2018 IEEE Intelligent Vehicles Symposium (IV)
Changshu, Suzhou, China, June 26-30, 2018

Road Infrastructure Indicators for Trajectory Prediction*

Geetank Raipuria[1,2], Floris Gaisser[1] and Pieter P. Jonker[2]

***Abstract*— Safe and comfortable path planning in a dynamic urban environment is essential to an autonomous vehicle. This requires the future trajectories of all other road users in the environment of the vehicle. These trajectories are predicted through modeling the motion and behaviour of these road users. In this work we state that for efficient trajectory prediction only motion indicators are not sufficient. Therefore, we propose using a curvilinear coordinate system with curvature as road infrastructure indicators to improve motion modeling and trajectory prediction. With experiments, we show that the curvilinear coordinate system with curvature sufficiently incorporates the road structure. Furthermore, we show that a sequence-to-sequence RNN model is suitable to incorporate road curvature indicators directly into the modeling and prediction.**

I. INTRODUCTION

A key aspect in autonomous driving is path planning through a dynamic environment. This environment, such urban roads contain also other human driven vehicles. For safe and comfortable driving, it is essential for an autonomous vehicle to ensure timely detection of possible collisions, while avoiding false collision warnings.

Such path planning can be done by incorporating the future trajectories of the vehicle driving in the vicinity of an autonomous vehicle. The trajectories of these vehicles are unknown and need to be estimated. Human drivers do this intuitively by considering various indicators such as past motion, road structure, turn or braking lights etc. These indicators can also be obtained by the autonomous vehicle and should be considered in the prediction of the future trajectory. However, in the state of the art methods for trajectory prediction, mostly only the past motion is considered in prediction [9]. In a few methods these indicators are used to estimate the intention of the road user and adjust the model for trajectory prediction accordingly [13], [14].

Therefore, our aim in this work is to integrate such indicators in the process of trajectory prediction itself. First, Section II discusses the state of the art methods on indicators and modeling required for trajectory prediction. Next, our proposed method on infrastructure indicators is introduced in Section III. Furthermore, our adaptation of the sequence-to-sequence RNN for trajectory prediction is detailed. Through experiments in Section IV the applicability of our proposed methods are evaluated for an autonomous vehicle. Finally, the results are discussed and a conclusion is given in Section V and VI.

*This work was supported by the Interreg IAT project

[1]Faculty of Mechanical, Maritime and Material Engineering, Department of Intelligent Vehicles - Cognitive Robotics Delft University of Technology, 2628CD Delft, The Netherlands f.gaisser@tudelft.nl

[2]Robot Engineering Systems, P.O. Box 96967, 2509 JJ The Hague, The Netherlands

II. RELATED WORK

To predict the future trajectory of other road users, generaly a model is created from the available relevant past information. Therefore, this section first discusses various methods to model the state of the object. Second, the relevant data required for the model is discussed.

A. Modeling

Modeling the state of a road user from continuous information is generally refered to as tracking. An extensive amount of research exists on this topic. Motion models are essential to the task of tracking [2] and trajectory prediction [17]. This section first discusses work on tracking methods and next how these can be used to predict the trajectory of objects.

Modeling the motion of other road users often uses multiple noisy and partial observations of the latent state. By expressing the motion as a linear transformation with added Gaussian noise, this can be modeled as a linear dynamic system. A Kalman Filter (KF) [2] or extensions such as an extended KF (EKF) [17] or Unscented KF (UKF) [20] can be used to model these linear dynamic systems.

Generally, the motion observed from road users can be modeled with a linear dynamic system, though not with only a single model. A pedestrian walking along the road has a constant speed and therefore the model assumes the acceleration to have no effect (Constant Speed model). However a pedestrian standing still has no speed, thus the model would require a different linear representation where the speed also has no effect (Constant Position model). Therefore, a tracking method that can model different behaviour of road users should contain different types of models and switch between them [2], [17].

Such a change in motion is often instantiated by the intention of the road user or other environmental causes. However, these are difficult to directly observe. One approach is to switch between models by fitting all models and determining the best fit or mixture of models such as in IMMs [2]. Another approach is by estimating the intention or modeling the switching directly by Bayesian filters [13], [17]. This allows tracking and motion modeling of objects with changing behaviors.

Three types of methods for trajectory prediction are described in [14]; A *Physics* method is solely based on motion properties such as one of the motion models described above. The *Maneuvre* method predicts a trajectory using a motion model selected by the intention of the object. The *Interaction* method also includes the influence of other road users.

For maneuvre methods the intention and the resulting behavior are estimated, such as *following road* or *changing lane* [14], [8], [6]. Each of these maneuvres have a corresponding motion model that can be used to predict the trajectory of the object. However, an alternative model can only be selected with additional information indicating a different intention.

