2018 26th Mediterranean Conference on Control and Automation (MED)
Zadar, Croatia, June 19-22, 2018

Infrastructure-based Tracking of Road Users in Urban Intersections for Partially Available, Variable-Delay Sensor Data*

Tobias Quack, Frank-Josef Heßeler and Dirk Abel[1]

Abstract— **Reliable and accurate tracking of road users is crucial for many applications in intelligent transport systems. Especially in urban intersections, where traffic safety, optimal traffic efficiency and the operation of autonomous vehicles are still technically challenging, the requirements on environment perception systems are very high. In this paper we address the problem of an infrastructure-based multiple object tracking system which makes use of the sensor information provided by a combination of both infrastructure and mobile sensors. Due to the characteristics of wireless V2X communication used to transfer the sensor data, the signals are unsynchronized, have variable delay times and are only partially available. Our sensor fusion approach, which is specifically designed to handle these characteristics, is validated using simulated sensor data and experimentally tested in an urban test environment.**

I. INTRODUCTION

For future mobility concepts, automated driving of road vehicles is currently regarded as one of the most essential components. Replacing the human driver by automated systems has the potential to improve traffic safety, traffic efficiency, comfort, flexibility and accessibility of transportation systems. In addition, connected traffic infrastructure, smart sensors and mobile devices carried by the traffic participants can play an important role in future traffic solutions by providing additional information that would otherwise not be available to the different agents. This information is particularly valuable in urban scenarios where many different road users with conflicting intentions share a limited space. In these surroundings, automation requires a detailed understanding of the traffic situation including the positions as well as the past and estimated future trajectories of all traffic participants.

In our work, we therefore focus on urban intersections where we consider the problem of tracking road users based on a network of multiple connected sensors. The network uses wireless data transfer for some sensor systems so that sensor information regularly arrives with a considerable delay which varies depending on the wireless connection quality. The goal of our work therefore consists in optimally fusing unsynchronized sensor information in order to track road users in an urban intersection. Out-of-sequence measurements, i.e. measurements that arrive at the processing center in a nonchronological sequence with respect to their time of validity should be included explicitly. We validate our algorithm using both simulated sensor data and real-life data and prove that the algorithm is feasible for real-time execution in an experimental test scenario.

The problem of tracking an unknown number of objects based on multiple sensor signals has been studied intensively in literature. Many approaches deal with in-vehicle object detection based on vehicle environment sensors such as radar, camera and LiDAR. For this problem, methods based on random-finite sets, namely probability hypothesis density (PHD) filtering, are very popular due to their ability to handle the uncertainties in both the number of objects and the states of the objects [1]. Data association, i.e. the decision, by which of multiple objects a certain measurement was caused, is then incorporated in a stochastic way. Studies which use variants of PHD filtering for solving multiple object tracking problems can be found in [2], [3] and [4], among others. Apart from PHD filtering approaches, the in-vehicle object tracking problem has been solved with probabilistic data association (PDA) filtering techniques [5] or rule-based strategies [6]. However, for in-vehicle applications, the sensor data can generally be assumed to arrive reliably at a fixed sample rate. In addition, the time delay between the validity of a measurement and its processing by the fusion computer is usually constant and small compared to the sample rate of the algorithm.

Similarly to the focus of our paper, [7] consider an infrastructure-based tracking system at an urban intersection. However, the authors do not exploit delayed information from mobile agents and assume synchronized measurements with matching sample rates from multiple camera and LiDAR sensors. A comprehensive review of alternative methods for multi sensor data fusion including Kalman filtering or Monte Carlo algorithms such as particle filters is given in [8].

Many algorithms for the explicit correction of time-delayed measurements in sensor fusion problems have been proposed for Kalman filters and Bayes filters in general. [9] introduce multiple methods to compensate for delayed measurements in a Kalman filter and analyze the respective computational burden. The multi-rate problem i.e. the presence of multiple sensors operating with different sample times is studied in [10] and [11] with the latter combining multi-rate with delayed measurements in a scheme consisting of two linked Kalman filters. [12] include delayed camera measurements in an extended Kalman filter for position and orientation estimation of mobile robots. However, none of the above mentioned studies consider the data association problem which results from tracking multiple objects.

The contribution of our work is the design of a real-

*This work was supported by the German Federal Ministry of Education and Research (BMBF).

