Trajectory Prediction on an ego-bike with IMU and camera using a Multi-State Constraint Kalman Filter

MAOUHEB BESSI, University of Twente, The Netherlands

Fig. 1. Trajectory Prediction

ABSTRACT

Trajectory prediction is a crucial task for ensuring the safety and reliability of transportation systems, including bicycles. While the Multi-State Constraint Kalman Filter (MSCKF) algorithm has been successfully applied to trajectory prediction for drones and cars, its effectiveness for predicting bicycle trajectories remains uncertain. In this study, we investigate the feasibility and accuracy of using the MSCKF algorithm for bicycle trajectory prediction. The aim is to utilize a cost-efficient setup, employing a simple low-cost camera and an inexpensive IMU. We compare the performance of the different algorithms on existing data sets to determine whether the MSCKF is a suitable algorithm. This study serves as a significant step toward the development of more effective and accurate trajectory prediction methods tailored specifically for bicycles.

Additional Key Words and Phrases: Accuracy, MSCKF, OpenVINS, Bicycle safety, Autonomous bicycles, Trajectory prediction

1 INTRODUCTION

Accurately predicting bicycle trajectories is crucial for ensuring safety and efficiency in modern transportation systems. The amount of deadly bicycle accidents in the Netherlands has increased with 54% 2, therefore it is important to implement measures to help reduce these numbers. This study investigates whether MSCKF is a feasible method for the trajectory prediction of bicycles. Despite encountered challenges, this research sheds light on limitations and provides insights for improving bicycle trajectory prediction.

The problem of predicting bicycle trajectories accurately and efficiently is important for the safety and reliability of bicycles in modern traffic. Despite the development of trajectory prediction methods for other types of autonomous vehicles, such as drones and self-driving cars, there has been limited research on these methods for the prediction of bicycle trajectories. One of the reasons which make it a challenge for bicycles is due to the characteristics of a

Tabel 3: dodelijke verkeersslachtoffers verdeeld naar vervoerswijze (2016-2022)

Vervoerswijze	2016	2017	2018	2019	2020	2021	2022
Personenauto	231	201	233	237	195	175	225
Fiets	189	206	228	203	229	207	291
Voetganger	51	58	54	49	41	43	57
Motor/scooter	45	51	42	52	44	52	46
Bromfiets en snorfiets	41	41	38	42	33	49	47
Brommobiel	3	5	5	3	3	1	4
Gemotoriseerd invalidenvoertuig	38	25	44	42	34	32	39
Bestelauto	23	19	20	22	19	12	17
Vrachtauto	6	6	6	6	4	2	8
Overig en onbekend	2	1	8	5	8	9	3
Totaal	629	613	678	661	610	582	737

Fig. 2. CBS data on deadly accidents

TScIT 39, July 7, 2023, Enschede, The Netherlands

bicycle, its limited capacity for the hardware and their limited computing power. Therefore, there is a need for research to investigate the feasibility and accuracy of using trajectory prediction methods for predicting bicycle trajectories, as well as identifying potential areas for improvement in these methods. The first step is to research the different algorithms that exist for the prediction of a trajectory. This leads to the following research question: **What trajectory prediction algorithm is best suitable for a bicycle given its constraints?**

1.1 Research Questions

(1) What hardware should we use?
(2) What algorithms are applicable to a trajectory prediction?
(3) How is the MSCKF the most suitable algorithm for a bicycle trajectory prediction?

2 ANALYSIS OF THE STATE-OF-THE-ART

In order to conduct the research on bicycle trajectory prediction, several sources were used to obtain relevant research papers. The sources used were Google Scholar, IEEE, and ScienceDirect, all of which provided access to various articles on the subject. Keywords such as "MSCKF", "OpenVINS", "Trajectory Prediction", "Bicycle Trajectory Prediction" and "Autonomous Vehicle Safety" have been used to retrieve the research papers.

As for the field of autonomous vehicles, a lot of research has been conducted, which can be categorized into surveys [5, 6], experimental studies [1, 13], and theoretical studies [7, 11, 15]. Surveys provide an overview of the current state of the field, highlighting possible advancements and identifying areas for future improvement. Experimental studies involve actual testing of autonomous vehicles to evaluate their performance in various situations. Theoretical studies dive into the underlying algorithms and mathematical models of the autonomous vehicles.

