Comparative Performance Analysis of Scalar and Vector GNSS Receiver Architectures

António Pedro Bretes Negrinho
antonio.negrinho@tecnico.ulisboa.pt

Instituto Superior Técnico, Lisboa, Portugal

November 2019

Abstract

This work addresses the performance comparison between scalar and vector architectures for GNSS receivers, regarding mainly code tracking. First, a review on the important literature is undertaken, alongside a mathematical description of the code tracking process and multipath effect. The main differences between scalar and vector receiver architectures are also highlighted. Two navigation estimation algorithms are described: the least-squares algorithm and the extended Kalman filter (EKF), as well as their integration in scalar and vector receivers. All the receiver architectures, and algorithms are implemented in MATLAB and simulated in a variety of signal reception scenarios and dynamics: pedestrian, car and aircraft. Different signal modulations are utilized: BPSK, BOCs(1,1) and CBOC(6,1,1/11), these last two being used by the Galileo system. Additionally, a new code discriminator based on a bank of correlators was devised for processing of the CBOC(6,1,1/11), which was not possible with the traditional one. Simulation results demonstrate that vector architectures outperform scalar ones in almost all cases, especially in scenarios of severe attenuation, regardless of the estimation algorithm used. Additionally, a vector algorithm, with simultaneous code tracking and frequency tracking, was implemented with some performance results displayed.
Keywords: Vector tracking, Scalar tracking, VDFLL, VDLL, GNSS

1. Introduction

Originally devised for military applications, the Global Positioning System (GPS) was the first global navigation satellite system (GNSS) to be put in operation by the United States of America in the 1970's [1], [2]. Because of the widespread usage of GNSS, its shortcomings must be tackled in order to ensure that these type of systems are robust and trustworthy so that other services and activities can rely on them. The first shortcoming is the power of the received signals, which can be as low as 10^{-16} watts [3]. This leads to easy interference of signal reception, either intentional or not, implying receivers must be able to operate with very attenuated signals. Another problem in signal reception is the blockage of incoming signals by buildings, trees or any other sort of object, natural or man-made. A GNSS receiver needs to maintain line-of-sight with the satellites whose signals it is tracking and the loss of one or more incoming signals negatively affects the performance of a receiver in a drastic way. This prolonged loss of signal tracking will usually lead receivers to re-do the time consuming task of re-acquiring the lost signals [2], [4]. Signal reception can also be affected by other phenomena such as ionospheric scintillation, multipath or high receiver dynamics, all of which degrade the receiver's performance.

A method to overcome these hindrances in signal reception is the utilization of vector architectures in GNSS receivers [1], [4]. There is some literature that shows vector receivers have been able to carry out the task of tracking severely attenuated signals, high receiver dynamics and recover from the momentary blockage of incoming signals [5], [6], [7], [8].

1.1. Objectives

The main objective of this document is the performance comparison between traditional (scalar) and vector receiver architectures for satellite navigation, in several simulation scenarios and using different navigation signals, where the code tracking operation was implemented and the frequency tracking operation was considered perfect. For a vector receiver, this type of architecture is called the Vector Delay Lock Loop (VDLL), where we only consider the code tracking operation as a vector structure. In the final part of the document, the condition of perfect frequency tracking was relaxed, by implementing an architecture - the Vector De-

lay/Frequency Lock Loop (VDFLL) - where both the code and carrier were tracked. The proposed objectives of this document are now stated: Generate a satellite constellation; Generate both GPS and Galileo navigation signals; Implement a scalar and vector receiver architecture for code tracking, with two different estimation algorithms; Devise several simulation scenarios to test the receiver architectures; Compare the obtained results between both the scalar and vector architectures and draw conclusions on that comparison; and implement and obtain some results for the VDFLL. All these tasks were carried out using the MATLAB software for implementation, testing and validation.

2. Theoretical Background

2.1. Code Tracking

Figure 1 describes the tracking part of a conventional GNSS receiver. It is constituted by a carrier frequency wipe-off block, three complex correlators (E = early, P = prompt and L = late), being complemented with a code and carrier discriminator. The synchronization errors generated at the discriminators feed low-pass filters (LPF) which provide feedback signals to the carrier and spreading signal numerically-controlled oscillators (NCO).

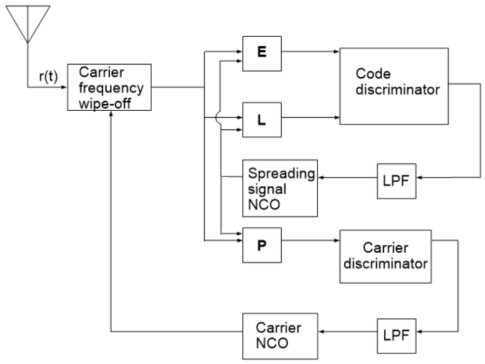


Figure 1: Block diagram of one channel of the conventional GNSS receiver.

The early and late correlators, alongside the code discriminator, are responsible for tracking the incoming signal's code delay - code tracking - forming a feedback structure called the Delay Lock Loop (DLL). On the other hand, the prompt correlator and the carrier discriminator form a feedback structure, through the carrier frequency wipe-off block, responsible for tracking the Doppler frequency shift of the incoming signal - carrier tracking . This structure is called a Frequency Lock Loop (FLL) if the carrier discriminator output depends on the frequency error or a phase lock loop (PLL) if it depends on the phase error. Herein we analyze the code tracking process, assuming frequency tracking is carried out perfectly. Let us consider the conventional GNSS receiver is fed by an incoming signal, generated from N satellites in view. The received signal is of the following type:

$$r(t) = \sum_{i=1}^{N} A_i X_i(t) \cos(\omega_0 t + \omega_{d_i} t + \theta_i) + w(t) \quad (1)$$

with A_i and $X_i(t)$ being, respectively, the amplitude of satellite n and encoded data signal; ω_0 and ω_{d_n} being, respectively, the carrier wave frequency and the Doppler frequency shift, in rad/s, and $w(t)$ being *Additive White Gaussian Noise* (AWGN), with power spectral density of $G_w(f) = N_0/2$. Equation (1) can be split into its inphase and quadrature components, after low-pass filtering:

$$\begin{bmatrix} r_I(t) \\ r_Q(t) \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{N} A_i X_i(t) \cos(\omega_{d_i} t + \theta_i) \\ \sum_{i=1}^{N} A_i X_i(t) \sin(\omega_{d_i} t + \theta_i) \end{bmatrix} + \begin{bmatrix} n_I(t) \\ n_Q(t) \end{bmatrix} \quad (2)$$

where $n_I(t)$ and $n_Q(t)$ are low-pass Gaussian white noises, with power spectral densities $G_{n_I} = G_{n_Q} = N_0 \Pi(f/2B)$, with B representing the bandwidth of the spreading signal $X_i(t)$ and $\Pi(f/2B)$ denoting a rectangle of duration $2B$ centered at the origin.

