Online Mobile User Speed Estimation: Performance and Tradeoff Considerations

Majed Haddad, Dalia-Georgiana Herculea, Chung Shue Chen, Eitan Altman, Veronique Capdevielle

Online Mobile User Speed Estimation: Performance and Tradeoff Considerations

Majed Haddad*, Dalia Georgiana Herculea†, Chung Shue Chen†, Eitan Altman‡ and Véronique Capdevielle§

*CERI/LIA, University of Avignon, Avignon, France
†Bell Labs, Nokia, France
‡INRIA Sophia-Antipolis, Sophia-Antipolis, France
§Mobile Networks, Nokia, France

Abstract—This paper presents an online algorithm for mobile user speed estimation in 3GPP Long Term Evolution (LTE)/LTE-Advanced (LTE-A) networks. The proposed method leverages on uplink (UL) sounding reference signal (SRS) power measurements performed at the base station, also known as eNodeB (eNB), and remains effective even under large sampling period. Extensive performance evaluation of the proposed algorithm is carried out using field traces from realistic environment. The online solution is proven highly efficient in terms of computational requirement, estimation delay, and accuracy. In particular, we show that the proposed algorithm can allow for the first speed estimation to be obtained after 10 seconds and with an average speed underestimation error of 14 kmph. After the first speed acquisition, subsequent speed estimations can be obtained much faster (e.g., each second) with limited implementation cost and still provide high accuracy.

Index Terms—Cellular networks, LTE, mobility state estimation, shadowing, memory consumption, complexity, estimation delay.

I. INTRODUCTION

Past few years have witnessed tremendous advancement in mobility management thanks to the new technical possibilities that agile network management offers. For instance, advanced mobility management implementations which monitor contextual situations have been recently investigated to support for unlimited seamless mobility across all networks and technologies [1]. Mobility robustness optimization (MRO) features were introduces by 3GPP to LTE self-optimization functions in order to address these issues and also dynamically improve the network performance of handover (HO), provide enhanced high quality of experience (QoE) for the users as well as increased network capacity. This can be done by automatically adapting cell HO thresholds based on feedback of performance indicators.

Mobility management is a challenging issue in heterogeneous network (HetNet) deployments or in macro multi-carrier deployments. In these contexts, the mobility state estimation (MSE) is crucial for optimizing HOs so as to reduce call drop and networking signaling flow, optimize traffic scheduling, and

achieve resource optimization. Fig. 1 illustrates a particularly interesting use case for UE mobility management. The main idea is to favor the handover of high mobility users, called user equipments (UEs), towards large coverage cells (e.g., macro cells in HetNet) or towards higher coverage carriers (e.g., LTE 700 MHz) in multi-carrier scenarios. On the other hand, for macro cell offloading purpose, the handover of low or medium mobility UEs to small cells is highly desirable. Besides, the optimal adjustment of HO parameters (e.g., offsets, hysteresis, timers, filtering coefficients, etc.) is mobility dependent.

Knowledge of UE mobility can also be very beneficial to transmission scheduling [2], mobility load balancing [3], channel quality indicator (CQI) feedback enhancement [4], energy efficiency [5], and many resource management scenarios. Choosing either frequency selective scheduling (FSS) or frequency diversity scheduling (FDS) for LTE, could depend on the user speed as FSS shall be preferred for low speed UEs and FDS for high speed UEs.

New mobility estimation enhancement is expected to provide high flexibility, accuracy and robustness that new 5G architectures would require. However, there are many issues to be addressed before the concept of MRO for next generation mobile networks (NGMN) can be maturely implemented and commercialized.

A. Existing Speed Estimation Techniques

Although there is a rich literature on methods for optimizing HO parameters based on MSE, enhancements to MSE procedures for practical mobile networks have been presented in only several papers. There are crossing based methods which counts the number of times that received signal strength crosses a certain level [6], covariance based methods that estimate the speed by computing the covariance function of the received signal [7], maximum likelihood based methods that rely on periodic channel estimation [8], and power spectrum based methods that estimate the Doppler frequency based on the characteristics of the power spectrum of the fading channel. As these algorithms aim at analyzing the speed dependent fast fading characteristics of the signals, they do not cope with the large periodicity of LTE sounding reference signal (SRS) measurements.

