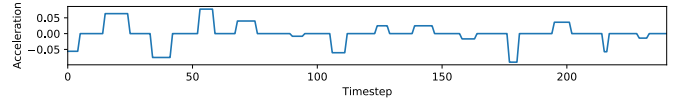


Fig. 6: One sample path of A_t with renewal intervals $I_{0,k}$ given by $\Gamma(\alpha = 5, \beta = 0.5)$, $I_{1,k}$ given by $\Gamma(\alpha = 30, \beta = 0.5)$ and $X_k \sim \mathcal{N}(0, 0.05^2)$.

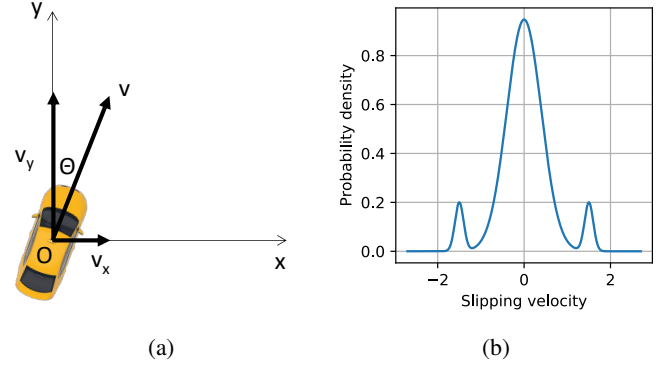


Fig. 7: (a) Illustration of the yawing angle θ , the lateral velocity v_x and the longitudinal velocity v_y . (b) Probability density of the Slipping Velocity Distribution with three modes indicating changing to the left lane, lane keeping and changing to the right lane.

a *stationary* distribution function $F_a(u)$, resulting in the acceleration impulses model A_t :

$$A_t = \sum_{i=1}^{+\infty} (M_i \times \mathbf{1}_{\omega_i \leq t \leq \omega_{i+1}} \times \delta_{\{i \equiv 0[2]\}}), \quad M_i \sim F_a(u) \quad (4)$$

A path sample is shown in Fig. 6. It is also worth mentioning that the RAIM model is a particular case of the more general Piecewise Markov Process and the Markov Renewal Process. Other interesting properties, including expended and remaining times in a segment, can be further found in [32].

B. Random Yawing Impulses Model (RYIM)

We now shift our focus onto the lateral component of the trajectory. As before, we assume that the set of lateral trajectories $\{W_t^{(i)}\}_{i=1}^{N_v}$ are independent path samples of a stochastic process Z_t . The RYIM model assumes:

$$Z_t = \mathbb{E}[Z_t] + \int_{s=0}^t B_s ds. \quad (5)$$

Introduce the coordinate system as depicted in Fig. 7(a), where Oy is aligned with the centre marking. With a small yawing angle θ , it follows that $v_x \approx v\theta$ and $v_y \approx v$. The dynamics of θ can be attributed to the driver’s steering wheel input, and this gives a physical interpretation to the RYIM model. As a vehicle navigates a road, small adjustments to the steering wheel by the driver keep the vehicle in its lane, leading to θ oscillating around zero. Occasionally, a significant shift in θ might be observed when the driver decides to change lanes.

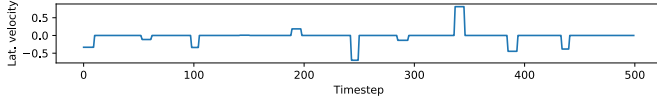


Fig. 8: The slipping impulses generated by process B_t with renewal intervals $I_{0,k}$ given by $\Gamma(\alpha = 30, \beta = 0.25)$, $I_{1,k}$ given by $\Gamma(\alpha = 150, \beta = 0.25)$ and $R_i \sim \mathcal{N}(0, 0.5^2)$.

In comparison with the RAIM model, there are still 2 states $\{0, 1\}$ which correspond to lane-keeping and lane-changing respectively. The only difference is that Z_t only has one integration of the random impulses process rather than two, as shown in (2).

As above, the “slipping activity process” \mathcal{S}_t is defined as:

$$\mathcal{S}_t = \sum_{i=1}^{+\infty} (\mathbf{1}_{\omega_i \leq t \leq \omega_{i+1}} \times \delta_{\{i \equiv 0[2]\}}). \quad (6)$$

The slipping impulse model is then defined as

$$B_t = \sum_{i=1}^{+\infty} R_i \times \mathbf{1}_{\omega_i \leq t \leq \omega_{i+1}} \times \delta_{\{i \equiv 0[2]\}} \quad \text{in which} \quad R_i \sim F_{lc}(u). \quad (7)$$

The distribution F_{lc} dictates the behavior of drivers in terms of lane-keeping and lane-changing, while the distributions $I_{.,k}$ manage the frequency and reaction rates of drivers. The sample paths of the slipping activity process, denoted as \mathcal{S}_t , are shown in Fig. 5, and of B_t are presented in Fig. 8, with its corresponding F_{lc} showcased in Fig. 7(b).

IV. VALIDATION OF THE MODEL THROUGH COMPUTATION OF THE COVARIANCE KERNEL AND ITS EIGENFUNCTIONS

The RAIM and RYIM models operate on the assumption that the residual components of the trajectory time series, once the mean trajectories have been subtracted, originate from a Markov process. However, the validity of such assumptions still needs verification. To address this, we analytically derive the covariance kernel of the processes X_t, Z_t and the corresponding eigenfunctions. The stationary process assumption could be deemed valid if the convergence of the covariance function can be established via the law of large numbers.

A. RYIM Model's Covariance Kernel and Eigenfunctions

We study the time integration of B_t in Eq. (7). Analysis for the acceleration process (4) should be similar.

$$B_t = \sum_{i=1}^{+\infty} R_i \times \mathbf{1}_{\omega_i \leq t \leq \omega_{i+1}} \times \delta_{\{i \equiv 0[2]\}}. \quad (8)$$

Proposition 1. *The covariance kernel for the process Z_t defined in (5) converges to the covariance kernel for the Wiener process if the regeneration time $I_0 + I_1$ goes to zero.*

Proof. The RYIM model assumes:

$$\begin{aligned} Z_t &= \mathbb{E}[Z_t] + \int_0^t B_s ds \\ &= \begin{cases} \mathbb{E}[Z_t] + \sum_{1 \leq k \leq \frac{i}{2}} M_k I_{1,k} & \text{if } \omega_i \leq t \leq \omega_{i+1} \text{ and } i \text{ is even;} \\ \mathbb{E}[Z_t] + \sum_{1 \leq k \leq \frac{i-1}{2}} M_k I_{1,k} + M_{\frac{i+1}{2}}(t - \omega_i) & \text{if } \omega_i \leq t \leq \omega_{i+1} \text{ and } i \text{ is odd.} \end{cases} \end{aligned}$$

We define $\overline{Z}_t, \underline{Z}_t$ as the almost sure upper and lower bounds of Z_t respectively:

$$\underline{Z}_t \leq Z_t \leq \overline{Z}_t \quad \text{almost surely.}$$

where

$$\underline{Z}_t = \mathbb{E}[Z_t] + \sum_{1 \leq k \leq \lfloor \frac{i}{2} \rfloor} M_k I_{1,k} \quad \text{if } \omega_i \leq t \leq \omega_{i+1},$$

and

$$\overline{Z}_t = \mathbb{E}[Z_t] + \sum_{1 \leq k \leq \lfloor \frac{i+1}{2} \rfloor} M_k I_{1,k} \quad \text{if } \omega_i \leq t \leq \omega_{i+1}.$$

We define $F_\pi(u)$ as the product distribution of $F_{lc}(u)$ and $F_k(u)$. Obviously $F_\pi(u)$ is stationary if the component distributions are stationary. Let $Q_k = M_k I_{1,k}$ then $Q_k \sim F_\pi(u)$. The lower bound process could be rewritten into a more suggestive form:

$$\underline{Z}_t = \mathbb{E}[Z_t] + \sum_{1 \leq k \leq \zeta_t} Q_k, \quad (9)$$

where ζ_t is the the number of regenerations before t of the process $I_0 + I_1 \triangleq \{I_{1,k} + I_{0,k}\}_k$. In other words, the process bounds are essentially renewal-reward processes, whose reward function is Q_k , and the holding times $I_{1,k} + I_{0,k}$ indicate the time between two consecutive renewals into state 1. Let $\rho_k \triangleq \omega_{2k+1}$, then $\{\rho_1, \rho_2, \dots\}$ are the times when $I_0 + I_1$ regenerates. From the Key Renewal Theorem

$$\mathbb{E}[\zeta_{\tau+h}] - \mathbb{E}[\zeta_\tau] \rightarrow \frac{h}{\mu} \quad \text{as } \tau \rightarrow +\infty, \quad (10)$$

where

$$\mu = \mathbb{E}[I_0] + \mathbb{E}[I_1].$$

Equation (10) implicitly assumes that the process has been running *long enough*. We also assume that the longer the run time is, the more renewals into state 1 there will be, with probability one:

$$\zeta_t < \zeta_s \Rightarrow t < s \quad \text{a.s.} \quad (11)$$

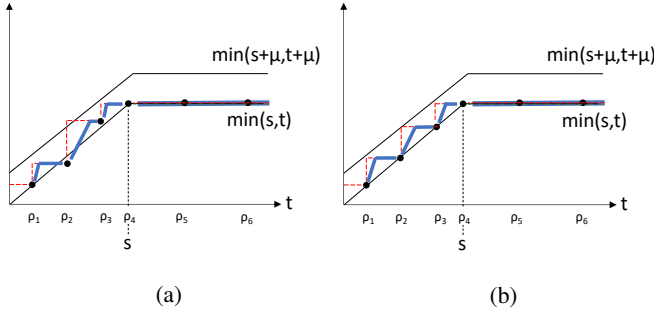


Fig. 9: (a) depicts one instance of $\mu \min(\zeta_t, \zeta_s) \text{Var}[Q_i] | \zeta_t, \zeta_s$, with s fixed, as the blue graph. The black dots are regeneration points. The expectation over all ζ_t, ζ_s , when $\tau \rightarrow +\infty$ gives the covariance function shown on (b). The lower bound of the process coincides with $\min(s, t)$, while the upper bound coincides with $\min(s + \mu, t + \mu)$. Note the relative position of the regeneration points with respect to the two bounds $\min(s, t)$ and $\min(s + \mu, t + \mu)$.

