THE GPS CODE SOFTWARE RECEIVER AT AALBORG UNIVERSITY

Nicolaj Bertelsen, Kai Borre, Peter Rinder

Danish GPS Center, Aalborg University
Niels Jernes Vej 14, DK-9220 Aalborg Ø, Denmark
{nb,borre,rinder}@gps.aau.dk

INTRODUCTION

In the early spring of 2004, the development of a fully-functional GPS software receiver was initiated at Aalborg University. This development started as a student project/Master thesis. The result from this project was a complete L1 software receiver implemented in MATLAB capable of computing positions accurate to about 10 meters. Based on these results, it was decided to continue the software receiver development.

The development of a software receiver contains enormous perspective regarding numerous different areas and applications. First of all, the development establishes an in-depth knowledge of the GPS signal structure and signal processing algorithms. Second, by definition the software receiver is already prepared to be customized to future changes in the GPS signal structure and more important by the future European positioning system GALILEO. Finally, the possibility of supplying a test-bed to the development of receiver algorithms with focus on multi-path mitigation is very interesting.

In general we refer to Dennis Akos [1] regarding software receivers and to Strang and Borre [4] regarding position computations.

RECEIVER OVERVIEW

The software receiver follows the most commonly used GPS receiver structure. This structure that can be seen in Fig. 1 is divided into an analog part and a digital part.

The digital part contains a GPS antenna, an RF front-end, and an A/D converter. The digital part contains N receiver channels and a position computation block. The position computation block can be replaced by any other user algorithm block if an output different than the receiver position is needed.

EMPLOYED HARDWARE

As seen in the previous section, the GPS software receiver is not only based on software being executed on a multi-purpose processor. As any other GPS receiver, this receiver also needs an antenna. Even though the removal of interfering signals could be performed on digital signals, it was decided that this should be done on analog signals. An RF front-end mainly handles the filtering, but additionally down-converts the signal to an intermediate frequency (IF). This is done to lower the needed sampling frequency used in the A/D conversion. Initially, the design of an antenna and RF front-end was not part of the development project at Aalborg University. As a result, a GPS antenna and a ready-made front-end was acquired from the company Simrad that develops marine electronics.

The front-end involves a two-step down-conversion of the L1 GPS signal to an IF of 3.563 MHz. In addition to the IF signal, the front-end supplies a local oscillator (LO) signal of 11.999 MHz as an output. The design of the front-end suggests that this LO signal is used as sampling clock signal to avoid possible phase jitter between the LO used for down-conversion and an internal sampling clock signal.

Based on the suggestions from the front-end designers, the demands for the A/D converter included the possibility of supplying an external sampling clock. Peripheral Component Interconnect (PCI) was chosen as the interface standard between front-end and computer. This was chosen based on high availability and a fairly simple programming interface.

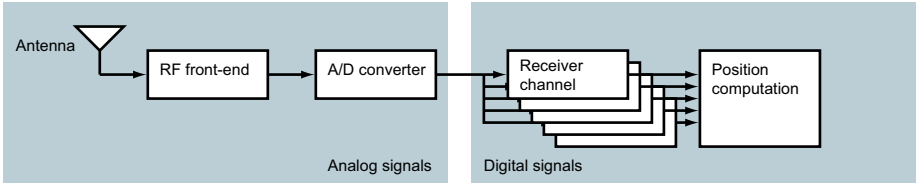

Fig. 1: GPS software receiver overview

The chosen A/D converter is an ICS-652, which is a 2 channel, 65 MHz, 14-bit data acquisition card designed to be used in software receiver applications.

The computer used for the software receiver is a Fujitsu-Siemens Scaleo with a Pentium 4 2.8 GHz processor with 1 GB RAM. The computer is running Windows XP Professional and MATLAB 7.0 (R14).

