MOTORCYCLE STATE ESTIMATION USING VISUAL-INERTIAL ODOMETRY

Martin Pryde
Lamri Nehaoua
Hicham Hadj-Abdelkader
Hichem Arioui
Laboratoire IBISC, Université d'Evry Val-d'Essonne, Université Paris-Saclay
France

Paper number 23-0225

ABSTRACT

The authors propose a visual-inertial algorithm to estimate the kinematic states of a motorcycle traveling at high speeds along an extra-urban road. The approach comprises the following steps: First, a monocular camera takes video of the road ahead. Key features from sequential video frames of the road surface are extracted using the Harris corner detector. Matching features are identified using the Fast retina keypoint descriptor (FREAK). Next, correct the perspective warping of the feature locations by applying inverse perspective mapping. The motion of the transformed features is registered using the Singular Value Decomposition (SVD) variant of the Iterative Closest Point (ICP) algorithm. Finally, this measurement is combined with readings from inertial navigation system using a Kalman filter to produce a filtered estimate and correct integrator drift. The approach was validated using data from simulations of three scenarios created in BikeSim. In the first, the motorcycle performs a series of slaloms along a straight road at 50 km/h. In the second, the motorcycle navigates an S-shaped bend at 80 km/h. Lastly, the motorcycle performs a double-lane change across both lanes of a straight road at 110 km/h.

INTRODUCTION

The term *Powered Two-Wheeled Vehicles* (P2WV) encompasses the class of self-propelled road vehicles whose two wheels are arranged in tandem. *Riders* of P2WVs continue to be overrepresented in severe road accidents [1]. By contrast, automobile road safety has steadily improved over the past decade. Among many factors in this trend has been the introduction of Advanced Driver Assistance Systems (ADAS) into modern cars [2]. In this context, researchers at the IBISC laboratory are investigating the modeling and control of P2WVs with the aspiration of developing technologies for *Advanced Rider Assistance System (ARAS)*. Various rider aids are already commercially available such as Forward Collision Warning (FCW) and Adaptive Cruise Control (ACC) that are analogous to their automobile counterparts. We are particularly interested in developing an ARAS analog of the *Electronic Stability Control* (ESC) now ubiquitous among recent car models. Consider a warning system capable of detecting and alerting the rider to dangerous steering situations. In critical situations, such a system could even intervene semi-autonomously to mitigate an accident.

Developing such a system is not as simple as implementing existing ESC on motorcycles: Their slim profiles permit a larger envelope of lateral motion compared to a car. Furthermore, a novice rider often has trouble judging the appropriate lateral position and heading relative to the road to safely navigate a bend. Hence, riders can attain larger magnitudes of relative lateral velocity compared to cars. Consider also that a rider must not only judge their trajectory but must also *lean* into a turn to balance out the overturning moment caused by the road-tire interaction. Therefore, the key difference from cars is that the body lateral velocity, yaw rate and steering angle can no longer be assumed to act in the road plane. Alternative formulations for the front and rear wheel *slip angles*, the angles formed between the direction in which a wheel is pointing and in which it is traveling, must be used for an ESC equivalent for P2WVs to effectively characterize dangerous over and under-steering behaviors.

Visual-Inertial Odometry (VIO) has shown encouraging performance in measuring translational velocities in *Uncrewed Aerial Vehicles* (UAVs) [3]. VIO estimates *ego-motion* in real-time using a camera alongside an *Inertial Measurement Unit* (IMU). The readings from these sensors are fed to a motion estimation algorithm such as *Optical Flow* (OF), the *Direct Linear Transform* (DLT) and *Iterative Closest Point* (ICP). Research has also been conducted into implementing VIO for *Uncrewed Ground Vehicles* (UGVs). The planar constraints of UGVs simplify the motion estimation problem to two dimensions compared to UAVs. VIO for UGVs has shown promise during indoor navigation tasks performed across multiple works [4],[5]. Outside of laboratory conditions, the works of Song et al. show a real-world demonstration of VIO for a small UGV capturing video of the ground beneath itself. We note that to date, these implementations were conducted only at low speeds ($\leq$ 50 km/h).

We take inspiration from VIO for UAVs alongside the work conducted thus far for UGVs and investigate its potential in the real-time estimation of velocity for P2WVs. We seek to know if the VIO algorithms which perform well at high velocities on UAVs perform comparably on a motorcycle. Previous works from IBISC have investigated the use of Inverse Perspective Mapping (IPM) to further simplify the motion estimation problem for road vehicles [6]. Thus, we summarize our estimation algorithm: We extract matching features from successive images taken by a camera mounted on the front of a motorcycle. Next, we use IPM to remove the perspective warping from the feature sets and obtain two point sets. We apply ICP to estimate the rigid transform between these two sets. From here, we express this motion estimate in the body frame of the motorcycle and use it as the drift-correcting component of a Kalman filter-based *Inertial Navigation System* (INS). Finally, we re-express this filtered estimate in the road plane and apply the appropriate kinematic expressions for P2WVs to calculate the wheel slip angles.

Motivation

Referring to the seminal works on motorcycle dynamics by Cossalter et al. [7], we define two cornering radii: The *ideal* radius R_0 of the path taken by the motorcycle assuming there is no lateral wheel slip and the actual path radius R. The ratio of these is termed the *steering ratio* ξ and defines whether a motorcycle is over, neutral or under-steering as shown in (1). Cossalter et al. derives and validates the approximations for these radii in kinematic terms shown in (2) and (3) where α_f and α_r are the front and rear slip angles respectively [8].

$$\xi = \frac{R_0}{R} \begin{cases} \xi < 1 & \text{under-steering} \\ \xi = 1 & \text{neutral steering} \\ \xi > 1 & \text{over-steering} \end{cases} \tag{1}$$

$$R_0 = \frac{l_{\text{wb}}}{\tan \Delta} \tag{2}$$

$$R = \frac{l_{\text{wb}}}{\tan(\Delta - \alpha_r)\cos(\alpha_r) + \sin(\alpha_f)} = \frac{V}{\dot{\psi}} \tag{3}$$

The definitions for all symbols used in this work are listed in Table 1. V is the *velocity magnitude* defined as the norm of the longitudinal and lateral body velocity components (4). Care should be taken to distinguish the handlebar steering angle δ from its projection onto the road plane Δ termed the *kinematic steering angle* which can be well-approximated using (5).

$$V = \sqrt{v_x^2 + v_y^2} \tag{4}$$

$$\Delta = \arctan\left(\frac{\cos(\epsilon)}{\cos(\varphi)}\tan(\delta)\right) \tag{5}$$

Relationships first derived by Robin Sharp express the front and rear slip angles in terms of the kinematic variables defined thus far. Note that in (6) and (7) as well as in (4) the longitudinal and lateral components v_x and v_y are expressed in the road plane and correspond to the velocity of the *projection* of the motorcycle Center of Mass (CoM) onto this plane.

$$\alpha_f = \arctan\left(\frac{v_y + l_f\dot{\psi} - l_{\text{tr}}\dot{\delta}}{v_x}\right) - \delta\cos(\epsilon) \tag{6}$$

$$\alpha_r = \arctan\left(\frac{v_y - l_r\dot{\psi}}{v_x}\right) \tag{7}$$

Thus one concludes that accurate measurements of velocity, yaw rate and steering angle are essential when attempting to estimate the slip angles. Steering angle and rate can be obtained using a position encoder in the steering bearing while the yaw rate can be accurately measured by even low-cost MEMS gyroscopes. However, velocity is a more challenging state to reconstruct: One can suppose that longitudinal velocity can be measured by the vehicle's engine odometer under the assumption that there is no longitudinal slip between the tires and the road. Lateral velocity is more challenging still: Many approaches assume the availability of this state through *Global Navigation Satelite System* (GNSS). In reality, the latency and precision of most consumer GNSS modules are, on their own, inadequate for real-time estimation and control applications according to tests carried out by the Connected Motorcycle Consortium (CMC) at the Technische Hochschule Ingolstadt in Germany [9].