Let $t = \tau + t', s = \tau + s'$, we can calculate the covariance function of Z_t :

$$\begin{aligned}
 \text{Cov}(t, s) &= \mathbb{E}[(Z_t - \mathbb{E}[Z_t])(Z_s - \mathbb{E}[Z_s])] \\
 &= \mathbb{E}[\mathbb{E}[(Z_t - \mathbb{E}[Z_t])(Z_s - \mathbb{E}[Z_s]) | \zeta_t, \zeta_s]] \\
 &= \mathbb{E} \left[\sum_{1 \leq i \leq \zeta_t} \sum_{1 \leq j \leq \zeta_s} \mathbb{E}[(Q_i - \mathbb{E}[Q_i])(Q_j - \mathbb{E}[Q_j])] \right] \\
 &= \mathbb{E}[\min(\zeta_t, \zeta_s) \text{Var}[Q_i]] \\
 &= \mathbb{E}[\min(\zeta_{\tau+t'}, \zeta_{\tau+s'}) \text{Var}[Q_i]] \\
 &\rightarrow \frac{1}{\mu} \min(\rho_{\zeta_t}, \rho_{\zeta_s}) \text{Var}[Q_i]
 \end{aligned} \tag{12}$$

where we have used the law of total probability to go from the second line to the third line, independence of Q_i to go from the third line to the fourth, and (10) to go to the final line if $t < s$ implies $\zeta_t < \zeta_s$ (11).

By definition, $\rho_{\zeta_{t'}} = \frac{t'}{\mu}$, $\text{Cov}(t, s) = (1/\mu) \min(t, s) \text{Var}[Q_i]$. Similarly for \bar{Z}_t :

$$\begin{aligned}
 \overline{\text{Cov}}(t, s) &= \mathbb{E}[(\bar{Z}_t - \mathbb{E}[\bar{Z}_t])(\bar{Z}_s - \mathbb{E}[\bar{Z}_s])] \\
 &= \mathbb{E}[\mathbb{E}[(\bar{Z}_t - \mathbb{E}[\bar{Z}_t])(\bar{Z}_s - \mathbb{E}[\bar{Z}_s]) | \zeta_t, \zeta_s]] \\
 &= \mathbb{E} \left[\sum_{1 \leq i \leq \zeta_t + 1} \sum_{1 \leq j \leq \zeta_s + 1} \mathbb{E}[(Q_i - \mathbb{E}[Q_i])(Q_j - \mathbb{E}[Q_j])] \right] \\
 &= \mathbb{E}[\min(\zeta_t + 1, \zeta_s + 1) \text{Var}[Q_i]] \\
 &= \mathbb{E}[\min(\zeta_{\tau+t'} + 1, \zeta_{\tau+s'} + 1) \text{Var}[Q_i]] \\
 &\rightarrow \frac{1}{\mu} \min(\rho_{\zeta_t + \mu}, \rho_{\zeta_s + \mu}) \text{Var}[Q_i]
 \end{aligned} \tag{13}$$

and we also have $\overline{\text{Cov}}(t, s) = (1/\mu) \min(t + \mu, s + \mu) \text{Var}[Q_i]$. Figure 9 shows the covariance functions' upper and lower bounds. Obviously, if $I_0 + I_1$ regenerates fast enough, $\mu \rightarrow 0$ and we can approximate:

$$\text{Cov}[(t, s)] \approx \frac{1}{\mu} \min(t, s) \text{Var}[Q_i]. \tag{14}$$

□

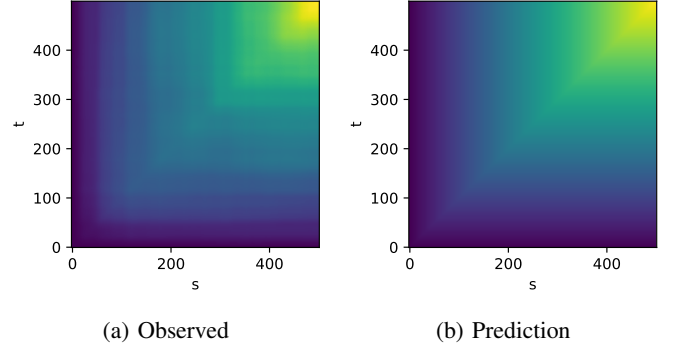


Fig. 10: (a) Computed covariance kernel from the NGSIM dataset (b) Predicted covariance kernel from the RYIM model.

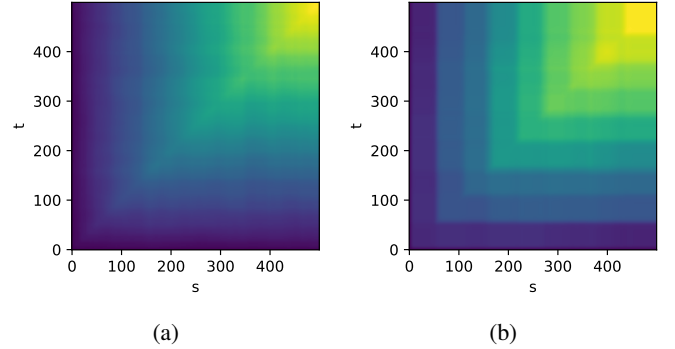


Fig. 11: Prediction of covariance kernel by RYIM model with different idle time distributions: (a) $\Gamma(\alpha = 75, \beta = 0.05)$, (b) $\Gamma(\alpha = 175, \beta = 0.25)$.

Because the covariance kernel's eigenfunctions are also the KL basis [33], we will use the terms interchangeably.

Corollary 1. *The KL decomposition for Z_t is also the same as the Wiener process and is given by:*

$$Z_t = \mathbb{E}[Z_t] + \sum_{k=1}^{+\infty} \beta_k \sqrt{2} \sin \left(\left(k - \frac{1}{2} \right) \pi t \right). \tag{15}$$

where α_k are normally distributed Gaussian random variables.

Proof. See [34]. □

The validity of the model is confirmed by Fig. 10. The similar structure of the covariance kernel attests to the assumptions proposed in Proposition 1, which implies that the process may be considered Markovian with impulses being independent and identically distributed samples from a stationary distribution. The eigenvectors, exhibiting a harmonic shape, also lend weight to this finding. However, the wobbly patterns indicate that the conditions for convergence are not ideal. This is also illustrated in Fig. 11, which displays covariance kernels associated with larger idle times between regenerations. If the regeneration times are large, the kernels will appear more “pixelated”.

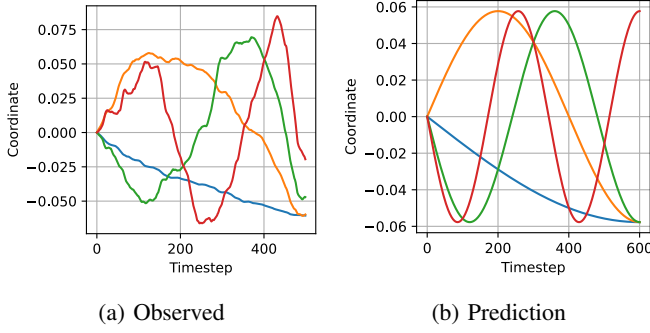


Fig. 12: (a) KL basis obtained by numerically eigendecomposing the empirical covariance kernel (b) KL basis predicted by the RYIM model. Functions that are inverted vertically are the same since the corresponding coefficients can be replaced with an opposite sign.

B. RAIM Model's Covariance Kernel and Eigenfunctions

The derivation of the covariance kernel and eigenfunctions of RAIM is more involved.

Proposition 2. *Under the assumptions made in Proposition 1, the covariance kernel eigenfunctions are given by:*

$$\phi(t) = \sum_{i=1}^{+\infty} \frac{c_i}{\pi^2(i-1/2)^2} e_i(t).$$

and the covariance kernel can be computed from:

$$\begin{aligned} R(s, t) &= \mathbb{E}[(X_t - \mathbb{E}[X_t])(X_s - \mathbb{E}[X_s])] \\ &= \sum_{k=1}^{+\infty} \sum_{l=1}^{+\infty} \frac{\mathbb{E}[\beta_k \beta_l]}{(k-1/2)^2 \pi^2} e_k(t) e_l(s). \end{aligned}$$

Proof. Consider the RAIM given by Eq. (2):

$$X_t = \mathbb{E}[X_t] + \int_{s=0}^t \int_{u=0}^s (A_u du) ds. \quad (16)$$

Restricting to $t \in [0, 1]$, and since A_t and B_t are similar, we can reuse Equation (15):

$$X_t = \mathbb{E}[X_t] + \int_0^1 \left(Z_0 + \sum_{k=1}^{+\infty} \beta_k \sqrt{2} \sin \left(\left(k - \frac{1}{2} \right) \pi t \right) \right) dt. \quad (17)$$

Without any loss of generality, we can assume $Z_0 = 0$. A direct integration gives:

$$X_t = \mathbb{E}[X_t] + \sum_{k=1}^{+\infty} \frac{-\beta_k}{k - \frac{1}{2}\pi} \sqrt{2} \left(\cos \left(\left(k - \frac{1}{2} \right) \pi t \right) - 1 \right). \quad (18)$$

While (18) is a series expansion of X_t , it is far from being the eigenfunctions (or the KLT). This is clear from the fact that $\{\cos((k-1/2)\pi t) - 1\}_k$ do not even form an orthogonal basis. To avoid cluttering of notations, we denote:

$$e_k(t) = \sqrt{2} \left(\cos \left(\left(k - \frac{1}{2} \right) \pi t \right) - 1 \right).$$

Then (18) can be rewritten as:

$$X_t = \mathbb{E}[X_t] + \sum_{k=1}^{+\infty} \frac{-\beta_k}{k - \frac{1}{2}\pi} e_k(t).$$

The covariance function can be calculated directly as:

$$R(s, t) = \mathbb{E}[(X_t - \mathbb{E}[X_t])(X_s - \mathbb{E}[X_s])] \quad (19)$$

$$= \sum_{k=1}^{+\infty} \sum_{l=1}^{+\infty} \frac{\mathbb{E}[\beta_k \beta_l]}{(k-1/2)^2 \pi^2} e_k(t) e_l(s), \quad (20)$$

and independence of β_k leads to:

$$R(s, t) = \sum_{k=1}^{+\infty} \frac{\mathbb{E}[\beta_k^2]}{(k-1/2)^2 \pi^2} e_k(t) e_k(s).$$

