Message Design for a Robust Time Signal using Distance Measuring Equipment (DME) Pulse Pair Position Modulated (PPPM) Pseudo lite

Sherman Lo[*], Yu-Hsuan Chen[*]

[*]Stanford University
Stanford, USA
email: sherman.lo@stanford.edu, shinge@stanford.edu

1. Introduction

The Federal Aviation Administration (FAA) Alternative Position Navigation and Timing (APNT) program developed several possibilities for improving distance measuring equipment (DME) such as DME pseudolite (PL) systems to provide data and passive ranging in addition to traditional DME operations. These enhanced DME (eDME) were designed to serve as an operational back up to GPS/GNSS, particularly to support future airspace and Next Generation Air Transportation System (NextGen) operations. While eDME concepts are not part of the current upgrades to the US DME system, there is continued interested in the ideas as they can support other airspace needs and help meet federal directives to increase the resiliency of Positioning, Navigation, and Timing (PNT) services [1][2].

Timing is a critical service used by many parts of the FAA infrastructure. Robust time distribution for FAA assets and other US federal infrastructure using high power terrestrial signals under federal control is attractive. Hence, the eDME efforts are still relevant and it is being examined by the Department of Transportation to provide alternative timing. DME PL based on pulse pair position modulation (PPPM) eDME design that is compatible with existing equipment and can be implemented with an applique [3][4]. This concept was tested on air in 2015 [5]. The paper examines the design of a DME PPPM transmission for providing robust and trusted precise time. First, it analyzes the required and desired data to provide precise time and frequency services to the national airspace (NAS). The paper then analyzes on-air test results to determine the forward error correction (FEC) needed to support the design and provide a high level of availability.

This paper examines improvements and updates to the DME PPPM message design for the on-air testing of a robust timing service. Specifically, it analyzes the data needed for absolute time and robust authentication and examines the design of FEC based on prior testing. It details the required and desired data for eDME to support robust and authenticatable time.

2. Background

DME PPPM is designed a means of providing a pseudoranging, timing and data capability on existing DME transponders without modifying existing field equipment. It utilizes the inherent operations of the DME transponder (i.e. ground station). The transponder receives an interrogation pulse pair on its interrogation frequency and transmits a pulse pair response to that interrogation on a separate frequency (its reply frequency) after a fixed delay known as the reply delay. By knowing the interrogation and reception time of these pulse pairs, one can calculate the round-trip time and hence true slant range from the aircraft to the interrogator.

The DME PPPM generator transmits a pseudo random sequence of DME interrogations (i.e. pulse pairs) that produce that exact sequence with DME replies. This acts as a synchronization, timing and pseudo ranging signal. Parts of the sequence can also be used to provide data. In this way, the DME PPPM generator works by acting as a DME interrogator and the DME PPPM functionality can be added to an existing DME transponder as an applique. This is shown in Figure 1.

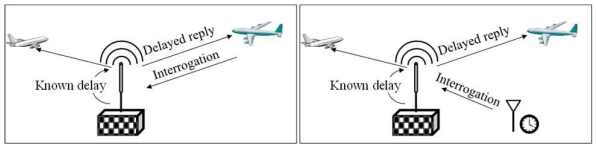

Figure 1. DME (Left) and DME PPPM Operations (Right)

The PPPM design provides both a timing (synchronization or synch) and a data signal [5]. The synchronization signal is a known pseudo random sequence of pulse pairs and is useful for providing a regular heartbeat transmission to range or get a stable time/frequency counter. It also establishes the reference time or frame for data transmissions which utilize DME pulse pairs at established time offsets to the start of synchronization sequence hence pulse pair position modulation. This synchronization and data design was made to be robust to loss of PPPM pulse pairs as PPPM signals have no priority in the DME transponder. "Reception" loss of PPPM pulse pairs occur mostly due to these pulse pairs not being transmitted as the DME transponder was responding to the interrogation signals from another aircraft or had priority pulse pairs to send out. This means that more aircraft in the airspace using the DME transponder will cause a higher level of interference to DME and DME PPPM use as there will more transmitted interrogation pulse pairs.