Assuming we can erase the ω_d component from (1), as we consider perfect frequency tracking, we can correlate, after low-pass filtering, signal n (with $n \in [1, \ldots, N]$) from $r(t)$ with two locally (in the receiver) generated copies of the spreading signal, one advanced, $X_n(t + \Delta/2 - \epsilon_n)$, and one delayed, $X_n(t - \Delta/2 - \epsilon_n)$, where Δ represents the DLL early-late spacing and ϵ_n represents the delay between the locally-generated spreading signal replica and the spreading signal of the incoming signal. Let us analyze, for example, the early inphase correlator output:

$$\begin{aligned} IE_n &= \frac{1}{T} \int_0^T \tilde{r}_{I_n}(t)\, X_n \left(t + \frac{\Delta}{2} - \epsilon_n \right) dt \\ &= A_n R_X \left(\epsilon_n - \frac{\Delta}{2} \right) \cos(\theta_n) + N_{IE_n} \end{aligned} \quad (3)$$

where R_X is the autocorrelation function of the spreading code signal $X_n(t)$ and T is the correlation interval. The noise component of (3) can be described as:

$$N_{IE_n} = \frac{1}{T} \int_0^T \tilde{n}_{I_n}(t)\, X_n \left(t + \frac{\Delta}{2} - \epsilon_n \right) dt \quad (4)$$

with zero mean and variance $\sigma^2 = \frac{N_0}{T}$. The other correlator outputs are:

$$\begin{aligned} IL_n &= A_n R_X \left(\epsilon_n + \frac{\Delta}{2} \right) \cos(\theta_n) + N_{IL_n} \\ QE_n &= A_n R_X \left(\epsilon_n - \frac{\Delta}{2} \right) \sin(\theta_n) + N_{QE_n} \\ QL_n &= A_n R_X \left(\epsilon_n + \frac{\Delta}{2} \right) \sin(\theta_n) + N_{QL_n} \end{aligned} \quad (5)$$

The inphase components $[N_{IE_n}\ N_{IL_n}]^T$ and quadrature components $[N_{QE_n}\ N_{QL_n}]^T$ are independent from one another. However, the elements of each of those vectors are correlated with $E\{N_{IE_n}\ N_{IL_n}\} = \frac{N_0}{T}R_X(\Delta)$ and $E\{N_{IE_n}\ N_{IL_n}\} = E\{N_{QE_n}\ N_{QL_n}\}$.

Let us now define function $D_c(\epsilon_k)$ as the response of a normalized non-coherent early-late power discriminator (NELP), with the following expression:

$$D_c(\epsilon_n) = \frac{(IE_n^2 + QE_n^2) - (IL_n^2 + QL_n^2)}{IE_n^2 + QE_n^2 + IL_n^2 + QL_n^2} \quad (6)$$

By using a normalized non-coherent discriminator, the output will be independent of the carrier phase θ and amplitude A_n.

The DLL's purpose is to provide an estimate $\hat{\tau}_{\text{code}}$ of the delay τ_{code} of the spreading signal $X(t)$. As such, the DLL's estimate of the code delay is updated, at instant $(k+1)T$, for each satellite, through:

$$\hat{\tau}_{\text{code}_n}(k+1) = \hat{\tau}_{\text{code}_n}(k) - \gamma_c D_c(\epsilon_n) \quad (7)$$

where γ_c represents the DLL's code discriminator gain.

Code tracking is an essential task of a GNSS receiver as the incoming signal's code delay, from satellite i, is related to the pseudorange between them through the following relation:

$$\rho_i = c \cdot \tau_{\text{code}_i} \quad (8)$$

where ρ_i represents the pseudorange between a satellite i and the receiver and c represents the speed of light in vacuum. We can define, in the ECEF frame, the pseudorange ρ_i between satellite i with coordinates (X_i, Y_i, Z_i) and a receiver with coordinates (x, y, z) near the Earth's surface as:

$$\rho_i = \sqrt{(X_i - x)^2 + (Y_i - y)^2 + (Z_i - z)^2} + u \quad (9)$$

where $u = c \cdot t_u$, with c being the speed of light in vacuum ($3 \times 10^8 m/s$) and t_u represents a relative clock error, where $u < 0$ corresponds to having a retarded receiver clock, regarding the satellite clock, and $u > 0$ corresponds to the opposite case, where the receiver clock is advanced relative to the satellite clock.

2.2. Multipath

A multipath scenario occurs when, besides the signal in line-of-sight (LOS), replicas of that same signal reach a GNSS receiver, with a delay in time τ. Thus, considering that only one of these replicas reaches the receiver, the received signal can be expressed as:

$$\begin{aligned} r(t) =& AX(t)\cos(\omega_0 t + \theta) \\ &+ \alpha A X(t-\tau)\cos(\omega_0 t + \theta + \phi) + \omega(t) \end{aligned} \quad (10)$$

where $X(t)$ represents the encoded data signal, τ represents the additional delay of the reflected ray, ϕ represents a phase offset due to the extra delay and α represents the attenuation of the multipath signal. If we consider, for simplicity's sake a non-normalized NELP discriminator (just the numerator from eq. (6)), we can define the multipath envelopes, which represent the worst case scenarios for measured pseudorange errors, as:

$$\begin{aligned} &R_X\left(\epsilon - \frac{\Delta}{2}\right) - R_X\left(\epsilon + \frac{\Delta}{2}\right) \\ &= \pm\alpha\left[R_X\left(\epsilon - \tau - \frac{\Delta}{2}\right) - R_X\left(\epsilon - \tau + \frac{\Delta}{2}\right)\right] \end{aligned} \quad (11)$$