ACQUISITION

During the software receiver development, three different methods of acquisition were implemented and tested for their properties regarding computation speed and robustness. The three methods counted *Serial search acquisition*, *Parallel frequency space search acquisition*, and *Parallel code phase search acquisition*. The latter method was chosen based on its superior computation speed resulting from the parallelism of the algorithm. The parallelism of the algorithm makes the two-dimensional search from the well-known serial search acquisition method unnecessary. The parallel code phase search acquisition method parallelizes the code phase dimension, thus eliminating the need for performing an exhaustive search in this dimension. The block diagram that describes the parallel code phase search acquisition algorithm is seen in Fig. 2.

The theory of the parallel code phase search acquisition method is based on the theory on circular correlation as described in [3].

The discrete Fourier transform (DFT) of a finite length sequence $x(n)$ with length N is calculated as

$$X(k) = \sum_{n=0}^{N-1} x(n) e^{-j2\pi kn/N}. \tag{1}$$

The cross-correlation between two finite length sequences $x(n)$ and $y(n)$ both with length N is calculated as

$$z(n) = \sum_{m=0}^{N-1} x(m) y(n+m). \tag{2}$$

In (2) we omit a possible scaling factor. The combination of (1) and (2) gives the DFT of the cross-correlation between x and y

$$Z(k) = \sum_{n=0}^{N-1} \sum_{m=0}^{N-1} x(m) y(n+m) e^{-j2\pi kn/N} \tag{3}$$

$$= \sum_{m=0}^{N-1} x(m) e^{j2\pi km/N} \sum_{n=0}^{N-1} y(n+m) e^{-j2\pi k(n+m)/N} \tag{4}$$

$$= X^*(k) Y(k) \tag{5}$$

where * denotes complex conjugation.