Authors of [9] propose a mobility estimation procedure that updates the reference signal received power (RSRP) thresholds
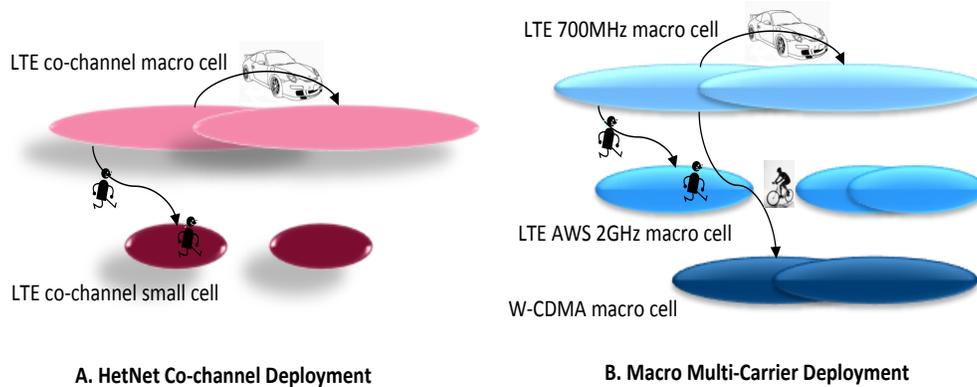
Fig. 1: Mobility state estimation scenarios.

for HO. The procedure is shown to have higher accuracy than the legacy procedure but has significant computational complexity. In [10], authors introduce region based approach to determine the speed of a vehicle using sensor data. [11] presents a method for estimating user mobility at the handover time. These methods estimate the speed of UEs in some specific use cases. In addition, some are not efficient for small observation windows, some are sensitive to noise or small Doppler spreads and do not consider implementation complexity, which is crucial to practical systems.

B. Contributions

In [12], we addressed the problem of determining UE speed and mobility class for mobility management in LTE radio access network (RAN). We proposed two methods for UE speed estimation: a spectral analysis method (SAM) and a time-based spectrum spreading method (TSSM). Both these methods belong to *network-based* solutions according to 3GPP-LTE standard, since they are based on UL SRS power measurements conducted at LTE base stations. Both solutions exploit the speed dependent time variations of the shadowing in the SRS measurements. The computed metrics are compared with a reference curve or look-up table (database), with respect to the shadowing decorrelation distance D (see Fig. 2).

In this paper, we will focus on time-based spectrum spreading method (TSSM) since it has very low implementation complexity and computational requirement, from a system deployment standpoint. One major objective is to investigate the online deployment of TSSM and its design so as to enable real-time application of the method to mobile cellular networks.

The main contributions of the paper can be summarized as follows:

1) *Online speed estimator:* a new online implementation of TSSM which allows for a continuous, real-time and customizable speed estimation (speed tracking). The solution captures the trade-off between the accuracy

and implementation cost, and is particularly suitable for applications where fast update of the speed estimation is crucial.

2) *Realistic simulation setup:* data traces that include path loss, multi-path, shadowing, and fractional power control, following 3GPP specifications, various mobility scenarios and practical environments.

3) *Extensive performance evaluation:* numerous tests and investigation of key performance metrics including the cumulative distribution function (CDF) of speed estimation error, probability of correct estimation, and speed under- and overestimation averages.

The paper is organized as follows. We first give a brief presentation of the TSSM method in Section II and describe the background. We introduce the new online algorithm in Section III and then present the simulation setup and performance evaluation in Section IV. Section V concludes the paper.

