Tracking Improvement for Vehicular Applications Due to Use of a Vector Delay Lock Loop

D. Massaro[1], M. Leonardi[1], M. Mascolo[2], G. Galati[1], L. Sfarzo[2]
[1]Tor Vergata University, Department of Electronic Engineering Via del Politecnico 1, 00133 Rome, Italy
e-mail: davide.massaro3@gmail.com; mauro.leonardi@uniroma2.it; gaspare.galati@uniroma2.it
[2]T.R.S. spa, Via della Bufalotta 378, 00139 Rome, Italy
e-mail: michele.mascolo@trs.it; luca.sfarzo@trs.it

ABSTRACT
In this paper a Kalman Filter is introduced in the GPS-L1 signal tracking loop to improve its robustness in adverse environments. The proposed algorithm uses two different approaches with respect to the carrier and to the code: the carrier tracking is performed by a single three-state Kalman Filter for each channel, in addition to the traditional PLL Costas loop. A vector loop is used, instead, to track the code by an Extended Kalman Filter (EKF) to process the information provided by all the receiver channels.
First, the mathematical model of the system is described. After the Kalman Filter parameter tuning, three different tests are performed tracking both simulated and real data by a Software-Defined Radio GNSS receiver implemented in *Matlab*© (based on an Open Source code provided by **[1]**).
In the end an exhaustive analysis of the results is made by comparing them to the performances of a traditional tracking loop.

INTRODUCTION
The continuous innovation of the GNSS technology has improved reliability in many applications, however there are some operational situations in which the traditional algorithms might have problems. In cases such as tunnels or urban environments a receiver may lose the lock of some satellites and be forced to re-acquire them. This event will likely cause larger position error and may require longer re-acquisition time. Such an event, in some circumstances, may even cause a loss of continuity of the positioning service, given the operational requirements under which it is used. This problem can be mitigated by exploiting an algorithm able to make predictions when measures are not available.

A traditional tracking loop consists of two different sections: the first tracks the carrier of the signal and often exploits a Costas PLL loop; the second tracks the Pseudo Random Noise (PRN) sequence, widely used in Code Division Multiple Access (CDMA) transmissions, exploiting a DLL loop. Both operate by generating a local carrier/code replica, similar to the received carrier/code, and by correlating it to the received signal. The resulting error is filtered using a Loop Filter and then it is used to generate a corrected carrier/code replica. So the receiver tracks the continuous phase and frequency variations of the signal which are consequences of the relative motion between the user and the satellite **[2]**.

The aim of using Kalman Filters in GNSS tracking loops is to improve the robustness of the receivers exploiting the estimation and prediction capability of this widely used algorithm. In similar cases as a complete signal blockage (for example when the user passes through a tunnel) a traditional tracker could not keep on tracking the signal when it is available again but, exploiting an algorithm which propagates the user position estimation and having all the ephemeris, it is possible to estimate the phase and frequency expected in the predicted user position **[3]**. This could be fundamental when a continuous navigation solution is needed.

SYSTEM DESCRIPTION
The tracker implemented in this work uses two different techniques to track the code and the carrier, as in **[4]**. The carrier tracking is very critical given the high sensitivity and promptness required by the receiver in order to keep track of quick carrier variations: in some high dynamics scenarios , noisy measurements could affect the performances of the other channels in case of a vector approach. Hence, concerning carrier tracking, it could be convenient to use a traditional approach maintaining each channel independent. On the other hand, in the case of satellite blockage with measurements not available, the system would not be able to work correctly. Using a Kalman Filter it is possible to predict the Doppler frequency according to a theoretical model still maintaining independent channels. Since the code tracking is not affected by the aforementioned problem, it seems more appropriate to employ a vector EKF approach to estimate the code phase and frequency and a single filter to estimate the user position.
The filter used to track the carrier in each channel consists of a three-state Kalman Filter. Its model is the following **[5]**:

$$\begin{bmatrix}\theta_{e,k}\\ f_{d,k}\\ a_{d,k}\end{bmatrix}=\begin{bmatrix}1 & T & \frac{T^2}{2}\\ 0 & 1 & T\\ 0 & 0 & 1\end{bmatrix}\begin{bmatrix}\theta_{e,k-1}\\ f_{d,k-1}\\ a_{d,k-1}\end{bmatrix}-\begin{bmatrix}T\\ 0\\ 0\end{bmatrix}f_d^{NCO}+\begin{bmatrix}w_\theta\\ w_d\\ w_a\end{bmatrix} \quad \textbf{(1)}$$

where $\theta_{e,k}$ is the phase error (in radians) between the incoming signal and the local replica, $f_{d,k}$ is the estimated Doppler Frequency (*Hz*), f_d^{NCO} is the Doppler Frequency generated by the Numerically-Controlled Oscillator (NCO) and $a_{d,k}$ is the Doppler rate (*Hz/sec*). T is the Kalman Filter update time which corresponds with the PLL integration time, set to 1 *ms* in this work. The vector $[w_\theta,\ w_d,\ w_a]^T$ represents the noise process whose Covariance Matrix is defined as follows:

$$\mathbf{Q}=Q_\theta\begin{bmatrix}T & 0 & 0\\ 0 & 0 & 0\\ 0 & 0 & 0\end{bmatrix}+Q_d\begin{bmatrix}\frac{T^3}{3} & \frac{T^2}{2} & 0\\ \frac{T^2}{2} & T & 0\\ 0 & 0 & 0\end{bmatrix}+Q_a\begin{bmatrix}\frac{T^5}{20} & \frac{T^4}{8} & \frac{T^3}{6}\\ \frac{T^4}{8} & \frac{T^3}{3} & \frac{T^2}{2}\\ \frac{T^3}{6} & \frac{T^2}{2} & T\end{bmatrix}. \quad \textbf{(2)}$$

Q_θ and Q_d describe respectively the statistics of the phase error and Doppler Frequency estimations, they depend on the clock stability so their values can be set using the Allan Variances **[6]**:

$$Q_\theta = 2h_0 \quad \textbf{(3)}$$

$$Q_d = 8\pi^2 h_{-2} \quad \textbf{(4)}$$

where h_0 and h_{-2} can be obtained by Table 1:

Clock Type	h_0	h_{-2}
Crystal	$2\cdot10^{-19}$	$2\cdot10^{-20}$
Ovenized Crystal	$8\cdot10^{-20}$	$4\cdot10^{-23}$
Rubidium	$2\cdot10^{-20}$	$4\cdot10^{-29}$

Table 1 Clock parameters for different clock types.

Q_a depends on the scenario and must be set manually.
An arctangent PLL Discriminator is employed in order to provide a measure of the phase error to the Kalman Filter:

$$\theta_{e,k}^{mea}=\tan^{-1}\left(\frac{Q_k}{I_k}\right) \quad \textbf{(5)}$$

where Q_k and I_k are the Quadrature and In-Phase correlation samples integrated over a period T. The *connection matrix* $\mathbf{H_\theta}$ can be obtained by considering the the average of the phase error over a period of length T:

$$\hat{\theta}_{e,k}^{mea}=\frac{1}{T}\int_0^T\theta_{e,k}(\tau)\ \ d\tau \quad \textbf{(6)}$$

and substituting equation (1) we get:

$$\hat{\theta}_{e,k}^{mea}=\left[1,\frac{T}{2},\frac{T^2}{6}\right]\begin{bmatrix}\theta_{e,k-1}\\ f_{d,k-1}\\ a_{d,k-1}\end{bmatrix}-\frac{T}{2}f_d^{NCO}+\nu_k \quad \textbf{(7)}$$

where ν_k is the measurement noise and

$$\mathbf{H_\theta}=\left[1,\frac{T}{2},\frac{T^2}{6}\right]\ . \quad \textbf{(8)}$$

The only parameter used to estimate the carrier frequency is $\theta_{e,k}$ while $f_{d,k}$ and $a_{d,k}$ are needed to evaluate the former parameter and are not propagated outside the Kalman Filter.

The approach followed to track the code is quite different and uses a single Extended Kalman Filter (EKF) which process the information provided by all the tracking channels, as described in the Vector Delay Lock Loop (VDLL) theory **[7]**. The EKF is the application of the Kalman Filter to a non-linear system: in this case it is performed by linearizing the navigation equations system (related to the pseudorange error) in an approximate user position; hence the state vector does not contain the position but the position error.

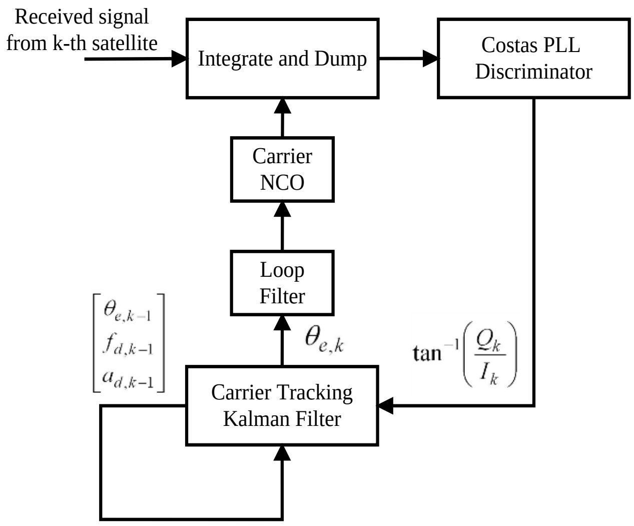

Figure 1 – single channel Kalman filter for carrier tracking

The same technique can be applied to the pseudorange rate error equations. The corresponding EKF state vector can be written as **[8]**:

$$\boldsymbol{\delta}\mathbf{X_k} = \begin{bmatrix} \delta x_k \\ \delta y_k \\ \delta z_k \\ \delta v_{x,k} \\ \delta v_{y,k} \\ \delta v_{z,k} \\ \delta b_k \\ \delta t_{d,k} \end{bmatrix} \qquad \textbf{(9)}$$

where δx_k, δy_k, δz_k represent the position error in the x, y and z coordinates of the Earth-Centered Earth-Fixed (ECEF) reference system. $\delta v_{x,k}$, $\delta v_{y,k}$ and $\delta v_{z,k}$ are the user velocity errors, δb_k the error of the clock bias estimation and $\delta t_{d,k}$ the error related to the bias drift. All the state vector components are measured in terms of position and velocity, so their measure units are *m* and *m/s*. Before running the VDLL it is necessary to know some parameters to initialize the system: they are provided by a traditional tracking loop. The VDLL algorithm can be divided into three steps: prediction, measurement and estimation.

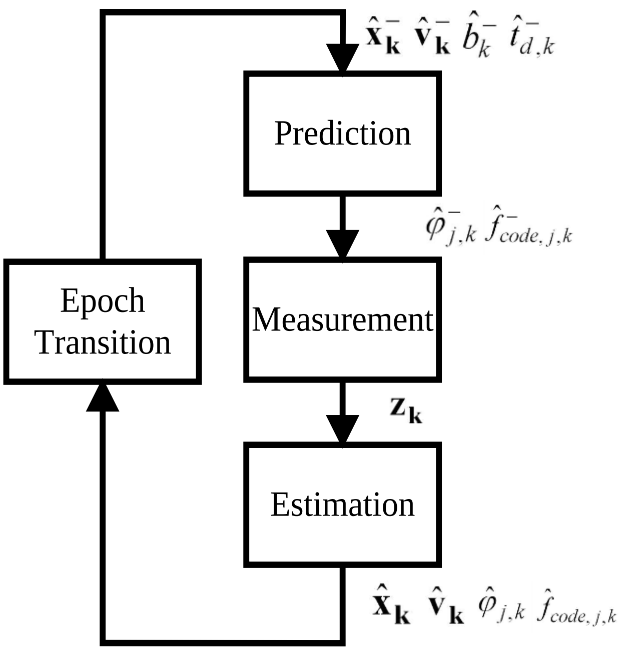

Figure 2 – VDLL algorithm scheme

(A) Prediction:

Starting from the estimation of user and available satellites positions, the forecast of the expected values of the code phase and frequency are given by the following equations:

$$\hat{\varphi}_{j,k}^- = \hat{\varphi}_{j,k-1} + \frac{f_{code}}{c}\left[\left(\mathbf{\Delta x_{j,k,k-1}} - T\hat{\mathbf{v}}_\mathbf{k-1}\right)^T \mathbf{a_{j,k}}\right] \qquad \textbf{(10)}$$

$$\hat{f}_{code,j,k}^- = f_{code} - \frac{f_{code}}{c}\left[t_{d,k-1} - \left(\mathbf{v_{j,k-1}} - \hat{\mathbf{v}}_\mathbf{k-1}\right)^T \mathbf{a_{j,k}}\right] \qquad \textbf{(11)}$$

where φ is the code phase in chips, $t_{d,k}$ is the bias drift in *m/s*, $\mathbf{\Delta x_{j,k,k-1}}$ is a vector containing the variation (in metres) of the satellite position from epoch *k* and *k-1*, $\mathbf{v_{j,k-1}}$ is the velocity vector (*m/s*) of the j-th satellite at epoch *k-1*, f_{code} is the nominal code frequency (1.023 *Mchip/sec*), *c* is the speed of light. Considering the equation **(10)** the terms in the round brackets are the corrections needed to compensate the phase created by the variation of the Line Of Sight (LOS) distance between the user and the satellite. In equation **(11)** the velocities are used to compensate the Doppler Frequency caused by the relative speed between user and satellite while the bias drift is used to compensate the clock instability. Hence the use of the first one is implied by the dynamical scenario while the second is used to keep track of the not ideal oscillator.

(B) Measurement

The code phase/frequency information obtained by equations **(10)**, **(11)** are used to generate the local replica of the signals to correlate to the incoming signal. After correlation, the phase/frequency error is evaluated using typical DLL/FLL discriminators. The DLL discriminator has the following expression:

$$L_{\varphi,k} = \frac{1}{2}\left(\frac{\sqrt{I_{E,k}^2 + Q_{E,k}^2} - \sqrt{I_{L,k}^2 + Q_{L,k}^2}}{\sqrt{I_{E,k}^2 + Q_{E,k}^2} + \sqrt{I_{L,k}^2 + Q_{L,k}^2}}\right) \qquad \textbf{(12)}$$