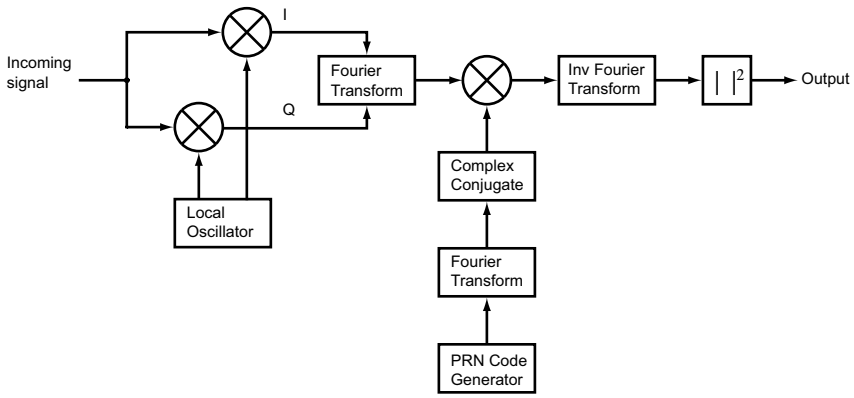

Fig. 2: Structure of the parallel code phase search acquisition method

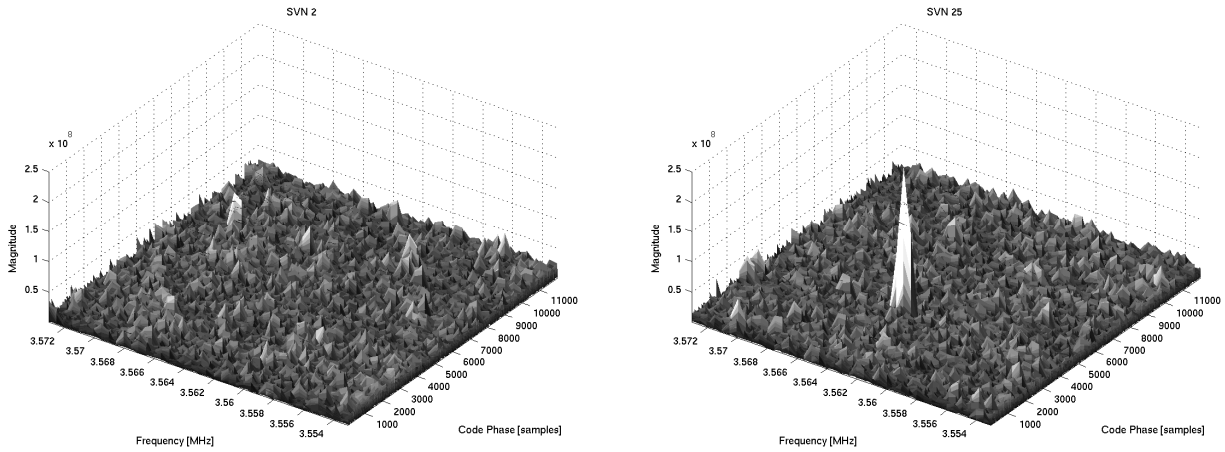

Fig. 3: Results from parallel code phase search acquisition of satellite not visible to the receiver and satellite visible to the receiver, respectively

From (5) it is seen that the DFT of the cross-correlation between sequences x and y can be calculated as the multiplication of the frequency domain representations of the two sequences X and Y where one of the sequences is complex conjugated. The desired time-domain cross-correlation is found as the inverse Fourier transform of the result of the multiplication.

As indicated, the parallel code phase search acquisition performs circular correlation between the incoming signal and a local code. However, this does not influence the need of searching through possible carrier frequencies. That is, the circular correlation must be performed for all possible carrier frequencies. The search frequencies lie within the interval ± 10 kHz to take the maximum possible Doppler shift into account, in case of a very fast moving receiver. The searched Doppler frequency bins are chosen to be 500 Hz which turned out to be a good compromise between frequency error and computational demands. The Doppler frequency bins are each of 500 Hz so the maximum frequency error is ± 250 Hz which the initial settings of the carrier tracking loop must be able to handle.

The parallel code phase search acquisition supplies a code phase with sampling time precision which is very important. The serial search acquisition method normally searches with $1/1$ chip or $1/2$ chip precision ($\approx 1\,\mu$s, $\approx 0.5\,\mu$s). The sampling precision of the code phase is estimated with e.g. $0.1\,\mu$s in case of 10 MHz sampling. That is, the acquisition supplies a very accurate code phase. An example of the output from the implemented acquisition algorithm is seen in Fig. 3 where the left figure shows the output in case the algorithm searches for a satellite currently *not* visible to the receiver, and the right shows the output in case the algorithm searches for a satellite that *is* currently visible.

CODE TRACKING

The code tracking block of the software receiver is implemented using the method of Early-Late code tracking, that involves correlation with three different generated codes known as the early (E), the prompt (P), and the late (L) codes. The code tracking block demands a carrier input to remove the carrier component from the signal before performing the correlation. The carrier input to the code tracking block is supplied by the carrier tracking block. To add robustness to the code tracking block, it is designed to track the code on both an I and a Q component of the signal. The implemented code tracking loop is seen in Fig. 4.

In the initial state of the carrier tracking loop the carrier phase of the signal is unknown. In that case, the energy from the correlations might as well be located in the Q branch correlators as in the I branch correlators. When the carrier tracking loop has tracked the carrier phase successfully, most of the energy is located in the I branch correlators. This property makes the code tracking faster as it can track the code independently of the carrier phase. Additionally, it supplies better conditions to the carrier tracking loop with a faster solution of the code phase.

The output from the six correlators are combined to provide an input to the code generator so the code phase of the generated PRN code will be properly adjusted. The input to the code generator is computed through a tracking loop discriminator. The used discriminator is referred to as the normalized early minus late power discriminator and is

