An Ultra-Sensitive Software GPS Receiver for Timing and Positioning

Nobuhiro Kishimoto, *CEO, Magellan Systems Japan*
Jon Vavrus, *CTO, Magellan Systems Japan*
Dr. Lawrence R. Weill, *Chief Scientist, Magellan Systems Japan*

BIOGRAPHIES

Nobuhiro Kishimoto

Mr. Kishimoto entered the Tokyo University of Science in 1976, then he moved to the Kwansei Gakuin University where he received a Bachelor's degree in Theology in 1984. Mr. Kishimoto has many years of experience in the development of semiconductor processing techniques and 22 years in the GPS industry.

In 1988 he founded Magellan Systems Japan, Inc., which has long been exclusively involved in representing the GPS OEM business, as well as being an international distributor for a worldwide well known GPS manufacturer in the US (formerly Magellan Systems Corporation) in the Japanese market.

In addition to marketing experience, in 1999 Mr. Kishimoto spearheaded innovative developments in high-sensitivity assisted GPS receiver technology for application to cell phone position location. Recently he has been in charge of an advanced generation of this technology and hybrid GNSS solutions for some advanced peripheral technologies.

Jon Vavrus

Mr. Vavrus received a B.S. degree in Chemistry from the California Institute of Technology in 1979. He then entered upon a career in software and system engineering starting at Pasadena's Jet Propulsion laboratory where he worked on very long baseline interferometry systems and massively parallel (multi-processor) operating systems.

In the late 1980s Mr. Vavrus moved into the GPS industry, first at several companies and then with a long stint at Magellan Systems Corporation. At Magellan he held positions of Director of SW Engineering, VP of Software Engineering, and CTO. In the last few years he has been the CTO of Magellan Systems Japan.

For Magellan Systems Corporation and Magellan Systems Japan, Mr. Vavrus has been actively involved in developing the core technologies for: Low power/low cost slow sequencing receivers; low power/low cost parallel tracking receivers; high performance multi-channel fast acquisition receivers; multipath mitigation techniques; and super-sensitive timing reference software based receivers.

Dr. Lawrence R. Weill

Dr. Weill received B.S. and M.S. degrees in Electrical Engineering from the California Institute of Technology in 1960 and 1961, respectively. In 1968 he earned the M.S. Degree in Mathematics at San Diego State University, and was awarded the Ph.D. in Mathematics in 1974 at the University of Idaho. He is currently Chief Scientist at Magellan Systems Japan and Professor of Mathematics Emeritus from California State University, Fullerton. He has operated his own consulting firm for 31 years.

Dr. Weill is also one of the three technical founders of Magellan Systems Corporation, which in 1989 produced the world's first low-cost handheld GPS receiver for the consumer market.

As an active researcher, Dr. Weill has published numerous papers on signal processing for GNSS, radar, sonar, optical sensor, and satellite communication systems. He has made substantial contributions to both the theoretical foundations and practical aspects of GNSS signal compression and multipath mitigation, and is currently developing new approaches for high performance GNSS receivers.

ABSTRACT

This paper describes the latest achievement by Magellan Systems Japan in high-sensitivity GPS technology, an ultra-sensitive software assisted GPS (AGPS) receiver for selected timing and positioning applications. A major application area is provision of accurate time for fixed Femtocell and Picocell base stations which are located indoors and typically have very weak GPS signal reception. In these applications the base station is at a fixed location and the receiver undergoes no dynamics.

The receiver is designed to provide excellent weak-signal reception at low cost and to be adaptable to many platforms. Since all signal processing is executed in software, all that is needed for full operation is an antenna, conventional RF module with a TCXO, and a host microprocessor in which the software is embedded. Extensive testing of this system with both GPS signal simulators and off-the-air signals has shown that acquisition of −163 dBm signals (C/N_0 = 11 dB-Hz) can occur when assisting with the small time and/or frequency uncertainty characteristic of typical CDMA AGPS systems. On the other hand, reliable acquisition of −160 dBm signals has been demonstrated with assisting information coming only from the internet and having much larger uncertainties. In both cases signals at −163 dBm can be tracked. Unlike many high-sensitivity receiver designs, there is no reliance on at least one relatively strong signal to enable acquisition. However, if any strong signals are found, the acquisition search strategy subsequently takes advantage of them to reduce the time to first fix (TTFF).

For high acquisition sensitivity, the receiver uses both long-term coherent processing and a final stage of non-coherent processing. The design has features which greatly reduce two well-known difficulties inherent in long-term coherent processing, which are signal-to-noise ratio (SNR) loss due to TCXO phase instability and increased false alarm probability due to a large number of frequency search cells. Additionally, the extreme sensitivity of the system has been enabled by an innovative combination of pseudorange quality algorithms.

Sufficient time transfer accuracy for Femtocell and Picocell applications (at the microsecond level) is also achieved.

The software is highly optimized to reduce computational demands imposed by the large amount of throughput required for acquisition in applications with larger time and frequency uncertainties in the assisting data.

1. INTRODUCTION

In 1998 Magellan Systems Japan (MSJ) initiated development of a super-sensitive GPS receiver with low-level signal processing performed in hardware. Completed in 2001, this receiver was capable of positioning with signals at −162 dBm and has been widely acknowledged throughout the world as the first practical indoor GPS solution. Subsequent to this pioneering work, MSJ produced the next generation of this receiver which was demonstrated at ION GNSS 2004 with the companion paper "The Next Generation of a Super Sensitive GPS System" [1]. Several core parts of this solution have been broadly used for autonomous applications as well as AGPS. Intellectual property related to MSJ's high sensitivity technology is covered by patents [2-7] and pending patents.

In the last decade there has been an increasing number of requests for semi-software and full-software solutions for applications requiring an embedded GPS receiver. Such solutions have the major advantages of lower costs and easier incorporation into the host applications. Accordingly, MSJ is working on shifting from the previous hardware solutions to software solutions in view of these market needs. This progression is illustrated in Figure 1.

This paper describes the latest achievement by MSJ in high-sensitivity GPS technology, which is a sophisticated ultra-sensitive software AGPS receiver for accurate timing and positioning applications. It is based upon more than a decade of technical experience in this field, and has been recognized as one of the best timing sources for small cell phone network base stations, otherwise known as Femtocell or Picocell in 3G, LTE, and WIMAX networks. WIMAX technology is one of the wireless communication specifications stipulated by IEEE, with relatively broad coverage. Although the system described in the paper uses GPS signals, it is readily adaptable to other GNSS systems (for example, Galileo).

Trends in the Femtocell Market

One of the key missions for the leading cell phone operators is to resolve areas where it is currently difficult or impossible for signals to reach, and which cannot be covered by the existing base stations (macro-cells) provided by the operators. The provision of Femtocells at individual houses and offices of the respective contracting parties (including corporate customers) free of charge or at low prices should reduce areas (particularly indoor ones) where it is currently difficult to use cell phones, thus making connections for mobile operation much easier. In addition, every contracting party can benefit from the reduction in communication charges. Another advantage will be the reduction of network load through the existing base-station constellation (for example, existing cell towers), thanks to automatic and seamless re-routing of calls through the internet when end users are within range of a Femtocell or Picocell base station, as illustrated in Figure 2.

Furthermore, expected MFC (Mobile-Fixed Convergence) can be stably realized with reasonable prices at a faster-than-expected pace as the adoption of Femtocells spreads widely. MFC refers to a system that has the functions and benefits of both fixed and cell phones inside one telephone set. Fixed phone services are available indoors via Femtocells. At the same time, access to macro-cell-based cell phone networks outdoors is also possible, which allows truly seamless communication networks to be established.

The Need for Accurate Timing

For cell phone networks, especially for the CDMA2000 infrastructure, sophisticated time synchronization among different relay/base stations is an indispensable technological requirement. Consequently, high-precision time data from GPS satellites is currently used to achieve network time synchronization.

2. DESIGN OBJECTIVES

The major application for which the receiver is designed is embedding in a Femtocell or Picocell base station located indoors. The purpose of the base station is to provide accurate timing at the microsecond level which will permit automatic routing of cellphone communications through the internet via the base station when the cellphone is nearby (perhaps within 100-200 meters), and seamlessly switch to normal routing through a cell tower when at greater distances from the base station. This should significantly increase the capacity of the cellular network by relieving traffic routed through the cell towers.

Since for this application the GPS receiver must function indoors, a reliable acquisition sensitivity of at least −160 dBm (C/N_0 = 14 dB-Hz) is desired. The receiver position is fixed, so the design does not have to accommodate dynamics. Generally the TTFF requirement can be more lenient than for most GPS receivers—up to tens of minutes TTFF might be acceptable for acquiring very weak signals. Single-fix positioning errors are generally allowed to be larger than in typical low-cost GPS receivers, because time is usually available for long-term averaging of multiple fixes to reduce positioning error to an acceptable level of under about 30 meters. Once receiver position has been established to the desired degree of accuracy, there is no need for further position fixes unless the receiver is moved. Subsequently, the primary function of the receiver becomes time transfer at the sub-microsecond accuracy level to enable handover between internet and cell tower routing.

The receiver must be capable of operating with assisting data, including timing and ephemeris data, obtained either from a CDMA-type network, or from an internet connection which generally has more timing uncertainty. Since the receiver is intended for high-volume production into a consumer market, low cost is a high priority. The receiver must also be easily embedded into a variety of computer platforms with acceptably small memory and processing loads. Power consumption is not an issue, because in most cases the receiver will have a fixed location within a building with readily available AC power.