Equation (11) represents the solution for obtaining the multipath envelopes, which can be seen below, in figure 2, for the GPS C/A signal and the Galileo BOCs(1,1) and CBOC(6,1,1/11) pilot signals, using a NELP discriminator, with $\Delta/T_c = 0.1$ (T_c represents the spreading code's chip time) and $\alpha = 0.5$:

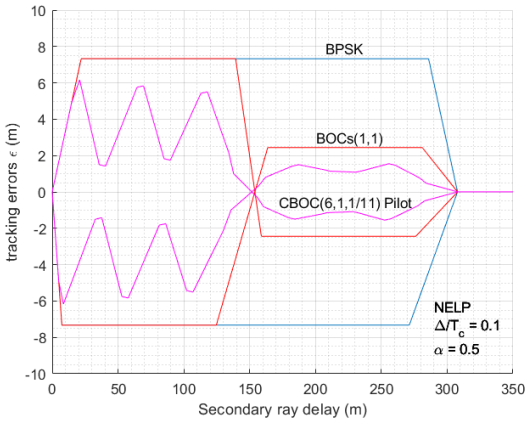


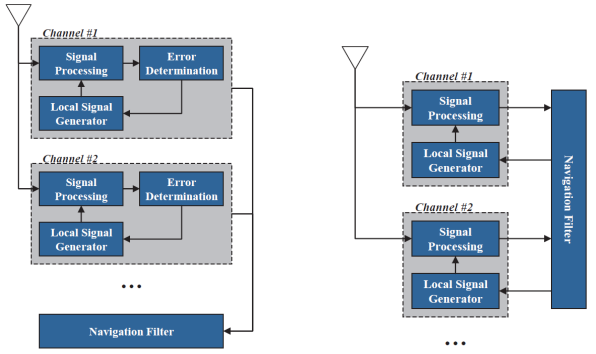
Figure 2: Multipath envelopes for several signals.

Notice that for small secondary ray delays (< 20m) the various solutions in figure 2 are equivalent. However, for longer delays, the BOCs(1,1) and CBOC(6,1,1/11) pilot signals exhibit better performance. This is especially true for delays larger than approximately 150 meters.

2.3. Vector tracking architectures

The traditional GNSS receiver architecture, a scalar tracking architecture, tracks the received GNSS signals (with four being the needed minimum) independently from one another, using a set of tracking loops, for code and Doppler phase or frequency. These tracking loops then feed an estimation algorithm, in order to obtain an estimate of the receiver's position, velocity, and time (PVT). Vector tracking architectures are an advanced and more complex form of processing the several GNSS signals that reach a given receiver. They are capable

of operating at lower carrier-to-noise ratios (C/N_0) and tolerating higher receiver dynamics. The vector architecture, contrarily to a traditional scalar architecture, processes all signals jointly through one big loop of code and frequency discriminators, coupled through an estimation algorithm (Least-Squares, extended Kalman filter, etc.) [1], [4], [9]. Figure 3 illustrates the differences between the scalar and vector architectures.



(a) Basic scalar tracking architecture

(b) Basic vector tracking architecture

Figure 3: Comparison of the scalar and vector tracking architectures [5].

Vector receivers can present three types of architecture [1], [4]: the Vector Delay Lock Loop (VDLL), where the code tracking portion of the receiver forms a vector structure coupled with the estimation algorithm, while the carrier tracking portion is separated as in the scalar architecture; the Vector Frequency Lock Loop (VFLL), with the carrier tracking portion being coupled with the estimation algorithm and the code tracking part being separated; and, finally, the Vector Delay/Frequency Lock Loop (VDFLL) where both the code and carrier tracking portions of the receiver are coupled, through an estimation algorithm, in a single feedback loop.

3. Scalar and Vector Receiver Algorithms Analysis

3.1. Least-Squares algorithm

Let us start by recalling equation (9), that defines the pseudorange between satellite i and a receiver. Using the linear Taylor series expansion on (9) we obtain the following increments equation:

$$\Delta\rho_i = \frac{\partial\rho_i}{\partial x}\Delta x + \frac{\partial\rho_i}{\partial y}\Delta y + \frac{\partial\rho_i}{\partial z}\Delta z + \frac{\partial\rho_i}{\partial u}\Delta u \quad (12)$$

This equation can also be expressed in matrix form, resulting in:

$$\underbrace{\begin{bmatrix}\Delta\rho_1\\ \vdots\\ \Delta\rho_N\end{bmatrix}}_{\Delta\rho} = \underbrace{\begin{bmatrix}\frac{\partial\rho_1}{\partial x} & \frac{\partial\rho_1}{\partial y} & \frac{\partial\rho_1}{\partial z} & 1\\ \vdots & \vdots & \vdots & \vdots\\ \frac{\partial\rho_N}{\partial x} & \frac{\partial\rho_N}{\partial y} & \frac{\partial\rho_N}{\partial z} & 1\end{bmatrix}}_{G} \cdot \underbrace{\begin{bmatrix}\Delta x\\ \Delta y\\ \Delta z\\ \Delta u\end{bmatrix}}_{\Delta X} \quad (13)$$

with G being the geometry matrix. A general least-squares solution for equation (13) can be written using the pseudo-inverse of G:

$$\Delta X = (G^T G)^{-1} G^T \Delta\rho \quad (14)$$

Equation (14) allows to iteratively solve, for $[x\, y\, z\, u]^T$, the system composed of $N \geq 4$ equations of the type presented in (9). We assume that the $[X_i\, Y_i\, Z_i]^T$ positions of the N satellites are known and that we have an initial estimate $[\hat{x}_0\, \hat{y}_0\, \hat{z}_0\, \hat{u}_0]^T$ for the receiver's state. We also assume that:

$$\Delta\rho = \rho_{measured} - \rho_{estimated} \quad (15)$$

where we can measure $\rho_{measured}$ and $\rho_{estimated}$ is obtained through (9). We then update the estimate of the receiver's state through:

$$[\hat{x}_1\, \hat{y}_1\, \hat{z}_1\, \hat{u}_1]^T = [\hat{x}_0\, \hat{y}_0\, \hat{z}_0\, \hat{u}_0]^T + \underbrace{[\Delta x\, \Delta y\, \Delta x\, \Delta u]}_{\Delta X} \quad (16)$$

In a scalar receiver, the least-squares algorithm described earlier is an "off the shelf" solution as it can be used immediately, receiving the measured pseudoranges ρ_i from the several DLLs assigned to track each satellite in use.