Many maneuvres include another road user such as *following vehicle* or *overtaking vehicle*. Therefore, [14] described the interaction method as one of the trajectory prediction methods. This interaction allows to select different motion models based on the interaction. For example the speed of the preceding vehicle limits the speed of the tracked vehicle. However, the speed of this vehicle is unknown for future moments in time. Thus the current speed of the vehicle is generally used for trajectory prediction and any change in speed of the preceding vehicle is not incorporated.

Therefore, solely switching of a motion model is not sufficient to incorporate intention and interaction of vehicles. We state that inputs which influence the motion model should be directly incorporated into the model. In the example of following a preceding vehicle, the speed of the preceding vehicle should be incorporated into the linear dynamics of the motion model. However, modeling external influence as part of the linear dynamic systems is in many cases very difficult or impractical.

Furthermore, consider a vehicle simply following the road. With a constant velocity motion model the predicted trajectory is a straight line. While for a constant turning rate or acceleration model the trajectory is making a curve (Figure 1). However, the vehicle is not only driving in a straight line or only making a (single) turn. The road consists of various straight and curved parts, and thus influencing the vehicles direction and limiting the position. Therefore, in this work we argue that all variations in a vehicles motion are caused by interaction with various aspects of the environment. Furtunately, the environment contains indicators that shows how the environment is influencing the motion. These indicators will be discussed in the next section.

B. Indicators

In this section we discuss different types of information that can indicate how the future trajectory of an object is influenced. Therefore, we categorize this data into different type of indicators:

- *motion indicators* discribe the kinematics of the vehicle, directly used in motion models
- *object indicators* is information displayed by the object.
- *infrastructure indicators* is how the road, traffic signs etc. influence the object.
- *interaction indicators* is how objects influence each other.

The most used and important indicator is that of motion information. This information is described by position, velocity etc. at multiple instances and form the past trajectory of the vehicle. This trajectory is the direct result of the intention of the object. Therefore, it is used by many motion model methods to extract the intention of the object. In turn the intention can be used to select a corresponding motion model that can be used to predict the future trajectory [18], [5].

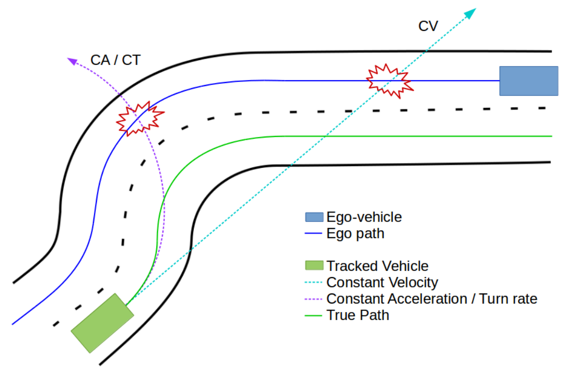


Fig. 1. Trajectory prediction using motion model on a curved road section

However, a change in motion is generally the result of a change in manoeuvre. In most cases, knowledge about an intented change of manoeuvre is prefered before it is being executed. Humans can predict a change in manoeuvre quite easily, because they also use many other indicators. Therefore, current research is focussed at including other information in the intention extraction [11], [3], [12], [15] .

As described before objects influence each other. For example when entering the highway [3] or following another vehicle [14]. A vehicle changes its trajectory based on the interaction with the other object. A vehicle on the highway may slow down or accelerate to create space to allow another vehicle to enter the highway. Accordingly the vehicle that wants to enter the highway will adjust its trajectory.

To support this interaction between objects, many road users indicate their intention before hand to others. For example a vehicle intending to change the lane is by law required to use the *turn indicator light* before hand. Also pedestrians often indicate their intention, though more indirectly. In [12], [4] the body pose and head orientation is used to estimate the persons orientation and intention whether the person is going to cross the road.

Also the structure of the road and other infrastructure components such as traffic lights and signs influence the trajectory of road users. In [15] the structure of intersections is used to extract the intention of a cyclist and select a corresponding motion model for trajectory prediction.

Selecting a specific motion model for every intention will require many different motion models, while they only differ in minor aspects. Therefore, we propose that the motion model should be extented to integrate the specific differences into the model, such as the velocity of the preceding vehicle or the curvature of the road.

To achieve this we limit current work to predicting the trajectory of cars following the road. These roads can have various shapes, though have no intersections. Furthermore, we propose a method to effectively include the shape of the road as an integral part of the modeling and prediction. Also, this approach should be easily extendable with other type of indicators.

