1

GPS Integrity Monitoring

Tom Temple

April 15, 2005

1 Introduction

The Global Positioning System (GPS) has proved to be very useful in a number of applications. Aircraft navigation is among the most important. But in the safety-of-life circumstance of precision approach and landing, we do not have sufficient guarantees of accuracy. While differential GPS (DGPS) is precise enough most of the time, a controller need to be able to put strict error bounds on the position estimate in order to be able to use it for precision approach.

In this work I will compare the effectiveness of various learning algorithms at estimating these error bounds in real time.

2 Problem Specification

Rather than solve the “Is it safe to land, or not?” question, I will attempt to answer a slightly more general problem, “what is the greatest the error could be? Given a set of satelite ranges (and potentialy their evolution over time), I would like to compute an upper bound on position error. This bound is allowed to underestimate less than once in 10 million trials. Simultaneously, one would want the service to be available as much of the time as possible. In other words, we would like to keep false negatives to a minimum while maintaining the strict

*Lincoln Laboratory

rule on false positives.

As the number of satellite ranges available increases, the equations that the receiver must solve become increasingly over-specified. The goal is to use this over-specification to generate a number of partially independent estimates of position. The distribution of these estimates can be used to estimate the distribution from which the position measurement (using all satellites) was selected.

3 Previous Work

I am primarily building on work by Misra and Bednarz, *Robust Integrity Monitoring*, which appeared in **GPS World** in April, 2004. They proposed an algorithm, called LGG1, which consists of three elements.

- A method of selecting subsets of satellites with good geometries
- A characterization of the distribution of subset positions
- And a “rule function that turns this distribution into an error-bound.

They demonstrated that such an algorithm, if sufficiently conservative, would give an error bound that was sufficiently stringent regardless of the size or shape of the underlying range error distribution. The recent subject of my research has been testing and improving the algorithm.

4 Proposal

I retain all three elements of the original algorithm but change each of them. For this work I will be focusing on determining the rule function. I will explore using learning techniques to determine a rule function that exploits more expressive characterizations of the subset position distributions.

The original paper used a single number to quantify the distribution of subset positions. The number that it used was the distance between the furthest two positions and they called it “scatter. The rule was a constant times this scatter.

An extension of the algorithm utilizes the fact that there are many more numbers that we can use to describe the distribution. So rather than characterize the distribution with one summary number, I use a set of what I am going to call “P-scatters computed as follows.

$$S_p = \left(\frac{\sum |x - \hat{x}|^p}{n} \right)^{1/p}$$

You will note that S_2 is the standard deviation and S_{infinity} is similar to the metric proposed by Misra. A value p need not be an integer nor be positive. Currently, the values of p that are used are chosen by hand through trial and error. Hopefully a learning algorithm can be applied to determining which values are the most telling of the distribution.

One property desirable in a rule function is that it be scale independent. If all the errors double, the scatter should double and the error bound should also double. A quick check reveals that the P-scatters have this property. So will the rule function as long as it is a linear combination of the P-scatters.
This means our rule function can be represented as a vector of weights. A vector of weights falls under the purview of a very long list of available techniques. I will apply a few of the newest and compare their performance to some of the more traditional.

5 Work Outline

The main idea is that the subset position distribution is somehow similar to the distribution from which the estimate is sampled. If we can determine how the distributions are related, we can use the position distribution to estimate the error distribution and determine an error bound. If the errors were truly Gaussian, we could simply estimate the variance, S_2, and simply multiply it by six and call it a day. Alas, they are not and it is not so easy.

For this project, I would like to try to classify this relationship. I will try to determine a function that takes a position distribution and returns an error bound and which has the properties that I described in Section **??**. I plan on using a genetic algorithm, Q-learning, neural nets and more classical least squares approaches and comparing them in simulation.

On the first spiral of work, I will generate a benchmarking framework that allows for the most meaningful comparison of the various algorithms. Most of this work I have already done for my research. The model for satellites and receivers is already done. What remains is generating a suite of cases on which to test the algorithms.

On the second spiral, I will implement and test the genetic algorithm and the recursive least squares algorithm and iron out the difficulties in applying these techniques to such an asymmetrical error-cost problem. On the third spiral I will try to implement Q-learning and a neural net. I must also bear in mind that an adequate experiment for benchmarking is a rather large simulation that is an entire weekend affair. I will be careful not to leave all the benchmarking until the end.

6 Timeline

Element	Completion date
Simulation framework	done
Genetic algorithm	4/22
Q-learning algorithm	4/29
Nueral Net	4/29
RLS algorithm	4/15
Final Benchmarking	5/4
Writeup	5/9