Using (15), we can further simplify the equation to:

$$R(s, t) = \sum_{k=1}^{+\infty} \frac{1}{(k-1/2)^4 \pi^4} e_k(t) e_k(s). \quad (21)$$

The KL basis is solution to the Fredholm equation:

$$\int_{t=0}^1 R(s, t) \phi(t) dt = \lambda \phi(s), \quad (22)$$

which is a special case of the Fredholm integral equation of the second kind with separable kernel. In general, there are infinitely many solutions corresponding to the zero eigenvalue of the integral operator of kernel $R(s, t)$. But it is of our interest to find solutions with the largest positive eigenvalues which are written under the form:

$$\phi(t) = \sum_{i=1}^{+\infty} \frac{c_i}{\pi^2(i-1/2)^2} e_i(t). \quad (23)$$

Substitution into (22) gives:

$$\begin{aligned} \lambda \phi(s) &= \int_{t=0}^1 \left(\sum_{i=1}^{+\infty} \frac{1}{(i-1/2)^4 \pi^4} e_i(s) e_i(t) \right) \\ &\quad \left(\sum_{j=1}^{+\infty} \frac{c_j}{\pi^2(j-1/2)^2} e_j(t) \right) dt \\ &= \sum_{i=1}^{+\infty} \sum_{j=1}^{+\infty} \frac{e_i(s)}{(i-1/2)^2 \pi^2} c_j \int \frac{e_i(s)}{(i-1/2)^2 \pi^2} \frac{e_j(s)}{(j-1/2)^2 \pi^2} dt. \end{aligned}$$

Let:

$$\begin{aligned} A_{ij} &= \int \frac{e_i(s)}{(i-1/2)^2 \pi^2} \frac{e_j(s)}{(j-1/2)^2 \pi^2} ds \\ &\triangleq \left\langle \frac{e_i(s)}{(i-1/2)^2 \pi^2}, \frac{e_j(s)}{(j-1/2)^2 \pi^2} \right\rangle. \end{aligned}$$

Then:

$$\lambda \sum_{i=1}^{+\infty} \frac{c_i}{\pi^2(i-1/2)^2} e_i(t) = \sum_{i=1}^{+\infty} \sum_{j=1}^{+\infty} A_{ij} \frac{e_i(s)}{(i-1/2)^2 \pi^2} c_j. \quad (24)$$

Because $e_i(t) \neq 0$, we can match the coefficients of $e_i(t)$:

$$Ac = \lambda c, \quad (25)$$

where $A = [A_{ij}]$, $c = [c_1, c_2, \dots]^\top$. Hence, an eigendecomposition of A should reveal the eigenvectors of coefficients c and the KLT basis is given by (23). The coefficients of the first 3 KL basis functions are given in Table I. \square

TABLE I: 7 first coefficients of the first 3 KL basis to be used in Equation (23).

c_1	c_2	c_3	c_4	c_5	c_6	c_7
-9.99e-01	-3.21e-02	-1.78e-03	-8.45e-04	-2.05e-04	-1.28e-04	-5.03e-05
-9.42e-01	3.30e-01	5.48e-02	4.71e-03	3.85e-03	9.49e-04	8.11e-04
-0.72	0.55	-0.39	-0.10	-0.01	-0.01	0

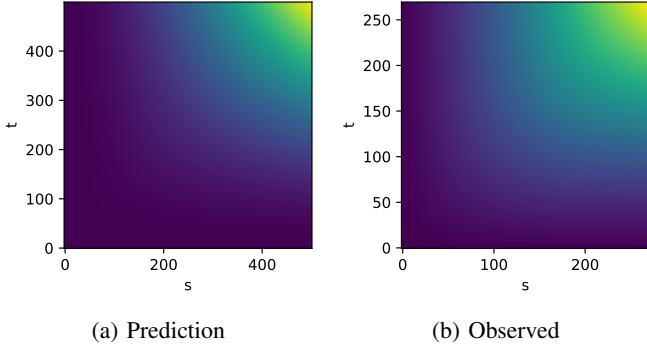


Fig. 13: The covariance kernels from (a) RAIM and (b) NGSIM dataset.

The striking similarity between the covariance kernels predicted by RAIM and those numerically computed from the datasets is readily observable (Fig. 13). Their eigenfunctions also display a strong similarity (Fig. 14). These observations provide robust support for the assumptions presented in Proposition 1, but to a significantly greater degree than the lateral component.

V. FUNCTIONAL PCA (FPCA) APPLICATIONS

Dimensionality reduction, a cornerstone of machine learning, promises innovative solutions in predictive modeling, anomaly detection, and classification tasks. Among these methods, PCA stands out. Extensively utilized in fields like biology and engineering, PCA effectively mitigates the curse of dimensionality, enhancing the performance of inference methods in low-dimensional space.

It turns out that the eigenfunctions introduced in Eq. (15) and (23) are also the basis functions required to perform

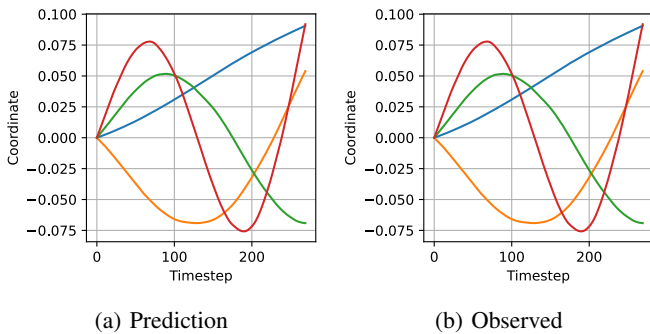


Fig. 14: (a) KL basis obtained from RAIM (b) KL basis obtained numerically from NGSIM dataset. Note that some components in (a) and (b) were inverted, but they are the same since we can invert the sign of the corresponding score component.

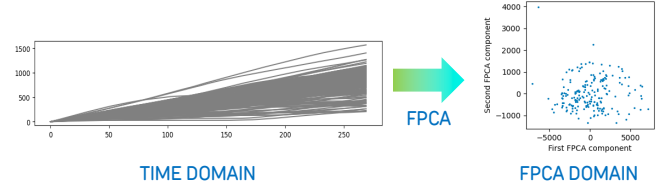


Fig. 15: In the decomposition of trajectories using FPCA, each original time series is treated as a functional—equivalent to a \mathbb{R}^{150} vector in the discrete domain—as depicted by a curve in the left graph. Each of these curves is subsequently transformed into a 2D point, which can be thought of as equivalent to a \mathbb{R}^2 vector. The horizontal axis is β_1 , and the vertical axis is β_2 .

dimensionality reduction (Fig. 15) [35], [36]. In particular, let $f(t)$ be the time series that is either the longitudinal or lateral component of the trajectory, and $e_k(t)$ representing the k -th eigenfunction as denoted by (15) or (23). Then:

$$\beta_k = \int_t e_k(t) f(t) dt \quad (26)$$

Then, given some $N \in \mathbb{N}^*$, $f(t) \in \mathbb{R}^\infty$ can be approximated by a *truncated series*:

$$f(t) \approx \tilde{f}_N(t) \triangleq \sum_{k=1}^N \beta_k e_k(t), \quad (27)$$

in other words, $f(t)$ can now be represented by a vector $\beta = [\beta_1, \dots, \beta_N]$ with a finite dimension of N .

We illustrate multiple applications of this dimensionality reduction method, encompassing data compression, identifying anomalies, and completing trajectories, providing comparisons whenever feasible.

A. Dimensionality Reduction

Definition 1 (N-terms basis restriction error and Normalized mean Squared Error). *Let the N-terms approximated functional of $f(t)$ be given by (27), we define the N-terms basis restriction error as:*

$$\epsilon_N = \int_{t=-\infty}^{+\infty} (f(t) - \tilde{f}_N(t))^2 dt, \quad (28)$$

and the related normalized mean squared error:

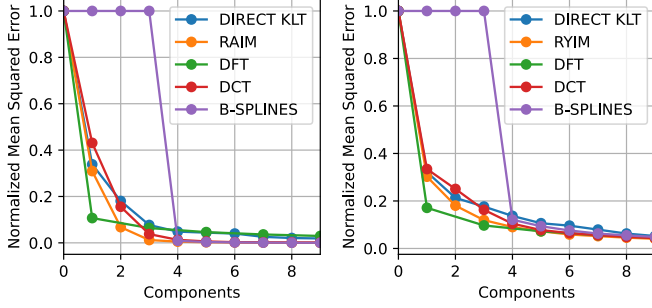
$$\tilde{\epsilon}_N = \frac{\epsilon_N}{\int_{-\infty}^{+\infty} f(t)^2 dt} \quad (29)$$

Then it has been shown in [37] that the FPCA decomposition whose score coefficients given by (26) minimize the N-terms basis restriction error.

Typically, in the traditional FPCA workflow, the basis functions must be computed from a sample set, referred to as the *learning dataset*, and then applied to carry out the decomposition. However, according to Proposition 1, these basis functions remain invariant provided the prescribed assumptions are maintained.

TABLE II: Approximation power of various harmonic transforms for longitudinal component of vehicle trajectories. Values for AVG, STD, MAX, MIN are in normalized square error.

N. components	AVG			STD			MAX			MIN		
	3	6	9	3	6	9	3	6	9	3	6	9
RAIM	0.05	0.00	0.00	0.10	0.00	0.00	0.55	0.01	0.00	0.00	0.00	0.00
DIRECT KLT	0.17	0.05	0.02	0.15	0.03	0.02	0.60	0.14	0.08	0.00	0.00	0.00
DFT	0.07	0.03	0.02	0.05	0.02	0.01	0.18	0.08	0.05	0.00	0.00	0.00
DCT	0.14	0.01	0.00	0.19	0.01	0.00	0.82	0.07	0.01	0.00	0.00	0.00



(a) Longitudinal

(b) Lateral

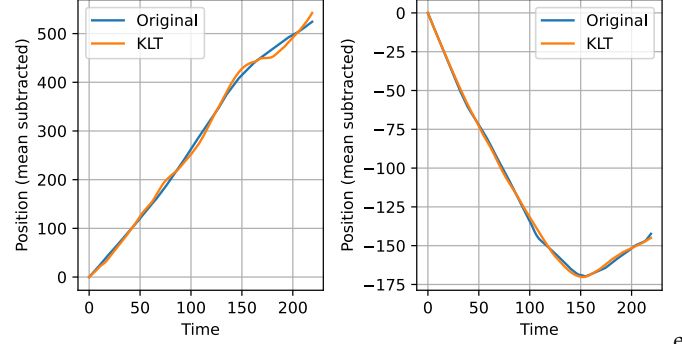
Fig. 16: Average N-terms basis restriction error of longitudinal component (a) and lateral component (b). RAIM and RYIM symbolize the predicted basis functions from the RAIM and RYIM models respectively.

