



Modelling and Comparison of Tracking Loops for GPS Signal

B.N. Vu and M. Andrlé*

Department of Aerospace Electrical Systems, University of Defence, Czech Republic

The manuscript was received on 8 September 2016 and was accepted after revision for publication on 19 June 2017.

Abstract:

Nowadays, Global Positioning System (GPS) applications are widely spread and affect all aspects of our modern lives. They can be found in many different fields, such as aviation, environment, marine, space, surveying, mapping, and military operations. In military, the GPS is employed in missile guidance, vessels and aircraft navigation, location, communication network timing, etc. The GPS receiver is required to be more and more robust to withstand harsh condition, for example losing signal or jamming. This paper demonstrates the implementation and simulation of an advanced tracking technique that is used in modern GPS receivers – a vector tracking loop. Brief related principles of a GPS receiver are introduced. The principle of traditional tracking loop is shown and compared to the vector tracking loop. An extended Kalman filter is employed as a vector tracking algorithm. The extended Kalman filter uses pseudo range rate, which is computed from both the code and the carrier to constrain the receiver velocity. The acquired results are compared and advantages of the vector tracking algorithm against the traditional tracking loop are presented in this paper, including characteristics of the vector tracking loop.

Keywords:

vector tracking loop, Kalman filter

1. Introduction

A typical GPS receiver must perform at least four tasks in baseband processing to determine its own position. These tasks are satellites acquisition, signal tracking, navigation data decoding and navigation equation calculation. In a conventional GPS receiver, each satellite which is revealed at an acquisition stage is tracked in a single channel using a traditional tracking loop or a scalar tracking loop (STL). In the STL, its phase is derived by mixing and integrating the incoming signal with a local generated signal, while the channels work independently. On the other hand, the vector tracking loop (VTL) couples all tracking the channels into a single algorithm to calculate the phase of

* Corresponding author: Department of Aerospace Electrical Systems, University of Defence, Kounicova 65, 662 10, Brno, Czech Republic.

Phone: +420 973 44 51 74, fax: +420 973 44 52 35, E-mail: milos.andrle@unob.cz

the incoming signal. The advantage of the vector tracking is that the algorithm uses not only information from the signal integration, but also both the position and the velocity of the receiver and the satellites to predict the phase of the incoming signal. In case of one or more satellites being blocked, due to an obstacle or jamming, the algorithm can still track the phase of the signal in that channel by setting information from the other channels – like the position and the velocity of the satellite and the receiver.

2. Scalar Signal Tracking

A signal tracking stage plays an important role in any GPS receiver. At this stage, the navigation data is demodulated, and both the phase of the code and the carrier are extracted. The GPS signal uses CDMA technique and is modulated with a pseudo random noise (PRN code). Thus, successful code phase tracking is the key to open the carrier phase and then the navigation data. Signal tracking generally consists of two tasks: a code tracking and a carrier tracking. The code tracking can be implemented by a delay locked loop (DLL) and the carrier tracking task is implemented by a phase locked loop (PLL) or the Costas loop. The two tracking loops are coupled in one schema as shown in Fig. 1.

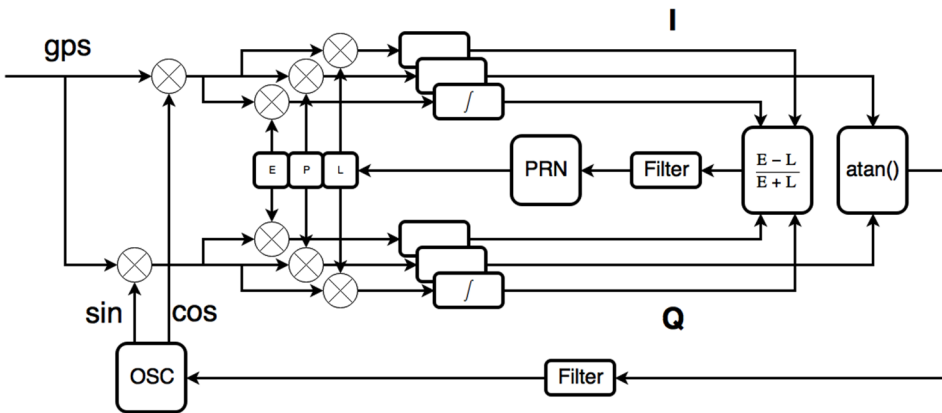


Fig. 1 Code and carrier tracking in one channel [1]

The DLL employs three local PRN codes called early, prompt and late code (*E*, *P*, *L*), whose phase differences are a half chip. The incoming code is multiplied with the early, prompt and late codes and afterwards the results are integrated. Waveforms of codes and multiplication are shown in Fig. 2. The output of the integrator is an auto correlation function. If incoming code and the prompt code are exactly aligned together, the prompt integration reaches maximum values, while the early and late integrations are equal. If the prompt code is faster than the incoming code, the integration increases in an early branch and decreases in a late branch. See Fig. 3.