III. Method

The contributions of this work consist of three parts; First, describing the motion information as a function of the road shape in order to integrate the road structure. Second, extract the curvature of the road for better trajectory prediction in curved sections. Last, we propose a sequence-to-sequence RNN to model vehicles and predict their trajectory with multiple indicators.

A. Curvilinear Coordinate System

In [8] the longitudinal and lateral position as well as velocity of the vehicle with respect to the road is extracted for tracking. To achieve this, a non-linear coordinate system is defined as a function of the shape of the road. This section will describe how to obtain this coordinate system and model the motion of a vehicle as a function of the road shape.

Figure 2 shows a curved road section, with $[X^G, Y^G]$ in the *Global Cartesian Coordinate System* (GCCS) and $[X^C, Y^C]$ in the *Curvilinear Coordinate System* (CCS). The road geometry is defined as a piecewise qubic spline as defined in Equation 1. Where X^G and Y^G is the position in GCCS, s the parametric variable in the range of $[0...k]$ and $a_x, b_x, c_x, d_x, a_y, b_y, c_y, d_y$ constants of the spline.

$$X^G = a_x * s^3 + b_x * s^2 + c_x * s + d_x \quad (1)$$
$$Y^G = a_y * s^3 + b_y * s^2 + c_y * s + d_y$$

The position $[X^C, Y^C]$ of the vehicle in CCS is defined by the projection of the position to point c_p on the s-axis and the lateral distance n_p as shown in Figure 2. To find c_p the function f in Equation 2 is minimized with a non-linear optimization. Where s is the unknown parameter and $a_x, b_x, c_x, d_x, a_y, b_y, c_y, d_y$ are the constants of the spline.

$$f = (a_x * s^3 + b_x * s^2 + c_x * s + d_x - X^G)^2 + (a_y * s^3 + b_y * s^2 + c_y * s + d_y - Y^G)^2 \quad (2)$$

Note that s is the parametric variable of the spline and is not the distance along the spline to point c_p. To obtain this distance l_p Equation 3 can be used.

$$l = \int_0^{s=s_p} dl \cdot ds = \int_0^{s=s_p} \sqrt{\left(\frac{dx}{ds}\right)^2 + \left(\frac{dy}{ds}\right)^2} ds \quad (3)$$

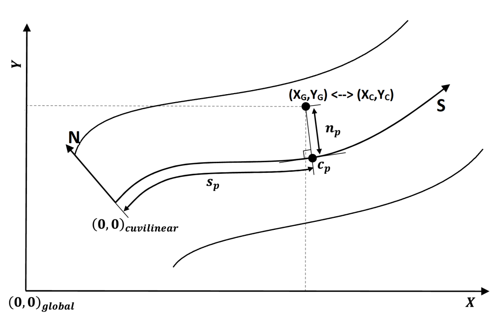


Fig. 2. The Curvilinear Coordinate system, image adopted from [8]

Fig. 3. Vehicle trajectory plotted on map in GPS coordinates

In contrast to [8] a CCS is defined for each direction of the road, such that the driving direction is always positive and the curvature of the inner and outer curves match the road structure. For more details on how the Curvilinear Coordinate System is used we refer to the work [16].

Generally, it can be assumed that vehicles follow the road and don't go off-road intentionally. Therefore, we propose to use the CCS to model the motion of the vehicle as a function of the road structure. To illustrate the benefit of this, Figure 1 shows a vehicle driving on a road section. This section of road first makes a slight left turn and then a sharper right turn. Both constant velocity and acceleration models show that the vehicle will go off road, but also intersect with the path of the ego-vehicle. However, the actual trajectory of the vehicle will follow is that of the road and can be modeled with the CCS.

The effect of modeling the motion in CCS is illustrated with a real world example of a vehicle taking a left curve (Figure 3). The motion in CCS as shown in Figure 4 is similar to how humans think, more left or right of the center of the lane and further down the road. From the described position within the lane in CCS it is also clear that the driver cut the corner. This is can also be noticed in Figure 3 when the lane is drawn.

Also for the velocity there is a large difference: Before and after the turn the X and Y-velocity change significantly in GCCS as shown in Figure **??**. However, in CCS the velocity is related to the motion along the road. The S-velocity is the speed along the road and the N-velociy is lateral to the road. When making a perfect turn the S-velocity is the true speed shown on the speedometer. Also, the N-velocity describes the change of position within the lane. Again, the cutting of the corner is observable from the lateral N-velocity in Figure 5.

B. Curvature

The motion in CCS is described as longitudinal and lateral movement along the road. Unfortunately, this also does not provide information of a change in direction of the road, thus information about turns. Since, drivers generally reduce the velocity because there is a curve, an additional feature is required that reintroduces this relavant information of the road.

