

**THE REPUBLIC OF TURKEY
BAHÇEŞEHİR UNIVERSITY**

**INERTIAL SENSOR FUSION FOR
3D CAMERA TRACKING**

M.S. Thesis

NURİ ÖZER

İSTANBUL, 2012

**T.C.
BAHÇEŞEHİR ÜNİVERSİTESİ**

**THE GRADUATE SCHOOL OF NATURAL AND APPLIED
SCIENCES**

ELECTRICAL AND ELECTRONICS ENGINEERING

**INERTIAL SENSOR FUSION FOR
3D CAMERA TRACKING**

M.S. Thesis

NURİ ÖZER

**Supervisor: Assoc. Prof. Çiğdem EROĞLU ERDEM
Co-Supervisor: Prof. Dr. A. Tanju ERDEM**

İSTANBUL, 2012

T.C.
BAHÇEŞEHİR ÜNİVERSİTESİ

THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

ELECTRICAL AND ELECTRONICS ENGINEERING

Title of Thesis: Inertial Sensor Fusion for 3D Camera Tracking
Name/Last Name of the Student: Nuri ÖZER
Date of Thesis Defense: February 7, 2012

The thesis has been approved by the Graduate School of Natural and Applied Sciences.

Assoc. Prof. Tunç BOZBURA
Director

This is to certify that we have read this thesis and that we find it fully adequate in scope, quality and content, as a thesis for the degree of Master of Science.

Assoc. Prof. Ufuk TÜRELİ
Program Coordinator

Examining Committee Members:

Signature

Assoc. Prof. Çiğdem EROĞLU ERDEM

Prof. Dr. A. Tanju ERDEM

Asst. Prof. A. Özer ERCAN

Asst. Prof. Devrim ÜNAY

Asst. Prof. K. Egemen ÖZDEN

ACKNOWLEDGEMENTS

I would like to start by thanking my supervisor Assoc. Prof. ıgdem Erođlu Erdem and my co-supervisor Prof. A. Tanju Erdem for all their supports. Especially Prof. A. Tanju Erdem was abundantly helpful and offered invaluable assistance, support and guidance

I would like to express my deepest gratitudes to Asst. Prof. A. zer Ercan for his help and guidance. My special thanks to Asst. Prof. Hasan Fatih Uđurdađ for his support during my masters studies.

I would like to thank to my thesis committee members for their time. I would also like to thank to my group members, Ahmet Kermen and Instructor Tarkan Aydın for sharing their knowledge and their invaluable assistance.

My special thanks also goes to VESTEK Electronics R&D Corp. and their members, specially Mr. Metin Salt and all my colleagues.

In addition I would like to thank TBİTAK for supporting me with BİDEB scholarship program in 2009 and 2010.

Finally, I would like to thank my family for their unending support and encouragement.

İstanbul, February 2012

Nuri zer

ABSTRACT

INERTIAL SENSOR FUSION FOR 3D CAMERA TRACKING

Nuri Özer

Electrical and Electronics Engineering

Thesis Advisor: Assoc. Prof. Çiğdem Eroğlu Erdem

Thesis Co-Advisor: Prof. Dr. A. Tanju Erdem

February 2012, 150 pages

3D motion tracking becomes more important in computer vision with increase of robotics and augmented reality's (AR) applicable areas such as medical education, remote robot control, entertainment and cultural heritage. In order to achieve a realistic feeling of immersion, the rendering of the virtual content has to be in alignment with real objects in the video and this requires a high-accuracy 3D tracking. The methods using only camera measurements generally perform well at slow camera motion; however they become less accurate at high velocities and accelerations due to motion blur. Inertial sensors on the other hand measure the derivatives of the camera pose and hence can be employed to improve the tracking performance at high velocities and accelerations, but cannot perform well at slow motion because of the error drift. Therefore, we present a high-accuracy 3D camera tracking method using inertial sensors but not require placing any devices or points on the scene. 3D information of scene where 3D motion tracking is done is previously known. The method consists of an Extended Kalman filter (EKF) that fuses the information from visual and inertial sensors. A hybrid filter combining the Bayesian filter and the direct linear transformation (DLT) is also used instead of EKF. The biases of the inertial sensors are also considered during the motion. In addition to performance comparison of these two filter, the performance of using both or one of accelerometer and gyroscope measurements as control input is compared to using both or one of accelerometer and gyroscope measurements as measurement. It is concluded via simulations that using inertial sensors in 3D camera tracking gives more accurate results and using inertial sensors as measurement or control input does not affect the performance of 3D camera tracking, while providing a lower complexity tracker. Also, EKF always performs better than the hybrid filter in simulations.

Keywords: 3D Motion Tracking, EKF, Visual and Inertial Sensor Fusion, Hybrid Filter, DLT

ÖZET

3B KAMERA TAKİBİ İÇİN EYLEMSİZLİK ALGILAYICILARININ BİRLEŞTİRİLMESİ

Nuri Özer

Elektrik-Elektronik Mühendisliği

Tez Danışmanı: Doç. Dr. Çiğdem Eroğlu Erdem

Tez II. Danışmanı: Prof. Dr. A. Tanju Erdem

Şubat 2012, 150 sayfa

Robotların ve eklenmiş gerçeklik uygulamalarının tıp eğitimi, robotların uzaktan kullanımı, eğlence ve kültürel miras gibi kullanım alanlarının artması ile birlikte, 3B (3 boyutlu) takip sistemlerinin Bilgisayarlı Görü alanında önemi biraz daha artmaktadır. Eklenmiş gerçeklik uygulamalarında gerçeklik hissinin yüksek olması için canlandırma sırasında kullanılan sanal karakterlerin mekan içerisinde doğru bir şekilde hizalanması çok önemlidir. Bunun için 3B takip sisteminin doğruluğu artırılmalıdır. Sadece video verisi kullanan 3B takip sistemleri hızlı hareketin olduğu durumlarda görüntü çok değişken olacağından, yeterince iyi izleme sonuçları vermeyebilirler. Eylemsizlik algılayıcıları ise hızlı hareket olan durumlarda iyi izleme yapabilirler, ancak az hareketin olduğu durumlarda ise ölçüm hatalarının birikmesi nedeniyle iyi çalışmayabilirler. Bu sebeple, hareket takibinin doğruluğunu artırmak için bu tezde eylemsizlik algılayıcılarının verilerinden de yararlanılacaktır. Aynı zamanda bu takip sistemi için mekana herhangi bir cihaz veya işaret yerleştirmeye gerek duyulmamaktadır. 3B hareket takibi video kameralar ve eylemsizlik algılayıcıları kullanılarak yapılmıştır. Hareket takibinin yapılacağı mekanın 3B bilgisi önceden çıkarılmıştır. Kamera verilen gelen verilerle eylemsizlik algılayıcılarından gelen veriler bir döngüsel Bayes kestirimi çerçevesinde birleştirilmiştir. Ayrıca, bu tezde 3B hareket takibi için hareketten yapı çıkarma yöntemi ile döngüsel Bayes süzgeçleme yöntemini birleştiren karma bir süzgeçte geliştirilmiştir. Eylemsizlik algılayıcılarının sapma modelleri de hareket takibi sırasında göz önüne alınmıştır. Döngüsel Bayes süzgeç ile karma süzgeçin performanslarının karşılaştırılmasının yanında eylemsizlik algılayıcılarından gelen verilerin Bayes veya karma süzgeçte ölçüm yerine kontrol girdisi olarak kullanılmasının takip performansını nasıl etkilediği gözlenmiştir. Simülasyonlar sonuçlarına bakarak eylemsizlik algılayıcılarının 3B hareket takibinde kullanılması daha doğru sonuçlar bulmasını sağlamış, eylemsizlik algıyacılarından gelen verilerin ölçüm veya kontrol girdisi olarak kullanılması performansı neredeyse

hiç etkilememesine rağmen süzgeç kullanımdaki karmaşıklığı azaltarak maliyeti düşürdüğü gözlemlenmiştir. Bununla birlikte döngüsel Bayes süzgeç kullanmak karma süzgeç kullanmaya göre her zaman daha iyi sonuçlar verdiği gözlemlenmiştir.

Anahtar Kelimeler: 3B Hareket Takibi, Döngüsel Bayes Kestirimi, Görsel ve Eylemsizlik Algılayıcı Verilerinin Birleştirilmesi, Karma Süzgeç, Hareketten Yapı Oluşturma Yöntemi

TABLE OF CONTENTS

LIST OF TABLES	XI
LIST OF FIGURES	XII
LIST OF ABBREVIATIONS	XVI
1. INTRODUCTION.....	1
1.1 MOTIVATION.....	1
1.2 SCOPE OF THE THESIS.....	2
1.3 LITERATURE REVIEW	4
1.3.1 Tracking Methods Using GPS & GSM.....	4
1.3.2 Tracking Methods Using IR & RFID.....	4
1.3.3 Tracking Methods Using Inertial Sensors	4
1.3.4 Tracking Methods Using Vision Data with Visual Markers.....	5
1.3.5 Tracking Methods Using Vision Data without Visual Markers	5
1.3.6 Hybrid Tracking Methods.....	6
1.3.7 Map Estimation Methods	7
1.4 CONTRIBUTIONS OF THE THESIS	9
1.5 OUTLINE OF THE THESIS.....	9
2. BACKGROUND	11
2.1 BAYESIAN FILTERING	11
2.1.1 Kalman Filter	12
2.1.2 Extended Kalman Filter	14
2.2 USAGE OF INERTIAL SENSORS IN BAYESIAN FILTER.....	17
2.3 INERTIAL SENSORS	18
2.3.1 Gyroscope	18
2.3.2 Accelerometer	19
2.4 COMBINATION OF CAMERA AND INERTIAL SENSORS.....	20
2.5 FEATURE POINTS DETECTION	21
2.6 CAMERA GEOMETRY	22
2.7 DIRECT LINEAR TRANSFORMATION	26
3. BAYESIAN FILTER DESIGN.....	28
3.1 EKF DESIGN.....	28

3.1.1 Both Angular Velocity and Linear Acceleration Data Used As Measurement.....	29
3.1.2 Angular Velocity Data Used As Control Input, Linear Acceleration Data Used As Measurement.....	31
3.1.3 Angular Velocity Data Used As Measurement, Linear Acceleration Data Used As Control Input	33
3.1.4 Both Angular Velocity and Linear Acceleration Data Used As Control Input.....	35
3.1.5 Angular Velocity Data Used As Measurement.....	37
3.1.6 Angular Velocity Data Used As Control Input	39
3.1.7 Linear Acceleration Data Used As Measurement	41
3.1.8 Linear Acceleration Data Used As Control Input.....	43
3.1.9 Both Angular Velocity and Linear Acceleration Data Not Used.....	45
3.2 HYBRID FILTER DESIGN	46
3.2.1 Both Angular Velocity and Linear Acceleration Data Used As Measurement.....	47
3.2.2 Angular Velocity Data Used As Control Input, Linear Acceleration Data Used As Measurement.....	48
3.2.3 Angular Velocity Data Used As Measurement, Linear Acceleration Data Used As Control Input	49
3.2.4 Both Angular Velocity and Linear Acceleration Data Used As Control Input.....	50
3.2.5 Angular Velocity Data Used As Measurement.....	51
3.2.6 Angular Velocity Data Used As Control Input	52
3.2.7 Linear Acceleration Data Used As Measurement	53
3.2.8 Linear Acceleration Data Used As Control Input.....	54
3.2.9 Both Angular Velocity and Linear Acceleration Data Not Used.....	55
4. EXPERIMENTAL SETUP AND RESULTS	57
4.1 SIMULATION STRUCTURE	57
4.2 NOISES	59
4.2.1 Measurement Noises	60
4.2.2 Time Model Noises	61

