Improving the Energy Consumption in Mobile Phones by Filtering Noisy GPS Fixes with Modified Kalman Filters

Isaac M. Taylor and Miguel A. Labrador
Department of Computer Science and Engineering
University of South Florida
Tampa, Florida 33620
E-mail: itaylor3@mail.usf.edu; labrador@cse.usf.edu

***Abstract*— Real-time location-based tracking applications require of continuous GPS calculations and transmissions, which consume a considerable amount of the phone's battery. As a result, methods have been devised to reduce the amount of GPS calculations and transmissions without sacrificing the tracking capabilities of the applications. One of these methods is based on a state machine that dynamically changes the frequency of GPS updates according to the user direction, speed, received signal strength, and other factors. However, the state machine, although efficient in terms of energy savings, still presents one major problem: it does not take into account the presence of noise in GPS data. In order to distinguish between actual GPS data and noise, three versions of the Kalman filter have been implemented within the state machine. These modified Kalman filters remove noisy GPS fixes with little to no input from the user in a very efficient manner. The filters are discussed in detail and tested against one another to determine which one removes GPS noise better and which one reduces the energy consumption in the cellular phone more with no loss of valuable tracking data. Experiments conducted show the Adaptive Kalman Filter as the best performer. No loss of valuable tracking data is seen while it introduces a significant decrease in the number of "asleep" fixes. The Adaptive Robust Kalman Filter is the second best performer of the three filters. It shows no loss of tracking data, while a slightly less decrease in "sleep" fixes. Testing shows that the Robust Kalman Filter is the worst performer of the three. This is because the Robust Kalman Filter is the slowest version to "wake up" and make transitions to a "sleep" state.**

I. Introduction

One major concern when using mobile devices is the efficient use of the battery's energy [1]. Using the GPS to obtain real-time location information consumes a large amount of energy. It is imperative that location data be collected and transmitted only when it is valid and useful. To ensure that GPS data is only processed when necessary, CUTR implemented the Location Aware State Machine into TRACIT [2]. The state machine determines if valid and useful GPS data are being obtained and based on the GPS data and some other parameters, it automatically adjusts the frequency of GPS calculations [5]. For example, if the user is stationary, the GPS is constantly returning redundant location information, the state machine would then decrease the frequency at which GPS data are obtained. When the state machine determines that useful data are once again needed, it changes the calculation interval again, so the application's needs are met [2]. Figure 1 displays the current design of the location aware state machine.

One of the primary issues with the state machine relates to one major fallback: the GPS fixes can be very noisy. Noise is defined as unwanted data that are usually included with GPS data [12]. Noise is created by various sources such as sensor failure, weak signals, and interference [7]. The location aware state machine implemented in TRACIT does not take into account the presence of this noise, and therefore, may take inappropriate decisions in terms of the frequency at which GPS fixes are calculated.

Figure 2 displays the activity of the state machine executing on a Sanyo Pro 200 cellular phone [4]. The blue line corresponds to the time between GPS calculations and the black line corresponds to when the user is moving and when the user is stationary. The two areas circled in red correspond to errors in the state machine. In these two areas, the state machine incorrectly increased the frequency of GPS calculations since the phone was immobile.

Although the location aware state machine reduces the energy consumption by more than half, one serious concern causes the state machine to fall short of determining when to change the frequency of GPS polling: outliers can severely impact the efficiency of the state machine. Unfortunately, outliers are also very common in GPS data. An outlier is considered to be any point that falls "outside some overall pattern of distribution" [8]. Since the state machine decreases or increases the frequency of GPS calculations, incorrect data can impose two significant effects:

- It may erroneously increase the number of GPS calculations, thus wasting the cellular phone's energy unnecessarily
- It may erroneously decrease the number of GPS calculations, thus losing important tracking information for the application

Both of these effects can diminish the usefulness of TRACIT. It is important to preserve the battery's energy without compromising valuable location data. Therefore, it is imperative that

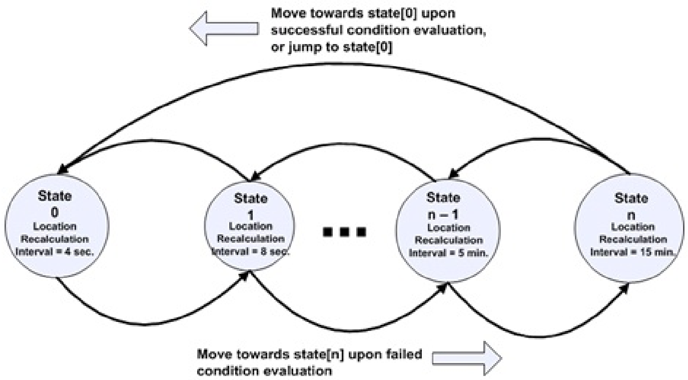

Fig. 1. The location aware state machine.