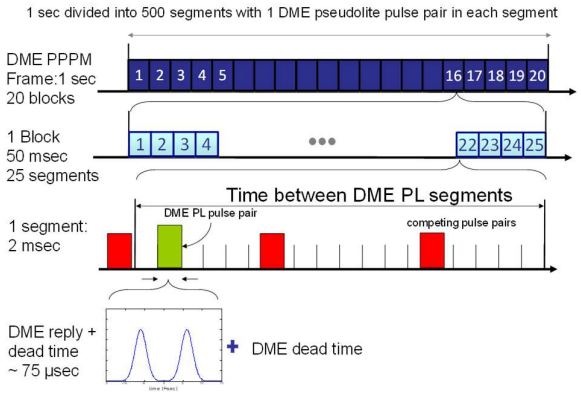

Figure 2. Basic Structure of Preliminary DME PPPM Design

To understand this a little better, we describe the preliminary DME PPPM transmission design used in our initial testing and use it to illustrate these issues. This design uses 500 DME reply pulse pairs per second which represents a little less than 20% of the capacity of many common DMEs. Figure 2 shows the basic structure of the DME PPPM design. Each frame is 1 second long and divided into 20 blocks of 50 milliseconds (ms). Within each of the 20 block, there will be 25 equally long segments. Only one DME PPPM pulse pair is used in each of the 2 ms long segment. This pulse pair will be transmitted at one of a fixed number of possible time slots within that segment. In the preliminary design, the segment is divided into 64 (i.e. 2^6) equally spaced allowable start times relative to the start of the segment. For blocks dedicated to ranging/synchronization, the start times within each segment is known. For data blocks, the start time of the DME PPPM pulse pair indicates that data symbol transmitted. For

example, the pulse pairs starting at start times 1, 2, 3 may represent a 0, 1, 2, respectively. So the 64 or 2^6 start times per segment allows transmission of 6 bits per segment. In the test design, we divide the frame into six sync and 14 data blocks as shown in Figure 2.

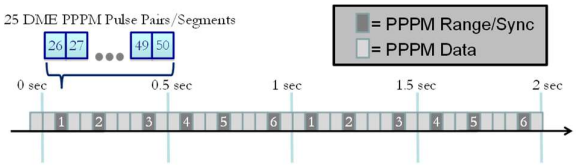

Figure 3. Sync and Data DME PPPM Blocks Used in March 2015 Tests

To create a robust signal, the system needs to be tolerant of losses and interference from aircraft replies, even at peak air traffic conditions and other normal DME functions. The synchronization sequence is found by correlation and hence losses just reduce the correlation peak. This is a nearly identical process to acquisition of the standard DME ranging and is quite loss tolerant. For the data sequence, several types of problems can occur when there are loses and/or replies to other aircraft. These are illustrated in Figure **2**. The first issue is data erasure and this can happen in three general ways. One way is if the signal is not transmitted – i.e. the DME PPPM pulse pair is preempted by a reply to an aircraft or another DME transmission. Another way is if there is a reply to an aircraft that is transmitted at one of allowable DME PPPM data start times within the same segment. As all DME transponder transmissions including the DME PPPM are the same (i.e. DME reply), receiver thus finds two possible DME PPPM pulse pairs in the segment where there should only be one. Hence, the user cannot determine which is the correct DME PPPM symbol and must declare an erasure. The final way an erasure can occur is an extension of the previous example. In this case, we have multiply replies to aircraft that starts at valid DME PPPM data start times within the segment. Hence, the receiver finds multiple possible DME PPPM pulse pairs in the segment. The other issue is data error where the receiver determines the wrong symbol. This can occur only if the PPPM signals preempted or interfered with such that it is not received and a non PPPM DME signal is received in such a way the receiver thinks it is a DME PPPM signal. That is, it is a DME reply transmitted at a DME PPPM start time, even though it is not actually DME PPPM signal (initiated by the DME PPPM applique).

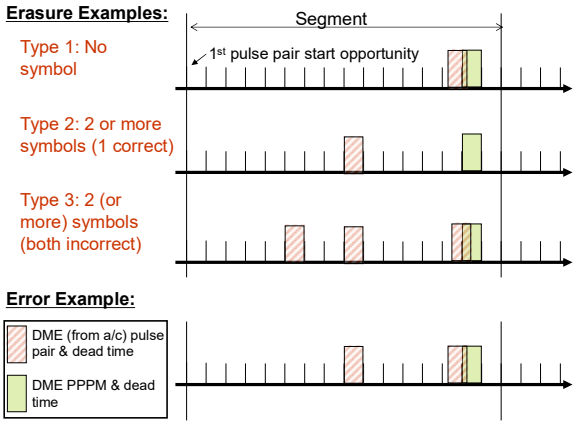

Figure 4. Types of Error and Erasures in DME PPPM Data (when there is overlap, only the earlier pulse pair in the overlap is transmitted)

As the DME PPPM pulse pair is a DME reply and the only way to identify it is if it starts at a valid DME PPPM start time. As there may be some clock drift in the user receiver clock from

when it synchronized time, the previous statement is more precisely stated that DME PPPM pulse pair is identified if the pulse pair starts at a valid DME PPPM start time within some margin of error. We term this margin of error as the acceptance tolerance (for accepting a pulse pair as PPPM) and is the maximum allowable time deviation from nominal start times.