4.3 EKF RESULTS.....	61
4.3.1 Both Angular Velocity and Linear Acceleration Data Used As Measurement.....	62
4.3.2 Angular Velocity Data Used As Control Input, Linear Acceleration Data Used As Measurement.....	64
4.3.3 Angular Velocity Data Used As Measurement, Linear Acceleration Data Used As Control Input	66
4.3.4 Both Angular Velocity and Linear Acceleration Data Used As Control Input.....	68
4.3.5 Angular Velocity Data Used As Measurement.....	70
4.3.6 Angular Velocity Data Used As Control Input	72
4.3.7 Linear Acceleration Data Used As Measurement.....	74
4.3.8 Linear Acceleration Data Used As Control Input.....	76
4.3.9 Both Angular Velocity and Linear Acceleration Data Not Used.....	78
4.4 HYBRID FILTER RESULTS	79
4.4.1 Both Angular Velocity and Linear Acceleration Data Used As Measurement.....	80
4.4.2 Angular Velocity Data Used As Control Input, Linear Acceleration Data Used As Measurement.....	82
4.4.3 Angular Velocity Data Used As Measurement, Linear Acceleration Data Used As Control Input	84
4.4.4 Both Angular Velocity And Linear Acceleration Data Used As Control Input.....	86
4.4.5 Angular Velocity Data Used As Measurement.....	88
4.4.6 Angular Velocity Data Used As Control Input	90
4.4.7 Linear Acceleration Data Used As Measurement	92
4.4.8 Linear Acceleration Data Used As Control Input.....	94
4.4.9 Both Angular Velocity And Linear Acceleration Data Not Used.....	96
4.5 COMPARISON OF EKF AND HYBRID FILTER.....	97
4.6 ACTUAL EXPERIMENT SETUP	104
4.7 ACTUAL EXPERIMENT RESULTS	108
5. CONCLUSIONS AND FUTURE WORKS.....	111

REFERENCES	113
APPENDIX A. CAMERA MEASUREMENTS	119
APPENDIX B. BAYESIAN FILTER LINEARIZATION	130
APPENDIX C. QUATERNIONS	142
CURRICULUM VITAE	149

LIST OF TABLES

Table 4.1: RMSE values from EKF and hybrid filter simulations.....	98
Table 4.2: Standard deviation values from EKF and hybrid filter simulations	99

LIST OF FIGURES

Figure 1.1: Examples of AR systems	1
Figure 1.2: General structure of the algorithm.....	2
Figure 1.3: Hybrid algorithm	3
Figure 2.1: The workflow of inertial navigation system.....	19
Figure 2.2: Camera-inertial sensor 3D coordinates and relations	20
Figure 2.3: Standard perspective projection.....	22
Figure 2.4: Transformation from world coordinate to camera coordinate.....	24
Figure 2.5: Camera calibration representation	24
Figure 4.1: Translational movement	57
Figure 4.2: Rotational movement.....	58
Figure 4.3: Estimated and true motion of the camera when both angular velocity and linear acceleration data are used as measurement in EKF	62
Figure 4.4: Change in RMSE of the position and orientation of the camera with rotational and translational period of the camera when both angular velocity and linear acceleration data are used as measurement in EKF.....	63
Figure 4.5: Estimated and true motion of the camera when angular velocity data is used as control input, linear acceleration data is used as measurement in EKF...	64
Figure 4.6: Change in RMSE of the position and orientation of the camera with rotational and translational period of the camera when angular velocity data is used as control input, linear acceleration data is used as measurement in EKF	65
Figure 4.7: Estimated and true motion of the camera when angular velocity data is used as measurement, linear acceleration data is used as control input in EKF ..	66
Figure 4.8: Change in RMSE of the position and orientation of the camera with rotational and translational period of the camera when angular velocity data is used as measurement, linear acceleration data is used as control input in EKF	67
Figure 4.9: Estimated and true motion of the camera when both angular velocity and linear acceleration data are used as control input in EKF.....	68

Figure 4.10: Change in RMSE of the position and orientation of the camera with rotational and translational period of the camera when both angular velocity and linear acceleration data are used as control input in EKF	69
Figure 4.11: Estimated and true motion of the camera when angular velocity data is used as measurement in EKF	70
Figure 4.12: Change in RMSE of the position and orientation of the camera with rotational and translational period of the camera when angular velocity data is used as measurement in EKF.....	71
Figure 4.13: Estimated and true motion of the camera when angular velocity data is used as control input in EKF.....	72
Figure 4.14: Change in RMSE of the position and orientation of the camera with rotational and translational period of the camera when angular velocity data is used as control input in EKF.....	73
Figure 4.15: Estimated and true motion of the camera when linear acceleration data is used as measurement in EKF	74
Figure 4.16: Change in RMSE of the position and orientation of the camera with rotational and translational period of the camera when linear acceleration data is used as measurement in EKF	75
Figure 4.17: Estimated and true motion of the camera when linear acceleration data is used as control input in EKF	76
Figure 4.18: Change in RMSE of the position and orientation of the camera with rotational and translational period of the camera when linear acceleration data is used as control input in EKF.....	77
Figure 4.19: Estimated and true motion of the camera when both angular velocity and linear acceleration data are not used in EKF.....	78
Figure 4.20: Change in RMSE of the position and orientation of the camera with rotational and translational period of the camera when both angular velocity and linear acceleration data are not used in EKF	79
Figure 4.21: Estimated and true motion of the camera when both angular velocity and linear acceleration data are used as measurement in hybrid filter.....	80

Figure 4.22: Change in RMSE of the position and orientation of the camera with rotational and translational period of the camera when both angular velocity and linear acceleration data are used as measurement in hybrid filter	81
Figure 4.23: Estimated and true motion of the camera when angular velocity data is used as control input, linear acceleration data is used as measurement in hybrid filter.....	82
Figure 4.24: Change in RMSE of the position and orientation of the camera with rotational and translational period of the camera when angular velocity data is used as control input, linear acceleration data is used as measurement in hybrid filter	83
Figure 4.25: Estimated and true motion of the camera when angular velocity data is used as measurement, linear acceleration data is used as control input in hybrid filter.....	84
Figure 4.26: Change in RMSE of the position and orientation of the camera with rotational and translational period of the camera when angular velocity data is used as measurement, linear acceleration data is used as control input in hybrid filter	85
Figure 4.27: Estimated and true motion of the camera when both angular velocity and linear acceleration data are used as control input in hybrid filter	86
Figure 4.28: Change in RMSE of the position and orientation of the camera with rotational and translational period of the camera when both angular velocity and linear acceleration data are used as control input in hybrid filter	87
Figure 4.29: Estimated and true motion of the camera when angular velocity data is used as measurement	88
Figure 4.30: Change in RMSE of the position and orientation of the camera with rotational and translational period of the camera when angular velocity data is used as measurement.....	89
Figure 4.31: Estimated and true motion of the camera when angular velocity data is used as control input in hybrid filter	90

Figure 4.32: Change in RMSE of the position and orientation of the camera with rotational and translational period of the camera when angular velocity data is used as control input in hybrid filter	91
Figure 4.33: Estimated and true motion of the camera when linear acceleration data is used as measurement in hybrid filter.....	92
Figure 4.34: Change in RMSE of the position and orientation of the camera with rotational and translational period of the camera when linear acceleration data is used as measurement in hybrid filter	93
Figure 4.35: Estimated and true motion of the camera when linear acceleration data is used as control input in hybrid filter	94
Figure 4.36: Change in RMSE of the position and orientation of the camera with rotational and translational period of the camera when linear acceleration data is used as control input in hybrid filter	95
Figure 4.37: Estimated and true motion of the camera when both angular velocity and linear acceleration data are not used in hybrid filter	96
Figure 4.38: Change in RMSE of the position and orientation of the camera with rotational and translational period of the camera when both angular velocity and linear acceleration data are not used in hybrid filter.....	97
Figure 4.39: RMSE (position) comparison of different usage of EKF and hybrid.....	100
Figure 4.40: RMSE (orientation) comparison of different usage of EKF and hybrid ..	101
Figure 4.41: RMSE (position) comparison of different usage of EKF and hybrid.....	102
Figure 4.42: RMSE (orientation) comparison of different usage of EKF and hybrid ..	103
Figure 4.43: Simulation setup	104
Figure 4.44: Image sequences used to find 3D map of the scene	105
Figure 4.45: Camera calibration pattern.....	106
Figure 4.46: Positions of the inertial sensors and camera on the setup	107
Figure 4.47: Relations of the inertial sensors' coordinate systems.....	108
Figure 4.48: Feature point search for best match in an ellipse.....	109
Figure 4.49: Projected and detected feature points	109

LIST OF ABBREVIATIONS

AR	: Augmented Reality
VR	: Virtual Reality
HMD	: Head Mounted Display
3D	: 3 Dimensional
SLAM	: Simultaneous Localization and Mapping
EKF	: Extended Kalman Filter
GPS	: Global Positioning System
GSM	: Global System for Mobile Communication
UMTS	: Universal Mobile Telecommunications System
IR	: Infrared
RF	: Radio Frequency
RFID	: Radio Wave Transmitters
DLT	: Direct Linear Transformation
RMSE	: Root Mean Square

1. INTRODUCTION

1.1 MOTIVATION

Augmented reality (AR) is a system which enables users to see real and virtual objects together in the same place. What distinguishes AR systems from virtual reality (VR) systems is that the image is created entirely using computer graphics in VR. AR defines a point between the real world and the virtual reality. Instead of replacing the real world totally, AR systems require to complete and enrich the real world (Azuma 1997).

AR is applicable to many areas such as medical education, remote control, entertainment, and cultural heritage (Papagiannakis *et al.* 2007; Papagiannakis *et al.* 2005; Azuma *et al.* 2001; Azuma 1997). In Figure 1.1.1, the examples of the AR that are the revitalization of historic cultural heritage and the placement of the virtual furniture in a room can be seen.

Figure 1.1: Examples of AR systems



Source : Papagiannakis *et al.* 2007; Davison *et al.* 2007

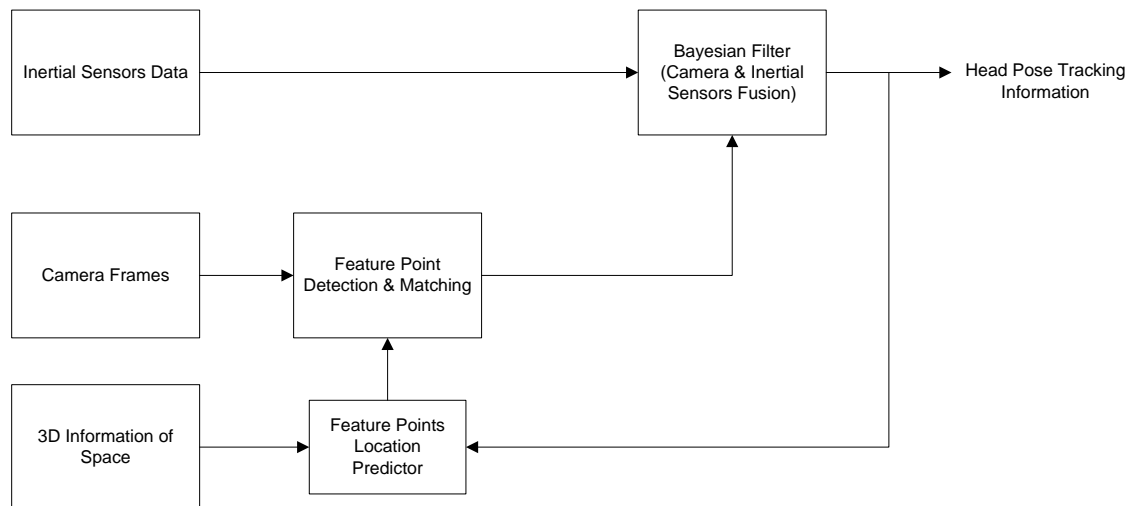
Head mounted displays (HMDs) are commonly used in AR applications to display the virtual objects on top of the real world. The rendering of the virtual content must be in line with the real objects in the video to maintain a realistic feeling of immersion. This can be done only with accurate tracking of the 3D pose of the user's head (Papagiannakis *et al.* 2005; Azuma 1997). Otherwise, it is hard to convince the human's

sense of sight to that the real objects and the virtual objects created by computer are together in the same place.