To demonstrate the approximation power, we compare the normalized mean squared error when performing the decomposition using the analytical basis functions (KLT transform) provided by (15) and (23), the basis functions computed numerically from the learning set, and other harmonic linear transforms such as Discrete Fourier Transform (DFT) and Discrete Cosine Transform (DCT). We also include the Cubic Splines Basis for reference due to its popularity, even though B-splines is not typically considered a linear transform.

The remarkable efficacy of the RAIM and RYIM models is clearly demonstrated in Fig. 16, where they exhibit the quickest error convergence for the longitudinal component of the trajectories. Moreover, they rank second for the speediest convergence on the lateral component.

Table II and III presents a more detailed comparison of the performance of the KLT, as predicted by the RAIM and RYIM models, against other harmonic transforms. This includes the numerical KLT, DFT, and DCT. It is clear that RAIM not only delivers superior performance in terms of approximation error, but also exhibits consistency across all trajectories. This consistency is reflected in the smaller standard deviations for all values, including the number of components, as well as the maximum and minimum squared error. For RYIM model, DFT is quicker to converge and also more consistent than RYIM's eigenfunctions. This is thought to be attributed to the non-compliance of the ideal convergence criteria of Proposition 1.

It's noteworthy to mention that this error convergence rate was even faster than when using the empirical basis numerically computed from the covariance kernel. This phenomenon can be linked to the inherent numerical errors and variances



ex

Fig. 17: Approximation with 9 components of the longitudinal component of two sample trajectories. The mean component was subtracted for better visualization.

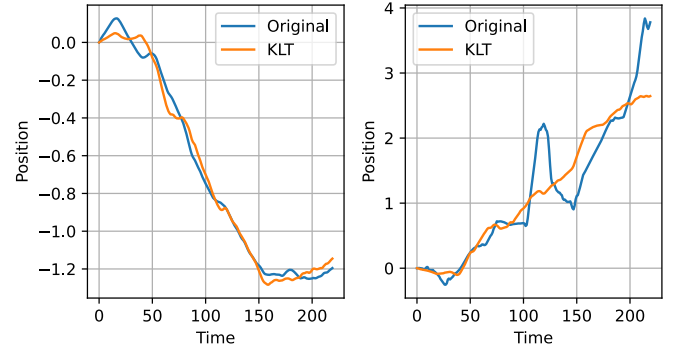


Fig. 18: Approximation with 9 components of the lateral component of two sample trajectories.

resulting from working with a sampled training set. However, it's anticipated that this discrepancy will diminish as more data is incorporated into the training set.

Figures 17 and 18 demonstrate the approximation of trajectories using only nine FPCA scores. It is immediately obvious that FPCA is comparable to the role of DFT and DCT in image processing where the truncated basis similarly reveals a denoising ability. There's a direct relationship between the number of components used in the transformation and the smoothness of the trajectory - fewer components result in smoother trajectories.

B. Data Compression

Drawing parallels with renowned digital data formats such as MP3 and JPEG, the KLT carries strong potential akin to the DCT for data compression. This potential is particularly important in the context of vehicular network. For instance,

TABLE III: Approximation power of various harmonic transforms for lateral component of vehicle trajectories. Values for AVG, STD, MAX, MIN are in normalized square error.

N. components	AVG			STD			MAX			MIN		
	3	6	9	3	6	9	3	6	9	3	6	9
RYIM	0.15	0.06	0.04	0.17	0.08	0.07	0.80	0.54	0.50	0.02	0.01	0.00
DIRECT KLT	0.20	0.10	0.06	0.19	0.10	0.07	0.85	0.63	0.38	0.03	0.03	0.01
DFT	0.09	0.04	0.02	0.10	0.05	0.03	0.61	0.26	0.18	0.02	0.01	0.00
DCT	0.23	0.07	0.05	0.17	0.09	0.07	1.00	0.63	0.53	0.03	0.01	0.00

instead of transmitting vehicle coordinates for each of the 150 time steps, it becomes feasible to transmit only 2-3 FPCA score coefficients, resulting in a far more efficient scheme. Details can further be found in our previous work [38], where we successfully demonstrated the implementation of such transform in devising a data compression strategy for the Cooperative Awareness Message (CAM) transmission within the V2X network setting.

The earlier implementation, however, has some by practical challenges. Specifically, it required continuous collection of trajectories in the learning set and recurrent computation of the basis functions. This paper addresses these challenges and offers a significant improvement: we demonstrate that a single, unique basis set can be used, which simplifies the process considerably. This development makes the approach more amenable for hard coding on embedded platforms.

C. Trajectory Prediction and Completion

FPCA simplifies the dataset by reducing its dimensions, thereby facilitating a more intuitive and straightforward inference process. To this, we propose a Bayesian inference framework as an example. This framework is designed to “fill in the blanks” of the unobserved part of the longitudinal trajectory, using partial observations obtained up to a specific timestep.

Formally, let $y_{1:n}$ be the set of n noisy observations identically and independently obtained from a vehicle’s trajectory whose measurement model is $y = \hat{y} + \epsilon_y$ where $\epsilon_y \sim \mathcal{N}(0, \sigma_y^2)$ and $\mathcal{D} = \{y_{1:q}^i; 1 \leq i \leq m\}$ be the training set of m trajectories whose length is $q > n$. We would like to find the best estimate for $y_{n+1:q}$. Furthermore, let $a \in \mathbb{R}^N$ be the corresponding FPCA score vector of the trajectory $y_{1:q}$ computed from (26).

The general idea is to fit a distribution (presumably Gaussian in this example) on the FPCA score vectors corresponding to the trajectories of \mathcal{D} , giving us the *prior distribution* $p(a) = \mathcal{N}(0, \Sigma_a)$. Then, we compute the posterior (Fig. 19).

$$p(a|y_{1:n}) = \frac{p(y_{1:n}|a)p(a)}{p(y_{1:n})} \propto \prod_{i=1}^n p(y_i|a)p(a) \quad (30)$$

In the context that all distributions are Gaussians, the maximum-a-posteriori coincides with the minimum mean square (MMSE) estimate of a is obtained from Algorithm 1.

Figure 20 provides an example of our efforts to complete a longitudinal trajectory. Remarkably, Fig. 21 reveals that the prediction error remains within acceptable bounds (with error tolerance $\epsilon=5\text{ft}$) for as many as 90 timesteps, equivalent to 9 seconds. This is in contrast to the average of only 20 timesteps

Algorithm 1 Trajectory Representation Estimation

```

1:  $\phi \leftarrow [e_k[0], 1 \leq k \leq q]$ 
2:  $\mu' \leftarrow 0_q \triangleq [0, 0, \dots, 0]$  ( $q$  zeroes)
3:  $\Sigma' \leftarrow \Sigma_a$ 
4: for  $i = 1$  to  $n$  do
5:    $\Sigma' \leftarrow (\Sigma_a^{-1} + 1/\sigma_y^2(\phi_i \otimes \phi_i))^{-1}$ 
6:    $\mu' \leftarrow \mu' + 1/\sigma_y^2(y_i - \mu_i)\phi_i^\top$ 
7:    $a^* \leftarrow \Sigma' \mu'$ 
8: end for

```

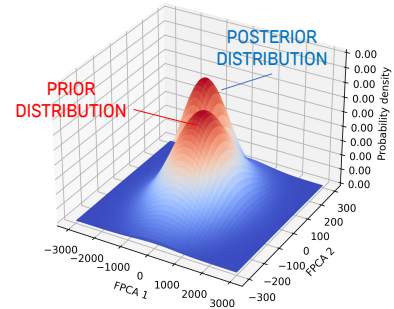


Fig. 19: Illustration of the estimation process: the observations provide information to “narrow down” the distribution of the value of a in the FPCA domain, which explains the more “pointy” look of the posterior distribution.

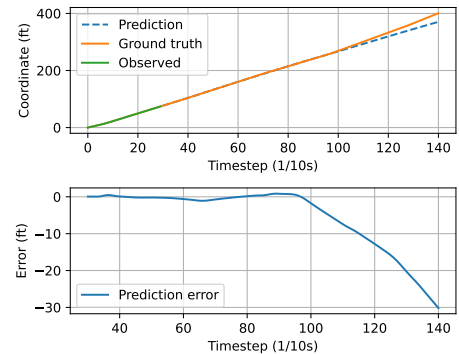


Fig. 20: The trajectory completion is illustrated for observations up to timestep 30. The upper graph presents the completed trajectory, while the lower one displays the error between the actual and predicted trajectories.

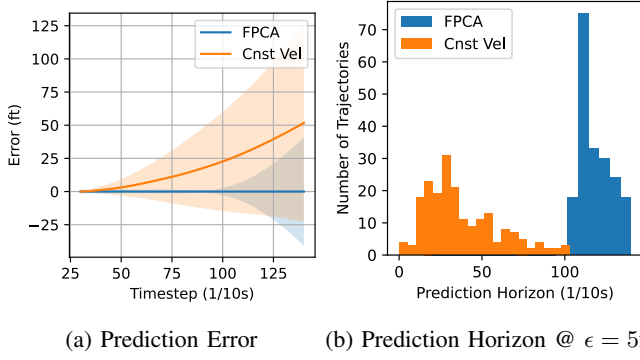


Fig. 21: A comparison was conducted between our Random Impulses FPCA estimate and the Constant Velocity models. This included the mean and standard deviation of the prediction error (a), and a histogram of the prediction horizon (b) for 100 test trajectories sampled from the NGSIM dataset. The prediction error tolerance was set at $\epsilon=5\text{ft}$, and 30 timesteps of observations were made available for prediction.