[1]Tobias Quack, Frank-Josef Heßeler and Dirk Abel are with Institute of Automatic Control, RWTH Aachen University, 52074 Aachen, Germany `t.quack@irt.rwth-aachen.de`

978-1-5386-7891-6/18/$31.00 ©2018 IEEE

time feasible sensor fusion algorithm for infrastructure-based tracking of multiple objects in an urban environment capable of processing delayed, multi-rate sensor data from arbitrary sensor systems.

The remainder of this paper is structured as follows: Section II provides detailed information about the sensor and communication setup considered in this work including the specific characteristics of the resulting multiple object tracking problem. In Section III, the algorithms we propose for our centralized sensor fusion system are introduced and explained. The validation of our approach using a simulation environment as well as experimental results obtained at the test intersection is presented and discussed in Section IV and a conclusion and outlook to our work are given in Section V.

II. PROBLEM STATEMENT

The scenario we consider in our research is traffic in a large urban intersection which is fitted with intelligent network and sensor infrastructure. In order to provide infrastructure-based services to the mobile agents, a central intersection computer (CIC) is located at the intersection which runs algorithms for the detection and tracking of road users as well as the prediction of future trajectories. Communication between the CIC and the mobile agents is realized wirelessly, so that considerable time delays for the transfer of sensor data have to be taken into account. Due to the characteristics of our scenario, the following requirements to the sensor fusion system were identified: The algorithm needs to cope with an **unknown and varying number of objects** and perform **data association** of the measurements to the tracked states. The properties of the different sensor systems impose a **multi-rate problem** with **out-of-sequence measurements** including potentially **high delay times**. In addition, the system should take into account **object classifications** provided by different sensor systems and apply suitable prediction principles so that **multiple models** for pedestrians and vehicles are required. The desired update rate of the sensor fusion results is $50\,\mathrm{Hz}$.

The aforementioned connected intersection concept has been physically implemented on the RWTH Aachen University proving ground "Aldenhoven Testing Center" [13], [14]. Details on the proving ground and the sensor and communication hardware at the test intersection are given in the following sections II-A and II-B.

A. Test Environment

The test intersection used for the experimental validation of our centralized sensor fusion approach is a 4-way intersection with a varied layout of turning lanes, bicycle lanes and pedestrian crossings. An aerial image of the intersection is given in Figure 1.

For vehicle-to-infrastructure (V2I) communication, the 802.11p (ITS G5) technology is employed at the test site. Smartphones and mobile GNSS sensors carried by pedestrians are included using conventional 802.11b/g/n/ac WiFi and LTE 5G cellular networks. As a common framework for data acquisition and processing on the CIC, we use the robotics middleware "Robot Operating System" (ROS). By doing so, our software framework is structured in a modular way so that nodes for additional sensors can easily be integrated and standard interfaces are available for data exchange between different nodes. Our multiple object tracking algorithm is realized as a ROS node and written in C++. For a live visualization of all sensor data together with the fusion results, a graphical front end based on the ROS visualization tool RViz has been developed.

Fig. 1. Aerial view of the test intersection at Aldenhoven Testing Center

On the test site, positions are measured with respect to a common, Cartesian coordinate frame which follows the ENU (east north up) convention for the orientation of the coordinate axes. The coordinate origin is located on a reference point of the test intersection which also serves for converting WGS84 coordinates obtained from GNSS receivers to the local frame. For details on the required coordinate conversions, refer to [15].

B. Sensor Setup

The sensor setup we consider for the localization of traffic participants comprises a variety of different sensor systems some of which are directly connected to the CIC whereas others are carried by mobile agents and therefore use wireless data transfer to communicate the respective information to the central sensor fusion. Infrastructure-based sensors include a system of four IBEO Lux3 **LiDARs**, multiple **cameras** and an **ultrasonic** sensor system all of which are connected using Gigabit-Ethernet. Pedestrians can connect their **smartphones** to the CIC so that GPS localization data is provided wirelessly. For research purposes, highly precise mobile **GNSS receivers** are available. Vehicles with V2X communication capabilities send their **ego localization** as well as data from their **environment perception sensors**, e.g. LiDAR and camera.

