LEAPFROG NAVIGATION SYSTEM

Guttorm R. Opshaug

Department of Aeronautics and Astronautics, Stanford University

ABSTRACT

There may be situations where positioning is required, but no service is available either due to signal blockage or cost of installing infrastructure.

I suggest an architecture where the navigation system is carried along as a group of people/devices are moving. For simple 2-D positioning, a set of 4 ranging transponders are needed. Initially, 2 of these devices are placed in fixed and "known" local positions (e.g. by a doorway), while the two others can roam freely into the area of interest. Their position can be calculated with respect to the "known" initial position. At some point one or both of the mobile units are commanded to stop, and one or both of the fixed units are allowed to move. As time progresses, all units take turns acting as fixed or mobile stations, and the entire group can leapfrog their way towards a common goal.

Positioning accuracies will deteriorate predictably as a function of network topology and individual ranging accuracies. Applications may range from firemen navigating through burning buildings or tunnels to groups of rovers exploring distant planets.

1 INTRODUCTION

The Leapfrog architecture crystallized from working with indoor navigation systems. This architecture is meant to provide local positioning services in areas where other navigation signals might be blocked or navigation infrastructure is too costly to install. In short, why don't we bring the navigation system along?

Inertial Navigation Systems (INS) are excellent examples of such self-contained navigation systems [1]. However, INS-es are generally very expensive, and positioning errors grow as function of time.

Several companies and academic groups work on self-configuring navigation systems. The Mars Rover project at Stanford University use GPS transceivers and motion algorithms to resolve both cycle ambiguities and individual relative locations of a rover and 3 fixed stations [2].

A group at Worchester Polytechnic Institute has suggested a real-time deployable geolocation system based on UWB transponders [3]. Such a system consists of a fixed reference station, several "thrown" pseudolites in random but fixed locations and multiple mobile users.

Æther Wire and Location Inc. are working on self-configuring networks of UWB localizers [4].

A group at the Seoul National University GPS Lab (SNUGL) also has a novel way of surveying locations of indoors pseudolites [5].

In its simplest 2-D form a Leapfrog navigation system consists of 4 units. Whereas two of the units are mobile, the two others are initially in fixed and "known" local positions. Positioning is done relative to the fixed units, and the mobiles are free to rove into the area of interest. At some point, one or both of the mobiles are commanded to stop, releasing one or both of the fixed units to move. As long as at least two units are fixed, the location of any of the 4 units can be referenced back to the initial two fixed positions. Positioning errors accumulate every time a unit switches state, but error bounds can be predicted based on ranging error statistics and local geometry.

The analysis presented in this paper is generic, but the ultimate intent is to use ranging transponders either based on GPS or UWB technology.

2 THEORY

In the following I have assumed use of transponder based ranging devices with the following basic navigation equation:

$$j_1^{(2)} = 2d_1^{(2)} + \boldsymbol{d}^{(2)} \qquad \textbf{Eq. 1}$$

In the above equation $j_1^{(2)}$ is the round-trip delay measurement between transponders *1* and *2*, $d_1^{(2)}$ is the distance between the transponders, and $d^{(2)}$ is the (known) processing delay of transponder *2* (ignoring any error terms for now). We may want to use measurements going both ways in order to reduce error effects. Thus, we can expand Eq. 1 as follows

$$\frac{j_1^{(2)}+j_2^{(1)}}{4}-\frac{d^{(1)}+d^{(2)}}{4}=d_1^{(2)} \qquad \textbf{Eq. 2}$$

If we stack all available measurements into a vector and linearize about the estimated positions of mobile units *3* and *4*, we get the matrix equation below (units *1* and *2* fixed).