II. TIME-BASED SPECTRAL SPREADING METHOD (TSSM)

Given the large period of UL SRS measurements, speed dependent fast fading effects can hardly be exploited for determining the user speeds. TSSM aims to exploit slow fading such as shadowing since it characterizes large scale variations of the received signal which do not vary much over the period of use. Shadowing plays an important role in the performance analysis of HO process [13], [14]. Extensive measurements were performed to characterize the empirical correlation of shadowing over distance for different environments and different frequencies [15]–[17]. Shadowing can be modeled as a spatial correlated log-normal process, where the correlation between shadow fading at two points separated by a distance δ decreases with the increase of their distance. This property is represented by Gudmundson's correlation model for shadowing [15], where the spatial autocorrelation between shadow fading at two points separated by distance δ is given
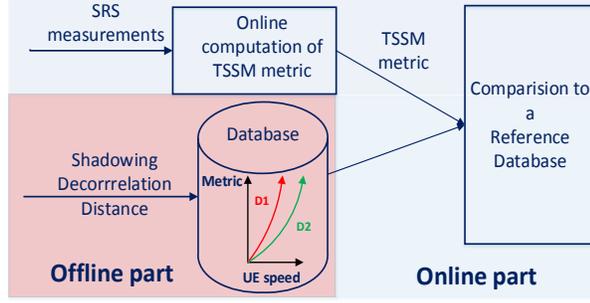
Fig. 2: The functional elements of the proposed online estimation solution.

Fig. 3: TSSM metric as a function of decorrelation distance.

by:

$$
\begin{aligned}
\mathcal{R}_\psi(\delta) &= \mathbb{E}\Big[\Big(\psi(d-\delta)-\mu_\psi\Big)\Big(\psi(d)-\mu_\psi\Big)\Big] \\
&= {\sigma_\psi}^2 e^{-\frac{\delta}{D}} \quad\quad\quad\quad\quad\quad\quad\quad (1)
\end{aligned}
$$

where ψ is the spatial Gaussian shadowing process, σ_ψ and μ_ψ are respectively the shadow standard deviation and the area mean both in decibels (dB), and D is the decorrelation distance at which the signal autocorrelation equals to $1/e$ of its maximum value. It is on the order of the size of the blocking objects or clusters of these objects. As an example, for outdoor systems, D typically ranges from 50 to 100 meters [18]. For a user moving with velocity v, the time autocorrelation can be obtained from (1) by substituting δ by $v\tau$ such that:

$$
\mathcal{R}_\psi(\tau) = \sigma_\psi^2 e^{-\frac{v\tau}{D}}. \quad\quad\quad\quad (2)
$$

The second derivative of the autocorrelation function of the shadowing ψ is given by:

$$
\left.\frac{\partial^2 \mathcal{R}_\psi(\tau)}{\partial^2 \tau}\right|_{\tau=0} = \frac{{\sigma_\psi}^2 v^2}{D^2}. \qu\quad\quad (3)
$$

In [12], the second derivative of the autocorrelation function of the received signal amplitude $r(t)$ is computed as:

$$
\left.\frac{\partial^2 \mathcal{R}_{rr}(\tau)}{\partial^2 \tau}\right|_{\tau=0} = \mathbb{E}\Big[\big(\dot{r}(t)\big)^2\Big]. \qu\quad\quad (4)
$$

Combining (3) and (4), we can obtain the dependency between the speed and the derivative of the SRS measurement expressible as:

$$
\mathbb{E}\Big[\big(\dot{r}(t)\big)^2\Big] \sim \frac{{\sigma_\psi}^2 v^2}{D^2}, \qu\quad\quad\quad (5)
$$

which implies that the standard deviation of the derivative of the measurements is a linear function of v/D. Let $\bar{r}$ be the normalized SRS signal, i.e., $\bar{r}(t) = \frac{r(t)}{\sigma_\psi}$. We obtain the TSSM metric as following:

$$
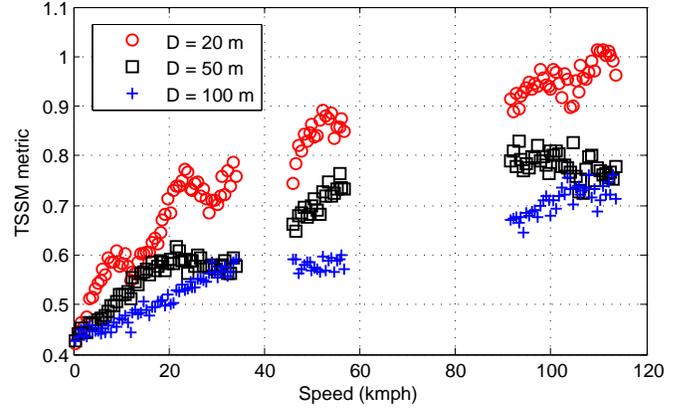\sqrt{\mathbb{E}\Big[\big(\dot{\bar{r}}(t)\big)^2\Big]}, \qu\quad\quad\quad\quad (6)
$$

which is equal to v/D and corresponds to the spectrum spreading of the received signal amplitude.

