

WAAS MOPS: Practical Examples

Todd Walter
Stanford University

ABSTRACT

The WAAS MOPS describes the method by which a Wide Area Augmentation System (WAAS) transmits its differential GPS corrections and integrity information to users in 250 bit messages. These messages must be decoded and interpreted every second. The corrections even for a single satellite are distributed across several individual messages. The corrections for individual satellites must be combined with receiver measurements and other local information to form the navigation solution and protection bounds. The user must reconstruct and apply all of this information correctly to achieve the required integrity level of 10^{-7} .

While the rules for coordinating these messages are defined in the Minimum Operational Performance Standards (MOPS), it is still helpful to provide some background, describe the intent, and show some examples. This paper provides examples of what the message order might look like and how the information should be interpreted. Both nominal and alarm situations are presented. Particular attention is paid to situations in which the user has missed some of the broadcast messages, but still has enough information to continue the procedure.

Better understanding the intent of the authors will further assist the user implementation of the MOPS. The MOPS is a large and fairly involved document. To new users in the process of implementing this standard, its rules may seem arbitrary. By showing how the different messages interact and work to protect integrity, the new user may more readily understand how to implement the MOPS compliant messages.

INTRODUCTION

The Wide Area Augmentation System (WAAS) Minimum Operational Performance Standards (MOPS) [1] is a large document written by committee to describe a complicated system. It has evolved slowly over time and

some of the nomenclature and writing reflects a history of ideas and approaches. As such, it can be an intimidating document to the new initiate. This paper is intended to provide some concrete examples to assist the reader in understanding how the different message types connect together to form a differential GPS correction. Only the interpretation and application of the messages themselves is investigated here. The MOPS cover numerous other issues which will not be addressed. This paper cannot substitute for the MOPS, and will be of little value to a reader not already familiar with Appendix A of the MOPS. Any discrepancy between the MOPS and this paper must be resolved in favor of the MOPS. The examples are intended to aid in the interpretation of the application of the corrections and not to take credit/blame for the contents of the MOPS.

The MOPS had several restrictions imposed on it for the definition of the messages, many of which apply specifically to precision approach [2]. These include:

- A severe bandwidth limitation of 250 bits per second.
- Stringent user integrity: $< 10^{-7}$ chance of receiving Hazardously Misleading Information (HMI).
- 6 second time to alarm for any failure that could lead to HMI.
- High availability, the system should be usable $> 99.9\%$ of the time.
- Support for global or near global coverage.
- Flexibility to support different service providers.

The MOPS was produced by a collaborative effort; many people contributed to its contents. It is remarkable that agreement could be achieved on a format that met all of these restrictions, especially when much of the framework of the MOPS was created before there was significant supporting data from the National Satellite Test Bed (NSTB) [3] [4] [5]. However, as with any joint venture no one person is entirely satisfied with the outcome. There is no doubt that these goals could be achieved in different ways. This author is not interested in debating the merits of one method versus another. Results have been put into the MOPS after laborious debate in which

authors have satisfied themselves that the methods meet the stringent requirements.

The corrections are broken into two categories: clock-ephemeris corrections and ionospheric corrections. The most confusing aspect of the clock and ephemeris corrections is that the information is distributed among many message types all of which must be tied together correctly. The most difficult aspect of the ionospheric corrections is that the user must determine the correct pieces to use and how to translate them to the specific user conditions.

MESSAGE FORMAT

The messages come once per second and contain 212 bits of correction data. Eight additional bits are used for acquisition and synchronization, 6 more bits to identify the message type and the remaining 24 bits designated for parity to protect against the use of corrupted data (see [2] and A.4.3 of [1]). Complementary message types must be stored and connected to the other individual components to form a single correction and confidence bound per satellite.

Because of the constraints, the bits must be used as efficiently as possible. Thus they are sometimes used for more than one purpose. Often, meeting bandwidth requirements took precedence over ease of implementation or clarity. The overriding consideration is integrity. However, integrity must be considered together with availability [6]. In order to maintain integrity and provide availability, some extra actions are required [7] [8]. Some of the corrections require additional effort to construct so as to guarantee integrity to all users.

Message Time-Out Periods

Most messages have an associated time-out period. These time-out periods prevent old data from being used beyond the time which it is valid. Some of these time-out periods are constant values (see Table A-24 of [1]) and some are broadcast to the user in the messages (see Table A-8 of [1]). For Category I precision approach, the time-out intervals are set so that users may miss one of any individual message, and still form a navigation solution using older messages. For Non-Precision Approach (NPA) and En Route (ER) phases of flight, in many cases, a user can miss two in a row of any individual messages. Thus users can continue to operate even if they do not have the latest corrections.

The service provider cannot know if a particular user has received each of the most recent messages and if not, which ones were lost. Therefore the format must protect these users without penalizing users who have not missed

any messages. Hence different rules govern the implementation of corrections and confidences when one is missing data. This can potentially lead to confusion.

The time of applicability of each message is defined to be coincident with the first bit of the message. However, the time-out period begins with the last bit of the message. Since, the message is one second in length, the time-out periods must be increased by one second when they are being compared to the time of applicability.

Issue Of Data (IOD)

Because information must be coordinated across different messages and with the information broadcast from the GPS satellites, there must be a way to identify which data may be used in combination. Issue numbers, termed Issue Of Data (IOD), must match between messages splitting information. When the IODs do not agree, the user knows that they are missing crucial pieces of information. This maintains the high level of integrity mandated by the system.

There are at least five defined types of IOD. On the unaugmented GPS system there are IODs to coordinate clock (IODC) and ephemeris (IODE) information. Each satellite has its own individual values [9]. The IODE represents the 8 least significant bits of the IODC. The IODE also enables the WAAS service provider to uniquely identify which ephemeris information is being corrected. The user must ensure that the IODE in the WAAS correction matches that in the GPS satellite broadcast.