The characteristic of this discriminator is practically linear in an input error range of +-0.5 chips and it does not depend on the input dynamic due to the normalization **[9]**. The FLL discriminator functions is:

$$cross = I_{P,k-1}Q_{P,k} - I_{P,k}Q_{P,k-1} \qquad \textbf{(13)}$$

$$dot = I_{P,k-1}I_{P,k} - Q_{P,k-1}Q_{P,k} \qquad \textbf{(14)}$$

$$L_{f,k} = \frac{cross \cdot sign(dot)}{2\pi T\left(I_{P,k}^2 + Q_{P,k}^2\right)} \qquad \textbf{(15)}$$

For each tracking channel the measures of the phase error and of the frequency error are converted into a pseudorange and a pseudorange error and then are sent to the EKF. The measurement vector $\mathbf{z_k}$ can be defined as follows:

$$\mathbf{z_k} = \begin{bmatrix} \frac{f_{code}}{c} L_{\varphi,1,k} \\ \vdots \\ \frac{f_{code}}{c} L_{\varphi,N,k} \\ \frac{f_{L1}}{c} L_{f,1,k} \\ \vdots \\ \frac{f_{L1}}{c} L_{f,N,k} \end{bmatrix} = \begin{bmatrix} \Delta\rho_1 \\ \vdots \\ \Delta\rho_N \\ \Delta\dot{\rho}_1 \\ \vdots \\ \Delta\dot{\rho}_N \end{bmatrix} \qquad \textbf{(16)}$$

(C) Estimation

Using the measure vector it is possible to calculate the EKF innovation. The connection matrix is obtained by linearization of the navigation equation system:

$$\mathbf{H_k} = \begin{bmatrix} a_{1,x,k} & a_{1,y,k} & a_{1,z,k} & 0 & 0 & 0 & 1 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ a_{N,x,k} & a_{N,y,k} & a_{N,z,k} & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & a_{1,x,k} & a_{1,y,k} & a_{1,z,k} & 0 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & a_{N,x,k} & a_{N,y,k} & a_{N,z,k} & 0 & 1 \end{bmatrix} \quad \mathbf{(17)}$$

so it contains the unit LOS vectors.

The noise process covariance matrix, **Q**, is defined as follows:

$$\mathbf{Q} = \begin{bmatrix} \sigma_x^2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \sigma_y^2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \sigma_z^2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \sigma_{v,x}^2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \sigma_{v,y}^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \sigma_{v,z}^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \sigma_b^2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \sigma_d^2 \end{bmatrix} \quad \mathbf{(18)}$$

where the variance values are set according to the user dynamics.

The noise measurement covariance matrix, **R**, is defined as follows:

$$\mathbf{R} = \begin{bmatrix} \sigma_{1,code}^2 & 0 & \cdots & 0 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & \sigma_{N,code}^2 & 0 & \cdots & 0 & 0 \\ 0 & 0 & \cdots & 0 & \sigma_{1,carr}^2 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & 0 & \cdots & 0 & \sigma_{N,carr}^2 \end{bmatrix} \quad \mathbf{(19)}$$

. Widening of the EKF bandwidth can be obtained by inflating the values of the parameters contained in the matrixes **Q** and **R**.

The updated state vector can be obtained using the typical Kalman Filter equation:

$$\boldsymbol{\delta}\mathbf{X_k} = \boldsymbol{\delta}\mathbf{X_k^-} + \mathbf{K_k}\left(\mathbf{z_k} - \mathbf{H_k}\boldsymbol{\delta}\mathbf{X_k^-}\right) \quad \mathbf{(20)}$$

Using the position/velocity/clock corrections given by the state vector it is possible to correct the user position, velocity, the clock bias and drift:

$$\hat{\mathbf{x}}_\mathbf{k} = \hat{\mathbf{x}}_\mathbf{k}^- + \boldsymbol{\delta}\mathbf{x_k} \quad \mathbf{(21)}$$

$$\hat{\mathbf{v}}_\mathbf{k} = \hat{\mathbf{v}}_\mathbf{k}^- + \boldsymbol{\delta}\mathbf{v_k} \quad \mathbf{(22)}$$

$$\hat{b}_k = \hat{b}_k^- + \delta b_k \quad \mathbf{(23)}$$

$$\hat{t}_{d,k} = \hat{t}_{d,k}^- + \delta t_{d,k} \quad \mathbf{(24)}$$

Using the corrected parameters from equations **(21)**, **(22)** **(23)** and **(24)** the estimated code phase/frequency can be obtained:

$$\hat{\varphi}_{j,k} = \hat{\varphi}_{j,k}^- - \frac{f_{code}}{c}\left(\boldsymbol{\delta}\mathbf{x_k^T}\mathbf{a_{j,k}} + \delta b_k\right) \quad \mathbf{(25)}$$

$$\hat{f}_{code,j,k} = \hat{f}_{code,j,k}^- - \frac{f_{code}}{c}\left(\boldsymbol{\delta}\mathbf{v_k^T}\mathbf{a_{j,k}} + \delta t_{d,k}\right) \quad \mathbf{(26)}$$