III. MULTIPLE OBJECT TRACKING

As a solution to the specific multiple object tracking problem introduced in Section II, we implement a two stage

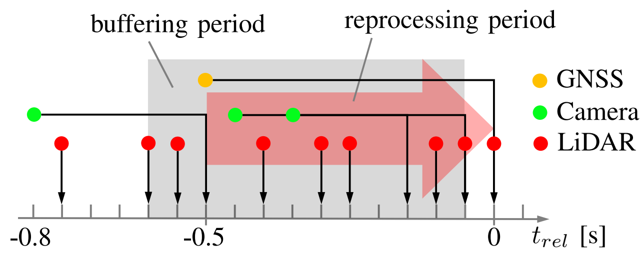

Fig. 2. Reprocessing principle for delayed measurements

fusion scheme which separates the data association and the filtering step. Each sensor reading is first mapped to the most likely existing or to a newly registered tracked object. Subsequently, the new position of each tracked object is calculated based on all associated measurements as well as the respective movement model using a multi-model discrete-time extended Kalman filter (EKF). Both processing stages take into account the time delay of each measurement explicitly. This structure of the tracking algorithm allows for the introduction of advanced heuristics which improve the tracking performance by making use of additional information such as self-identifying measurements which include the ID of the related object or knowledge about the fields of view of different sensors.

The following sections provide details about our approach for correcting time delays, the data association method and the filtering step.

A. Out-of-Sequence Measurements

The method we employ for handling the out-of-sequence measurements is based on buffering past states, past measurements and the respective covariance matrices of our discrete-time algorithm for a certain time interval. In principle, this allows us to access the state of the filter at an earlier time step at which a newly arrived, delayed measurement was taken. The measurement can then be processed at that matching time step before the filter prediction and update steps are recalculated up to the current time step. This basic method known as “reprocessing” is well established for single object tracking problems and Kalman filtering, see e.g. [9]. In the multiple object tracking case, the buffered states can also improve the data association results. By accessing the buffer, a delayed measurement can be compared to the positions of all tracked objects at the matching past time step in order to find the most likely association.

Due to the discrete-time processing scheme of our algorithm, the matching past state x_{k-n} is calculated from the constant sample time of the algorithm t_S, the delay time of the measurement t_{delay} and the current time step k by rounding to the nearest step with

$$n = \text{round}\left(\frac{t_{delay}}{t_S}\right). \tag{1}$$

The major advantage of the reprocessing principle is its optimality with respect to the available information. One possible drawback is related to the computational burden. The number of reprocessing steps required in each time step of the filtering algorithm increases with increasing delay times of the measurements. In our application, the maximum allowed delay was defined as $600\,\text{ms}$. Figure 2 visualizes the reprocessing principle at the example of an object being tracked by three sensors with different delay times. At the current time step with $t_{rel} = 0$, an undelayed LiDAR measurement and a GNSS measurement, which was taken at $t_{rel} = -0.5\,\text{s}$, were received. The algorithm selects the oldest received measurement, in this case the GNSS reading, retrieves the respective old states and covariance matrices from the buffer and starts reprocessing all time steps up to $t_{rel} = 0$. In this particular case, this involves the calculation of 10 filter steps with 9 measurement corrections, while a conventional algorithm would only calculate 1 filter step with the 2 new measurements. In Section IV-A the resulting implications on computation time are analyzed.

B. Data Association

The purpose of the data association stage is to determine whether a sensor reading can either be used to correct the position of an object which is already tracked, whether it should be discarded since it was caused by a sensor error or if it was caused by a new object which should be registered for subsequent tracking. Other association hypotheses like split or merge situations (see [16]) can be handled implicitly.

In our implementation, we use the Mahalanobis distance to determine the probability that a measurement belongs to a tracked object. In order to decide whether a given measurement should be used to correct a given state, the respective Mahalanobis distance D_M is calculated and compared to a threshold Ξ_M which is one major parameter of the data association algorithm. If $D_M < \Xi_M$ the accordance between measurement and state is considered to be sufficient so that data association is possible.