DME PPPM for Robust time

Data is required for absolute time and authentication. This section will discuss both topics. While real time data is not required for DME PL ranging, it is needed to provide absolute time. For ranging, we solely need to know the location of DME transponder and its relative time of transmission. DME transponders are fixed and identified by their interrogation and reply frequency and pulse pair characteristics which define the DME channel it is operating on. While these characteristics are not unique, they are unique in a given region. So as long as we have a rough idea of where we are, we can know the DME station and use a database to look up its exact location. Additionally, since DME PPPM provides sync pulse pair sequences that identify each second or better and since all DME signals should take well less than a second to propagate to the airborne user, we can calculate the pseudorange from each station without ambiguity. For precise absolute time, we need to know the absolute date which can be given like in GNSS by providing: 1) a week number since a specified start date and 2) the number of seconds since the start of the week. These values change continuously and needs to be transmitted since we cannot guarantee that the user clock is synchronized to universal time coordinated (UTC) at a second or better. Hence data is required for precise absolute time.

The operations to provide precise time is different depending on whether the user is static with known position or dynamic. This is because of the need to account for the propagation delay to get precise time synchronization. A static user with known position or distance to the DME transponder will only need the pseudo ranging signal. From that it can calculate the current DME transponder time as the time of transmission indicated by the signal plus the propagation delay. It can then determine its time offset from DME, presumably UTC, time. Even if the range is not known, it only needs to be calculated once for a static user to make this correction for propagation delay. For a dynamic user, the delay changes with movement and hence a true range will be regularly needed. Hence a dynamic user will need both the traditional DME true ranges as well as the pseudo ranging signal to get absolute time and time synchronization. We previously termed this use of DME as hybrid ranging [6][7].

Hence the data needed to provide absolute time are: 1) absolute time in seconds and 2) station location. The former can be provided like in GNSS as two parts: 1) seconds from the start of the week (12:00 AM Sunday) and 2) week number from a specified start date (e.g. midnight January 6 1980 for GPS). Seconds from the start of the week requires 20 bits of data. Week number depends on how often we want to have a week number rollover. GPS uses 10 bits for rollover about every 20 years though other systems use 11 bits (~39 years). 10 bits is likely the minimum we should use but 11 bits should be sufficient given the expected life of the system. Of course, more can be used. The station location can be provided in latitude, longitude and altitude. Latitude and longitude can be provided at a submeter level with 25 and 26 bits respectively. Altitude at meter level depends on the maximum and minimum expected station height but 12-13 bits is likely sufficient expecting that there will be no stations below -96 meters and above 4000 or 8096 meters. Finally, we may want a small flag (~ 2 bit) indicating if the service status. The total number of bits needed if all this data is included in one message is 96-97 bits. This data may not need to be sent every second, especially the

station location which should not change. So 96-97 bits per second (bps) represents the highest level of data that we need to support for the timing service.

The data necessary to support authentication depends on the scheme utilized. Both a public key and a symmetric key/Timed Efficient Stream Loss-tolerant Authentication (TESLA) based system can be used such as those proposed for the Wide Area Augmentation System (WAAS) [8][9]. A public key system is simpler but requires more data for the same rate of authentication. TESLA is more data efficient but is more complex and has vulnerability to time errors. Also, a TESLA concept also requires public key-based authentication but at a much lower rate. Preliminary thinking leans us towards TESLA for its data efficiency and its resistance to quantum computing attacks. Common and data efficient public key systems, such as elliptic curve (EC)-Schnorr or Digital Signatures Algorithm using EC (EC-DSA) which require roughly 400 bits for signature for satellite navigation applications [8], are vulnerable to quantum computing attacks [10]. While a TESLA based system still uses public keys, it is used much less frequency and the DME channel, as seen later, should have the bandwidth to support future quantum resistant public key concepts which require orders of magnitude more data for key distribution. Hence, for TESLA system, we need only 15 bits for a message authentication message, about 115 bits for the symmetric key and then very occasionally 400 bits public key-based authentication to update the base TESLA key.

As an aside, we are investigating quantum computing resistant (known as post-quantum) techniques for data authentication [10]. One that is currently being looked at for this purpose is Great Multivariate Short Signature (GeMSS). In GeMSS, each signature requires about 300 bits which is quite reasonable. However, the public key is rather large compared to current quantum computing vulnerable public keys. The keys are approximately 2.4 to 24 megabits (Mb). This is better than other quantum schemes that require much larger signatures. Since public key distribution is exceedingly rare in TESLA, we can do this either offline or over a longer time period. As we shall see later, after applying forward error correction (FEC), will have an available data rate of 732 bps. The usage given the above analysis is shown in Figure **5** with 87% unused which is 635 bps available. With 635 bps available, 2.4 Mb can be sent in approximately 1 hours. We will later show how this is achieved. Other post quantum methods should also be suitable [11].