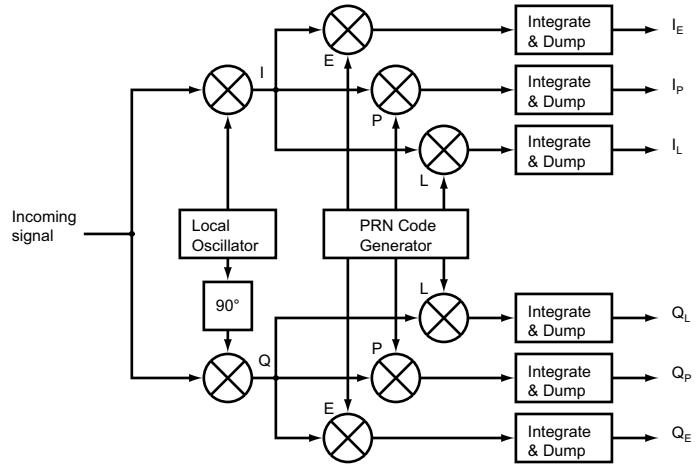

Fig. 4: I and Q code tracking loop with six correlators

calculated as

$$D = \frac{(I_E^2 + Q_E^2) - (I_L^2 + Q_L^2)}{(I_E^2 + Q_E^2) + (I_L^2 + Q_L^2)}. \tag{6}$$

The normalized early minus late power discriminator is chosen because it makes the code tracking loop independent of the carrier tracking loop as it estimates correlation in both the I and the Q branch of the code tracking loop [6].

CARRIER TRACKING

The carrier tracking block of the software receiver is implemented as a Costas Phase Lock Loop (PLL). A PLL measures the carrier phase error and adjusts the frequency of the local oscillator based on that error. Compared to Frequency Lock Loops (FLL) and other PLL's, the Costas PLL is capable of tracking the phase of the signal even with recurrent phase changes of 180°. In GPS, this property is especially useful as the navigation data bits involves such phase changes in the signal. That is, a Costas PLL is insensitive to navigation bit transitions [6]. The Costas loop containing a carrier loop discriminator, a carrier loop filter, and a numerically controlled oscillator (NCO) can be seen in Fig. 5.

The loop discriminator is implemented as an arctangent discriminator because of its high accuracy and insensitivity towards navigation bit transitions [5]. The arctangent discriminator output is calculated as

$$\phi = \tan^{-1}\left(\frac{Q^k}{I^k}\right) \tag{7}$$

where Q^k and I^k are the outputs from the low-pass filters in the I and Q branches for satellite number k.

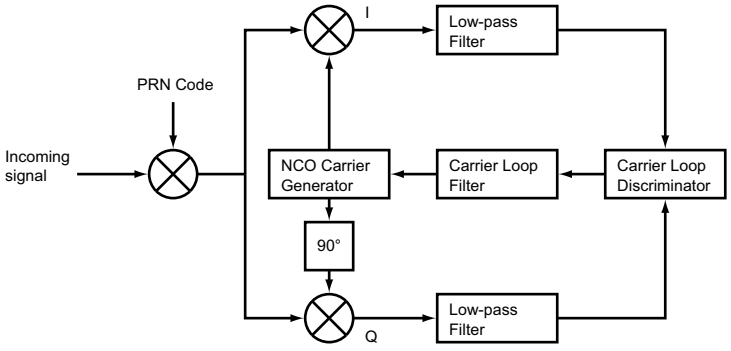

Fig. 5: Costas loop used for carrier phase tracking

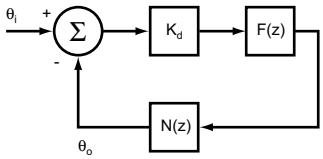

Fig. 6: Linearized digital second order PLL model

The implemented PLL is a second order system containing a first-order filter and an NCO. A linearized model for a digital second order PLL is shown in Fig. 6. The discriminator gain is denoted K_d, the loop filter transfer function is denoted $F(z)$, and the NCO transfer function is denoted $N(z)$.

Transfer functions for the loop filter and NCO are defined as

$$F(z) = \frac{(C_1 + C_2) - C_1 z^{-1}}{1 - z^{-1}} \tag{8}$$

$$N(z) = \frac{K_o z^{-1}}{1 - z - 1}. \tag{9}$$

Based on the transfer function in (8), a block diagram showing the structure of the loop filter is shown in Fig. 7. The connection between an analog and the digital PLL transfer function is described in [2] which supplies the resulting digital PLL transfer function

$$H(z) = \frac{K_o K_d (C_1 + C_2) z^{-1} - K_o K_d C_1 z^{-2}}{1 + (K_o K_d (C_1 + C_2) - 2) z^{-1} + (1 - K_o K_d C_1) z^{-2}} \tag{10}$$