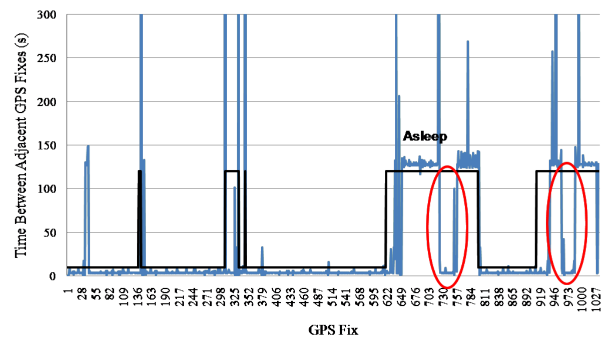

Fig. 2. Frequency of GPS calculations during normal execution of TRACIT.

the state machine receives accurate data to determine which actions to take.

In an attempt to remove these outliers from the location aware state machine, three versions of the Kalman filter have been researched and implemented. Kalman filters are used because they are fast to compute and they give a good estimate of where the user is. These filters are shown to improve the performance of the state machine by removing noise/outliers from raw GPS fixes. The rest of the paper is organized as follows. Section II explains the location aware state machine in more detail. Section III explains the Kalman filter and each of the Kalman filter versions implemented, the Robust Kalman Filter (RKF), the Adaptive Kalman Filter (AKF), and the Adaptive Robust Kalman Filter (ARKF). Section IV proves that each modified Kalman filter removes outliers in actual GPS data and shows the results of experiments with these filters integrated in the location aware state machine. Section V delves into conclusions and future work.

II. The Location Aware State Machine

The location aware state machine is responsible for determining the frequency of GPS calculations in TRACIT. Every new GPS fix that the phone receives is run through the state machine. It is important to understand how the state machine processes these points before any filter can be integrated.

The state machine has several different frequencies that it can adjust GPS calculations to. These frequencies include 4, 8, 16, 64, 150, and 256 seconds. The state machine initializes at the fastest frequency, and can slowly decrease the frequency to one calculation every 256 seconds if so needed. Also, if certain thresholds are surpassed, the state machine can move between states at a quicker rate. For instance, if 256 is the current frequency, and it is determined the user is moving at a fast rate, the state machine will quickly set the frequency to the fastest state (4 seconds) to ensure no tracking data is lost.

Each GPS fix that TRACIT receives contains some specific fields. These fields include speed, latitude, longitude, etc. Many of these fields are used by the location aware state machine, while others are simply stored in the database for future reference.

Once the state machine receives a new GPS fix, several checks are made. The first check is called the valid check. Each GPS fix contains a field that states whether new GPS coordinates were obtained or not. If the current GPS fix is invalid, the state machine lowers the frequency at which GPS calculations are made. If the point is valid, several other tests are run. Once the point is determined as a valid GPS fix, the speed field of this point is then analyzed. By looking at the current speed from this GPS point, the state machine can sometimes determine if the frequency needs to be increased or decreased. If the above tests cannot determine any useful information about this GPS fix, more detailed analysis must be done. The state machine is explained in more detail in [2].

III. The Kalman Filters

The Kalman filter has been heavily researched for use in many different fields over the past 50 years. The Kalman filter was developed by Rudolf E. Kalman who published the very detailed theory of the filter in 1960 [6]. The filter is a solution to the least-squares method of Carl Friedrich Gauss [10]. The filter is designed to predict and sort out random signals that are linear in nature [14]. The filter is the optimal estimator for linear Gaussian systems [11].