Therefore, we define a feature describing the curvature of the road obtained by taking the change in the direction with respect to the curve length as defined in Equation 4 [21].

$$k = \frac{\dot{x}\ddot{y} - \dot{y}\ddot{x}}{\left(\dot{x}^2 + \dot{y}^2\right)^{(3/2)}} \tag{4}$$

$$\text{with } \dot{x} = \frac{dx}{ds}, \dot{y} = \frac{dy}{ds}, \ddot{x} = \frac{d^2x}{ds^2}, \ddot{y} = \frac{d^2y}{ds^2}$$

Generally, a vehicle slows down before the curve, due to safety and control of the vehicle, which can be observed from Figure 6. However, information about the curvate at the position of the vehicle is not informative, because a driver slows down before the curve and speeds up in towards the end of the curve. Furthemore, since a fast driving vehicle has to look further ahead as it has to slow down more than a slow driving vehicle. Therefore, the curvature indicator is defined as the curvature 2 seconds (50 time steps) ahead of the vehicle. By modeling the slowing-down behaviour as a dynamic system, this look-ahead-time can be estimated more accurately, though more information about the driving style of the vehicle is also required. Therefore, the value of 2 seconds was approximated by observing the behaviour of the recorded vehicles.

C. Sequence-to-Sequence model

In [19] a sequence-to-sequence RNN model is used to encode a sentence in one language and decode it in a different language. We propose to use the encoder part of this type of network to model the state of the vehicle and use the decoder part for prediction of the vehicles future trajectory. In this section we describe how this network is adapted for vehicle trajectory modeling and prediction with motion and infrastructure indicators.

Figure 7 shows our network design, with Long Short Term Memory (LSTM) units [7] as RNN units. The encoder part is fed with the past information, in the form of position and velocity in CCS x_{t-n} as well as the curvature k_{t-n}. The LSTM units don't output any information, and only pass forward the hidden state to the next time step.

The decoder part is used for trajectory prediction. Every LSTM unit is fed with the vehicles position and velocity of the current time step $\widetilde{y}_{t+n}$ combined with the additional indicators such as the curvature k_{t+n}. For training $\widetilde{y}_{t+n}$ is known as x_{t+n}, though for deployment this is not known. As the output of an unit is the predicted state at the next time step and can be used as the input of the next unit. Therefore, we pass the predicted state $\widetilde{y}_{t+n-1}$ to the next unit in the sequence-to-sequence RNN.

However, this causes discrepancy in training and inference, which leads to poor performance [1]. A scheduled sampling during training can be used, where x_{t+n} is selected with a probability η or alternatively $\widetilde{y}_{t+n-1}$. At the start of training $\eta = 1$, selecting the training data. As training progresses η is reduced such that the network is trained with the same

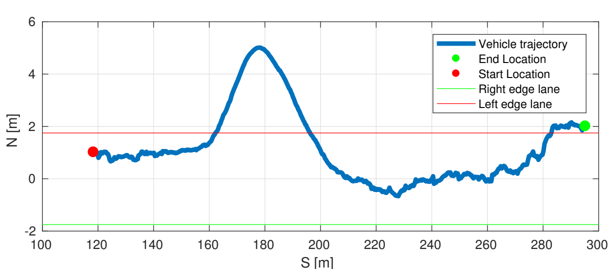


Fig. 4. Vehicle trajectory in Curvilinear Coordinate System corresponding to trajectory in 3

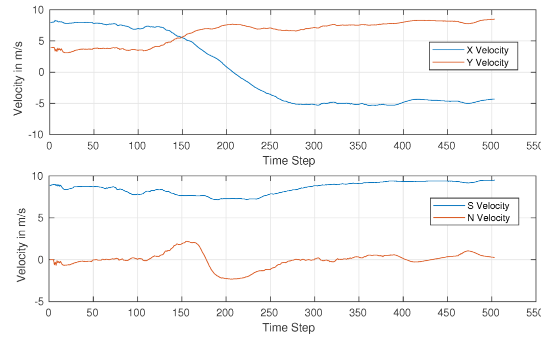


Fig. 5. Velocity in GCCS (top) and CCS (bottom), corresponding to trajectory in 3