Many of these papers provide detailed insights into trajectory prediction systems and suggest possible ways to enhance the accuracy of the prediction models on autonomous vehicles. However, there has been relatively little research conducted on bicycle trajectory prediction in specific. While some papers have presented theoretical methods for predicting certain trajectories of cyclists, such as starting and stopping motions [16], there is still much to be explored in this field.

With the increasing popularity of bicycles as a mode of transportation, the need for more effective and accurate trajectory prediction methods is becoming increasingly urgent. Accurate trajectory prediction is vital for ensuring the safety of cyclists, particularly when it comes to interactions with other autonomous vehicles.

2.1 Existing Algorithms

Existing algorithms for the problem can be broadly categorized into three types: physics-based models, data-based models, and hybrid models.

2.1.1 Model based. One area of research focuses on model-based approaches for trajectory prediction. Kalman Filters (KF) have been widely employed in trajectory prediction due to their ability to estimate the state of an object based on noisy measurements [12]. However, the assumption of linear motion dynamics limits its applicability to scenarios with simple motion patterns and noise characteristics.

To overcome the limitations of KF, researchers have explored extensions such as the Extended Kalman Filter (EKF) and Unscented Kalman Filter (UKF). The EKF linearizes the motion and measurement equations, enabling the estimation of nonlinear systems [12]. The UKF, on the other hand, employs a deterministic sampling technique to propagate the state distribution through the nonlinear motion model, resulting in more accurate predictions.

The Multi-State Constraint Kalman Filter (MSCKF) is an extension of the Kalman Filter that addresses the limitations of traditional KF and UKF approaches. The MSCKF represents the state as a set of both camera poses and IMU states and maintains a joint state for all the camera poses and the bicycle's dynamics [8]. This representation allows for better modeling of the system and better integration of sensor information.

2.1.2 Data based. Data-driven approaches have also gained significant attention in trajectory prediction. Machine learning techniques, such as Support Vector Machines (SVM), have been used to learn patterns from historical data and make predictions based on the learned model [5]. SVM-based methods extract relevant features from the data and learn decision boundaries that separate different classes of trajectories.

Deep learning approaches, particularly Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) networks, have shown promising results in trajectory prediction. RNNs and LSTMs are capable of understanding how things change over time in trajectory data and can use this sequential information to make accurate predictions [2, 5]. These deep learning models have been applied to various scenarios, including pedestrian trajectory prediction and autonomous driving.

2.1.3 Hybrid. Researchers have also investigated hybrid approaches that combine model-based and data-driven methods. One such approach is the use of Particle Filters, also known as Monte Carlo Methods. In Particle Filters, the object's state is represented by a group of particles, and their distribution is updated based on sensor measurements [5]. This combination of motion models and likelihood estimation helps in making reliable predictions, especially in complex situations.

Furthermore, there is a growing interest in leveraging deep learning techniques within the framework of model-based approaches. Deep Kalman Filters (DKF) have been proposed to learn the dynamics of object motion from data and perform probabilistic trajectory prediction [3]. These methods combine the modeling capabilities of deep neural networks with the probabilistic inference of Kalman Filters, leading to improved trajectory prediction accuracy.

2.1.4 Summary. In summary, trajectory prediction has been a subject of extensive research, with various algorithms and techniques

proposed in the literature. Model-based methods, such as Kalman Filters and their extensions, have been widely used but have limitations in handling nonlinear motion. Data-driven approaches, including SVM and deep learning models, offer promising results by learning patterns from data. Hybrid approaches that combine motion models and likelihood estimation, such as Particle Filters, provide reliable predictions. The recent integration of deep learning techniques into model-based approaches shows potential for further improving trajectory prediction accuracy.

2.2 Why Multi-State Constraint Kalman Filter?

For bicycle trajectory prediction, the Multi-State Constraint Kalman Filter (MSCKF) is a suitable choice of algorithm due to several reasons.

Firstly, the MSCKF is designed to combine information from different sensors, such as cameras and inertial measurement units (IMUs), to make accurate predictions about how the bicycle will move. This is especially beneficial in the context of bicycle trajectory prediction, as the use of multiple sensors allows for a better understanding of the bicycle's dynamics that may affect its motion.

Secondly, the MSCKF is known to perform well in different situations, even when there are obstacles in the environment or lighting conditions vary. This is crucial for bicycle trajectory prediction, as bicycles often navigate through complex environments where obstacles, such as vehicles, pedestrians, and road infrastructure, can significantly impact their motion. The ability of the MSCKF to handle such scenarios enhances its applicability for bicycle trajectory prediction.