$$\begin{bmatrix} \frac{j_1^{(3)}+j_3^{(1)}}{4} \\ \frac{j_1^{(4)}+j_4^{(1)}}{4} \\ \frac{j_2^{(3)}+j_3^{(2)}}{4} \\ \frac{j_2^{(4)}+j_4^{(2)}}{4} \\ \frac{j_3^{(4)}+j_4^{(3)}}{4} \end{bmatrix} - \begin{bmatrix} \frac{d^{(1)}+d^{(3)}}{4} \\ \frac{d^{(1)}+d^{(4)}}{4} \\ \frac{d^{(2)}+d^{(3)}}{4} \\ \frac{d^{(2)}+d^{(4)}}{4} \\ \frac{d^{(3)}+d^{(4)}}{4} \end{bmatrix} - \begin{bmatrix} d_{1,0}^{(3)} \\ d_{1,0}^{(4)} \\ d_{2,0}^{(3)} \\ d_{2,0}^{(4)} \\ d_{3,0}^{(4)} \end{bmatrix} = \begin{bmatrix} \mathbf{los}_1^{(3)} & \mathbf{0} \\ \mathbf{0} & \mathbf{los}_1^{(4)} \\ \mathbf{los}_2^{(3)} & \mathbf{0} \\ \mathbf{0} & \mathbf{los}_2^{(4)} \\ \mathbf{-los}_3^{(4)} & \mathbf{los}_3^{(4)} \end{bmatrix} \begin{bmatrix} \Delta x_3 \\ \Delta y_3 \\ \Delta x_4 \\ \Delta y_4 \end{bmatrix}$$

Eq. 3 Linearized navigation equations

All Δx and Δy are perturbations around the current position estimates, and all **los** are line-of-sight vectors between the different units. The equation set might not converge globally through straight iterations, but non-linear algorithms can resolve such problems [6].

3 COVARIANCE ANALYSIS

One can include previous error statistics of the initially fixed units in the covariance calculations, e.g. if those initial positions were found using GPS. In the matrix equation below, measurements on Eq. 2 form has been linearized around both fixed and mobile unit locations. However, the terms containing fixed units have been moved to the left side of the equation set since their positions already are "known."

$$\begin{bmatrix} \frac{j_1^{(3)}+j_3^{(1)}}{4} \\ \frac{j_1^{(4)}+j_4^{(1)}}{4} \\ \frac{j_2^{(3)}+j_3^{(2)}}{4} \\ \frac{j_2^{(4)}+j_4^{(2)}}{4} \\ \frac{j_3^{(4)}+j_4^{(3)}}{4} \end{bmatrix} - \begin{bmatrix} \frac{d^{(1)}+d^{(3)}}{4} \\ \frac{d^{(1)}+d^{(4)}}{4} \\ \frac{d^{(2)}+d^{(3)}}{4} \\ \frac{d^{(2)}+d^{(4)}}{4} \\ \frac{d^{(3)}+d^{(4)}}{4} \end{bmatrix} - \begin{bmatrix} d_{1,0}^{(3)} \\ d_{1,0}^{(4)} \\ d_{2,0}^{(3)} \\ d_{2,0}^{(4)} \\ d_{3,0}^{(4)} \end{bmatrix} + \begin{bmatrix} \mathbf{los}_1^{(3)} & \mathbf{0} \\ \mathbf{los}_1^{(4)} & \mathbf{0} \\ \mathbf{0} & \mathbf{los}_2^{(3)} \\ \mathbf{0} & \mathbf{los}_2^{(4)} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \Delta x_1 \\ \Delta y_1 \\ \Delta x_2 \\ \Delta y_2 \end{bmatrix} = \begin{bmatrix} \mathbf{los}_1^{(3)} & \mathbf{0} \\ \mathbf{0} & \mathbf{los}_1^{(4)} \\ \mathbf{los}_2^{(3)} & \mathbf{0} \\ \mathbf{0} & \mathbf{los}_2^{(4)} \\ \mathbf{-los}_3^{(4)} & \mathbf{los}_3^{(4)} \end{bmatrix} \begin{bmatrix} \Delta x_3 \\ \Delta y_3 \\ \Delta x_4 \\ \Delta y_4 \end{bmatrix}$$

Eq. 4 Nav. equations with fixed position

In order to more easily perform algebra on the above equation set I introduce the following substitutions.

$$A \cdot \vec{j} - B \cdot \vec{d} - \vec{d}_0 + G_F \cdot \vec{x}_F = G_M \cdot \vec{x}_M \qquad \textbf{Eq. 5}$$