Within the MOPS messages there is an IODP which allows the user to uniquely match the PRN number of the satellite being corrected to the location of the corrections and bounds in the messages. The IODF allows the integrity information in Message Type 6 to be traced back to a specific fast correction. The IODF also serves another purpose. It increments by one from one fast correction to the next, modulo 3. Thus, a user can detect when they have missed a message because the IODFs will not be sequential for the received messages. By determining that they are missing information the user can then take the prescribed steps to ensure that their integrity bound sufficiently covers the error. This will be discussed in further detail below. Finally, the IODF is used to signal an alarm condition. If the service provider should discover that information already broadcast would fail to protect all users, they must alert everyone to this fact. This situation is indicated by having $IODF = 3$.

The IODI allows latitude and longitude to be mapped onto the ionospheric correction information. If the information were not divided in this manner, it would be impossible to fit the data into the 250 bps bandwidth.

Nor could the information be recombined with sufficient integrity.

CLOCK AND EPHEMERIS CORRECTIONS

The satellite clock and ephemeris errors are corrected and bounded by information in Message Types 1-7, 9, 10, 17, 24, 25, and 27. The corrections are split into two types: fast corrections, which are scalar values common to all users; and slow corrections which are in the form of a four dimensional vector and affect users differently depending on their location. Most of the common errors, particularly Selective Availability (SA), are removed by the fast correction. The slow corrections primarily remove the satellite ephemeris errors. In addition they model the satellite's slowly varying clock component. Any discontinuities between one set of ephemeris parameters broadcast from a GPS satellite and the next set are also incorporated into the slow corrections. This is done for two reasons: to keep the fast corrections continuous, and to match specific ephemeris parameters, since only the slow corrections include the IODE.

The confidence bound on the clock and ephemeris corrections is called the User Differential Range Error (UDRE) and is broadcast to the user in the form of a bounding variance. Although it is called an error and has the form of a variance, it should be viewed as a statistical confidence bound [6].

Fast Correction Calculation

Message Types 2 - 5 and 24 contain fast corrections. These corrections primarily remove the error caused by SA. The pseudorange correction for only an individual epoch is broadcast. Users update these corrections over time by applying a range rate correction term formed from recent corrections. The range rate correction, RRC_n , is determined by differencing a newer fast clock correction with an older one (see A.4.4.3 of [1]) and dividing by the difference between the time of arrival, t_n , of the newer pseudorange correction, PRC_n , and the time of arrival of the older one, t_o :

$$RRC_n = \frac{PRC_n - PRC_o}{t_n - t_o} \quad (1)$$

The pseudorange correction as a function of time is given by

$$PRC(t) = PRC_n + RRC_n \cdot (t - t_n) \quad (2)$$

This correction is added to the user's measured pseudorange to remove the fast clock error. For precision approach, this correction is valid from time $t_n + 1$, when

the message is fully decoded, until the message times out, as specified by the fast correction time-out interval, I_{fc} , in Message Type 7, at time $t_n + I_{fc} + 1$ (see A.4.4.5 of [1]).

UDRE Degradation

The fast correction was estimated by the master station at some previous time, $t_n - t_l$, where t_l is the system latency time provided in Message Type 7. The bounding variance in the pseudorange correction includes estimation errors for fast clock, slow clock, and the projection of the ephemeris offset, quantization errors in those terms and some of the errors from propagating the correction forward in time, and is given by the bounding variance, σ_{UDRE}^2 , which is broadcast in Message Types 2 - 6 and 24. To be applicable for the present time, the uncertainty of the pseudorange correction must be updated by the remaining errors resulting from propagating the corrections forward in time.

Any error in the user's estimate of the range error rate will affect the confidence in the forward propagation of the correction. Calculation of the range rate correction in (1) has two primary error sources: quickly varying errors in the fast corrections estimates, such as quantization, and neglected higher order terms such as SA acceleration. The first error source is described by:

$$\frac{B_{rrc}}{t_n - t_o} \quad (3)$$

where B_{rrc} is a value greater or equal to the quantization error. The service provider estimates this term and broadcasts it to the user via Message Type 10 [1] [8].

The second error source can be described by the first neglected term in the linear model of the range error. The bounding acceleration estimate is called the Fast correction degradation factor, a , and is broadcast in Message Type 7. The range rate correction estimate formed in (1) is the best estimate for a time midway between the newer and older corrections, $(t_n + t_o) / 2$. Thus, even at t_n , the rate calculated in (1) is already out of date. In addition, the system latency, t_l , also adds to the uncertainty in the estimated range rate correction. Thus this second error source takes the form of:

$$a \cdot \left(t - \frac{t_n + t_o}{2} + t_l \right) \quad (4)$$

Both (3) and (4) are uncertainties on range rate. Their respective contributions to uncertainty on pseudorange correction are:

$$\frac{B_{rrc}}{t_n - t_o} \cdot (t - t_n) \quad (5)$$

and

$$\frac{a}{2} \cdot (t - t_n + t_1) \cdot (t - t_o + t_1) \quad (6)$$

The MOPS specifies the degradation on the fast correction to be modeled as

$$\frac{a}{2} \cdot (t - t_n + t_1)^2 \quad (7)$$

which is very similar to (6). When the true SA acceleration is very large, the broadcast value of σ_{UDRE}^2 must be increased to account for the discrepancy between (6) and (7) at the message time-out period.

The derivation above assumes that no data is missing. When the user fails to receive a fast correction, they will continue to apply the above corrections until a new fast correction is received or the old information times out. Upon receipt of the new fast correction there will be a break in the sequence of the IODFs indicating a missed message. The user continues to use (1) and (2) to generate the correction. When missing the previous fast correction, one must add an extra term to account for additional latency and quantization errors in the RRC_n . Provided neither of the messages indicate an alarm condition, this term is

$$\varepsilon_{rrc} \equiv \left(\frac{a \cdot I_{fc}}{4} + \frac{B_{rrc}}{t_n - t_o} \right) \cdot (t - t_n) \quad (8)$$

Alarm conditions are indicated when the IODF of either message is set to 3. In this case, ε_{rrc} has a slightly different form as given by (A-49) of [1].

Under alarm conditions, fast clock corrections may be as closely spaced as 1 second apart. A user who differences adjacent or closely spaced fast clock errors will suffer greater σ_{UDRE}^2 degradation (and consequently lower availability and higher risk of lost continuity) than a user who differences fast clock error terms with nominal or close to nominal spacing [8]. User algorithms will have better performance taking this latter approach, although it is not mandated. In addition the message may not be forward propagated more than eight times the time difference used to form the RRC_n . Thus two messages one second apart may only be used eight seconds into the future before the uncertainty grows too large to continue.