The Mahalanobis distance D_M between a position measurement $\vec{z}$ with $\vec{z} = [z_x, z_y]^T$ and a tracked object with positions $\vec{x} = [p_x, p_y]^T$ is calculated as follows:

$$D_M = \sqrt{(\vec{z} - \vec{x})^T \cdot Q \cdot (\vec{z} - \vec{x})} \tag{2}$$

For measurements containing additional coordinates like velocities or orientations, the Mahalanobis distance can be calculated by extending the measurement vector and the state vector accordingly. In equation 2, Q is a covariance matrix that contains the uncertainties related to the association of $\vec{z}$ and $\vec{x}$. For this matrix, two choices can be considered: If the covariance of the measurements is selected, the probability of associating a measurement to a given state increases with the uncertainty of the measurement. If the covariance of the state is selected, measurements are more likely associated with this state if the uncertainty of the state increases. Since both variants represent desirable mechanisms, we calculate the Mahalanobis distances for both the measurement covariance matrix and the state covariance matrix and choose D_M as:

$$D_M = \min\left(D_{M,meas}, D_{M,state}\right) \tag{3}$$

This choice guarantees that uncertain measurements can be assigned to well tracked states while precise measurements can still be associated with uncertain states.

For measurements that cannot be associated with any object tracked by the filter, new objects are created. However, in order to avoid ghost objects caused by measurement errors, each new object is considered as "unconfirmed" and not published by the filter algorithm until a minimum number of measurements has been associated with it within a time limit. If the confirmation criteria is met, the object status is changed to "confirmed".

Objects are removed from tracking if one of the following conditions is fulfilled: A timeout parameter defines the maximum time after which a new measurement must be associated with each object. If the timeout is exceeded, the respective object is deleted. An additional uncertainty threshold defines the maximum position uncertainty tolerated for a tracked object. Large uncertainties result in small Mahalanobis distances so that wrong associations would become more likely for objects with large covariance matrices.

Each tracked object is assigned a unique ID so that the trajectory traveled by each road user can be determined easily. For these IDs, two spaces are distinguished: "Local" IDs are assigned by the data association algorithm and only valid for one occurrence of the respective road user at the intersection. "Global" IDs are sent by the road users themselves, e.g. included in V2I messages [17] or transmitted by the portable GNSS units. For the tracking algorithm, these IDs are very valuable since they eliminate association uncertainties. Tracking of objects with "global" IDs is therefore rarely lost.

C. Filtering

For each tracked object, the respective states x are updated in each time step using a discrete-time Kalman filter [18]. The basic Kalman filter uses a prediction step which calculates an a-priori estimate $\widehat{x}_i$ for the state x at the considered time step i based on the state x_{i-1} and the respective control input u_{i-1}. The second main step of the Kalman filtering process known as "correction" corrects $\widehat{x}_i$ based on measurements z_i in order to obtain the a-posteriori estimate x_i.

One major design choice for the implementation of a Kalman filter for object tracking is the selection of the states to be tracked by the filter and the definition of a suitable model for the prediction of these states. This model should incorporate the basic principles of movement of the respective object. In our implementation, we use different models depending on the object type. The vehicle model consists of the 5 states $x = \left[p_x, p_y, \phi, \dot{\phi}, v\right]$ where p_x and p_y describe the vehicle's position, ϕ is the direction of movement, $\dot{\phi}$ the yaw rate around the vertical axis and v the absolute velocity. For the prediction, the following update functions with the sample time T are used:

$$p_{x,k} = p_{x,k-1} + v_{k-1}T \cdot \cos\left(\phi_{k-1} + 0.5 \cdot \dot{\phi}_{k-1}T\right) \quad (4)$$

$$p_{y,k} = p_{y,k-1} + v_{k-1}T \cdot \sin\left(\phi_{k-1} + 0.5 \cdot \dot{\phi}_{k-1}T\right) \quad (5)$$

$$\phi_k = \phi_{k-1} + \dot{\phi}_{k-1}T \quad (6)$$

$$\dot{\phi}_k = \dot{\phi}_{k-1} \quad (7)$$

$$v_k = v_{k-1}. \quad (8)$$

This model assumes a constant vehicle velocity and yaw rate and implies that the direction of movement changes with the yaw rate of the vehicle.

For pedestrians and unidentified road users, a more general, linear model is employed. Its 4 states are $x = [p_x, p_y, v_x, v_y]$ where v_x and v_y denote the components of the velocity vector in x and y-direction. The state transition matrix is

$$F = \begin{bmatrix} 1 & 0 & T & 0 \\ 0 & 1 & 0 & T \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (9)$$

In our application, the Kalman filter correction step is mostly trivial, since the object states are measured directly by the sensors. Standard deviations are identified for all sensor systems and assumed to be time invariant so that the respective values of the measurement covariance matrix R can be obtained easily. The measurement noise covariance matrix Q is tuned iteratively in simulations (see Section IV-A) in order to optimally balance smoothing vs. fast response of the different states.