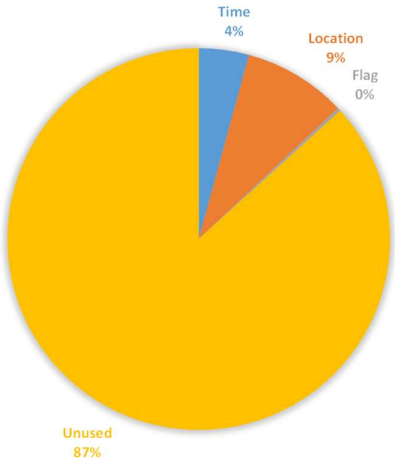

Figure 5. Data Usage on Designed DME PPPM Message

A useful examination for a robust PNT system is to examine its vulnerabilities under a reasonable threat model. It is the opinion of the author that no system is invulnerable and one can always think of ways to overcome it. However, these means may be too costly to implement (for example, building a new constellation of satellites to provide a spoof GNSS system). Hence, we should restrict the examination to attacks that are practical now and in the operational future of the system. When examining possible time spoofing of DME, this paper will limit itself to the attack model where the spoofer cannot deny the true DME signal. While it is possible to deny the true DME signal, directly exploiting this vulnerability will generally require significant power and some know how that makes detecting this attack easier. This is because typical en route DME transmit at 1 kilowatt (KW) peak power.

Examining vulnerabilities, TESLA prevents the data from being spoofed unless the user receiver is significantly time delayed (i.e. it is slow) from the true time (i.e. the true time is 12:01:00 and the user thinks it is 12:00:55). The true DME, which we assumed is not denied, provides absolute time with our proposed DME PPPM. The only way the spoofer can attack this is by providing a reply signal slightly earlier than the true reply to beat it out, which effectively reduces the calculated propagation delay and slows the receiver. This is seen in Figure **6** where the lines coming from the DME transponder are its true transmissions and the transmissions from the man in the middle (MITM) are spoofed ones. R' is spoofed transmission of the DME reply R. They should look the same for the user to accept but the spoofer can transmit it before or after the true reply once it receives the user interrogation I as that is the only knowledge needed to generate the reply pulse pair sequence. Hence the delay on R1 "delay" is d_{s1} where it can be any positive value but also some negative value. The corner case is immediately after the transmission of the interrogation – this is immediate reply. Hence, $d_{s1} > -(t_4-t_1+d_{reply})$ though the user knows that a reply cannot arrive until the reply delay time after the transmission of the interrogations. Hence, the limit is likely $d_{s1} > -(t_4-t_1)$ which is twice the propagation time. If the attacker only fakes the DME reply, as seen later, we derive that the attacker will alter the user clock by $d_{s1}/2$. Thus the attacker can at most slow the user clock by the propagation time. Given the useful range of DME less than 300 kilometers, this amounts to less than 1 ms of time shift.

The attacker may target the DME PPPM broadcast as also shown in Figure **6**. The attacker could replay the true broadcast DME PPPM message, M, which contains absolute time information. It would send message M', the spoofed version of M, which would be the same as M but delayed. The message must be the same since M is authenticated. We denote the replay delayed time as d_{s2}, where $d_{s2} > 0$ since it cannot be known a priori, it cannot be transmitted earlier. If the DME transponder signal cannot be easily denied, then this vulnerability is hard to exploit.

We can calculate the time offset between DME transponder (source) and the aircraft, θ. Equation (1) shows the calculated offset given nominal (unspoofed) interaction. With this offset, the time synchronized to the DME transponder is the time at the aircraft (Tc) plus the offset. With the added spoofing delays, we can calculate the spoof affected offset, θ_s, and compare it to the nominal offset and this is in (2). One can see that given the allowable range for d_{s2}, it will only cause a delay (slow the user clock) whereas d_{s1} can cause either a time delay or advancement. This portion is given as an example and further work is being done to examine this vulnerability.

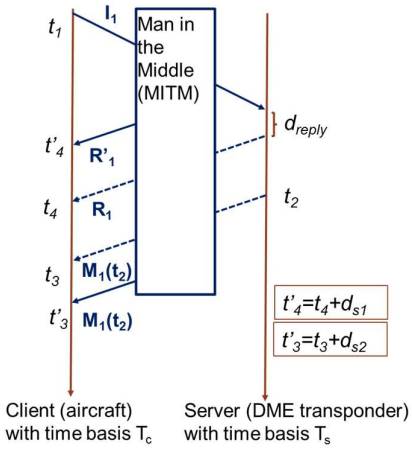

Figure 6. Time Transfer with DME and DME Pseudolite; Time is given in the time basis of the receiving site

$$\boldsymbol{\theta = (t_2 - t_3) + \frac{(t_4 - t_1) - d_{reply}}{2}} \qquad \textbf{(1)}$$

$$\theta_s = (t_2 - t'_3) + \frac{(t'_4 - t_1) - d_{reply}}{2} = \theta - ds2 + \frac{ds1}{2} \qquad \textbf{(2)}$$