Message Type 6

The above discussion assumed that the service provider was broadcasting in a mode where the fast corrections and the σ_{UDRE}^2 always arrive in the same messages. This is the expected case for when SA is turned on. However, in the future, when SA is turned off, it will not be necessary to send the fast corrections as often as they are sent currently. Significant bandwidth can be gained by sending these messages less frequently. The extra bandwidth can be used to improve performance by sending other messages more often or by defining future message types.

It will still be necessary to update the integrity status at the high rate. Thus, it is desirable to send the σ_{UDRE}^2 values every six seconds. Message Type 6 allows these values to be sent without also retransmitting the fast corrections. Message Type 6 contains σ_{UDRE}^2 values for all satellites that are being corrected (up to 51). Thus, one could receive a subset of Message Types 2-5 and 24 at one minute intervals and Message Type 6 every six seconds in between. Equations (1) and (2) remain valid. However, (7) will have to be updated as follows to allow for this possibility.

The receipt of Message Type 6 has the effect of retransmitting the last broadcast fast correction messages. It tells the user that the last fast correction broadcasts are still valid and can be used for another twelve seconds. This is the other and more important role of the IODF which increments for every new fast correction. Every Type 6 message sent in between successive fast corrections will have IODFs corresponding to the value in the last fast correction broadcast. Thus users will know unambiguously which fast corrections the Message Type 6 is updating.

The new equation for updating σ_{UDRE}^2 is unchanged in form from (7) but has one crucial difference. The degradation starts from the time of applicability, t_{UDRE} , of the last received σ_{UDRE}^2 with matching IODF < 3 . If Message Types 2-5 or 24 are the most recent source of σ_{UDRE}^2 then $t_{UDRE} = t_n$. If the most recent source is from Message Type 6 with matching IODF < 3 then $t_{UDRE} = t_6$ where t_6 is the time of applicability of that message. Finally, if the IODF in either message is set to three then $t_{UDRE} = t_n$ and is not updated to the time of the Type 6 message. Because the receipt of Message Type 6 with matching IODF < 3 is equivalent to another reception of the last fast correction, the Fast correction degradation term can be reset and t_n in (7) can be replaced with t_{UDRE} resulting in

$$\varepsilon_{fc} \equiv \frac{a}{2} \cdot (t - t_{UDRE} + t_1)^2 \quad (9)$$

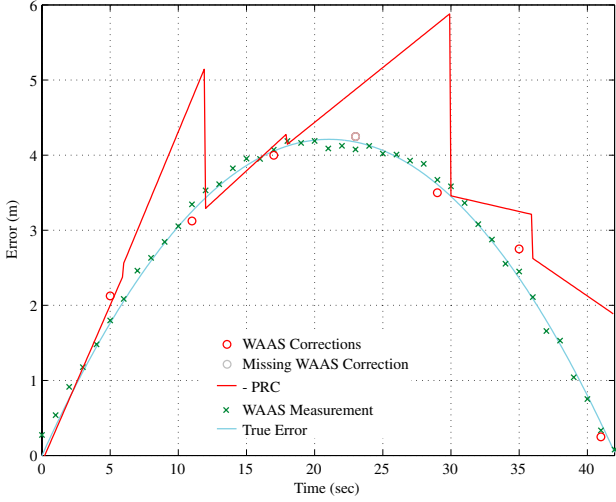


Figure 1. True clock error, the error as measured by the service provider, and the resulting corrections are shown for an extreme case with maximum acceleration.

where ε_{fc} is the degradation parameter for fast correction data (see A.4.5.1.1 of [1]). Thus even though fast corrections are coming at a slower rate, the Fast correction degradation is still reset every six seconds provided the user has not lost any messages and the IODFs are not equal to 3.

Fast Correction and UDRE Time-Outs

A user must verify that the newer fast clock error used in Equations (1) and (2) to generate the pseudorange correction be no older than the broadcast time-out period, I_{fc} , for that satellite. Thus, one must verify $t - t_n \leq I_{fc} + 1$, before applying the correction (2). If this time difference is greater than $I_{fc} + 1$, the user must not apply the correction and the satellite is flagged as not having a differential correction. Similarly, if the old fast clock error is too outdated, $t_n - t_o > I_{fc}$, then the user also cannot use (2). A user must have both a valid fast clock correction and range rate correction to differentially correct the satellite. In addition, they cannot project fast clock corrections beyond eight times the time difference used to form the range rate correction, $t - t_n > 8 \cdot (t_n - t_o)$.

The time-out for σ_{UDRE}^2 values is 12 seconds from the end of the message (or 13 seconds from the time of applicability of the σ_{UDRE}^2 information). Thus it also must be true that $t - t_{UDRE} \leq 13\text{sec}$ for a user to apply differential corrections to that satellite. When the user is missing a fast correction and the Type 6 IODF does not match, it is sufficient to use $t - t_6 \leq 13\text{sec}$ in place of the above equation. In all cases σ_{UDRE}^2 values set to “Do Not Use” or “Not Monitored” take precedence over any pseudorange correction calculations. Upon receipt of such information, a user must remove all other knowledge of the satellite and wait for the master station to declare it

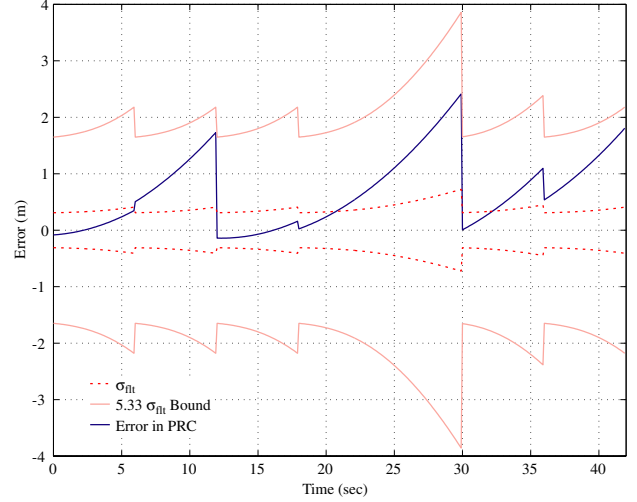


Figure 2. The error in the correction and the resulting $1\text{-}\sigma$ and 10^{-7} bounds are shown for the data in Figure 1.

safe and then fill in correction information for it once again.