1.2 SCOPE OF THE THESIS

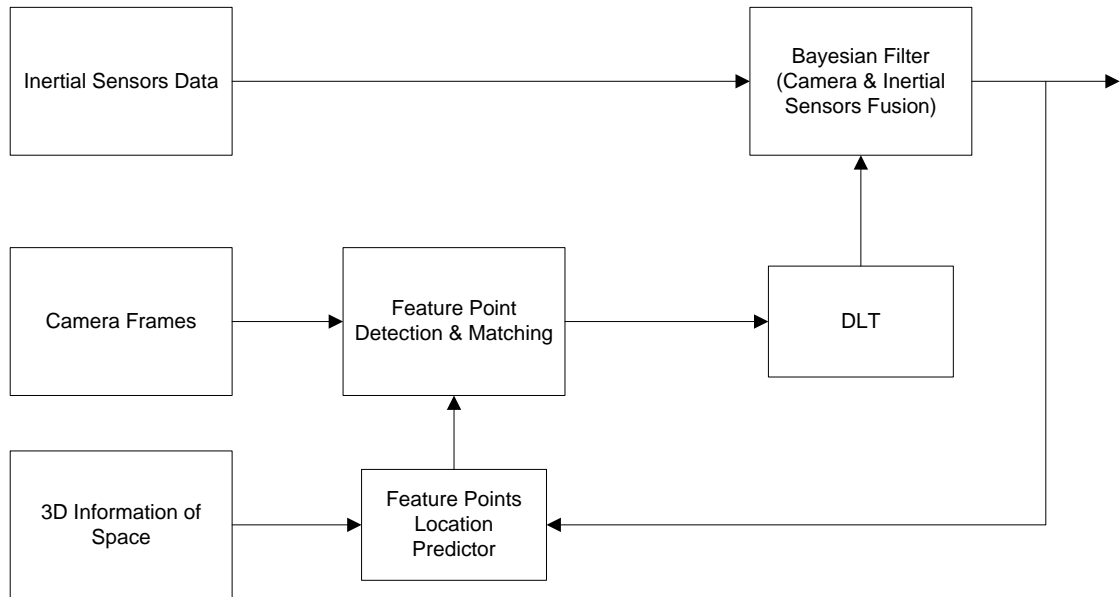
This thesis aims to develop high-accuracy 3D camera motion tracking algorithm using the video cameras and inertial sensors but not require placing any devices or points in the space. The method consists of a Bayesian filter that combines the feature points coming from video frames, the measurements coming from inertial sensors, and 3D information of the space. It is assumed that 3D information of the space is known in this thesis. In Figure 1.2.1, the general structure of the algorithm can be seen.

Figure 1.2: General structure of the algorithm



In addition to this algorithm, the hybrid algorithm shown in Figure 1.2.2 combines the Bayesian filtering algorithm and the structure from motion algorithm to track head motion. What makes this algorithm difficult is that reconstruction from video frames is impossible without the camera calibration parameter (Luong *et al.* 2001). The images of the calibration object or the lines perpendicular to each other in the place can be used to find the camera calibration parameter (Pentland 1987). Also, there are some auto-calibration methods that do not need any special process to find the camera calibration parameters (Fischler *et al.* 1981).

Figure 1.3: Hybrid algorithm



Bayesian filter used in this thesis is similar to SLAM (Simultaneous Localization and Mapping) which is a popular approach in computer vision and robotics. However, in this thesis, the maps already known in these algorithms consist of the 3D information of the feature points of the environment, so we have implemented only localization part of SLAM algorithms, not mapping part. The camera pose and 3D information of environment form the system state. Some statistics on the state are also shifted cyclically by a Bayesian filter, and updated by using the measurements of the camera, and the inertial sensors. At that point, an important observation is that the 3D information is connected to each other. With the help of this connection, the system state is converged to real 3D information of the environment, and the camera continues to track its pose precisely.

Systems using only video frames are successful in slow motion, but in fast motion they do not give accurate results. Accelerometers and gyroscope are appropriate for fast motion. They do not work well in slow motion because of the accumulation of the error and noise. In addition to video frames, the accelerometer and gyroscope measurement usage is proposed in Bayesian filter (Bleser *et al.* 2009; Armesto *et al.* 2007; Corke *et al.* 2007; Gemeiner *et al.* 2007; Kim *et al.* 2007; Schon *et al.* 2007). This approach is used in this thesis as well.

1.3 LITERATURE REVIEW

1.3.1 Tracking Methods Using GPS & GSM

Many methods have been proposed for 3D motion tracking in the literature. The first method is the tracking method using Global Positioning System (GPS), Global System for Mobile Communication (GSM) and Universal Mobile Telecommunications System (UMTS). Especially for outdoors, GPS system, one of the most appropriate methods, is used for the position tracking algorithm (Papagiannakis *et al.* 2008; Schmeil *et al.* 2006; Azuma *et al.* 2006). Since GPS needs signals coming from 4 different satellites, this method is not suitable for indoor applications. Schmeil (2006) used GPS system combining with angle sensors on the HMD in virtual assistant for outdoors. In this system, localization precision is related to satellite link and varies from 3m to 10m.

1.3.2 Tracking Methods Using IR & RFID

The second method is the tracking method using infrared (IR) light and radio frequency (RF). IR light-emitting LEDs emit light in a very narrow band. Thus, by the sensor, they can easily be detected using an appropriate filter. This method requires placing IR LEDs in the space (Papagiannakis *et al.* 2008). Similarly, there are other methods using active or passive radio wave transmitters (RFID). Steggle (2005) developed a system called UWB that provides tracking with 15cm accuracy using a network including small receivers and transmitters. However, according to (Azuma *et al.* 2001), it is not practical to place artificial cues on the scene, such as IR light emitters, RFID tags, markers, for some outdoor and mobile applications.

1.3.3 Tracking Methods Using Inertial Sensors

Because of the disadvantages of the IR light and RF tracking, the methods using the sensors placed only the HMD have been used in the literature. Gyroscope and accelerometer, such as inertial sensors, measure the derivatives of the motion and hence can be employed to boost tracking performance at high velocity and accelerations. Also,

they are not affected by the condition of current space. However, the algorithms using only head-mounted inertial sensors are similar to open-loop control systems (Azuma 1997). This is because there is no feedback on how well the real and virtual images coincide.

1.3.4 Tracking Methods Using Vision Data with Visual Markers

There are also tracking algorithms using visual markers placed in the space. In some AR applications (Cho *et al.* 1998), the marker placed on previously known locations are detected by using video processing algorithms and used in alignment process. In these algorithms, it is assumed that more than one feature points can be seen in all cases. If this assumption is provided, alignment can be done with one pixel accuracy (Azuma 1997).

1.3.5 Tracking Methods Using Vision Data without Visual Markers

Unlike the previous method, there are tracking algorithms not using visual markers placed in the space. These algorithms must be designed to overcome many difficulties, i.e. detecting feature points in the scene, variable lights, shadows, motion blur, overlaps, and real-time implementation condition, etc. Instead of calculating the camera pose relatively according to previous frame, estimation of the real pose prevents the error drift, and also allows the self-correction when the tracking is not working because of motion blur and overlaps (Papagiannakis et al. 2005).

Davison (2007) proposed a method determining the location of the camera in real time. The most important requirement of the method is the functionality in high frame rates (30fps). First of all, a pattern with known shape and location is shown to the camera to start tracking features and defining the certain depth measurement to system. Also, it assumed that the camera is moving with constant linear and angular velocity. The feature points (11x11) located in the room in which camera is moving are detected by the method implemented by Shi (1994). It is assumed that the feature points are on the plane and the vector is assigned to each of them. From a new perspective of the camera,

the projection of the feature points on the new perspective are found and used as templates. Before process of the cross-correlation between image from camera and feature points templates, the location of the template in the image is predicted. For this purpose, EKF method using motion model and modeling uncertainty with covariance matrix is used. Thereby, feature template is searched in the ellipsoidal area in the image. Advantage of this property is the real time functionality of the method. This technique performs satisfactorily at the slow motion, however it becomes less accurate at high velocities and accelerations due to motion blur.

1.3.6 Hybrid Tracking Methods

Using different tracking algorithms together can be useful. For example, combining video-based tracking with prediction-based tracking algorithms ensure that the system will continue to work when there is no sufficient and visible feature points (Azuma 1997). In these applications, the algorithms using compass and angular velocity compass on the HMD are combined with video processing algorithms (You *et al.* 1999). Also, the estimated values of perspective can be a good starting point for video-based tracking algorithms. Thus, delays in the system can be reduced (Azuma *et al.* 2001). It is important to use the right motion model for making an accurate prediction (Akatsuka & Bekey 1998). Sometimes a choice must be made between several different motion models (Chai 1999). Davison (2007) obtained a hybrid tracking algorithm by adding the data coming from accelerometer on the robot in the system.

Another hybrid method is the method where inertial sensors and cameras are used together (Azuma *et al.* 2001; Yokokohji *et al.* 2000). Techniques that use only camera measurements generally perform satisfactorily at slow head motion, however they become less accurate at high velocities and accelerations due to motion blur. Inertial sensors on the other hand measure the derivatives of the head pose and hence can be employed to boost the tracking performance at high velocities and accelerations. Therefore, hybrid methods where inertial sensors and cameras are used together have been proposed (Yokokohji *et al.* 2000; Newman *et al.* 2006; DiVerdi & Hollerer 2007). Yokokohji (2000) proposed a method combining video frames with accelerometers. In

this method, accelerometer data is used to predict head motion, thereby make the method using video frames more robust, and reduce delays in the system. It is indicated that this method work with average 6 pixels, maximum 11 pixels error when the head motion is fast (10 m/s^2 and 49 rad/s^2).

1.3.7 Map Estimation Methods

When the map is not known, structure from motion algorithms can be used to find the map of space. First of all, the motion of the camera capturing frames must be known to find the 3D structure. However, 3D structure must be known to find the motion of the camera. This is the main problem of the structure from motion algorithms. The other reason making the problem difficult is that the reconstruction from video frames is impossible without the camera calibration parameter (Luong & Faugeras 2001). The images of the calibration object or the lines perpendicular to each other in the place can be used to find the camera calibration parameter (Pentland 1987). Also, there are some auto-calibration methods that do not need any special process to find the camera calibration parameters (Fischler & Bolles 1981).

The most common and successful method for structure from motion is BA (Bundle Adjustment) algorithm (Chekhlov *et al.* 2006). (BA) is almost invariably used as the last step of every feature-based multiple view reconstruction vision algorithm to obtain optimal 3D structure and motion (i.e. camera matrix) parameter estimates. Provided with initial estimates, BA simultaneously refines motion and structure by minimizing the reprojection error between the observed and predicted image points. The minimization is typically carried out with the aid of the LM (Levenberg-Marquardt) algorithm (<http://www.ics.forth.gr/~lourakis/sba/> 2010).