3. DESCRIPTION OF BASIC SIGNAL PROCESSING

To keep costs low, the receiver is software-based and takes advantage of the computational capability available in any of a number of host microprocessors. In order to achieve the desired acquisition sensitivity, the original design used fully-coherent processing of a 2.56-second segment of signal captured in memory. Although the desired acquisition sensitivity of −160 dBm was closely approached, it was decided that more sensitivity was needed for marginal environments. This resulted in a design with noncoherent addition of two coherently processed blocks of signal.

Signal Capture

In the first step of signal processing, 4.88 seconds of digitally sampled intermediate frequency (IF) output from a GPS RF module is stored in a signal capture memory. The stored data contains all satellite signals additively superimposed. The RF module used for receiver development is a Rakon model GRM 8650 with a digitally sampled IF output centered at $128.\overline{3}$ kHz having a bandwidth of ±0.5 MHz about the center frequency. The sampling rate is $2.0458\overline{3}$ MHz. Each digital sample is complex-valued with 2 bits for I and 2 bits for Q. The stored samples are re-sampled at 1.024 MHz using nearest-neighbor interpolation and are replaced in the capture memory. Aliasing losses in re-sampling are minimized by the ±0.5 MHz bandwidth of the IF signal. Although this bandwidth is considerably smaller than the full null-to-null bandwidth of a C/A-coded GPS signal (±1 MHz), the resulting post-correlation SNR loss of only 0.6 dB and a small increase in pseudorange error were judged to be acceptable tradeoffs in lowering the sampling rate to keep memory and computational costs as low as possible.

Initial Acquisition of Satellite Signals

Figure 3 is a simplified tree-like diagram of the processing steps for initial acquisition of satellite signals. The figure shows the steps used for acquiring a single satellite (satellite *k*), in which the signal in capture memory, stored at a sampling rate of 1.024 MHz, is repeatedly accessed. Further details of the processing are shown in Figures 4a and 4b.

Coarse Doppler Search

The first dimension of the acquisition search is shown at the top of Figure 3 and consists of coarse frequency bins spaced 25 Hz apart. The number of bins searched depends primarily on the TCXO frequency uncertainty and to a lesser extent on the initial position uncertainty of the receiver. For example, a ±0.5 ppm TCXO frequency uncertainty at L-band and 100 km initial position uncertainty would result in approximately 71 coarse frequency bins being required. The 25 Hz bin spacing is chosen to give at most 0.9 dB of correlation loss (scalloping loss) across individual navigation data bits in subsequent processing.

For each coarse frequency bin searched, the contents of the capture memory are passed through a phase rotator that shifts the satellite *k* signal from its nominal $128.\overline{3}$ kHz IF frequency to baseband. The phase rotator also compensates for the estimated satellite *k* Doppler shift and Doppler rate computed from approximate receiver location and ephemeris data. The Doppler rate is included to avoid SNR loss in the very narrow bandwidth of subsequent coherent processing, and varies very slowly with position. The output of the phase rotator is stored in a phase rotator output memory for further processing.

Navigation Data Bit Boundary Search

The next dimension of the acquisition search is a coarse search to locate the approximate navigation data bit boundaries. The search is conducted at four trial locations spaced 5 msec apart. The average loss due to the 5 msec quantization is 0.5 dB. A finer quantization could have been used, but at the expense of increasing the search time.

Synchronous Summation and Correlation

Refer to Figures 3 and 4a. For each combination of coarse frequency search bin and trial bit boundary location, synchronous summation of 20 1-msec signal segments is performed on each of 244 20-millisecond blocks of signal to produce sums, each of 1-msec duration. Each such sum is correlated with a reference code to produce a 1024-point complex-valued correlation function. The result is 244 correlation functions, which are stored in a 1024 by 244 correlation matrix which spans 244 navigation bits of received signal. Each column contains a 1024-point correlation function obtained from 20 msec of signal. The correlation functions are generated by the standard fast Fourier transform (FFT) method (pointwise multiplication of the signal FFT and conjugate reference code FFT, followed by an inverse FFT).

The correlation matrix is then compensated for signal precession due to Doppler by cyclic vertical shifts of its columns. The amount of shift for each column is determined by the trial coarse signal frequency bin currently being searched. The compensated correlation matrix is maintained during all subsequent processing for the current combination of coarse frequency bin and trial bit boundary location.

Data Correlation

Refer to Figure 4b. The received signal contains 244 known navigation bits which are contained in a somewhat larger sequence of 244+B known bits obtained from an external aiding source. B is the bit uncertainty span corresponding to a time uncertainty span of 0.02B seconds. With the assisting server used for receiver development, the value of B is 4. To correlate the navigation data with the correlation function columns of the correlation matrix (data correlation), the known 244+B-bit sequence is shifted by X = 0,1, …, B bits, and for each shift value, 244 bit values (each value is ± 1) within the sequence are pointwise multiplied by each row of the correlation matrix. These multiplications are partitioned into two groups as shown in Figure 4b, and the results of 122 values each are the inputs to two 256-point FFTs which have been zero-padded with 134 zeros.

The known navigation data bit sequence is mostly from the ephemeris data in subframes 2 and 3 of the navigation data message. Because ephemeris data from different satellites is not the same, a serendipitous byproduct of the data correlation is increased immunity to false detection caused by cross-correlation of the reference code of the desired satellite with the received code of a strong signal from another satellite.

Formation of the Detection Statistic

As shown in Figure 4b, the magnitudes of corresponding output bins of the two FFTs are then summed to provide a sequence of 256 detection statistic values. However, the summation process involves a search over a few cyclic bin shifts of the FFT output magnitudes from the second FFT. These are used to mitigate SNR loss due to possible frequency wander of the TCXO from the first half of the received signal to the second half. Based on measurements conducted so far, only two shifts in each direction appear to be adequate.

A detection statistic value is computed for each combination of Doppler search frequency, one of the 4 trial navigation bit boundary positions, selected row of the correlation matrix (shown by the circled "A" in Figures 4a and 4b), bit shift of the known 244+B-bit sequence, output bin index (0-255) of the first FFT, and cyclic shift of the second FFT output bins. Whenever the value of the detection statistic exceeds the maximum value found so far, the maximum value is updated and the corresponding values of Doppler search frequency, trial bit boundary, correlation delay (i.e., row index of the correlation matrix), bit shift of the known 244+B-bit sequence, output bin index of the first FFT, and cyclic shift of the second FFT output bins are also all updated. After all combinations have been searched for a particular satellite, these resulting retained values constitute a provisional detection of the signal parameters for the satellite, which can be in error if the signal is too weak. To reduce the error, subsequent cross-checks are performed in software as described in Section 6. After detection, the correlation delay is refined by interpolation.

The 256-point FFTs provide a frequency estimate for each satellite having a resolution of 0.195 Hz. Additionally, the 134-point zero padding of these FFTs reduces what would otherwise be a worst-case scalloping loss of 3.9 dB down to a worst-case loss of 0.8 dB and an average loss of 0.3 dB.

4. THEORETICAL ACQUISITION SENSITIVITY

Lossless Processing Gain

The lossless processing gain of the receiver is computed as follows:

20-msec synchronous summations:	$10\log_{10} 20 = 13.0$ dB
1024-point correlations:	$10\log_{10} 1024 = 30.1$ dB
256-point FFTs:	$10\log_{10} 122 = 20.9$ dB
Summing of FFT output magnitudes:	$10\log_{10} 2 = \underline{3.0\text{ dB}}$
Total lossless processing gain (G_{PROC}):	**67.0 dB**

Note that the processing gain of the 256-point FFTs is realized from only the 122 signal input samples, and not the 134 zero-padding samples. Also, the loss in forming the FFT output magnitudes is not included.

Lossless SNR of Detection Statistic

The lossless SNR of the detection statistic is

$$SNR_{OUT} = P_S - P_N + G_{PROC} \quad \text{dB} \tag{1}$$

where P_S is the received signal power, P_N is the noise power, and G_{PROC} is the processing gain computed above. The SNR loss due to filtering of the signal in the RF module is not included. The noise power is the power within ±0.5 MHz of the received carrier frequency, and is given by

$$\begin{aligned} P_N &= kTB = \left(1.38\times10^{-23}\right)\left(290\right)\left(1\times10^{6}\right) \\ &= 4\times10^{-15} \text{ W or } -114 \text{ dBm} \end{aligned} \tag{2}$$

where k is Boltzmann's constant, T is temperature in degrees Kelvin (290^{o} is typically used for room temperature), and B is the bandwidth centered about the signal carrier. Expression (1) then becomes

$$\begin{aligned} SNR_{OUT} &= P_S - \left(-114\right) + 67 \\ &= P_S + 181 \text{ dB} \end{aligned} \tag{3}$$

where P_S is expressed in dBm. Thus, a −160 dBm received signal would produce a detection statistic with a SNR of 21 dB, assuming no losses.

Theoretical Lossless Initial Acquisition Sensitivity

Acquisition sensitivity depends not only on processing gain, but also on the false detection rate. The probability of false detection increases as the number of frequency/delay search cells increases. The probability of false detection was computed (not simulated) with a MATLAB program for various numbers of frequency/delay search cells, and the results are shown in Figure 5. Note that in going from one search cell to 10^8 cells, the acquisition sensitivity can drop by approximately 6 dB at the 10% false detection probability.

As an example of the number of cells searched in initial acquisition, the number of noise-independent frequency bins searched is the product of the 71 coarse frequency search bins (previously calculated for a TCXO frequency uncertainty of ±0.5 ppm and position uncertainty of ±100 km), the approximately 122 noise-independent FFT outputs, 1024 correlation delays, and an uncertainty of B = 4 bits in the navigation bit sequence. The result is N = 35,479,552 search cells, and the base-10 logarithm of this number is about 7.5. Using the curves in Figure 5 labeled $\log_{10}(N)$ = 7 and 8, it is seen that a single-satellite lossless false detection probability of 0.1 will occur at a received signal level of about −166.3 dBm.