Thirdly, the MSCKF provides real-time estimates of the bicycle's trajectory even with limited hardware, which is important when quick results are required. In applications such as real-time navigation assistance or collision avoidance systems for bicycles, having timely and accurate trajectory predictions is crucial for ensuring safety and efficiency. The MSCKF's ability to provide real-time estimates aligns well with the requirements of such applications.

While the MSCKF shows promise for bicycle trajectory prediction, it is important to note that further research is needed to validate its performance specifically in the context of bicycles. Factors such as the unique dynamics and maneuverability of bicycles, as well as the constraints imposed by limited hardware resources, should be considered and evaluated when implementing the MSCKF for bicycle trajectory prediction.

In conclusion, the Multi-State Constraint Kalman Filter (MSCKF) is a suitable algorithm for bicycle trajectory prediction because it can use information from different sensors, perform well in difficult situations, and provide real-time estimates, the algorithm has the advantage of being able to handle complex scenarios with reliable results. However, further investigation and experimentation are necessary to assess its effectiveness and optimize its parameters for the specific characteristics of bicycle motion.

3 HOW DOES A MULTI-STATE CONSTRAINT KALMAN FILTER WORK?

The Multi-State Constraint Kalman Filter (MSCKF) is a tool that helps estimate the position and movement of something in real-time. It does this by combining information from sensors that can sense motion (like accelerometers and gyroscopes) and cameras that can see the environment. The information was obtained from the following sources: [4, 8, 12, 14]

3.1 Representing the System

MSCKF keeps track of two main things:

- Information about the moving object: It remembers where the object is, how fast it is moving, which way it is facing, and other important details.
- Information about the visual things it sees: It stores the 3D positions of objects (like landmarks) that the camera captures.

3.2 Predicting the Future

The MSCKF uses the information from the IMU (like accelerometers and gyroscopes) and a mathematical model to guess where the object will be in the future based on where it was before and how it was moving. This helps it make real-time predictions about the object's position and movement.

The prediction equations can be represented as follows:

$$\mathbf{x}_t = f(\mathbf{x}_{t-1}, \mathbf{u}_{t-1}) + \mathbf{w}_t$$

Where:

- $\mathbf{x}_t$ is the estimated state of the object at time t.
- $\mathbf{u}_{t-1}$ represents the motion sensor data (e.g., accelerometers and gyroscopes) at time $t-1$.
- $f(\mathbf{x}_{t-1}, \mathbf{u}_{t-1})$ is a mathematical function that predicts the object's state based on the previous state and motion sensor data.
- $\mathbf{w}_t$ is a term that accounts for any uncertainty or errors in the prediction.

3.3 Updating with Camera Information

The MSCKF also looks at what the camera sees. The camera takes pictures, and from those pictures, the MSCKF can figure out the position of visual objects (landmarks) in 3D space. It then uses this information to update its estimate of the object's position and movement. An example of the visual odometry can be seen in 3.

4 IMPLEMENTATION

In this study, we tried to expirement with the MSCKF ourselves using a tool like OpenVINS. We collected the data ourselves on the bicycle. The first step to conduct this study is to research what modules are needed. We decided to use a Raspberry Pi as the central unit, a simple Arduino for the IMU, a Raspicamera for the camera input, a powerbank with sufficient power to power this setup and a ZED-F9P GPS-RTK module for the ground-truth Global Navigation Satelite System (GNSS) data. We ran into issues of insufficient power at first, the powerbank has to be at least 5 Volts and 2.5 Ampère. The ZED-F9P was chosen because of its very high accuracy (within

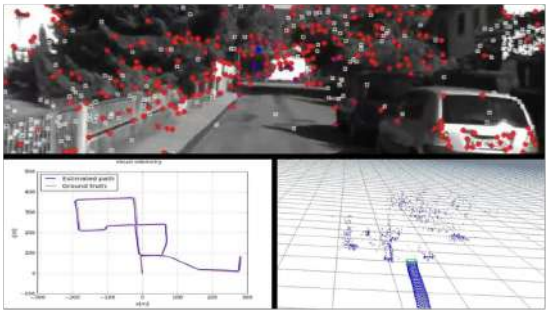

Fig. 3. Feature extraction from landmarks in an image

centimeter level). A centimeter level accuracy is needed for a bicycle to determine how accurate the predicted trajectory is compared to the ground-truth.