Another mapping algorithm is SLAM in the literature. SLAM is a group of algorithms developed to make a robot (or apparatus on the head) estimate both the position of itself and the map of the unknown environment (Durant-Whyte *et al.* 2006). There are different advantages and disadvantages of the SLAM and BA algorithms. Due to the large number of unknowns contributing to the minimized reprojection error, a general purpose implementation of the LM algorithm incurs high computational and memory

costs. So, the real time implementation of the BA does not seem possible (Pentland 1987). However, SLAM has certain errors for representation of the larger frames because the dimension of EKF is growing with the actual time. FastSLAM that is developed to run EKF based SLAM methods faster has the same problem with EKF-SLAM (Konolige & Agrawal 2008).

FastSLAM algorithm (Konolige & Agrawal 2008) does not use all feature points in contrast to EKF-SLAM. In FastSLAM, fewer feature points representing the trajectory are used and passed into smaller EKF. There are two disadvantages of FastSLAM:

1. The number of the feature points to represent the given frame is not obvious.
2. The computation cost is growing with the number of feature points.

EKF-SLAM is more disadvantageous than BA in the calculation of the wide range and rough land area because of the reasons stated above. FrameSLAM method is developed to eliminate disadvantages of SLAM method recently (Konolige & Agrawal 2008). FrameSLAM can be used in wide range and rough area because it chooses only particular subspaces of the feature points and frames, and works with them only. A reduction in the number of frame affects the performance less than a reduction in the number of feature points.

PTAM (Parallel Tracking and Mapping) is developed recently to implement BA algorithm in real time (Klein & Murray 2007). PTAM implements tracking and mapping algorithms in parallel. Tracking used in PTAM starts with the estimation of the camera pose by using new frame in motion model. Then the camera pose is updated by using the matching of the projection of the feature points found before and placed on map on current frame and the feature points on current frame. First, the update is made by using clearer 50 feature points matching, and then more detailed update is made by using approximately 1000 feature points matching.

In PTAM algorithm, tracking and mapping are not connected to each other, so any robust tracking method desired can be used. Indeed, data does not need to be shared between tracking and mapping.

When BA algorithm is compared to EKF-SLAM algorithm to build a map, BA algorithm gives better results (Konolige and Agrawal 2008; Klein and Murray 2008; Klein and Murray 2007).

1.4 CONTRIBUTIONS OF THE THESIS

The main contributions of this thesis are:

1. The inertial sensors (accelerometer and gyroscope) data is used to increase the accuracy of the motion tracking. Using the inertial sensors data with visual data (camera video frame) in Bayesian filter overcomes the limitations of the only visual based motion tracking. The biases of the inertial sensors are also tracked in time and the effects of the biases are minimized.
2. There are a total of eight approaches for fusing gyroscope and accelerometer sensor data with camera measurements, i.e., both of them used as measurements, both of them used as control inputs, one is used as control input while the other is used as measurement, and one is used as control input or measurement while the other is not used. Only three of these eight cases have been investigated in the literature.
3. The Hybrid method that combines the recursive Bayesian filtering method and the direct linear transformation (DLT) method for tracking 3D camera motion is developed. In this method, we design new Bayesian filter which uses the head pose estimated from DLT instead of using feature points directly.

1.5 OUTLINE OF THE THESIS

To be able to keep track of this thesis report, this chapter contains a simple outline that gives information about the contents of each chapter.

In the second chapter, background information used in this thesis is given.

In the third chapter, the Extended Kalman Filter (EKF) and Hybrid filter equations are derived. All equations and their derivation are explained in detail in Appendix.

Chapter four contains the experimental setup and the simulation results.

Chapter five and six give a summary and discussion together with possible future research directions.

2. BACKGROUND

2.1 BAYESIAN FILTERING

Bayesian filter is a general probabilistic approach for estimating an unknown probability density function recursively over time using incoming measurements and a mathematical process model.

A Bayesian filter is an algorithm used in computer science for calculating the probabilities of multiple beliefs to allow a robot to infer its position and orientation. Essentially, Bayesian filters allow robots to continuously update their most likely position within a coordinate system, based on the most recently acquired sensor data. This is a recursive algorithm. It consists of two parts: prediction and innovation. If the variables are linear and normally distributed the Bayesian filter becomes equal to the Kalman filter.

Bayesian filtering is to update the system state over time recursively by using the measurements coming from inertial sensors or video frame. Theoretically, it is difficult to implement the optimal Bayesian equation in computer because it requires the integral representation of the multi-dimensional distributions (Fox *et al.* 2005). There are lots of different Bayesian filter variants of this approach in practice. Kalman filter and extended Kalman filter (EKF) are the first and most widely used of these filters for SLAM (Davison *et al.* 2007; Durrant-Whyte & Bailey 2006; Fox *et al.* 2005; Guivant & Nebot 2001). The SLAM algorithms using EKF are called as EKF-SLAM.

Bayesian filter used in this thesis is similar to EKF-SLAM (Simultaneous Localization and Mapping) which is a popular approach in computer vision and robotics. SLAM is a group of algorithms developed to make a robot (or apparatus on the head) estimate both the position of itself and the map of the unknown environment (Durrant-Whyte & Bailey 2006). The maps in these algorithms consist of the 3D information of the feature points of the space. The robot pose and 3D information of environment form the system state. Some statistics on the state also are shifted cyclically by a Bayesian filter, and

updated by using the measurements of the camera, and the inertial sensors. Here, an important observation is that the 3D information is connected to each other. Thanks to this connection, the system state is converged to real 3D information of the environment, and the robot continues to track its pose precisely (Durrant-Whyte & Bailey 2006).

2.1.1 Kalman Filter

The Kalman filter is a recursive estimator. This means that only the estimated state from the previous time step and the current measurement are needed to compute the estimate for the current state. In order to use the Kalman filter to estimate the internal state of a process given only a sequence of noisy observations, one must model the process in accordance with the framework of the Kalman filter. This means specifying the following matrices: F_t , the state-transition model; H_t , the observation model; Q_t , the covariance of the process noise; R_t , the covariance of the observation noise; and sometimes B_t , the control-input model, for each time-step, t , as described below:

$$x_t = F_t x_{t-T} + B_t u_t + v_t \quad (2.1)$$

where F_t is the state transition model which is applied to the previous state x_{t-T} , B_t is the control input model which is applied to the control vector u_t and v_t is the process noise which is assumed to be drawn from a zero mean multivariate normal distribution with covariance Q_t , $v_t \sim N(0, Q_t)$.

At time t a measurement y_t of the true state x_t is made according to:

$$y_t = H_t x_t + e_t \quad (2.2)$$

where H_t is the observation model which maps the true state space into the observed space and e_t is the observation noise which is assumed to be zero mean Gaussian white noise with covariance R_t , $e_t \sim N(0, R_t)$.

The state of the filter is represented by two variables:

$\hat{x}_{t|t}$: the a posteriori state estimate at time t given observations up to and including at time t;

$P_{t|t}$: the a posteriori error covariance matrix.

The Kalman filter can be written as a single equation; however it is most often conceptualized as two distinct phases: "Predict" and "Update". The predict phase uses the state estimate from the previous time step to produce an estimate of the state at the current time step. This predicted state estimate is also known as the a priori state estimate because, although it is an estimate of the state at the current time step, it does not include observation information from the current time step. In the update phase, the current a priori prediction is combined with current observation information to refine the state estimate. This improved estimate is called as posteriori state estimate.

Typically, the two phases alternate, with the prediction advancing the state until the next scheduled observation, and the update incorporating the observation.

Predict:

Predicted (a priori) state estimate:

$$\hat{x}_{t|t-T} = F_t \hat{x}_{t-T|t-T} + B_t u_t \quad (2.3)$$

Predicted (a priori) estimate covariance:

$$P_{t|t-T} = F_t P_{t-T|t-T} F_t^T + Q_t \quad (2.4a)$$

where

$$P_{t|t'} \triangleq E[(x_t - \hat{x}_{t|t'})(x_t - \hat{x}_{t|t'})] \quad (2.4b)$$

Update:

Innovation or measurement residual:

$$z_t = y_t - H_t \hat{x}_{t|t-T} \quad (2.5)$$

Innovation (or residual) covariance:

$$S_t = H_t P_{t|t-T} H_t^T + R_t \quad (2.6)$$

Optimal Kalman gain:

$$K_t = P_{t|t-T} H_t^T S_t^{-1} \quad (2.7)$$

Updated (a posteriori) state estimate:

$$\hat{x}_{t|t} = \hat{x}_{t|t-T} + K_t z_t \quad (2.8)$$

Updated (a posteriori) estimate covariance:

$$P_{t|t} = P_{t|t-T} - K_t H_t P_{t|t-T} \quad (2.9)$$

2.1.2 Extended Kalman Filter

The basic Kalman filter is limited to a linear assumption. More complex systems, however, can be nonlinear. The non-linearity can be associated either with the process model or with the observation model or with both. The EKF is the nonlinear version of the Kalman filter that linearizes about an estimate of the current mean and covariance. For EKF, the nonlinear state-space model is given:

$$x_t = f(x_{t-T}, u_t, v_t) \quad (2.10a)$$

$$y_t = h(x_t, e_t) \quad (2.10b)$$

where x_t denotes the state vector, u_t denotes a known control input, v_t and e_t are the process and measurement noises which are both assumed to be zero mean multivariate Gaussian noises with covariance Q_t and R_t , respectively.

The function f can be used to compute the predicted state from the previous estimate and similarly the function h can be used to compute the predicted measurement from the predicted state. However, f and h cannot be applied to the covariance directly. Instead a matrix of partial derivatives (the Jacobian) is computed.

At each time step the Jacobian is evaluated with current predicted states. These matrices can be used in the Kalman filter equations. This process essentially linearizes the non-linear function around the current estimate (Kleeman 1996).

Predict:

Predicted (a priori) state estimate:

$$\hat{x}_{t|t-T} = f(\hat{x}_{t-T|t-T}, u_t, 0) \quad (2.11)$$

Predicted (a priori) estimate covariance:

$$P_{t|t-T} = F_t P_{t-T|t-T} F_t^T + V_t Q_t V_t^T \quad (2.12a)$$

where

$$P_{t|t'} \triangleq E[(x_t - \hat{x}_{t|t'})(x_t - \hat{x}_{t|t'})] \quad (2.12b)$$

The state transition matrices are defined to be the following Jacobians:

$$F_t = \left. \frac{\partial f}{\partial x} \right|_{(\hat{x}_{t-T|t-T}, u_t, 0)} \quad (2.13a)$$

$$V_t = \left. \frac{\partial f}{\partial v} \right|_{(\hat{x}_{t-T|t-T}, u_t, 0)} \quad (2.13b)$$

Update:

Innovation or measurement residual:

$$z_t = y_t - h(\hat{x}_{t|t-T}, 0) \quad (2.14)$$

Innovation (or residual) covariance:

$$S_t = H_t P_{t|t-T} H_t^T + R_t \quad (2.15)$$

Optimal Kalman gain:

$$K_t = P_{t|t-T} H_t^T S_t^{-1} \quad (2.16)$$

Updated (a posteriori) state estimate:

$$\hat{x}_{t|t} = \hat{x}_{t|t-T} + K_t z_t \quad (2.17)$$

Updated (a posteriori) estimate covariance:

$$P_{t|t} = P_{t|t-T} - K_t H_t P_{t|t-T} \quad (2.18)$$

where the observation matrices are defined to be the following Jacobians:

$$H_t = \left. \frac{\partial h}{\partial x} \right|_{(\hat{x}_{t|t-T}, 0)} \quad (2.19)$$

EKF-SLAM algorithms are not robust for the error of the matching feature points, and run only the maps including small number of feature points since computation cost of algorithm is high (Jacobians) (Fox *et al.* 2005).