3. On Air DME PPPM Test Results

DME PPPM was tested in on-air tests in March 2015 at Ohio University. A DME PPPM generator was coupled into a modern, commercial DME/TACAN (tactical air navigation), the Moog MM-7000. Test of a different 250 pulse pair per second (ppps) priority enhanced DME transmission was simultaneously conducted [12]. Additionally, as the DME/TACAN was not operational and so no other aircraft would be using it, artificial traffic was added to simulate a maximally loaded DME. Details on the test set up is given in [3]. DME PPPM ground and flight tests results from this test is given in Table 1 and Table 2 which shows the resulting DME PPPM pulse pair data erasure and error rates. The tables show results with different acceptance tolerance, the maximum allowable time deviation from nominal PPPM data start times to be still accepted as a PPPM data pulse pair. Data performance during flight data is worse as there is time additional uncertainty which creates more chances of misidentification – both missed and false detection. Regardless, an acceptance tolerance of 600 nanoseconds (ns) performs better than a larger acceptance tolerance of 1000 ns and those are the values will be used for the analysis in this paper. Reference [3] also provides results from analytic models. These results are slightly lower than the on-air flight results as they do not capture some sources of error. Hence, we will use the on-air results as the basis for a conservative determination of FEC. FEC allows for detection, and more importantly efficient correction of missing and erroneous data bits or symbols. This design will be used to support future on air testing in 2020 where Stanford University and the FAA examines the use of DME PPPM for robust time. Note, sync pulse pair erasure rates were similar though slightly better than the data erasure rates.

Table 1. Erasure Rate Statistics (600 & 1000 ns)

Test	Mean (%)	Standard Deviation (%)

0309, Ground 600 ns	29.3%	3.4%
0310 AM, Air 600 ns	36.4%	11.7%
0310 PM, Air 600 ns	39.0%	14.0%
0309, Ground 1000 ns	34.1%	3.8%
0310 AM, Air 1000 ns	39.5%	8.7%
0310 PM, Air 1000 ns	41.4%	10.5%

Table 2. Error Rate Statistics (600 and 1000 ns)

Test	Mean (%)	Standard Deviation
0309, Ground, 600 ns	3.5%	1.0%
0310 AM, Air, 600 ns	4.4%	1.9%
0310 PM, Air, 600 ns	4.6%	1.9%
0309, Ground, 1000 ns	5.2%	1.2%
0310 AM, Air, 1000 ns	6.6%	2.6%
0310 PM, Air, 1000 ns	6.8%	2.5%

Message Design

Theoretical and on-air test results on DME PPPM pulse pair availability are used to develop the level of FEC necessary in the signal design. We begin determining the level of FEC needed to provide 98% of better message availability given the pulse pair availability. For FEC, we use Reed Solomon (RS) for 6 bit symbols. Reed Solomon is a class of linear block codes that allow for efficient corrections of symbol errors where a symbol may be multiple bits [13][14]. Better symbol error correction performance, as opposed to typical bit (meaning 0 or 1) errors, is important as our information is packaged into 6 bit symbols which can be represented as one block. This is desirable as it protects against 6 bit burst errors or effective the amount of data loss when one pulse pair is lost. RS codes is usually described by two parameters (n, k) where n stands for the RS block length (in symbols) and k is the message size (i.e. the number of symbols of data actually available). Note, this is different the block of time used for PPPM and when discussing RS FEC, block will refer to the RS block. This results in a minimum Hamming distance of (n-k+1) symbols. The block length is 2^m-1 symbols where m is the number of bits per symbol. For our design, m equals 6. When the RS block has less than 2^m-1 symbols, zero symbols are used to pad the block until has enough symbols. This forms a truncated RS code where RS (n, k) code is used by described as RS (n-p, k-d) since there are p symbols that form the zero pad.