This leads to the following estimate approximation of the speed as a function of the received signal amplitude:

$$
v \sim D\sqrt{\mathbb{E}\Big[\big(\dot{\bar{r}}(t)\big)^2\Big]}. \qu\quad\quad\quad (7)
$$

III. ONLINE TSSM ALGORITHM

The functional elements of the speed estimation algorithm are presented in Figure 2. By exploiting the variance of the temporal derivative of the SRS signal, we detect, in the time domain, the speed dependent spread of the power spectral density. Following (5), the TSSM metric is increasing with the speed, and with the decrease of the decorrelation distance D. Figure 3 captures the behavior of the TSSM metric as function of the speed and the decorrelation distance. At the eNB, a database (DB) containing different TSSM metric curves for different values of the decorrelation distance D (depending on the environment) is stored. Once the DB is built offline, the speed estimation is derived online by mapping the TSSM metric obtained with the reference DB.

We now introduce an online algorithm for computing the TSSM metric, which allows a customization of the method in function of the required estimation accuracy, period of updating the estimation and implementation cost. The main functional implementation steps required for the online TSSM are illustrated in Fig. 4. Consider the block i of L SRS samples, $\boldsymbol{r_i} = [r_1 \ldots r_L]$. We first normalize $\boldsymbol{r_i}$ as follows:

$$
\bar{\boldsymbol{r}}_{\boldsymbol{i}} = \frac{\boldsymbol{r_i}}{\|\boldsymbol{r_i}\|}. \qu\quad\quad\quad\quad (8)
$$

Then, we compute the derivative for each element k of the block i. A new sample $\bar{r}_k$ is stored in a circular buffer of size n. The derivative d_k is computed by recursive filtering, accounting for the new input value. Below, we give an example with a derivative of order 1:

$$
d_k = \beta \cdot d_{k-1} + (1-\beta) \cdot \frac{\bar{r}_k - \bar{r}_{k-n+1}}{nT}; \text{ for } k = n, \ldots, L \qu(9)
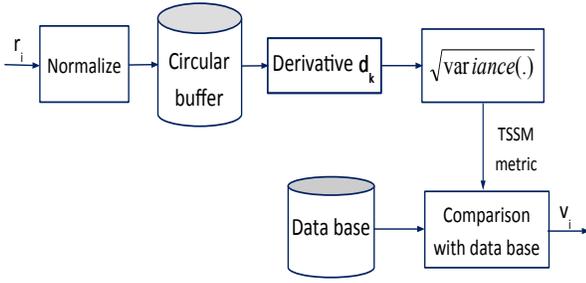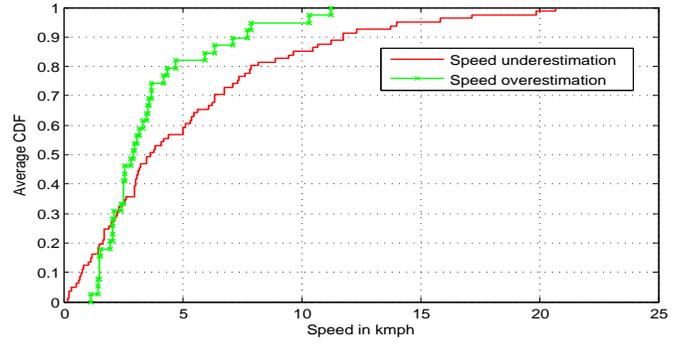$$

Fig. 4: The online TSSM algorithm.

Fig. 5: Average CDF of the algebraic speed estimation error.

