Combined lab report for L6 and L7

Boldizsár Markotics, Tzu-Chiao Lin

1 Assignment 6 – Compute rover position using relative positioning with code and phase observations

1.1 Introduction

utilized

The purpose of this lab report is to implement a relative positioning method using GPS measurements. The implemented methods are based on the document titled "Relative positioning with GPS". We followed its steps to improve and determine the accurate position of a rover receiver relative to a reference receiver.

During the task we computed single differences for code and phase observations, we calculated double differences using a selected reference satellite and determined coefficients for least squares estimation. After these steps we computed the variance-covariance matrix and the extracted the float ambiguities. Than to obtain fixed ambiguities we utilized the LAMBDA function provided by TU Delft. Finally, we performed ambiguity validation to assess the quality of the fixed ambiguities.

By doing the previously described steps, the goal is to determine the precise position of the rover receiver in relation to the reference receiver. Our results will be presented and discussed in the next subsections of this report. We used provided data from the file "Diff AssignmentL1.xlsx," and we checked our intermediate results from the file "Intermediate results.txt" for debugging purposes.

1.2 Methods

1.2.1 Differential positioning

Relative positioning is a commonly used technique in GPS data processing that enables the determination of the position of one receiver (rover) relative to another receiver (reference) using observations from common satellites. We had to do a number of steps to implement it in our task such as synchronize the observables, compute single and double differences, estimate coefficients, perform a least squares solution, and correct the rover position (as previously stated). Differential positioning compared to regular methods offers increased accuracy, error mitigation, rapid initialization, RTK capabilities, improved integrity monitoring, and the potential for ambiguity resolution.[1]

1.2.2 Lambda function

The LAMBDA function, available from TU Delft, is a fundamental tool in our laboratory task to determine the fixed ambiguities as well as to run the ratio test.

It takes the float ambiguities vector $\hat{\mathbf{a}}$ and the corresponding variance-covariance matrix $\hat{\mathbf{Q}}$ as inputs.

As its (utilized) outputs the LAMBDA function can provide us the following parameters:

- afixed: An array of size (n $\times$ ncands) containing the estimated integer candidates. The best candidate is in the first column.
- Qzhat: The variance-covariance matrix of decorrelated ambiguities. This matrix corresponds to the fixed subset of ambiguities in the case of Partial Ambiguity Resolution (PAR) method.

By utilizing the LAMBDA function, one can effectively estimate the integer candidates for the ambiguities. [3] We will write more about the importance of this method in the discussion part of the report.

1.3 Results

After implementing the described methods, the coordinate of the receiver was determined together with the associated standard deviation by the best set of integer values of ambiguities. This result is shown in Table 1.

X (m)	Y (m)	Z (m)	σ_X (m)	σ_Y (m)	σ_Z (m)
3099279.6973	1013348.7853	5463274.4824	0.003260824268544	0.002721671714098	0.010804197117885

Table 1: The receiver coordinates and their computed standard deviations

In the next calculation, the fixed ambiguities associated with the float ambiguities were calculated. Our results are shown in Table 2.

Floating ambiguities	Best fixed ambiguities	Next best fixed ambiguities
951566.0022423886	951566	951561
1014451.143872100	1014450	1014450
-1138240.422031580	-1138244	-1138243
-9347.421405921010	-9347	-9349
-1078921.556536572e	-1078924	-1078927

Table 2: The integer values of the ambiguities and their comparison with the float ambiguities

Due to the equation of ratio test $\frac{(v^T \cdot P \cdot v)_{nextbest}}{(v^T \cdot P \cdot v)_{best}}$, for a valid critical value, the next best value should be significantly larger than the best value, which means that next best value should have more error than the best value when doing the ambiguity fixing. The normal threshold for critical value is 3, we should obtain value larger than 3 to validate the ambiguities.However, in this assignment we obtain 0.642549891917138. It means that the integer ambiguities we obtain for the best guess are not good. It requires more epochs of data to improve the results, then we can get better fixed ambiguities.