When we apply RS to DME PPPM, this breaks down the message into sets of 2^6-1 or 63 symbols or less. So we divide the data in 1 second into 5 groups of 63 symbols plus one group of 35 symbols and apply RS to each. This is an important point for the analysis as we do not care so much about the average and standard deviation of individual pulse pair (i.e. symbol)

error but rather those statistics on a block level. Again let *n* be the number of symbols per block. Define mean and standard deviation of symbol error and erasure rates as (μ_{err}, σ_{err}) and ($\mu_{erasure}$, $\sigma_{erasure}$), respectively. Then the mean and standard deviation of the number of pulse pairs with errors ($\mu_{n,err,block}$, $\sigma_{n.err,\ block}$) or erasure ($\mu_{n,erasure,block}$, $\sigma_{n,erasure,\ block}$) per block is given by Equation (3) and (4), assuming independence between each pulse pair/symbol. This is a reasonable assumption given DME operations.

$$\boldsymbol{\mu_{n,erasure,block} = (n * \mu_{erasure});\ \ \mu_{n,err,block} = (n * \mu_{err})} \quad \textbf{(3)}$$

$$\sigma_{n,erasure,block} = \left(\sqrt{n} * \sigma_{erasure}\right);\ \ \sigma_{n,err,block} = \left(\sqrt{n} * \sigma_{err}\right) \quad \textbf{(4)}$$

The mean and standard deviation of percent of symbols error/erasure on a block basis is just the values above divided by the number of symbols per block. The mean percentage of symbol is the same but the standard deviation on a block basis decreases by square root of the block size. So it is helpful to have larger blocks.

Given the result above on a block basis, we can look at the percentage of symbols that need to be dedicated to FEC to cover different levels of errors and erasures. For Reed Solomon, one symbol is needed to correct every symbol erased and two symbols are needed to correct every symbol error. Hence to accommodate the mean plus N times the standard deviation of errors and erasures we need to dedicate at least $FEC_{symbols,N\ \sigma}$ symbols for FEC. The calculation is shown in Equation (5). We can also calculate on a percentage basis as shown in Equation (6).

$$\boldsymbol{FEC_{symbols,N\ \sigma} = \left(\mu_{n,erasure,block} + N * \sigma_{n,erasure,block}\right) + 2 * \left(\mu_{n,err,block} + N * \sigma_{n,err,block}\right)} \quad \textbf{(5)}$$

$$\boldsymbol{percent\ FEC_{symbols,N\ \sigma} = \left(FEC_{symbols,N\ \sigma}/N_s\right)} \quad \textbf{(6)}$$

From these results, we calculate the percent of symbols that need to be dedicated to FEC given the on-air erasure and error levels seen on each test. The results for $N = 2$ to 5 standard deviations are shown in the tables below. Note that there are several block sizes used with the standard being 63 symbols per block ($N_s = 63$). However, that leaves a final block of 35 and 7 symbols if we have 1 (five 63 symbol and one 35 symbol blocks) or 2 (eleven 63 symbol and one 7 symbol blocks) second long messages, respectively. Table 3 shows the results assuming 63 symbol blocks. It presents results from the same on-air test but processed with 2 different acceptance tolerance (600 and 1000 ns). As indicated before 600 ns performs better and the benefit of that performance versus 1000 ns can be seen in the table as reduction of 5-10% in symbol needed for FEC. Table 4 shows the results for 35 and 7 symbol blocks. Not surprisingly, the smaller the blocks, the more symbols are needed for FEC to provide the same level of availability. The results of these tables can also be visualized in Figure 7 and Figure 8.

Table 3. Percent FEC per Block based on Testing and 63 Symbol Blocks (assuming independent errors per symbol)

Test date, location & acceptance tolerance	μ+ 2σ	μ+ 3σ	μ+ 4σ	μ+ 5σ
3/09/15, Ground 600 ns	37.6%	38.3%	39.0%	39.7%
3/10/15 AM, Air 600 ns	49.1%	51.1%	53.0%	55.0%
3/10/15 PM, Air 600 ns	52.7%	54.9%	57.2%	59.4%
3/09/15, Ground 1000 ns	46.1%	46.8%	47.6%	48.4%
3/10/15 AM, Air 1000 ns	56.2%	58.0%	59.7%	61.5%
3/10/15 PM, Air 1000 ns	58.9%	60.9%	62.8%	64.8%

Table 4. Percent FEC per Block based on Testing and 35 or 7 Symbol Blocks (assuming independent errors per symbol)

Test date, location & truncated block size	μ+ 2σ	μ+ 3σ	μ+ 4σ	μ+ 5σ
3/09/15, Ground 600 ns, 35 symbols	38.1%	39.0%	40.0%	40.9%
3/10/15 AM, Air 600 ns, 35 symbols	50.4%	53.1%	55.7%	58.3%
3/10/15 PM, Air 600 ns, 35 symbols	54.2%	57.2%	60.2%	63.2%
3/09/15, Ground 600 ns, 7 symbols	40.4%	42.4%	44.5%	46.5%
3/10/15 AM, Air 600 ns, 7 symbols	56.9%	62.8%	68.6%	74.5%
3/10/15 PM, Air 600 ns, 7 symbols	61.7%	68.4%	75.1%	81.8%