The Kalman filter is a set of mathematical equations that take in measurements (in our case, GPS coordinates) that contain noise, and produce an output that is closer to the true values of the measurements. In addition to the new predicted state, the uncertainty of these values is also within the output [14]. Welch provides the notation used for the Kalman Filter variations below, as well as an explanation of the basic Kalman Filter [13].

The Kalman filter serves as the base for all of the following three modified Kalman filters. Each of the new filters modify the original Kalman filter to obtain several enhancements, such as removing noise other than simple white noise, and determining the level of noise in the system dynamically.

The following list equations summarize the operations of the Kalman filter. These equations, which are very well-known, are included here just as a reference and for the reader to be follow the variations introduced in the following versions. For more detailed information about these equations, see [13].

$$\hat{x}_k^- = A\hat{x}_{k-1} + Bu_k \tag{1}$$

$$P_k^- = AP_{k-1}A^T + Q \tag{2}$$

$$K_k = P_k^- H^T (HP_k^- H^T + R)^{-1} \tag{3}$$

$$\hat{x}_k = \hat{x}_k^- - K_k(z_k - H\hat{x}_k^-) \tag{4}$$

$$P_k = (I - K_k H)P_k^- \tag{5}$$

where B is the control matrix and U is the input matrix; P_k^- is the *estimate error covariance*; Q is the *process noise covariance* matrix; K if the *gain* of the filter; H is sometimes called the *measurement* matrix; R is known as the *measurement noise covariance*; P_k is the *error covariance* matrix; and I is simply the identity matrix with the same dimensions as P_k^-.

A. The Robust Kalman Filter (RKF)

Ting's Kalman filter reduces the complexity of the algorithm and allows it to be used in real time environments [11]. The most notable filter modification has to do with applying a weight factor to the noise matrix R. This weight is calculated in such a way that large outliers will not affect the system, while normal data will bypass the weight. The following equations are part of Ting's Kalman filter. Like the normal Kalman Filter, this version also has two prediction equations:

$$x_k^- = A\hat{x}_{k-1} \tag{6}$$

$$P_k^- = Q \tag{7}$$

Most notably, P_k^- no longer requires any calculations. The correction equations are modified in greater length. Also, part of the Kalman gain (K) equation has been broken up into two equations for simplicity.

$$S = H_k P_k^- H_k^T + \frac{R_k}{W_k}^{-1} \tag{8}$$

$$K = P_k^- H^T S \tag{9}$$

$$X_k = X_{k-1} + K(z_k - HP_k) \tag{10}$$

$$P_k = (I - KH)P_k^- \tag{11}$$

In this version, $P_k - 1$ has become much simpler, and W has been introduced as the divisor of R in Equation 8. This weighting factor is where the extra computation comes in. The equation to compute W is below:

$$w = \frac{a + \frac{1}{2}}{b + ((z_k - Hx_k^-)^T R^{-1} (z_k - Hx_k^-))} \tag{12}$$

In the weight equation above, a and b are both constant values. These are scaling factors that determine the range of the weighted values. According to Ting, $a = b = 1$ is usually a good choice, as this assumes that most data received are correct and are not outliers. The bottom of the fraction contains the residual squared multiplied by the inverse of the current R value.

B. The Adaptive Kalman Filter (AKF)

Rutan proposes the AKF for use in Analytical Chemistry [9]. A function that uses the residuals from past measurements is used to compute R, the measurement error, at each new correction step. A sliding window is required to keep track of which residuals the filter is currently looking at. This equation would be computed before the Kalman gain (K). The equation is as follows:

$$R_k = (1/q)\left(\sum_{j=1}^{q}(v_{k-j}^2)\right) - H_k^T P_k H_k \tag{13}$$

where q is the window size that R should be computed with, v_k is the residual of the original Kalman filter, H is the predicted measurement, and P is the error covariance matrix. In short, the residual calculations from the most current back to q are summed together and subtracted from the current predictions. Consequently, if the residuals are very large, the current predicted measurement will have little effect on R. This allows the Kalman filter to be "turned off" when an outlier is encountered by the system.

One other change is included in Rutan's equation(from Ting's filter, or the RKF). The simplification of the covariance estimate matrix is also incorporated in this Adaptive Kalman Filter. By making P_k^- a static variable, divergence is not observed when an outlier is present in the system. The filter now efficiently ignores outliers. This change also simplifies the prediction function call, as P_k^- no longer needs to be computed.