Here, $\vec{j}$ and $\vec{d}$ are the vectors of round trip delay measurements and transponder processing delays. *A* and *B* are the matrices that combine $\vec{j}$ and $\vec{d}$ to Eq. 4 format. $\vec{d}_0$ holds the current distance estimates between units. Whereas, $\vec{x}_F$ and $\vec{x}_M$ contain the fixed and mobile unit locations, G_F and G_M are their respective G-matrices. Let's find the covariance of the previous equation [7].

$$E\left(\left(A\cdot\vec{j}-B\cdot\vec{d}-\vec{d}_0+G_F\cdot\vec{x}_F\right)\cdot\left(A\cdot\vec{j}-B\cdot\vec{d}-\vec{d}_0+G_F\cdot\vec{x}_F\right)^T\right)$$
$$=E\left(\left(G_M\cdot\vec{x}_M\right)\cdot\left(G_M\cdot\vec{x}_M\right)^T\right)$$

Eq. 6 Fundamental covariance

Furthermore, it is assumed that $\vec{j}$, $\vec{d}$ and $\vec{x}_F$ all are unbiased separate Independent Identically Distributed (I.I.D.) Gaussian variables. $\vec{x}_F$ is also Gaussian, since any linear sum of Gaussian variables also is Gaussian. Algebra yields.

$$A\cdot E\left(\vec{j}\vec{j}^T\right)\cdot A^T + B\cdot E\left(\vec{d}\vec{d}^T\right)\cdot B^T + G_F\cdot E\left(\vec{x}_F\vec{x}_F^T\right)\cdot G_F^T$$
$$=G_M\cdot E\left(\vec{x}_M\vec{x}_M^T\right)\cdot G_M^T$$

Let's solve for the covariances of the mobile units (assuming a generally over-determined equation set).

$$\begin{aligned} E\left(\vec{x}_M\vec{x}_M^T\right) &= \left(G_M^TG_M\right)^{-1}\cdot G_M^TA\cdot E\left(\vec{j}\vec{j}^T\right)\cdot A^TG_M\cdot\left(G_M^TG_M\right)^{-1} \\ &+\left(G_M^TG_M\right)^{-1}\cdot G_M^TB\cdot E\left(\vec{d}\vec{d}^T\right)\cdot B^TG_M\cdot\left(G_M^TG_M\right)^{-1} \\ &+\left(G_M^TG_M\right)^{-1}\cdot G_M^TG_F\cdot E\left(\vec{x}_F\vec{x}_F^T\right)\cdot G_F^TG_M\cdot\left(G_M^TG_M\right)^{-1} \end{aligned}$$

Eq. 7 Covariance solution

The above equation relates position accuracies of the mobile units to variations in ranging measurements and processing delays as well as position uncertainties of the fixed stations. The same equation can be used to propagate covariances when units switch states between being mobile and stationary. If both mobile units were to switch to fixed mode at the same time, the current $E\left(\vec{x}_M\vec{x}_M^T\right)$ could be copied directly to the new $E\left(\vec{x}_F\vec{x}_F^T\right)$. All previous cross-unit terms of $E\left(\vec{x}_F\vec{x}_F^T\right)$ should be set to zero if only one mobile and one fixed unit switch at a time. The matrices on the next page show the transition in $E\left(\vec{x}_F\vec{x}_F^T\right)$ when fixing mobile unit 3 and releasing fixed unit 1.

$$\begin{bmatrix} s_{x_1}^2 & r_{x_1y_1} & r_{x_1x_2} & r_{x_1y_2} \\ r_{x_1y_1} & s_{y_1}^2 & r_{x_2y_1} & r_{y_2y_1} \\ r_{x_1x_2} & r_{x_2y_1} & s_{x_2}^2 & r_{x_2y_2} \\ r_{x_1y_2} & r_{y_2y_1} & r_{x_2y_2} & s_{y_2}^2 \end{bmatrix} \Rightarrow \begin{bmatrix} s_{x_3}^2 & r_{x_3y_3} & 0 & 0 \\ r_{x_3y_3} & s_{y_3}^2 & 0 & 0 \\ 0 & 0 & s_{x_2}^2 & r_{x_2y_2} \\ 0 & 0 & r_{x_2y_2} & s_{y_2}^2 \end{bmatrix}$$

The lower right hand terms are left untouched from the original matrix, and the upper left hand terms are copied from the corresponding terms in $E\left(\vec{x}_M\vec{x}_M^T\right)$.