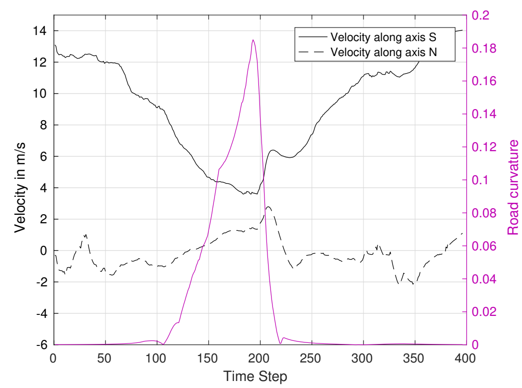


Fig. 6. Effect of Road Curvature on vehicle velocity

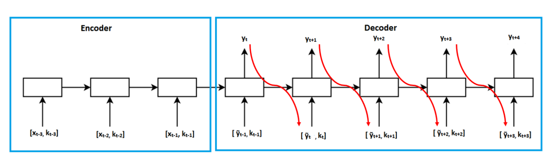


Fig. 7. Sequence to Sequence RNN model

settings as during inference.

In contrast to machine translation, for trajectory prediction, the inputs and outputs are a continuous sequence of trajectory data. As a result we do not reverse the order of the input data. Also we do not use the special end-of-sentence symbol in our model, as there is no specific end of the sequence.

IV. Experiments

The contributions of this work consists of two parts; First, the use of road structure and curvature as infrastructure indicators in trajectory prediction. Second, a novel modeling approach for trajectory prediction using these infrastructure indicators along with motion indicators. With these contributions we aim to improve trajectory prediction on curved road sections. We perform experiments to evaluate our proposed approach against a conventional motion based prediction method on real data.

For our experiments trajectories of human driven vehicles under natural driving condition were recorded along with road infrastructure information. The data was collected with a test vehicle, the WEpod, in the region of Wageningen, The Netherlands. The test vehicle is equipped with 6 IBEO LUX LIDAR sensors running at 25Hz. Each LIDAR has a 110 degree (horizontal) FOV and 4 vertical planes. Data from these LIDAR sensors is used to detect and record vehicles moving around the test vehicle.

The IBEO system provides vehicle position $[X, Y]$, velocity $[\dot{X}, \dot{Y}]$ and heading angle θ in the WEpods (ego-vehicle) reference frame. The vehicle states were converted to the Universal Transverse Mercator (UTM) coordinate system which acts as the GCCS, using GPS localization.

Figure 8 shows the road sections on which the vehicle trajectories were recorded. These were selected to record vehicle trajectories on roads of different curvature, while avoiding the influence of features like pedestrian crossings, complex road design like roundabouts and intersections with traffic lights.

We obtained 285 unique vehicle trajectories, with a minimum lenght of 7 seconds containing 175 timesteps. The trajectory data was segregated with a 4:1 ratio into a training and test set. For trajectories longer than 10 seconds, the trajectory was split into multiple parts at an interval of 3 seconds. This resulted in 496 training trajectories and 143 trajectories used for testing.

Additionally, an artificial dataset was created to pre-train the RNN model. This dataset consists of simulated vehicle trajectories with different driving behaviors in CCS along the longitudinal direction. We simulated vehicle trajectories on straight roads including constant velocity and de/acceleration followed by constant velocity. Vehicle motion on curved roads was simulated considering different radii. Based on the curve radius the maximum safe vehicle speed was calculated, and vehicles were simulated to decelerate to a speed below this value. Gaussian noise was added to all states, with mean and variance obtained from LIDAR measurements. The dataset consists of 1/5th simulated trajectories on road with no curvature, and the rest randomly generated for curves of different road curvatures. Training the network with the transfer learning regime gave an 8% improvement in performance and was subsequently used in all experiments.

To compare the trajectory prediction in CCS with GCCS a baseline Interactive Multiple Model filter [2] with constant velocity and acceleration models was used. To compare the performance of the RNN and the curvature indicator, we trained two RNN models, one with only motion features as input and another with both motion features as well as road curvature.

We determined experimentally the best LSTM network architecture, by varying the number of hidden layers between and the number of cells in each layer. This resulted in an architecture of two hidden layers of 275 and 160 cells. The output layer uses basic RNN cells with Rectified Linear Unit (ReLu) nonlinearity. Furthermore, we used Adams [10] optimization algorithm to train the network. The loss is calculated using as Mean Squared Error over the four output states $(X, Y, \dot{X}, \dot{Y})$. The network is trained with a constant learning rate of 10^{-3}.