C. The Adaptive Robust Kalman Filter (ARKF)

Although both the AKF and RKF should remove all types of outliers in theory, either one could be ineffective at detecting all possible outliers. This is the reason behind the creation of the Adaptive Robust Kalman Filter. This filter is a combination of the modifications used in the RKF and the AKF. By using both methods within the same filter, no outlier should go undetected within the system.

By computing both values at the same time, higher computational costs are introduced to the Kalman filter. However, this could still save battery life if it removes those outliers that went through undetected by the two other filters. Testing will need to be conducted to determine if combining both of the above filters into one is worth the computational increase, or if one of the above filters detects and removes outliers efficiently on its own.

D. Implementation in TRACIT

The location aware state machine implemented in TRACIT is based on the distance between the current GPS fix and the last valid GPS fix. Depending on how large this calculated distance is, the state machine determines whether to increase or decrease the frequency of GPS calculations. At this point, the state machine does perform some tests to check for outliers already. For instance, if the distance between the two points is greater than a certain threshold, the point is considered an outlier and no action is taken by the state machine.

E. Filter Integration within the Location Aware State Machine

Changes to the location aware state machine are kept as small as possible to ensure any changes in energy consumption are directly caused by the introduction of the filter to the system.

Most of the tests within the location aware state machine are not modified. Tests such as speed and accuracy checks are left unchanged; however, any check that requires a distance calculation has been modified. When these distance checks occur, the state machine uses a previously stored GPS point and the current GPS point to determine the distance between them. All of these checks now use the GPS fix that is taken from the filter instead of the raw GPS fix.

The state machine makes one major check on the data before anything else can be looked at. It must first check the valid variable of this GPS fix. If the fix is invalid, no computation is needed and the state machine can increase the GPS polling frequency. If the fix is valid, the filter is now executed via the `Run` method. This method takes the current latitude and longitude and returns the predicted fix from the filter. The state machine then runs all of its tests in a normal manner. As mentioned above, only several distance checks are modified within the location aware state machine.

Once the state machine has determined that no decisions can be made from looking at the speed and the accuracy values of the current GPS fix, more detailed checks are made. These checks involve the last two valid GPS points encountered. Three different distance checks are made using each of these points: the user has moved a small amount, a medium amount, or a large amount. These three checks are responsible for changing the GPS calculation frequency of the state machine. Each of these checks has been modified to now use the coordinates taken from the filter instead of the raw GPS coordinates. These are the primary modifications that have occurred within the location aware state machine.

IV. Experiments

A. Modified Kalman Filter Testing

To demonstrate that each filter removes outliers from real GPS data, each filter is run using a set of previously collected GPS points. These tests show that each of the three Kalman filter versions remove outliers without heavily modifying normal data. This dataset contains outliers that are commonly seen within TRACIT. The results of this test, depicted in Figure 3, show each of the filters moving the two outliers towards the majority of the points. For testing, the noise variables (Q and R) are initialized to .0001. This includes the individual filter testing, and the testing with TRACIT.

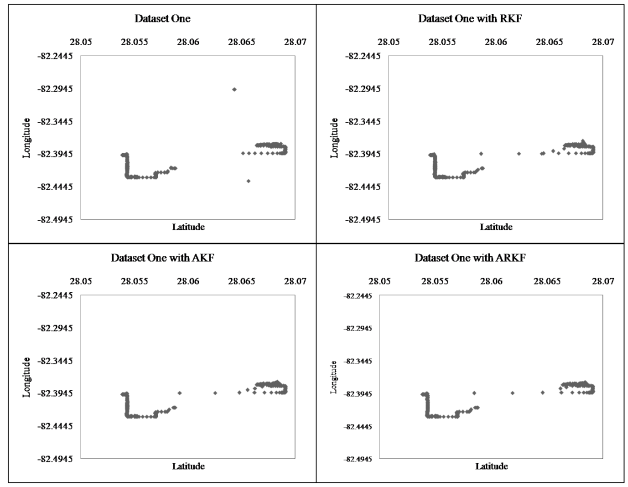

Fig. 3. Results of the modified Kalman filter experiments.