By comparing the early and late integrations, the phase of the incoming code can be revealed. Then, a code phase discriminator is implemented by the following equation:

$$\varphi_{\text{code}} = \frac{1}{2} \frac{E-L}{E+L} \tag{1}$$

where:

$$E = \sqrt{I_E^2 + Q_E^2},$$

$$L = \sqrt{I_L^2 + Q_L^2},$$

I_E, Q_E, I_L, Q_L are the integrations of the in-phase and the quadrant branches after being multiplied with both the early and the late codes.

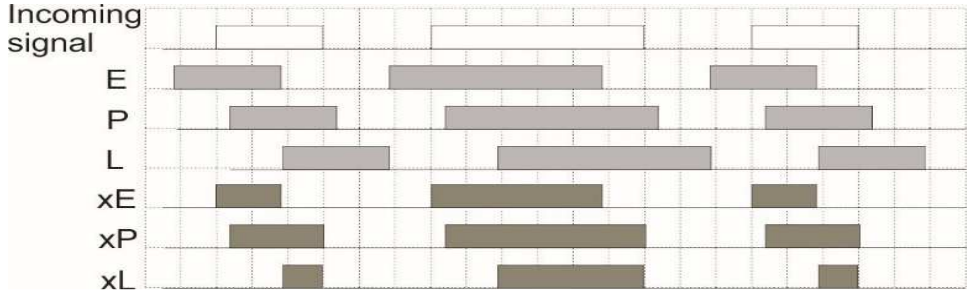


Fig. 2 Waveforms in delay locked loop

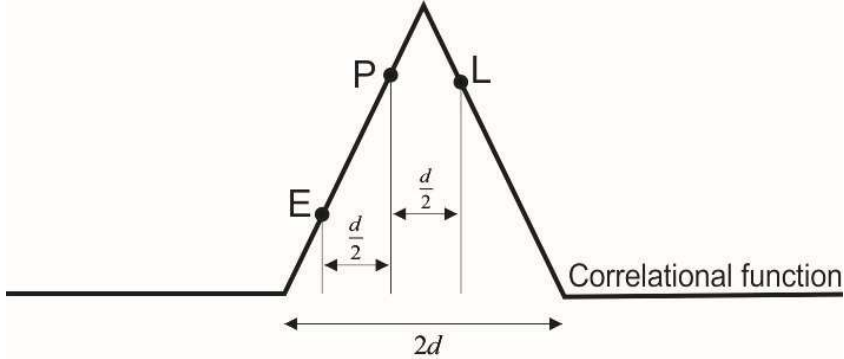


Fig. 3 Auto correlation function of PRN code [1]

The initial code phase is generally obtained at a correlation acquisition stage. Then, the code phase is tracked in a tracking block. The discriminator can track the phase difference within ± 0.5 chip. Otherwise, the code phase cannot be tracked and the signal is not de-spread [2].

Let us assume that the GPS signal is being modelled in (2) after the PRN code was removed by DLL. Then, the harmonic signal generator generates both the in-phase signal and the quadrant signal in intermediate frequency. They are expressed in (3) and (4):

$$x_{\text{gps}} = D_t \sin(\omega t + \varphi), \quad (2)$$

$$x_{\text{sin}} = \sin \omega t, \quad (3)$$

$$x_{\text{cos}} = \cos \omega t, \quad (4)$$

where D_t is the bit stream, -1 and $+1$, of navigation data that was modulated into a signal, ω is the angular velocity, $\omega = 2\pi f_i$ and φ is the carrier phase of the incoming signal.