In a vector receiver, the least-squares algorithm is fed with a pseudorange error vector $\Delta\rho$ that results from subtracting a vector of measured pseudoranges, obtained from a series of code discriminators, from a vector of predicted pseudoranges, that are initialized on the first iteration of the algorithm. This $\Delta\rho$ vector is related to the vector of position errors ΔX through the following relation:

$$\Delta\rho = G\,\Delta X \quad (17)$$

with G being the geometry matrix presented in (13).

The pseudorange errors can be expressed, in the linearity region, through the code discriminator outputs multiplied by a matrix of discriminator gains:

$$\Delta\rho = \Gamma \begin{bmatrix}D(\epsilon_1)\\ \vdots\\ D(\epsilon_N)\end{bmatrix} \quad (18)$$

with Γ being the diagonal matrix of N code discriminator gains: We can now write the least-squares

solution of (17) as:

$$\Delta X = (G^T G)^{-1} G^T \Gamma \begin{bmatrix} D(\epsilon_1) \\ \vdots \\ D(\epsilon_N) \end{bmatrix} \tag{19}$$

using ΔX, at instant k, to update the estimate for the position vector of the receiver, in instant $k+1$:

$$\hat{X}(k+1|k) = \hat{X}(k|k-1) - \Delta X(k) \tag{20}$$

Now, with $\hat{X}(k+1|k)$ obtained, it is possible to create a predicted pseudorange between satellite S_i and the receiver at instant $k+1$, assuming the position for satellite S_i, at that instant, is known $(X_i(k+1), Y_i(k+1), Z_i(k+1))$, using equation (9).

These predicted pseudoranges $\hat{\rho}_i(k+1|k)_{(i=1,\dots,N)}$ will be fed back to the code discriminators, in order to be compared with the pseudoranges measured by them. That comparison will result in a new $\Delta\rho$, beginning the explained loop anew. In a vector receiver, we cannot separate the code tracking loops from the estimation algorithm, as they are coupled, forming a structure called the VDLL.

3.2. Extended Kalman filter algorithm

The Extended Kalman filter is characterized by two steps, at each iteration (instant k in time): a prediction step and a filtering step [10]. It can be shown that the filtering step equations are [10]:

$$\begin{aligned} \hat{x}(k|k) =& \hat{x}(k|k-1) + K_k \left(z_k - h\left[\hat{x}(k|k-1)\right]\right) \\ P(k|k) =& \left[I - K_k H_k\right] P(k|k-1) \left[I - K_k H_k\right]^T \\ &+ K_k \widetilde{R}_k K_k^T \end{aligned} \tag{21}$$

where $\hat{x}(k|k)$ represents the filtered estimate of x at instant k, $\hat{x}(k|k-1)$ represents the predicted estimate of x at instant k, and $P(k|k-1)$ and $P(k|k)$ represent respectively the prediction and filtering error covariance matrices. The Kalman gain K_k and can be computed as:

$$K_k = P(k|k-1) H_k^T \left(H_k P(k|k-1) H_k^T + \widetilde{R}_k\right)^{-1} \tag{22}$$

According to [10], the prediction step equations can be expressed as:

$$\begin{aligned} \hat{x}(k+1|k) &= \Phi_k \hat{x}(k|k) \\ P(k+1|k) &= \Phi_k P(k|k) \Phi_k^T + Q_k \end{aligned} \tag{23}$$

The dynamics model utilized was the PVT model (position, velocity and time), where we have a state vector x_k with eight components: three for position in the x, y, z axes plus three for the respective velocities $\dot{x}$, $\dot{y}$, $\dot{z}$ and two for the receiver's clock state [11]. We define the state vector, at instant k, as $[x_{1,k}, \cdots, x_{8,k}]^T$, where $x_{1,k} = x_{u,k}$, $x_{2,k} = \dot{x}_{u,k}$, $x_{3,k} = y_{u,k}$, $x_{4,k} = \dot{y}_{u,k}$, $x_{5,k} = z_{u,k}$, $x_{6,k} = \dot{z}_{u,k}$, $x_{7,k} = c\,x_{\phi,k} = u_{u,k}$ and $x_{8,k} = c\,x_{f,k} = \dot{u}_{u,k}$.

Let us define the discrete-time state model corresponding to the receiver coordinate x_u:

$$\begin{bmatrix} \dot{x}_{1,k+1} \\ \dot{x}_{2,k+1} \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix}}_{a} \begin{bmatrix} x_{1,k} \\ x_{2,k} \end{bmatrix} + \begin{bmatrix} u_{1,k} \\ u_{2,k} \end{bmatrix} \tag{24}$$

where $\Delta t = T$ (correlation interval). We determine the dynamics noise covariance matrix as:

$$Q'_k = E\{[u_{1,k}\, u_{2,k}]^T\, [u_{1,k}\, u_{2,k}]\} = q_v \Delta t \begin{bmatrix} \frac{(\Delta t)^2}{3} & \frac{\Delta t}{2} \\ \frac{\Delta t}{2} & 1 \end{bmatrix} \tag{25}$$

For the y and z coordinates, the discrete-time state model is the same. The receiver's discrete-time clock state model is described as:

$$\begin{bmatrix} \dot{x}_\phi(t) \\ \dot{x}_f(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x_\phi(t) \\ x_f(t) \end{bmatrix} + \begin{bmatrix} u_\phi(t) \\ u_f(t) \end{bmatrix} \tag{26}$$

and the corresponding discrete-time version

$$\begin{bmatrix} x_{\phi,\,k+1} \\ x_{f,\,k+1} \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix}}_{A} \begin{bmatrix} x_{\phi,\,k} \\ x_{f,\,k} \end{bmatrix} + \begin{bmatrix} u_{\phi,\,k} \\ u_{f,\,k} \end{bmatrix} \tag{27}$$

The clock model noise covariance matrix, in discrete time, can be expressed as:

$$\widetilde{Q}_k = \begin{bmatrix} q_\phi \Delta t + \frac{q_f (\Delta t)^3}{3} & \frac{q_f (\Delta t)^2}{2} \\ \frac{q_f (\Delta t)^2}{2} & q_f \Delta t \end{bmatrix} \tag{28}$$

The variances q_ϕ and q_f can be described as [11]:

$$\begin{aligned} q_\phi &\approx \frac{h_0}{2} \\ q_f &\approx 2\pi^2 h_{-2} \end{aligned} \tag{29}$$

with h_0 and h_{-2} being the Allan variance parameters. In [11], several values for the Allan variance parameters can be found. When treating clock deviations as errors in meters, the Allan variance parameters must be multiplied by the speed of light squared ($c^2 = 9 \times 10^{16}$).