A. Experiment 1: Sequence-To-Sequence model

This experiment is designed to establish that, the RNN model can perform regression in a non-linear space and model an internal state over multiple samples. Sine-wave prediction is chosen as the regression task. The space is single dimensional, and is described by the function $x =$

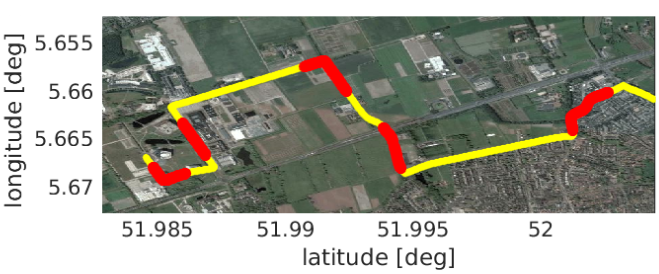


Fig. 8. Recorded road sections (in red)

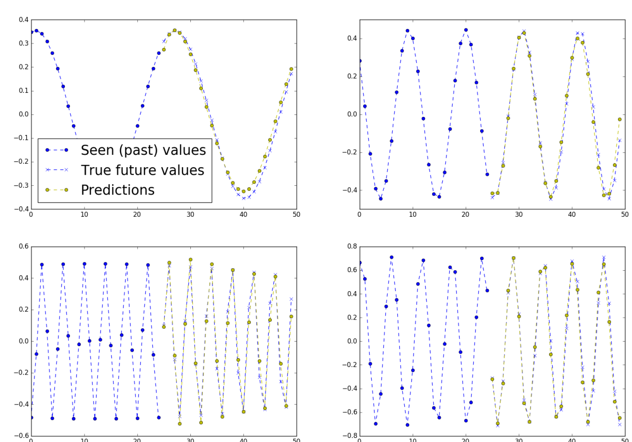
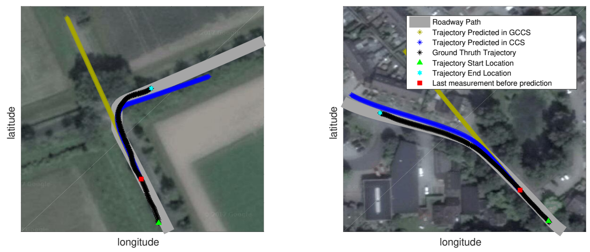


Fig. 9. Sine-wave prediction with Sequence-to-Sequence RNN



(a) (b)

Fig. 10. Vehicle trajectory predicted using IMM in GCCS and CCS

$a * sin(2\pi f t + \phi)$. Where a is the amplitude, f the frequency, ϕ the phase and t the independent variable.

The training dataset consists of randomly generated sine-waves with random amplitude, frequency and phase. The first 25 samples of the wave are provided as input to the model, which then predicts the next 25 samples. To make this prediction, the three variables a, f, ϕ need to be modeled internally by the encoder and the value x predicted.

For sin-wave prediction, the RNN model is found to perform best with one hidden layer of 40 LSTM cells. The neural network is trained using Adams optimization [10], with Mean Squared Error loss. Figure 9 shows some predicted sin wave from test set samples.

B. Experiment 2: Curvilinear Coordinate System

To compare trajectory predicted in CCS with one predicted in GCCS, we plot them both in GPS co-ordinates along with the ground truth trajectory. The IMM prediction model is provided with 2 seconds (50 time steps) of vehicle states as input data and predicts the trajectory for next 8 seconds (200 time steps). Figures 10 shows the position plots of two (real) examples of vehicle trajectories predicted in GCCS and CCS. In the first the trajectory, Figure 10(a), the road makes a sharp 90 degree turn, and in the second figure 10(b) the roadway makes a turn of about 30 degrees.

C. Experiment 3: Curvature Indicator

To make the performance comparison between the three models, IMM, Motion RNN, and Curvature RNN, the models are provided with input data for 25 time steps (1 second), and the error is reported for the predicted vehicle position at 25, 50, 100 and 150 time steps in meters. Error in velocity would be reflected in position, as a result the velocity predictions are not used as a separate metrics to compare the three models.

We segregate the test trajectories into three groups based on the curvature. Figure 11 shows the performance of the three models for each type of test trajectory. Table I gives the mean error over all trajectories for the three models.

V. Discussion

In this work the Curvilinear Coordinate System was used to incorporate the structure of the road into modeling vehicle motion. Experiment 2 aimed to show that CCS improves the trajectory prediction over GCCS. Since in CCS the motion