Assuming that the carrier phase of the local harmonic signals is zero, then, a phase error between the incoming signal and the local signal is φ . Outputs of multiplications in both the in-phase branch and the quadrant branch are expressed in (5) and (6), from

which the constant parts and high frequency parts are separated. Then, the high frequency part is filtered and the constant part remains with the navigation data. As integrators act as a low pass filter, the following equations are used:

$$x_{Imult} = D_t \sin(\omega t + \varphi) \cos \omega t = \frac{D_t}{2} [\sin \varphi - \sin(2\omega t + \varphi)], \quad (5)$$

$$x_{Qmult} = D_t \sin(\omega t + \varphi) \sin \omega t = \frac{D_t}{2} [\cos \varphi - \cos(2\omega t + \varphi)]. \quad (6)$$

Results of the integrations are:

$$x_{Iint} = \frac{D_t}{2} \int_0^T [\sin \varphi + \sin(2\omega t + \varphi)] dt, \quad (7)$$

$$x_{Qint} = \frac{D_t}{2} \int_0^T [\cos \varphi - \cos(2\omega t + \varphi)] dt, \quad (8)$$

$$x_{Iint} = -\frac{D_t}{2} T \cos \varphi - \frac{D_t}{2} [\cos(2\omega T + \varphi) - \cos 0], \quad (9)$$

$$x_{Qint} = \frac{D_t}{2} T \sin \varphi - \frac{D_t}{2} [\sin(2\omega T + \varphi) - \sin 0], \quad (10)$$

where T is the integration time measured in samples.

The usual amount of samples T is several thousands. Both equation parts $D_t[\cos(2\omega T + \varphi) - \cos 0]/2$, and $D_t[\sin(2\omega T + \varphi) - \sin 0]/2$ can be neglected, as their values are too small in comparison to $-D_t T/2 \cos \varphi$ and $D_t T/2 \sin \varphi$. Then, the outputs of the integrators remain as follows:

$$x_{Iint} = -\frac{D_t}{2} T \cos \varphi, \quad (11)$$

$$x_{Qint} = \frac{D_t}{2} T \sin \varphi, \quad (12)$$

$$\varphi_{err} = \text{atan} \frac{x_{Qint}}{x_{Iint}}, \quad (13)$$

If the PLL is locked, the phase error is supposed to be zero. Because $\cos \varphi = 1$ and T is a constant, the signal x_{Iint} in (11) is demodulated navigation data. Then, a carrier discriminator is described in the Eq. (13).

A low pass filter is employed to reduce a noise in a phase error and to provide a smoother control signal to the local generators. The schema and the coefficients of the loop filter are given in Tab. 1 and Fig. 4.

3. The Navigation Equation

A pseudo range equation is represented in (14) [2]:

$$p_i = \sqrt{(x - x_i)^2 + (y - y_i)^2 + (z - z_i)^2} + R, \quad (14)$$

$$\partial p_i = \frac{x-x_i}{r_i} \partial x + \frac{y-y_i}{r_i} \partial y + \frac{z-z_i}{r_i} \partial z + \partial R, \quad (15)$$

$$\partial P = H_{\text{los}} \partial X, \quad (16)$$

$$\dot{p}_{\text{mea},i} = H_{\text{los}} (V_{\text{rec}} - V_{\text{sat},i}), \quad (17)$$

$$p_{\text{rec},i} = p_{\text{mean},i} + H_{\text{los}} V_{\text{sat},i}, \quad (18)$$

where x, y, z are the receiver positions, x_i, y_i, z_i represent the position of the i^{th} satellite, R is the residual range, caused by a clock error, $r_i = \sqrt{(x-x_i)^2 + (y-y_i)^2 + (z-z_i)^2}$, and p_i is the pseudo range from the receiver to the i^{th} satellite.

The velocity and position of the satellites are computed using ephemeris, which are obtained from the navigation data. The algorithm that converts orbital parameters of the satellite into ECEF coordinates is described in [3, 4].

Tab. 1 Low pass filter coefficients

Order	Coefficients
First	$\omega_0 = 4B_n$ $a = \omega_0$
Second	$\omega_0 = 1/0.53B_n$ $a_1 = \omega_0^2$ $a_2 = 1.424 \omega_0$

where B_n is the noise bandwidth that is needed to calculate the coefficients a, a_1, a_2 .

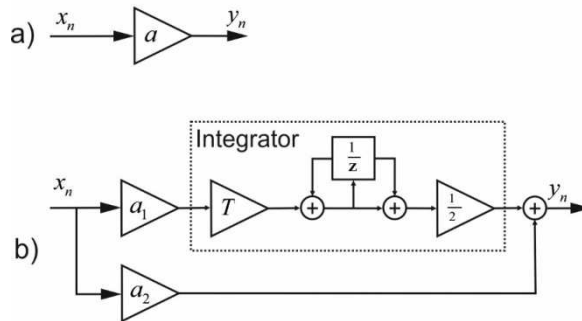


Fig. 4 a) First order and b) Second order discrete filters [2]

4. The Vector Tracking Loop and Kalman Filter

Fig. 5 shows the difference in the structure of the STL and the VTL [5, 6]. Most components of the STL remain in the VTL such as: a code generator, a carrier generator, integrators, multipliers, discriminators, etc. The difference is that each channel in the STL computes its own phase error to feed the local generator but the VTL combines all the channels together and computes the phase errors for all the channels by extended Kalman filter. To implement the extended Kalman filter, a code phase error and a carrier phase error are converted into pseudo range rates. The extended Kalman filter uses the pseudo range rates as a measurement vector. Then, the pseudo range rates are re-calculated and fed back to the local generators.