2.2 USAGE OF INERTIAL SENSORS IN BAYESIAN FILTER

Systems using only video frames are successful in slow motion, but in fast motion they do not give accurate results. Accelerometers and gyroscope are appropriate for fast motion. However; they do not work well in slow motion because of the accumulation of the error and noise. In SLAM, in addition to video frame, the accelerometer and gyroscope measurement usage is proposed in Bayesian filter (Bleser & Stricker 2009; Armesto *et al.* 2007; Corke *et al.* 2007; Gemeiner *et al.* 2007; Kim & Sukkarieh 2007; Schon *et al.* 2007). This approach is used in this thesis as well.

Bleser and Stricker (2009) proposed a head motion tracking algorithm that uses inertial sensors and video frames in Bayesian filter. They combine inertial sensors and video frames in four models:

1. Combining data from video frame and gyroscope.
2. Combining data from video frame, gyroscope, and accelerometer, but the accelerometer is used for stabilizing the camera attitude.
3. Combining data from video frame, gyroscope, and accelerometer, using all information given in the accelerometer measurement.
4. Combining data from video frame, gyroscope, and accelerometer, but accelerometer and gyroscope information used as not measurement, used as control input.

In their paper, they test all of the four models with slow and fast motion. Their experiments showed that in fast motion model 3 and 4 give better results. They stated that there is no significant difference in model 3 and 4 results, but because the accelerometer and gyroscope are used as control input in model 4, the computation cost is lower than model 3. They also track the biases of the gyroscope and accelerometer in addition to pose of head. The case that they track the biases gives better result than the case that they do not track the biases.

2.3 INERTIAL SENSORS

Inertial sensors measure the translational and rotational body kinematics. Inertial sensors are devices that use inertia to perform a measurement. As a practical matter, when people say “inertial sensor” they are referring to an accelerometer or a gyroscope. In this thesis, we are using two 2D gyroscopes, and one 3D accelerometer. By using inertial sensors, we get 3D linear acceleration, 3D angular velocity, and 3D earth gravitation force. We are using two 2D gyroscopes to get 3D angular velocity.

2.3.1 Gyroscope

A gyroscope is a device used primarily for navigation and measurement of angular velocity. Current gyroscopes can measure angular velocity in 1, 2, or 3 directions. 3-axis gyroscopes are often implemented with a 3-axis accelerometer to provide a full 6 degree-of-freedom (DoF) motion tracking system (SensorWiki.org 2009). The 3D gyroscope measures the angular velocity expressed as $\omega_{s,t}^{ws}$ in the sensor frame s . The direction of the angular velocity is from sensor coordinate system to global coordinate system (Bleser & Stricker 2009). In the next part, the difference of direction and the transformation of the coordinate systems can be found. When we stabilize the gyroscope, the gyroscope still gets some values. These values are called as bias term and it is not constant. The measured angular velocity from gyroscope includes the slowly varying bias $b_{s,t}^\omega$ and the zero mean white noise $e_{s,t}^\omega$. The calibrated gyroscope signal is:

$$y_{s,t}^\omega = \omega_{s,t}^{ws} + b_{s,t}^\omega + e_{s,t}^\omega \quad (2.20)$$

In this thesis, we thought that the bias term in this equation is varying and the bias term must be estimated at each time angular velocity is measured.

2.3.2 Accelerometer

An accelerometer is an electromechanical device that measures acceleration forces (SensorWiki.org 2009). These forces may be static, like the constant force of gravity pulling at your feet, or they could be dynamic - caused by moving or vibrating the accelerometer. Similarly with gyroscope, when we stabilize the accelerometer, the accelerometer still gets some values. These values are called as bias term and it is not constant. The calibrated accelerometer signal $y_{s,t}^a$ is corrupted by a slowly varying bias $b_{s,t}^a$ and zero mean white noise $e_{s,t}^a$:

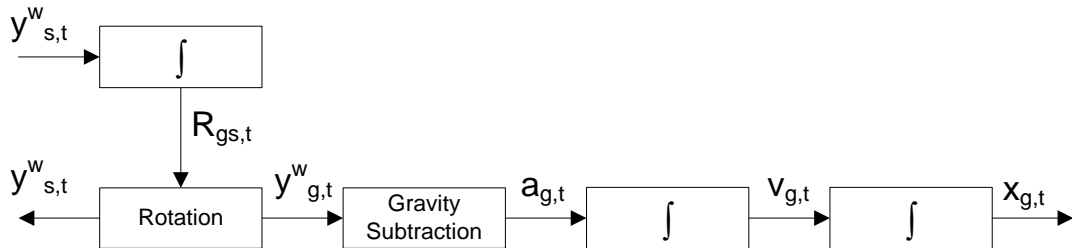
$$y_{s,t}^a = a_{s,t} - g_s + b_{s,t}^a + e_{s,t}^a \quad (2.21)$$

Because in the global coordinate system gravity force is constant, the accelerometer expression can be written in the global coordinate system:

$$y_{s,t}^a = R_{sg,t}(a_{g,t} - g_g) + b_{s,t}^a + e_{s,t}^a \quad (2.22)$$

where $R_{sg,t}$ denotes the rotation matrix from global coordinate system to sensor coordinate system. This rotation matrix can be found by using angular velocity data measured by gyroscope. Also, we can use the angular velocity data measured by gyroscope to calculate linear acceleration and the position of the system (Bleser & Stricker 2009).

Figure 2.1: The workflow of inertial navigation system



2.4 COMBINATION OF CAMERA AND INERTIAL SENSORS

Each inertial sensor and the camera have their own spatial and time domain. To combine the all camera and inertial sensor, we must synchronize them. In the time domain, the camera used in this thesis has 30 Hz frame rate. The camera is a multicolor camera and its frame rate can be changed. The inertial sensor frequency is faster than the camera. Their frequency can be up to 400 Hz. In our simulation, we used 120 Hz accelerometer and gyroscope. Our system works at 120 Hz, so the camera measurements are used once time in four periods.

When we look at the spatial domain of the camera and the inertial sensors, they are also different from each other. The relationship of the coordinate systems of the camera and the inertial system can be seen in Figure 2.2 (Bleser & Stricker 2009).

3D coordinate systems in Figure 2.4.1 are:

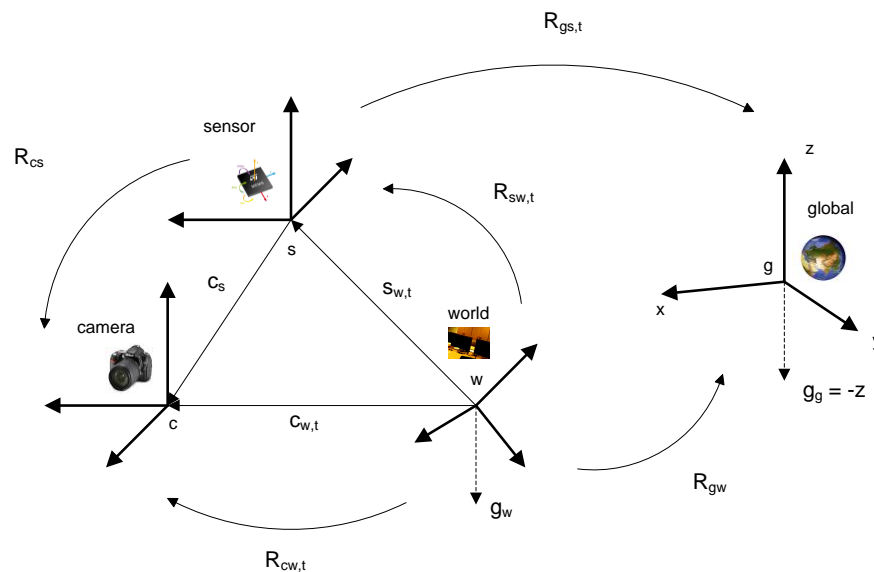
World (w): world frame is fixed to the target scene model

Camera (c): camera frame is fixed to moving camera

Sensor (s): sensor frame is fixed to moving sensors

Global (g): global frame is fixed to earth.

Figure 2.2: Camera-inertial sensor 3D coordinates and relations



In our system, the sensor coordinate system and the camera coordinate system are fixed and the relation of each coordinate system is constant. R_{cs} (rotation matrix from sensor to camera coordinate system), and c_s (distance between camera and sensor origins) are fixed and does not change in time. So, a point on camera coordinate can be easily transformed to sensor coordinate system. On the other hand, camera-sensor combination is moving in time, the coordinate systems of them change according to global coordinate system. For example, if you want to find the camera pose by using the sensor pose, we can convert sensor pose to camera pose by using:

$$R_{cw,t} = R_{cs}R_{sw,t} \quad (2.23)$$

$$c_{w,t} = s_{w,t} + R_{ws,t}c_s \quad (2.24)$$

A 3D point on world coordinate can be pointed on the camera coordinate by using Figure 2.3:

$$m_c = R_{cs}(R_{sw}(m_w - s_w) - c_s) \quad (2.25)$$

2.5 FEATURE POINTS DETECTION

When the camera frame is captured, the specific feature points must be detected and matched with the feature points on the previous frames. We need these feature points and their 2D/3D correspondence to use the camera measurements.

To find the feature points, SIFT (Scale-invariant feature transform) can be used. SIFT is an algorithm in computer vision to detect and describe local features in images (Lowe 2004).

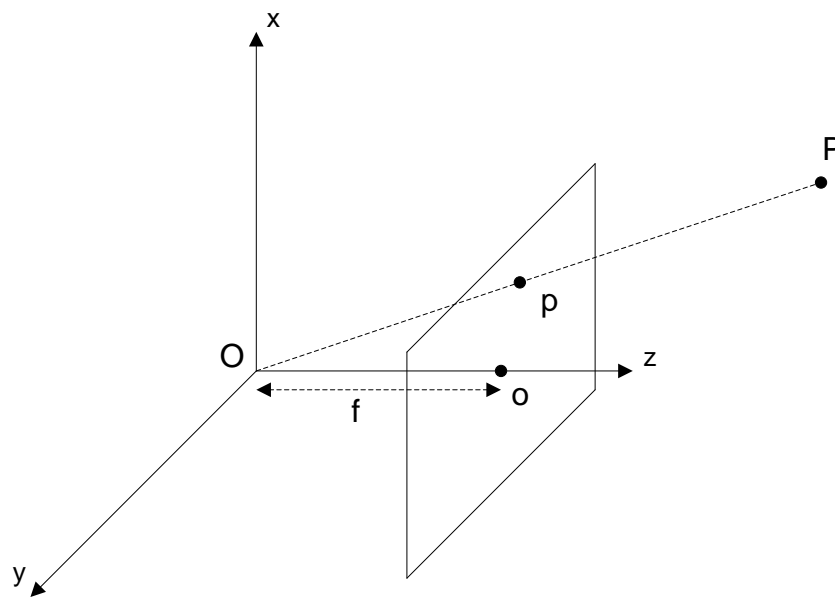
SIFT key points of objects are first extracted from a set of reference images (Lowe 2004) and stored in a database. An object is recognized in a new image by individually comparing each feature from the new image to this database and finding candidate matching features based on Euclidean distance of their feature vectors. From the full set of matches, subsets of key points that agree on the object and its location, scale, and

orientation in the new image are identified to filter out good matches. The determination of consistent clusters is performed rapidly by using an efficient hash table implementation of the generalized Hough transform. Each cluster of 3 or more features that agrees on an object and its pose is then subject to further detailed model verification and subsequently outliers are discarded. Finally, the probability that a particular set of features indicates the presence of an object is computed, given the accuracy of fit and number of probable false matches. Object matches that pass all these tests can be identified as correct with high confidence (Lowe 2004).