where the connection between the constants C_1 and C_2 and the corresponding analog PLL transfer function constants is given as

$$C_1 = \frac{1}{K_o K_d} \frac{8\zeta\omega_n T}{4 + 4\zeta\omega_n T + (\omega_n T)^2} \tag{11}$$

$$C_2 = \frac{1}{K_o K_d} \frac{4(\omega_n T)^2}{4 + 4\zeta\omega_n T + (\omega_n T)^2} \tag{12}$$

where $K_o K_d$ is the loop gain, ζ is the damping factor, ω_n is the natural frequency, and T is the sampling time. The pull-in sequence with three different frequency offsets is shown in Fig. 8

COMBINED CODE AND CARRIER TRACKING

The code and carrier tracking loops are working coherently being mutually dependent on each other. So instead of dividing these two functionalities into two different blocks they are merged into a combined code and carrier tracking loop. This does not simplify the algorithm structure but it makes some computational improvements possible. The improvement is based on the similar elements in the two algorithms. This includes multiplication of the incoming signal with a local carrier signal, multiply the incoming signal with a local PRN code, and finally low-pass filter the result to obtain an I and a Q.

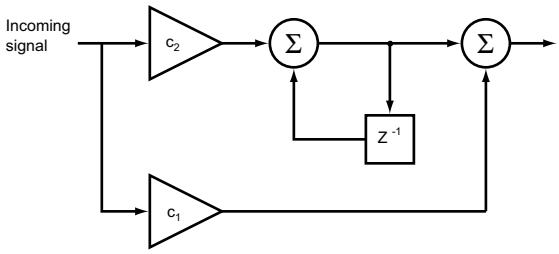

Fig. 7: Second order PLL filter

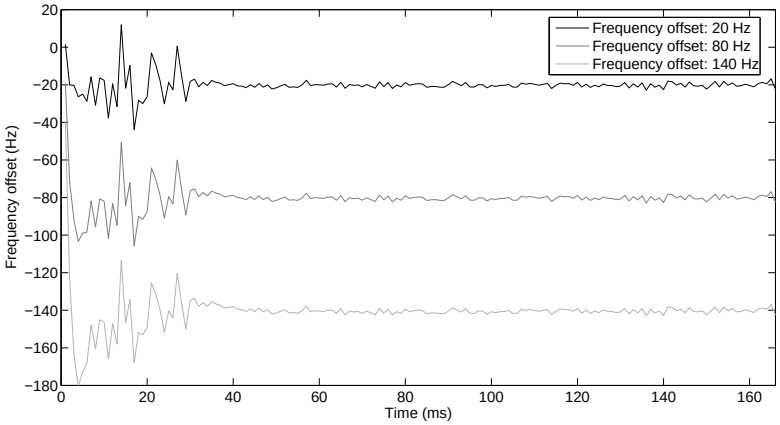

Fig. 8: PLL pull-in sequence

The complete combined code and carrier tracking block can be seen in Fig. 9: The common I and Q are the outputs from the correlation with the prompt code in the code tracking loop. It is used as input to the code loop discriminator in combination with the results from correlation with early and late codes. But it is also used as input to the carrier loop discriminator. In this way, the three multiplications in the ordinary carrier tracking loop are eliminated.

TEST RESULTS

The first system test should examine how the system would react on missing sequences in the signal. This could for instance occur if the receiver was located on a moving vehicle and some object obstructed the line-of-sight between antenna and satellite.

The focus on the test is especially the tracking loops that must be able to track the satellite after the signal has been reestablished. The test is performed on a sequence of real sampled data with extreme additive noise for 100 ms. The result from the carrier tracking loop can be seen in Fig. 10. As seen in the figure, the phase and the frequency are stabilizing shortly after the signal reappears.