4 SIMULATIONS

I've considered 3 operational scenarios and simulated positioning performance of the leapfrog system. First, planetary exploration with a group of 4 rovers. In this case, one has freedom in optimizing geometry of the rover group. Second, a single bootstrap configuration for use by e.g. firemen in a burning building. Motion will be more random, but certain constraints will be enforced on who gets to move and who must remain stationary. Third, dual bootstrap with 2 or 3 team members. Motion is still random, but the previous constraints are relaxed.

4.1 PLANETARY EXPLORATION

Future Mars missions may include rovers that will venture away from the comforts of the mother ship landing site. Such vehicles may be sent out to collect soil and rock samples for further analysis either back at the landing site or back on Earth. Not only would it be important to know where samples were taken, but also how to find the way back to the landing site.

Self-surveying of the original location can be done with techniques described in [6].

In the following I have assumed a ranging transponder system with fundamental ranging accuracies of 10cm and processing delay variations of 10cm for all units. Furthermore, I start with a 100m baseline between the two fixed units. The mobile units keep that same spacing, move away in lines perpendicular to the baseline between the fixed units, then stop and let those units leapfrog by.

The next plot shows horizontal root mean squared (HRMS) error for one mobile unit as function of distance from reference baseline.

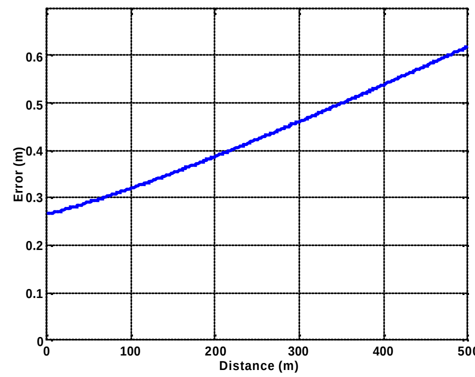

Figure 1 Horizontal RMS error vs. distance

Intuitively one might be inclined to choose the leapfrog distance (how far the mobiles go before they stop) that minimizes HRMS error (~0 in our case). That distance ensures minimum error growth per step, but might not give the minimum error growth as function of total distance traveled. This point is clearly made by the curve traces in the plot below.

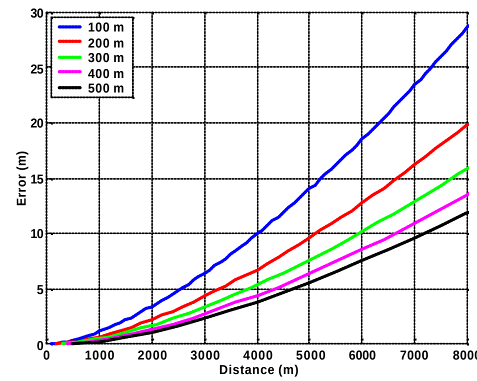

Figure 2 Error growth vs. leapfrog distance

Given a maximum error tolerance, one can find the maximum range of the system from Figure 2. The group of rovers could travel 3 km away from the landing site and still find back to their original locations to within 10 meters (half the distance out, the other half returning) if the leapfrog distance is more than 400m.

Greater system range would be gained if fundamental ranging errors and processing delay variations decrease, or the baseline distance is increased. This last point can be understood by the fact that along-track errors generally are small, but cross-track ones tend to be large. Increasing the baseline distance, will generally improve geometry in the cross-track direction. The figure on the next page shows the error ellipse of one of the mobile stations after having leapfrogged 6 km (400m leapfrog distance).