The state transition matrix is $\Phi = \mathrm{diag}\{a\, a\, a\, A\}$ and the dynamics noise covariance matrix is $Q_k = \mathrm{diag}\{Q'_k\, Q'_k\, Q'_k\, \widetilde{Q}_k\}$. We can express the observations matrix H_k as:

$$H_k = \left[\frac{\partial h_i[\hat{x}(k|k-1)]}{\partial x_j}\right]_{N \times l} \tag{30}$$

we consider $l = 8$ because we are using the PVT model and $h_{x_k} = [\rho_1, \ldots, \rho_N]$. The observation's noise covariance matrix is expressed as $\widetilde{R}_k = \text{diag}\{\sigma^2_{1,UERE}, \ldots, \sigma^2_{N,UERE}\}$.

In a scalar receiver the EKF can be used in an "off the shelf" approach. It just needs to be fed with a set of pseudoranges, at each t_k iteration from a set of DLL's.

For the vector receiver, we will follow an adaptation of the model first proposed in [12] for a VDFLL to describe the adaptation of the EKF to a vector structure, as a VDLL (assuming perfect carrier tracking). We now describe the innovations process through the outputs of the several code discriminators:

$$z_k - h\left[\hat{x}(k|k-1)\right] = \Gamma \begin{bmatrix} D_{c_1} \\ \vdots \\ D_{c_N} \end{bmatrix} \tag{31}$$

where $[D_{c_1}, \cdots, D_{c_N}]^T$ is the vector containing the outputs of the code discriminators and Γ represents a diagonal matrix of N code discriminator gains. The right-hand members of equation (31) constitute the innovations process $z_k - h\left[\hat{x}(k|k-1)\right]$ in the update of the state vector in (21).

Another difference, regarding the EKF's implementation in a scalar receiver, is the way the R_k matrix is defined. In a vector receiver, with EKF, the matrix $\widetilde{R}_k$ is diagonal with [12]:

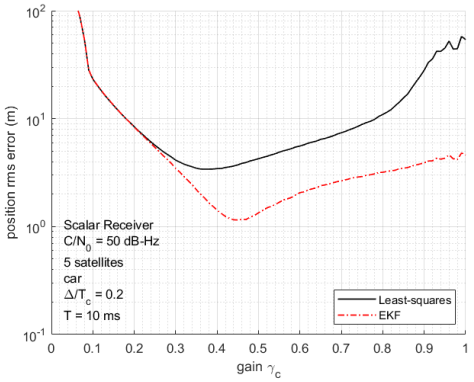
$$r_{ii} = -\frac{c^2 \Delta}{2p\left(\frac{C}{N_0}\right)_i T}, \qquad i = 1, \cdots, N \tag{32}$$

where $p = -\frac{1}{T_c}$ for BPSK signals, $p = -\frac{3}{T_c}$ for BOCs(1,1) signals and $p = -\frac{(53+2\sqrt{10})}{11T_c}$ for CBOC(6,1,1/11). T_c is the chip time of the respective signal.

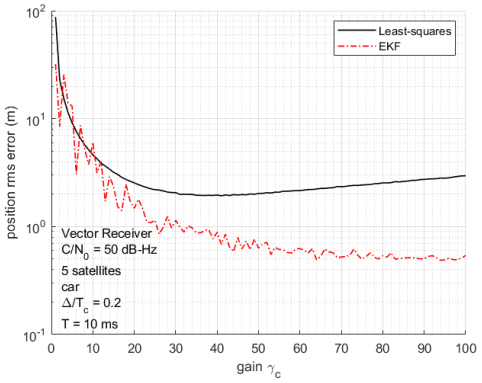
The feedback process in the vector receiver consists of computing $\hat{x}(k+1|k)$, utilizing the filtered estimate $\hat{x}(k|k)$, and rebuild a set of estimated pseudoranges, at the instant $k+1$, using equation (9), to be compared with the measured ones, through the code discriminator. We call the difference between the measured and predicted pseudoranges, the residuals and they are expressed through the code discriminator outputs D_{c_i}, with $i \in [1, \ldots, N]$.

4. Scalar and vector receiver simulation results

The simulations herein presented consider a GPS constellation and BPSK signals. A code discriminator integration time of $T = 10$ms and $\Delta/T_c = 0.2$ were also considered, disregarding ionospheric and troposhperic errors and assuming infinite bandwidth. The receiver was initially located at the IST Alameda campus on 20/02/2019 and presented $v_x = v_y = v_z = 20\,\text{m/s}$ in the ECEF coordinate system ($\approx 125\,\text{Km/h}$). A simulation run i corresponding to an interval of 100 s was considered for each value of DLL gain γ_c, with the position root mean square (rms) error being calculated for each run. In figure 4 we can observe the results for both the scalar and vector receiver with least-squares and the EKF algorithms, obtained with a constellation of 5 satellites with carrier-to-noise ratio $C/N_0 = 50$ dB-Hz and shared γ_c. We call this simulation scenario the standard scenario.



(a) Scalar receiver



(b) Vector receiver

Figure 4: Performance comparison between a scalar and a vector receiver.

The minimum rms errors registered, for the scalar receiver, were of $\approx 3\,\text{m}$ with the least-squares and $\approx 1\,\text{m}$ for the EKF. For the vector receiver, the minimum rms errors were of 2 m and $\approx 0.5\,\text{m}$ for the least-squares and EKF, respectively. The first conclusion is that for the same receiver architecture the EKF outperforms the least-squares. The vector receiver with the EKF presents the best results, whereas the worst are presented by the scalar receiver with the least-squares. Also, considering the same estimation algorithm, the vector receiver presents better results than the scalar one. We can conclude, therefore, that in terms of receiver architecture, the vector architecture presents better

results than the scalar one.