4.1 Installation

4.1.1 Modules. Installing the hardware modules correctly on the bicycle is a time consuming process. The first few data sets that we collected were invalid, because of the incorrect installation of the modules on the bicycle. The camera was pointed too much to the sky and it moved along with the wheel. Having taken that into consideration in the second attempt: we had to ensure that the camera and the IMU were attached as close to each other as possible in front of the bicycle, as can be seen in Figure 4. The camera had to point downwards to record the road in front of it, without the mudguard of the front wheel in the frame. A very important detail is that the camera and IMU have to be attached to a part of the bicycle that does not move along with the wheel. This is to ensure that the camera and the IMU are in sync and that the camera does not suddenly take a different angle while the IMU is going straight. We attached the GNSS antenna at the back of the bicycle, as can be seen in Figure 7. All the scripts from the collection of the data to the post-processing and ROS bag creation were written by ourselves.

4.2 Calibration

The next step is to calibrate the camera and IMU to retrieve their intrinsic and extrinsic values. Extrinsic values deal with the position and orientation of the camera and IMU relative to each other. They help align the data from both sensors correctly. Intrinsic values are specific to each sensor. For the camera, it includes things like focal length and lens distortion. For the IMU, it involves factors like biases and misalignments. Calibrating intrinsic values helps correct errors and ensure accurate measurements. By calibrating both extrinsic and intrinsic values, we can accurately relate data from the camera and IMU, correct errors, and obtain precise and synchronized information as an input to the MSCKF [9].

4.3 Data collection

All the data that was necessary for this study was collected during the research. We used OpenCV to capture the images while reading the GNSS data using a PyRTCM. The IMU was read over cable using Serial. The refresh rate for the GNSS is 1Hz, which is limited by

Fig. 4. Camera and IMU setup

Fig. 5. Raspberry Pi and GNSS module setup

Fig. 6. Powerbank setup

Fig. 7. GNSS antenna setup

Fig. 8. Long trajectory with turns

the caster of the reference station. The refresh rate for the IMU and camera are 119Hz and 75Hz respectively. The data contains gray-scaled images with timestamps, accelerometer and gyroscope from the IMU with its timestamps and the latitude and longitude from the GNSS module along with its timestamps.

4.3.1 Trajectories. We decided to take multiple rounds of data on each trajectory for a total of two different trajectories. The collected data has a constant speed of 13 kilometers per hour with the same cyclist in each data set. One of the trajectories contains many turns and was longer, as can be seen in Figure 8. The other trajectory are two simple straight paths, as can be seen in Figure 9.

4.4 Post-processing

The post-processing of the data is important to prepare it for the tools. The recorded images had a consistent 90-degree offset to the left, because of the setup of the camera on the bicycle. This required a rotation step in the script to align them correctly. All the images were saved as a gray-scale 8-bit PNG file with a 480x640 resolution. All the timestamps of the IMU, GNSS and images were saved in a date and time format. This had to be translated into nanoseconds since 1970 (also known as epoch time), as OpenVINS requires this.

4.4.1 ROS Bag. The next step involves creating a ROS bag to store the data from the IMU and camera. A ROS bag is a file that stores the received message data in a serialized format [10]. Think of it like a video playback that consists of frames, which are individual photos. Similarly, a ROS bag is a collection of IMU data and images that can be played back. ROS, the Robot Operating System, allows tools like OpenVINS to read and process this data, and then visualize it using a tool like RViz. For example, in Figure 10, we used an existing ROS bag called "neighborhood01.bag" that contains car data from the Ironsides dataset.