Theoretical Initial Acquisition Sensitivity with Losses

To get a realistic estimate of receiver sensitivity, various processing losses must be taken into account. The average loss from each source as obtained from analysis, measurements, or simulation is as follows:

RF front end noise figure (including ADC):	1.7 dB
RF ±0.5 MHz filtering:	0.6 dB
Noise aliasing (resampling to 1.024 MHz):	0.2 dB
Interpolation loss (resampling to 1.024 MHz):	0.2 dB
Phase rotator loss:	0.1 dB
Doppler precession compensation:	0.6 dB
Correlation function interpolation loss:	1.1 dB
Coarse frequency search scalloping loss:	0.3 dB
Navigation bit boundary quantization loss:	0.5 dB
256-point FFT scalloping loss with zero padding:	0.3 dB
Total average loss:	**5.6 dB**

The 5.6 dB loss could have been made significantly smaller by using a finer coarse frequency search, a larger RF bandwidth, a sampling rate higher than 1.024 MHz, finer navigation bit boundary search quantization, and/or more FFT zero padding. However, the required memory and computational load would rise quickly if these steps were taken, making the receiver cost too great for the anticipated competitive consumer marketplace.

Using the lossless sensitivity of −166.3 dBm as determined above, the expected single-satellite initial acquisition sensitivity with losses is −166.3 + 5.6 = −160.7 dBm. However, since the search is conducted for more than one satellite, the higher probability of acquiring at least one satellite permits calibration of the receiver TCXO, thus narrowing the searches for the remaining satellites. Therefore, on the average the actual sensitivity can be better than −160.7 dBm. This agrees quite well with actual sensitivity measurements, which at times were at the −163 dBm level. These sensitivities were obtained with assisting data from the internet, and did not require the smaller frequency and time uncertainties obtainable from CDMA-based assisting data.

Fast Acquisition Mode

The receiver has the option of using a fast acquisition mode as the first part of the acquisition process. If a relatively strong signal is present, this significantly reduces the TTFF. On the other hand, if no satellites are found, a seamless transition is made to the slower but higher sensitivity acquisition search. In the fast acquisition mode, only 2.44 seconds of captured data are used.

5. TRACKING

Unlike most GNSS receivers, our design tracks signals by reacquisition using the same architecture that is used for initial acquisition, thus greatly simplifying the software by avoiding the usual tracking loops, as well as the logical operations for switching from search to loop operation. The only significant difference between initial acquisition and tracking is that uncertainties in time, frequency, and code phase, are much smaller for tracking. Reacquisition for tracking purposes is very rapid. Further details can be found in Section 6.

6. SOFTWARE IMPLEMENTATION

The algorithms outlined in this paper were intended for time and frequency resolution in small cell phone base station implementations (Femtocell, Picocell, etc.). Therefore, we picked a typical communications platform for our initial implementation. This consisted of a picoChip PC7202 development board with a Rakon GRM 8650 GPS module attached. This picoChip platform provided us an ARM 926 processor running at 280 MHz with a version of Linux installed.

The Rakon module provided us with time stamped (in terms of TCXO counts) GPS data sampled at $2.0458\overline{3}$ MHz (derived from a 19.2 MHz TCXO). The GPS data was presented as 2 bits I and 2 bits Q (sign and magnitude). This module streamed the data using a standard SPI interface. The ARM 926 board did not have a suitable SPI available in hardware, so we used one of the board's small DSPs to implement this interface and provide the functionality of a SPI port connected to a DMA channel (the data was placed directly into SDRAM memory accessible to the processor without processor intervention).

The algorithms were implemented in C++ with some legacy code and support routines being implemented in straight C. The algorithms were implemented in such a way that they could be run on a PC using data from files (data could be captured live or simulated). This allowed for ease of debugging and development. In addition the code used 64-bit floating point for data storage in the PC environment; this allowed us to check of any limitations developing due to optimizations used in the ARM 926 environment.

The lack of hardware floating point support in our target platform(s) made us make an initial design decision to use 16-bit integer values for data storage on the ARM 926 platform. This would result in 32-bit complex numbers (two 16-bit integers).

The project used GNU-based tools for cross-compilation onto the ARM 926 board. For debug on the PC, Borland and Microsoft tools were used.

Software Implementation Problems

When it came time to implement the algorithms outlined in this paper in a real world system, we realized that there were several large challenges ahead. The algorithms require extensive use of FFTs. For example, searching a 25 Hz frequency bin with 0.1 seconds of time ambiguity nominally requires 2304 FFTs of size 1024, 2304 inverse FFTs of size 1024, and 92,160 FFTs of size 256. This is a potentially daunting amount of throughput given our target systems being integer-only RISC based processors.

In addition to the potential throughput problems the size of the memory buffers required is nominally very large. Working with approximately 5 seconds of $2.0458\overline{3}$ MHz data would require a capture buffer of 5.1 Mbytes. This data would need to be down-sampled and converted to 32-bit complex values, resulting in 10.2 Mbytes. It would then be phase rotated to compensate for Doppler, and the resulting buffer would be 10.2 Mbytes. The various FFT buffers would add more than 1 Mbyte to this total. Therefore, we were looking at a potential total memory usage of 27 Mbytes. This was too much for a set of software that would be embedded with other software in a Femtocell-type application.

Besides memory and throughput performance issues, we faced a requirement to deliver somewhat accurate position results. Our basic pseudorange measurement granularity is 1/1024000 seconds (coming from our use of a 1024 point FFT/IFFT for code correlation). This is approximately 293 meters, and would only provide positioning to several hundred meters of accuracy. Although we had no official position accuracy requirement, we felt that we should be able to position within 100 meters using weak signals.

Finally, we faced an issue with how to determine what was "real" data and what was noise. Analysis showed that there could exist "noise peaks" in random data on the order of −157 to −158 dBm. Since we were building a system that would find signals at the −160 dB level and sometimes lower, this was quite a problem, especially in the case of the initial search when our "search space" could be very large due to time and frequency uncertainty.

Initialization Data

The algorithms described in this paper inherently need some outside data to realistically process the captured, raw GPS data. The data needed is of 4 types:

Ephemeris

The system needs ephemeris data for the purpose of computing satellite position and time of transmission. It also has to be "bit accurate" ephemeris data in order to re-create the data bits used during the data bit correlation process. Bit accurate ephemeris is decoded/parsed ephemeris data that reflects the actual data stream transmitted by the satellites. This data is a subset of the E911 data that is available on most provider networks, so obtaining it operationally is not a problem. For development and test however, it is not so readily available. Most ephemeris data available on the internet is derived from independent curve fits, and does not reflect the actual transmitted data bits (this data is meant to provide a more accurate representation of the satellite orbits). Thus, for our development we needed to build our own "ephemeris servers".

These servers turned out to be fairly easy to develop, install and maintain. They consist of a netbook computer running Linux with an attached "smart antenna" GPS receiver (containing a Sirf chipset and connected via USB cable). A simple program runs on the computer and queries the Sirf receiver for ephemeris data at regular, short, intervals. If new ephemeris data is available it writes it out in standard RINEX format to a specific file in a specific directory. Then the soft GPS software we developed just FTPs the file from the ephemeris server onto its board, then reads that file to get its ephemeris data.

Time

An approximate time is needed by the software to limit the "time dimension" of the search space. Although it is theoretically possible to build the system to search over large time uncertainties (possibly hours or days) it would require extremely large amounts of CPU time and memory to do so. We have designed the software to allow time initialization with an uncertainty of up to 2 seconds (±1 second), which is considerably more than what is actually available in most applications.

Time is obtained over the internet through the use of the NTP (Network Timing Protocol) from an NTP server. There are many such servers around the world. Using a nearby server (one within hundreds of kilometers) we typically get a time initialization within 40-80 milliseconds. This allows for a considerable reduction in the throughput needed for an initial fix since the navigation data search space is typically only 4-5 data bits. Using a time server on a different continent (which we consider to be the worst case) still provides initialization within 250 milliseconds, which is well within the capability of the software.

Frequency

The system is capable of deriving frequency from no other initialization information but the oscillator's specifications. However, this results in a large frequency search space. In the case of ±0.5 ppm and ±100 km position uncertainty, the search space for one satellite would be 71 bins of 25 Hz at L-band (with a totally uncalibrated TCXO this could be up to 10 times larger), and subsequent huge CPU throughput requirements. To reduce this search space the software accepts a frequency initialization (including uncertainty range). In our test system we use some proprietary software provided by Rakon that samples NTP packet delivery (from an NTP server) to derive an approximate oscillator frequency. Typically this approach can deliver a frequency within 100 ppb (parts per billion) in less than 20 minutes. This results in a search space of up to (and often less than) twenty-one 25 Hz bins, which is a reasonable number.

Position

An approximate position is needed by the system. Although this can be the center of the Earth (for a true "cold" start), large search times can result. For each potential error of 1 km in position the frequency search space must be increased by approximately 1 Hz. Thus, for a cold start this would add 6400 Hz to the search range. For our test and development we provide the system with an approximate position and its uncertainty. The system works well with position uncertainties up to 100 km. Beyond that, the first fix time is noticeably increased due to the larger frequency search space. Operationally, providers that we have talked to generally indicate that at installation/setup the system could easily be set to a position within 100 km (often far less) through use of the location of the town/city in which it is installed, U.S. zipcode location, or something similar.