A user also cannot form a range rate correction (1) across fast clock errors if the σ_{UDRE}^2 information has been set to “Do Not Use” or “Not Monitored” in between reception of those fast clock errors (Types 2 -5 and 24). After the reception of an integrity alarm with either of those two conditions, a user must receive two fast clock correction messages before applying corrections to that satellite. If there is a change of integrity status from safe to unsafe, that integrity message is broadcast four times in a row. A user who misses four messages in a row must assume that every σ_{UDRE}^2 has been set to Not Monitored.

The error terms ε_{rrc} and ε_{fc} are added to the broadcast UDRE differently depending on the value of the RSS_{UDRE} bit. If the bit is set to zero, the contributions are summed as sigmas and then squared to form a variance. If the RSS_{UDRE} bit is set to one, the contributions are squared and then summed as variances. There are other contributions from long term corrections and extra missing messages that will be discussed later. The WAAS system will initially use the RSS_{UDRE} set to 0 [8]. However, in this paper, we will set RSS_{UDRE} to 1 [7]. In this case, the variance of the corrections described so far is given by

$$\sigma_{fit}^2 = \sigma_{UDRE}^2 + \varepsilon_{fc}^2 + \varepsilon_{rrc}^2 \quad (10)$$

where σ_{fit}^2 is the bounding variance of all the error terms discussed. Equation (A-44) in [1] describes the equation and terms in full.

To see how these concepts connect together we will look at some examples.

Time (s)	Last Message Time(s)	Last Message PRC (m)	Last Message IODF	PRC (m)	Error (m)	σ_{fit}^2 (m)
0	-1	0.500	0	0.083	-0.083	0.309
3	-1	0.500	0	-1.167	0.052	0.338
6	5	-2.125	1	-2.563	0.505	0.309
9	5	-2.125	1	-3.875	1.045	0.338
12	11	-3.125	2	-3.292	-0.140	0.309
15	11	-3.125	2	-3.792	-0.071	0.338
18	17	-4.000	0	-4.146	0.238	0.309
21	17	-4.000	0	-4.583	0.373	0.338
24	17	-4.000	0	-5.021	0.893	0.412
27	17	-4.000	0	-5.458	1.584	0.544
30	29	-3.500	2	-3.458	0.008	0.311
33	29	-3.500	2	-3.333	0.479	0.354
36	35	-2.750	0	-2.625	0.537	0.309
39	35	-2.750	0	-2.250	1.100	0.338

Table 1. Values of Pseudorange Correction (PRC), true error and bounding variance, σ_{fit}^2 , are listed for various times along with the corresponding message times, broadcast corrections, and IODFs.

Example 1: No Type 6 messages are broadcast and a user is not aware of any missing messages from Types 2-5 or 24. This corresponds to the data in Figures 1 and 2 and Table 1 from times 0 to just before 30, and 36 to 42.

Example 2: Same as Example 1 except now the user has non-consecutive IODFs so they know they have missed a fast correction. This corresponds to the data in Figures 1 and 2 and Table 1 from times 30 to 36.

These examples are illustrated in Figures 1 and 2 and in Table 1. Figure 1 shows a simulation of true clock error. There is an underlying quadratic representing the

true SA over time (light solid line). This is corrupted by measurement noise and biases to form the instantaneous service provider estimates indicated by the x's. These measurements are used to forward predict an estimate of the correction four seconds ahead which is broadcast once every six seconds as indicated by the circles. Finally the messages are extrapolated as described by (1) and (2) to form a pseudorange correction. The pseudorange correction, or rather, its negative, is shown as the dark solid line. Also shown in Figures 1 and 2 is the effect of the user not receiving the correction broadcast at time 23. Notice that the RRC_n from time 30 to 36 is also less accurate due to the lost message.

Figure 2 and Table 1 show the error remaining after correction and the error bound as calculated according to the MOPS. Until time 30 the user calculates the bound as though they are missing no data. At time 30 they finish decoding a fast correction whose IODF is out of sequence (2 instead of 1). At that point they must add the additional degradation term (8) to account for the additional error in the RRC_n term. At time 36 they receive another new fast correction and the calculation of the error bound resumes to its nominal form as no data is missing.

These examples represent a worst possible case, with the true pseudorange acceleration matching the specified not to exceed value of 19 mm/s^2 [9]. The vast majority of the time the actual acceleration will be less than half this value [10]. The additional broadcast parameters necessary to calculate σ_{fit}^2 are listed in Table 2.

Parameter	Examples 1 & 2	Examples 3, 4, & 5
σ_{UDRE}^2	0.0924 m^2	$0.0520 \text{ m}^2 \dagger$ $0.0924 \text{ m}^2 \dagger$
a	4.60 mm/s^2	0.30 mm/s^2
I_{fc}	12 sec	66 sec
FC update rate	6 sec	30 sec
t_l	4 sec	4 sec
B_{rrc}	0.15 m	0.15 m

Table 2. The parameter values used to calculate the error bounds for examples 1-5.

\dagger The upper value was broadcast three times per 30 second fast correction update interval (with the fast corrections, 6 seconds after and 12 seconds after), and the lower value twice (18 seconds after the fast correction and 24 seconds after).

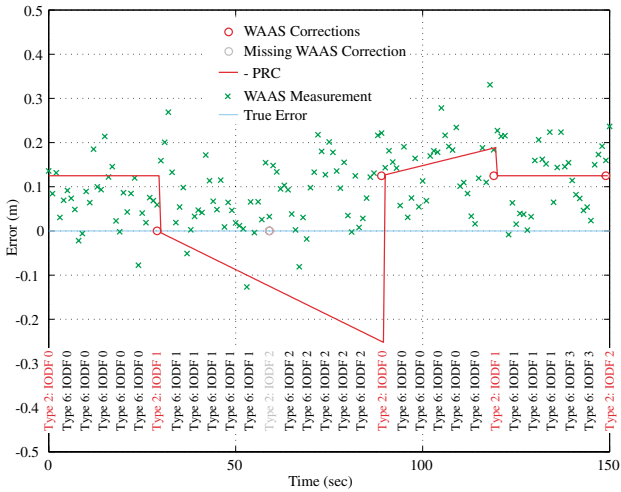


Figure 3. True clock error, the error as measured by the service provider, and the resulting corrections are shown for a case with no clock acceleration. The times of the Type 2 and Type 6 messages are shown and the IODFs indicated.