Table 1.
List of states and their definitions (absent of point and frame definitions)

symbol	state	units	symbol	parameter	units
$\boldsymbol{a}$	translational acceleration	m/s^2	l_f	arm between G_r and C_f	m
$\boldsymbol{\omega}$	angular velocity	rad/s	l_r	arm between G_r and C_r	m
$\boldsymbol{v}$	translational velocity	m/s	l_{rk}	rake	m
ϕ	roll	rad	l_{tr}	normal trail $R_f \sin(\epsilon) - l_{\text{rk}}$	m
θ	pitch	rad	R_f	front wheel radius	m
ψ	yaw	rad	R_r	rear wheel radius	m
φ	lean	rad	ϵ	caster angle	rad
δ	steer	rad	μ	camera tilt	rad
Δ	kinematic steer	rad	l_{wb}	wheelbase $l_f + l_r$	m
α_f	front slip	rad			
α_r	rear slip	rad			

State observer approaches hypothesize that it is possible to reconstruct lateral velocity from measurements of other states and a sufficiently faithful dynamical model. The most widely used of these is the Sharp 1971 linearized model [10]. However, observers based on the Sharp model and variations of it have delivered mixed results to date across multiple works [11],[12],[13]. Ahead of this work, we re-confirmed the results of [12] in Figure (1) by implementing their *Linear Parameter-Varying* (LPV) variant of the Sharp model and comparing against results from BikeSim. We note that while the yaw and lean dynamics estimates are satisfactory, the estimate for the lateral velocity deviates significantly from the ground truth. Furthermore, model-based observers rely on accurate *a priori* knowledge of P2WV dynamic parameters, such as mass and moment of inertia, which vary widely depending on the rider [14]. Hence, a key technical motivation for this work is to obtain a more accurate and robust result for lateral velocity compared to available model-based observer approaches.

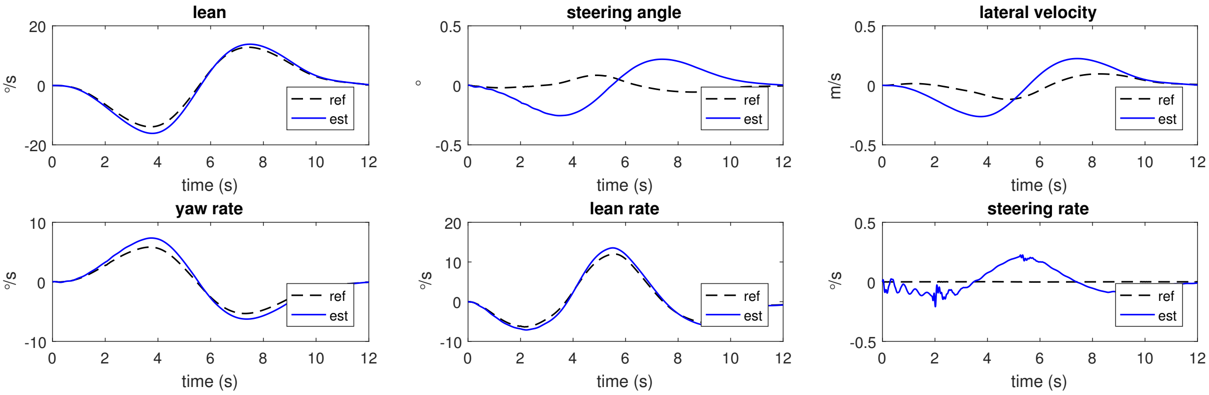

Figure 1. Lateral motion results for an LPV model of a motorcycle. Note the significant deviation from ground truth lateral velocity in the top right plot.

Notation

We summarize the notation used in this work as follows:

- The frame originating from the point P is denoted $\mathfrak{R}_P$.
- $\mathbf{x}$ and $\boldsymbol{x}$ are column vectors while $\mathbf{A}$ and $\boldsymbol{A}$ are matrices.
- The position vector $\boldsymbol{p}_a^b \in \mathbb{R}^3$ is the Euclidean position of the point a expressed in $\mathfrak{R}_b$.
- The distance vector $\boldsymbol{r}_{a,b}^b$ begins at point a, ends at point b and is expressed in $\mathfrak{R}_b$.
- The time derivative of $\boldsymbol{p}_a^b$ is $\dot{\boldsymbol{p}}_a^b$.
- the velocity vector $\boldsymbol{v}_a^b$ is the velocity of the point a expressed in $\mathfrak{R}_b$.
- $\boldsymbol{A}^{\mathrm{T}}$ is the transpose of $\boldsymbol{A}$ while $\boldsymbol{A}^{-1}$ is it's inverse.
- We favor the skew-symmetric operator $[\mathbf{x}]_\times$ over the cross product ×.
- The identity and null matrices are denoted by $\mathbf{I}$ and $\mathbf{0}$ respectively.
- The det operator returns the determinant of a matrix.

SYSTEM DESCRIPTION

Consider a motorcycle traveling at high speed along an extra-urban road. We propose to attach a monocular camera in front of the steering head fixed to the motorcycle rear body as shown in Figure 2. When the *lean* φ of the motorcycle is zero, the camera is located at a height h_0 above the road and at a distance l_f from G_r shown in Figure 3. Concerning environmental conditions, assume the road surface is smooth and well-illuminated in good weather so that no raindrops and little jitter are present in the camera images.

Additionally, we consider that this motorcycle is fitted with an IMU located at the Center of Mass (CoM) of the rear body G_r. The IMU measures the angular velocity $\boldsymbol{\omega}^{G_r}$ acceleration $\boldsymbol{a}^{G_r}$ in the rear body frame $\mathfrak{R}_{G_r}$. Assume that the IMU module is programmed with an *Attitude Heading and Reference System* (AHRS) sensor fusion algorithm which outputs the roll ϕ, pitch θ and yaw ψ angles of G_r. Note that we make a distinction between the roll and lean: The roll is a rotation about the x-axis of a frame that has previously been rotated about the local z and y-axes by the yaw and pitch respectively. In contrast, the lean is a rotation about the x-axis of $\mathfrak{R}_V$.

Lastly, assume that the *steer* angle of the steering head is measured with an encoder that also provides its time derivative the *steer rate*. Steer measured in the front body frame $\mathfrak{R}_{G_f}$ is denoted by δ. The projection of this steer onto the road surface is denoted by Δ and is expressed in $\mathfrak{R}_V$. The point V is the projection of G_r onto the road surface and its frame $\mathfrak{R}_V$ is orthogonal to the road while translating with the motorcycle.

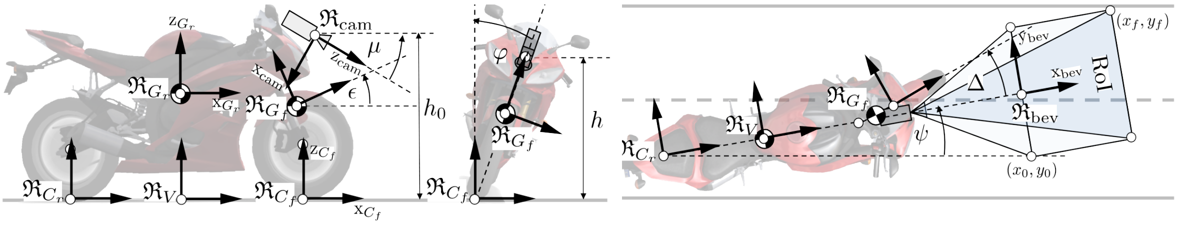

Figure 2. Illustration of the proposed motorcycle-IMU-camera setup

We adopt the following terminology when referring to the motorcycle geometry: The angle through which the front body is rotated with respect to the road normal is the *caster* angle ϵ. The horizontal distance in $\mathfrak{R}_{G_f}$ from the front wheel center to the motorcycle steering axis is the *rake* l_{rk}. The camera is rotated by a static pitch μ in $\mathfrak{R}_{G_r}$ so that the road ahead dominates its *Field of View* (FoV).