Optimization for Speed

In order to decrease the throughput used by the software three methods were undertaken:

1. Removal of all unnecessary copying of data (as from one buffer to another).

2. Exporting all operations from loops that were not required within the loop (initialization, memory allocation/de-allocation, etc.). This included a careful analysis of the code to make sure no explicit or implicit allocation/de-allocation of C++ objects was occurring within loops.

3. Developing a highly optimized integer FFT.

The first two steps were accomplished through traditional methods; measurements of where CPU time was being used and code analysis. The optimization of the FFT required the bulk of the work in this area.

It became apparent from research and trials of various algorithms and packages that any FFT routine implemented in C or C++ was not going to have the performance that was needed. Therefore, we began researching and developing a 16-bit integer complex FFT in assembly language. The final routines that we developed were based on algorithms and routines described in [8]. This enabled us to develop a radix-4 FFT optimized for the ARM9e processing core. This routine can compute a 256-point FFT in a little over 13,000 processor cycles, giving a theoretical rate of some 15,000 FFTs per second on our target hardware where we only use 70% of the processor, on average. For the 1024-point FFTs the number of processor cycles is a little over 66,000 processor cycles for a rate of 2,900 FFTs per second. This gave us acceptable performance without any hardware acceleration. We feel that these optimizations can be adapted to any RISC platform.

Table 1 gives the measured times to obtain a fix with some representative uncertainty values. The first three columns represent first fix conditions and the last column represents a subsequent fix.

Table 1. Measurements of Search Times With Various Uncertainties

	First fix	First fix	First fix	Reacqui-sition
Time uncertainty (seconds)	0.2400	0.1000	0.0200	0.000025
Frequency uncertainty (Hertz)	325	100	50	1
Position uncertainty (kilometers)	1	1	0.50	0.50
1024-point FFT/IFFTs needed	269,600	83,000	41,500	128
256-point FFTs needed	11,513,000	1,476,000	147,600	26
Total time on ARM 926 board (minutes)	**29.74**	**9.14**	**4.95**	**0.68**

Optimization for Memory Size

Reducing our memory footprint was done in several steps. The first was to keep the data in 2 bit I/2 bit Q format after re-sampling from $2.0458\overline{3}$ MHz to 1.024 MHz. This results in almost 7.7 Mbytes of savings.

We had also intended to place this re-sampled data into the original capture buffer. However, since the memory mapping on the ARM 926 platform probably did not have optimization for this type of operation as a goal, there were severe throughput problems (access to the capture buffer by the ARM 926 processor was slow). However, if this could be done there would be another 2.6 Mbytes of savings.

Analysis showed that we did not need 32 bits to maintain precision in the phase rotated sample data. We therefore changed the output of the phase rotator to be 16-bit complex values (two 8-bit integers). These values are then converted to 32-bit complex values when they are accumulated prior to the code correlation FFT. This saved 5.1 Mbytes.

Additionally we made extensive re-use of the FFT buffers by, for example, putting FFT results into the input buffers. This saves several hundred Kbytes.

The result is that the current software has a memory footprint of less than 15 Mbytes, including the initial capture buffer. Discussions with Femtocell designers have indicated that this is within the "acceptable" range.

Validation of Pseudorange Data

The algorithms can often detect a signal and derive a pseudorange for a satellite down to −163 dBm. Unfortunately, noise can cause correlation spikes on the order of at least −158 dBm. We faced the challenge of telling one from the other. Once a fix has been computed, this is a relatively straightforward problem. Knowing the position to within a few hundred meters, the oscillator frequency to a few Hz (at L-band), and the time to tens of microseconds allows the software to do basic pseudorange and pseudorange rate residual checking to determine what "peaks" are from noise. On a first fix, however, things are quite different. There are two primary stages to the validation of the pseudorange data; pre-fix, and actual position fix processing. We use the process of determining a position fix as a key element in the validation of our data.

For pre-fix pseudorange validation we use a basic technique which maintains information for several "peaks", those with the largest power signatures, for each satellite instead of just a single maximum. With multiple choices for potential true signals for each satellite the software checks each candidate for each satellite for agreement in residual Doppler with the satellite which has the strongest signal. This test takes into account the position uncertainty to see if a given peak's data has a residual Doppler within appropriate limits from the strongest satellite signal's residual Doppler.

Although noise peaks are possible, their likelihood for a given satellite is small enough that there is a large chance that the strongest satellite signal is "true" and not noise. If this is not the case the above tests will fail for all (or a large majority) of the signals and the software will abort and gather more data for another try. Most of the time this process will yield a set of pseudorange results which use the maximum power level peak for all except 1 or 2 satellites. These other satellites will have identified one of their other search result candidates as the probable true signal. Some satellites, of course, develop no matching data as there is no signal from them being received.

Once a candidate set of pseudoranges has been determined, they are processed through a residual checking method called Multi-Fix Residual Reduction (MFRR). MFRR was developed for Magellan System Japan's autonomous GPS IP, which is part of the position fix determination process. MFRR takes the N satellites' pseudorange data and determines a series of N different fixes, each of which use N−1 satellites' pseudorange data. Once each fix is determined, the pseudorange residual for the satellite that was not used is measured. If one or more of these residuals is beyond a limit (for this application we use a limit of hundreds of meters), an appropriate pseudorange is marked "bad" and removed from the candidate set. "Appropriate" is determined by how many of the residuals are large. If it is only one, then that satellite's data is removed; if it is more than one, a determination of which satellite is "probably" causing the problems is done. The MFRR process is then repeated until the resultant residuals are small, or until there is not enough data to compute a fix.

Once a fix is computed the resulting Doppler residuals are compared (the residuals having been recomputed based on the final fix). If there is any significant disagreement the fix is declared "bad". If this is not a first fix, it is also compared to the previously determined position to do a validation against it having "jumped" too far.

Finally, we maintain the concept of "trusted" and "untrusted" position fixes. The position solution does not become "trusted" until it meets the criteria of being over-determined (more than 4 satellites, all of which have signal levels above our nominal noise peak level). If we are getting "untrusted" fixes the solution can also become "trusted" after several consecutive fixes that give the same position. Certain validation tests on the pseudorange data are not performed until the position solution is "trusted". This keeps the software from rejecting good data if it gets a "bad" fix. Additionally an "untrusted" position solution is assigned a higher uncertainty level (larger error circle).

The above algorithms allow the software to determine, with a high level of reliability, which signals are noise and which are real GPS signals. This allows the receiver to get a first fix at or below signal levels of −160 dBm. Table 2 shows some test results for maximum sensitivity of first fix (given a single pass/collection of GPS data, 1 km position uncertainty and 0.08 seconds of time uncertainty with signals generated by a NavLabs multi-channel GPS simulator).

Table 2. Test Results for Maximum Sensitivity of First Fix

Frequency uncertainty at L-Band (Hertz)	80	80	395	395
Signal strength (dBm)	−159	−162	−159	−160
First fix success rate (single set of data)	100%	20%	100%	60%

Epoch Ambiguity Resolution

The ability to resolve the proper code epoch (millisecond resolution of satellite transmission time) proved difficult with this system. The low signal levels gave poor discrimination (in signal power) between adjacent epochs. The software added 2 algorithms to work around this problem.

The first thing that is done is modification of all the pseudorange data to correspond to a "reference" epoch. The "reference" epoch is determined from one of three different possibilities:

1. The previously determined "correct" epoch (see below)

2. The signal with the highest power level, if its SNR is higher than that needed to have unambiguously determined the correct epoch.

3. The common epoch resolution which has been determined from at least 3 satellite signals.

The second algorithm is to determine the correct epoch via either a strong signal, or through multiple fixes that give the same epoch resolution. The software therefore marks the position/time solution with a flag indicating whether the epoch has been correctly determined. When the epoch is correctly determined it remains determined through subsequent fixes.

Before the epoch is correctly determined the system can still operate, but with limitations. In such a case the epoch ambiguity error will be common to all satellites and therefore have no observable effect on the derived position and derived oscillator frequency error. The derived time may be in error by 1-2 milliseconds. So, when in this condition, the software will not report a time uncertainty of less than 3 milliseconds.

7. SOLUTION ACCURACY

The system is intended to be used to aid small cell phone base stations (Femtocells, etc.), which are assumed to be stationary. As such, we are most interested in the frequency (in order to correct the primary oscillator's frequency error) and time (for network synchronization) components of our GPS solution. In addition, position is desired to make sure the cell site has not been moved and is operating in the provider's licensing area.

Frequency

Measurements have shown that the system provides corrections to the oscillator that are accurate to better than 1 ppb (part per billion). This leads to a sustained frequency accuracy of 1 ppb. "Sustained" means that the oscillator will hold to this accuracy from time of correction to the availability of the next correction (90 seconds). Figure 6 shows the typical frequency corrections.

Time

The time solution is typically accurate to better than 0.5 microseconds at the time of the fix, leading to a sustained time accuracy of 1-2 microseconds. "Sustained" meaning that time derived from the oscillator will hold to this accuracy from time of correction to the availability of the next correction (90 seconds). We feel that with proper averaging/conditioning that a sustained accuracy of 0.5 microseconds is possible, although we have not, as yet, undertaken this effort. Figure 7 shows the typical time/clock correction history. This shows a distribution of the magnitudes of the corrections of 1.6 microseconds (1 sigma) and does not represent any smoothing/filtering of the data which could be applied in an operational system to obtain more accurate tracking of time between corrections.

Position

The system's pseudorange resolution is approximately 293 meters, which is caused by the 1024 point FFT/IFFT used for PRN code correlation. When PDOP and a factor that we include to reflect probable multipath problems are taken into account (this being an indoor system, it will almost always have bad multipath), position uncertainties on the order of kilometers can result. It was felt that we had to provide a more accurate position (many cell phone providers want position so that they can be sure they are operating in allowed areas for regulatory purposes).