Fig. 6: Variation of the average speed overestimation error with the window size for different step sizes.

where $0 < \beta < 1$ stands for the update parameter and T is the measurements sampling period. Once the buffer is full, we take a new block of n SRS measurement samples and iterate the same procedure. This procedure smooths the signal in order to further reduce the fast fading and Doppler variations and capture the slow variations of shadowing. The TSSM metric is then computed per block i from the set of all derivatives d_k following Equation (6). It is obtained from an empirical estimation of the variance on the derivatives according to:

$$TSSM_i = \sqrt{\frac{1}{K}\sum_{k=1}^{K}(d_k - \mathbb{E}[d_k])^2}, \qquad (10)$$

where K is the number of derivatives computed for the tagged window. Finally, we compare the estimated metric in (10) to the reference DB. The outputted value for the estimated speed will be selected online by choosing the closest value of speed form the DB that maps to the estimated TSSM metric. The operation is restarted once a number of new samples equal to m arrive at the buffer. In the sequel, we will refer to $W = nT$ as the speed estimation window size (in seconds) and $S = mT$ as the speed estimation periodicity or the step size (also in seconds).

IV. NUMERICAL RESULTS

A. Simulation Setup

Evaluation is performed under 3GPP-LTE setup by using realistic eNB power measurements of UL SRS provided by Nokia. The signals are instantaneous wideband SRS with 10 MHz bandwidth, 1.9 GHz carrier frequency, a time sampling period T of 80 ms and a Doppler frequency varying from 1 to 166 Hz (low, medium and high Doppler frequencies). The radio channel follows the 3GPP Extended Typical Urban (ETU) environment requirement. The collected samples contain already deployed fractional power control (FPC) with Open Loop Power Control (OLPC) and Closed Loop Power Control (CLPC) periods of 80 ms, P_0 nominal of -78 dBm and a FPC parameter of 0.8. The path-loss model follows 3GPP TS 36.942 urban macro path-loss model [19].

Speeds are assumed to be stationary, taking values from 3 to 120 kmph with a step of 3 kmph. We further consider 9 trajectories for UEs out of which 7 are in straight line with an angle of 45 degrees, one is a circle around the eNB and one is
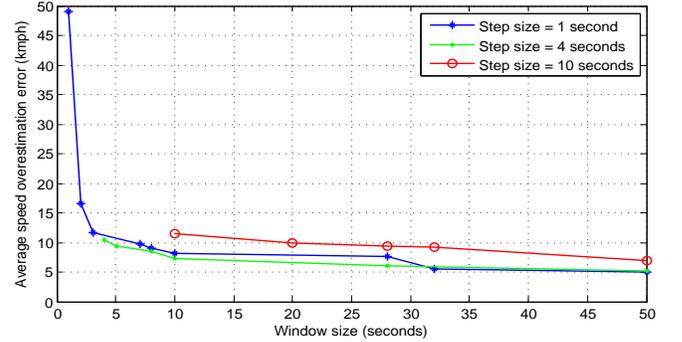
erratic. Each trace corresponds to a distance of 400 meters. The performance results are averaged over all mobility patterns.

B. Simulation Results

Let us first fix the speed estimation window size W to 10 seconds and the step size S to 4 seconds.

1) Speed under- and overestimation error: In Figure 5, we analyze the CDF of the algebraic speed estimation error, defined as the difference between the estimated value of the speed and the actual value of the speed. A close look at Figure 5 shows that the maximum speed underestimation error for 90% of the cases is 12 kmph and the maximum speed overestimation error for 90% of the cases is 7 kmph.