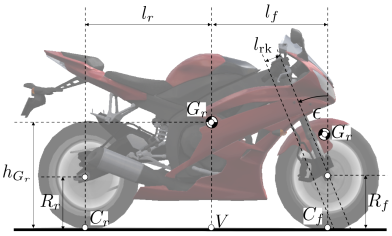

Figure 3. Geometric parameters of the P2WV

FEATURE DETECTION & MATCHING

We define *features* as regions within at least two images of the same scene which could be recognized by a computer vision algorithm. Points, corners or patterns strongly contrasting with their environment make good feature candidates. Road scenes have been challenging for feature detection due to their relative homogeneity. However, digital camera resolutions and embedded processing power have improved at a geometric rate since the early development of computer vision in the 1980s. Hence, we hypothesize that it is now possible to capture enough detail of surface flaws present on roads to robustly extract features. Theoretically, the feature detection ought to work even better on roads with deteriorated lane markings and where the road surface has become potted over time.

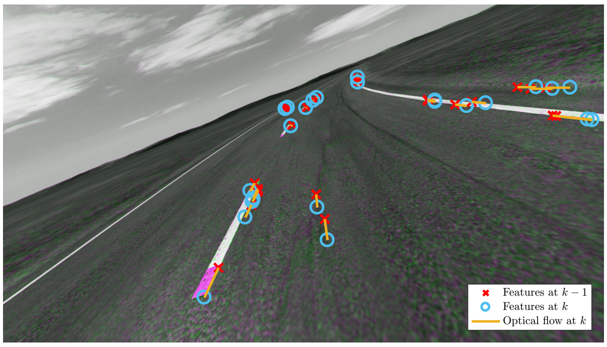

Figure 4. Superimposed image pair taken at timesteps $k-1$ and k. Overlayed are the locations of the detected features at the respective time steps and the optical flow vectors between them.

The Harris corner detector [15] is one of the earliest developed robust feature detection algorithms. Consider a small $n \times n$ window of pixels W in a grayscale image I. A given W is considered a corner if displacing the window in any direction always results in a large change in the intensity gradient. Passing this window over the entire image produces a map of corner features. Referring to Algorithm 1, the implementation is as follows: For every pixel $I\left(x,y\right)$, compute the image gradients in the x and y directions I_x and I_y of the window W surrounding it. Here, we calculate the gradients by convolving with the Sobel kernels G_x and G_y. Construct the structure tensor M from these and compute the *cornerness* score R for each pixel. A higher score indicates the presence of a corner. Finishing steps for the algorithm typically include thresholding R for higher scores followed by non-maximum suppression. There are many more modern feature detectors such as the Scale-Invariant Features Transform (SIFT) and the Features from Accelerated Segment Test (FAST). While we may choose to implement one of these in our approach at a later date, a comparison of feature detectors is not the focus of this work.

Algorithm 1 The Harris corner detector

```
function HarrisCorners($I$,$n$)
    $G_x \leftarrow \begin{bmatrix}1 & 2 & 1\end{bmatrix}^\mathrm{T} * \begin{bmatrix}1 & 0 & -1\end{bmatrix}$
    $G_y \leftarrow \begin{bmatrix}1 & 0 & -1\end{bmatrix}^\mathrm{T} * \begin{bmatrix}1 & 2 & 1\end{bmatrix}$
    $w \leftarrow \frac{1}{2}\left[-n..n\right]$
    $\kappa \in \left[0.04, 0.06\right]$
    for all $x \in I$ do
        for all $y \in I$ do
            $W \leftarrow I\left(x+w, y+w\right)$
            $I_x \leftarrow G_x * W$
            $I_y \leftarrow G_y * W$
            $M \leftarrow \sum_{(x,y)\in W} \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix}$
            $R(x,y) \leftarrow \det M - \kappa \left(\operatorname{tr} M\right)^2$
        end for
    end for
    return $R$
end function
```

The next task is to identify matching features between sequential video frames of the road. We convert each feature pixel window into a *binary descriptor* which describes differences in intensity values in the window. Descriptors have the advantage of simplifying the matching process into thresholding the Hamming distance between two binary strings. Popular descriptor schemes include Binary Robust Independent Elementary Features (BRIEF) and Binary Robust Invariant Scalable Keypoints (BRISK). For this work, we chose the widely-used Fast Retina Keypoint (FREAK) [16].

INVERSE PERSPECTIVE MAPPING

The positions of the features detected in Section are expressed in a *persepctive projection* of the world frame $\mathfrak{R}_\mathrm{w}$ at this stage. Furthermore, these positions are given in pixels rather than meters. IPM constructs a synthetic Bird's Eye View (BEV) from an image taken in perspective. Consider a rectangular *Region of Interest* (RoI) in the real world we wish to view from above defined by corner points $\boldsymbol{p}_0$ and $\boldsymbol{p}_f$ in meters as shown in Figure 2. We convert these corner points into *homogeneous coordinates* $\tilde{\boldsymbol{p}}_0$ and $\tilde{\boldsymbol{p}}_f$ through division by the camera height.

$$\tilde{\boldsymbol{p}}_i = \begin{bmatrix} \frac{1}{h}\boldsymbol{p}_i^\mathrm{T} & 1 \end{bmatrix}^\mathrm{T} \tag{8}$$

We infer the existence of a *virtual camera* looking down from above at the RoI of resolution $m_\mathrm{VC} \times n_\mathrm{VC}$. The pixel intensity values in the images taken by the real camera and those taken by the virtual camera are related by a *Homography* matrix $\mathbf{H}$ [17].

$$\mathbf{H} = \mathbf{K}\left(\mathbf{R}_\mathrm{VC} - \frac{1}{h}\mathbf{t}\mathbf{n}^\mathrm{T}\right)\mathbf{K}_\mathrm{VC}^{-1} \tag{9}$$

Where $\mathbf{R}_\mathrm{VC}$ is the rotation matrix defining the orientation of the virtual camera VC with respect to the real one. $\mathbf{t}$ is the displacement of the virtual camera with respect to the real camera and $\mathbf{n}$ is the unit normal to the road surface. $\mathbf{K}$ and $\mathbf{K}_\mathrm{VC}$ are the *intrinsic parameter matrices* of the real and virtual cameras respectively. We can approximate the inverse of the virtual camera intrinsic matrix by expressing the desired size of each of the pixels in the BEV in world coordinates [6].

$$\mathbf{K}_\mathrm{VC}^{-1} = \begin{bmatrix} \dfrac{\tilde{x}_f - \tilde{x}_0}{m_\mathrm{VC}} & 0 & -\tilde{x}_f \\ 0 & \dfrac{\tilde{y}_f - \tilde{y}_0}{n_\mathrm{VC}} & -\tilde{y}_f \\ 0 & 0 & 1 \end{bmatrix} \tag{10}$$

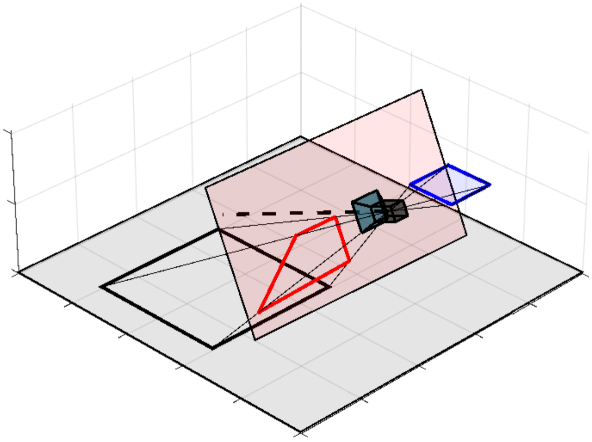

Figure 5. Geometry of the IPM process. First, we define a plane for the BEV (highlighted in blue above) in world frame units. Next, we project this plane onto the camera image plane (highlighted in red). Thus, it becomes our region of interest in the real world (highlighted in black)