1.4 Discussion

The ambiguities are related to the cycle slip in the position calculation. We need to change the floating ambiguities into integers to count how many wavelengths is it counting. We are using the Lambda method to obtain the likely candidates of fixed ambiguities. If we obtain a bad guess on fixed ambiguities, for each satellite we obtain the error $\mathbf{n}_{error} \cdot 0.19m$, which $\mathbf{n}_{error}$ is the miscount of fixed ambiguity and 0.19m is the wavelength of the GNSS L1 band signal. Therefore, the validation of ambiguity is really important, or the error will make the calculation unreliable.

For the quality of ambiguities, we highly depend on the critical ratio we obtain by the ratio test. We can assume that if our best candidates are really close to the true situation which means that we have every element in v close to 0. $v^T \cdot P \cdot v$ will also be close to 0 in this case and have a large difference to the second candidate, which makes our critical value a large number. Hence, we can use the ratio test to ensure the quality of the integer ambiguities

2 Assignment 7 – Kalman Filtering

2.1 Introduction

This report investigates the use of Kalman filtering and smoothing techniques on a given example task. We will use these methods to compute optimal estimates of a moving vehicle's 2D position and velocity. The goal is to implement Kalman filtering and smoothing and check our results and draw conclusions from them. By implementing these techniques we would like to obtain accurate and reliable estimates of the vehicle's position and velocity. In this part of the report we will provide the used methods and their theoretical overview for this and than we will present our results, findings.

2.2 Methods

2.2.1 Kalman filtering

If we need to reduce noise on a set of measurements, Kalman filtering is a powerful mathematical tool for estimating the state of a dynamic system. It combines predictions from a system model with measurements, accounting for uncertainties in both system dynamics and measurements. In the steps of our implemented Kalman filtering algorithm, we followed the steps described in the document provided for the measurement as follows.

1. Initialisation
$$\mathbf{x}_0, \quad \mathbf{Q}_{\mathrm{x}0} = \mathrm{var}\left[\mathbf{x}_0\right] \tag{1}$$
2. Time propagation
$$\mathbf{x}_{\mathrm{k}}^{-} = \mathbf{T}_{\mathrm{k}-1,\mathrm{k}}\mathbf{x}_{\mathrm{k}-1}, \quad \mathbf{Q}_{\mathrm{x,k}}^{-} = \mathbf{T}_{\mathrm{k}-1,\mathrm{k}}\mathbf{Q}_{\mathrm{x,k}-1}\mathbf{T}_{\mathrm{k}-1,\mathrm{k}}^{\mathrm{T}} + \mathbf{Q}_{\mathrm{k}} \tag{2}$$
3. Gain calculation
$$\mathbf{K}_{\mathrm{k}} = \mathbf{Q}_{\mathrm{x,k}}^{-}\mathbf{H}_{\mathrm{k}}^{\mathrm{T}}\left[\mathbf{R}_{\mathrm{k}} + \mathbf{H}_{\mathrm{k}}\mathbf{Q}_{\mathrm{x,k}}^{-}\mathbf{H}_{\mathrm{k}}^{\mathrm{T}}\right]^{-1} \tag{3}$$
4. Measurement update
$$\mathbf{x}_{\mathrm{k}} = \mathbf{x}_{\mathrm{k}}^{-} + \mathbf{K}_{\mathrm{k}}\left[\tilde{\mathbf{L}}_{\mathrm{k}} - \mathbf{h}_{\mathrm{k}}\left(\mathbf{x}_{\mathrm{k}}^{-}\right)\right] \tag{4}$$
5. Covariance update
$$\mathbf{Q}_{\mathrm{x,k}} = \left[\mathbf{I} - \mathbf{K}_{\mathrm{k}}\mathbf{H}_{\mathrm{k}}\right]\mathbf{Q}_{\mathrm{x,k}}^{-} \tag{5}$$

The matrices and equations needed to implement the filter were created according to the provided information sheet. [2]

2.2.2 Kalman smoothing

In addition to Kalman filtering, we will also employ Kalman smoothing. This method refines past state estimates by observing and incorporating future observations. By doing this, we can improve the accuracy and reduce uncertainty in the estimates. Analogous to Kalman filtering, we also used the provided document to implement the smoothing algorithm.