B. TRACIT Testing

The next test is conducted to determine if the integration of the modified Kalman filters improve the performance of TRACIT, and if they decrease the energy consumption while still obtaining the same amount of valid travel data as with the location aware state machine without a filter. This test involves a user carrying four phones, one with no filter, and the other three with the Robust Kalman Filter, the Adaptive Kalman Filter, and the Adaptive Robust Kalman Filter. These phones are carried for one day's worth of travel behavior. This test is conducted with four Sanyo Pro 200 cellular phones with a standard battery. These phones were carried for a total of 30 days during normal activity. The majority of the trips consist of several traveling periods, while mainly consisting of stationary periods for the phones. Several of the days contain more traveling points than others. The GPS data from each of these phones are then analyzed to produce the results.

Before any analysis of a test can occur, some basic work must be done on a dataset. Each point in this data set is then labeled as a "stationary" or as a "traveling" data point. These points are determined by referencing a travel log that states exactly when the user is moving or traveling throughout the day.

1) State Machine Performance: The first metric is called the Incorrect State Point Percentage (ISP). An ISP of the lowest possible value is desirable, as the lower the value, the more accurate the state machine is. ISP uses the time difference between two consecutive GPS fixes. This difference effectively is the frequency that GPS fixes are being calculated at that time. For example, the state machine intervals discussed in the previous section are 4, 8, 16, 64, 150, and 256 seconds. Therefore, the frequency between two GPS calculations should be one of the state machine intervals at any given time. We consider the difference of two consecutive GPS calculations to represent the current state machine state. An Incorrect State Point is a location when the state machine is at an incorrect state. For example, when the user is moving and the state machine is asleep, or stationary. Another example would be when the user is not moving and the state machine is at state 4, or awake. There are two criteria for a difference to be considered an Incorrect State Point:

- A point has a difference value of less than five and is labeled "stationary"
- A point has a difference value equal or greater than eight and is labeled "traveling"

The value five is chosen because if the location aware state machine has a value less than five, then it is considered awake. If the value is equal or greater than 8, then the state machine is in a state other than the awake state. All of the states above 8 are considered "stationary".

Also, differences that have a value of 0 or a value greater than 500 are not counted. This is because the state machine does not have an effect on these points. They are attributed to problems with obtaining a GPS fix at the time. The Incorrect State Points are summed together and divided by the total number of GPS calculations for that day.

The results of this experiment are presented in Table I, which shows that the integration of these modified Kalman filters improve the performance of the location aware state machine. The "% Difference" row of the table shows the difference with respect to the version without a Kalman Filter. For example, the RKF shows a 2.88% increase in the number of ISPs with respect to the Normal version. Of the three filters, the Adaptive Kalman Filter shows the largest performance increase of 4 percent, while the ARKF has 1.65 percent. Also, tests with lower movement generally have higher ISP values for every version tested. These lower values are a result of the period of travel where the user changes from traveling to stationary, or vice versa, contribute a much greater percentage of the total points for these tests.

2) TRACIT Energy Consumption: One other metric used involves summing up the total number of "stationary" and "traveling" points, respectively. By removing outliers, one should see a decrease in the number of "stationary" fixes calculated, while the number of "traveling" fixes should be unchanged. By lowering the amount of stationary fixes, considerable energy is saved. However, at the same time, we do not want to lose valuable travel data. By having a similar number of "traveling" fixes with and without a filter, we show that the addition of a filter does not cause a loss of travel data. Tables II and III contain the results of this experiment. This test shows the AKF having the same amount of awake fixes (.02% difference), which proves that no location data is lost. The ARKF is next with a 1 percent difference. The greatest energy savings are also with the AKF. The test shows a 17.5 percent decrease in the number of stationary points over the version without a filter. The ARKF shows a 5.5 percent decrease, while the RKF is the worst performer of the three, actually showing a increase of 24 percent.