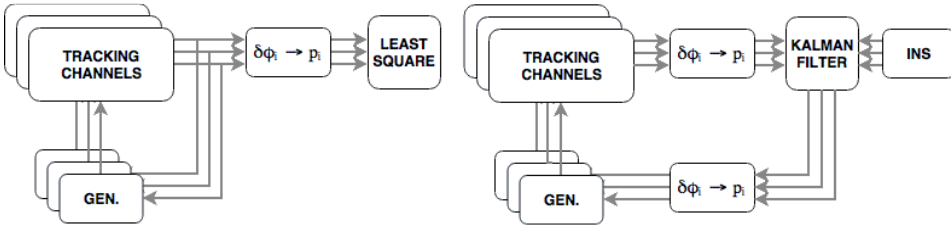


Fig. 5 Comparison structure of STL (left) and VTL (right) [5, 6]

The Kalman filter is a mathematical algorithm, which probabilistically combines different resources of information together to produce the best output. A model of the Kalman filter is described in (19) and (20):

$$\mathbf{x}_k = \mathbf{G}\mathbf{x}_{k-1} + \mathbf{B}\mathbf{u}_{k-1} + \mathbf{w}_{k-1}, \quad (19)$$

$$\mathbf{z}_k = \mathbf{H}\mathbf{x}_k + \mathbf{v}_k, \quad (20)$$

Equation (19) describes the dependence of the system state on time and is called a time transition equation, where \mathbf{x} is the state vector of the system, \mathbf{G} is the transition matrix, \mathbf{B} is the control matrix, \mathbf{u} is the control vector, k represents the time step and \mathbf{w} is the white noise, which is given by probability function $p(\mathbf{w}) = N(0, \mathbf{Q})$.

Equation (20) is called a measurement equation, where \mathbf{z} is the measuring vector of the state \mathbf{x} , \mathbf{H} is the measurement matrix and \mathbf{v} is the white noise, which is characterised by the probability function $p(\mathbf{v}) = N(0, \mathbf{R})$.

Application of the Kalman filter to the system, modelled in (19) and (20), can be described in two stages: A time update stage (or prediction) and a measurement update stage [7, 8].

The time update stage:

$$\bar{\mathbf{x}}_k = \mathbf{G}\mathbf{x}_{k-1} + \mathbf{B}\mathbf{u}_{k-1}, \quad (21)$$

$$\bar{\mathbf{P}}_k = \mathbf{G}\mathbf{P}_{k-1}\mathbf{G}^T + \mathbf{Q}, \quad (22)$$

The measurement update stage:

$$\mathbf{K}_k = \frac{\bar{\mathbf{P}}_k \mathbf{H}_k^T}{\mathbf{H}_k \bar{\mathbf{P}}_k \mathbf{H}_k^T + \mathbf{R}}, \quad (23)$$

$$\mathbf{x}_k = \bar{\mathbf{x}}_k + \mathbf{K}_k (\mathbf{z} - \mathbf{H}_k \bar{\mathbf{x}}_k), \quad (24)$$

$$\mathbf{P}_k = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \bar{\mathbf{P}}_k, \quad (25)$$

where the notation “ $\bar{\cdot}$ ” represents the prior state, in which the current state is updated from the previous state, but not updated by the measurement yet.

If the transition equation and the measurement equation are not linear, the extended Kalman filter is used instead. The extended Kalman filter is the Kalman filter applied to a linearized form of a non-linearized model.

5. Implementation and Simulation

The simulation model is built in MATLAB using a GPS signal, which is recorded by GNSS software receiver. The incoming GPS signal is converted into 5.5 MHz of intermediate frequency, sampled at 20.480 MHz and then sent to a computer in 2-bits format. As the PRN code rate is about 1.023 MHz, each chip in code is sampled 20 times (20.480 / 1.023). The local PRN code is generated using the algorithm described in [3]. The code discriminator and the carrier discriminator are described in (1) and (13).