IV. RESULTS

For the validation of our work, we conduct both simulation and real world experiments. Since our algorithm is implemented in C++ using the ROS framework, it is designed to subscribe to a custom message format which can be generated by our simulation environment as well as by the sensors at our test intersection. The identical algorithm could thus be validated in simulation before integrating it on the proving ground without the need to adapt any interfaces.

A. Simulation Results

In order to generate realistic simulation data for the validation of our algorithm, we implement a modular simulation environment. Table I shows an overview of the different sensor systems considered in this study together with sensor specific additional parameters. The error of the position measurements provided by each sensor is modeled as a normally distributed, Gaussian white noise so that a standard deviation σ_{xy} can be assumed. t_{sample} defines the fastest sample rate at which the sensor system generates new measurements, p_{det} is the probability of detection, i.e. the likelihood that an object located within the range of the sensor is actually detected. The delay t_{delay} measures the time between the validity of a measurement and its availability at the sensor fusion computer. It therefore includes delay due to data processing at the sensor stage as well as communication delay.

For the generation of simulated objects, vehicles and pedestrians were modeled, that appear and disappear on a

TABLE I

SENSOR PARAMETERS

Sensor	$t_{sample}[s]$	$t_{delay}[s]$	$\sigma_{xy}[m]$	$p_{det}[\%]$
Infrastructure LiDAR	0.01	0.001	0.015	22
Infrastructure Camera	0.5	0.5	0.5	80
Pedestrian GNSS	0.5	0.3	0.5	100
Vehicle GNSS	0.1	0.2	0.5	100

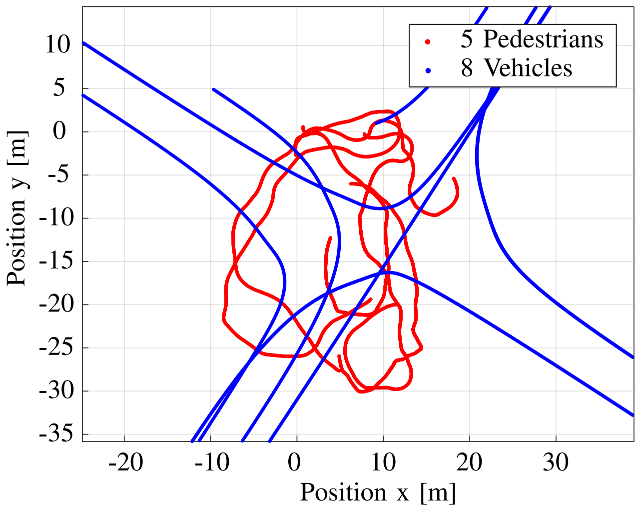

Fig. 3. Trajectories of the simulated road users in the intersection area

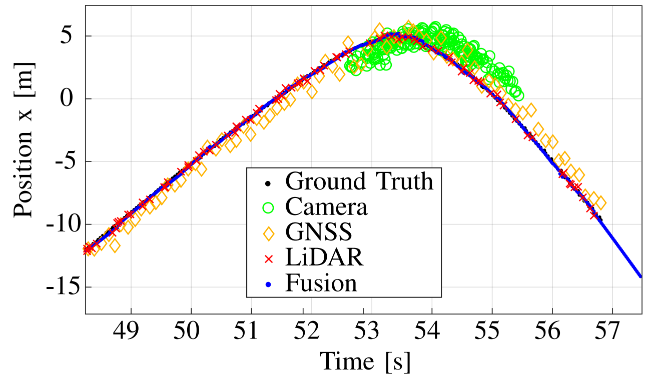

Fig. 4. Fusion position x vs. time for simulated vehicle with time delayed measurements