Now, we define the orientation of the real camera with respect to VC. Note that in Figure 2 that the real camera z-axis is colinear with the axis of the focal point. First, we re-orient the camera axes so that it is the x-axis instead which projects forwards using $\mathbf{R}_c$.

$$\mathbf{R}_c = \mathbf{R}_y\left({}^{\pi}/_{2}\right)^{\mathrm{T}}\mathbf{R}_x\left({}^{\pi}/_{2}\right) \tag{11}$$

Next, we undo the rotation by the tilt so that the camera position is expressed in the motorcycle body frame $\mathfrak{R}_{G_r}$. Finally, VC yaws with the motorcycle so we need to undo only the roll followed by the pitch to arrive in $\mathfrak{R}_V$, the reference frame in which the position of the virtual camera is defined.

$$\mathbf{R}_{\mathrm{VC}} = \mathbf{R}_c\left(\mathbf{R}_\theta\mathbf{R}_\phi\mathbf{R}_\mu\right)^{\mathrm{T}} \tag{12}$$

We form the intrinsic matrix $\mathbf{K}$ of the real camera using its focal length f and the pixel size $m_x \times m_y$. The translations t_x and t_y compensate for the difference in coordinate origins between an image, where the origin is the top left corner, and the BEV, where the origin is at the bottom center.

$$\mathbf{K} = \begin{bmatrix} \mathrm{f}m_x & 0 & m_xt_x \\ 0 & \mathrm{f}m_y & m_yt_y \\ 0 & 0 & 1 \end{bmatrix} \tag{13}$$

Finally, we construct the homography matrix $\mathbf{H}_{\mathrm{VC}}$. Note that there is no translation: Figure 5 shows that in reality IPM is a *warping* of the RoI in perspective as if the real camera were orthogonal to the world plane facing down, thus the translation is implicitly incorporated into the design of the $\mathbf{K}_{\mathrm{VC}}$.

$$\mathbf{H}_{\mathrm{VC}} = \mathbf{K}\mathbf{R}_{\mathrm{VC}}\mathbf{K}_{\mathrm{VC}}^{-1} \tag{14}$$

Recall the use of homogeneous coordinates in (8): A point $\boldsymbol{p}_{\mathrm{im}}$ in the original image (in pixels) is expressed in the BEV as $\boldsymbol{p}_{\mathrm{bev}}$ using the *inhomogeneous* form of the projective transform where $h_{i,j}$ is an element of $\mathbf{H}$ located at row i and column j.

$$x_{\mathrm{bev}} = \frac{h_{11}x_{\mathrm{im}} + h_{12}y_{\mathrm{im}} + h_{13}}{h_{31}x_{\mathrm{im}} + h_{32}y_{\mathrm{im}} + h_{33}} \tag{15}$$

$$y_{\mathrm{bev}} = \frac{h_{21}x_{\mathrm{im}} + h_{22}y_{\mathrm{im}} + h_{23}}{h_{31}x_{\mathrm{im}} + h_{32}y_{\mathrm{im}} + h_{33}} \tag{16}$$

Having removed the perspective warp from the feature set, we can convert each feature point $\boldsymbol{p}_i^V$ from pixels into meters by multiplying each feature by the inverse of the virtual camera intrinsic matrix to obtain our point sets on the road surface expressed in $\mathfrak{R}_V$.

$$\boldsymbol{p}^V = \mathbf{K}_{\mathrm{VC}}^{-1}\begin{bmatrix} x_{\mathrm{bev}} & y_{\mathrm{bev}} & 1 \end{bmatrix}^{\mathrm{T}} \tag{17}$$

An example of a perspective-corrected point set superimposed onto a synthetic BEV created using IPM and bilinear interpolation is shown in Figure 6. Note that this entire process is reversible by simply multiplying the perspective-corrected points by the inverse of $\mathbf{H}$.

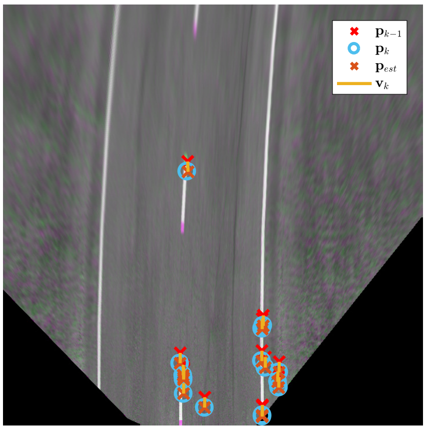

Figure 6. Features transformed into the planar coordinates. Note this time that the motions indicated in yellow are the calculated translational velocities from the ICP and not the optical flow.

Image registration is the process of determining the *affine* transformation between two images of the same scene. Thanks to the IPM, the registration problem here is restricted to two dimensions. We assume that the camera and IPM induce no distortion in the BEV, hence there exists a *rigid transformation* that optimally aligns the perspective-corrected point sets $\boldsymbol{P}_k$ and $\boldsymbol{P}_{k-1}$. Let this transformation be composed of a rotation $\mathbf{R}_\vartheta \in \mathbb{R}^{2\times 2}$ about the virtual camera focal axis by ϑ and a translation $\boldsymbol{d} \in \mathbb{R}^2$ [18].

$$\begin{bmatrix} \boldsymbol{P}_{k-1} \\ \mathbf{I} \end{bmatrix} = \begin{bmatrix} \mathbf{R}_\vartheta & \boldsymbol{d} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{P}_k \\ \mathbf{I} \end{bmatrix} \tag{18}$$

However, it is never the case that the features in image $k-1$ are a perfect rigidly transformed copy of those in image k, but a close approximation can be found using algorithms such as Iterative Closest Point. There exist several variations of ICP including such as point to point, point to plane and generalized ICP. Each of these minimizes a slightly different metric and vary in terms of robustness against the presence of outliers. In all cases though, ICP seeks to minimize the overlap error $\boldsymbol{E}$ between point sets in the least-squares sense [19].

$$\min \sum_{k=1}^{n} \frac{1}{n} \|\boldsymbol{E}\|^2 = \sum_{k=1}^{n} \frac{1}{n} \|\boldsymbol{P}_{k-1} - \mathbf{R}_\vartheta \boldsymbol{P}_k - \boldsymbol{d}\|^2 \tag{19}$$

While iterative least-squares algorithms are more robust to outliers, this work is intended as a proof of concept rather than a final implementation. Thus, we select the well-known Singular Value Decomposition (SVD) variant of ICP summarized in Algorithm 2: First, center both point sets on zero by subtracting their means. Next, form the structure tensor $\boldsymbol{A}$ from these centered sets and take its SVD. Verify that the product of $\boldsymbol{U}$ and $\boldsymbol{V}^{\mathrm{T}}$ is orthogonal and multiply the end column by -1 if necessary. Finally, recover the translation using the difference between $\boldsymbol{P}_{k-1}$ and the rotation-corrected $\boldsymbol{P}_k$.

Algorithm 2 The Iterative Closest Point algorithm

```
function ICP($\boldsymbol{P}_{k-1}$,$\boldsymbol{P}_k$)
    $\boldsymbol{A} \leftarrow \left[\boldsymbol{P}_{k-1} - \bar{\boldsymbol{P}}_{k-1}\right]\left[\boldsymbol{P}_k - \bar{\boldsymbol{P}}_k\right]^{\mathrm{T}}$
    $\boldsymbol{U}, \boldsymbol{\Sigma}, \boldsymbol{V}^{\mathrm{T}} \leftarrow \operatorname{svd} \boldsymbol{A}$
    if $\det \mathbf{R} = -1$ then
        $\mathbf{R} \leftarrow \left(\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \mathbf{R}^{\mathrm{T}}\right)^{\mathrm{T}}$
    end if
    $\mathbf{R} \leftarrow \boldsymbol{U}\boldsymbol{V}^{\mathrm{T}}$
    $\vartheta \leftarrow \arctan\left(\frac{\hat{\boldsymbol{i}}^{\mathrm{T}}\boldsymbol{r}_3}{\hat{\boldsymbol{k}}^{\mathrm{T}}\boldsymbol{r}_3}\right)$
    $\boldsymbol{d} \leftarrow \bar{\boldsymbol{P}}_{k-1} - \mathbf{R}\bar{\boldsymbol{P}}_k$
    return $\vartheta$, $\boldsymbol{d}$
end function
```

It is worth mentioning that the rotation between successive images $k-1$ and k can also be recovered from the IMU yaw rate measurement $\tau\dot{\psi}$ where τ is the system sample time [20].