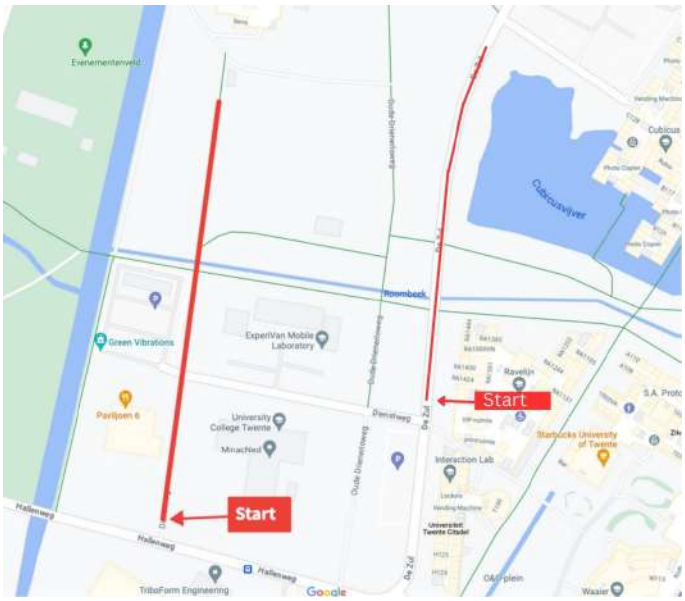

Fig. 9. Straight trajectory

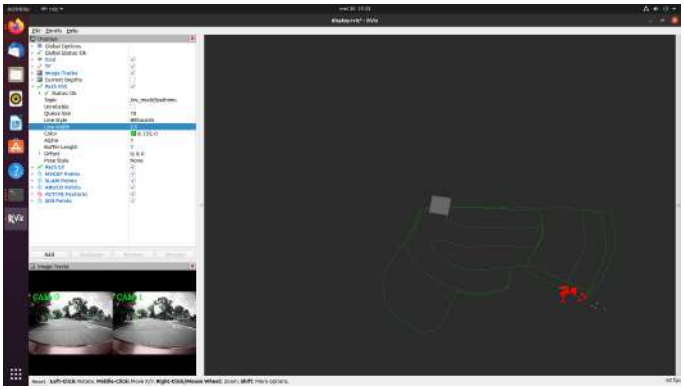

Fig. 10. neighborhood01.bag in OpenVINS

4.5 GNSS

We utilized the longitude and latitude coordinates from the GNSS to reconstruct the path taken during the experiment. The resulting path is visualized in Figures 11, 12, and 13. These figures depict the actual route we followed, which is further illustrated in Figure 8. The red dots represent the first collected dataset, and the blue dots represents the second dataset. By comparing the predicted trajectory with this ground-truth path, we can evaluate the accuracy of the trajectory prediction.

4.6 Simulation

Unfortunately, we encountered an unsolvable issue with our cost-efficient setup. The problem lies in the camera we are using, which either captures images at a high rate but compresses them, or it operates at a slow speed without compression. This poses a major

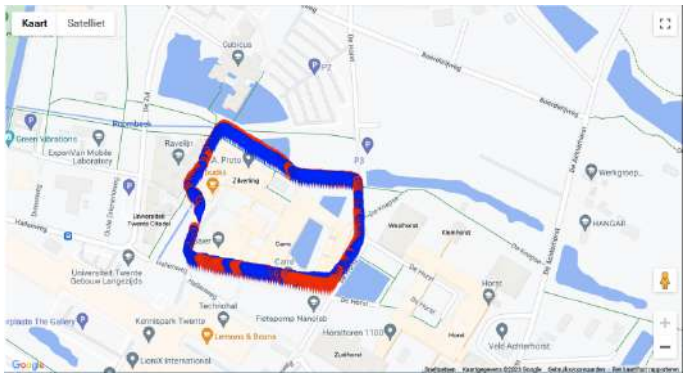

Fig. 11. GNSS locations from the first dataset

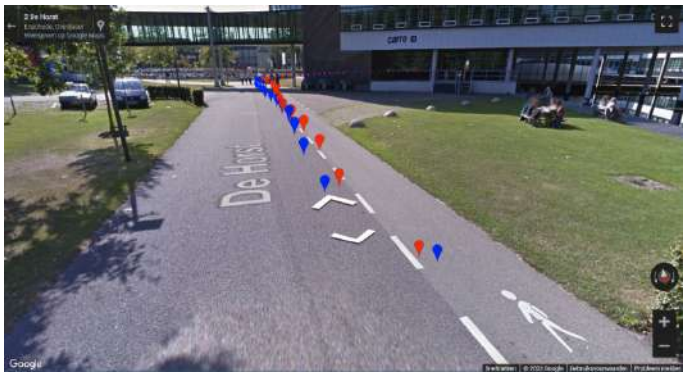

Fig. 12. GNSS locations from the first dataset

Fig. 13. GNSS locations from the first dataset

challenge for OpenVINS, as it requires the original, uncompressed images to function properly. The built-in compression functionality of OpenCV makes it faster than the Picamera2 library, but the latter only operates at a maximum rate of 1-2Hz, which is insufficient for accurate trajectory prediction. In order for OpenVINS to work, it relies on a calculation that involves the size of the images, represented by the equation:

$$step = width * bytedepth * numchannels$$

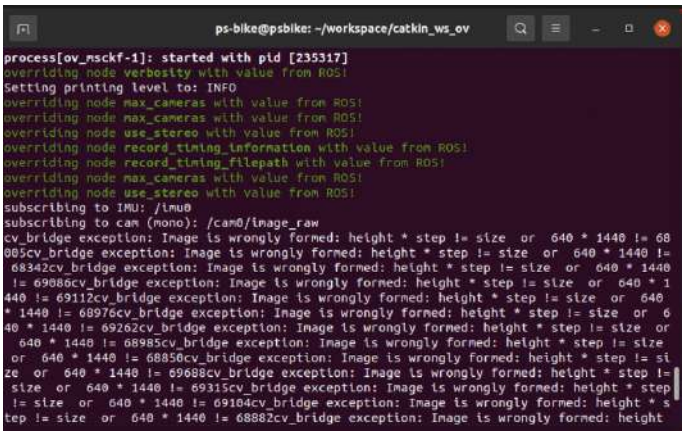

Fig. 14. Error output from OpenVINS

This equation needs to be satisfied:

$$height * step == filesize$$

However, since the images have been compressed, they no longer meet this requirement, resulting in the error message shown in Figure 14. To resolve this issue, it is essential to ensure that the images are not compressed and maintain their original size. Additionally, all images should have the same file size. Due to this issue, we were not successful in obtaining an output for the predicted trajectory of the bicycle.

5 RESULTS

In this section, we present the results of our analysis, comparing the performance of the different algorithms for trajectory prediction. Figures 15 and 16 show the comparison table and performance table, respectively, sourced from Huang et al. (2022) [5].

Figure 15 provides a comprehensive comparison between the physics models and data models based on the root-mean-square error (RMSE). The table highlights that the Kalman Filter is on par with the LSTM in short-term predictions.

Figure 16 presents the performance table of the different types of algorithms. This table, obtained from the results, demonstrates the prediction horizon and the computational cost for the algorithms. The physics based models overall are better for short prediction horizons with small computational cost.

These findings strongly support the choice of the MSCKF over other algorithms for bicycle trajectory prediction. The MSCKF demonstrates on par performance in terms of RMSE, which confirms its ability to provide more accurate predictions of bicycle trajectories. Bicycles have very limited capabilities in terms of computing power, making MSCKF the most suitable for a bicycle.

Overall, these results validate the selection of the MSCKF as a superior algorithm for bicycle trajectory prediction, showcasing its potential to enhance navigation systems, collision avoidance, and other applications requiring accurate and real-time bicycle trajectory estimation. However, the actual performance of the MSCKF algorithm on a bicycle dataset has to be evaluated to confirm its effectiveness in bicycle trajectory prediction. It is important to note

TABLE VII
COMPARISON FOR TRAJECTORY PREDICTION METHODS FOR AVS BASED ON THE HIGHWAY DRIVING DATASET NGSIM

Classification	Models	RMSE(m)				
		1S	2S	3S	4S	5S
Single Trajectory methods	Constant Velocity [102]	0.73	1.78	3.13	4.78	6.68
Kalman Filtering methods	IMM-KF [32]	0.58	1.36	2.28	3.37	4.55
HMM	C-VGMM+VIM [55], [152]	0.66	1.56	2.75	4.24	5.99
RNN	M-LSTM [82]	0.58	1.26	2.12	3.24	4.66
RNN	MFP-1 [84]	0.54	1.16	1.90	2.78	3.83
CNN and RNN	CS-LSTM(M) [102]	0.62	1.29	2.13	3.20	4.52
Attention Mechanism	MHA-LSTM [117]	0.41	1.01	1.74	2.67	3.83
GNN	GRIP++ [128]	0.38	0.89	1.45	**2.14**	**2.94**
GNN	GISNet [135]	**0.33**	**0.83**	**1.42**	2.14	3.23
Generative Model	MATF-GAN [152]	0.66	1.34	2.08	2.97	4.13
Generative Model	TS-GAN [153]	0.60	1.24	1.95	2.78	3.72
IRL	L-IRL [164], [202]	1.12	2.29	2.31	3.38	4.45
GAIL	GAIL-GRU [164], [182]	0.69	1.51	2.55	3.65	4.71
DIRL	MEDIRL [164], [187]	1.35	2.57	2.83	3.69	4.88
DIRL	DN-IRL [164], [203]	0.54	1.02	1.91	2.43	3.76