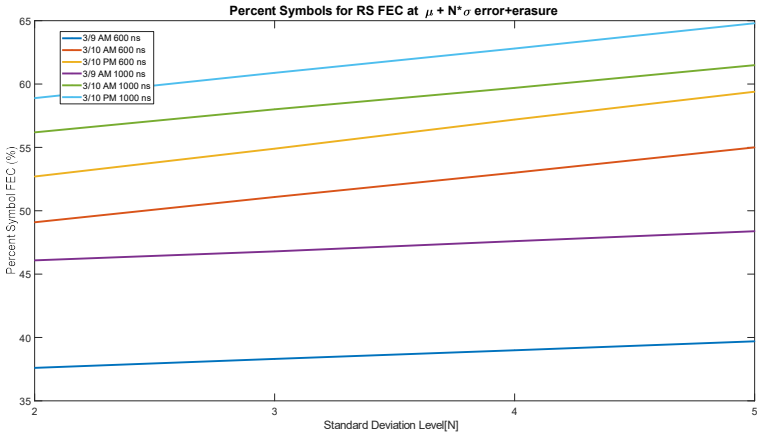

Figure 7. Percent FEC per Block based on Testing and 63 Symbol Blocks (assuming independent errors per symbol) vs. Level of Availability (in terms of mean + standard deviation of errors and erasures)

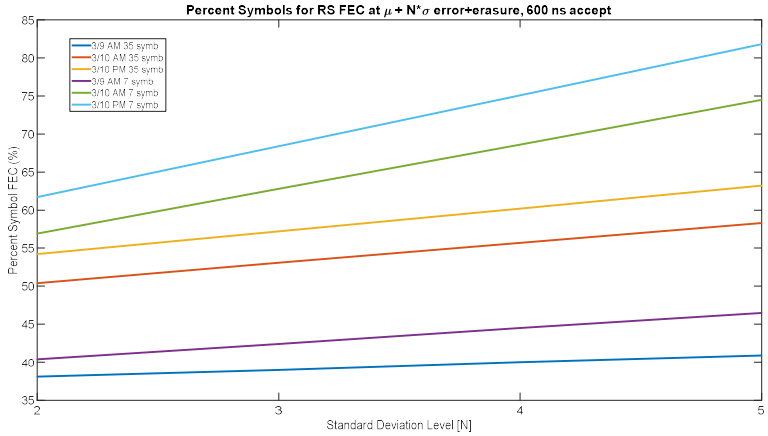

Figure 8. Percent FEC per Block based on Testing and 35 or 7 Symbol Blocks (600 ns acceptance tolerance) vs. Level of Availability (in terms of mean + standard deviation of errors and erasures)

This gets us nearly to the desired result of determining the number of symbols that we need to dedicate to FEC. An important consideration is determining the level to correct to, i.e. the number of standard deviations of error and erasure we need to protect against. The results above are on a per block basis. Ultimately, data availability is examined on a per message basis, so we translate the per block basis results to a per message basis result. Assuming that the error and erasure results are Gaussian distributions, we can calculate the result as follows where N_b is the number of blocks per message. The results for our one second, 6 block message (five 63 symbol blocks and one 35 symbol block) is shown in Table 5 which the equation to go from block availability to message availability shown in Equation (5). For 99% message availability, correcting at slightly above 3 standard deviation seems adequate. So we examine both the three and four standard deviation level as seen in Table 6. For the table, since the percentages calculated do not result in integer number of symbols for FEC, we round up to be conservative. The table shows that for FEC that covers up to 3 and 4 standard

deviations, we have 155 and 144 symbols available for data, respectively per 1 second message. In other words, we have 930 and 864 bps for 3 and 4 standard deviation FEC. The result is more than adequate to provide the data needed for timing and authentication each second.

Now we will detail the design and calculation of the FEC and data symbols for the above result. The three standard deviation design uses RS (63, 28) for each of its five 63 symbol blocks and truncated RS (35, 15) for the remaining 35 symbol block. This results in 5 blocks of 28 data symbols and one block of 15 data symbols. Another way of seeing that is the 63 and 35 symbol blocks use 35 and 20 symbols for FEC, respectively. For four standard deviation design, RS (63, 26) and RS (35, 14) is used for the 63 and 35 symbol blocks, respectively.

$$p_{avail,message} = \left(p_{avail,block}\right)^{N_b} \qquad (5)$$

Table 5. Block and Message (6 blocks) Availability for FEC Correcting to Various Standard Deviation Levels

	σ	2σ	3σ	4σ	5σ
Avail Per Block	68.3%	95.4%	99.7%	99.99%	99.99994%
Avail Per 6 Block Message	10.1%	75.6%	98.4%	99.96%	99.99966%

Table 6. FEC symbols used and Available Data for 6 Block Message

	3 σ	Available Symbols	4 σ	Available Symbols
FEC symbols (bits)	5*35+20	155 (930)	37*5+21	144 (864)