Next, we give some insights on the impact of the speed estimation window size W and the step size S on the speed estimation. Figures 6 and 7 depict respectively the average speed overestimation and the average speed underestimation as function of W for different S. An overall observationis that, in general, the overestimation error is always lower that the underestimation error. As an example, for a window size of $10s$, a first speed estimation can be obtained after $4s$, with an average speed underestimation error of 14 kmph and an average speed overestimation error of 7 kmph. If we decrease the step size to $1s$, the estimated speed is updated more frequently (each second) and the accuracy stays high
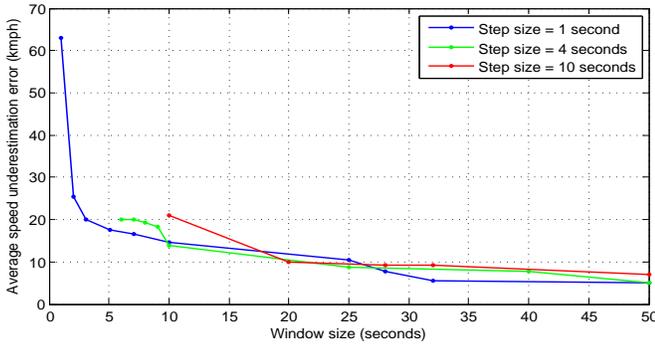
Fig. 7: Variation of the average speed underestimation error with the window size for different step sizes.
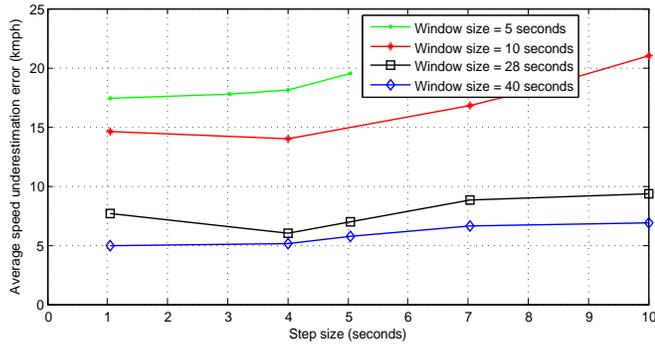

Fig. 8: Variation of the average speed underestimation error with the step size for different window sizes.

with an average speed underestimation error of 15 kmph and an average speed overestimation error of 8 kmph. Decreasing the window size to $7s$ preserves the accuracy at 16 kmph of underestimation and 8 kmph of overestimation. The window size can be further decreased to $4s$ while maintaining the estimation periodicity to $1s$. In this case, the average speed underestimation error becomes 18 kmph and the average speed overestimation error reaches 10 kmph. Further decreasing the window size to $1s$, leads to unacceptable errors of up to 60 kmph. This shows that the online implementation of the algorithm enables the use of speed estimation method in a customized manner, allowing trade-offs between accuracy, estimation update and complexity which we will address next.

Memory consumption: is defined as the size of the memory buffer required by the algorithm, and is equal to the window size W. One can observe from Fig. 6 and Fig. 7 that the accuracy improves with the increase of the window size, although leading to a higher memory utilization. The more we increase the speed estimation window, the lower speed under- and overestimation error is obtain. Results with acceptable accuracy have been obtained even for smaller sizes of the window. Therefore, the online implementation of the TSSM solution allows to choose the best configuration of the parameters depending on the system requirements regarding

the estimation accuracy and memory consumption.

Computational complexity: The step size S of the algorithm is the parameter that determines after what period a new value for the estimated speed will be outputted. For the estimation of one value of the speed, around 10 operations (normalization, computation of derivatives, etc.) are required. Therefore, a configuration of the estimator can be done with trade-offs between computational complexity and estimation delay. For example, if the step size S is set to 10, a new value of the estimated speed is outputted every $10s$, yet resulting in 60 operations per minute. Decreasing the step size to $4s$ results in 150 operations per minute. A fast estimation can be done every $1s$, increasing the computational complexity to 600 operations per minute.

Estimation delay: Fig. 8 shows that we may have a better estimate if we take a large or a small step size S. A good choice of S shall however take into account the dynamic of the input signal. Typically, a high step size corresponds to the case where the speed estimation is computed based on new samples each iteration, which is suitable when the input signal varies a lot. A low step size is fit when the speed estimation is mostly computed based on samples from the previous iteration, implying that the UE does not move rapidly. Accordingly, it is advisable to configure a low S, in order do update samples less frequently. However, a low S makes the algorithm not reacting sufficiently faster when the signal varies a lot, which may degrade the accuracy of the algorithm. This justifies the need for a balance between the performance of the TSSM algorithm and the estimation delay.