Fig. 15. Comparison table from [5]

TABLE IX
THE PERFORMANCE OF THE TRAJECTORY PREDICTION METHODS

Methods	Accuracy	Prediction Horizon	Computation Cost	Applications
Physics-based	High in short-term prediction, low in other prediction horizon	Short	Small	Collision risk analysis
Classic Machine Learning-based	Good at recognizing maneuvers but generalization ability is poor	Medium	Medium	Maneuver recognition
Deep Learning-based	High in considering some factors	Long	Relatively high	More and more applied in real-world
Reinforcement Learning-based	Relatively high, prediction methods are relatively few	Long	High	More applied in planning

Fig. 16. Performance table from [5]

that the characteristics of a bicycle are different from those of an autonomous vehicle, which are used in trajectory prediction algorithm evaluations. A bicycle is more susceptible to centimeter-level disturbances, rider-induced movements, and external factors such as road conditions and wind.

To assess the performance of the MSCKF on a bicycle dataset, specific evaluation metrics and methodologies need to be employed. These metrics should include the Root Mean Square Error (RMSE), so it can be compared to the performance of other algorithms. Additionally, the dataset used for evaluation should be representative of real-world bicycle trajectories, accounting for factors such as varying speeds, turning maneuvers, and environmental conditions.

Conducting experiments and comparisons against other trajectory prediction algorithms on dedicated bicycle datasets will provide valuable insights into the accuracy and reliability of the MSCKF for bicycle trajectory prediction. This evaluation will help establish the algorithm's suitability in real-world bicycle navigation scenarios, accounting for the unique characteristics and challenges of bicycle motion.

Note: The figures referenced above are sourced from Huang et al. (2022) [5].

6 CONCLUSION

In conclusion, our analysis compared the performance of various trajectory prediction algorithms for bicycle trajectory estimation. The results, obtained from the comparison table and performance table sourced from Huang et al. (2022) [5], highlighted the suitability of the Multi-State Constraint Kalman Filter (MSCKF) for bicycle trajectory prediction.

The comparison table showed that the MSCKF performed on par with the Long Short-Term Memory (LSTM) algorithm in terms of root-mean-square error (RMSE) for short-term predictions. This

indicates that the MSCKF is just as capable of providing accurate predictions for bicycle trajectories.

Furthermore, the performance table demonstrated that physics-based models, including the MSCKF, are recommended when it comes to short prediction horizons and computational cost compared to data models. This is particularly important for bicycles, which have limited computing power, making the MSCKF a very suitable choice for bicycle trajectory prediction.

However, it is essential to evaluate the performance of the MSCKF algorithm on a dedicated bicycle dataset to account for the unique characteristics and challenges associated with bicycle motion. This evaluation should include specific metrics such as RMSE and consider factors like varying speeds, turning maneuvers, and environmental conditions.

In summary, based on the available results and the understanding of the specific requirements and limitations of bicycle trajectory prediction, the MSCKF emerges as a promising algorithm for accurate and real-time estimation of bicycle trajectories. Further evaluation on dedicated bicycle datasets will be needed to validate its effectiveness and enhancing its applicability in real-world bicycle navigation systems, collision avoidance, and other related applications.

REFERENCES

[1] Samer Ammoun and Fawzi Nashashibi. 2009. Real time trajectory prediction for collision risk estimation between vehicles. In *2009 IEEE 5th International Conference on Intelligent Computer Communication and Processing*. 417–422. https://doi.org/10.1109/ICCP.2009.5284727

[2] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long Short-Term Memory. *Neural Computation* 9, 8 (1997), 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735

[3] Siavash Hosseinyalamdary. 2018. Deep Kalman Filter: Simultaneous Multi-Sensor Integration and Modelling; A GNSS/IMU Case Study. *Sensors* 18, 5 (May 2018), 1316. https://doi.org/10.3390/s18051316 Number: 5 Publisher: Multidisciplinary Digital Publishing Institute.