So now we can get to the final message design with FEC. To gain an additional margin for error, we use a level slightly above that determined for 5 standard deviations with over 65% based on error and erasure results in Table 3 and Table 4 for 600 ns and 35 and 63 symbol blocks. For the implemented one second message, we use RS (63, 22) and RS (35, 12) is used for the 63 and 35 symbol blocks, respectively. Hence the 63 and 35 symbol blocks have

Figure 9. Preliminary data payload for 1 second DME PPPM message after FEC (732 bits total)

3. Summary

First we conduct analysis to determine the data necessary to transmit absolute time. It was determined that at most 96 bps is needed to support this functionality with DME PPPM pseudo lite. Next we examine the error correction necessary and developed error correction for the message design.

Prior analysis and flight data is used to determine level of FEC needed. Given this, using about 60% of the symbols for FEC is sufficient in high traffic scenarios. Our design adds a little conservatism and uses a minimum of 65% of symbols for FEC. This leaves 732 bps of data for use which is enough for the small of amount data needed for absolute time and station location. After providing absolute time and station location, approximately 635 bps of data remain for other purposes. This result indicates that there can be significant additional bandwidth for authentication, even possibly post-quantum authentication schemes.

References

[1] Executive Order (EO) 13905: Executive Order on Strengthening National Resilience through Responsible Use of Positioning, Navigation, and Timing (PNT) Services, February 12 2020, https://www.whitehouse.gov/presidential-actions/executive-order-strengthening-national-resilience-responsible-use-positioning-navigation-timing-services/

[2] National Security Presidential Directive (NSPD)--39: U.S. Space-Based Position, Navigation, and Timing Policy, December 15, 2004

[3] S. Lo, P. Enge, Per, and M. Narins, “Design of a Passive Ranging System Using Existing Distance Measuring Equipment (DME) Signals & Transmitters,” NAVIGATION, Journal of The Institute of Navigation, Vol 62, No. 2, Summer 2015 pp. 131– 149. doi: 10.1002/navi.83

[4] S. Lo, P. Enge, "Signal Structure Study for a Passive Ranging System using Existing Distance Measuring Equipment (DME)," Proceedings of the 2012 International Technical Meeting of The Institute of Navigation, Newport Beach, CA, January 2012, pp. 97-130.

[5] S. Lo, Y.-H. Chen, P. Enge, W. Pelgrum, K. Li, G. Weida, A. Soelter, "Flight Test of a Pseudo Ranging Signal Compatible with Existing Distance Measuring Equipment (DME) Ground Stations", NAVIGATION, Journal of The Institute of Navigation, Volume 67, No. 3, Fall 2020

[6] Chu, Jiangping, "Mixed One-way and Two-way Ranging to Support Terrestrial Alternative Position Navigation & Timing," Proceedings of the 25th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS 2012), Nashville, TN, September 2012, pp. 2034-2039.

[7] S. Lo, Y.-H. Chen, P. Enge, “Hybrid APNT: Terrestrial Radionavigation to Support Future Aviation Needs," Proceedings of the 27th International Technical Meeting of the

Satellite Division of The Institute of Navigation (ION GNSS+ 2014), Tampa, Florida, September 2014, pp. 3029-3039.

[8] A. Neish, T. Walter, J. D. Powell, “Design and Analysis of a Public Key Infrastructure for SBAS Data Authentication.” NAVIGATION, Journal of The Institute of Navigation. Volume 66, No. 4, Winter 2019 pp. 831-844.https://doi.org/10.1002/navi.338

[9] A. Neish, T. Walter, J. D. Powell, “SBAS Data Authentication: A Concept of Operations” Published in Proceedings of the 32nd International Technical Meeting of The Satellite Division of the Institute of Navigation (ION GNSS+ 2019), Miami, Florida, September 2019

[10] A. Neish, T. Walter, P. Enge, “Quantum-resistant authentication algorithms for satellite-based augmentation systems,” NAVIGATION, Journal of The Institute of Navigation; Volume 66, No. 1, Spring 2019, pp. 199–209.https://doi.org/10.1002/navi.287

[11] M. Mendiola, J. Gillis, A. Binder, R. Haddad, "Post-Quantum Authentication Schemes," Proceedings of the 33rd International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2020), September 2020, pp. 3812-3825.

[12] K. Li, W. Pelgrum, "Enhanced DME Carrier Phase: Concepts, Implementation, and Flight-test Results", NAVIGATION, Journal of The Institute of Navigation, Vol. 60, No. 3, Fall 2013, pp. 209-220.

[13] S. Haykin, “Digital Communications,” John Wiley & Sons, 1988

[14] Reed-Solomon error correction, Wikipedia, https://en.wikipedia.org/wiki/Reed-Solomon_error_correction