The schema of the simulation is described in the state machine in Fig. 6. To retrieve the initial code phase, an acquisition block is employed to correlate the incoming signal with the local signal. Doppler frequency of the GPS signal falls into the interval ± 9.2 kHz [9, 10]. The chosen range (± 10 kHz) is then divided into 40 intervals, 500 Hz each. At these stages, the acquisition engine searches for satellites in two dimensions: 32 PRN codes and 40 frequency bins. The outputs of the acquisition stage are satellites currently in view, their initial code phase and their carrier Doppler shift. The receiver delivers the satellites with their initial code phase, PRN number and their Doppler shift to the scalar tracking stage. Each channel tracks one satellite and the integration time in the tracking loops is 1 millisecond. The tracking stage runs until the length of the tracked signal is at least 5 sub-frames. Each sub-frame, which contains 300 navigation bits at 50 Hz of a bit rate, is about 6 seconds long. Then, the receiver jumps to the decoding stage, decodes the navigation data and sends a navigation message to the positioning stage. The positioning stage uses a code phase and a carrier phase, which were obtained from the scalar tracking stage, to formulate the pseudo ranges and to compute the receiver position. After calculating the receiver position, the receiver returns to the scalar tracking stage only if the vector tracking option is not enabled. Otherwise, the receiver jumps to the vector tracking stage.

In the extended Kalman filter, the state vector is the receiver velocity $X_{(4 \times 1)} = [v_x, v_y, v_z, \dot{R}]$. The pseudo range rate can be derived from both the code phase and the carrier phase errors, thus the measurement matrix of light is a double size of the sight matrix. Transition matrices $G_{(4 \times 4)}$ and measurement matrices $H_{(10 \times 4)}$ are given in (26) and (27). The measurement matrices consist of a pseudo range rate, which is invoked by the movement of the receiver. They are calculated by (28), (29), (30) and (31).

$$G_{(4 \times 4)} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad (26)$$

$$H_{(10 \times 4)} = \begin{pmatrix} H_{\text{los}} \\ H_{\text{los}} \end{pmatrix} = \begin{pmatrix} \frac{x-x_1}{r_1} & \frac{y-y_1}{r_1} & \frac{z-z_1}{r_1} & 1 \\ \frac{x-x_2}{r_2} & \frac{y-y_2}{r_2} & \frac{z-z_2}{r_2} & 1 \\ \dots & \dots & \dots & \dots \\ \frac{x-x_1}{r_1} & \frac{y-y_1}{r_1} & \frac{z-z_1}{r_1} & 1 \\ \frac{x-x_2}{r_2} & \frac{y-y_2}{r_2} & \frac{z-z_2}{r_2} & 1 \\ \dots & \dots & \dots & \dots \end{pmatrix}, \quad (27)$$

$$\dot{p}_{\text{code}} = \frac{\varphi_{\text{code}}}{20480} c, \tag{28}$$

$$f_{\text{dop}} = f_{\text{acq}} + 1000 \frac{\varphi_{\text{carrier}}}{2\pi}, \tag{29}$$

$$\dot{p}_{\text{carrier}} = f_{\text{dop}} \frac{c}{f_{L1}}, \tag{30}$$

$$z = \begin{pmatrix} \dot{p}_{\text{code},1} \\ \dot{p}_{\text{code},2} \\ \dots \\ \dot{p}_{\text{carrier},1} \\ \dot{p}_{\text{carrier},2} \\ \dots \end{pmatrix} + \text{diag} \left[\begin{pmatrix} \frac{x-x_1}{r_1} & \frac{y-y_1}{r_1} & \frac{z-z_1}{r_1} & 1 \\ \frac{x-x_2}{r_2} & \frac{y-y_2}{r_2} & \frac{z-z_2}{r_2} & 1 \\ \dots & \dots & \dots & \dots \\ \frac{x-x_1}{r_1} & \frac{y-y_1}{r_1} & \frac{z-z_1}{r_1} & 1 \\ \frac{x-x_2}{r_2} & \frac{y-y_2}{r_2} & \frac{z-z_2}{r_2} & 1 \\ \dots & \dots & \dots & \dots \end{pmatrix} V_{\text{sat}} \right], \tag{31}$$

where f_{dop} is Doppler shift, f_{acq} is Doppler frequency acquired in an acquisition stage and c is the speed of light.

After each step of the Kalman filter, a code phase and a carrier phase are re-calculated using newly the value of the receiver velocity.

To test the performance of the VTL, the phase error feedback is interrupted to simulate a blocked signal. Instead of using the correct code phase, a local PRN random code phase is applied incorrectly to multiply with the incoming signal. Since the GPS signal spectrum is spread by the PRN code, the code phase is critical to unlock and compress the spectrum. And as the GPS signal level on ground is about -158 dBW [3], much lower than the thermal noise level, the interrupting code phase takes effect in the same way as when the incoming signal is absent.