The corrections shown in equations **(25)** and **(26)** are refinements of the predictions **(10)**, **(11).** The EKF state transition from epoch *k* to *k+1* is performed according to the following equations:

$$\boldsymbol{\delta}\mathbf{X_{k+1}^-} = \mathbf{F} \quad \boldsymbol{\delta}\mathbf{X_k} \quad \mathbf{(27)}$$

where

$$\mathbf{F} = \begin{bmatrix} 0 & 0 & 0 & T & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & T & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & T & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & T \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad \mathbf{(28)}$$

$$\hat{\mathbf{x}}_\mathbf{k+1} = \hat{\mathbf{x}}_\mathbf{k} + T\hat{\mathbf{v}}_\mathbf{k} \quad \mathbf{(29)}$$

$$\hat{\mathbf{v}}_\mathbf{k+1} = \hat{\mathbf{v}}_\mathbf{k} \quad \mathbf{(30)}$$

PERFORMED TESTS

The tests performed in this work can be related to two different types of dataset: first, a static scenario has been analyzed and then the system has been tested by a dynamic user. In both cases a traditional loop has been used as well to compare the performances.

(A) Static User

The static signal has been collected by the *GNSS@RadarLab* receiver of Tor Vergata University. Its set up are a sampling frequency f_s = 4 *MHz* and an intermediate frequency f_{IF} = 1 *MHz*. The acquisition threshold (i.e. the ratio between the main lobe and the highest secondary lobe of the correlation performed by the acquisition function) has been set to 3.5.

.

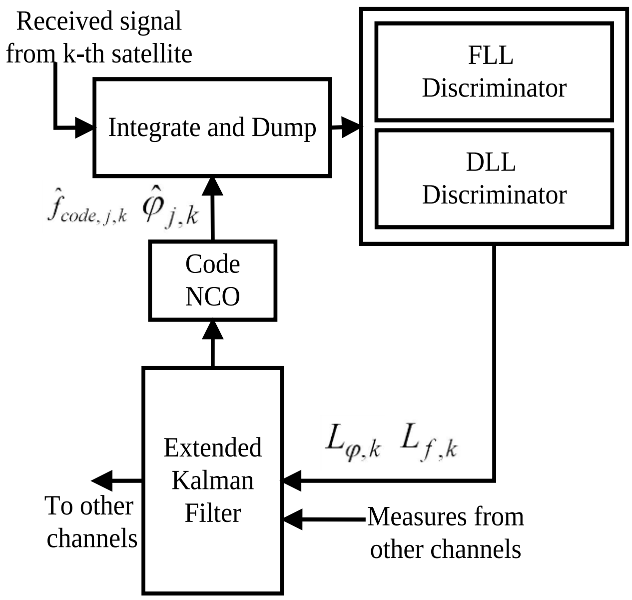

Figure 3 – scheme of performed VDLL

The parameters used by this dataset are summarized in Table 2 and Table 3Table 4:

Carrier Tracking Parameters	
h_0	$2 \cdot 10^{-19}$
h_{-2}	$2 \cdot 10^{-20}$
Q_a	10^{12} *Hz/s*
R	1 rad^2
Damping Ratio, ζ	0.7
Noise Bandwidth, B_n	85 *Hz*

Table 2 Carrier Tracking Parameters for static user

Code Tracking Parameters	
σ^2_{pos}	20 m^2
σ^2_{vel}	3 m^2/s^2
σ^2_b	10^{-7} m^2
σ^2_d	0.1 m^2/s^2
σ^2_{code}	1500 m^2
σ^2_{carr}	900 m^2/s^2

Table 3 Code Tracking paremeters for static user

In the first test a signal block of 50 *s* duration was processed: both the traditional tracker and the VDLL were able to track the signal, as shown, for one channel, in Figure 4**:**

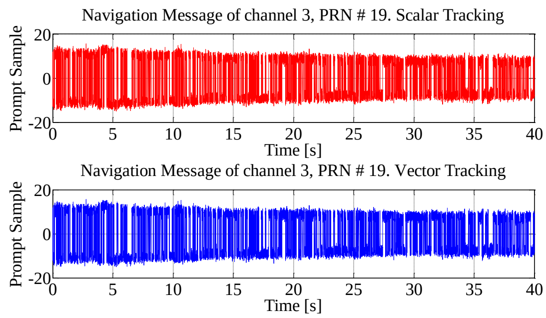

Figure 4 – In phase signal component as output of the scalar and vector tracker

In the second test a complete satellite blockage of 10 *s* was simulated (from seconds 31 to 41). In this case the VDLL is able to predict the carrier and code parameters by exploiting its theoretical model due to the unavailability of the measures. The traditional tracker does not rely on any model to propagate code and carrier, and relies only on measurements. So, the VDLL, under such circumstances, has more chances to re-lock the signal without performing the acquisition again. This is clearly shown in Figure 5.