MOTORCYCLE KINEMATICS

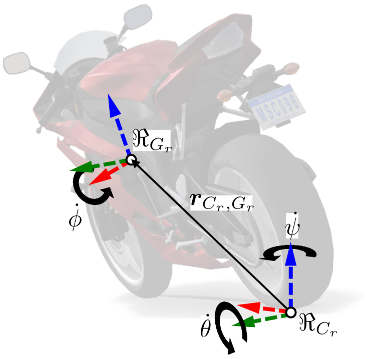

Figure 7. illustration of the relationship between the rear body center of mass and the rear wheel contact point

We assume that the translational velocity estimate from Section is measured in $\mathfrak{R}_V$. Let us approximate the motorcycle wheels as thin disks: Referring to Figure 7 and applying the rotations in the correct order, we obtain $\mathbf{R}_{\theta\phi}$, the rotation matrix transforming from G_r to C_r. From Screw theory, we know that the *twist* of a point i on a rigid body comprised of a translational velocity $\boldsymbol{v}_i^b$ and an angular velocity $\boldsymbol{\omega}^b$ can be expressed in the world frame $\mathfrak{R}_w$ using the *velocity adjoint mapping.* Note that there also exists an inverse mapping to convert from the world frame back into the body frame.

$$\begin{bmatrix} \boldsymbol{v}_i^w \\ \boldsymbol{\omega}^w \end{bmatrix} = \begin{bmatrix} \mathbf{R}_b^w & [\boldsymbol{p}_i]_\times \mathbf{R}_b^w \\ \mathbf{0} & \mathbf{R}_b^w \end{bmatrix} \begin{bmatrix} \boldsymbol{v}_i^b \\ \boldsymbol{\omega}^b \end{bmatrix} \tag{20}$$

$$\begin{bmatrix} \boldsymbol{v}_i^b \\ \boldsymbol{\omega}^b \end{bmatrix} = \begin{bmatrix} \mathbf{R}_w^b & -\mathbf{R}_w^b [\boldsymbol{p}_i]_\times \\ \mathbf{0} & \mathbf{R}_w^b \end{bmatrix} \begin{bmatrix} \boldsymbol{v}_i^w \\ \boldsymbol{\omega}^w \end{bmatrix} \tag{21}$$

Recall from the system description that the IMU measures $\boldsymbol{\omega}_{G_r}$ at G_r in $\mathfrak{R}_{G_r}$. With this in mind, we apply the Adjoint maps to express the ICP velocity estimate from the Section in $\mathfrak{R}_{G_r}$.

$$\begin{bmatrix} \boldsymbol{v}_{G_r}^{G_r} \\ \boldsymbol{\omega}^{G_r} \end{bmatrix} = \begin{bmatrix} \mathbf{R}_{\theta\phi}^{\mathrm{T}} & -\mathbf{R}_{\theta\phi}^{\mathrm{T}} \left[\boldsymbol{r}_{G_r,C_r}^{C_r}\right]_\times \\ \mathbf{0} & \mathbf{R}_{\theta\phi}^{\mathrm{T}} \end{bmatrix} \begin{bmatrix} \boldsymbol{v}_{G_r}^{C_r} \\ \boldsymbol{\omega}^{C_r} \end{bmatrix} \tag{22}$$

$$= \begin{bmatrix} \mathbf{R}_{\theta\phi}^{\mathrm{T}} & \mathbf{I} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \boldsymbol{v}_{G_r}^{C_r} \\ \boldsymbol{\omega}^{G_r} \end{bmatrix} \tag{23}$$

Next, we must derive the velocity of V to express the measured velocity at the front and rear contact points. This time applying the forwards Adjoint mapping, we obtain the following:

$$\begin{bmatrix} \boldsymbol{v}_V^V \\ \boldsymbol{\omega}^V \end{bmatrix} = \begin{bmatrix} \mathbf{R}_{\theta\phi} & \left[\boldsymbol{r}_{G_r,V}^{G_r}\right]_\times \mathbf{R}_{\theta\phi} \\ \mathbf{0} & \mathbf{R}_{\theta\phi} \end{bmatrix} \begin{bmatrix} \boldsymbol{v}_{G_r}^{G_r} \\ \boldsymbol{\omega}^{G_r} \end{bmatrix} \tag{24}$$

Finally, we obtain the front and rear slip angles from the x and y components of $\boldsymbol{v}_V^V$ and the application of Equations (6) and (7). As discussed in the introduction, the leaning motion of the motorcycle means that the yaw rate is not equal to the angular velocity about the z-axis of $\mathfrak{R}_{G_r}$. However, it can be recovered through the orientation with respect to the road plane and the application of the proper *Rate Jacobian.*

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & 0 & -\sin(\theta) \\ 0 & \cos(\phi) & \sin(\phi)\cos(\theta) \\ 0 & -\sin(\phi) & \cos(\phi)\cos(\theta) \end{bmatrix}^{-1} \boldsymbol{\omega}_{G_r} \tag{25}$$

Note that due to the thin disk approximation, we expect to over-estimate the velocity somewhat compared to the ground truth as in reality $\boldsymbol{r}_{C_r,G_r}$ will not remain constant due to tire compression during transient phases.

SENSOR FUSION

The estimator described thus far is summarized in Figure 8: The IMU consisting of a MEMS accelerometer and gyroscope feeds its readings to a standard INS, which outputs roll, pitch and the derivative of body velocity $\dot{\boldsymbol{v}}_{G_r}^{G_r}$. The camera feeds images to Algorithm 1 which outputs features to the IPM. From here, the ICP computes the motion between point sets and provides a planar velocity and yaw rate estimate. The planar velocity is converted into body velocity using the gyroscope measurement and the Adjoint mapping (22). Finally, the integrated INS velocity estimate is corrected for integrator drift using the vision estimate within the Kalman Filter. The velocity is fed back into the INS and the algorithm repeats for all time steps k.

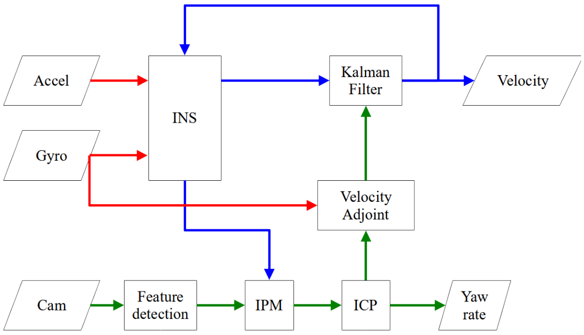

Figure 8. The proposed visual-inertial estimation algorithm: Measurements from the IMU sensors are highlighted in red. Readings from the proposed vision component are highlighted in green. State estimates are highlighted in blue.