This simulation is for illustrative purposes only. Although the 10^{-7} confidence bound (see Appendix J of [1]) was always larger than the actual error, it did become larger than $4.4 \cdot \sigma_{flt}^2$, which should occur less than one second out of a day. In fact much of the time the error is outside of the one sigma bound. If this situation had really occurred, the service provider would have further increased σ_{UDRE}^2 , or the broadcast degradation parameter a . Merely bounding the errors in the pseudorange domain is not sufficient to protect the user in the position domain. Large errors such as those in Figures 1 and 2 must also have a sufficiently low probability of occurrence [6].

Fast Correction Degradation Equation Use with Message Type 6

At all times, each σ_{UDRE}^2 is updated at least every six seconds, but if Message Type 6 is implemented, the fast clock errors may be sent out less frequently. Message Type 6 performs a virtual update of the fast clock error if the IODF is not equal to 3. A user who receives this message is effectively being told that any changes to fast clock error since it was last broadcast are small, and are covered by the new σ_{UDRE}^2 included in that message.

Example 3: The fast corrections are broadcast every 30 seconds with four Type 6 messages in between. The user is not aware of any missing messages and there are no alarm conditions. This corresponds to the data in Figures 3 and 4 and Table 3 from times 0 to 90, and 120 to 138.

Example 4: The same as above except that the user is missing a fast correction, but still no alarms. This corresponds to the data in Figures 3 and 4 and Table 3 from times 90 to 120.

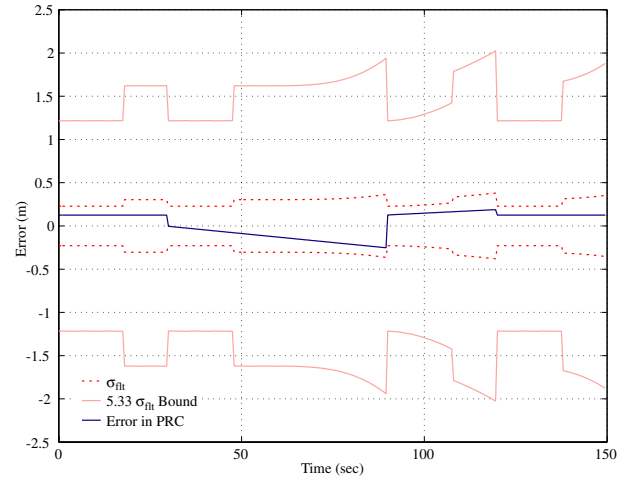


Figure 4. The error in the correction and the resulting $1\text{-}\sigma$ and 10^{-7} bounds are shown for the data in Figure 3.

Time (s)	Last Message Time (s)	Last Message Type	Last Message IODF	t_{UDRE} (s)	σ_{flt} (m)
0	-1	2	0	-1	0.228
6	5	6	0	5	0.228
12	11	6	0	11	0.228
18	17	6	0	17	0.304
24	23	6	0	23	0.304
30	29	2	1	29	0.228
36	35	6	1	35	0.228
42	41	6	1	41	0.228
48	-	-	-	41	0.229
54	53	6	1	53	0.304
60	-	-	-	53	0.304
66	65	6	2	53	0.307
72	71	6	2	53	0.314
78	77	6	2	53	0.329
84	83	6	2	53	0.355
90	89	2	0	89	0.228
96	95	6	0	95	0.234
102	101	6	0	101	0.248
108	107	6	0	107	0.335
114	113	6	0	113	0.357
120	119	2	1	119	0.228
126	125	6	1	125	0.228
132	131	6	1	131	0.228
138	137	6	3	119	0.314
144	143	6	3	119	0.329
150	149	2	2	149	0.228

Table 3. Tabular values are listed for the message types, times, IODFs, t_{UDRE} , and the confidence bound, σ_{flt} , corresponding to Figures 3 and 4. The - indicates a lost message.

Example 5: The same as Example 3 except that one or more IODFs are set to 3. This corresponds to the data in Figures 3 and 4 and Table 3 from times 138 to 150.

These examples are illustrated in Figures 3 and 4 and in Table 3. Figure 3 shows another simulation of true fast clock error. There are no acceleration or velocity terms on the true pseudorange; it is constant and zero (light solid line). This is corrupted by measurement noise and biases to form the instantaneous service provider estimates indicated by the x's. These measurements are used to forward predict an estimate of the correction four seconds ahead which is then broadcast once every thirty seconds as indicated by the circles. In between these messages, every six seconds, Message Type 6 is broadcast to update σ_{UDRE}^2 . The Type 6 messages in no way affect the extrapolation of the pseudorange correction. The pseudorange correction (again, actually the negative) is shown as the dark solid line. Also shown is the effect of the user not receiving the Message Type 6 at time 47, or the fast correction broadcast at time 59. Additionally, at times 138 and 144 the IODFs the Type 6 messages are set to 3. These examples represent an expected case with SA turned off. The parameters affecting the calculation of σ_{fit}^2 are shown in Table 2.

Figure 4 shows the error remaining after correction and the error bound as calculated according to the MOPS. Until time 90 the user calculates the bound as though they are missing no data. At time 66 they finish decoding a Message Type 6 whose IODF is out of sequence (2 instead of 1). The t_{UDRE} remains at 53 s and will remain there until the receipt of the next fast correction. Because the IODF was not set to 3 and the user has not missed 4 messages in a row, they know that the service provider is monitoring this and other combinations of old data and it is safe for them to continue its use. Thus, t_6 is updated and even though $t - t_{UDRE} > 13\text{sec}$, it is safe to continue operation. When the new fast correction is decoded at time 90, the RRC_n is formed using corrections with IODFs that are out of sequence, requiring the inclusion of the degradation parameter ϵ_{rrc} (8). At time 120 another new fast correction is received and the calculation of the error bound resumes its nominal form as no data is missing. At times 138 and 144, the user decodes Type 6 messages with the IODF set to 3. This sets the t_{UDRE} back to 119 s, the time of applicability of the last received fast correction. If the user had missed the fast correction at 119 s, then t_{UDRE} would have been set to 89 s.