An interpolation scheme was added to the software that allows us to halve the resolution for signals of non-weak (such as those at −155 dBm and above) satellites. Additionally, a Doppler-based correction was added that also halved our average pseudorange resolution. The result was a resolution of about 150 meters for weak signals and 70-80 meters for non-weak signals. This resulted in position uncertainties on the order of 100-200 meters when the use of over-determined position solutions was included (the software uses pseudoranges from all the satellites it can acquire, often 7 or more).

In addition to all of this, we average the positions as they are determined. This is possible because of the stationary nature of our environment and our feeling that the multipath problems will average out (in general) over a period of several hours.

The dependence of positioning performance on averaging using actual signals received in the U.S. can be seen in Figures 8a and 8b. These are plots of the horizontal position error of the averaged position produced by the system using an indoor antenna (active patch antenna located on the author's desk). The first figure (8a) shows the results from a running average (alpha/beta filter using a minimum of 1/40 for beta). This shows that even with a running average position accuracy of 100 meters is achievable in less than an hour (after the initial fix, the system provides position updates at a rate of approximately one every 90 seconds on our hardware platform), and potentially 50 meter accuracy with proper filter tuning. The second figure (8b) shows that with a continuous average the accuracy is within 50 meters at 2 hours and better than 30 meters after 6 hours.

Positioning performance tests with actual signals using the running average described above were also conducted in Japan. Figure 9a is a scatterplot of positions obtained with the receiver located at a central spot on the 5th floor of a 6-story solid concrete building, the Magellan Systems Japan headquarters in Osaka, Japan. Figure 9b is a tally of the signal levels from the satellites that were used. It can be seen that there were many weak signals (from −156 dBm to −163 dBm).

8. SUMMARY

The work presented in this paper has resulted in a practical low-cost software AGPS receiver with very high sensitivity primarily designed for Femtocell and Picocell applications. It can operate with all assisting data (including timing) coming solely from an internet connection. Extensive testing with real and simulated signals has shown that reliable position fixes with −160 dBm signals and tracking to −163 dBm can be achieved without requiring at least one strong satellite signal. The receiver can provide positioning accuracy of 30 meters and timing accuracy at the microsecond level.

9. REFERENCES

1. L. R. Weill, Nobuhiro Kishimoto, Seiichiro Hirata, and Kevin Xinhua Chin, "The Next Generation of a Super Sensitive GPS System," *Proceedings of ION GNSS-2004*, Long Beach, CA, September 2004, pp. 1924-1935.

2. S. Hirata and S. Lyusin, "GPS Position Measuring System and GPS Position Measuring Apparatus," U.S Patent No. 6,329,946, December 11, 2001.

3. S. Hirata and S. Lyusin, "Global Positioning System and Global Positioning Apparatus," U.S Patent No. 6,483,457, November 19, 2002.

4. S. Hirata and S. Lyusin, "Global Positioning System and Global Positioning Apparatus," U.S Patent No. 6,674,402, January 6, 2004.

5. L. Weill, "Process for Accurate Location Determination in GPS Positioning System," U.S Patent No. 7,362,265, April 22, 2008.

6. N. Kishimoto and S. Hirata, "GPS Position Measuring Method and GPS Position Measuring System," U.S Patent No. 7,408,504, August 5, 2008.

7. N. Kishimoto and S. Hirata, "Method and System for GPS Position Measuring and Frequency Error Detecting," U.S Patent No. 7,528,771, May 5, 2009.

8. A. Sloss, D. Symes, and C. Wright, *ARM System Developer's Guide, Designing and Optimizing System Software*, Morgan Kaufmann Publishers/Elsevier, 2004, ISBN-13: 978-1-55860-874-0.

10. FIGURES

Figures are shown on the following pages.

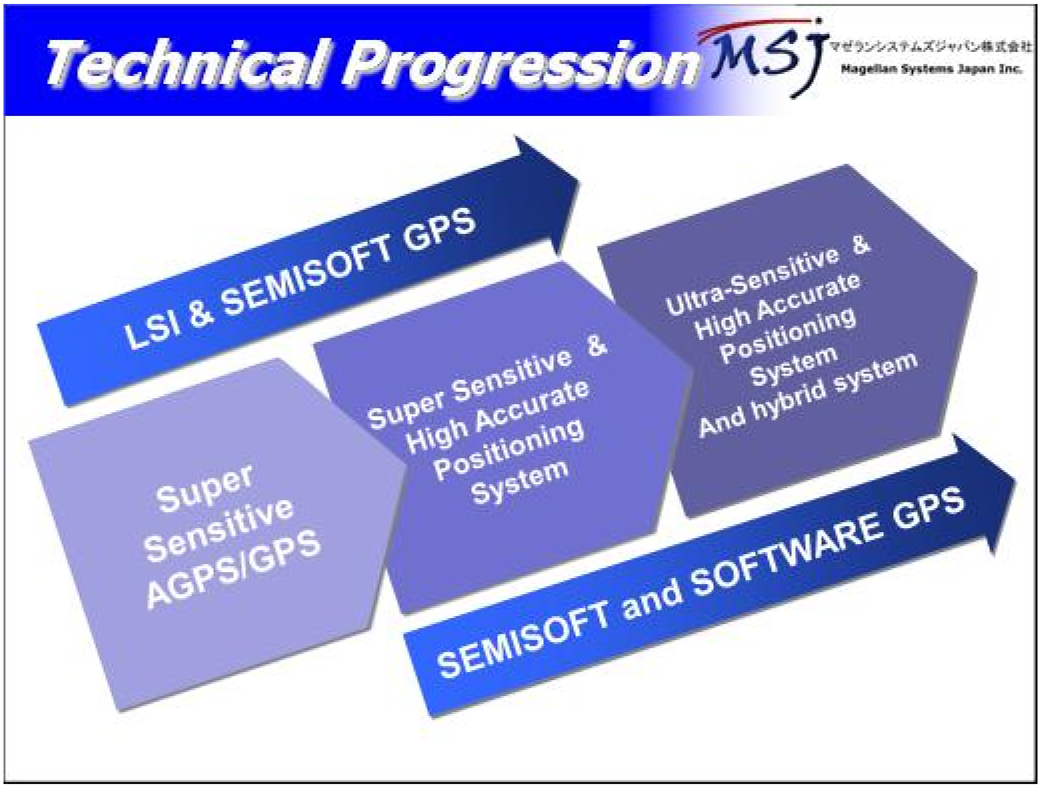
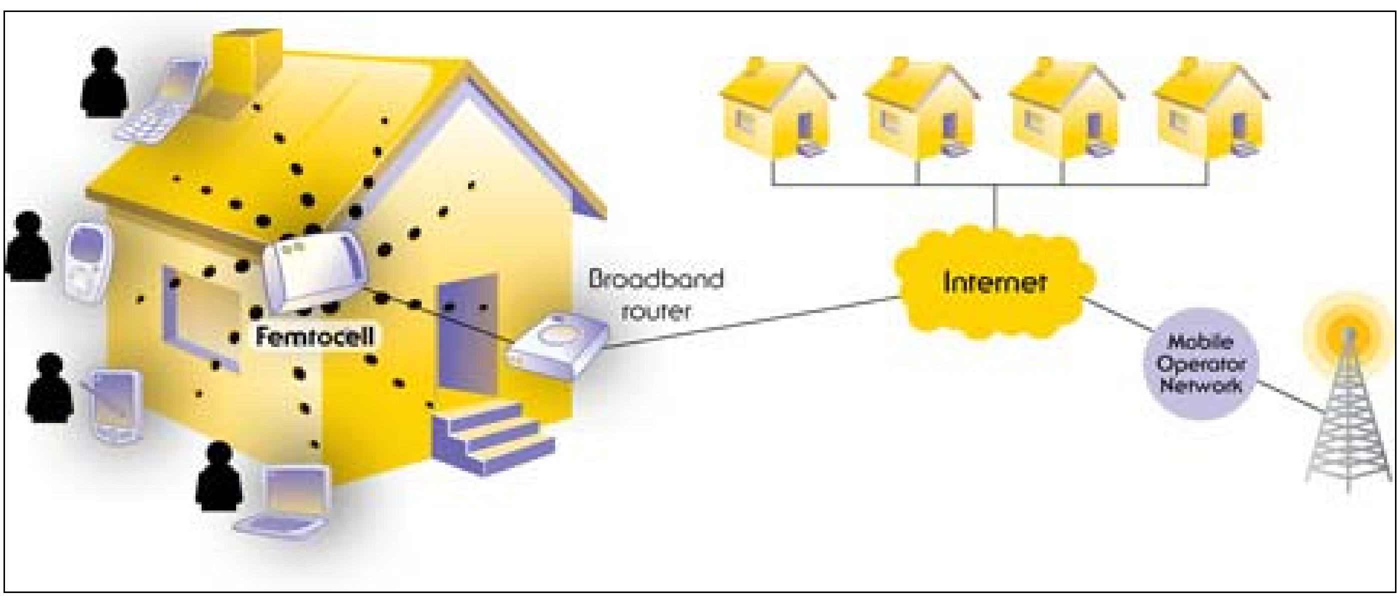