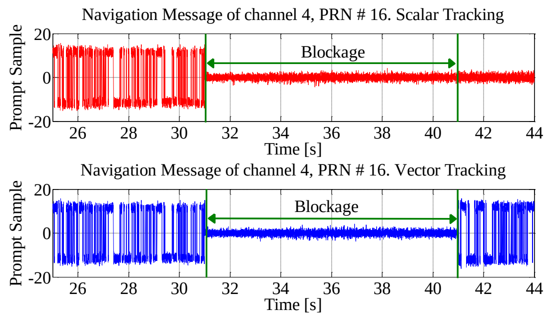

Figure 5 - In phase signal component as output from scalar and vector tracker, in blockage scenario

Something similar happens when there are small blockages, as shown in Figure 6.

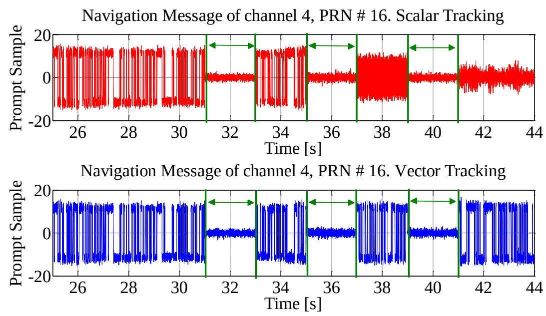

Figure 6 - In phase signal component as output from scalar and vector tracker, in multi blockage scenario

It is cleat that exploiting a vectorial tracking and, in a wider view, a Kalman filter, improves the robustness of the tracking function. The recovery time for VDLL is quite near to zero meaning no delays to re-lock the satellite signal.

(B) Dynamic User

The dynamic signal has been collected by a *Spirent 12-channel* GPS simulator, property of University of Colorado at Boulder. In this case the sampling frequency was $f_s = 4\ MHz$ and the intermediate frequency was $f_{IF} = 1\ MHz$. The parameters used for this dataset are summarized in Table 4 and Table 5.

Carrier Tracking Parameters	
h_0	$2 \cdot 10^{-19}$
h_{-2}	$2 \cdot 10^{-20}$
Q_a	$10^2\ Hz/s$
R	$10^{-6}\ rad^2$
Damping Ratio, ζ	0.7
Noise Bandwidth, B_n	$25\ Hz$

Table 4 Carrier Tracking Parameters for dynamic user

Code Tracking Parameters	
σ^2_{pos}	$20\ m^2$
σ^2_{vel}	$3\ m^2/s^2$
σ^2_{b}	$10^{-7}\ m^2$
σ^2_{d}	$0.1\ m^2/s^2$
σ^2_{code}	$1500\ m^2$
σ^2_{carr}	$900\ m^2/s^2$

Table 5 Code Tracjing Parameters for dynamic user

The main difference between the Kalman Filter parameters used for this dataset and the ones of the static test can be seen in the behavior of the Carrier Tracking section since the VDLL parameters depend only on the user dynamics and, in a dynamics of a car, for example, the user could be moving or not.

In the first test system stability has been analyzed by processing the signal for 120 *s* . The results are similar to the ones of the traditional tracker: the Kalman Filter process introduces a smoothing on carrier and code parameters which makes them less noisy. In Figure 7 and in Figure 8 the tracking performances and the user trajectory are shown.

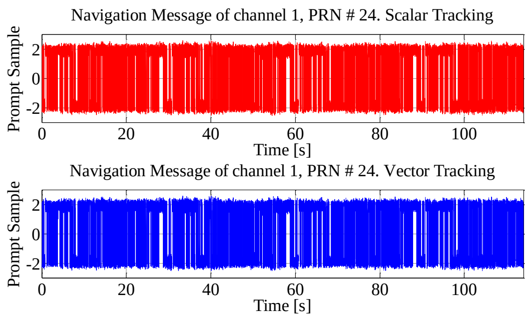

Figure 7 - In phase signal component as output from scalar and vector tracker

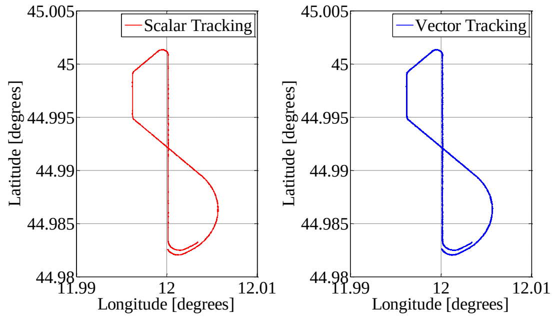

Figure 8 – Track points computed using respectively the scalar tracker and EKF

It should be considered that the trajectories in Figure Figure 8 are calculated using two different methods: in the scalar tracking a navigation computation function based on the time of arrival (measured in terms of signal samples) is used, while in the VDLL test the EKF provides the position/velocity errors based on the code/carrier measures so they can be propagated obtaining the user position.