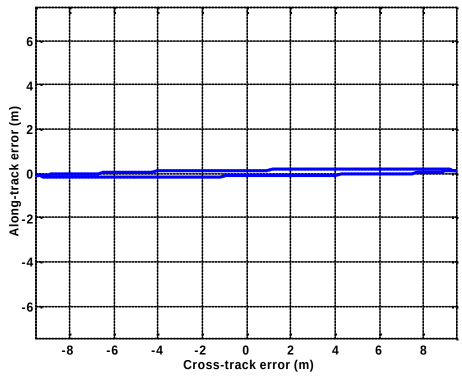

Figure 3 Error ellipse

4.2 SINGLE BOOTSTRAP

There may be situations where firemen require a navigation aid, but few or no services exist. I suggest a system where every person in a team is outfitted with their own ranging device (e.g. GPS transceiver or UWB transponder). To minimize moment-arm errors, such devices should be located close to the floor, i.e. on somebody's boot. Hence, *bootstrap*.

Operationally, a team of 4 people could be outfitted with such transponders. Each teammate would have a light indicator showing green for free-to-move, or red for stop. Two of the teammates would go just inside the doorway of a building, and their initial positions would serve as reference. The two other teammates would then be free to move farther into the building. At some point, one or both of the roaming teammates stop and one or both of the fixed ones gets to move. The process is repeated, and the unit as a whole leapfrogs into the building.

In bootstrap mode one doesn't have the luxury of optimizing path geometry, as the situation may be rather chaotic. Instead, I used Monte-Carlo simulations to predict navigation system performance. In addition to the constraints of having two people stopped at all times, I constrained mode switching to regions with favorable HDOP. Simulations were done with different HDOP thresholds and with goals of either maximizing or minimizing the baseline between stopped units. These scenarios were simulated in a 3-by-20 meter hallway. The figure below shows all mobile and fixed unit locations for one simulation run with HDOP threshold of 2.5 and the goal of maximizing fixed-unit baseline distance.

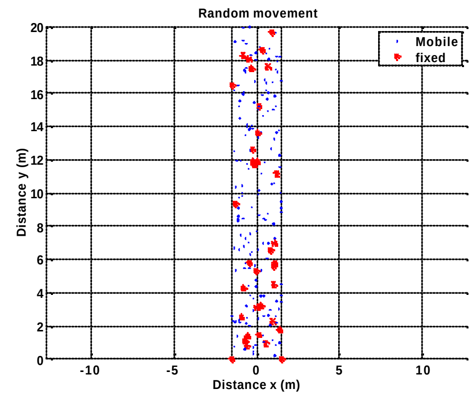

Figure 4 Unit locations

The next plot shows mobile-unit ensemble-statistics of HRMS errors after 200 simulation runs.

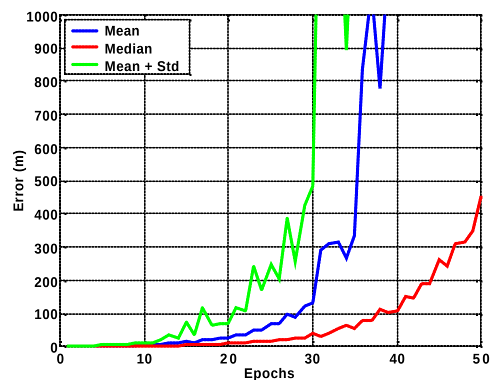

Figure 5 HRMS errors. Maximize distance

At each time step none, one or both mobile units stop, and the corresponding fixed units are released. This process is decided by the given HDOP at each mobile unit. If only one unit stops, baseline distance concerns decide which of the fixed units are released.

The next plot shows HRMS error statistics using the same HDOP threshold as above, but instead trying to minimize baseline distance.