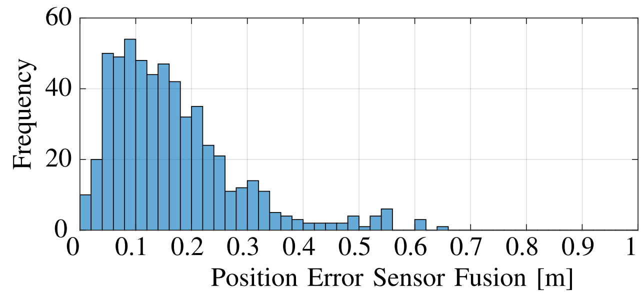

Fig. 5. Histogram of the fusion position error for the selected vehicle

stochastic basis. The trajectories of the simulated vehicles are given by the intersection layout, i.e. vehicles are able to perform all road legal maneuvers on the intersection. Target velocity and maneuver type are chosen randomly. Simulated pedestrians are programmed to move across the whole intersection area. Sudden changes of direction, standing still and accelerating are triggered randomly. Figure 3 shows the resulting trajectories traveled by 5 pedestrians and 8 vehicles in a $45\,\mathrm{s}$ time interval of our test scenario.

In Figure 4, the fusion result for the trajectory of one vehicle is depicted as a plot of the coordinate x vs. the simulation time t. The figure shows the ground truth as generated by the simulation environment, the fusion result and the different types of simulated sensor signals associated by the tracking algorithm. Measurements from the camera sensors are only available in a small part of the trajectory because the cameras' fields of view cover the central intersection area only. Since for the generation of the plot, the times t_{proc} of measurement processing at the CIC were used instead of the times t_{valid}, at which the measurements were taken, the delay of the different sensor systems can be deduced from the figure. Despite the delayed measurements, the fusion result tracks the real position of the vehicle very well. Systematic errors such as a lag are effectively avoided.

In order to analyze the accuracy of the tracking, we generate a histogram of the Euclidean errors which is shown in Figure 5. The fusion error never exceeds $0.65\,\mathrm{m}$, the mode is at $0.1\,\mathrm{m}$. For further analysis, we define the percentage of sensor fusion positions with Euclidean errors exceeding $0.5\,\mathrm{m}$ for cars as $e_{0.5,Car}$ and the respective percentage for pedestrians with Euclidean errors exceeding $0.3\,\mathrm{m}$ as $e_{0.3,Ped}$.

Table II shows performance criteria of the object tracking algorithm which have been determined as average values over all objects appearing during the whole simulation scenario. The identical scenario has been simulated for three cases: In the case "without t_{delay}", all sensor readings were made available to the sensor fusion immediately. This scenario can serve as a reference. Our approach for the fusion of delayed measurements was tested in the case "t_{delay} corrected". Here, the sensor readings were delayed as shown in table I. The result of ignoring the time delays in the tracking algorithm can be found in the case "t_{delay} ignored". In the table, p_A, p_B and p_C represent the percentages of three different cases for the data association results. "A"-measurements were associated with the correct object, "B"-measurements were associated with non existing, duplicate objects created by the fusion algorithm and "C"-measurements were associated with the wrong existing object.

The results show that the data association stage of our algorithm can compensate for the delayed measurements very effectively. The percentage of correct associations is practically identical with the value achieved for non delayed measurements. When the delay is ignored, the percentage of data association errors rises from about 1 % to over 5 %. In terms of the tracking accuracy, the results are similar. While

TABLE II

OBJECT TRACKING PERFORMANCE

	without t_{delay}	t_{delay} corrected	t_{delay} ignored
$p_A[\%]$	99.00	98.95	94.86
$p_B[\%]$	0.57	0.82	4.93
$p_C[\%]$	0.43	0.23	0.21
$e_{0.3,Ped}[\%]$	4.30	5.71	23.56
$e_{0.5,Car}[\%]$	5.24	6.70	41.31

TABLE III

TIMING BUDGET FOR PROCESSING STEPS

	t_{delay} corrected	t_{delay} ignored
Data Association [ms]	1.72 (33.2%)	1.72 (68.9%)
Prediction [ms]	2.37 (45.7%)	0.44 (17.7%)
Correction [ms]	1.04 (20.0%)	0.28 (11.3%)
Overhead [ms]	0.05 (1.1%)	0.06 (2.1%)
Total Step Time [ms]	5.19	2.50

in case of corrected sensor delays, 5.71 % of the sensor fusion positions for pedestrians exceed the error of $0.3\,\mathrm{m}$, this percentage increases to 23.56 % without correction. For vehicles, the correction is even more critical due to the higher average speeds which result in increased position errors due to lag.