The above discussions neglected the contributions of the long term error estimation. The σ_{UDRE}^2 broadcast to the user must also include these error projections. Since they are a function of user location, the value sent corresponds to the largest projection within the service

region. This term can overwhelm the noise in the clock estimation which has been greatly exaggerated in these examples.

Long Term Corrections

The long term corrections are far less complicated than the fast corrections. These corrections are contained in the back half of Message Type 24 or in Message Type 25 for GPS satellites. For Geostationary satellites they are contained in Message Type 9. These packets contain only correction information as the error bounds for these message types are contained in the σ_{UDRE}^2 . There are terms that degrade the error bound when missing one of these correction terms. These factors are relatively straight forward and described in Section A.4.5.1.3 in [1]. One potentially confusing item is the form of Equation (A-50). If the current time is between the time of applicability of the velocity code = 1 message, t_0 , and that time plus the $v=1$ update interval, $I_{lc,v1}$, then the first term, $C_{lc,lsb}$, should also be set to zero. Thus for the time span $0 \leq t - t_0 \leq I_{lc,v1}$ there is no applied degradation factor.

En Route Degradation

For precision approach a user is only allowed to miss one of any particular message. However, the user can still operate in less stringent phases of flight even if they have missed two of any particular fast or slow correction messages. When users apply corrections beyond the precision approach specified time-out, but before the En Route, Terminal, NPA time-out period (i. e. $I_{lc} + 1 < t - t_n \leq \frac{3}{2} I_{lc} + 1$), another degradation term is added to account for the potential error. While performance is probably worse than for a user not missing any messages, it will likely still provide sufficient navigation for these other phases of flight. This degradation term is described in Section A.4.5.1.4 of [1].

IONOSPHERIC CORRECTIONS

The ionospheric corrections and integrity bound information are broadcast in Message Types 18 and 26. Message Type 18 defines a “mask” of activated Ionospheric Grid Points (IGP). This mask allows the user to determine the latitude and longitude of the corrections and confidences in the Type 26 messages.

The application of ionospheric corrections requires the user to interpolate corrections for their measurements from a predefined grid of vertical delay values, τ_{vi} . The user must determine which grid points to use for interpolation and then apply the proper weights to each one to form their vertical delay estimate and confidence. This vertical delay estimate at the user's Ionospheric Pierce Point (IPP), τ_{vpp} , is then scaled by the obliquity factor to

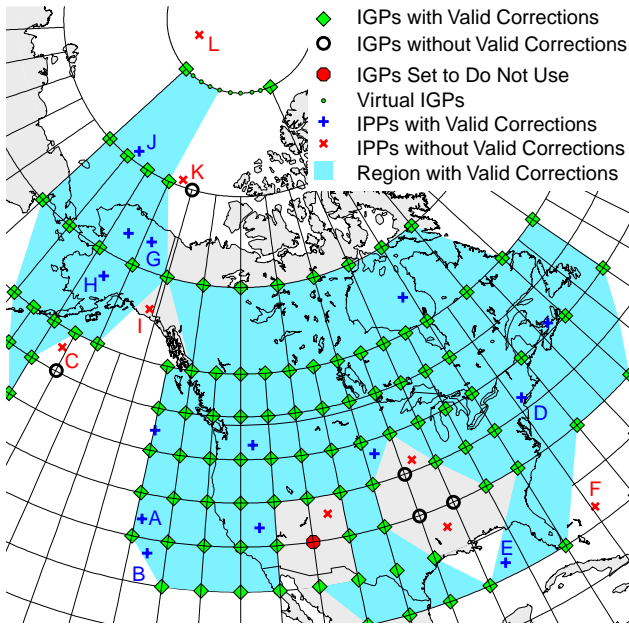


Figure 5. An example grid of Ionospheric Grid Points (IGPs) is shown with the corresponding region of valid correction. IGPs defined in the mask are indicated by diamonds if they have valid corrections by octagons if they have been set to “Do Not Use”, or by circles if they are designated “Not Monitored”. User Ionospheric Pierce Points (IPPs) are also shown along with the indication of whether they can be corrected or not. [11]

convert it to a slant range correction. The mathematical formulations for interpolated vertical IPP delay, τ_{vpp} , are functions of IPP latitude, ϕ_{pp} , longitude, λ_{pp} , and number of grid points used for the interpolation (see Section A.4.4.10.3 of [1]).

As depicted in Figure 5, the earth is divided into four interpolation regions:

- 1) $|\phi_{pp}| \leq 55^\circ$
- 2) $55^\circ < |\phi_{pp}| \leq 75^\circ$
- 3) $75^\circ < |\phi_{pp}| \leq 85^\circ$
- 4) $|\phi_{pp}| > 85^\circ$

The first two regions use rectangular grids with equal spacings in latitude and longitude. To save bandwidth, the service provider has the option of defining both 5° and 10° squares in Region 1. In Region 3 the user creates virtual $10^\circ \times 10^\circ$ cells for interpolation. Region 4 is physically a circular region and the interpolation has slightly different form. In all regions the user’s IPP must be surrounded by active grid points with valid data. The user first seeks to use the four active surrounding IGPs defined in the mask to create a square which can be used to interpolate to the location of the IPP. If there is no surrounding square the user then checks for a surrounding triangular region. If this too is unavailable, the user cannot form a differential ionospheric correction. The selection criteria for choosing surrounding grid points is given in Section A.4.4.10.2 of [1].