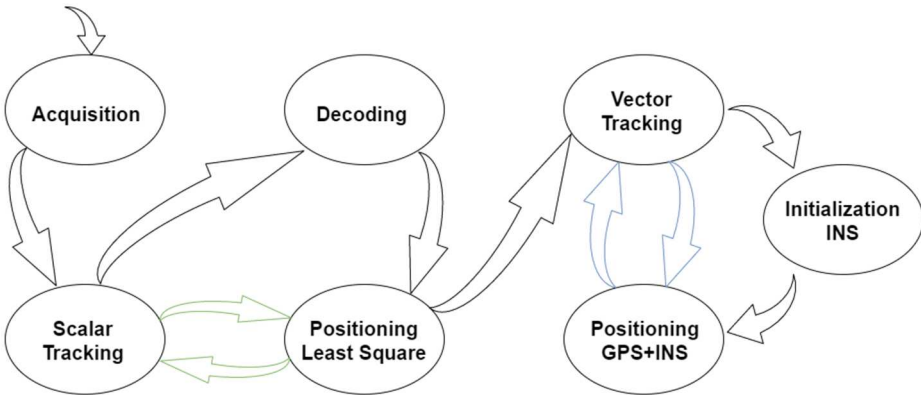


Fig. 6 State machine of simulation

6. Results

6.1. Comparison of the VTL and the STL

After 74 seconds, the acquisition and tracking blocks acquire 5 satellites with PRN number: 8, 11, 14, 22 and 27. Fig. 7 shows the position of the satellites currently in-view in the sky. Information about the state of the satellites such as their Doppler frequency, pseudo range and pseudo range rate are described in Tab. 2. Then, the receiver runs the simulation for the vector tracking and the scalar tracking. In this simulation, the satellites were set up for a random interruption time, which is described in Tab. 3. The simulation runs for 16 seconds.

Tab. 2 Satellite parameters

Satellites	8	11	14	22	27
Doppler [Hz]	-685	2 524	1 439	-1 997	-2 793
Pseudo range [m]	20 524 418	21 197 291	22 250 490	21 507 245	21 707 262
Range rate [m/s]	1 306	-479	-272	380	532

Tab. 3 Interrupt time of satellites

Satellite	8	11	14	22	27
Interrupting time [s]	2 ÷ 4	2 ÷ 6	3 ÷ 6	4 ÷ 11	7 ÷ 12

Fig. 8, 9 and 10 show the state of three channels in the first simulation. The top subplot expresses a code phase error; the following subplot is a code phase that is an integration of the code phase error; and the two bottom subplots are amplitudes of the incoming signals after de-spreading. Blue lines are signals of the scalar tracking and the orange lines are signals of the vector tracking algorithm. The green vertical markers indicate the time when the signal started interrupting and stopped interrupting.

The raising level of the de-spread signal indicates that the code phase was tracked correctly. Figs. 8, 9 and 10 show that the VTL tracked the signal correctly in all available channels. The situation in the STL was opposite. Channel 14 suffered and all the rest lost the code phase. In the STL when the signal was turned off, the phase error was the only noise random value. The integration of the random signal produced a random walking signal, thus the code phase in the scalar tracking channels diverged after the signal was turned off. When the signal returned, without a correct code phase, the incoming signal could not be de-spread and the amplitude of the signal remained a noise. That is what happened to channels 8, 11, 22, 27. The code phase subplot shows that the code signal was completely different to the correct value. In channel 14, the code phase diverged while the time signal was turned off, however, the difference was not too big and it could still catch the correct phase after the signal returned. This error was reduced when the signal was turned off for a longer time.

6.2. The influence of quantity of channels upon VTL performance

This simulation demonstrated the influence of quantity of channels on the performance of the VTL. The simulation was run for four times. First time, only channel 8 was interrupted so it was supported by all four channels. Second time, channels 8 and 11 were interrupted, so that channel 8 was supported by only three channels. Then, 11 and 14 were interrupted and in the last run, channels 8, 11, 14 and 22 were interrupted. Interrupting time was 30 seconds, from second one to second 31.