2.6 CAMERA GEOMETRY

Perspective projection can be defined as in Figure 2.3. The *center of projection* is at the origin O of the 3D reference frame of the space. The image plane is parallel to the (\vec{x}, \vec{y}) plane and displaced a distance f (*focal length*) along the \vec{z} axis from the origin. The 3D point P projects to the image point p . The orthogonal projection of O onto image plane is the *principal point* o , and the \vec{z} axis which corresponds to this projection line is the *principal axis*.

Figure 2.3: Standard perspective projection



Let (x_c, y_c) be the 2D coordinates of p and (X_c, Y_c, Z_c) the 3D coordinates of P . A direct application of Thales theorem shows that:

$$x_c = f \frac{X_c}{Z_c}, \quad y_c = f \frac{Y_c}{Z_c} \quad (2.26)$$

We can assume that $f = 1$ as different values of f just correspond to different scaling of the image. Below, we will incorporate a full camera calibration into the model. In homogeneous coordinates, the above equations become:

$$\begin{bmatrix} x_c \\ y_c \\ 1 \end{bmatrix} \sim \begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix} \quad (2.27)$$

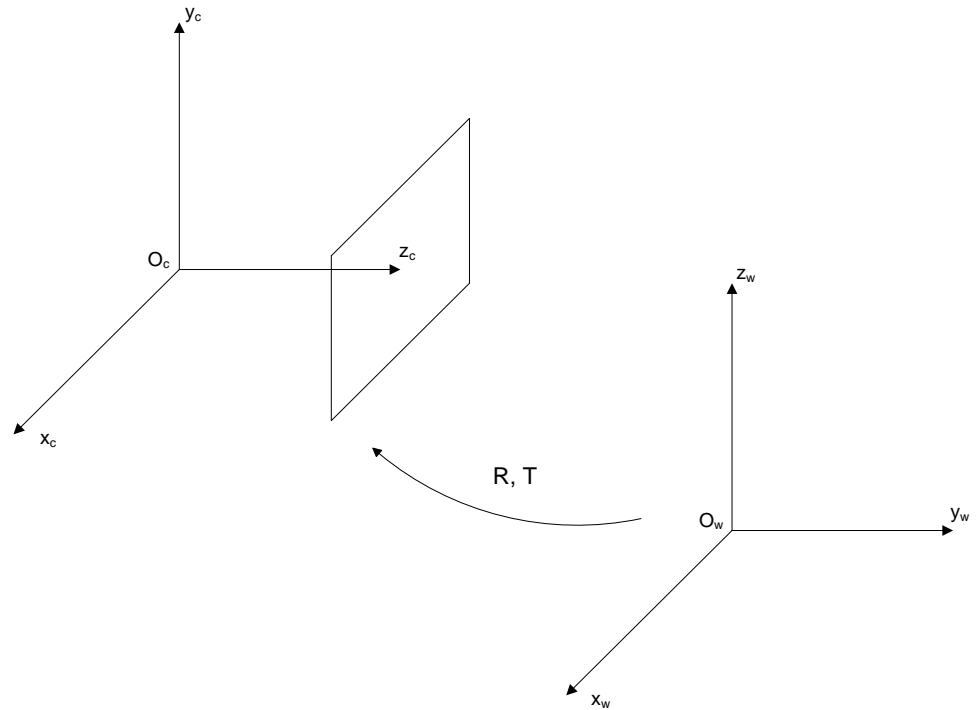
The world coordinate system does not usually coincide with the perspective reference frame, so the 3D coordinates undergo a Euclidean motion described by some matrix M . M gives the 3D position and pose of the camera and therefore has six degrees of freedom which represents the *exterior* (or *extrinsic*) camera parameters. In a minimal parameterization, M has the standard 6 degrees of freedom of a rigid motion. The Euclidean transformation between the camera and world coordinates is:

$$X_c = RX_w + T \quad (2.28)$$

$$\begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix} = \begin{bmatrix} R & T \\ 0^T & 1 \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} \quad (2.29)$$

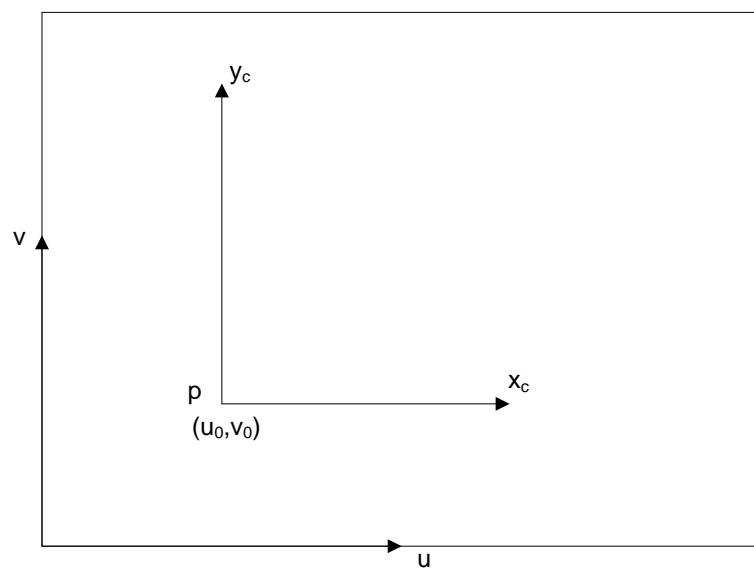
Transformation from world frame to camera frame can be seen in Figure 2.4:

Figure 2.4: Transformation from world coordinate to camera coordinate



In real images, the origin of the image coordinates is not the principal point and the scaling along each image axis is different, so the image coordinates undergo a further transformation described by some matrix K .

Figure 2.5: Camera calibration representation



$$k_u x_c = u - u_0 \quad (2.30)$$

$$k_v y_c = v - v_0 \quad (2.31)$$

where units of k are [pixels/length].

$$x_i = \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f k_u & 0 & u_0 \\ 0 & f k_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_c \\ y_c \\ f \end{bmatrix} = K \begin{bmatrix} x_c \\ y_c \\ f \end{bmatrix} \quad (2.32)$$

K is independent of the camera position. It contains the *interior* (or *intrinsic*) parameters of the camera. It is usually represented as an upper triangular matrix, called camera calibration matrix:

$$K = \begin{bmatrix} \alpha_u & 0 & u_0 \\ 0 & \alpha_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.33)$$

where $\alpha_u = f k_u$, $\alpha_v = f k_v$.

Finally, when we concatenating the three matrices:

$$x = \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = K \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} R & T \\ 0^T & 1 \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} = K[R|T] \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} \quad (2.34)$$

which defines the 3x4 projection matrix from Euclidean 3-space to an image:

$$x = P \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \quad (2.35)$$

$$P = K[R|T] \quad (2.36)$$

2.7 DIRECT LINEAR TRANSFORMATION

The equations of the direct linear transformation (DLT) are given below. The matrix given below consists of the known 3D position of the feature points in world coordinate system (X, Y, Z) and the 2D position of the feature points on camera image (x, y) .

$$A = \begin{bmatrix} X_1 & Y_1 & Z_1 & 1 & 0 & 0 & 0 & 0 & -x_1X_1 & -x_1Y_1 & -x_1Z_1 \\ 0 & 0 & 0 & 0 & X_1 & Y_1 & Z_1 & 1 & -y_1X_1 & -y_1Y_1 & -y_1Z_1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ X_N & Y_N & Z_N & 1 & 0 & 0 & 0 & 0 & -x_NX_N & -x_NY_N & -x_NZ_N \\ 0 & 0 & 0 & 0 & X_N & Y_N & Z_N & 1 & -y_NX_N & -y_NY_N & -y_NZ_N \end{bmatrix}, N \geq 6 \quad (2.37)$$

P matrix given below is the camera projection matrix. In our simulation, camera matrix is assumed to be unit matrix, so the left 3x3 part of the P matrix gives the rotation matrix of the camera (Q). The right 3x1 vector part of the P matrix gives the 3D position vector.

$$k = [x_1 \quad y_1 \quad \cdots \quad x_{12} \quad y_{12}]^T \quad (2.38a)$$

$$P = \begin{bmatrix} P_{11} & P_{12} & P_{13} & P_{14} \\ P_{21} & P_{22} & P_{23} & P_{24} \\ P_{31} & P_{32} & P_{33} & 1 \end{bmatrix} \Rightarrow p = \begin{bmatrix} P_{11} \\ P_{12} \\ P_{13} \\ P_{14} \\ P_{21} \\ P_{22} \\ P_{23} \\ P_{24} \\ P_{31} \\ P_{32} \\ P_{33} \end{bmatrix} \quad (2.38b)$$

$$k = A * p \quad (2.38c)$$

$$p = \text{pinv}(A) * k \quad (2.38d)$$

$$Q(q_{sw}) = \begin{bmatrix} P_{11} & P_{12} & P_{13} \\ P_{21} & P_{22} & P_{23} \\ P_{31} & P_{32} & P_{33} \end{bmatrix} \quad (2.38e)$$

$$\begin{bmatrix} s_{wx} \\ s_{wy} \\ s_{wz} \end{bmatrix} = -inv(Q(q_{sw})) * \begin{bmatrix} P_{14} \\ P_{24} \\ 1 \end{bmatrix} \quad (2.38f)$$

$pinv$ represents Moore-Penrose pseudoinverse matrix. A is not square, then $inv(A)$ does not exist. In this case, $pinv(A)$ has some of the properties of $inv(A)$.

The 3D position and 3D angle data obtained from above equations are used in the measurement model of the Hybrid filter.

$$y_{s,t}^q = q_{sw,t} + e_{s,t}^q \quad (2.39a)$$

$$y_{s,t}^s = s_{w,t} + e_{s,t}^s \quad (2.39b)$$

3. BAYESIAN FILTER DESIGN

In this section, two types of Bayesian filter are examined:

1. Extended Kalman Filter (EKF)
2. Hybrid Filter

3.1 EKF DESIGN

9 different EKF methods using video frames, accelerometer and gyroscope sensors are analyzed. Video frames are used in all models. In addition to video frames, angular velocity and linear acceleration data are used as measurement or control input in EKF. If any data is used as control input, it will be used in prediction part of EKF as a known control input. It will improve the prediction performance of EKF. On the other hand, if any data used as measurement, it will be used in correction part of EKF as a measurement. It will improve the correction performance of EKF. The methods investigated in this thesis are:

1. Both angular velocity and linear acceleration data used as "measurement"
2. Angular velocity data used as "control input", linear acceleration data used as "measurement"
3. Angular velocity data used as "measurement", linear acceleration data used as "control input"
4. Both angular velocity and linear acceleration data used as "control input"
5. Angular velocity data used as "measurement"
6. Angular velocity data used as "control input"
7. Linear acceleration data used as "measurement"
8. Linear acceleration data used as "control input"
9. Both angular velocity and linear acceleration data "not" used

For each case, EKF equations will be updated according to the usage of inertial sensors. General EKF equations are given in Section 2.1. Some of them are same for all cases; whereas some are different for each case.