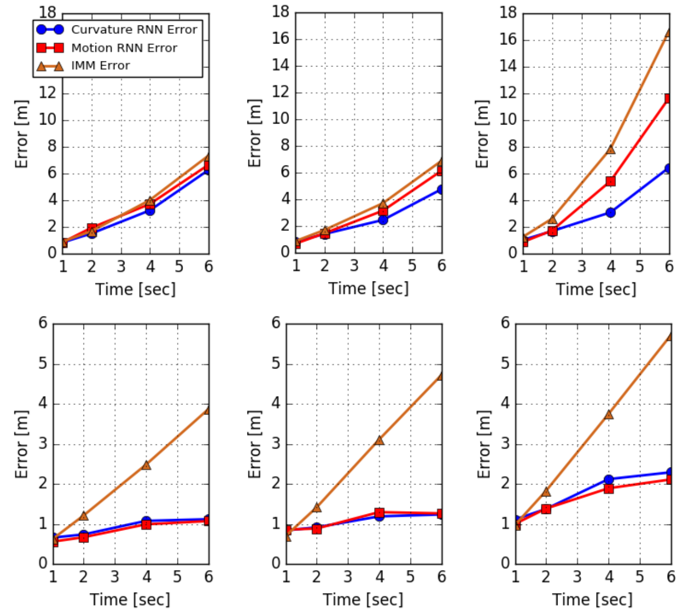


Fig. 11. Prediction Error (in m) on the X (top) and Y (bottom) axis for test trajectories over prediction time. Left: trajectories with almost no curvature; Center: slight curvature; Right: large curvature.

is a function of the road structure, modeling this information is directly integrated. This was then also clearly showed with two results, in Figure 10, where trajectory prediction in GCCS has no knowledge about the curve and predicted a straight line.

Since information about a change in direction of the road is not provided by CCS, the velocity prediction does not correspond to that of normal driving behaviour. Generally a vehicle slows down before a curve. Therefore, an indicator was introduced in the form of curvature. Experiment 3 shows with Figure 11 that for sharp curves the longitudinal prediction error is much reduced using road curvature. For 4 seconds prediction the mean error is 3 meters, which is less than the length of a car and can be used for path planning. For longer periods the error is increasing as for 6 seconds prediction the mean error is 6 meters. This can be explained by the fact that for longer periods of time more unmodeled factors influence the vehicle and its trajectory.

It is interesting to note that for the lateral position the curvature provides no improvement in prediction, though this was to be expected. The lateral position within the lane is generally not a result of a curve, but of other factors. However, the RNN seems to learn that vehicles stay within

	IMM		**Motion RNN**		**Curvature RNN**	
axis:	X	Y	X	Y	X	Y
1 sec	0.95	0.70	**0.82**	**0.73**	0.87	0.80
2 sec	1.90	1.39	1.78	**0.86**	**1.55**	0.90
4 sec	4.87	2.91	3.98	**1.24**	**3.19**	1.30
6 sec	9.47	4.48	7.61	**1.33**	**6.20**	1.38

TABLE I

Mean error [m] for all test trajectories their lane. Which was one of the goals of this work, but was not explicitly defined in training the RNN.

In order to model the motion and predict the trajectory of the vehicle, while directly integrating additional indicators such as curvature, we introduced the sequence-to-sequence RNN. This RNN was used for language translation, but hasn't been used in trajectory prediction. Therefore, we devised a toy-example in Experiment 1 with prediction sine-wave patterns. A simple LSMT with only a few cell could already model and predict these non-linear patterns. In Experiment 3 the RNN model was shown to clearly outperform IMM in trajectory prediction especially as it was able to model some level of driver behaviour such as keeping its lane.

VI. Future Work and Conclusion

From the predicted trajectories we noticed that the position and velocity outputs did not correspond. Integrating the velocities over time gave a different position than the position output itself. This is because the RNN model has no understanding of the laws of physics and does not adhere to the kinematic rules of motion. Therefore, in future work this constraint will be added to the loss such that $v_t = (x_t - x_{t-1})/dt$.

One of the aims of this work was to find a modeling method that could be used to incorporated additional indicators in the motion modeling and trajectory prediction. We proposed the sequence-to-sequence RNN and showed with the curvature indicator this was attainable. When studying the results some unmodeled factors became appearant, which we intent to include as indicators in future work.

Some recorded trajectories had to be excluded as a vehicle or bicycle was preciding the vehicle and limiting the velocity of the tracked vehicle. By modeling this interaction with an indicator describing the speed of a preceding road user, this can included in the motion model.

When observing the predicted velocities of vehicles we noticed that most vehicles would drive about 40 km/h, though some roads had a maximum speed of 30 km/h and others had 50 km/h. Therefore, we also intend to include an maximum speed indicator, such that the trajectory prediction will reflect normal driving behaviour on various type of roads.

One of the limitations we set was that only road sections without intersections were used in this work. An intersection adds multiple possible paths which can be modeled by adding a node in the piece-wice spline where multiple splines branch off. Also, intersections have traffic rules to regulate the interaction between road users, which also have to be modeled. For example traffic lights and preference rules have to be included. Furthermore, an approach has to be found that indicates which of the choosen trajectories is most probable. Fortunately, vehicles often indicate that already with their turn-light indicator. Therefore, also object indicators such as turn and brake-light should be incorporated.