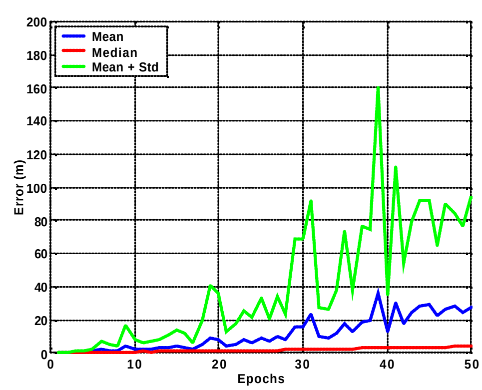

Figure 6 HRMS errors. Minimize distance

Eventually both strategies led to huge HRMS errors, but minimizing baseline distance leads to more well-behaved error statistics. In Figure 5 the 1σ error bound has a monotonically increasing trend, and it crosses 1 km after only 30 epochs. Similarly, Figure 6 shows the same general trend, but the error bound never goes much beyond 50 m even after 50 epochs.

4.3 DUAL BOOTSTRAP

The single bootstrap scenario puts a few constraints on the operators in the unit; a minimum of 4 people is required and any 2 must always be stationary.

In order to loosen some of the previous constraint, I also considered a case where each team member has two ranging transponders; one on each boot. The minimum team size shrinks to 2 (4 transponders total). Now no one has to remain fixed, but at least 2 boots must be on the ground at all times. Practically speaking this would mean walking, but no running or jumping.

Unit control would also be significantly simpler, since each boot could be outfitted with a pressure switch to tell whether it is on the ground or not (no need for stop/go light indicator).

The price to pay for relaxed operator constraints is total system complexity. Now everybody has to carry twice as many ranging units.

For the dual bootstrap simulations I literally implemented a "random walk" algorithm. Each person always makes a step that's uniformly distributed between 1 and 3 feet and ranges in a semicircle around each foot. Everything is further constrained by the same 3-by-20 hallway, and by not bumping into one another. The figure below shows possible steps for each foot.

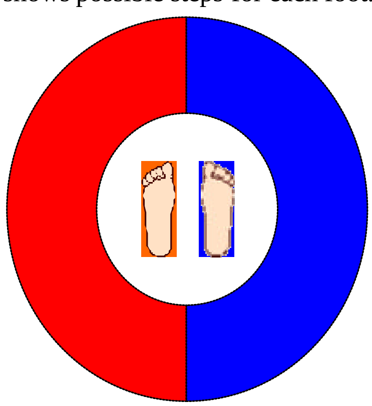

Figure 7 Dual bootstrap step

The next plot shows RMS error ellipses around the first 20 step locations.

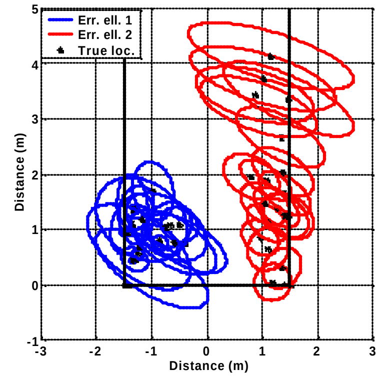

Figure 8 Error ellipses 4-unit dual bootstrap

Figure 9 shows HRMS error statistics of 200 simulation runs for 500 steps.

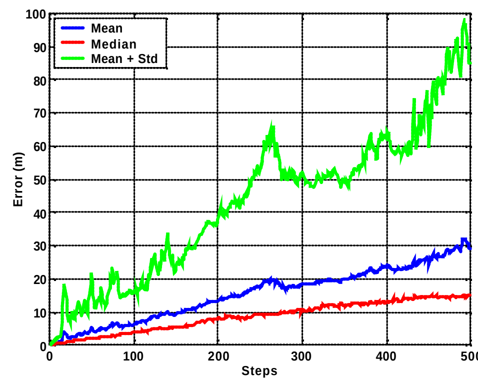

Figure 9 HRMS errors 4-unit dual bootstrap

Another simulation was done with a group of 3 firemen for comparison reasons. Error ellipses for the first 20 steps are plotted in Figure 10.