IPP	Region	# of IGPs in Mask	# of Valid IGPs	IPP Within Triangle	Valid Correction
A	1	4	4	-	yes
B	1	3	3	yes	yes
C	1	4	3	no	no
D	1	4	4	-	yes
E	1	4	3	yes	yes
F	1	3	3	no	no
G	2	4	3	yes	yes
H	2	4	4	-	yes
I	2	3	3	no	no
J	3	4	4	-	yes
K	3	4	3	no	no
L	4	2	2	-	no

Table 4. For the indicated IPPs in Figure 5, the following are listed: region, number of surrounding IGPs defined in the mask, subset of IGPs having valid corrections, determination of IPP containment within a valid triangle, and possibility of differential correction.

The user always begins by checking the mask. If the user’s IPP is in Region 1 and the four grid points corresponding to the $5^\circ \times 5^\circ$ surrounding grid cell are activated in the mask, the user then interpolates using at least 3 of those 4 IGPs. If the correction information for those IGPs does not support a valid correction the user may not attempt to use the IGPs corresponding to a surrounding $10^\circ \times 10^\circ$ cell. If no $5^\circ \times 5^\circ$ square or surrounding triangle is defined in the mask, the user may then check for a surrounding $10^\circ \times 10^\circ$ cell. If the four possible larger cells are not defined or do not have valid corrections creating a triangular region containing the IPP, then no ionospheric correction can be formed for that IPP.

The rules for selecting IGPs in Region 2 are identical to Region 1 except only $10^\circ \times 10^\circ$ cells are possible. If the four surrounding IGPs are defined and have valid corrections then all are used to interpolate the ionospheric corrections. If only three are defined and/or valid and they form a triangle containing the IPP then the user may also form a correction unless the fourth point has been set to “Do Not Use”. In this case, no interpolation is allowed in any of the four squares surrounding that IGP. If the user is outside of the triangle or there are fewer than three defined and valid IGPs then no correction can be formed.

Region 3 has 10° longitudinal spacing at the 75° latitude line but 90° spacing at the 85° line. The user must linearly interpolate “virtual” grid points along this line to form $10^\circ \times 10^\circ$ cells. Once the virtual $10^\circ \times 10^\circ$ cell is formed the same rules apply to this region as described

for Region 2. If both “true” IGPs are not defined at 85° latitude, no virtual IGPs may be created.

In Region 4 there are only four possible IGPs so selection is straight forward. Instead of being in square or triangular regions the user must either be in the full circle or the semi-circle described by three points.

To provide examples of how these rules work, Figure 5 shows a particular mask and valid IGP configuration. There is a shaded region corresponding to all the areas where a differential correction is possible according to the rules of selection. Also shown are some sample IPPs and whether they can be differentially corrected (see Table 4). Most of the examples are relatively obvious, for example, IPPs A and H are inside well defined square cells, while IPPs B and G are inside well defined triangular cells. IPPs D, E, and F are in region 1 but have neither square nor triangular 5°x5° cells defined in the mask around them. Therefore, one looks to see if the IPPs are surrounded by 10°x10° cells. IPPs D and E are surrounded by a square and a triangular cell respectively, but IPP F is in neither.

The IPP marked C is interesting because it is outside of the 5° triangle of valid grid points, but if the non valid IGP below and right of it had not been defined in the mask, this IPP would have been contained in a 10°x10° right triangle and could have been corrected. However, since that IGP is in the mask and has either not been received, timed-out, set to “Do Not Use” or set to “Not Monitored”, the IPP cannot be differentially corrected as stands. IPP I is in Region 2 and clearly falls out of the triangle of valid correction.

IPP J and K fall in Region 3 first requiring interpolation along 85° latitude to form virtual IGPs. These points can be used to form a 10°x10° box around J so that it can be corrected. However, IPP K falls outside of the triangular region created by the two virtual IGPs and the lower true IGP so it cannot be corrected. Finally IPP L lies in Region 4, but only two IGPs are defined in this example so no differential correction is possible. This figure is for illustrative purposes only and is not meant to represent a likely scenario.

Finally, we turn to the calculation of the weights used to form the correction at the IPP from the IGPs. These weights are described in Section A.4.4.10.3 of [1].

Example 6. Suppose a user has a pierce point at (36° N, 122°W) in Region 1. The 5° bounding rectangle for this IPP is 35 - 40°N and 120 - 125°W.

If these IGPs are specified and valid, the user interpolates using all four points. The weights the user applies would be given by (A-24) - (A-32)

$$\begin{aligned} W_{(40^\circ \text{ N}, 125^\circ \text{ W})} &= 0.08, & W_{(40^\circ \text{ N}, 120^\circ \text{ W})} &= 0.12, \\ W_{(35^\circ \text{ N}, 125^\circ \text{ W})} &= 0.32, & W_{(35^\circ \text{ N}, 120^\circ \text{ W})} &= 0.48 \end{aligned} \quad (11)$$

If the IGP at (40° N, 125° W) is not monitored, a user could still perform triangular interpolation, because the IPP is in the region bounded by the other three grid points. For this case the weights become (A-34) - (A-36)

$$\begin{aligned} W_{(40^\circ \text{ N}, 125^\circ \text{ W})} &= 0.00, & W_{(40^\circ \text{ N}, 120^\circ \text{ W})} &= 0.20, \\ W_{(35^\circ \text{ N}, 125^\circ \text{ W})} &= 0.40, & W_{(35^\circ \text{ N}, 120^\circ \text{ W})} &= 0.40 \end{aligned} \quad (12)$$

Example 7. Suppose a user has a pierce point at (81° N, 104°W) in Region 3. The bounding rectangle for this IPP is 75 - 85°N and 100 - 110°W in Region 3.