When we got the last value in the state vector $\mathbf{x}_N$ and in the covariance matrix $\mathbf{Q}_{x,N}$ we started to calculate the smoothed values (equation 6-7).After this the covariance matrix $\hat{\mathbf{Q}}_{\mathrm{x,N}}$ can be calculated with equation 8. Then the standard deviation for smoothing can be derived from this matrix.[2]

$$\hat{\mathbf{x}}_{\mathrm{k}} = \mathbf{x}_{\mathrm{k}} + \mathbf{D}_{\mathrm{k}}\left[\hat{\mathbf{x}}_{\mathrm{k}+1} - \mathbf{x}_{\mathrm{k}+1}^{-}\right] \tag{6}$$
$$\mathbf{D}_{\mathrm{k}} = \mathbf{Q}_{\mathrm{x,k}}\mathbf{T}_{\mathrm{k}-1,\mathrm{k}}^{\mathrm{T}}\left(\mathbf{Q}_{\mathrm{x,k}+1}^{-}\right)^{-1} \tag{7}$$
$$\hat{\mathbf{Q}}_{\mathrm{x,k}} = \mathbf{Q}_{\mathrm{x,k}} + \mathbf{D}_{\mathrm{k}}\left[\hat{\mathbf{Q}}_{\mathrm{x,k}+1} - \mathbf{Q}_{\mathrm{x,k}+1}^{-}\right]\mathbf{D}_{\mathrm{k}}^{\mathrm{T}} \tag{8}$$

2.3 Results

Table 3 shows our results, these are the filtered and smoothed coordinates and their corresponding velocities and standard deviations.

Time (s)	$e_{filtered}$	$\sigma\ e_{filtered}$	$n_{filtered}$	$\sigma\ n_{filtered}$	$Ve_{filtered}$	$\sigma\ Ve_{filtered}$	$Vn_{filtered}$	$\sigma\ Vn_{filtered}$	$e_{smoothed}$	$\sigma\ e_{smoothed}$	$n_{smoothed}$	$\sigma\ n_{smoothed}$	$Ve_{smoothed}$	$\sigma\ Ve_{smoothed}$	$Vn_{smoothed}$	$\sigma\ Vn_{smoothed}$
0	-9.820000	10.000000	0.060000	10.000000	3.530000	3.000000	0.860000	3.000000	-2.808066	1.795754	1.624741	2.081276	4.833037	0.267691	2.554815	0.309193
2	10.051200	2.876384	5.065056	2.903707	5.053768	0.779440	1.252944	2.532695	6.859969	1.496170	6.737929	1.678597	4.834531	0.228153	2.558268	0.275980
4	17.895222	2.150347	9.532762	2.680356	4.684276	0.503325	2.083468	1.505641	16.542198	1.282483	11.857196	1.377775	4.850245	0.199555	2.559823	0.242028
6	24.412063	1.883902	20.392859	2.501884	4.153610	0.394462	3.966181	0.870273	26.297456	1.136594	16.975948	1.181473	4.907373	0.180528	2.556161	0.210500
8	33.689743	1.779033	26.141086	2.193881	4.340360	0.351258	3.552425	0.528339	36.179944	1.042616	22.080725	1.073034	4.974226	0.168389	2.549645	0.184458
10	44.433701	1.732321	28.982607	2.062542	4.658292	0.317433	2.867445	0.407510	46.168454	0.987893	27.163613	1.021893	5.012106	0.158369	2.536197	0.167991
12	53.825774	1.687345	33.917164	1.974566	4.720841	0.285848	2.789451	0.348284	56.229123	0.959620	32.242312	1.000772	5.046629	0.150887	2.544275	0.158707
14	64.785844	1.649324	36.880579	1.900455	4.862363	0.266001	2.440803	0.313746	66.337560	0.946179	37.345251	0.992892	5.057373	0.146338	2.560598	0.153669