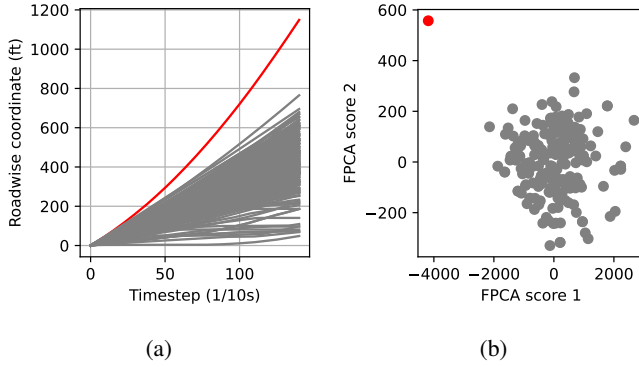


Fig. 22: (a) The speeding vehicle's longitudinal component (b) The corresponding FPCA scores. KNN could easily single out the FPCA representation of this speeding vehicle from the rest.

predicted by the Constant Velocity model. It's also noteworthy that our FPCA-based method begins to exhibit divergence in error from timestep 100 onwards. We attribute this divergence to the finiteness of the time series, a phenomenon similar to performing a Fourier Transform on a windowed signal.

D. Anomaly Detection

Anomaly detection tasks are also straightforward as algorithms such as K-nearest-neighbor (KNN) and clustering can be effectively performed on this domain (Fig. 22). While this concept has been partially explored in previous studies like [39], [40], the utilization of the *same* basis functions for different traffic scenarios to decompose trajectories is a novel finding. This feature, akin to data compression applications, allows us to bypass the learning process, favoring a hard-coded transformation, making these results more impactful.

VI. FILTERING AND SMOOTHING OF VEHICLE TRAJECTORIES EXTRACTED FROM AERIAL VIDEOS

In the domain of extracting high-resolution vehicle trajectories, established tracking frameworks such as the Simple Online and Realtime Tracking (SORT) [17] commonly involve methods such as Kalman or Particle Filter for data assimilation, in conjunction with the Hungarian algorithm to associate measurements with their respective targets. SORT, for instance, presupposes the following *process model*:

$$x_{k+1} = x_k + \epsilon_k, \epsilon_k \sim i.i.d. \mathcal{N}(0, Q) \quad (31)$$

where $x_k = [u, v, s, r, \dot{u}, \dot{v}, \dot{s}, \dot{r}]^T$; here, s and r symbolize the scale, and u, v signify the horizontal and vertical coordinates of the target center, respectively. However, the model makes some questionable assumptions. For instance, the assumption about the nature of the model being perturbed by random walk noise is not adequately justified. Additionally, the characterization of Q as a full-rank covariance matrix is not fully appropriate.

We now propose a novel, comprehensive filtering and smoothing framework based on the previously validated RAIM and RYIM models. Despite extensive research on filtering, smoothing has been relatively under-explored. To the best of our knowledge, this is the first study to suggest a smoothing framework for this application.

A. Approximated Form for Lateral Component Dynamics

Let $\mu_{x,k}$ be the mean of the lateral component of the trajectory at time k . The RYIM model suggests the following approximated model:

$$\begin{aligned} x_{k+1} &= x_k + \mu_{x,k+1} - \mu_{x,k} + \eta_{x,k} \\ y_k &= y_k + \zeta_{x,k} \end{aligned} \quad (32)$$

where:

$$\begin{aligned} \zeta_{x,k} &\sim i.i.d. \mathcal{N}(0, \sigma_x^2) \\ \eta_{x,k} &\sim i.i.d. \alpha_l \mathcal{N}(\mu_{v,l}, \sigma_{v,l}^2) + \alpha_c \mathcal{N}(0, \sigma_{v,c}^2) + \alpha_r \mathcal{N}(\mu_{v,r}, \sigma_{v,r}^2) \end{aligned}$$

In other words, $\eta_{x,k}$ samples from a Gaussian Mixture composing of three modes corresponding to switching to left lane, lane-keeping and switching to right lane (Fig. 7 (b)), rather than from a unimodal Gaussian distribution. By definition:

$$\mathbb{E}[\eta_{x,k}] = 0 \Rightarrow \mu_{v,l} = -\mu_{v,r} \quad (33)$$

and

$$\alpha_l + \alpha_c + \alpha_r = 1 \quad (34)$$

B. Lateral Component Filtering

In the following, the filtering expressions for the particular case of RYIM model will be presented. For a more general case of Gaussian Mixture process and measurement models, we refer to [41]. Recall the fundamental equations of the Bayesian filtering problem are:

$$p(x_k | y_{1:k}) = \frac{p(y_k | x_k) p(x_k | y_{1:k-1})}{p(y_k | y_{1:k-1})} \quad (35)$$

$$p(x_{k+1} | y_{1:k}) = \int p(x_{k+1} | x_k) p(x_k | y_{1:k}) dx_k \quad (36)$$

The equations can be developed into a recursive algorithm, with q being the length of the trajectory and $p(x_1)$ being the prior distribution of the vehicle's lateral position. The algorithm consists of two steps: in the measurement assimilation step, we go from:

$$p(x_k|y_{1:k-1}) = \sum_{l=1}^{N_{k|k-1}} w_{k|k-1}^l \mathcal{N}(\hat{x}_{k|k-1}^l, \sigma_{k|k-1}^{l^2}) \quad (37)$$

to:

$$p(x_k|y_{1:k}) = \sum_{l=1}^{N_{k|k}} w_{k|k}^l \mathcal{N}(\hat{x}_{k|k}^l, \sigma_{k|k}^{l^2}) \quad (38)$$

by using the Kalman update for each individual component l of the predictive mixture $p(x_k|y_{1:k-1})$. In the prediction step, we go from the previous posterior distribution $p(x_k|y_{1:k})$ to the predictive distribution:

$$p(x_{k+1}|y_{1:k}) = \sum_{l=1}^{N_{k+1|k}} w_{k+1|k}^l \mathcal{N}(\hat{x}_{k+1|k}^l, \sigma_{k+1|k}^{l^2}) \quad (39)$$

Given that the process model's Gaussian mixture consists of three components, the quantity of components in the posterior will increase exponentially as iterations progress. To mitigate this, we implement the Kullback-Leibler (KL) GMM Reduction, as outlined in Algorithm 1 of [42]. The main idea is to pairwise merge the components that result in the KL divergence staying bounded to some threshold limit. The filtering procedure for RYIM model is depicted in Algorithm 2.

C. Smoothing of Lateral Component

For a more precise estimation, the distribution of the state vector can be tailored based on future observations, rather than solely relying on the observations up until the estimation time. This leads to a *smoother*, which is usually formulated as:

$$p(x_k|y_{1:q}) = p(x_k|y_{1:k}) \int \frac{p(x_{k+1}|x_k)p(x_{k+1}|y_{1:q})}{p(x_{k+1}|y_{1:k})} dx_{k+1} \quad (40)$$

The conventional form described is used by the renowned *Rauch-Tung-Streifel (RTS) Smoother*. However, as evidenced in [43], an analytical solution doesn't exist when a Gaussian mixture is involved as in the case of RYIM model. Consequently, an alternative method known as the *Two-Filter approach* should be applied:

$$p(x_k|y_{1:q}) = \frac{p(y_{k+1:q}|x_k)p(x_k|y_{1:k})}{p(y_{k+1:q}|y_{1:k})} \quad (41)$$

$$p(y_{k+1:q}|x_k) = \int p(y_{k+1}|x_{k+1})p(y_{k+2:q}|x_{k+1})p(x_{k+1}|x_k)dx_{k+1} \quad (42)$$

The precise formulas of the Two-Filter, applied to a general Gaussian mixture for both process and measurement cases, are detailed in [42]. A standout feature is the utilization of the information form to circumvent computational singularity tied to the uniform distribution prior. In Algorithm 3, we introduce

Algorithm 2 GMKF. Abbreviations: LC - Left Change, LK - Lane Keeping, RC - Right Change, Comp - Mixture Component.

```

Initialize  $N_{1|0} = 1, \hat{x}_{1|0} = x_1, p_{1|0}^1 = \sigma_{x_1}^2$   $\triangleright$  A unimodal
prior of the lateral position  $\mathcal{N}(x_1, \sigma_{x_1}^2)$ 
2: for  $1 \leq k \leq q$  do
     $\triangleright$  Measurement assimilation (Eq. 35)
4:    $N_{k|k} = N_{k|k-1}$   $\triangleright$  Unimodal measurement model
    for  $1 \leq l \leq N_{k|k-1}$  do
6:      $e_k^l = x_k - \hat{x}_{k|k-1}^l$   $\triangleright$  Prediction error
        $\sigma_k^{l^2} = \sigma_{k|k-1}^{l^2} + \sigma_x^2$ 
8:      $K_k^l = \sigma_{k|k-1}^{l^2} / \sigma_k^{l^2}$   $\triangleright$  Kalman gain
        $\hat{x}_{k|k}^l = \hat{x}_{k|k-1}^l + K_k^l e_k^l$   $\triangleright$  Kalman update
10:     $\sigma_{k|k}^{l^2} = (1 - K_k^l) \sigma_{k|k-1}^{l^2}$   $\triangleright$  Posterior comp. var
        $\bar{w}_{k|k}^l = \frac{\bar{w}_{k|k-1}^l \exp(-1/2 e_k^{l^2} / \sigma_k^{l^2})}{\sqrt{2\pi\sigma_k^{l^2}}}$ 
12:    Normalize  $\{\bar{w}_{k|k}^l\}$   $\triangleright$  Posterior comp. weight
    end for
14:     $\triangleright$  Prediction (Eq. 36)
    for  $1 \leq s \leq N_{k|k}$  do
16:      for  $1 \leq j \leq 3$  do  $\triangleright$  3 modes: LC, LK, RC.
         $N_{k+1|k} = 3N_{k|k}$ 
18:         $l = 3(s-1) + j$ 
         $\hat{x}_{k+1|k}^l = \mu_{x,k+1} - \mu_{x,k} + \hat{x}_{k|k}^l$   $\triangleright$  Position for
        LC, LK, RC respectively
20:         $\sigma_{k+1|k}^{l^2} = \sigma_{k|k}^{l^2} + \sigma_p^2$  where  $\sigma_p^2 = \sigma_{v,l}^2$  if  $j = 1$ ,
         $\sigma_{v,c}^2$  if  $j = 2$  and  $\sigma_{v,r}^2$  if  $j = 3$ 
         $w_{k+1|k}^l = w_{k|k}^s \alpha_p$  where  $\alpha_p = \alpha_l$  if  $j = 1$ ,  $\alpha_c$ 
        if  $j = 2$  and  $\alpha_r$  if  $j = 3$ 
22:      end for
    end for
24:    Perform Mixture Component pruning using KL Diver-
    gence (see Algorithm 1 of [42])
end for

```

the expressions for the Backward Information Filter, specific to the RYIM model, which noticeably incorporates the mean trajectory as the control input of the process model. Owing to this unique aspect, we've named this algorithm *Smoothing over Expectation*, or SmoothiE.