In terms of the computational load, we were able to execute our algorithm in real time on a conventional desktop computer with an Intel i5-3470 CPU running at $3.2\,\mathrm{GHz}$. The target sample rate of $50\,\mathrm{Hz}$ could be maintained with a safe margin. Table III shows the amount of time required for the different processing stages in each time step. When comparing the times required without delay correction to the times of our final algorithm, the increased computational load due to the reprocessing steps is obvious. While the data association takes exactly the same amount of time in both cases, the prediction and correction times have increased significantly. The total processing time for one time step is approximately doubled by the delay correction. However, compared to the maximum sample time of $20\,\mathrm{ms}$ required for real-time execution, the safety factor is still about 4 in our implementation.

B. Experimental Results

For experimental validation, we test our algorithm at the urban intersection of "Aldenhoven Testing Center". During the test, 4 pedestrians and 2 vehicles were present at the test field. In terms of sensor systems, we had access to the infrastructure LiDAR, vehicle on-board data and multiple mobile GNSS receivers. The tracking result for one pedestrian is depicted in Figure 6. It can be seen that continuous tracking is achieved even based on the delayed GNSS measurements only. Where available, the LiDAR measurements improve the tracking quality. At $t = 19.3\,\mathrm{s}$, three GNSS measurements with different delay times arrive at the CIC simultaneously due to wireless communication outages. However, as the individual delay times are known to the tracking algorithm, the tracking result is not impaired.

In terms of the parametrization of the filter, the figure shows that the algorithm has been set up to smooth noise

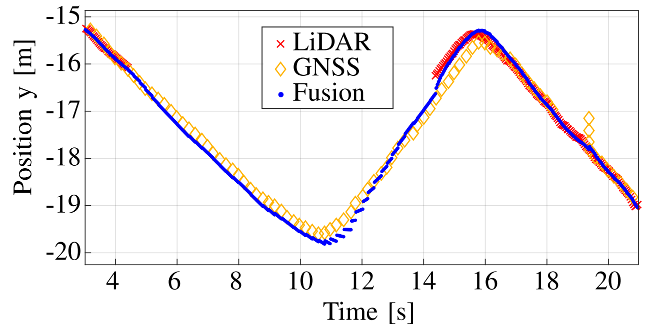

Fig. 6. Fusion position x vs. time for real pedestrian with time delayed measurements

from the measurements i.e. the model noises are assumed to be relatively low. At $t = 11\,\mathrm{s}$, the tracked pedestrian performs a sudden change of direction while being detected by GNSS only. At that time, the direction of movement predicted by the filter only changes slowly which results in the characteristic spikes. At the second change of direction at $t = 15.5\,\mathrm{s}$, LiDAR measurements are available, as well, so that the fusion follows the real trajectory more quickly.

The selected parametrization has been found to be particularly useful in the presence of noisy data from multiple sensors where it ensures smooth trajectories being calculated by the sensor fusion. In the subsequent processing steps – e.g. the prediction of future road user trajectories [19] – these fusion results can then be used seamlessly.

V. CONCLUSION

In our work, we have proposed an algorithm for tracking road users in an urban intersection based on multiple sensor systems. By explicitly correcting for time delayed measurements, we are able to include both infrastructure and wirelessly connected mobile sensor systems. Our algorithm is capable of tracking an unknown number of pedestrians and vehicles using specific movement models for each type. Tracking is based on measurements from multiple sensor systems with varying rates, delay times and availabilities. For the validation of our approach, we have presented results from a tailor made simulation environment and conducted an experimental study at a test intersection. Due to the explicit correction of time delays, the tracking performance was found to be excellent; the accuracy of our algorithm is close to the optimal case without time delays. In terms of the computational load, the feasibility of our method for real-time execution on a conventional desktop computer was proven. Future work includes further studies with an upgraded sensor setup and the extension of the algorithm for the exploitation of additional sources of information such as statistical data about road user behavior or digital maps.

ACKNOWLEDGMENT

This research was funded by the German Federal Ministry of Education and Research (BMBF) within the Project "I2EASE". The authors acknowledge the fruitful collaboration with the project partners.

REFERENCES

[1] S. Reuter, *Multi-object tracking using random finite sets*. Ulm: Schriftenreihe des Instituts für Mess-, Regel- und Mikrotechnik, 2014.

[2] M. Vasic, D. Mansolino, and A. Martinoli, "A system implementation and evaluation of a cooperative fusion and tracking algorithm based on a Gaussian Mixture PHD filter," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2016, pp. 4172–4179.