16	76.331560	1.613406	41.678147	1.844916	5.162211	0.252215	2.472175	0.295827	76.479631	0.939928	42.509654	0.989607	5.081160	0.143598	2.603425	0.151482
18	86.774978	1.584616	46.265427	1.803725	5.115087	0.244421	2.404498	0.286351	86.598698	0.936639	47.729832	0.987515	5.035821	0.142225	2.615692	0.150631
20	97.872488	1.562944	50.142247	1.774449	5.205652	0.240341	2.303557	0.281570	96.646499	0.934430	52.980205	0.985816	5.011006	0.141605	2.632523	0.150404
22	107.312892	1.546175	56.351582	1.755736	5.089143	0.237808	2.467906	0.279789	106.639937	0.932658	58.243562	0.984766	4.983646	0.141351	2.625580	0.150412
24	116.874007	1.536728	63.189938	1.742333	4.991151	0.237619	2.654511	0.278275	116.589880	0.931207	63.465143	0.984831	4.966927	0.141490	2.591432	0.150247
26	124.709933	1.533457	68.532444	1.731815	4.724616	0.238698	2.635605	0.276698	126.521879	0.930338	68.601864	0.986077	4.965804	0.141763	2.543355	0.149994
28	134.802920	1.533975	72.465653	1.723740	4.793629	0.239912	2.492507	0.275448	136.485233	0.930381	73.650967	0.988148	4.993511	0.141995	2.503582	0.149817
30	144.789110	1.533775	80.107039	1.720132	4.827178	0.239690	2.772309	0.275600	146.491603	0.931522	78.609186	0.990463	5.008431	0.142191	2.449781	0.149880
32	155.568499	1.536583	84.971748	1.715848	4.879757	0.240725	2.659626	0.274631	156.500035	0.933953	83.440207	0.992389	4.995898	0.142858	2.381637	0.150124
34	165.624529	1.537840	90.403972	1.714105	4.906592	0.240579	2.668876	0.274687	166.491041	0.938488	88.155251	0.993482	4.993316	0.143909	2.334834	0.151465
36	175.918783	1.538594	94.869946	1.713256	4.896796	0.240420	2.541321	0.274730	176.461381	0.946843	92.786553	0.994876	4.975253	0.146002	2.299853	0.154698
38	187.224479	1.537684	97.433680	1.714001	5.101036	0.239742	2.291127	0.275217	186.414096	0.962580	97.384480	1.001257	4.976920	0.149752	2.302343	0.161326
40	195.971875	1.532951	102.251457	1.717915	4.936913	0.237794	2.314263	0.276754	196.343230	0.992193	101.991188	1.023895	4.953758	0.156175	2.304950	0.172948
42	206.234422	1.529559	105.460234	1.720913	4.920930	0.237233	2.134192	0.277197	206.239425	1.046355	106.611368	1.082981	4.941178	0.166897	2.316007	0.190526
44	216.384734	1.525142	110.014117	1.724756	4.974751	0.236221	2.173755	0.278020	216.141415	1.140471	111.270293	1.205440	4.960806	0.182946	2.341241	0.214741
46	225.884663	1.521771	116.139937	1.727737	4.969469	0.235807	2.384173	0.278359	226.075774	1.292815	115.965088	1.416219	4.974057	0.205832	2.351314	0.244498
48	236.014466	1.521451	120.661358	1.727974	4.963927	0.236573	2.345076	0.277686	236.014466	1.521451	120.661358	1.727974	4.963927	0.236573	2.345076	0.277686