[4] Guoquan Huang. 2019. Visual-Inertial Navigation: A Concise Review. In *2019 International Conference on Robotics and Automation (ICRA)*. 9572–9582. https://doi.org/10.1109/ICRA.2019.8793604 ISSN: 2577-087X.

[5] Yanjun Huang, Jiatong Du, Ziru Yang, Zewei Zhou, Lin Zhang, and Hong Chen. 2022. A Survey on Trajectory-Prediction Methods for Autonomous Driving. *IEEE Transactions on Intelligent Vehicles* 7, 3 (Sept. 2022), 652–674. https://doi.org/10.1109/TIV.2022.3167103 Conference Name: IEEE Transactions on Intelligent Vehicles.

[6] Stéphanie Lefèvre, Dizan Vasquez, and Christian Laugier. 2014. A survey on motion prediction and risk assessment for intelligent vehicles. *ROBOMECH Journal* 1, 1 (July 2014), 1. https://doi.org/10.1186/s40648-014-0001-z

[7] Caroline Bianca Santos Tancredi Molina, Jorge Rady de Almeida, Lúcio F. Vismari, Rodrigo Ignacio R. González, Jamil K. Naufal, and João Camargo. 2017. Assuring Fully Autonomous Vehicles Safety by Design: The Autonomous Vehicle Control (AVC) Module Strategy. In *2017 47th Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshops (DSN-W)*. 16–21. https://doi.org/10.1109/DSN-W.2017.14 ISSN: 2325-6664.

[8] Anastasios I. Mourikis and Stergios I. Roumeliotis. 2007. A Multi-State Constraint Kalman Filter for Vision-aided Inertial Navigation. In *Proceedings 2007 IEEE International Conference on Robotics and Automation*. 3565–3572. https://doi.org/10.1109/ROBOT.2007.364024 ISSN: 1050-4729.

[9] Janosch Nikolic, Michael Burri, Igor Gilitschenski, Juan Nieto, and Roland Siegwart. 2016. Non-Parametric Extrinsic and Intrinsic Calibration of Visual-Inertial Sensor Systems. *IEEE Sensors Journal* 16, 13 (July 2016), 5433–5443. https://doi.org/10.1109/JSEN.2016.2556662 Conference Name: IEEE Sensors Journal.

[10] ROS. [n. d.]. Bags - ROS Wiki. http://wiki.ros.org/Bags

[11] Tessa van der Heiden, Naveen Shankar Nagaraja, Christian Weiss, and Efstratios Gavves. 2019. SafeCritic: Collision-Aware Trajectory Prediction. http://arxiv.org/abs/1910.06673 arXiv:1910.06673 [cs, stat].

[12] E.A. Wan and R. Van Der Merwe. 2000. The unscented Kalman filter for nonlinear estimation. In *Proceedings of the IEEE 2000 Adaptive Systems for Signal Processing, Communications, and Control Symposium (Cat. No.00EX373)*. 153–158. https://doi.org/10.1109/ASSPCC.2000.882463

[13] Yijing Wang, Zhengxuan Liu, Zhiqiang Zuo, Zheng Li, Li Wang, and Xiaoyuan Luo. 2019. Trajectory Planning and Safety Assessment of Autonomous Vehicles Based on Motion Prediction and Model Predictive Control. *IEEE Transactions on Vehicular Technology* 68, 9 (Sept. 2019), 8546–8556. https://doi.org/10.1109/TVT.2019.2930684 Conference Name: IEEE Transactions on Vehicular Technology.

[14] Greg Welch. 1997. An Introduction to the Kalman Filter. (1997).

[15] Guotao Xie, Hongbo Gao, Lijun Qian, Bin Huang, Keqiang Li, and Jianqiang Wang. 2018. Vehicle Trajectory Prediction by Integrating Physics- and Maneuver-Based Approaches Using Interactive Multiple Models. *IEEE Transactions on Industrial Electronics* 65, 7 (July 2018), 5999–6008. https://doi.org/10.1109/TIE.2017.2782236 Conference Name: IEEE Transactions on Industrial Electronics.

[16] Stefan Zernetsch, Sascha Kohnen, Michael Goldhammer, Konrad Doll, and Bernhard Sick. 2016. Trajectory prediction of cyclists using a physical model and an artificial neural network. In *2016 IEEE Intelligent Vehicles Symposium (IV)*. 833–838. https://doi.org/10.1109/IVS.2016.7535484