[3] J. Gan, M. Vasic, and A. Martinoli, "Cooperative multiple dynamic object tracking on moving vehicles based on Sequential Monte Carlo Probability Hypothesis Density filter," in *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, 2016, pp. 2163–2170.

[4] F. Zhang, C. Buckl, and A. Knoll, "Multiple Vehicle Cooperative Localization with Spatial Registration Based on a Probability Hypothesis Density Filter," *Sensors*, vol. 14, no. 1, pp. 995–1009, 2014.

[5] M. Maehlisch, W. Ritter, and K. Dietmayer, "De-cluttering with Integrated Probabilistic Data Association for Multisensor Multitarget ACC Vehicle Tracking," in *2007 IEEE Intelligent Vehicles Symposium*, 2007, pp. 178–183.

[6] S. Hwang, N. Kim, Y. Choi, S. Lee, and I. S. Kweon, "Fast multiple objects detection and tracking fusing color camera and 3D LIDAR for intelligent vehicles," in *2016 13th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI)*, 2016, pp. 234–239.

[7] D. Meissner, S. Reuter, E. Strigel, and K. Dietmayer, "Intersection-Based Road User Tracking Using a Classifying Multiple-Model PHD Filter," *IEEE Intelligent Transportation Systems Magazine*, vol. 6, no. 2, pp. 21–33, 2014.

[8] B. Khaleghi, A. Khamis, F. O. Karray, and S. N. Razavi, "Multisensor data fusion: A review of the state-of-the-art," *Information Fusion*, vol. 14, no. 1, pp. 28–44, 2013.

[9] T. D. Larsen, N. A. Andersen, O. Ravn, and N. K. Poulsen, "Incorporation of time delayed measurements in a discrete-time Kalman filter," in *37th IEEE Conference on Decision and Control*, 1998, pp. 3972–3977.

[10] D.-J. Lee and M. Tomizuka, "Multirate optimal state estimation with sensor fusion," in *Proceedings of the 2003 American Control Conference, 2003*, vol. 4, 2003, pp. 2887–2892 vol.4.

[11] A. Fatehi and B. Huang, "Kalman filtering approach to multi-rate information fusion in the presence of irregular sampling rate and variable measurement delay," *Journal of Process Control*, vol. 53, pp. 15–25, 2017.

[12] K. Berntorp, K. E. Årzén, and A. Robertsson, "Sensor fusion for motion estimation of mobile robots with compensation for out-of-sequence measurements," in *2011 11th International Conference on Control, Automation and Systems*, 2011, pp. 211–216.

[13] P. Themann, D. Raudszus, L. Eckstein, T. Kopacz, and D. Heberling, "Modulare Testumgebung zur Analyse kooperativer Assistenzsysteme basierend auf V2X-Technologien im kontrollierten Feld," in *Automotive meets Electronics, Beiträge der 6. GMM-Fachtagung*, vol. 1, 2015.

[14] T. Quack, F.-J. Heßeler, and D. Abel, "Zentrale Sensorfusion für kooperative Fahrerassistenzsysteme in städtischen Kreuzungen," in *Positionierung und Navigation für intelligente Verkehrssysteme POSNAV*, Berlin, 2016.

[15] J. Wendel, *Integrierte Navigationssysteme: Sensordatenfusion, GPS und inertiale Navigation*, 2nd ed. München: Oldenbourg, 2011.

[16] A. G. A. Perera, C. Srinivas, A. Hoogs, G. Brooksby, and W. Hu, "Multi-Object Tracking Through Simultaneous Long Occlusions and Split-Merge Conditions," in *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, vol. 1, 2006, pp. 666–673.

[17] ETSI, "Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Part 2: Cooperative Awareness Basic Service, European Telecommunications Standards Institute ETSI TS 102 637-2," 2011.

[18] D. Simon, *Optimal State Estimation: Kalman, H Infinity, and Nonlinear Approaches*. Wiley-Interscience, 2006.

[19] C. Framing, T. Quack, F.-J. Heßeler, and D. Abel, "Lokalisierung und Bewegungsprädiktion für vernetzte Fahrzeuge an städtischen Kreuzungen," in *8. VDI/VDE-Fachtagung AUTOREG*, Berlin, 2017.