3.1.1 Both Angular Velocity and Linear Acceleration Data Used As Measurement

In this method; all video frames, angular velocity, and linear acceleration data are used as measurement. The state vector \mathbf{x}_t comprises:

$$\mathbf{x}^T = [s_w^T \quad v_w^T \quad a_w^T \quad b_s^a{}^T \quad q_{sw}^T \quad \omega_s^T \quad b_s^\omega{}^T] \quad (3.1)$$

where s_w denotes the position of camera, v_w the linear velocity of camera, a_w the linear acceleration of camera, b_s^a the biases of accelerometer inertial sensor, q_{sw} the orientation quaternion, ω_s the angular velocity of camera, b_s^ω the biases of gyroscope inertial sensor.

The time model assumes constant linear acceleration and constant angular velocity. Time model is (Bleser & Stricker 2009):

$$\mathbf{x}_t = \begin{bmatrix} s_{w,t} \\ v_{w,t} \\ a_{w,t} \\ b_{s,t}^a \\ q_{sw,t} \\ \omega_{s,t} \\ b_{s,t}^\omega \end{bmatrix} = \begin{bmatrix} s_{w,t-T} + T v_{w,t-T} + \frac{T^2}{2} a_{w,t-T} + \frac{T^2}{2} \varepsilon_{w,t}^a \\ v_{w,t-T} + T a_{w,t-T} + T \varepsilon_{w,t}^a \\ a_{w,t-T} + \varepsilon_{w,t}^a \\ b_{s,t-T}^a + \varepsilon_{s,t}^{b^a} \\ \exp\left(-\frac{T}{2}(\omega_{s,t-T} + \varepsilon_{s,t}^\omega)\right) \odot q_{sw,t-T} \\ \omega_{s,t-T} + \varepsilon_{s,t}^\omega \\ b_{s,t-T}^\omega + \varepsilon_{s,t}^{b^\omega} \end{bmatrix} \quad (3.2a)$$

$$\mathbf{u}_t^T = 0 \quad (3.2b)$$

$$\mathbf{v}_t^T = [\varepsilon_{w,t}^a \quad \varepsilon_{s,t}^\omega \quad \varepsilon_{s,t}^{b^a} \quad \varepsilon_{s,t}^{b^\omega}] \quad (3.2c)$$

where $\varepsilon_{w,t}^a$, $\varepsilon_{s,t}^{b^a}$, $\varepsilon_{s,t}^\omega$, and $\varepsilon_{s,t}^{b^\omega}$ denote time independent Gaussian process noise that is uncorrelated in all components.

Since all data is used as measurement, all of them are in the measurement model. The measurement model below includes the linear acceleration, angular velocity, and the position of the feature points on the video frames, respectively:

$$y_t = \begin{bmatrix} y_{s,t}^a \\ y_{s,t}^\omega \\ m_t \end{bmatrix} \quad (3.3a)$$

$$e_t^T = [e_{s,t}^a \quad e_{s,t}^\omega \quad e_t^c] \quad (3.3b)$$

where $y_{s,t}^a$ denotes the linear acceleration, $y_{s,t}^\omega$ denotes the angular velocity, $e_{s,t}^a$, $e_{s,t}^\omega$, $e_{x,t}^c$, and $e_{y,t}^c$ denote time independent Gaussian measurement noise. Because the linear acceleration and gravity acceleration are in world coordinates in time model, they must be converted to inertial sensor coordinate by using $Q_{sw,t}$.

The other equations of EKF used in this method are provided below. Time model equations are:

$$\hat{x}_{t|t-T} = f(\hat{x}_{t-T|t-T}, 0,0,0,0) \quad (3.4a)$$

$$P_{t|t-T} = F_t P_{t-T|t-T} F_t^T + V_t Q_t V_t^T \quad (3.4b)$$

Measurement model equations are:

$$z_t = y_t - h(\hat{x}_{t|t-T}, 0,0,0) \quad (3.5)$$

Since time model and measurement model are both nonlinear, these models must be linearized in order to use EKF. The linearization of the time model equations are:

$$F_t = \frac{\partial f}{\partial x} \Big|_{(\hat{x}_{t-T|t-T}, 0, 0, 0, 0)} \quad (3.6a)$$

$$V_t = \frac{\partial f}{\partial v} \Big|_{(\hat{x}_{t-T|t-T}, 0, 0, 0, 0)} \quad (3.6b)$$

$$Q_t = \text{diag}([\Sigma_{\varepsilon_w, t}^a \quad \Sigma_{\varepsilon_s, t}^\omega \quad \Sigma_{\varepsilon_s, t}^{b^a} \quad \Sigma_{\varepsilon_s, t}^{b^\omega}]) \quad (3.6c)$$

The linearization of the measurement model equations are:

$$H_t = \frac{\partial h}{\partial x} \Big|_{(\hat{x}_{t|t-T}, 0, 0, 0)} \quad (3.7a)$$

$$R_t = \text{diag} \left(\left[\sigma_{e_{sx, t}}^2 \quad \sigma_{e_{sy, t}}^2 \quad \sigma_{e_{sz, t}}^2 \quad \sigma_{e_{sx, t}}^2 \quad \sigma_{e_{sy, t}}^2 \quad \sigma_{e_{sz, t}}^2 \quad \sigma_{e_{x, t}}^2 \quad \sigma_{e_{y, t}}^2 \right] \right) \quad (3.7b)$$

3.1.2 Angular Velocity Data Used As Control Input, Linear Acceleration Data Used As Measurement

In this method; video frames, and linear acceleration data are used as measurement. However, angular velocity data is used as control input. Thus, the state vector \mathbf{x}_t does not include angular velocity:

$$\mathbf{x}^T = [s_w^T \quad v_w^T \quad a_w^T \quad b_s^{aT} \quad q_{sw}^T \quad b_s^{\omega T}] \quad (3.8)$$

The angular velocity data is still used, so the biases of the gyroscope are still in the state vector \mathbf{x}^T .

As in the previous method, the time model assumes constant linear acceleration and constant angular velocity. The angular velocity is used as control input in this method, so the angular velocity is not used in state vector, but it is used in time model. Time model is:

$$x_t = \begin{bmatrix} s_{w,t} \\ v_{w,t} \\ a_{w,t} \\ b_{s,t}^a \\ q_{sw,t} \\ b_{s,t}^\omega \end{bmatrix} = \begin{bmatrix} s_{w,t-T} + T v_{w,t-T} + \frac{T^2}{2} a_{w,t-T} + \frac{T^2}{2} \varepsilon_{w,t}^a \\ v_{w,t-T} + T a_{w,t-T} + T \varepsilon_{w,t}^a \\ a_{w,t-T} + \varepsilon_{w,t}^a \\ b_{s,t-T}^a + \varepsilon_{s,t}^{b^a} \\ \exp\left(-\frac{T}{2}\left((y_{s,t}^\omega - b_{s,t-T}^\omega) + \varepsilon_{s,t}^\omega\right)\right) \odot q_{sw,t-T} \\ b_{s,t-T}^\omega + \varepsilon_{s,t}^{b^\omega} \end{bmatrix} \quad (3.9a)$$

$$u_t^T = y_{s,t}^\omega \quad (3.9b)$$

$$v_t^T = [\varepsilon_{w,t}^a \quad \varepsilon_{s,t}^\omega \quad \varepsilon_{s,t}^{b^a} \quad \varepsilon_{s,t}^{b^\omega}] \quad (3.9c)$$

In the measurement model, there is only the data used as measurement. Because of that angular velocity data is not used in measurement model. The measurement model includes the linear acceleration, and the position of the feature points on the video frames, respectively:

$$y_t = \begin{bmatrix} y_{s,t}^a \\ m_t \end{bmatrix} \quad (3.10a)$$

$$e_t^T = [e_{s,t}^a \quad e_t^c] \quad (3.10b)$$

The other equations of EKF used in this method are given below. Time model equations are:

$$\hat{x}_{t|t-T} = f(\hat{x}_{t-T|t-T}, y_{s,t}^\omega, 0, 0, 0, 0) \quad (3.11a)$$

$$P_{t|t-T} = F_t P_{t-T|t-T} F_t^T + V_t Q_t V_t^T + V_m Q_m V_m^T \quad (3.11b)$$

Measurement model equations are:

$$z_t = y_t - h(\hat{x}_{t|t-T}, 0, 0) \quad (3.12)$$

The linearization of the time model equations are:

$$F_t = \left. \frac{\partial f}{\partial x} \right|_{(\hat{x}_{t-T|t-T}, 0, 0, 0, 0)} \quad (3.13a)$$

$$V_t = \left. \frac{\partial f}{\partial v} \right|_{(\hat{x}_{t-T|t-T}, \mathcal{Y}_{s,t}^\omega, 0, 0, 0, 0)} \quad (3.13b)$$

$$Q_t = \text{diag}([\Sigma_{e_w,t}^a \quad \Sigma_{e_s,t}^\omega \quad \Sigma_{e_s,t}^{b^a} \quad \Sigma_{e_s,t}^{b^\omega}]) \quad (3.13c)$$

$$V_m = \left. \frac{\partial f}{\partial \mathcal{Y}_{s,t}^\omega} \right|_{(\hat{x}_{t-T|t-T}, \mathcal{Y}_{s,t}^\omega, 0, 0, 0, 0)} \quad (3.13d)$$

$$Q_m = \sigma_{e_{s,t}^\omega}^2 \quad (3.13e)$$

The linearization of the measurement model equations are:

$$H_t = \left. \frac{\partial h}{\partial x} \right|_{(\hat{x}_{t|t-T}, 0, 0)} \quad (3.14a)$$

$$R_t = \text{diag}([\sigma_{e_{sx,t}}^2 \quad \sigma_{e_{sy,t}}^2 \quad \sigma_{e_{sz,t}}^2 \quad \sigma_{e_{x,t}}^c \quad \sigma_{e_{y,t}}^c]) \quad (3.14b)$$

3.1.3 Angular Velocity Data Used As Measurement, Linear Acceleration Data Used As Control Input

In this method; video frames, and angular velocity data are used as measurement. However, linear acceleration data is used as control input. Since linear acceleration data is used as control input, the state vector \mathbf{x}_t does not include linear acceleration:

$$\mathbf{x}^T = [s_w^T \quad v_w^T \quad b_s^a{}^T \quad q_{sw}^T \quad \omega_s^T \quad b_s^{\omega T}] \quad (3.15)$$

The linear acceleration data is still used, so the biases of the accelerometer are still in the state vector \mathbf{x}^T .