TABLE I

INCORRECT STATE PERCENTAGE (ISP)

Test	Normal	RKF	AKF	ARKF	Movement
1	11.10%	7.80%	2.00%	8.70%	High
2	29.10%	35.45%	9.39%	12.70%	Low
3	3.25%	32.32%	0.00%	3.70%	High
4	1.46%	22.15%	2.56%	5.40%	High
5	15.79%	7.19%	3.43%	6.80%	High
6	10.13%	12.13%	12.71%	9.30%	High
7	20.96%	13.72%	11.13%	16.63%	Low
8	18.50%	20.21%	9.13%	15.40%	Low
9	14.55%	16.84%	14.07%	8.89%	Low
10	16.34%	21.50%	11.82%	12.35%	Low
11	16.88%	6.59%	8.43%	15.85%	Low
12	18.44%	14.63%	17.24%	18.83%	Low
13	10.61%	7.34%	4.43%	15.50%	High
14	17.90%	17.65%	15.95%	20.42%	Low
15	0.51%	7.36%	0.00%	1.47%	High
16	15.35%	24.47%	14.37%	6.41%	Low
17	1.18%	12.43%	0.74%	5.54%	Low
18	4.06%	20.08%	3.94%	14.10%	Low
19	11.20%	22.00%	11.45%	23.32%	Low
20	10.46%	8.86%	5.05%	5.44%	Low
21	10.14%	9.57%	4.67%	6.93%	High
22	20.49%	13.10%	16.62%	15.61%	Low
23	5.11%	17.17%	5.02%	4.09%	Low
24	9.81%	11.55%	8.97%	16.78%	Low
25	6.26%	9.24%	11.88%	5.25%	Low
26	7.37%	1.38%	0.00%	1.10%	Low
27	7.80%	24.87%	11.12%	6.56%	Low
28	3.32%	5.42%	5.06%	2.58%	High
29	26.44%	5.09%	1.77%	7.98%	Low
30	3.42%	6.33%	5.44%	4.82%	High
Average	**11.60%**	**14.48%**	**7.61%**	**9.95%**	
% Difference		**-2.88%**	**3.98%**	**1.65%**	

V. CONCLUSIONS

The three variations of the Kalman filter are embedded in a location aware state machine included in TRACIT that varies the frequency of GPS calculations and posterior transmissions according to some parameters. The main idea of the state machine is to save energy avoiding unnecessary GPS calculations and transmissions while obtaining the necessary number of fixes needed by the application. One of the major problems of the state machine is that it does not consider the noise of GPS fixes, which make the state machine to make wrong decisions. We implement three variations of the Kalman filter to filter out those noisy fixes and improve the performance of the state machine. The results from the experiments show the AKF as the best performer. No loss of valuable tracking data is seen, while it introduces a significant decrease in the number of "asleep" fixes. The ARKF is the second best performer of the three filters. It shows no loss of tracking data, while a slightly less decrease in "sleep" fixes. Testing shows that the RKF is the worst performer of the three. This is because the RKF seems to be the slowest version to "wake up" and also the slowest version to make transitions to a "sleep" state. Both of these timings could be tweaked by changing the noise values with this filter (or possibly all three). By increasing these values (from .0001 to .001), these transition time periods could be lessened, while still removing distance outliers. These

TABLE II

TRAVELING FIX COUNT

Test	Normal	RKF	AKF	ARKF	Movement
1	844	901	946	875	High
2	248	207	277	209	Low
3	1634	1561	1495	1487	High
4	1541	1590	1456	1615	High
5	2583	2573	2561	2483	High
6	2585	2564	2652	2642	High
7	333	317	317	301	Low
8	370	324	361	320	Low
9	566	555	484	487	Low
10	522	534	588	598	Low
11	130	162	94	114	Low
12	114	142	152	117	Low
13	743	739	770	731	High
14	281	287	238	299	Low
15	750	708	725	661	High
16	143	141	202	98	Low
17	609	665	585	540	Low
18	597	661	599	585	Low
19	145	181	148	173	Low
20	361	392	393	329	Low
21	706	692	712	672	High
22	271	280	299	276	Low
23	320	310	279	529	Low
24	246	313	351	265	Low
25	379	350	252	372	Low
26	159	144	104	163	Low
27	135	140	160	149	Low
28	1967	1992	1943	1929	High
29	92	165	212	172	Low
30	1373	1565	1399	1315	High
Average	**691.57**	**705.17**	**691.70**	**683.53**	
% Difference		**-1.97%**	**-0.02%**	**1.16%**	