The notations used in Algorithm 3 are as follows: $p(y_{k+2:q}|x_{k+1})$ is the Gaussian mixture whose l th component is characterized by the information $L_{k+1|k+2}^l$, $s_{k+1|k+2}^l$ is the information vector and $t_{k+1|k+2}$ is a helpful variable for the weight normalization.

D. Approximated Form for Longitudinal Component Dynamics

Let z_k represent longitudinal position and let $\mu_{y,k}$ stand for the longitudinal mean time series value at timestep k .

$$z_{k+1} = z_k + v_k$$

$$z_{k+1} - \mu_{y,k+1} = z_k - \mu_{y,k+1} - \mu_{y,k} + \mu_{y,k} + v_k$$

Algorithm 3 Smoothing of lateral component (SMOOTHIE).

Initialize $L_{q|q+1}^1 = 0, s_{q|q+1} = 0, t_{q|q+1} = 0, N_q = 1$
for $q - 1 \geq k \geq 1$ **do**
 \triangleright Measurement assimilation (Eq. 41)
 for $1 \leq l \leq N_{k+1}$ **do**
 $L_{k+1|k+1}^l = L_{k+1|k+2}^l + 1/\sigma_x^2$
 $o_{k+1} = -y_{k+1}$
 $s_{k+1|k+1}^l = s_{k+1|k+2}^l + o_{k+1}/\sigma_x^2$
 $t_{k+1|k+1}^l = t_{k+1|k+2}^l + o_{k+1}^2/\sigma_x^2 + \ln(2\pi\sigma_x^2)$
 end for
 \triangleright Backward prediction (Eq. 42)
 for $1 \leq l \leq N_{k+1}$ **do**
 for $1 \leq i \leq 3$ **do**
 $s = 3(l-1) + i$
 $\sigma_p^2 = \sigma_{v,l}^2$ if $i = 1$, $\sigma_{v,c}^2$ if $i = 2$ and $\sigma_{v,r}^2$ if $i = 3$
 $\alpha_p = \alpha_l$ if $i = 1$, α_c if $i = 2$ and α_r if $i = 3$
 $O_{k+1}^s = \frac{1}{L_{k+1|k+1}^s + 1/\sigma_p^2}$
 $L_{k|k+1}^s = \frac{1}{\sigma_p^2}(1 - O_{k+1}^s/\sigma_p^2)$
 $b_k^i = \mu_{x,k+1} - \mu_{x,k} + \mu_{v,l}$ if $i = 1$, $\mu_{x,k+1} - \mu_{x,k}$
 if $i = 2$, $\mu_{x,k+1} - \mu_{x,k} + \mu_{v,r}$ if $i = 3$
 $s_{k|k+1}^s = \frac{1}{\sigma_p^2} \left(b_k^i + O_{k+1}^s \left(s_{k+1|k+1}^l - \frac{b_k^i}{\sigma_p^2} \right) \right)$
 $t_{k|k+1}^s = t_{k+1|k+1}^l - O_{k+1}^s s_{k+1|k+1}^l + \ln \sigma_p^2 - 2 \ln \alpha_p$
 $-\ln |O_{k+1}^s| + b_k^i/\sigma_p^2 \left(b_k^i + 2O_{k+1}^s s_{k+1|k+1}^l - O_{k+1}^s b_k^i/\sigma_p^2 \right)$
 end for
 end for
 \triangleright Convert to state-covariance form
 for each comp l **of the predictive mixture** $k|k+1$ **do**
 $\sigma^{l^2} = 1/L^l$
 $\mu^l = -\sigma^{l^2} s^l$
 $w^l = 0.5(\ln |2\pi\sigma^{l^2}| + L^l \mu^{l^2} - t^l)$
 end for
 Perform Mixture Component pruning using KL Divergence (see Algorithm 1 of [42])
 \triangleright Convert back to information form
 for each comp l **of the predictive mixture** $k|k+1$ **do**
 $L^l = 1/\sigma^{l^2}$
 $s^l = -L^l \mu^l$
 $t^l = \mu^{l^2}/\sigma^{l^2} + \ln |2\pi\sigma^{l^2}| - 2w^l$
 end for
end for

Let $\tilde{z}_k \triangleq z_k - \mu_{y,k}$:

$$\begin{aligned}\tilde{z}_{k+1} - \tilde{z}_k &= \mu_{y,k} - \mu_{y,k+1} + v_k \\ \tilde{v}_k &= -\bar{v}_k + v_k\end{aligned}$$

where \bar{v}_k is the numerical derivative of the mean trajectory, or the mean velocity timeseries. The RAIM model then assumes:

$$\tilde{v}_{k+1} = \tilde{v}_k + \eta_{v,k}, \quad \eta_{v,k} \sim_{i.i.d.} \mathcal{N}(0, \sigma_v^2) \quad (43)$$

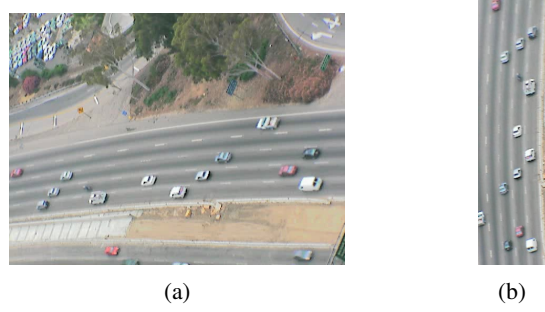


Fig. 23: (a) An original frame from the NGSIM 101 dataset and (b) the frame whose perspective is corrected.

As the result, the full process model for the longitudinal state vector z is:

$$\begin{cases} z_{k+1} = z_k + v_k \\ v_{k+1} = \bar{v}_{k+1} + \tilde{v}_k + \eta_{y,k} = (\bar{v}_{k+1} - \bar{v}_k) + v_k + \eta_{v,k} \end{cases} \quad (44)$$

With a slight abuse of notation of the measurement being y_k for the *longitudinal position* at time k :

$$y_k = z_k + \zeta_{y,k}, \quad \zeta_{y,k} \sim_{i.i.d.} \mathcal{N}(0, \sigma_y^2) \quad (45)$$

Obviously, the longitudinal component is much simpler than the lateral component since all distributions are unimodal Gaussians. This results in a familiar state-space model:

$$\begin{bmatrix} z_{k+1} \\ v_{k+1} \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} z_k \\ v_k \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} (\bar{v}_{k+1} - \bar{v}_k) + \begin{bmatrix} 0 \\ \eta_{v,k} \end{bmatrix} \quad (46)$$

$$y_k = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} z_k \\ v_k \end{bmatrix} + \zeta_{y,k} \quad (47)$$

E. Filtering and Smoothing of Longitudinal Component

The aforementioned state-space model enables conventional implementation of the Kalman Filter [44] for the longitudinal component. When considering smoothing, one might be inclined to apply the *Two-Filter Approach*, as detailed in preceding sections.

Equation (46) implies a process model covariance matrix of

$$Q = \begin{bmatrix} 0 & 0 \\ 0 & \sigma_v^2 \end{bmatrix} \quad (48)$$

which does not have full-rank, complicating calculations like O_{k+1}^s in Algorithm 3. We suggest utilizing the RTS smoother to avoid this, as it doesn't require inversion of the process covariance matrix.

F. Experimental Results

We derive vehicle trajectories from the NGSIM 101 dataset's aerial footage. Due to unavailable camera calibration information, we manually rectified perspective (Fig. 23), using pixel coordinates directly. While this doesn't yield measurements in meters, it is sufficient to demonstrate the proposed filter and smoother's effectiveness.

The full trajectory extraction framework is presented in Fig. 24. The car positions and bounding boxes on the perspectived

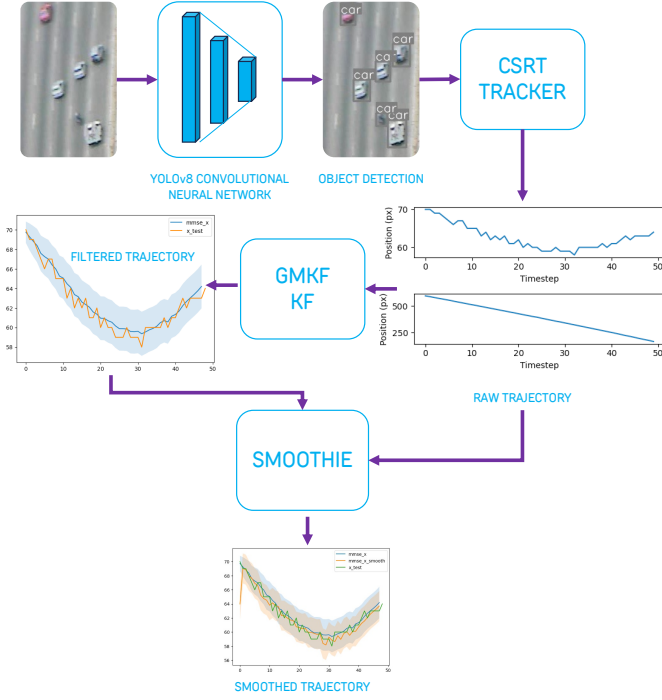


Fig. 24: Overview of the trajectory extraction framework.