Figure 1. Progression Toward Software GPS



http://www.femtoforum.org/femto/aboutfemtocells.php

Figure 2. Femtocell Concept

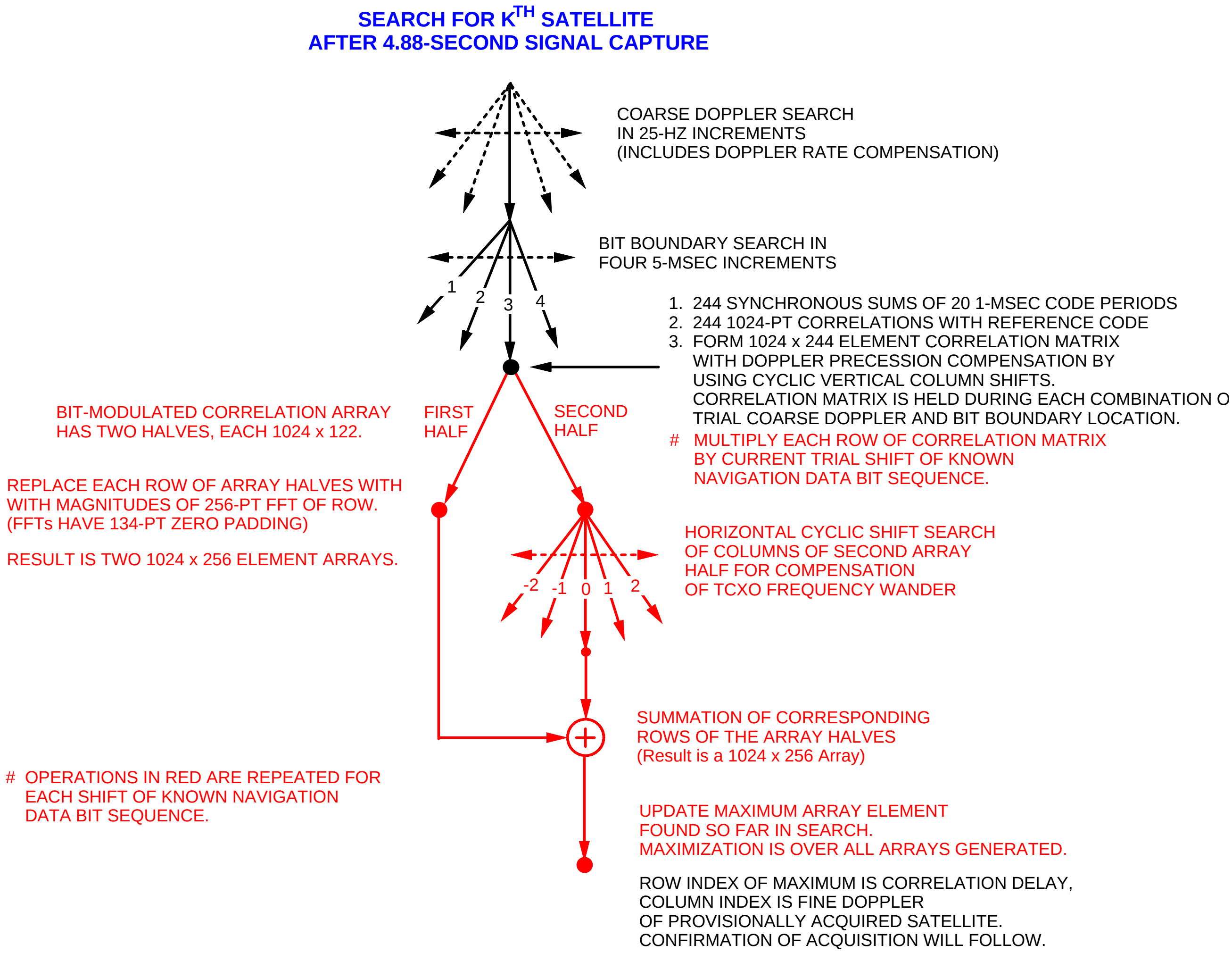


Figure 3. Acquisition Signal Processing

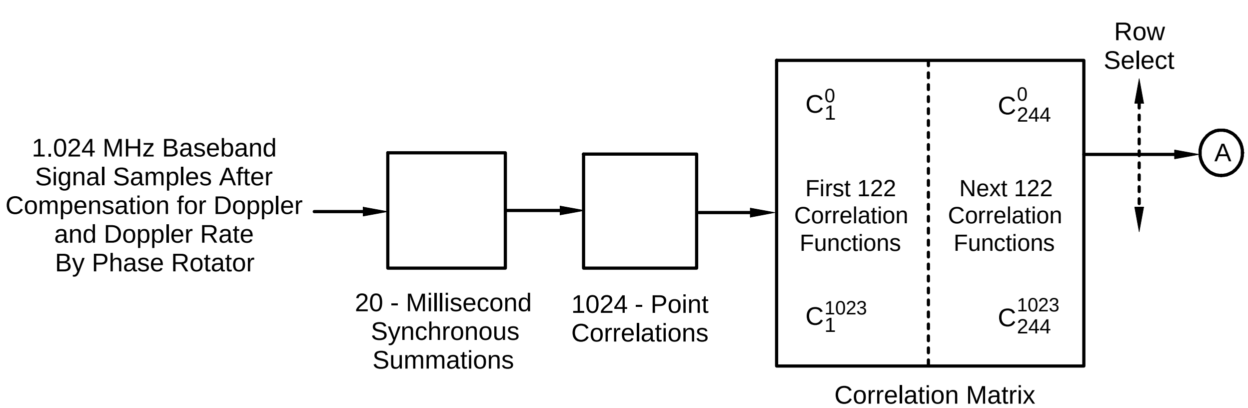


Figure 4a. Synchronous Summation and 1024-Point Correlations

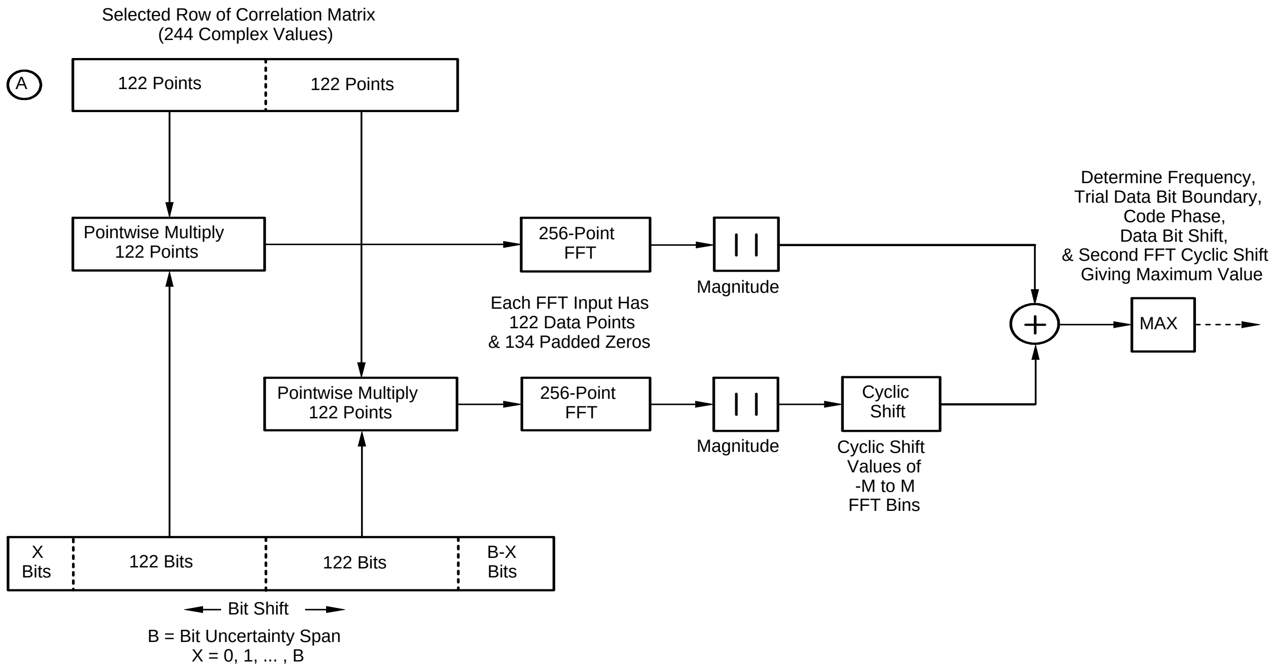