TABLE III

STATIONARY FIX COUNT

Test	Normal	RKF	AKF	ARKF	Movement
1	155	99	53	123	High
2	476	517	265	278	Low
3	211	1065	139	350	High
4	240	829	265	346	High
5	774	409	238	356	High
6	524	593	596	464	High
7	187	113	113	138	Low
8	149	156	99	128	Low
9	401	523	325	322	Low
10	407	465	401	381	Low
11	344	248	250	334	Low
12	336	309	341	345	Low
13	303	242	246	366	High
14	177	223	132	230	Low
15	221	325	197	224	High
16	287	431	299	214	Low
17	323	389	267	335	Low
18	339	609	314	507	Low
19	227	539	223	376	Low
20	308	342	241	267	Low
21	260	348	188	208	High
22	339	216	278	294	Low
23	287	453	267	254	Low
24	294	267	251	337	Low
25	308	322	430	314	Low
26	370	363	237	382	Low
27	224	431	234	232	Low
28	352	461	371	317	High
29	430	267	240	279	Low
30	207	226	310	241	High
Average	**315.33**	**389.33**	**260.33**	**298.07**	
% Difference		**-23.47%**	**17.44%**	**5.48%**	

noise settings have less of an impact on the AKF and ARKF because these two filters constantly recalculate noise variables, while the RKF only changes the weight on the noise variables. Further evidence is seen with the Incorrect State Percentages in Table I. The RKF shows a decrease in performance from the version without a filter, while the ARKF shows a 1.65 percent increase, and the AKF shows a 4 percent increase. This is also due to the slow transition time of the RKF. By modifying the weighting values in the RKF, better results could be achieved.

REFERENCES

[1] D. P. Aguilar. A framework for evaluating the computational aspects of mobile phones. *USF Thesis*, 1:1, 2008.

[2] S. Barbeau, M. A. Labrador, A. Perez, P. Winters, N. Georggi, D. Aguilar, and R. Perez. Dynamic management of real-time location data on gps-enabled mobile phones. In *Proc. Second Int. Conf. Mobile Ubiquitous Computing, Systems, Services and Technologies UBICOMM '08*, pages 343–348, 2008.

[3] Sean J Barbeau, Miguel A Labrador, Nevine Labib Georggi, Philip L Winters, and Rafael A Perez. Trac-it: Software architecture supporting simultaneous travel behavior data collection and real-time location-based services for gps-enabled mobile phones. *Transportation Research Board Annual Meeting*, 88:09 – 3175, 2010.

[4] Sean J Barbeau, Rodney Nelson, and Mike McMullen. Location-based services - best practices and mobile applicaiton optimization. In *Sprint Open Developer's Conference*, October 2009.

[5] Sean J Barbeau, Rafael A Perez, Miguel A Labrador, Alfredo Perez, Philip L Winters, and Nevine Labib Georggi. Laisyc - a location-ware framework to support intelligent real-time applications for gps-enabled mobile phones.

[6] Rudolph Emil Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME–Journal of Basic Engineering*, 82:35–45, 1960.

[7] Xiaoying Kong. A shaping filter approach to model gps errors. *International Journal of Vehicle Information and Communication Systems*, 1:181 – 194, 2008.

[8] D.S. Moore and G.P. McCabe. *Introduction to the Practice of Statistics*. W.H Freeman & Company, 1999.

[9] Sarah C. Rutan. Adaptive kalman filtering. *Analytical Chemistry*, 63:1103A1109A, 1991.

[10] H. W. Sorenson. Least-squares estimation: from gauss to kalman. *Spectrum, IEEE*, 7(7):63 –68, july 1970.

[11] J.-A. Ting, E. Theodorou, and S. Schaal. A kalman filter for robust outlier detection. In *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems IROS 2007*, pages 1514–1519, 2007.

[12] Jo-Anne Ting, Evangelos Theodorou, and Stefan Schaal. Learning an outlier - robust kalman filter. In *Machine Learning: ECML*, volume 4701/2007 of *Lecture Notes in Computer Science*, pages 748 – 756, 2007.

[13] Greg Welch and Gary Bishop. An introduction to the kalman filter. In *ACM SIGGRAPH Intl. Conf. on Computer Graphics and Interactive Techniques*, Chapel Hill, NC, USA, 1995. University of North Carolina at Chapel Hill.

[14] Ka-Veng Yuen, Ka-In Hoi, and Kai-Meng Mok. Selection of noise paramters for kalman filter. *Earthquate Engineering and Enginerring Vibration*, 6:1671–3664, 2007.