In the second test a whole satellite signal blockage was simulated for a period of 20 *s*. In a real environments it could happen, for example, if a user passes through a tunnel or through an environment with an high density of natural or man-made obstacles (a city or a forest, for example). In this case a GNSS receiver would not have any available signal and the measures would be based only on thermal noise: for this reason the code and carrier parameters would be estimated randomly, and a traditional tracker will lose lock. The advantage of a VDLL or a Kalman Filter aided tracker lies in the possibility to exploit a theoretical model, which could reduce the parameters estimation errors respect to what we yield using a traditional tracker. Exploiting EKF, the system propagates the position errors and the carrier frequency for each channel. The simulation has been made by substituting thermal noise to the signal:

$$s_{rx}(t) = s(t) + n(t)$$
$$\downarrow$$
$$s_{rx}(t) = n(t)$$
$$\downarrow$$
$$s_{rx}(t) = s(t) + n(t)$$

where $s_{rx}(t)$ is the received signal, $s(t)$ contains the information and $n(t)$is the thermal noise.

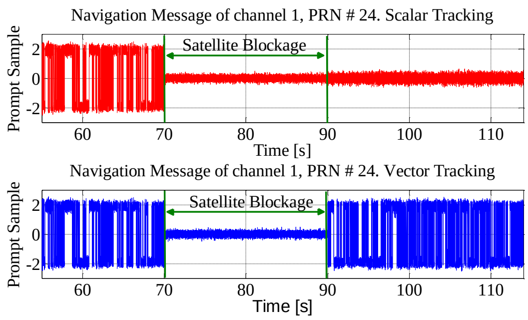

Figure 9 – In-phase signal component as output from scalar and vector tracker, in blockage scenario

As it can be observed in Figure 9, the VDLL suddenly re-locks the signal due to the Kalman Filter aiding while the traditional tracker loses lock and must re-acquire the satellites again so wasting time and making the user position no more available. Not necessarily all the channels will lose lock, since the code and carrier parameters estimation is based on noise and it might be possible that the estimation is near to the actual parameters. But it is a purely random and so unpredictable result.

A similar behavior, concerning Doppler frequency, can be seen in Figure 10. After signal blockage, the scalar algorithm is no more able to provide a correct measurement while the vector one, exploiting its estimation capabilities, continues to provide a correct measurement when the signal is received again.

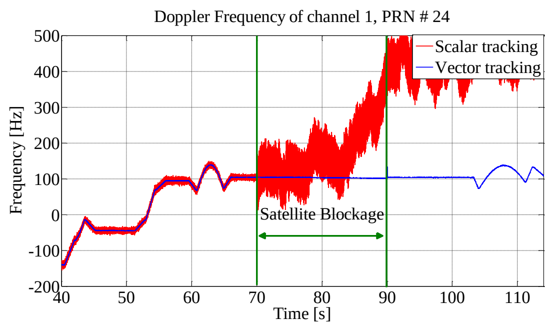

Figure 10 – Doppler frequency computed by scalar and vectorial tracker, in blockage scenario

A variant of this test has been performed by blocking the signal of some satellites making less than four satellites available to the receiver. In this case the position calculated by the traditional tracking clearly diverges from the real trajectory, as shown in Figure 11.

This situation represents a real scenario in which the VDLL could be used in order to avoid the loss of lock and so a loss of the position information. Considering an vehicular application which relies on navigation by exploiting a vehicular receiver which produces position information to a central block, the loss of service continuity (current position) may produce an hazard for the entire system. In this sense exploiting a VDLL algorithms in this kind of application could improve the safety of the system.

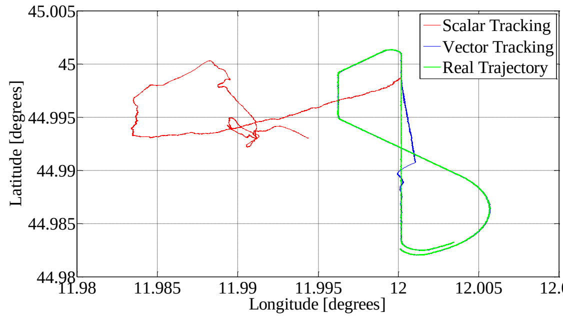

Figure 11 - Track points computed using respectively the scalar tracker and EKF, in blockage scenario

The last test shows what might usually happen in an urban environment: a set of four consequent satellite blockages of 4 *s* length were realized. The results highlight a higher robustness in the VDLL due to what previously explained.