We model noise present in the vision measurement as Average White Gaussian Noise (AWGN). We use a discrete *Linear Time-Invariant* (LTI) Kalman filter to produce an estimate of the motorcycle velocity $\boldsymbol{v}_{G_r}^{G_r}$. Our prediction model is a simple integration on the input $\dot{\boldsymbol{v}}_{G_r}^{G_r}$ provided by the INS.

$$\boldsymbol{x} = \begin{bmatrix} v_x^{G_r} & v_y^{G_r} \end{bmatrix}^{\mathrm{T}} \tag{26}$$

$$\boldsymbol{u} = \begin{bmatrix} \dot{v}_x^{G_r} & \dot{v}_y^{G_r} \end{bmatrix}^{\mathrm{T}} \tag{27}$$

$$\boldsymbol{F} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{28}$$

$$\boldsymbol{B} = \begin{bmatrix} \tau & 0 \\ 0 & \tau \end{bmatrix} \tag{29}$$

Where $\boldsymbol{x}$ and $\boldsymbol{u}$ denote the state and input vectors. Concordantly, $\boldsymbol{F}$ and $\boldsymbol{B}$ are the state and input transition matrices. Let $\boldsymbol{P}$ and $\boldsymbol{Q}$ be the error and process noise covariance matrices. The *prediction* equations are summarized below where the superscript $^-$ denotes an *a priori* state or covariance estimate.

$$\hat{\boldsymbol{x}}_k^- = \boldsymbol{F}\hat{\boldsymbol{x}}_{k-1}^- + \boldsymbol{B}\boldsymbol{u}_k \tag{30}$$

$$\boldsymbol{P}_k^- = \boldsymbol{F}\boldsymbol{P}_{k-1}^-\boldsymbol{F}^{\mathrm{T}} + \boldsymbol{Q} \tag{31}$$

Referring to Figure 8 we apply (22) to the translational velocity measured by our vision algorithm to obtain our Kalman filter measurement. We summarize our measurement model below where $\boldsymbol{z}$ is the measurement vector and $\boldsymbol{H}$ is the measurement model.

$$\boldsymbol{z} = \begin{bmatrix} v_x^{\mathrm{ICP}} & v_y^{\mathrm{ICP}} \end{bmatrix}^{\mathrm{T}} \tag{32}$$

$$\boldsymbol{H} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{33}$$

We complete the *update* step of Kalman filtering by computing the Kalman gain $\boldsymbol{K}$ and the *a posteriori* estimate and covariance matrix. Note that $\boldsymbol{R}$ is the measurement covariance matrix.

$$\boldsymbol{K}_k = \boldsymbol{P}_k^-\boldsymbol{H}^{\mathrm{T}}\left(\boldsymbol{H}\boldsymbol{P}_k^-\boldsymbol{H}^{\mathrm{T}} + \boldsymbol{R}\right)^{-1} \tag{34}$$

$$\hat{\boldsymbol{x}}_k = \hat{\boldsymbol{x}}_k^- + \boldsymbol{K}_k\left(\boldsymbol{z}_k - \boldsymbol{H}\hat{\boldsymbol{x}}_k^-\right) \tag{35}$$

$$\boldsymbol{P}_k = \left(\boldsymbol{I} - \boldsymbol{K}_k\boldsymbol{H}\right)\boldsymbol{P}_k^- \tag{36}$$

RESULTS

To validate our approach, we designed three scenarios in BikeSim mechanical simulation software. BikeSim allows developers to test their ARAS desgins on realistic multi-body simulations of motorcycles. We tested our vision algorithm on three different driving scenarios. The first scenario sees the motorcycle perform a series of *slaloms* along a straight road at 50 km/h. This scenario was selected in order to test our estimator at high values of steering angle. The second scenario sees the motorcycle negotiate an S-shaped bend at 80 km/h. We chose this scenario to test our estimator's ability to cope with curved roads as well as straights. The third is the *Double Lane-Change* (DLC) often featured in literature where the motorcycle changes from one lane to another and back again on a straight road. We perform the DLC at 110 km/h. Here, the scenario was chosen in order to test the

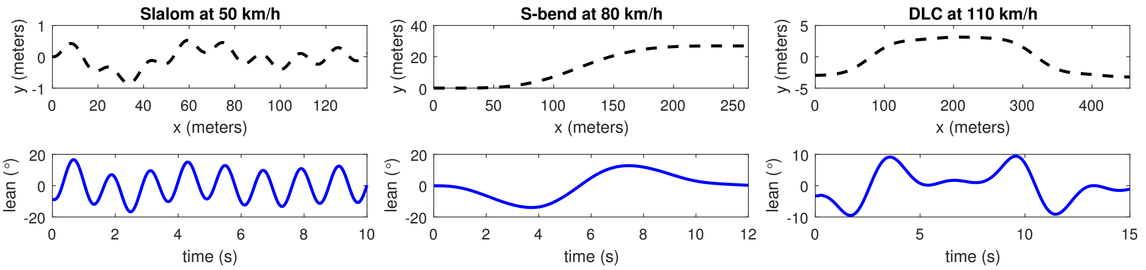

Figure 9. Trajectory and lean profiles of the three scenarios

algorithm's ability to cope with elevated speeds typically experienced by riders on European extra-urban roads. The paths and lean profiles of each scenario are illustrated in Firgure 9.

We implemented our estimator in MATLAB/Simulink in co-simulation with BikeSim at a sampling rate of 60 Hz. The outputs from the camera were 720 × 1280 RGB images. We defined our RoI with corner points $(x_0, y_0) = (3\,\text{m}, -10\,\text{m})$ and $(x_f, y_f) = (23\,\text{m}, 10\,\text{m})$ in $\mathfrak{R}_V$. The BEV resolution was 720 × 720 and we added noise to the accelerometer reading $\boldsymbol{a}_{G_r}^{\text{accel}}$ with a Signal to Noise Ratio SNR $= 30$ to test our estimator's robustness to integrator drift. The process and measurement covariance matrices were set to $\boldsymbol{Q} = 1$ and $\boldsymbol{R} = 1 \times 10^5$.

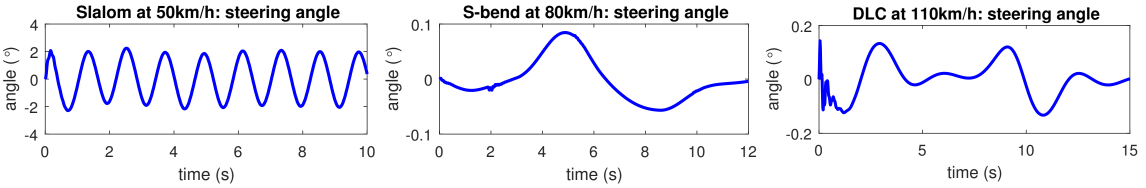

Figure 10. Steering angle inputs for each scenario

The following subsections present the results of the estimates obtained presented against ground truth references generated in BikeSim. In Figures 11, 12 and 13, the acronyms ICP, DR, and KF denote Iterative Closest Point, Dead-Reckon (of the INS velocity rate estimate) and Kalman Filter respectively.

Longitudinal Velocity

We observe from Figure 11 that the ICP estimate rests around the reference value, if not slightly underestimating. This is an encouraging result as other estimation algorithms such as Optical Flow fail to register large displacements between images which is not the case here. However, in the Slalom scenario, the variance is much higher compared to the S-bend and DLC.

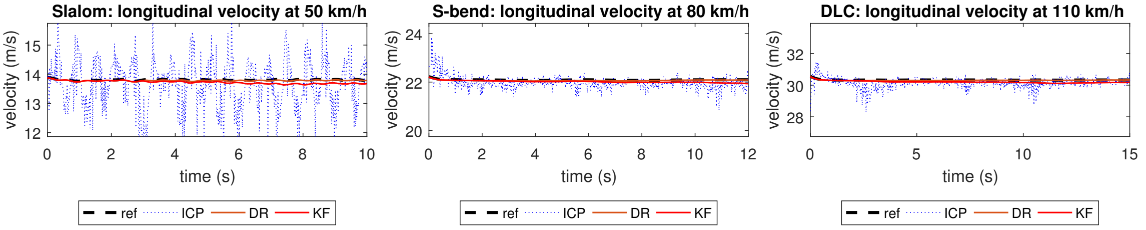

Figure 11. Longitudinal velocity results

Lateral Velocity

Recall from the motivation that a key technical goal for our estimator is the reconstruction of the lateral velocity. The results displayed in Figure 12 are encouraging: We note that in the Slalom scenario, the vision system struggles to keep pace with the rapid succession of transients. However, this is compensated for by the INS measurement where integrator drift is less of an issue due to the rapidly-varying input. We observe that the both the INS and vision measurements track the reference very well in the S-Bend scenario.