Another simulation performed was considering the previous scenario but now with a constellation of 4 satellites where, for a simulation interval of 10 minutes, we severely degrade the C/N_0 of the received signal of one of the satellites for 5 minutes. We call this scenario the shadowing scenario, since the receiver is shadowed regarding one of the satellites. In this scenario, we kept a constant optimal value of γ_c and measured the instantaneous position vector error. The aim of this simulation is to measure how much attenuation on one of its received signals can the receiver tolerate before it loses tracking capability. Tracking capability was considered to be lost when the instantaneous position vector error would not return to the values prior to the shadowing period, after it ended. The results obtained are now presented in table format, in table 1, as to avoid presenting a large amount of plots.

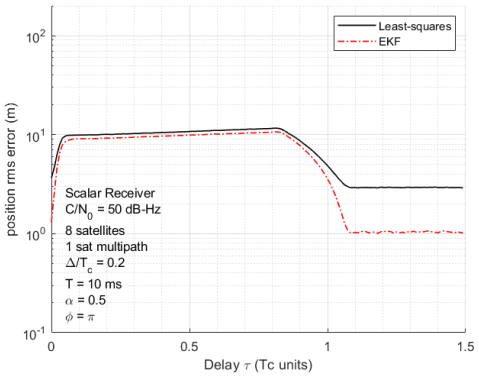
	Max. Attenuation (dB)	
	Scalar	Vector
Least-squares	20	30
EKF	20	30

Table 1: Scalar vs Vector receiver performance in a shadowing scenario (EKF).

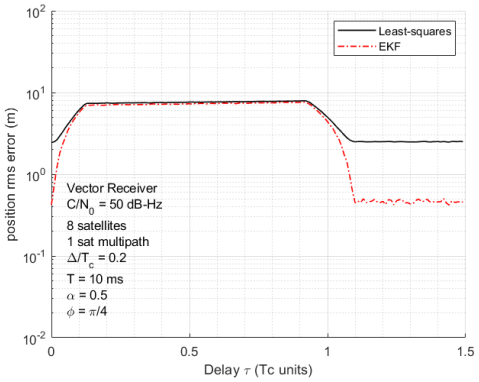
From table 1 we can observe that, firstly, the chosen estimation algorithm does not influence the obtained results: only the receiver architecture is important. The vector receiver is clearly better than the scalar one, withstanding 10 dB more of attenuation before losing tracking. This translates, in linear units, in the vector receiver being able to function with a C/N_0 ten times lower than the one registered for the scalar one. Therefore, we can infer that the vector architecture is more robust to low signal power than the scalar one.

Finally, we considered the receiver to be utilizing a constellation of 8 satellites, where one of them was subject to the effects of multipath. We considered an attenuation of the reflected ray $\alpha = 0.5$ and a phase offset of $\phi = \pi$, for the scalar receiver and $\phi = \pi/4$ for the vector receiver. These phase offset values proved to present the worse case scenarios for this simulation. The results for this multipath scenario are displayed in figure 5.

For the scalar receiver there is a small variation in maximum rms, with it being 10 m for the EKF and 12 m for the least-squares. For the vector receiver, we can see that there are practically no changes in the obtained results due to the chosen estimation algorithm, with the maximum rms error registered being 8 m. As such, it can be observed that the vector receiver once more outperforms the scalar one



(a) Scalar receiver



(b) Vector receiver

Figure 5: Performance comparison between a scalar and a vector receiver in multipath.

(regardless of the estimation algorithm chosen), being more robust to multipath, despite the overall poor performance results obtained from this simulation. Also, it should be noted that for both receivers, for $\tau \geq 1.1\,T_c$ the effects of multipath no longer influence the obtained results.

A performance analysis regarding the utilized modulation was also considered. We simulated the BPSK signal, the BOCs(1,1) signal and the CBOC(6,1,1/11) pilot signal, utilized in the Galileo system. For this last signal, a new code discriminator based on a bank of correlators was devised as the NELP discriminator was prone to false code lock. The response of the bank of correlators is described as:

$$D_c(\epsilon) = \frac{\sum_{i=-M}^{M} \mu_i \left[\left(I_n^{(i)}\right)^2 + \left(Q_n^{(i)}\right)^2 \right]}{\sum_{i=-M}^{M} \left(I_n^{(i)}\right)^2 + \left(Q_n^{(i)}\right)^2} \tag{33}$$

with

$$\begin{aligned} I_n^{(i)} &= A_n R_X \left(\epsilon_n - \frac{iT_c}{M}\right) \cos(\theta_n) + N_{I_n}^{(i)} \\ Q_n^{(i)} &= A_n R_X \left(\epsilon_n - \frac{iT_c}{M}\right) \sin(\theta_n) + N_{Q_n}^{(i)} \end{aligned} \tag{34}$$

where we consider $2M+1$ correlators and μ_i are the weight coefficients of each correlator. These weight coefficients can be computed considering equation (33) in the absence of noise:

$$D_c(\epsilon) = \frac{1}{S(\epsilon)} \sum_{i=-M}^{M} \mu_i R_X^2 \left(\epsilon - \frac{iT_c}{M}\right) \tag{35}$$

with

$$S(\epsilon) = \sum_{i=-M}^{M} R_X^2 \left(\epsilon - \frac{iT_c}{M}\right) \tag{36}$$

We consider a set of $\epsilon_j = \epsilon_1, \ldots, \epsilon_J$ delays and solve equation (35) for the several μ_i, also considering $D_c(\epsilon) = \widetilde{D}_c(\epsilon)$ where $\widetilde{D}_c(\epsilon)$ is a desired ideal response for the bank of correlators.

Below in tables 2 through 5 we observe the obtained results, where we considered both receiver architectures and both estimation algorithms.

Scalar	Min. rms error (m)	
Least-squares	NELP	Bank
BPSK	≈ 3.5	4
BOCs(1,1)	2	$\approx$ 2.5
CBOC(6,1,1/11)	≈ 6	≈ 2.5

Table 2: Scalar receiver performance for different modulations (least-squares).

Scalar	Min. rms error (m)	
EKF	NELP	Bank
BPSK	≈ 1	≈ 1.5
BOCs(1,1)	$\approx$ 0.7	$\approx$ 0.9
CBOC(6,1,1/11)	≈ 2.5	0.9

Table 3: Scalar receiver performance for different modulations (EKF).

Vector	Min. rms error (m)	
Least-squares	NELP	Bank
BPSK	0.7	0.8
BOCs(1,1)	0.5	0.6
CBOC(6,1,1/11)	≈ 2.5	0.6

Table 4: Vector receiver performance for different modulations (least-squares).