2) Probability of speed class estimation: In general MSE problems, the goal is **not** to obtain UE speed estimates that are highly accurate, but only to categorize UEs to improve HO performance. To this end, 3GPP [1] classified the estimated speed into 4 speed (mobility) classes: $[0, 30)$ kmph, $[30, 60)$ kmph, $[60, 90)$ kmph and the last class for the speed values higher or equal to 90 kmph. Next, we will see how the **speed** estimation performances translate in terms of **speed class** estimation. We define the following performance metrics:

- P_0 is the probability of correct speed classification,
- P_1^+ is the probability of classifying the UE speed into the upper speed class,
- P_1^- is the probability of classifying the UE speed into the lower speed class.

We fix the speed estimation window size W to 10 seconds and the step size S to 4 seconds. In Figure 9, one can observe that the probability of correct speed class estimations, P_0, is close to 100% except for values close to speed class frontiers (i.e., 30, 60 and 90 kmph). This is an expected phenomenon, as we observed in Fig. 5, an error of 12 or 7 kmph around the real speed value may lead to an error in speed class estimation when the speed is located in the speed class borders. In these cases, the adjacent class is estimated as it can be further
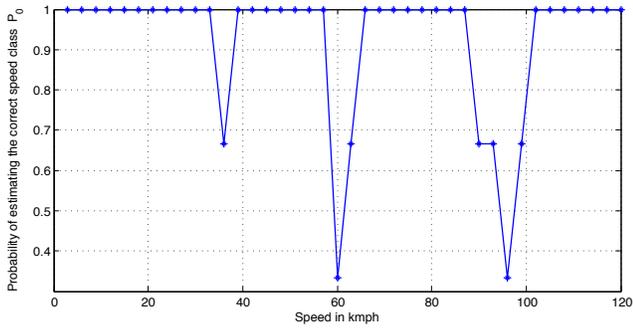
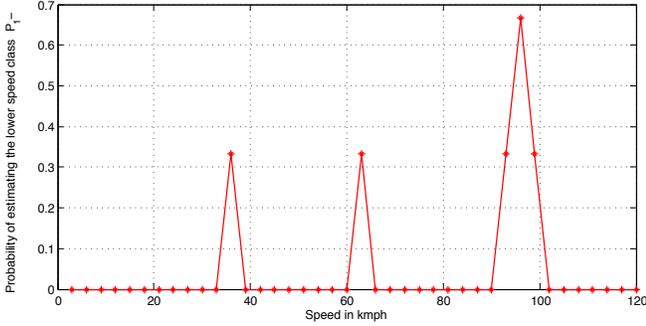Fig. 9: The probability of correct speed class estimation.

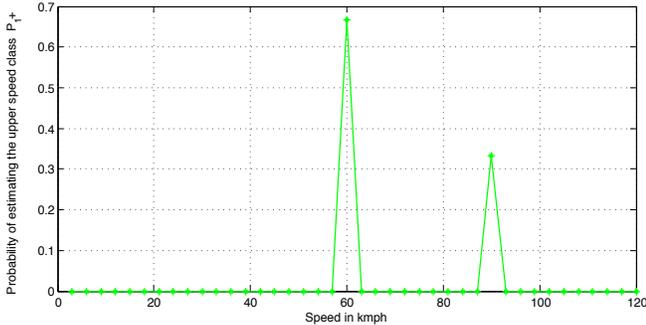Fig. 10: The probability of estimating the lower speed class.

Fig. 11: The probability of estimating the upper speed class.

noticed in Fig. 10 (P_1^-) and Fig. 11 (P_1^+). Notice that speed class underestimation is the most critical case regarding to MSE in HetNet since high speed users may be erroneously handed-over to small cells, which may cause an increase in HO failures and call drop rate.

V. CONCLUSION

We have proposed an online MSE method, called TSSM. We have shown how the online implementation of the TSSM solution allows to adapt the configuration of the parameters to the requirements dictated by each situation with respect to the estimation accuracy, memory consumption, computational complexity, and estimation delay. Performance evaluation based on realistic LTE system data has been carried out to illustrate the effectiveness of the proposed solution. More specifically, for a window size of $10s$, a first speed estimation

can be obtained after $4s$, with an average speed overestimation error of 7 kmph. After this coarse first speed acquisition, subsequent speed estimations can be obtained more frequently (e.g., each second) and the accuracy decreases by only few kmph. Moreover, we have observed that the probability of correct speed class estimation remains always close to 100% except for values close to speed class frontiers. This suggests that the proposed online TSSM algorithm is amenable to fast and low cost implementation.