Figure 4b. Data Correlation, 256-Point FFT Processing, and Signal Detection

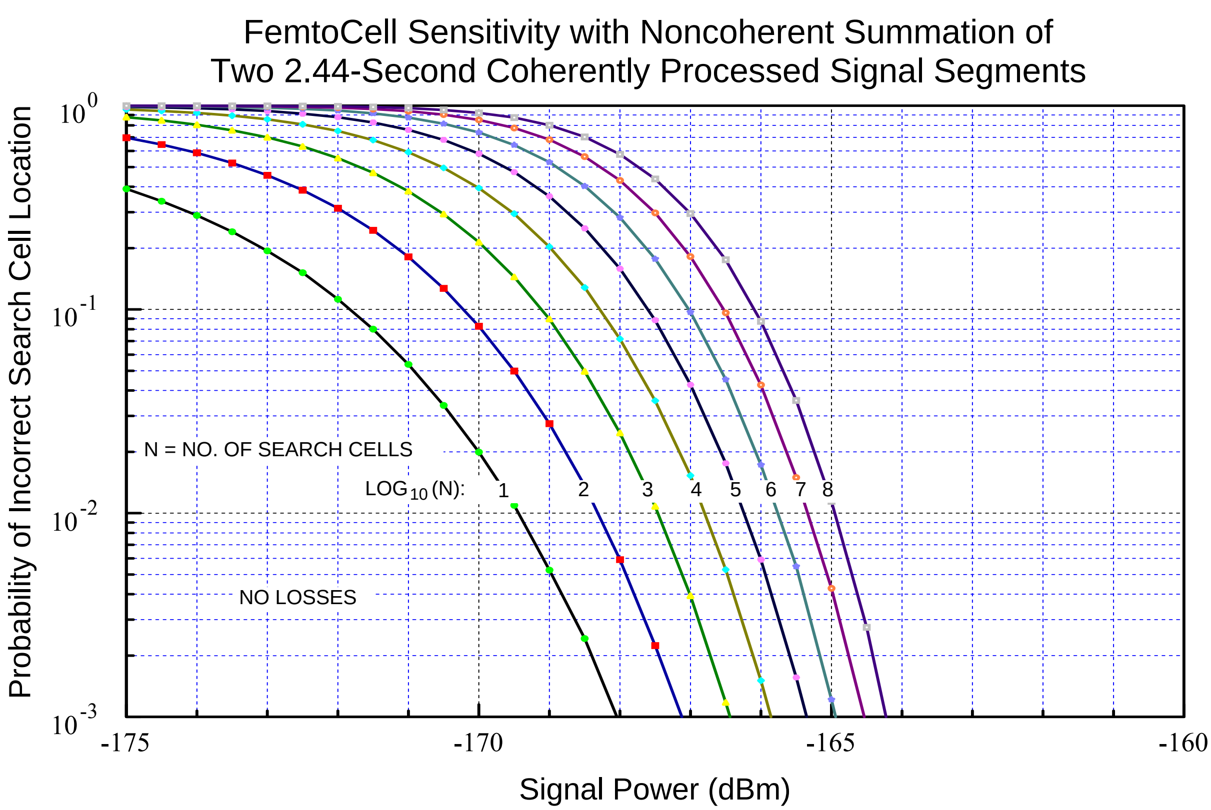


Figure 5. Theoretical Sensitivity Parameterized by Number of Search Cells

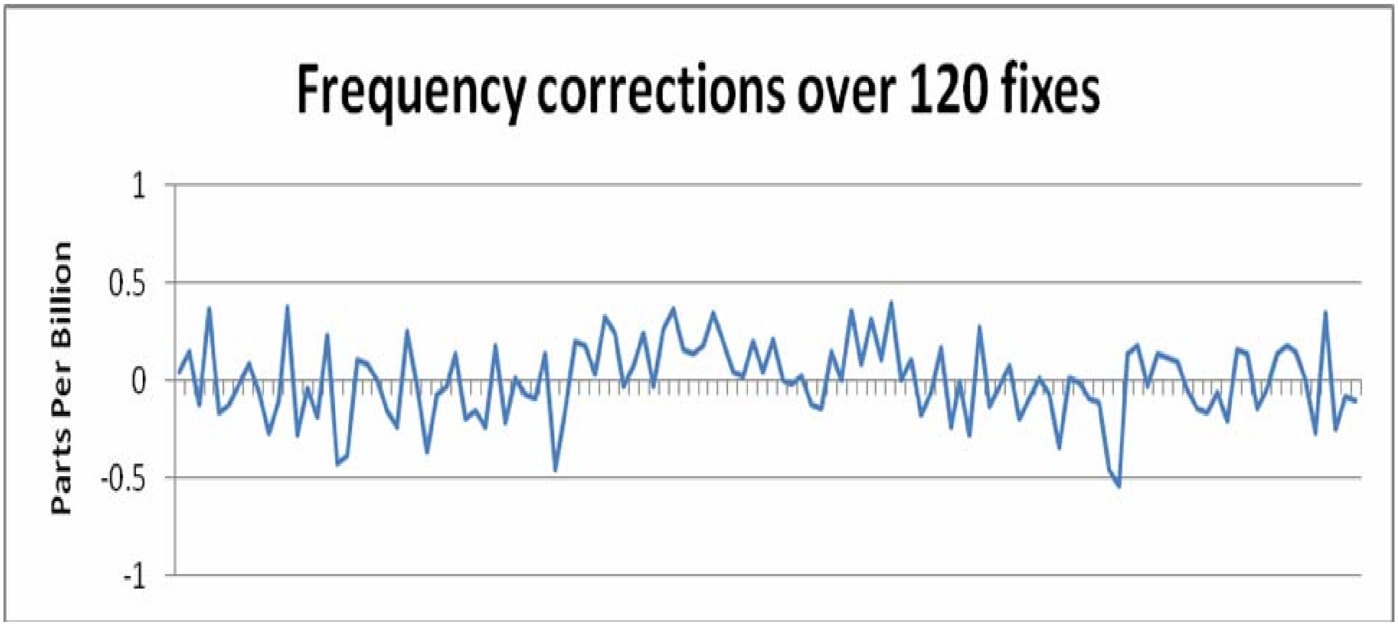


Figure 6. Frequency Corrections

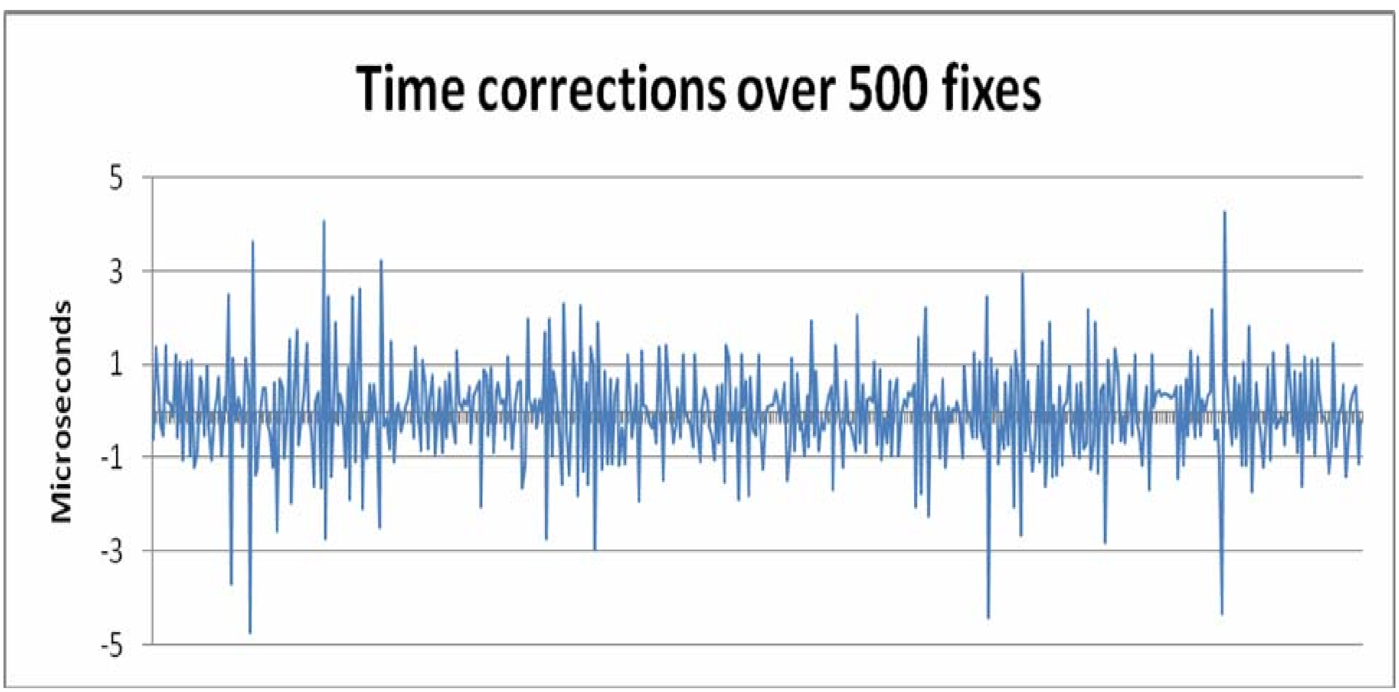


Figure 7. Time Corrections

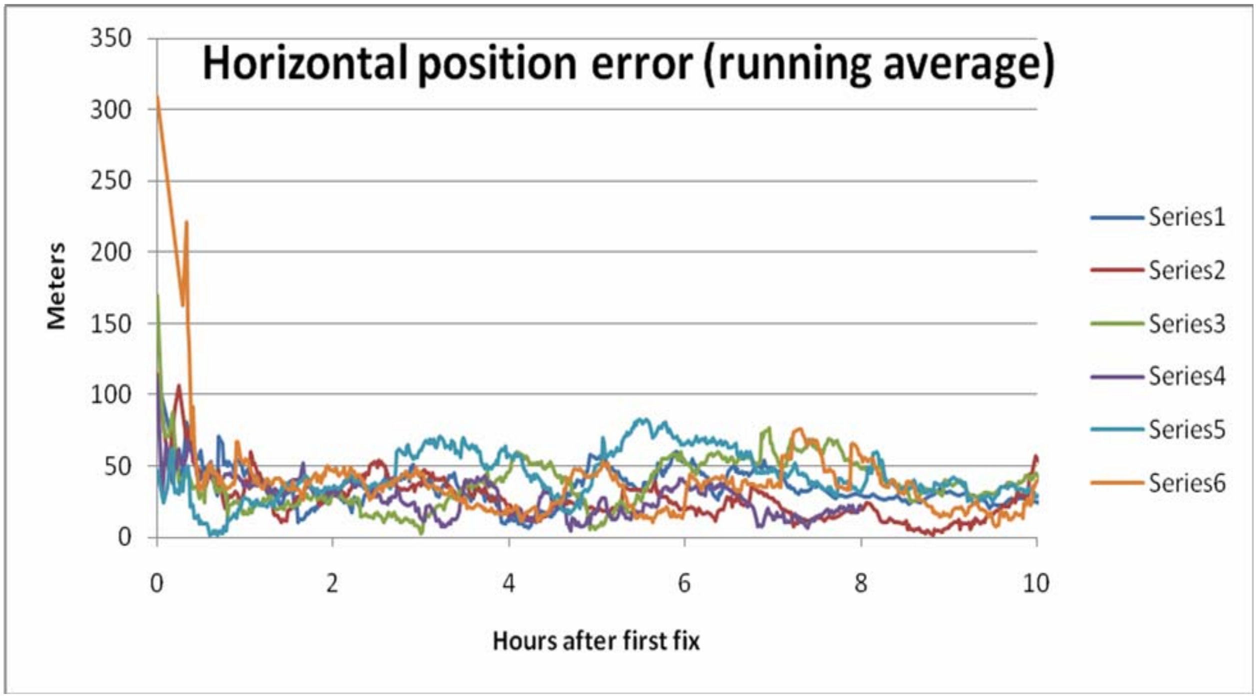