Finally, we note that in the DLC the ICP estimate deviates slightly from the reference especially during transients corresponding to sudden changes in lean (see Figure 9). In all cases, we note that the integrated INS alone slowly drifts from the reference as the errors from the noise added to the accelerometer reading accumulate. As hypothesized, our vision algorithm measurement successfully compensates for this drift and drags the filtered estimate back toward the ground truth.

Yaw Rate

So far, there has been significant noise present in all the ICP estimates obtained. It is clear from Figure 13 that this noise is especially present in the yaw rate estimate obtained from the estimated rotation matrix in

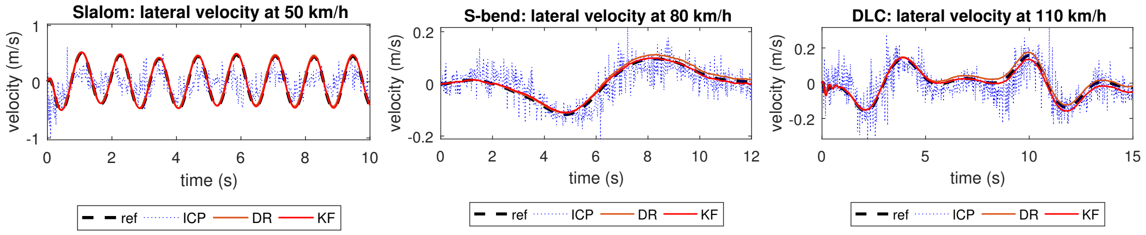

Figure 12. Lateral velocity results

Algorithm 2. While the mean of this noise appears to track the reference well, we note in the S-bend and DLC scenarios that there is slight overshoot at the peaks of motion.

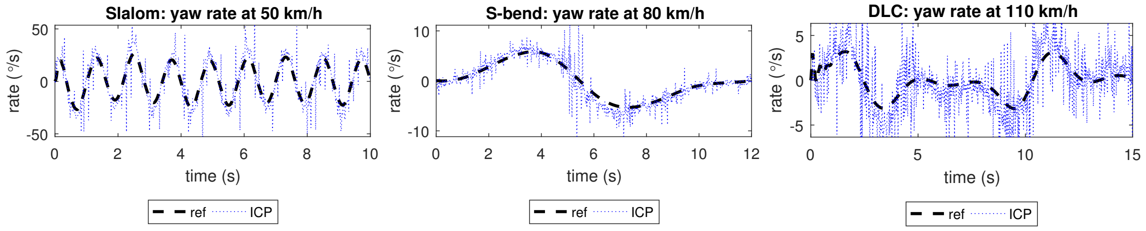

Figure 13. Yaw rate results

Front Slip

Having obtained our velocity estimates, we re-express them in $\mathfrak{R}_V$ using (24) and pass the results to (6) and (7). It should be noted that these expressions for front and rear slip are themselves linearized approximations of the ground truth therefore there is an upper bound on achievable performance. Recall that a front slip larger than the rear slip indicates under-steering: We observe from Figure 14 good tracking in the Slalom scenario where front slip is highest. Note also from the steering inputs in Figure 10. that the majority of the angle between direction and velocity in $\mathfrak{R}_{C_f}$ is due to the steering angle. We note an underestimation in the S-bend and DLC.

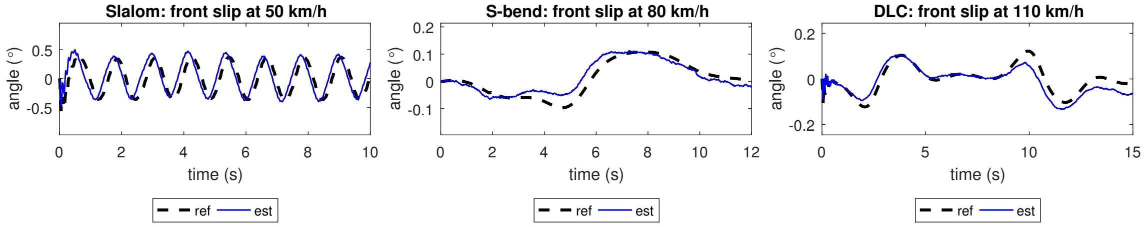

Figure 14. Front slip results

Rear Slip

The results for the rear slip angle are displayed in Figure 15. The S-bend and DLC estimates track much better than their front slip counterparts though we note that the magnitude of the references here are more significant. We observe that the estimate lags in the Slalom scenario.

DISCUSSION

The initial evaluation of our results is encouraging. Referring to Figures 12 we note that our results compare favorably with the Kalman-based approach of Teerhuis and Jansen [21] despite there being no a priori model in our estimator. Regarding performance against observer-based designs, note that the estimate during the peak regions of the DLC outperforms the estimation by the UI-HOSM [11] while performing the same maneuver. This comparison is of particular importance since the results in [11] were also obtained using BikeSim. Recall that in [22], the absolute error magnitude in their observer lateral velocity estimation during a DLC using a nominal LPV model was just over 0.12 m/s. We note an error in our DLC result of under 0.05 m/s, outperforming that paper's LPV observer. Of particular importance is that while the control inputs of the scenarios in [22] are generated using BikeSim, they are validated against a theoretical model. Thus, we can conclude that our estimator outperforms theirs even in the best case. In [13], a highly sophisticated multi-model observer based on Tagaki-Sugeno and *Linear Matrix Inequalities* (LMI) techniques is presented and demonstrates the most promising results for observer-based lateral velocity estimates to date. While their observer does reach the correct final value in steady-state periods, it fails to reconstruct the waveforms of transients compared to our approach.

In the Motivation, we mentioned that longitudinal velocity can be recovered from the vehicle engine odometer under the assumption that no longitudinal slip is present between the road and the tires. This is nevertheless a harsh assumption to make, especially in road conditions where such slip is likely to be present such as wet

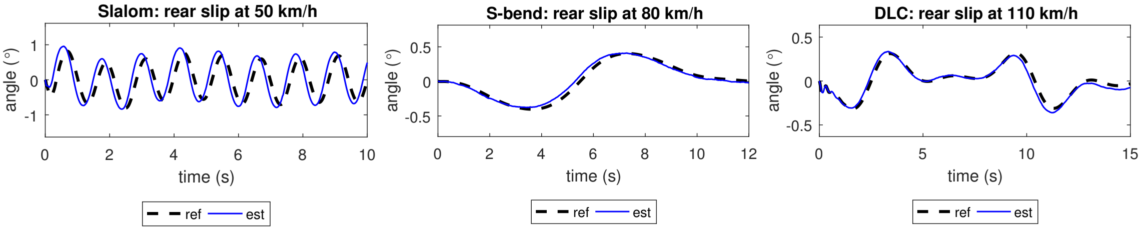

Figure 15. Rear slip results

weather. For this work, we chose to work with constant longitudinal velocities since the primary motivation is to reconstruct lateral motion and the front and rear slip angles. Future works on our algorithm should test its ability to estimate a varying longitudinal velocity.

Earlier works published by our research team attempted to measure lateral velocity using only information from BEV images of the center lane markers [23]. We identified two major issues with this approach which in part motivated this new design: The sudden appearance and removal of transitioning lane markers entering the BEV induced a large-amplitude low-frequency oscillation in the vision system measurement. We have overcome this by using road surface detailing as tracked features as well as lane markers as explained in Section . The second major issue was computational complexity: Producing a full grayscale BEV using image interpolation is extremely intensive for any application. We proposed to reduce complexity through image binarization. However, this introduced a new source of noise into our measurement. Here, we transform only the features detected in the camera images into the inverse perspective, greatly reducing computational complexity whilst simultaneously decreasing noise.