REFERENCES

[1] 3GPP TR 36.902 V9.3.1 (2011-03), "Evolved universal terrestrial radio access (E-UTRA); mobility enhancements in heterogeneous networks," in *Tech. Rep. V11.1.0*, Jan. 2013.

[2] J. Niu, D. Lee, T. Su, G. Li, and X. Ren, "User classification and scheduling in LTE downlink systems with heterogeneous user mobilities," *IEEE Transactions on Wireless Communications*, Dec. 2013.

[3] S. M. Musa and N. F. Mir, "An analytical approach for mobility load balancing in wireless networks." *Journal of Computing and Information Technology*, vol. 19, no. 3, pp. 169–176, 2011.

[4] A. Dua, F. Lu, and V. Sethuraman, "Speed-adaptive channel quality indicator (CQI) estimation," Jun. 28 2012, US Patent App. 13/253,039.

[5] A. Prasad, P. Lunden, O. Tirkkonen, and C. Wijting, "Energy-efficient flexible inter-frequency scanning mechanism for enhanced small cell discovery," in *IEEE VTC-Spring*, 2013.

[6] H. Zhang and A. Abdi, "Mobile speed estimation using diversity combining in fading channels," in *IEEE GLOBECOM*, Nov 2004.

[7] Y. Zheng and C. Xiao, "Mobile speed estimation for broadband wireless communications over Rician fading channels," *IEEE Transactions on Wireless Communications*, vol. 8, no. 1, pp. 1–5, Jan 2009.

[8] H. Arslan, L. Krasny, D. Koilpillai, and S. Chennakeshu, "Doppler spread estimation for wireless mobile radio systems," in *IEEE Wireless Communications and Networking Conference*, 2000.

[9] P. Deogun, M. Mehta, A. Karandikar, and N. Akhtar, "Trajectory based mobility state estimation for heterogeneous cellular networks," *IEEE Wireless Communications and Networking Conference*, 2016.

[10] D.-H. Kim, K.-H. Choi, K.-J. Li, and Y.-S. Lee, "Performance of vehicle speed estimation using wireless sensor networks: a region-based approach," *The Journal of Supercomputing*, 2014.

[11] D. Herculea, C. S. Chen, M. Haddad, and V. Capdevielle, "Straight: Stochastic geometry and user history based mobility estimation," in *ACM HotPOST '16*.

[12] M. Haddad, D.-G. Herculea, E. Altman, N. Ben Rached, V. Capdevielle, C. S. Chen, and F. Ratovelomanana, "Mobility state estimation in LTE," in *IEEE WCNC*, 2016.

[13] N. Zhang and J. Holtzman, "Analysis of handoff algorithms using both absolute and relative measurements," in *IEEE Vehicular Technology Conference*, 1994.

[14] G. Pollini, "Trends in handover design," *IEEE Communications Magazine*, vol. 34, no. 3, pp. 82–90, Mar 1996.

[15] M. Gudmundson, "Correlation model for shadow fading in mobile radio systems," *Electronics Letters*, vol. 27, no. 23, pp. 2145–2146, Nov 1991.

[16] J. Weitzen and T. Lowe, "Measurement of angular and distance correlation properties of log-normal shadowing at 1900 MHz and its application to design of PCS systems," *IEEE Transactions on Vehicular Technology*, Mar 2002.

[17] A. Algans, K. Pedersen, and P. Mogensen, "Experimental analysis of the joint statistical properties of azimuth spread, delay spread, and shadow fading," *Selected Areas in Communications, IEEE Journal on*, 2002.

[18] M. Marsan and G. Hess, "Shadow variability in an urban land mobile radio environment," *Electronics Letters*, vol. 26, pp. 646–648, May 1990.

[19] 3GPP Technical Specification Group RAN, "E-UTRA; LTE RF system scenarios," in *TS 36.942*, 2008-2009.