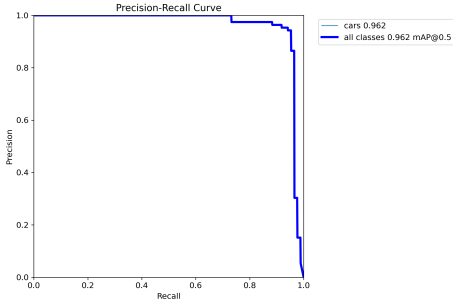
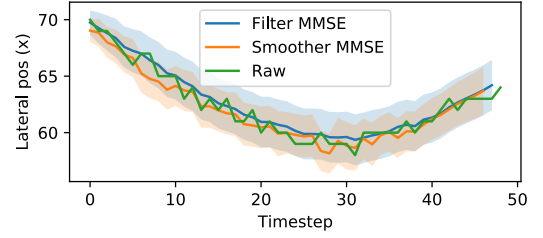
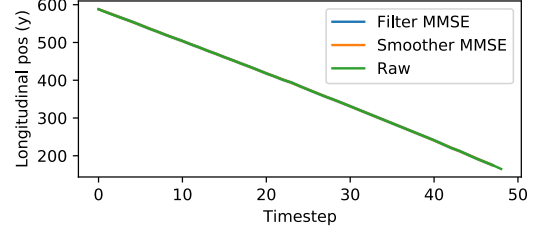


Fig. 25: Precision-recall characteristics of the deep convolutional neural network for detection of cars in video frames.

transformed frame are detected with YOLOv8 [45] trained with 200 manually collected examples. The Precision-Recall Curve is shown in Fig. 25.

Raw trajectory extracted from the CSRT tracker, a direct application of Algorithm 2 and 3 for the lateral component, along with the KF and RTS smoother for the longitudinal component reveal the results shown in Fig. 26-28. Here we have used the following parameters: $\sigma_x = \sigma_y = 0.3$, $\sigma_v = 0.2$, $\alpha_l = 0.05$, $\mu_l = -1.5$, $\sigma_{v,l} = 0.1$, $\alpha_{v,c} = 0.9$, $\sigma_{v,c} = 0.1$, $\alpha_{v,r} = 0.05$, $\mu_{v,r} = 1.5$, $\sigma_{v,r} = 0.1$.

The data clearly demonstrates that both the filter and the smoother produced similar outcomes concerning the lateral and longitudinal position of the vehicle. Nonetheless, the smoother's performance displayed a significant enhancement with approximately a twofold decrease in the confidence interval of the velocity. The method's uniqueness lies in the provision of the smoothed MMSE, coupled with the availability of the confidence interval, distinguishing it from conventional denoising techniques where a confidence interval

Fig. 26: Estimation of lateral position x .Fig. 27: Estimation of longitudinal position y . The difference between estimates are negligible.

is not usually obtainable. In scenarios where the vehicle's velocity does not factor into the application, employing the filtered result could be sufficient for the sake of simplicity.

G. Estimation of the Mean Trajectory

As a prerequisite to both RAIM and RYIM models, the mean trajectory has to be computed for both the lateral and longitudinal components.

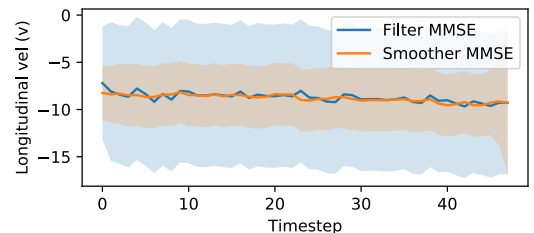
Proposition 3. *The unbiased estimate of the mean trajectory is the empirical mean of the trajectory samples. The growth rate of the sample set size is proportional to n^2 , where n signifies the length of the mean trajectory.*

Proof. From the approximated RYIM model (32):

$$x_{k+1} = x_k + \mu_{x,k+1} - \mu_{x,k} + \eta_{x,k} + \zeta_{x,k}$$

then with n being the length of the trajectory, we have:

$$x_n - x_1 = \mu_{x,n} - \mu_{x,1} + \sum_{k=1}^{n-1} (\eta_{x,k} + \zeta_{x,k})$$

Fig. 28: Estimation of velocity v .

Suppose the set of samples for $x_n - x_1$ is $\{x_n^{(i)} - x_1^{(i)}\}_{i=1}^m$, using Central Limit Theorem (CLT):

$$\frac{\sum_{i=1}^m (x_n^{(i)} - x_1^{(i)}) - m(\mu_{x,n} - \mu_{x,1})}{\sqrt{m}} \rightarrow_D \mathcal{N}(0, (n-1)(\text{Var}[\eta_{x,k}] + \text{Var}[\zeta_{x,k}]))$$

or:

$$\frac{1}{m} \left(\sum_{i=1}^m (x_n^{(i)} - x_1^{(i)}) - (\mu_{x,n} - \mu_{x,1}) \right) \rightarrow_D \mathcal{N} \left(0, \frac{n-1}{m} (\text{Var}[\eta_{x,k}] + \text{Var}[\zeta_{x,k}]) \right)$$

In other words, the size of the dataset required for estimation of the mean lateral component is proportional to the length of the trajectory:

$$m = O(n) \quad (49)$$

For the longitudinal component, we have:

$$y_n - y_1 = \sum_{k=1}^{n-1} v_k$$

We also have:

$$v_k - v_1 = \mu_{v,k} - \mu_{v,1} + \sum_{j=1}^{k-1} \eta_{v,j}$$

Then, it is not hard to see that:

$$y_n - y_1 = \sum_{k=1}^{n-1} (v_1 + \mu_{v,k} - \mu_{v,1}) + \sum_{k=1}^{n-1} \sum_{j=1}^{k-1} \eta_{v,j} + \zeta_{y,n}$$

Notice that $\mathbb{E} v_1 = \mu_{v,1}$, and that both v_1 and $\zeta_{y,n}$ are dominated by the $\sum_{k=1}^{n-1} \sum_{j=1}^{k-1} \eta_{v,j}$ term if n is sufficiently large, using CLT again:

$$\frac{1}{m} \left((y_n^{(i)} - y_1^{(i)}) - \sum_{k=1}^{n-1} \mu_{v,k} \right) \rightarrow_D \mathcal{N} \left(0, \frac{(n-2)(n-1)}{2m} \text{Var}[\eta_v] \right)$$

So the size of the sample set m grows with the square of the length of the vehicle trajectory:

$$m = O(n^2) \quad (50)$$

□

VII. DISCUSSION

The validation of the Random Impulses Model demonstrates that the Markovian assumption regarding random acceleration and yawing impulses is reasonable, although the assumption is more valid for the longitudinal component than the lateral. The models further reveal that the discrepancy originates from the less frequent regeneration of yawing impulses compared to acceleration impulses. Consequently, the approximated models (32) and (44) generally hold, but to varying degrees.

The models also offer a more intuitive approximation of the underlying physical processes. Rather than modeling the driver's reactions as functions of various external factors, they

represent cumulative distance-keeping behavior using a single component, namely the mean trajectory. The residual component reflects the variations in inputs, leading to a significantly simpler model. However, despite this simplicity, the model still maintains the powerful characteristics seen in car-following models, such as a long prediction horizon.

With the validation of these models and the availability of covariance kernels, the applications of FPCA can be extended. Conventionally, FPCA necessitates the derivation of basis functions from a *training set*, raising the question of the *robustness* of these functions in relation to time and traffic conditions. This paper introduces a novel finding from Proposition 1, asserting that the basis functions remain invariant under mild conditions of impulse regeneration time and amplitude samples from a stationary distribution. These assumptions appear valid for highway scenarios, and we speculate that they likely hold for most traffic scenarios, provided conditions do not rapidly change, such as with the sudden presence of reckless vehicles.

The implications of these findings are significant; the learning phase can be entirely bypassed, and the same basis functions can be hard-coded into signal processors. This can lead to considerable memory and cost savings for vehicular and ITS applications.

Moreover, this paper presents a comprehensive probabilistic framework for filtering and, more importantly, smoothing of trajectories from aerial videos. While the position estimates of the vehicles don't differ substantially between filtering and smoothing, velocity estimation proves much more effective, as indicated by a confidence interval reduction of approximately 50%. This paper also provides the theoretical background, justification, and reference implementation for the use of random walks as process noise models for vehicle tracking. While the application is currently limited to trajectory extraction from aerial videos, we believe these principles could be applicable in various automotive contexts, such as radar tracking and antenna beamforming.

However, we acknowledge several unaddressed shortcomings that warrant further research. First, a more detailed experimental or numerical study that compares filtering and smoothing results against other common trackers is needed. Second, further refinement of the models is necessary if multiple modes of traffic are available. Thirdly, akin to the Fourier Transform, it is crucial to account for the window effect on the time domain signal transform. This helps to prevent the divergence of error towards the end of the signal.

VIII. CONCLUSION

In conclusion, this paper introduces a kinematic statistical model for road vehicle trajectory and investigates its potential in various applications. These include FPCA, as well as the approximated forms for probabilistic filtering and smoothing of trajectories derived from aerial video extraction.

The model's assumptions were validated through the computation of covariance kernels, while a convergence study illuminated the approximation power of the RYIM and RAIM models. The simplicity of these models, combined with the

theoretical developments presented in this paper, advocate for increased confidence in the use of the derived FPCA basis functions within ITS and vehicular embedded platforms.

Moreover, the filtering and smoothing framework enriches existing literature on the use of random walks as a process noise model. Ultimately, this research not only contributes to the current understanding of vehicle trajectory modeling, but also paves the way for further exploration and application in this field.

ACKNOWLEDGMENTS

This research was supported by the AI Interdisciplinary Institute ANITI, funded by the French "Investing for the Future – PIA3" program (Grant n°ANR-19-PI3A-0004). We also acknowledge Google Colab's generosity in providing free GPU for YOLOv8 training.