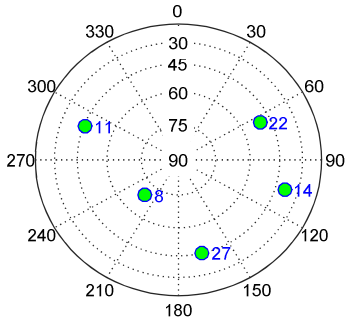


Fig. 7 Sky plot of the in-view satellites

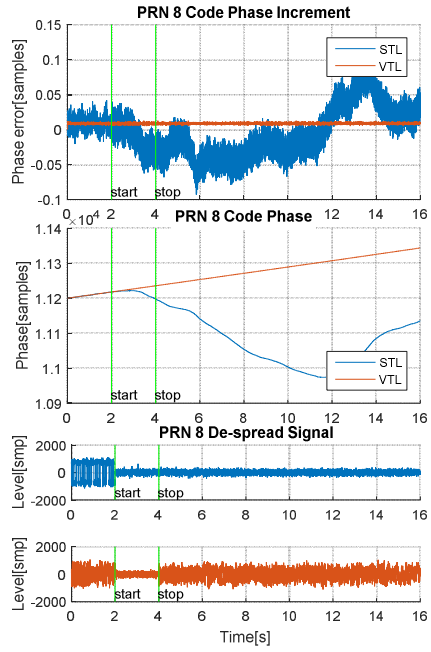


Fig. 8 Signal in channel 8

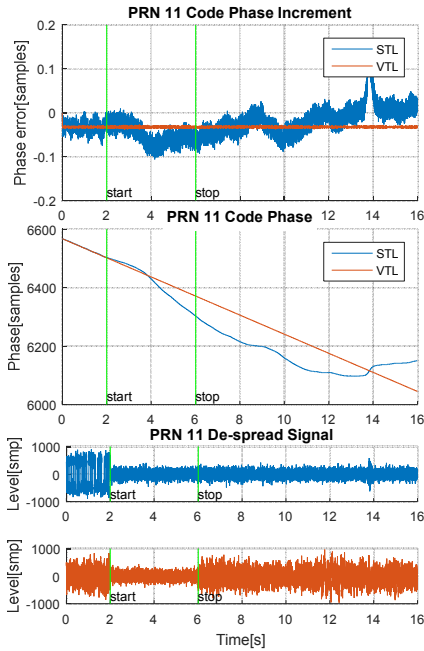


Fig. 9 Signal in channel 11

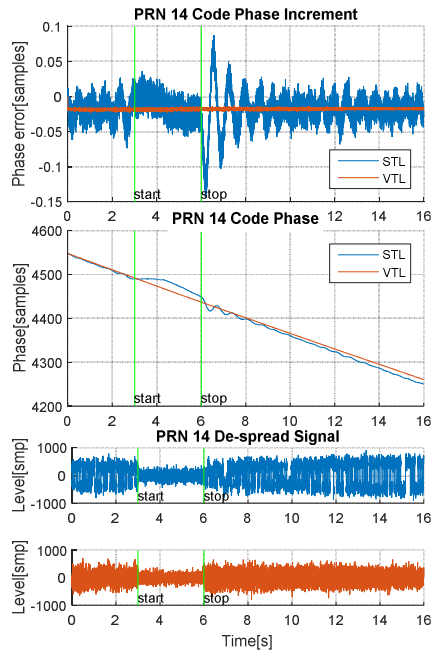


Fig. 10 Signal in channel 14

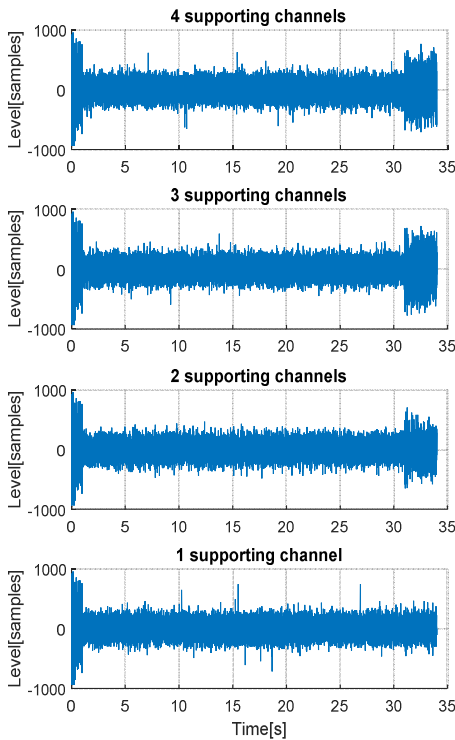


Fig. 11 De-spread signal PRN 8 in difference numbers of supporting channels

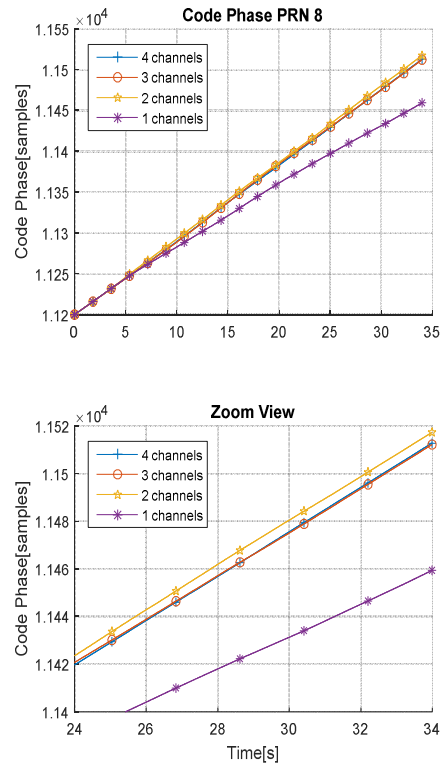


Fig. 12 Code phase PRN 8 in difference numbers of supporting channels

In Fig. 11, the raising amplitude of the de-spread signal at 31th second indicates that the code phase was tracked correctly with the support of three or four channels. With two supporting channels, the amplitude was dropped a little. With only one supporting channel, the VTL was not able to track channel 8 during the 30 seconds of the signal being turned off. Fig. 12 shows the code phase of channel 8. The top subplot is the full time code phase in the simulation. The bottom subplot is a zoom view at the end of the simulation. The code phases, in the situation with three or four supporting channels, gathered relatively close to each other. With two supporting channels, the code phase diverged a little, and in case of one supporting channel, the difference is relatively large.

This simulation showed that the channels do influence the VTL and the larger the amount of channels used in the loop, the better tracking ability of the whole algorithm.

7. Conclusion

This paper presented the implementation, simulation and analysis of the VTL characteristic by using the extended Kalman filter. Principles of the tracking loops and the VTL were shown. Structure of the extended Kalman filter was described by equations.

The experiments shown in chapter 4 were conducted to compare the STL and the VTL in the first case and robustness of the VTL itself in the second case. Because the