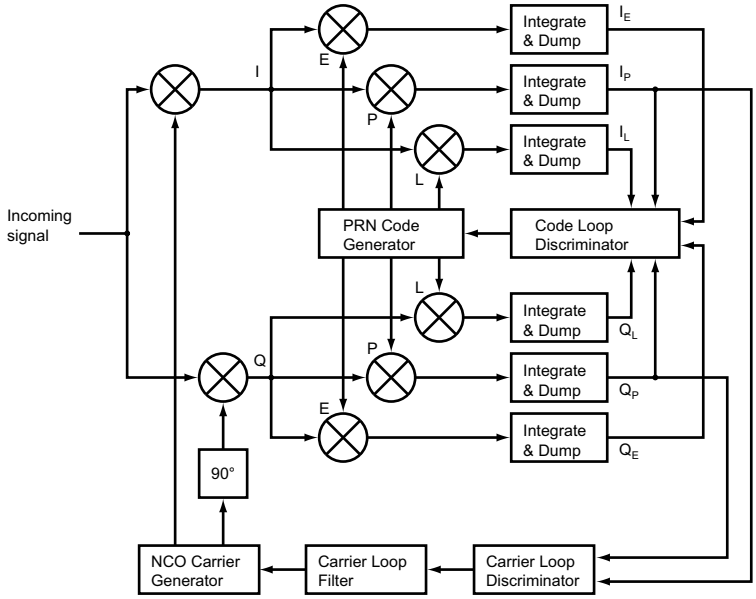

Fig. 9: The complete combined code- and carrier tracking loops

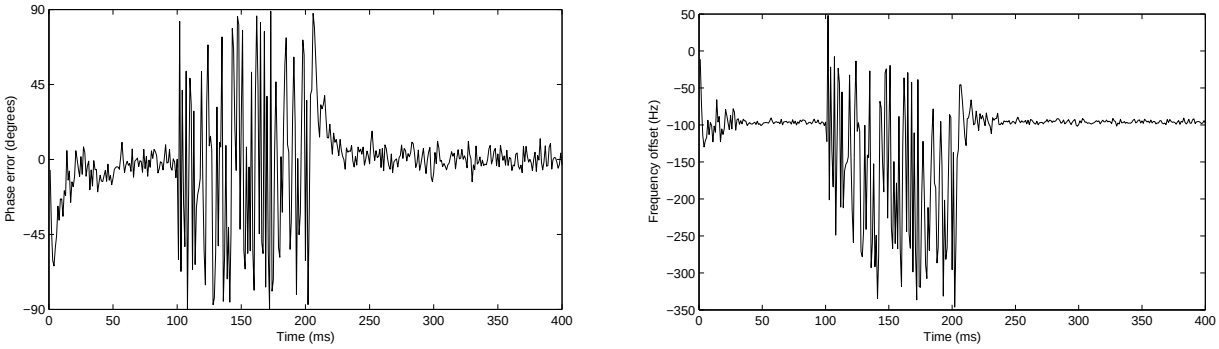

Fig. 10: The response of the phase lock loop. Left: The phase error between the input carrier and the local carrier. Right: The frequency offset of the carrier compared to the acquired frequency

An important output from the tracking block is the navigation data bits. The data bits during the same time interval is shown in Fig. 11.

The second test investigates the position accuracy of the software receiver. Obviously, this test demonstrates all implemented algorithms working together. The test is run on a 30 s sequence that includes one complete frame of navigation data and thus all transmitted ephemerides. This information makes it possible to compute satellite positions and pseudo-ranges that are needed in the position computations. The result from the position computation test is seen in Fig. 12. As seen in the figure, the position computations have an accuracy of approximately 10 m.

CONCLUSIONS

A single frequency C/A code software receiver that runs in MATLAB on a common PC has been implemented and tested. The employed hardware includes a Simrad antenna and front-end, an ICS data acquisition card, and a Pentium 4 PC. The software receiver algorithms were implemented in MATLAB. The receiver algorithms include GPS signal acquisition, code tracking, carrier tracking, and additional algorithms needed to supply a receiver position once every millisecond.

The implemented software receiver has been tested regarding temporary signal interference and position accuracy. Even though it was not possible to make the receiver run completely in real-time due to problems with real-time import of data from the data acquisition card into MATLAB, algorithm improvements have increased the computational speed and thus making the software receiver capable of running in real-time.