Table 3: The filtered and smoothed value

To illustrate the values obtained, they are shown in Figure 1. Here one can see the differences between the measured, filtered and smoothed trajectories.

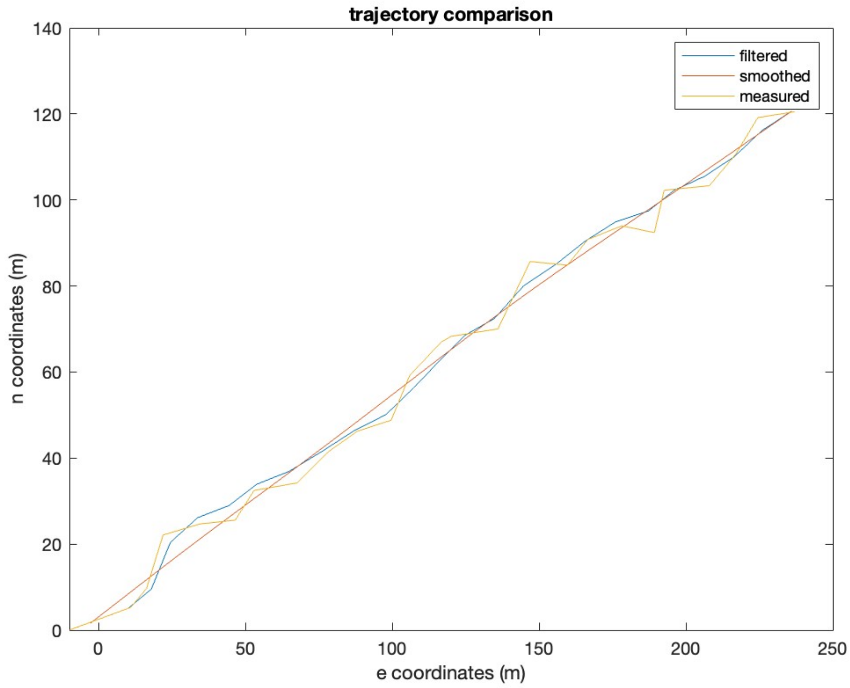
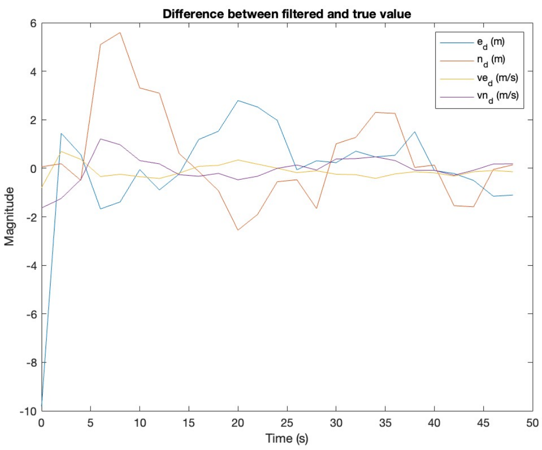
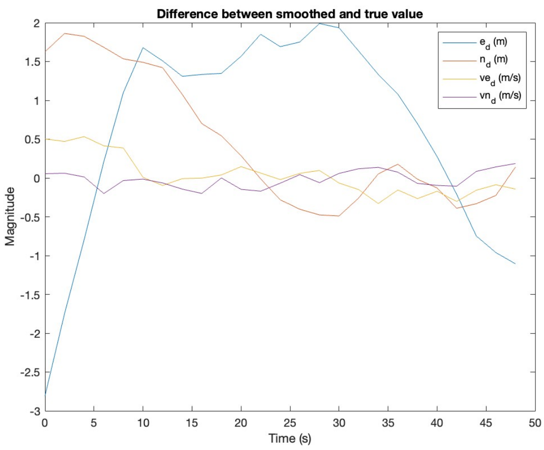


Figure 1: Filtered, smoothed, and Measured trajectory

The state variables were then compared in true-filtered and true-smoothed arrangements. The results are shown in Figure 2.



(a) Difference of filtered and true value



(b) Difference of smoothed and true value

Figure 2: Differences between the true, smoothed and filtered state variables

The figures show that the filtered values show a smaller difference compared to the true values, and the smoothed values are smoothest as expected.

2.4 Discussion

PSD will affect the matrix $\mathbf{Q}_k$, which will largely affect the standard deviations and change the trajectories. If we take a larger PSD value, the smoothed and filtered trajectories will be similar to the measured trajectory, and we will have bigger standard deviations, which means we are also farther from the true values. On the other hand, smaller PSD will give us a closer result to the true trajectories.

The standard deviation of measurements will affect the construction of the R matrix, which is the measurement noise covariance matrix. If the matrix has a larger value, the corresponding signal will affect the results much less. Therefore, the bigger the value of the standard deviation of measurements, the smaller the difference between our calculation to the true value.

For the standard deviation of initial state variables, it works in the same mechanism as the measurement noise covariance matrix. The larger the value, the more likely it will affect the output. Therefore, if we put a larger value, the Kalman filter will weigh the initial states less. On the other hand, if we have a smaller value for the standard deviation of initial state variables, the Kalman filter will weight it more.

References

[1] Ken Doucet et al. "A Real-Time GPS-Based Differential Positioning System". In: *Kinematic Systems in Geodesy, Surveying, and Remote Sensing.* Ed. by Klaus-Peter Schwarz and Gérard Lachapelle. New York, NY: Springer New York, 1991, pp. 178–187. ISBN: 978-1-4612-3102-8.

[2] Milan Horemuž. *Kalman filtering.*

[3] *Verhagen, S., Li, B., and Teunissen, P.J.G. (2012). LAMBDA - Matlab implementation, version 3.0. Delft University of Technology and Curtin University.*