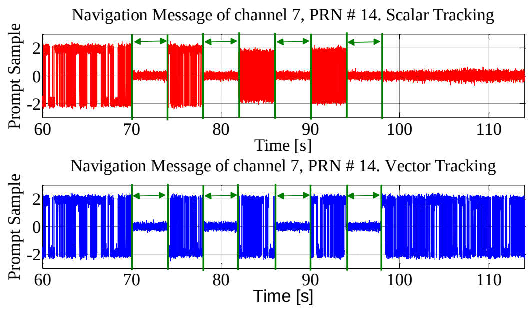

Figure 12 - In phase signal component as output from scalar and vector tracker, in multi blockage scenario

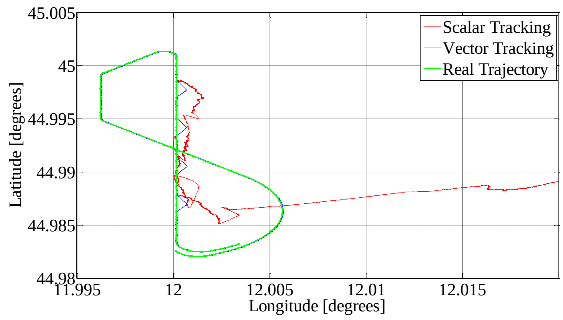

Figure 13 - Track points computed using respectively the scalar tracker and EKF, in multi blockage scenario

CONCLUSIONS

In this work a Vector Delay Lock Loop for vehicular receivers was designed, realized and tested using a GPS Software-Defined Radio Receiver. The first part of the paper includes an exhaustive description of the mathematical model used to realize the system and the steps of the related algorithm are shown. The Kalman Filter algorithm has been used in two different ways according to code and carrier tracking: we exploited a vector approach for the code and a scalar approach for the carrier. In the scalar approach a Kalman Filter is used independently on each channel and it introduces a better smoothing attitude as well as prediction capabilities than standard PLL algorithms. The vector tracking, instead, filters the information from all the receiver channels by exploiting a single Extended Kalman Filter. Then the tests performed are shown: initially, the system was used to track the signal recorded by a fixed receiver and then by a simulation of a dynamic receiver placed in a moving car. In both cases three tests were realized:

- a nominal condition test, in which the signals were tracked without any interference or blockage, to evaluate the system stability;
- a 20 *s* blockage of all the satellite signals to simulate what might happen when the user passes through a tunnel or a forest;
- a set of small blockages to simulate what a user can find in an urban environment.

In these cases, the Kalman Filter aided tracker showed better performances than the traditional tracker, and a higher robustness as well. It is the consequence of the Kalman Filter use, as the algorithm is able to make a dynamic filtering of the measures by exploiting a theoretical model as reference, too. This approach may give a big advantage when the receiver is placed on a moving vehicle. For example, a system which relies on the knowledge of the current car position for some reason (i.e. a car sharing system for billing reasons)with car moving in an urban scenario, the loss of lock situation may represents an hazard for the system for multiple aspects: safety of vehicles (if the track is loss someone could steal the car), to guarantee legal conditions (ensure in case of crash the exact coordinates of vehicle), for management (if a car is parked under a tree, the system does not know where the car is) and so on. Many of this unwanted situations could be mitigated by introducing the vector tracking. For these reasons the Kalman approach can be an important technique to develop GNSS based applications which will be applied in urban and/or harsh environments.

However the system is strongly dependant on the Filter tuning process and could be improved by exploring some automatic methods to set the Filter parameters: it could be interesting, for example, to use an Adaptive Kalman Filter (AKF) instead of a traditional one. Further work can be performed in mixing the typical DLL, FLL and PLL sensors with other types of sensors such as gyroscopes or inertial, thus exploiting the capability of the Kalman Filter to filter measures coming from different types of sensors.

ACKNOWLEDGMENT

The authors would like to point out the great kindess of Dr. Sihao Zhao and Prof. Dennis Akos for the help and the material provided. The signal used in the dynamic tests has been simulated and recorded by them at University of Colorado at Boulder.

REFERENCES

[1] A Software-defined GPS and Galileo Receiver - Kai Borre, Dennis A. Akos, Nicolaj

[2] Understanding GPS: Principles and Applications, Second Edition [Hardcover] by Elliott D. Kaplan, Christopher Hegarty

[3] Global Positioning System: Theory and Applications, Volume 1, Bradford W. Parkinson, James J. Spilker

[4] Vector Delay/Frequency Lock LoopImplementation and Analysis Matthew Lashley, Navigation Technology Associates David M. Bevly, Auburn University

[5] P. Lian. Improving tracking performance of pll in high dynamic applications. Master's thesis, University of Calgary, 2004.

[6] R.G. Brown; P.Y.C. Hwang. Introduction To Random Signals And Applied Kalman Filtering. Third Edition. John Wiley and Sons, Inc, 1997.

[7] B.W. Parkinson; J.J. Spilker Jr. Global Positioning System: Theory and Applications. Volume I. American Institute of Aeronautics and Astronautics, Inc, 1996.

[8] S. Zhao; D.M. Akos. An open source gps/gnss vector tracking loop - implementation, filter tuning, and results. In Proceedings of the 2011 International Technical Meeting of The Institute of Navigation, pages 1293–1305, San Diego, CA, USA, January 2011

[9] E.D. Kaplan; C.J. Hegarty. Understanding GPS. Principles and applications. Second Edition. Artech House, 2006.