The first conclusion to be drawn is that the bank of correlators is only useful for the CBOC(6,1,1/11) signal, presenting worse results for the BPSK signal. On the other hand, the NELP is not recommended for reception of the CBOC(6,1,1/11) signal.

Vector	Min. rms error (m)	
EKF	NELP	Bank
BPSK	0.5	0.6
BOCs(1,1)	0.3	0.4
CBOC(6,1,1/11)	3	0.4

Table 5: Vector receiver performance for different modulations (EKF).

For both receiver architectures we observe considerable improvements in performance from utilizing the EKF instead of the least-squares, with the exception of the vector receiver plus the NELP discriminator, for the CBOC(6,1,1/11). We can, however, disregard this exception as the NELP discriminator is not intended for the reception of the CBOC(6,1,1/11) signal.

It is interesting to observe that for a vector receiver and a bank of correlators (regarless of estimation algorithm) we obtain better performance from a GPS C/A signal than with the same signal for a scalar receiver and the optimal code discriminator for this signal: the NELP. This observation further attests to the benefits of using a vector architecture. It can be assumed that the same situation (vector receiver with sub-optimal discriminator outperforming a scalar receiver with the optimal one) does not happen for the CBOC(6,1,1/11) as it is a much more complex signal, demanding a more specific discriminator block.

Regarding the results obtained for the BOCs(1,1) signal, we observe that the performance registered with this signal increases if a vector architecture is utilized, instead of a scalar one. Contrarily to what was observed for the CBOC(6,1,1/11), the BOCs(1,1) signal presents better results when a NELP discriminator is being used. These results can be somewhat misleading, as using this modulation with the NELP discriminator can lead to false-code lock. As such, the bank of correlators presents itself as an alternative where that phenomenon does not occur and with very little difference in the minimum rms error values achieved, in comparison to the NELP. On a last note, it should be acknowledged that even if BOCs(1,1) and CBOC(6,1,1/11) present the same performance results with the ban of correlators (regardless of architecture and estimation algorithm), the latter presents more robustness to multipath as demonstrated through the multipath envelopes in figure 2.

5. Vector delay/frequency lock loop characterization and results

5.1. VDFLL characterization

Let us now consider a vector receiver in which we also implement the carrier tracking part, instead

of considering perfect, as in 2. As such, besides the early and late correlators and the NELP discriminator decribed in section 2, we now consider inphase and quadrature prompt correlators IP and QP, which correlate the incoming signal with a locally generated centered replica $X_n(t-\epsilon_n)$. Also, since we no longer consider perfect Doppler wipe-off, new terms appear in equation (3), becoming:

$$\begin{aligned} IE_n =& A_n R_X \left(\epsilon_n - \frac{\Delta}{2}\right) \cos\left(\frac{\tilde{\omega}_n T}{2} + \theta_n\right) \\ & \cdot \operatorname{sinc}\left(\tilde{f}_n T\right) + N_{IE_n} \end{aligned} \tag{37}$$

where $\tilde{\omega}_n = (\omega_{d_n} - \omega_e) - (\hat{\omega}_{d_n} - \hat{\omega}_e)$ with ω_{d_n} and $\hat{\omega}_{d_n}$ being the true and estimated Doppler frequencies and ω_e and $\hat{\omega}_e$ being the true and estimated oscillator frequency errors. Also, $\tilde{f}_n = \tilde{\omega}_n/2\pi$. The IP_n, IL_n, QE_n, QP_n and QL_n correlator outputs are described analogously to (37). We define the normalized code discriminator $D_f(\tilde{\omega}_n)$ as:

$$D_f(\tilde{\omega}_n) = \frac{IP_n(k-1)QP_n(k) - IP_n(k)QP_n(k-1)}{IP_n(k)^2 + QP_n(k)^2} \tag{38}$$

Notice that the frequency discriminator function depends on signals generated at two different instants [2]: k and $k-1$.

Considering a EKF is in use, the innovations process is now described as:

$$z_k - h\left[\hat{x}(k|k-1)\right] = \widetilde{\Gamma} \begin{bmatrix} D_{c_1} \\ \vdots \\ D_{c_N} \\ D_{f_1} \\ \vdots \\ D_{f_N} \end{bmatrix} \tag{39}$$

The $\widetilde{\Gamma}$ diagonal matrix now contains $2N$ discriminator gains, half for the code correlators and the other half for the frequency ones: $\gamma = \operatorname{diag}\left[\gamma_{c_1}, \ldots, \gamma_{c_N}, \gamma_{f_1}, \ldots, \gamma_{f_N}\right]^T$, where γ_c are the code discriminator gains and γ_f the frequency ones.

The observations matrix is expressed by expanding the matrix in equation (30) doubling its number of rows $(2N \times l)$, as we now have to accommodate the observations derived from the frequency tracking portion of the VDFLL. These observations are the estimates for the Doppler frequency for each of the N satellites' signals. As such, we have $h_{x_k} = \left[\rho_1, \ldots, \rho_N, -2\pi\frac{f_c}{c} \cdot \frac{d}{dt}\rho_1, \ldots, -2\pi\frac{f_c}{c} \cdot \frac{d}{dt}\rho_N\right]$.

Matrix $\widetilde{R}_k$ is now described as [12]:

$$r_{ii} = \begin{cases} -\frac{c^2\Delta}{2p\left(\frac{C}{N_0}\right)_i T}, & i = 1, \ldots, N \\ \frac{1}{\left(\frac{C}{N_0}\right)_i T^3}, & i = N+1, \ldots, 2N \end{cases} \tag{40}$$

The entries of matrix $\widetilde{R}_k$, represented in equation (40), had to be manually adjusted in order to improve the obtained results. Also, the receiver's clock was now modeled as double integrated brownian motion.