VTL keeps code phase and carrier phase via Kalman filtering, the VTL is able to almost instantaneously recover tracking of previously lost signal. The presented simulation also showed that within 2 to 4 seconds of the signal not being available, the code phase was mostly lost in the STL, while the VTL only suffered during that interval. The results allow us to state that the VTL demonstrated superior performance against the traditional STL. In the second case of experiments, the VTL proved the robustness of the tracking loop in case of single or multiple PRN signal losses. The simulation showed us that if you want to track the code phase in channel suffering signal loss, the VTL only needs two other supporting channels (meaning other existing signals with different PRN). However, the VTL also has a disadvantage, because it is a very computationally demanding process requiring the state of the art computers and DSP devices.

Acknowledgement

The work presented in this paper has been supported by the Ministry of Defence and the Ministry of Education, Youth and Sports of the Czech Republic (K206 Student research program “Implementation of modern technologies in avionics systems”).

References

- [1] VU, B.N. and ANDRLE, M. The code and carrier tracking for GPS Signal. In *16th Mechatronika 2014*. Brno: University of technology, 2014, p. 569-574.
- [2] KAPLAN, E.D. and HEGARTY, CH.J. *Understanding GPS – Principles and Applications*. Norwood, MA: Artech House, 2006.
- [3] *Global Positioning System, Standard Positioning Service, Performance Standard*. Washington: Pentagon, 2008.
- [4] *GLONASS, Interface Control Document, Navigational radiosignal in bands L1, L2*. Moscow: Russian Institute of Space Device Engineering, 2008.
- [5] VU, B.N. and ANDRLE, M. Implementation Of Vector Tracking Loop Algorithm In Modern Gps Receiver. In *Digital Avionics Systems Conference (DASC), 2014 IEEE/AIAA 33rd*. Colorado Springs, CO, 2014, p. 3E1-1 – 3E1-9.
- [6] MATTHEW, L. and BELVY D. Comparison of traditional tracking loops and vector based tracking loops for weak GPS signals. In *National Technical Meeting of The Institute of Navigation*. Manassas, VA, 2008, p. 310-316.
- [7] VASEGHI, S.V. *Advanced Digital Signal Processing and Noise Reduction*. London: Wiley, 2008.
- [8] WELCH, G. and BISHOP, G. *An Introduction to the Kalman Filter*. Chapel Hill: University of North Carolina, 2006.
- [9] DIGGELEN, F.V. *Assisted GPS, GNSS and SBAS*. London: Artech House, 2009.
- [10] BAO-YEN, J.T. *Fundamentals of Global Positioning System Receivers: A Software Approach*. London: Wiley, 2005.