Figure 8a. Horizontal Positioning Error with Running Averaging

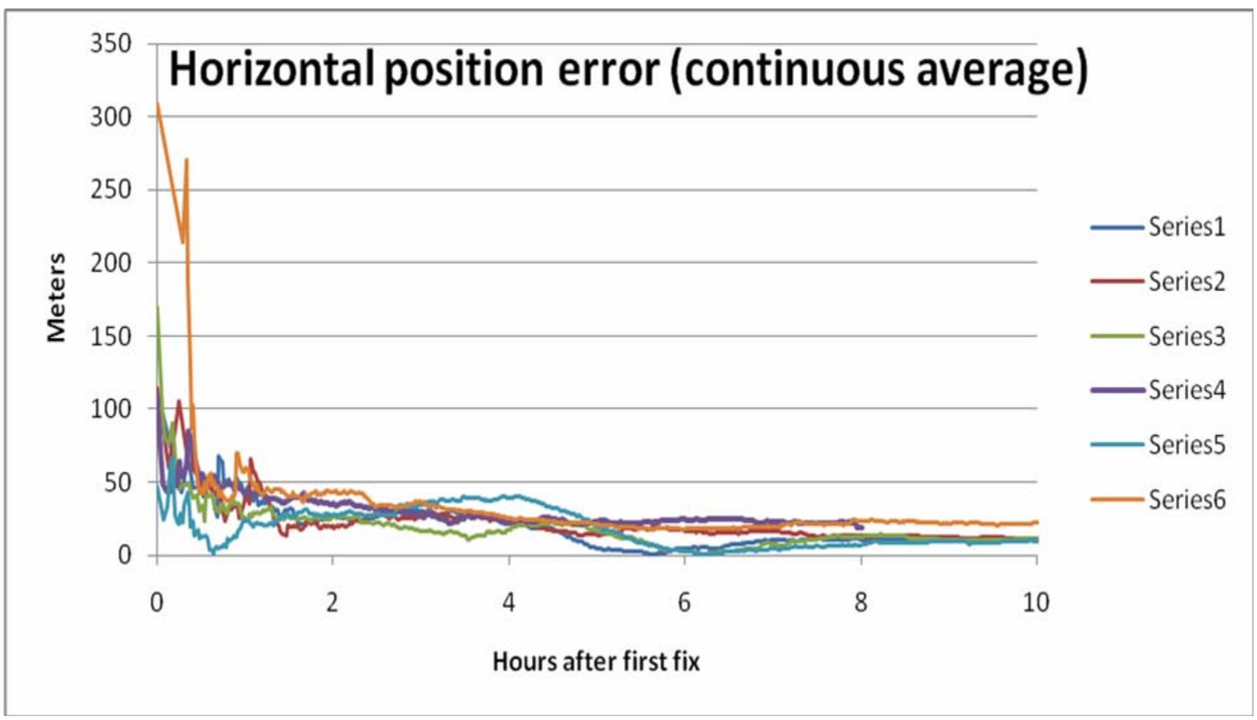


Figure 8b. Horizontal Positioning Error with Continuous Averaging

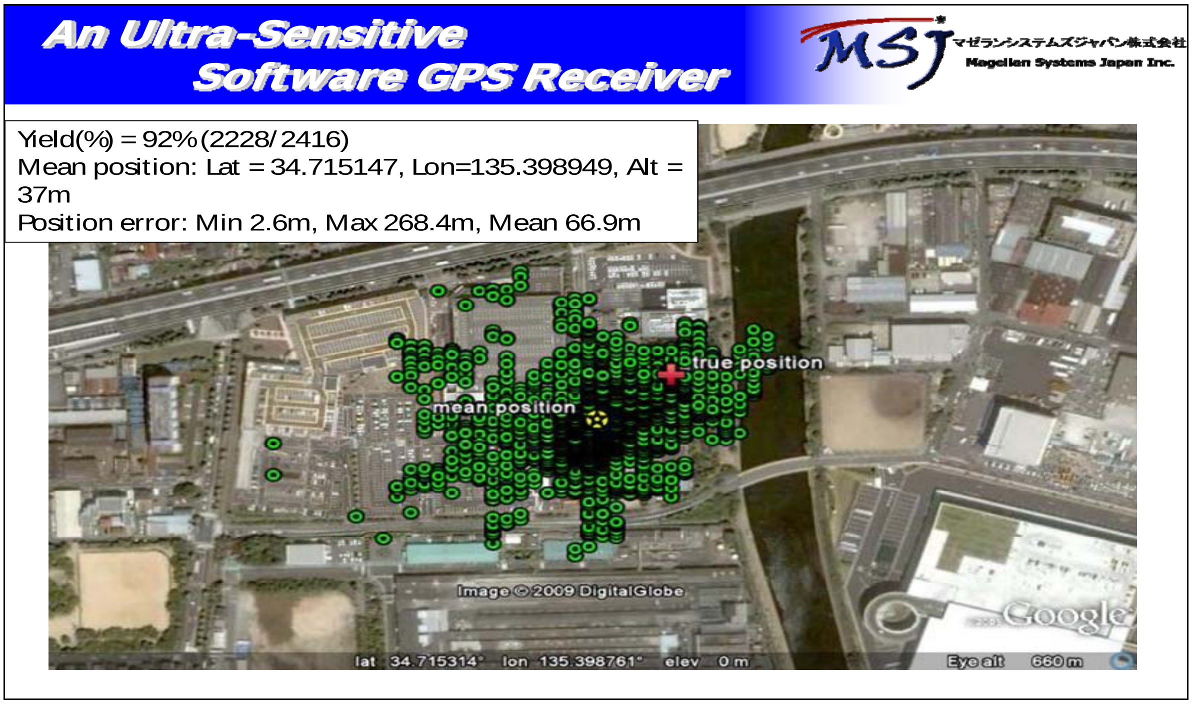


Figure 9a. Scatterplot of Positions Obtained in Osaka, Japan

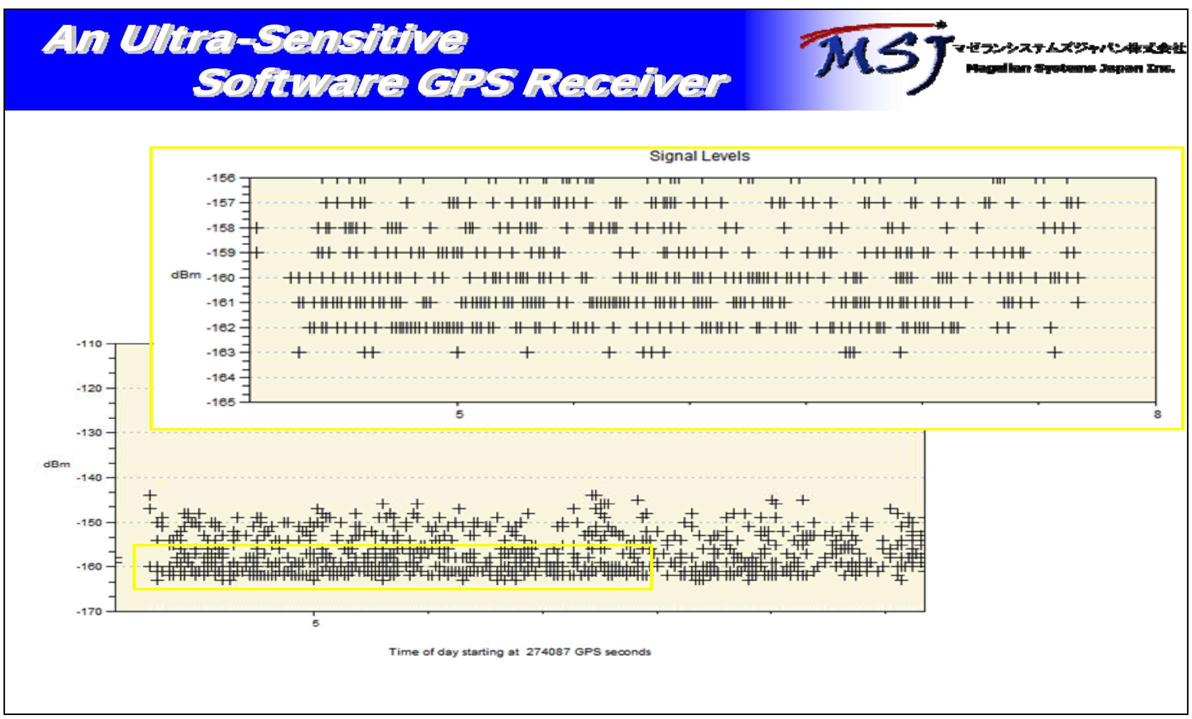


Figure 9b. Signal Levels for Positions in Figure 9a