5.2. VDFLL simulation results

The VDFLL was simulated with a constellation of 8 satellites, all with $C/N_0 = 50$dB-Hz. The code correlators were assumed to share a common value for their gain γ_c and the same was assumed for the frequency correlators with γ_f. These gain values were swept from 0 to 100, for γ_c and γ_f. We consider a simulation interval of 100 seconds, car dynamics and no ionospheric or tropospheric errors. The obtained results are sketched in figure 6

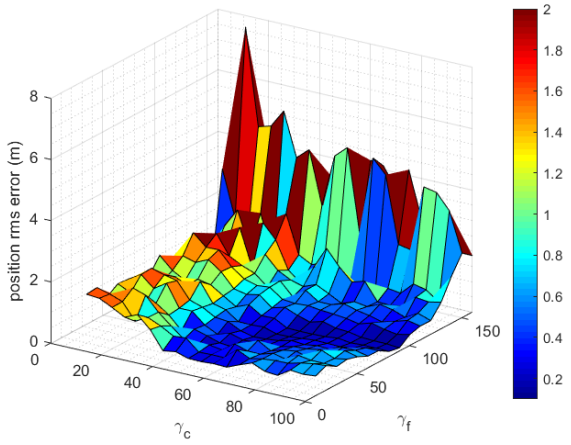


Figure 6: Multipath envelopes for several signals.

We can observe that the results are best in the $\gamma_c > 50$ and $\gamma_f < 130$ region, with performance worsening outside of those boundaries. Within that optimum region, the best results appear to be obtained with $80 \leq \gamma_c \leq 100$ and $\gamma_f = 100$, where the rms error reaches values as low as 0.2 m. We obtain better performance with the VDFLL, than with the VDLL, despite we now simulate a component that was assumed to be perfectly modelled, because we utilized 8 satellites for this simulation, whereas in figure 4 we only utilized 5.

It should be stated that the performance of the VDFLL also greatly depends on other parameters, such as the Q_k and $\widetilde{R}_k$ matrices, with this last one requiring some delicate tuning in order to stabilize the algorithm, as stated previously. The adopted clock model also affects the obtained results, where a poor modelling would render the VDFLL inoperable.

6. Conclusions

In this work we implemented both a scalar and vector GNSS receiver with two different estimation

algorithms - the least-squares and the EKF - assuming perfect carrier tracking. The differences between a scalar and vector receiver were laid out and we analyzed the code thoroughly the code tracking process as well as the effects of multipath on signal reception. The least-squares and EKF algorithms were described, as well as their integration in the two considered receiver architectures.

The standard scenario results show that for the same estimation algorithm, the vector receiver clearly outperforms the scalar one, presenting lower values of rms error. The best performance results were obtained with the vector architecture plus the EKF and the worse with the scalar receiver with plus least-squares. Also, for a given architecture, the EKF always outperformed the least-squares.

In the shadowing scenario, performance could be analyzed regardless of the adopted estimation algorithm. It was shown that the vector receiver was able to sustain much more attenuation to one of the 4 satellites in use, in this simulation, before losing tracking abilities, than the scalar receiver.

For the multipath scenario the results demonstrated that the vector receiver was marginally better than the scalar one, albeit still producing poor positioning results. Utilizing an extended Kalman filter over a least-squares algorithm also slightly improved the results obtained with the scalar receiver, whereas the vector receiver appeared unaffected regarding the choice of estimation algorithm.

Regarding the signal comparison analysis, the vector receiver almost always produced better results than the scalar one, with the NELP discriminator being the best choice for the BPSK signal and the bank of correlators only being useful for the CBOC(6,1,1/11) signal, for which the NELP performs very poorly.

Finally, a thorough description of the VDFLL was presented, analyzing the newly implemented frequency tracking operation and the alterations to be made to the EKF. Some performance results were also presented, regarding the VDFLL's dependence on the feedback gains of the code and frequency tracking loops, with optimal values for those gains being pointed out.

References

[1] M. Lashley. *Modeling and Performance Analysis of GPS Vector Tracking Algorithms*. PhD thesis, Graduate Faculty, Auburn University, Alabama, 2009.

[2] J. W. Betz. *Engineering Satellite-Based Navigation and Timing. Global Navigation Satellite Systems, Signals and Receivers*. John Wiley and Sons, INC., Hoboken, New York, 1^{st} edition, 2016. ISBN:978-1-118-61597-3.

[3] P. Misra and P. Enge. *Global Positioning System - Signals, Measurement and Performance*. Ganga-Jamuna Press, Lincoln, Massachusetts, 2^{nd} edition, 2004. ISBN:0-9709544-1-7.

[4] S. Bhattacharyya. *Performance Integrity Analysis of the Vector Tracking Architecture of GNSS Receivers*. PhD thesis, Graduate School, University of Minnesota, Minnesota, 2012.

[5] M. G. Petovello and G. Lachapelle. Comparison of Vector-Based Software Receiver Implementations With Application to Ultra-Tight GPS/INS Integration. In *Proceedings of ION GNSS 2006*, pages 1790 – 1799, September 2006. Fort Worth, TX.

[6] T. Pany and B. Eissfeler. Use of a Vector Delay Lock Loop Receiver for GNSS Signal Power Analysis in Bad Signal Conditions. In *Proceedings of IEEE/ION Position Location and Navigation Symposium Conference*, pages 893–903, April 2006. Coronado, CA.

[7] D. Benson. Interference Benefits of a Vector Delay Lock Loop (VDLL) GPS Receiver. In *Proceedings of the 63rd Annual Technical Meeting of The Institute of Navigation (2007)*, pages 749 – 756, April 2007. Cambridge, MA.

[8] Jong-Hoon Won, Dominik Dotterbock, and Bernd Eissfeller. Performance Comparison of Different Forms of Kalman Filter Approaches for a Vector-Based GNSS Signal Tracking Loop. *NAVIGATION: Journal of the Institute of Navigation*, 57(3):185 – 199, Fall 2010.

[9] M. Lashley and D. M. Bevly. Vector Delay/Frequency Lock Loop Implementation and Analysis. In *Proceedings of the 2009 International Technical Meeting of The Institute of Navigation*, pages 1073 – 1086, January 2009. Anaheim, CA.

[10] A. Gelb. *Applied Optimal Estimation*. MIT Press, Massachusetts Institute of Technology, Cambridge, Massachusetts, 1^{st} edition, 1974. ISBN:0-262-57048-3.

[11] R. G. Brown and P. Y. C. Hwang. *Introduction to Random Signals and Applied Kalman Filtering*. Wiley, 3^{rd} edition, 1997. ISBN:0-471-12839-2.

[12] F. M. G. Sousa and F. D. Nunes. Characterization and Performance Analysis of a VDFLL GNSS Receiver Architecture. In *NAVITEC 2012*, December 2012. Noordwijk, Netherlands.