Nonetheless, there is strong noise present in all of our ICP measurements originating outside of the sources discussed. A known issue with point registration is how one deals with *outliers*. A common technique is to detect them using RAndom Sample Consensus (RANSAC) [24]: Future works on this approach should investigate the influence of outliers on our results. We are also limited by the BikeSim software itself as the simulated IMU readings from the physics engine and the simulated image stream from the visualizer are asynchronous. This leads to imperfect IPM, especially at high roll angles and we hypothesize that this is the cause of the deviations from the ground truth observed in Figures 12 and 13. It may be possible to reduce measurement noise by increasing the sample rate of the video stream and the resolution of the camera.

As mentioned in the front slip results, there is an upper limit to the possible accuracy of the slip estimates using (6) and (7). Nevertheless, our estimates do approximate the transient behavior well if under-estimating their magnitude. It would be worthwhile in future works to implement our estimator in its current form as an ARAS to detect critical steering scenarios and test it against simulations of motorcycle crashes in BikeSim.

CONCLUSIONS

In this article, we propose a method of estimating the velocity and wheel lateral slip angles of a motorcycle traveling at high speed along a single-carriageway road. We extract feautures from images captured by a camera fixed to the front of the motorcycle and use Harris corner detection to extract features. We remove the perspective projection from the point sets using IPM and recover ego-motion using Iterative Closest. Finally, we obtain a velocity estimate using sensor fusion with an Inertial Navigation System and a Kalman filter. We validated our approach against three different BikeSim simulation scenarios and compared the obtained results with other approaches. We conclude that our approach compares very well to the state of the art and surpasses it in a few cases. We recommend that future works should investigate the effects of outliers, sensor resolution and frame rate on the accuracy of our vision measurements. We will refine our algorithm in simulations under non-ideal conditions such as rain, poor marker visibility, and a rough road surface. Furthermore, we will perform tests under more diverse scenarios such as periods of longitudinal acceleration and simulated crashes. Eventually, we hope to validate our approach using real-world data from an experimental P2WV rig situated at IBISC.

REFERENCES

[1] ONSIR. Le parc automobile des ménages. resreport, Observatoire national interministériel de la sécurité routière, September 2021. URL {`https://www.onisr.securite-routiere.gouv.fr/etudes-et-recherches/vehicules/parc-des-vehicules/le-parc-deux-roues-motorises-des-menages`}.

[2] Andrea D. Furlan, Tara Kajaks, Margaret Tiong, Martin Lavallière, Jennifer L. Campos, Jessica Babineau, Shabnam Haghzare, Tracey Ma, and Brenda Vrkljan. Advanced vehicle technologies and road safety: A scoping review of the evidence. *Accident Analysis & Prevention*, 147, 2020. ISSN 0001-4575. doi: 10.1016/j.aap.2020.105741.

[3] Peter Corke. An inertial and visual sensing system for a small autonomous helicopter. *Journal of Robotic Systems*, January 2004. doi: 10.1002/rob.10127.

[4] D. Ortín and J. M. M. Montiel. Indoor robot motion based on monocular images. *Robotica*, 19(3):331–342, 2001. doi: 10.1017/S0263574700003143.

[5] Bojian Liang and N. Pears. Visual navigation using planar homographies. In *Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No.02CH37292)*, volume 1, pages 205–210 vol.1, May 2002. doi: 10.1109/ROBOT.2002.1013362.

[6] Pierre-Marie Damon, Hicham Hadj-Abdelkader, Hichem Arioui, and Kamal Youcef-Toumi. Inverse perspective mapping roll angle estimation for motorcycles. In *2018 15th International Conference on Control, Automation, Robotics and Vision (ICARCV)*, pages 349–354, November 2018. doi: 10.1109/ICARCV.2018.8581182.

[7] Vittore Cossalter. *Motorcycle Dynamics.* Lulu, second edition, October 2006. ISBN 1430308613.

[8] V. Cossalter, R. Lot, and M. Peretto. Steady turning of motorcycles. *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, 221(11):1343–1356, 2007. doi: 10.1243/09544070JAUTO322.

[9] CMC. Accurate localisation for motorcycles remains a challenge. Technical report, Connected Motorcycle Consortium, April 2019. URL `https://www.cmc-info.net/news/archives/04-2019`.

[10] R. S. Sharp. The Stability and Control of Motorcycles. *Journal of Mechanical Engineering Science*, 13(5): 316–329, 1971. doi: 10.1243/JMES_JOUR_1971_013_051_02.

[11] Lamri Nehaoua, Dalil Ichalal, Hichem Arioui, Jorge Davila, Saïd Mammar, and Leonid M. Fridman. An unknown-input hosm approach to estimate lean and steering motorcycle dynamics. *IEEE Transactions on Vehicular Technology*, 63(7):3116–3127, September 2014. ISSN 1939-9359. doi: 10.1109/TVT.2014.2300633.

[12] Pierre-Marie Damon. *Estimation pour le développement de systèmes d'aide à la conduite des véhicules à deux-roues motorisés.* Theses, Université Paris-Saclay ; Université d'Evry-Val-d'Essonne, November 2018. URL `https://hal.archives-ouvertes.fr/tel-02024804`.

[13] M. Fouka, L. Nehaoua, H. Arioui, and S. Mammar. Interconnected observers for a powered two-wheeled vehicles: Both lateral and longitudinal dynamics estimation. In *2019 IEEE 16th International Conference on Networking, Sensing and Control (ICNSC)*, pages 163–168, May 2019. doi: 10.1109/ICNSC.2019.8743290.

[14] P-M Damon, D. Ichalal, L. Nehaoua, H. Arioui, and S. Mammar. Rider weight consideration for observer design with an application to the estimation of the lateral motorcycle dynamics and rider's action. In *2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 3237–3242, 2017. doi: 10.1109/SMC.2017.8123127.

[15] Chris Harris and Mike Stephens. A combined corner and edge detector. In *Proceedings of the Alvey Vision Conference*, 1988. doi: 10.5244/C.2.23.

[16] Alexandre Alahi, Raphael Ortiz, and Pierre Vandergheynst. FREAK: Fast Retina Keypoint. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 510–517, June 2012. doi: 10.1109/CVPR.2012.6247715.

[17] Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision.* Cambridge University Press, second edition, 2004. ISBN 978-0-521-54051-3. doi: 10.1017/CBO9780511811685.

[18] Y. Liu, T.S. Huang, and O.D. Faugeras. Determination of camera location from 2-d to 3-d line and point correspondences. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(1):28–37, January 1990. ISSN 1939-3539. doi: 10.1109/34.41381.

[19] Fang Wang and Zijian Zhao. A survey of iterative closest point algorithm. In *2017 Chinese Automation Congress (CAC)*, pages 4395–4399, October 2017. doi: 10.1109/CAC.2017.8243553.

[20] Pierre-Marie Damon, Majda Fouka, Hicham Hadj-Abdelkader, and Hichem Arioui. Vision-based lane crossing point tracking for motorcycles. In *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, pages 3399–3404, October 2019. doi: 10.1109/ITSC.2019.8917206.

[21] A. P. Teerhuis and S. T. H. Jansen. Motorcycle state estimation for lateral dynamics. *Vehicle System Dynamics*, 50(8):1261–1276, 2012. doi: 10.1080/00423114.2012.656655.

[22] P. M. Damon, H. Dabladji, D. Ichalal, L. Nehaoua, and H. Arioui. Estimation of lateral motorcycle dynamics and rider action with Luenberger observer. In *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, pages 2392–2397, November 2016. doi: 10.1109/ITSC.2016.7795941.

[23] Martin Pryde, Lamri Nehaoua, Hicham Hadj-Abdelkader, and Hichem Arioui. Visual-inertial lateral velocity estimation for motorcycles using inverse perspective mapping. In *The 17th International Conference on Control, Automation, Robotics and Vision (ICARCV 2022)*, December 2022.

[24] Davide Scaramuzza. 1-point-RANSAC structure from motion for vehicle-mounted cameras by exploiting non-holonomic constraints. *International Journal of Computer Vision*, 2011. doi: 10.1007/s11263-011-0441-3.