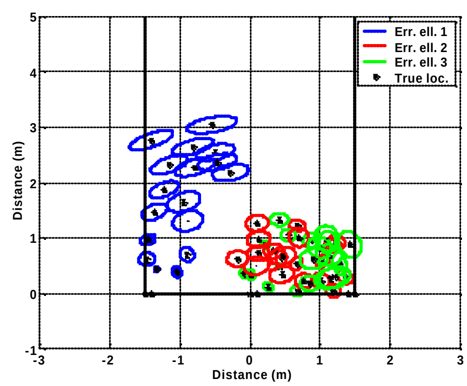

Figure 10 Error ellipses 6-unit dual bootstrap

The corresponding HRMS error statistics are plotted in the Figure 11.

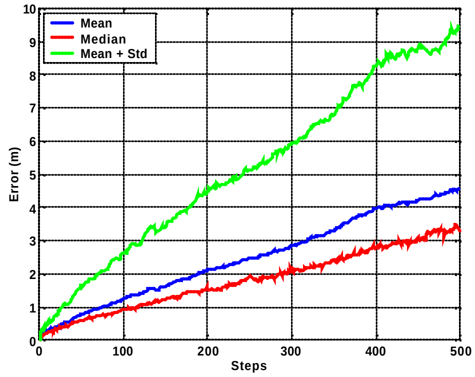

Figure 11 HRMS errors 6-unit dual bootstrap

5 CONCLUSION

In the research presented in this paper I find that the leapfrog architecture has great potential for solving navigation needs in areas where other services are scarce.

For planetary exploration, simulations show that a group of rovers could travel several kilometers away from a landing site, and still find their way back with position errors of only 10 meters.

In a constrained space scenario I find that error growth is minimized if a minimum-baseline strategy is applied for single bootstrap. It may sound more intuitive to have a large baseline distance between fixed stations. However, the area around the baseline has very poor DOP, and the larger the baseline distance the greater the area of poor DOP.

Dual bootstrap generally performs better than single bootstrap, which also constrains user operation more. Further positioning improvement is gained when the total number of navigation units is increased. This point comes as no surprise since adding units improves geometry. There is a price to pay for adding units, though. With $n \cdot (n-1)/2$ cross-link baselines, and $2n$ unit coordinates to solve for, the G-matrix increases quickly.

A leapfrog navigation system could also be integrated with other navigation devices. Augmenting standard GPS with a leapfrog system could improve total positioning performance even when SV visibility is good [8]. As GPS gets obstructed, the units go over to leapfrog mode. However, scattered GPS fixes may reset error growth, much like GPS can be used to update an INS. Given an error tolerance, curves in e.g. Figure 11 can be used to predict how often such a fix must occur.

In this paper I only considered 2-D navigation, and 3-D is left for future research.

Acknowledgement

I would like to thank everybody at the Stanford University GPS Laboratory for all their help and support. In particular I'd like to thank C. O. Lee Boyce Jr. for reading and editing all my papers.

References

[1] Gebre-Egziabher, D., Hayward R. and J. D. Powell **A Low-Cost GPS/Inertial Attitude Heading Reference System (AHRS) for General Aviation Applications** IEEE PLANS '98, April 1998

[2] LeMaster, E. and S. Rock **A Local Area GPS Pseudolite-Based Mars Navigation System** IEEE 10th International Conference on Advanced Robotics, Budapest, Hungary Aug. 2001.

[3] Orr, A. and D. Cyganski **Personnel Geo-Location Equipment** DARPA NETEX Industry day Sept. 2001.

[4] Flemming, R. and C. Kushner **Integrated Ultra-wideband localizers** UWB conf. Sept. 1999

[5] Kee, C., Yun, D., Jun, H., Parkinson, B., Pullen, S. and T. Lagenstein **Centimeter-Accuracy Indoor Navigation Using GPS-Like Pseudolites** GPS World Nov. 2001.

[6] LeMaster, E. and S. Rock **An Improved Solution Algorithm for Self-Calibrating Pseudolite Arrays** ION-NTM Jan 2002.

[7] Brown, R. and P. Hwang **Introduction to Random Signals and Applied Kalman Filtering** 3. edition. Wiley and Sons, 1997

[8] Opshaug G. and P. Enge **Integrated GPS and UWB Navigation System: (Motivates the Necessity of Non-Interference)** Accepted to IEEE UWBST 2002, May 2002