This work aimed at improving the accuracy of vehicle trajectory prediction over a longer duration of time in complex urban settings. Therefore, a Sequence-to-Sequence RNN was introduced to model the motion of a vehicle combined with road structure and curvature. Through experiments on real world data we have shown that trajectory prediction is significantly improved. Finally, we have set a road map of features that have to be included to succesfully implement this on a vehicle.

References

[1] S. Bengio, O. Vinyals, N. Jaitly, and N. Shazeer. Scheduled sampling for sequence prediction with recurrent neural networks. In *Advances in Neural Information Processing Systems*, pages 1171–1179, 2015.

[2] S. Blackman and R. Popoli. Design and analysis of modern tracking systems(book). *Norwood, MA: Artech House, 1999.*, 1999.

[3] C. Dong, J. M. Dolan, and B. Litkouhi. Intention estimation for ramp merging control in autonomous driving. In *Intelligent Vehicles Symposium (IV), 2017 IEEE*, pages 1584–1589. IEEE, 2017.

[4] F. Flohr, M. Dumitru-Guzu, J. F. Kooij, and D. M. Gavrila. A probabilistic framework for joint pedestrian head and body orientation estimation. *IEEE Transactions on Intelligent Transportation Systems*, 16(4):1872–1882, 2015.

[5] E. Galceran, A. G. Cunningham, R. M. Eustice, and E. Olson. Multipolicy decision-making for autonomous driving via changepoint-based behavior prediction: Theory and experiment. *Autonomous Robots*, pages 1–16, 2017.

[6] T. Gindele, S. Brechtel, and R. Dillmann. A probabilistic model for estimating driver behaviors and vehicle trajectories in traffic environments. In *Intelligent Transportation Systems (ITSC), 2010 13th International IEEE Conference on*, pages 1625–1631. IEEE, 2010.

[7] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[8] K. Jo, M. Lee, J. Kim, and M. Sunwoo. Tracking and behavior reasoning of moving vehicles based on roadway geometry constraints. *IEEE Transactions on Intelligent Transportation Systems*, 18(2):460–476, 2017.

[9] B. Kim, C. M. Kang, S. H. Lee, H. Chae, J. Kim, C. C. Chung, and J. W. Choi. Probabilistic vehicle trajectory prediction over occupancy grid map via recurrent neural network. 2017.

[10] D. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[11] S. Klingelschmitt, M. Platho, H.-M. Groß, V. Willert, and J. Eggert. Combining behavior and situation information for reliably estimating multiple intentions. In *Intelligent Vehicles Symposium Proceedings, 2014 IEEE*, pages 388–393. IEEE, 2014.

[12] J. F. P. Kooij, N. Schneider, F. Flohr, and D. M. Gavrila. Context-based pedestrian path prediction. In *European Conference on Computer Vision*, pages 618–633. Springer, 2014.

[13] C. Laugier, I. E. Paromtchik, M. Perrollaz, M. Yong, J.-D. Yoder, C. Tay, K. Mekhnacha, and A. Nègre. Probabilistic analysis of dynamic scenes and collision risks assessment to improve driving safety. volume 3, pages 4–19. IEEE, 2011.

[14] S. Lefèvre, D. Vasquez, and C. Laugier. A survey on motion prediction and risk assessment for intelligent vehicles. *Robomech Journal*, 1(1):1, 2014.

[15] E. A. Pool, J. F. Kooij, and D. M. Gavrila. Using road topology to improve cyclist path prediction. In *Intelligent Vehicles Symposium (IV), 2017 IEEE*, pages 289–296. IEEE, 2017.

[16] G. Raipuria. Vehicle trajectory prediction using road structure. Master's thesis, Delft University of Technology - Mechanical Engineering, 12 2017.

[17] N. Schneider and D. M. Gavrila. Pedestrian path prediction with recursive bayesian filters: A comparative study. In *German Conference on Pattern Recognition*, pages 174–183. Springer, 2013.

[18] T. Streubel and K. H. Hoffmann. Prediction of driver intended path at intersections. In *Intelligent Vehicles Symposium Proceedings, 2014 IEEE*, pages 134–139. IEEE, 2014.

[19] I. Sutskever, O. Vinyals, and Q. V. Le. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112, 2014.

[20] J. Tao and R. Klette. Tracking of 2d or 3d irregular movement by a family of unscented kalman filters. *Journal of information and communication convergence engineering*, 10(3):307–314, 2012.

[21] E. W. Weisstein. "Curvature" from mathworld–a wolfram web resource.