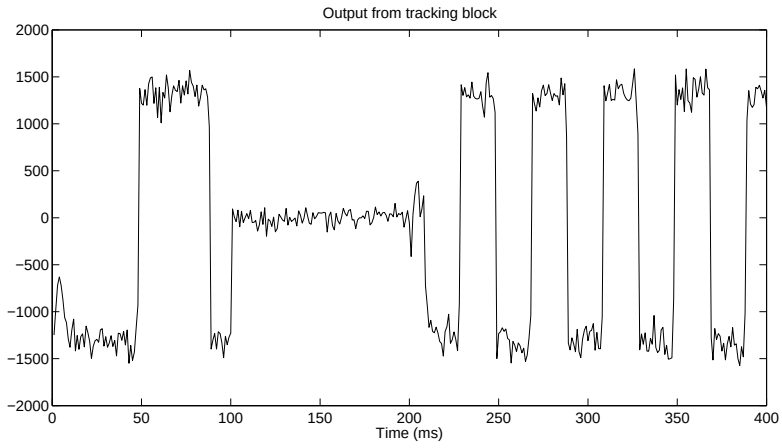

Fig. 11: The demodulated navigation bits from the tracking block

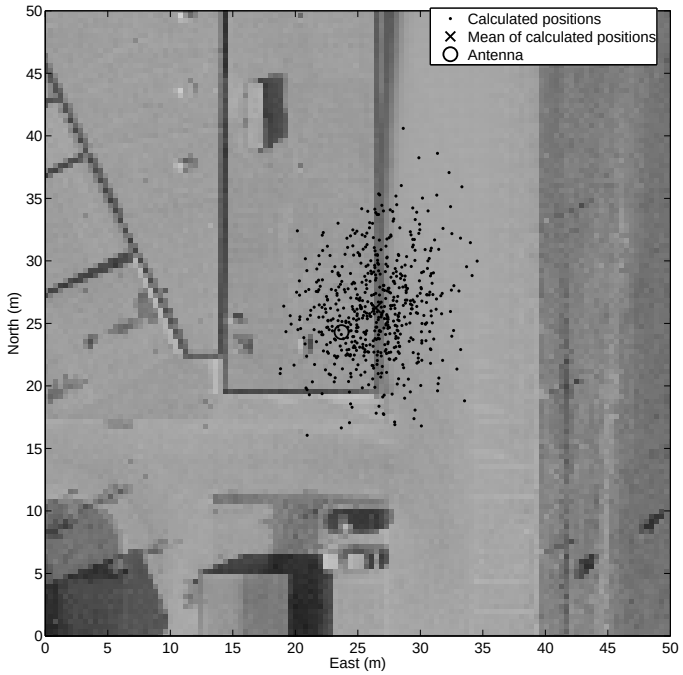

Fig. 12: Positions computed by the GPS software receiver

REFERENCES

[1] D. M. Akos. *A Software Radio Approach to Global Navigation Satellite System Receiver Design*. Ph.D. dissertation, Ohio University, 1997.

[2] B-Y. Chung, C. Chien, H. Samueli, and R. Jain. Performance analysis of an all-digital BPSK direct-sequence spread-spectrum IF receiver architecture. *IEEE Journal on Selected Areas in Communications*, 11:1096–1107, 1993.

[3] A. Oppenheim and R. Schäfer. *Discrete-Time Signal Processing*. Prentice-Hall, second edition, 1999.

[4] G. Strang and K. Borre. *Linear Algebra, Geodesy, and GPS*. Wellesley-Cambridge Press, 1997.

[5] A. J. Van Dierendonck. GPS receivers. In Bradford W. Parkinson and James J. Spilker, Jr., editors, *Global Positioning System: Theory and Applications*, number 163 in Progress In Astronautics and Aeronautics, chapter 8. American Institute of Aeronautics and Astronautics, Inc., Washington, DC, 1996.

[6] P. Ward. Satellite signal acquisition and tracking. In Elliot D. Kaplan, editor, *Understanding GPS Principles and Applications*, chapter 5. Artech House Publishers, 1996.