REFERENCES

- [1] H. Yu and M. Krstic, "Traffic congestion control for aw-rasle-zhang model," *Automatica*, vol. 100, pp. 38–51, 2019.
- [2] E. Yurtsever, J. Lambert, A. Carballo, and K. Takeda, "A survey of autonomous driving: Common practices and emerging technologies," *IEEE access*, vol. 8, pp. 58 443–58 469, 2020.
- [3] Y. Li and D. Sun, "Microscopic car-following model for the traffic flow: the state of the art," *Journal of Control Theory and Applications*, vol. 10, no. 2, pp. 133–143, 2012.
- [4] S. Moridpour, M. Sarvi, and G. Rose, "Lane changing models: a critical review," *Transportation letters*, vol. 2, no. 3, pp. 157–173, 2010.
- [5] Y. Ji, L. Wang, W. Wu, H. Shao, and Y. Feng, "A method for lstm-based trajectory modeling and abnormal trajectory detection," *IEEE Access*, vol. 8, pp. 104 063–104 073, 2020.
- [6] Y. Wang, G. Li, K. Li, and H. Yuan, "A deep generative model for trajectory modeling and utilization," *Proceedings of the VLDB Endowment*, vol. 16, no. 4, pp. 973–985, 2022.
- [7] G. Zhu, Y. Sang, W. Chen, and L. Zhao, "When self-attention and topological structure make a difference: Trajectory modeling in road networks," in *Web and Big Data: 6th International Joint Conference, APWeb-WAIM 2022, Nanjing, China, November 25–27, 2022, Proceedings, Part III*. Springer, 2023, pp. 381–396.
- [8] S. Lefèvre, D. Vasquez, and C. Laugier, "A survey on motion prediction and risk assessment for intelligent vehicles," *ROBOMECH journal*, vol. 1, no. 1, pp. 1–14, 2014.
- [9] S. Ammoun and F. Nashashibi, "Real time trajectory prediction for collision risk estimation between vehicles," in *2009 IEEE 5th International Conference on Intelligent Computer Communication and Processing*. IEEE, 2009, pp. 417–422.
- [10] N. Kaempchen, K. Weiss, M. Schaefer, and K. C. Dietmayer, "Imm object tracking for high dynamic driving maneuvers," in *IEEE Intelligent Vehicles Symposium, 2004*. IEEE, 2004, pp. 825–830.
- [11] A. Polychronopoulos, M. Tsogas, A. J. Amditis, and L. Andreone, "Sensor fusion for predicting vehicles' path for collision avoidance systems," *IEEE Transactions on Intelligent Transportation Systems*, vol. 8, no. 3, pp. 549–562, 2007.
- [12] V. G. Kovvali, V. Alexiadis, and L. Zhang PE, "Video-based vehicle trajectory data collection," Tech. Rep., 2007.
- [13] J. Apeltauer, A. Babinec, D. Herman, and T. Apeltauer, "Automatic vehicle trajectory extraction for traffic analysis from aerial video data," *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 40, no. 3, p. 9, 2015.
- [14] N. Balamuralidhar, S. Tilon, and F. Nex, "Multeye: Monitoring system for real-time vehicle detection, tracking and speed estimation from uav imagery on edge-computing platforms," *Remote Sensing*, vol. 13, no. 4, p. 573, 2021.
- [15] X. Chen, Z. Li, Y. Yang, L. Qi, and R. Ke, "High-resolution vehicle trajectory extraction and denoising from aerial videos," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 5, pp. 3190–3202, 2020.
- [16] E. Barmounakis and N. Gerolimimis, "On the new era of urban traffic monitoring with massive drone data: The pneuma large-scale field experiment," *Transportation research part C: emerging technologies*, vol. 111, pp. 50–71, 2020.
- [17] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Uprocroft, "Simple online and realtime tracking," in *2016 IEEE international conference on image processing (ICIP)*. IEEE, 2016, pp. 3464–3468.
- [18] N. Wojke, A. Bewley, and D. Paulus, "Simple online and realtime tracking with a deep association metric," in *2017 IEEE international conference on image processing (ICIP)*. IEEE, 2017, pp. 3645–3649.
- [19] H. Fu, L. Wu, M. Jian, Y. Yang, and X. Wang, "Mf-sort: Simple online and realtime tracking with motion features," in *Image and Graphics: 10th International Conference, ICIG 2019, Beijing, China, August 23–25, 2019, Proceedings, Part I 10*. Springer, 2019, pp. 157–168.
- [20] K.-C. Liu, Y.-T. Shen, and L.-G. Chen, "Simple online and realtime tracking with spherical panoramic camera," in *2018 IEEE International Conference on Consumer Electronics (ICCE)*. IEEE, 2018, pp. 1–6.
- [21] P. G. Gipps, "A behavioural car-following model for computer simulation," *Transportation Research Part B: Methodological*, vol. 15, no. 2, pp. 105–111, 1981.
- [22] K. I. Ahmed, "Modeling drivers' acceleration and lane changing behavior," Ph.D. dissertation, Massachusetts Institute of Technology, 1999.
- [23] K. Aghabayk, M. Sarvi, and W. Young, "A state-of-the-art review of car-following models with particular considerations of heavy vehicles," *Transport reviews*, vol. 35, no. 1, pp. 82–105, 2015.
- [24] J. Han, H. Shi, L. Chen, H. Li, and X. Wang, "The car-following model and its applications in the v2x environment: A historical review," *Future Internet*, vol. 14, no. 1, p. 14, 2021.
- [25] G. Agamennoni, J. I. Nieto, and E. M. Nebot, "A bayesian approach for driving behavior inference," in *2011 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2011, pp. 595–600.
- [26] J. Joseph, F. Doshi-Velez, A. S. Huang, and N. Roy, "A bayesian nonparametric approach to modeling motion patterns," *Autonomous Robots*, vol. 31, pp. 383–400, 2011.
- [27] G. Aoude, J. Joseph, N. Roy, and J. How, "Mobile agent trajectory prediction using bayesian nonparametric reachability trees," in *Infotech@ Aerospace 2011*, 2011, p. 1512.
- [28] R. Chandra, U. Bhattacharya, A. Bera, and D. Manocha, "Trophic: Trajectory prediction in dense and heterogeneous traffic using weighted interactions," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 8483–8492.
- [29] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.
- [30] K. Messaoud, I. Yahiaoui, A. Verroust-Blondet, and F. Nashashibi, "Attention based vehicle trajectory prediction," *IEEE Transactions on Intelligent Vehicles*, vol. 6, no. 1, pp. 175–185, 2020.
- [31] A. Barth and U. Franke, "Where will the oncoming vehicle be the next second?" in *2008 IEEE Intelligent Vehicles Symposium*. IEEE, 2008, pp. 1068–1073.
- [32] A. Kuczura, "Piecewise markov processes," *SIAM Journal on Applied Mathematics*, vol. 24, no. 2, pp. 169–181, 1973.
- [33] P. Deheuvels and G. Martynov, "Karhunen-Loeve expansions for weighted Wiener processes and Brownian bridges via Bessel functions," in *High Dimensional Probability III*. Springer, 2003, pp. 57–93.
- [34] R. Durrett, *Probability: theory and examples*. Cambridge university press, 2019, vol. 49.
- [35] T. H. Dinh, V. Martinez, and D. Delahaye, "Recognition of outlying driving behaviors: A data-driven perspective with applications to v2x collective perception," in *2021 IEEE Vehicular Networking Conference (VNC)*. IEEE, 2021, pp. 52–59.
- [36] H. L. Shang, "A survey of functional principal component analysis," *ASIA Advances in Statistical Analysis*, vol. 98, pp. 121–142, 2014.
- [37] J. Brown, Jr, "Mean square truncation error in series expansions of random functions," *Journal of the Society for Industrial and Applied Mathematics*, vol. 8, no. 1, pp. 28–32, 1960.
- [38] T. H. Dinh, V. Martinez, and D. Delahaye, "Spherical codec for v2x cooperative awareness trajectory compression: A preliminary study," in *2023 IEEE Vehicular Technology Conference (VTC)*. IEEE, 2023.
- [39] G. Jarry, D. Delahaye, F. Nicol, and E. Feron, "Aircraft atypical approach detection using functional principal component analysis," *Journal of Air Transport Management*, vol. 84, p. 101787, 2020.
- [40] G. Jarry, D. Delahaye, and E. Feron, "Trajectory approach analysis: A post-operational aircraft approach analysis tool," in *SID 2019, 9th SESAR Innovation Days*, 2019.
- [41] A. G. Wills, J. Hendriks, C. Renton, and B. Ninness, "A bayesian filtering algorithm for gaussian mixture models," *arXiv preprint arXiv:1705.05495*, 2017.
- [42] M. P. Balenzuela, J. Dahlin, N. Bartlett, A. G. Wills, C. Renton, and B. Ninness, "Accurate gaussian mixture model smoothing using a two-

filter approach,” in *2018 IEEE Conference on Decision and Control (CDC)*. IEEE, 2018, pp. 694–699.

- [43] G. Kitagawa, “The two-filter formula for smoothing and an implementation of the gaussian-sum smoother,” *Annals of the Institute of Statistical Mathematics*, vol. 46, pp. 605–623, 1994.
- [44] G. Welch, G. Bishop *et al.*, “An introduction to the kalman filter,” 1995.
- [45] Ultralytics. Ultralytics yolov8. [Online]. Available: <https://docs.ultralytics.com/>

Thinh Hoang Dinh is currently pursuing his PhD at the ENAC Optimization Laboratory in Toulouse, France. His research focuses on the application of artificial intelligence in the field of transportation, aiming to improve efficiency and sustainability within this sector.

Vincent Martinez received the M.Sc. degree in mathematical and statistical modeling from INSA Toulouse, France. He is working as a Senior Principal Engineer at NXP Semiconductors since 2006. He is currently a Wireless Signal Processing and Physical Layer Expert, and is contributing to various V2X standardization bodies.

Pierre Maréchal is Professor of Mathematics at the University of Toulouse III Paul Sabatier. Although his initial training was in engineering, he obtained a PhD in Image processing in 1997, and then turned to mathematics through a postdoc at Simon Fraser University. His areas of interest are mainly, but not exclusively, optimization and inverse problems.

Daniel Delahaye leads the Optimization and Machine Learning Team at ENAC Research Laboratory and chairs the “AI for ATM and Large Scale Urban Mobility” research at ANITI in Toulouse. After earning his engineering degree and master’s from ENAC and National Polytechnic Institute of Toulouse, he obtained his Ph.D. in automatic control and later tenure in applied mathematics at ENAC. He has worked at the French Civil Aviation Study Center and MIT. His research, conducted in collaboration with MIT, Georgia Tech, and NASA, centers on AI and mathematical optimization for airspace and aircraft trajectory optimization.