If the IGPs at (85° N, 180° W) and (85° N, 90° W) are specified and valid, the user forms virtual grid points at (85° N, 110° W) and (85° N, 100° W). The values of the virtual grid points would then be

$$\tau_{v,(85^\circ \text{ N}, 110^\circ \text{ W})} = \frac{2}{9} \tau_{v,(85^\circ \text{ N}, 180^\circ \text{ W})} + \frac{7}{9} \tau_{v,(85^\circ \text{ N}, 90^\circ \text{ W})} \quad (13)$$

$$\tau_{v,(85^\circ \text{ N}, 100^\circ \text{ W})} = \frac{1}{9} \tau_{v,(85^\circ \text{ N}, 180^\circ \text{ W})} + \frac{8}{9} \tau_{v,(85^\circ \text{ N}, 90^\circ \text{ W})}$$

The weights the user forms would be given by

$$\begin{aligned} W_{(85^\circ \text{ N}, 110^\circ \text{ W})} &= 0.24, & W_{(85^\circ \text{ N}, 100^\circ \text{ W})} &= 0.36, \\ W_{(75^\circ \text{ N}, 110^\circ \text{ W})} &= 0.16, & W_{(75^\circ \text{ N}, 100^\circ \text{ W})} &= 0.24 \end{aligned} \quad (14)$$

If the IGP at (75° N, 110° W) is not monitored, a user could still perform triangular interpolation, because the IPP is in the region bounded by the other three grid points. For this case the weights become

$$\begin{aligned} W_{(85^\circ \text{ N}, 110^\circ \text{ W})} &= 0.40, & W_{(85^\circ \text{ N}, 100^\circ \text{ W})} &= 0.20, \\ W_{(75^\circ \text{ N}, 110^\circ \text{ W})} &= 0.00, & W_{(75^\circ \text{ N}, 100^\circ \text{ W})} &= 0.40 \end{aligned} \quad (15)$$

Example 8. For a user with an IPP in Region 4 at (87° N, 104°W) the weights would be given by

$$\begin{aligned} W_{(85^\circ \text{ N}, 90^\circ \text{ E})} &= 0.059, & W_{(85^\circ \text{ N}, 0^\circ \text{ W})} &= 0.141, \\ W_{(85^\circ \text{ N}, 180^\circ \text{ W})} &= 0.235, & W_{(85^\circ \text{ N}, 90^\circ \text{ W})} &= 0.565 \end{aligned} \quad (16)$$

and for an IPP at (86° N, 72°W) the weights would be given by

$$\begin{aligned}
 W_{(85^\circ \text{ S}, 130^\circ \text{ E})} &= 0.030, & W_{(85^\circ \text{ S}, 40^\circ \text{ E})} &= 0.070, \\
 W_{(85^\circ \text{ S}, 140^\circ \text{ W})} &= 0.266, & W_{(85^\circ \text{ S}, 50^\circ \text{ W})} &= 0.634
 \end{aligned}
 \tag{17}$$

CONCLUSIONS

The WAAS MOPS achieves remarkable performance despite the severe constraints placed upon it. Although the bandwidth is extremely limited, the quantization error and restricted update rates do not currently limit system performance. The MOPS achieves its goals of creating and documenting an interface between the service provider and the user that is capable of satisfying the aviation requirements for accuracy, integrity, continuity and availability.

This paper represents the views of the author and does not necessarily convey the opinion of RTCA, the FAA or Stanford University. Although the author believes all of this information to be accurate, mistakes and misunderstandings may have occurred in this interpretation of the MOPS. Please check our web site at <http://waas.stanford.edu/mops.html> for updated versions of this manuscript. The author will correct any errors found with newer versions at that site. Note that the MOPS is still being revised. Any changes or additions will also be updated at this location. Comments questions and concerns can be registered at the same address.

This paper provides examples of how the different message types are connected together and how the integrity bounds are calculated. These examples are meant to aid in understanding the intent of the MOPS. It is important that these concepts be well understood to be implemented correctly. It is hoped that having examples to compliment the rules listed in the MOPS will be helpful to new readers. Correct implementation leads directly to flight safety.

ACKNOWLEDGMENTS

The author wishes to thank all the people who participated in the development of the MOPS. In particular JP Fernow and Bruce Declene provided many helpful insights. Sincere thanks also go to Andrew Hansen for his many contributions to this document and for creating the web site. Thanks also go to Per Enge for many helpful discussions. This work was sponsored by the FAA GPS Product Team (AND-730).

REFERENCES

- [1] RTCA Special Committee 159 Working Group 2, "Minimum Operational Performance Standards for Global Positioning System / Wide Area Augmentation System Airborne Equipment," RTCA Document Number DO-229A June, 1998.
- [2] Enge, P., "WAAS Messaging System: Data Rate, Capacity, and Forward Error Correction," NAVIGATION, Spring, 1997.
- [3] Comp, C., Walter, T., Fuller, R., Barrows, A. K., Alter, K., Gebre, D., Hayward, R., Jennings, C., Hansen, A., Phelts, E., Archdeacon, A., Enge, E., Powell, J. D., and Parkinson, B., "Demonstration of WAAS Aircraft Approach and Landing in Alaska," in proceedings of ION GPS-98, Nashville, TN, September 1998.
- [4] Tsai, Y. J., Enge, P., Chao, Y. C., Walter, T., Kee, C., Evans, J., Barrows, A., Powell, D. and Parkinson, P., "Validation of the RTCA Message Format for WAAS," in Proceedings of ION GPS-95, Palm Springs, CA., September, 1995.
- [5] Loh, R., Wullschleger, V., Elrod, B., Lage, M., and Haas, F., "The U.S. Wide-Area Augmentation System (WAAS)," NAVIGATION, Fall, 1995.
- [6] Walter, T., Enge, P., and Hansen, A., "A Proposed Integrity Equation for WAAS MOPS," in proceedings of ION GPS-97, pp. 475-484, Kansas City, MO, September 1997.
- [7] Comp, C., Gazit, R., Walter, T., and Enge, P., "Improving WAAS Integrity and Availability: UDRE and GIVE Time Updates," in proceedings of ION GPS-97, pp. 1315-1324, Kansas City, MO, September 1997.
- [8] Slattery, R., Peck, S., Anagnost, J., and Moon, M., "Guaranteeing Integrity for all Users of Active Data," to be published in Global Positioning System: Papers Published in Navigation Volume VI.
- [9] Global Positioning System Standard Positioning Service Signal Specification, June, 1995.
- [10] Chao, Y. C. and Parkinson, B. W., "The Statistics of Selective Availability and its Effect on Differential GPS," Proceedings of ION GPS-93, Salt Lake City, Utah, September, 1993.
- [11] Wessel, P. and Smith, W. H. F., "New Version of the Generic Mapping Tools Released," EOS Trans. Amer. Geophys. U., vol. 76, pp 329, 1995.