As in the previous methods, the time model assumes constant linear acceleration and constant angular velocity. The linear acceleration is used as control input in this method,

so the linear acceleration is not used in state vector, but it is used in time model. Time model is:

$$x_t = \begin{bmatrix} s_{w,t} \\ v_{w,t} \\ b_{s,t}^a \\ q_{sw,t} \\ \omega_{s,t} \\ b_{s,t}^\omega \end{bmatrix} = \begin{bmatrix} s_{w,t-T} + Tv_{w,t-T} + \frac{T^2}{2}\varepsilon_{w,t}^a + \frac{T^2}{2}Q_{ws,t-T}(y_{s,t}^a - b_{s,t-T}^a) + \frac{T^2}{2}g_w \\ v_{w,t-T} + T\varepsilon_{w,t}^a + TQ_{ws,t-T}(y_{s,t}^a - b_{s,t-T}^a) + Tg_w \\ b_{s,t-T}^a + \varepsilon_{s,t}^{b^a} \\ \exp\left(-\frac{T}{2}(\omega_{s,t-T} + \varepsilon_{s,t}^\omega)\right) \odot q_{sw,t-T} \\ \omega_{s,t-T} + \varepsilon_{s,t}^\omega \\ b_{s,t-T}^\omega + \varepsilon_{s,t}^{b^\omega} \end{bmatrix} \quad (3.16a)$$

$$u_t^T = y_{s,t}^a \quad (3.16b)$$

$$v_t^T = [\varepsilon_{w,t}^a \quad \varepsilon_{s,t}^\omega \quad \varepsilon_{s,t}^{b^a} \quad \varepsilon_{s,t}^{b^\omega}] \quad (3.16c)$$

Linear acceleration data is not included in measurement model since measurement model only consist data used as measurement. The measurement model includes the angular velocity, and the position of the feature points on the video frames, respectively:

$$y_t = \begin{bmatrix} y_{s,t}^\omega \\ m_t \end{bmatrix} \quad (3.17a)$$

$$e_t^T = [e_{s,t}^\omega \quad e_t^c] \quad (3.17b)$$

The other equations of EKF used in this method are given below. Time model equations are:

$$\hat{x}_{t|t-T} = f(\hat{x}_{t-T|t-T}, y_{s,t}^a, 0,0,0,0) \quad (3.18a)$$

$$P_{t|t-T} = F_t P_{t-T|t-T} F_t^T + V_t Q_t V_t^T + V_m Q_m V_m^T \quad (3.18b)$$

Measurement model equations are:

$$z_t = y_t - h(\hat{x}_{t|t-T}, 0,0) \quad (3.19)$$

The linearization of the time model equations are:

$$F_t = \frac{\partial f}{\partial x} \Big|_{(\hat{x}_{t-T|t-T}, y_{s,t}^a, 0, 0, 0, 0)} \quad (3.20a)$$

$$V_t = \frac{\partial f}{\partial v} \Big|_{(\hat{x}_{t-T|t-T}, y_{s,t}^a, 0, 0, 0, 0)} \quad (3.20b)$$

$$Q_t = \text{diag}([\Sigma_{\varepsilon_w^a, t} \quad \Sigma_{\varepsilon_s^\omega, t} \quad \Sigma_{\varepsilon_s^{b^a}, t} \quad \Sigma_{\varepsilon_s^{b^\omega}, t}]) \quad (3.20c)$$

$$V_m = \frac{\partial f}{\partial y_{s,t}^a} \Big|_{(\hat{x}_{t-T|t-T}, y_{s,t}^a, 0, 0, 0, 0)} \quad (3.20d)$$

$$Q_m = \sigma_{e_{s,t}^a}^2 \quad (3.20e)$$

The linearization of the measurement model equations are:

$$H_t = \frac{\partial h}{\partial x} \Big|_{(\hat{x}_{t|t-T}, 0, 0)} \quad (3.21a)$$

$$R_t = \text{diag}([\sigma_{e_{sx,t}^\omega}^2 \quad \sigma_{e_{sy,t}^\omega}^2 \quad \sigma_{e_{sz,t}^\omega}^2 \quad \sigma_{e_{cx,t}^\omega}^2 \quad \sigma_{e_{cy,t}^\omega}^2]) \quad (3.21b)$$

3.1.4 Both Angular Velocity and Linear Acceleration Data Used As Control Input

In this method; only video frames are used as measurement. However, linear acceleration and angular velocity data are used as control input. Since linear acceleration and angular velocity data are used as control input, the state vector \mathbf{x}_t does not include linear acceleration and angular velocity:

$$\mathbf{x}^T = [s_w^T \quad v_w^T \quad b_s^{aT} \quad q_{sw}^T \quad b_s^{\omega T}] \quad (3.22)$$

The linear acceleration and angular velocity data are still used, so the biases of the accelerometer and gyroscope are still in the state vector \mathbf{x}^T .

As in the previous methods, the time model assumes constant linear acceleration and constant angular velocity. The linear acceleration and angular velocity are used as control input in this method, so the linear acceleration and angular velocity are not used in state vector, but they are used in time model. Time model is:

$$x_t = \begin{bmatrix} s_{w,t} \\ v_{w,t} \\ b_{s,t}^a \\ q_{sw,t} \\ b_{s,t}^\omega \end{bmatrix} = \begin{bmatrix} s_{w,t-T} + Tv_{w,t-T} + \frac{T^2}{2}\varepsilon_{w,t}^a + \frac{T^2}{2}Q_{ws,t-T}(y_{s,t}^a - b_{s,t-T}^a) - \frac{T^2}{2}g_w \\ v_{w,t-T} + T\varepsilon_{w,t}^a + TQ_{ws,t-T}(y_{s,t}^a - b_{s,t-T}^a) - Tg_w \\ b_{s,t-T}^a + \varepsilon_{s,t}^{b^a} \\ \exp\left(-\frac{T}{2}\left((y_{s,t}^\omega - b_{s,t-T}^\omega) + \varepsilon_{s,t}^\omega\right)\right) \odot q_{sw,t-T} \\ b_{s,t-T}^\omega + \varepsilon_{s,t}^{b^\omega} \end{bmatrix} \quad (3.23a)$$

$$u_t^T = [y_{s,t}^a \quad y_{s,t}^\omega] \quad (3.23b)$$

$$v_t^T = [\varepsilon_{w,t}^a \quad \varepsilon_{s,t}^\omega \quad \varepsilon_{s,t}^{b^a} \quad \varepsilon_{s,t}^{b^\omega}] \quad (3.23c)$$

The linear acceleration and angular velocity data are not used in measurement model. The measurement model includes only the position of the feature points on the video frames:

$$y_t = m_t \quad (3.24a)$$

$$e_t^T = e_t^c \quad (3.24b)$$

The other equations of EKF used in this method are given below. Time model equations are:

$$\hat{x}_{t|t-T} = f(\hat{x}_{t-T|t-T}, y_{s,t}^a, y_{s,t}^\omega, 0,0,0,0) \quad (3.25a)$$

$$P_{t|t-T} = F_t P_{t-T|t-T} F_t^T + V_t Q_t V_t^T + V_m Q_m V_m^T \quad (3.25b)$$

Measurement model equations are:

$$z_t = y_t - h(\hat{x}_{t|t-T}, 0) \quad (3.26)$$

The linearization of the time model equations are:

$$F_t = \frac{\partial f}{\partial x} \Big|_{(\hat{x}_{t-T|t-T}, y_{s,t}^a, y_{s,t}^\omega, 0, 0, 0, 0)} \quad (3.27a)$$

$$V_t = \frac{\partial f}{\partial v} \Big|_{(\hat{x}_{t-T|t-T}, y_{s,t}^a, y_{s,t}^\omega, 0, 0, 0, 0)} \quad (3.27b)$$

$$Q_t = \text{diag}([\Sigma_{\varepsilon_w,t}^a \quad \Sigma_{\varepsilon_s,t}^\omega \quad \Sigma_{\varepsilon_s,t}^{b^a} \quad \Sigma_{\varepsilon_s,t}^{b^\omega}]) \quad (3.27c)$$

$$V_m = \frac{\partial f}{\partial y_{s,t}} \Big|_{(\hat{x}_{t-T|t-T}, y_{s,t}^a, y_{s,t}^\omega, 0, 0, 0, 0)} \quad (3.27d)$$

$$Q_m = \text{diag}([\sigma_{e_{s,t}^a}^2 \quad \sigma_{e_{s,t}^\omega}^2]) \quad (3.27e)$$

The linearization of the measurement model equations are:

$$H_t = \frac{\partial h}{\partial x} \Big|_{(\hat{x}_{t|t-T}, 0)} \quad (3.28a)$$

$$R_t = \text{diag}([\sigma_{e_{x,t}^c}^2 \quad \sigma_{e_{y,t}^c}^2]) \quad (3.28b)$$

3.1.5 Angular Velocity Data Used As Measurement

In this method; the video frames and angular velocity data are used as measurement. However, linear acceleration data is not used in this method, so the state vector \mathbf{x}_t does not include linear acceleration:

$$\mathbf{x}^T = [s_w^T \quad v_w^T \quad q_{sw}^T \quad \omega_s^T \quad b_s^{\omega T}] \quad (3.29)$$

The linear acceleration data are not used, so the biases of the accelerometer are not in the state vector \mathbf{x}^T .

Unlike previous methods, the time model assumes constant linear velocity and constant angular velocity because there is no linear acceleration data. The linear acceleration is

not used in this method, so the linear acceleration is not used in time model like state vector. Time model is:

$$x_t = \begin{bmatrix} s_{w,t} \\ v_{w,t} \\ q_{sw,t} \\ \omega_{s,t} \\ b_{s,t}^\omega \end{bmatrix} = \begin{bmatrix} s_{w,t-T} + T v_{w,t-T} + T \varepsilon_{w,t}^v \\ v_{w,t-T} + \varepsilon_{w,t}^v \\ \exp\left(-\frac{T}{2}(\omega_{s,t-T} + \varepsilon_{s,t}^\omega)\right) \odot q_{sw,t-T} \\ \omega_{s,t-T} + \varepsilon_{s,t}^\omega \\ b_{s,t-T}^\omega + \varepsilon_{s,t}^{b^\omega} \end{bmatrix} \quad (3.30a)$$

$$u_t^T = 0 \quad (3.30b)$$

$$v_t^T = [\varepsilon_{w,t}^v \quad \varepsilon_{s,t}^\omega \quad \varepsilon_{s,t}^{b^\omega}] \quad (3.30c)$$

where $\varepsilon_{w,t}^v$ denote time independent Gaussian process noise.

The measurement model includes the angular velocity, and the position of the feature points on the video frames, respectively:

$$y_t = \begin{bmatrix} y_{s,t}^\omega \\ m_t \end{bmatrix} \quad (3.31a)$$

$$e_t^T = [e_{s,t}^\omega \quad e_t^c] \quad (3.31b)$$

The other equations of EKF used in this method are provided below. Time model equations are:

$$\hat{x}_{t|t-T} = f(\hat{x}_{t-T|t-T}, 0, 0, 0) \quad (3.32a)$$

$$P_{t|t-T} = F_t P_{t-T|t-T} F_t^T + V_t Q_t V_t^T \quad (3.32b)$$

Measurement model equations are:

$$z_t = y_t - h(\hat{x}_{t|t-T}, 0, 0) \quad (3.33)$$

The linearization of the time model equations are:

$$F_t = \left. \frac{\partial f}{\partial x} \right|_{(\hat{x}_{t-T}|_{t-T}, 0, 0, 0)} \quad (3.34a)$$

$$V_t = \left. \frac{\partial f}{\partial v} \right|_{(\hat{x}_{t-T}|_{t-T}, 0, 0, 0)} \quad (3.34b)$$

$$Q_t = \text{diag}([\Sigma_{\varepsilon_w^v, t} \quad \Sigma_{\varepsilon_s^\omega, t} \quad \Sigma_{\varepsilon_s^{b\omega}, t}]) \quad (3.34c)$$

The linearization of the measurement model equations are:

$$H_t = \left. \frac{\partial h}{\partial x} \right|_{(\hat{x}_t|_{t-T}, 0, 0)} \quad (3.35a)$$

$$R_t = \text{diag}([\sigma_{e_{sx,t}^\omega}^2 \quad \sigma_{e_{sy,t}^\omega}^2 \quad \sigma_{e_{sz,t}^\omega}^2 \quad \sigma_{e_{x,t}^c}^2 \quad \sigma_{e_{y,t}^c}^2]) \quad (3.35b)$$

3.1.6 Angular Velocity Data Used As Control Input

In this method; only the video frames are used as measurement. However, linear acceleration data is not used, but angular velocity data is used as control input in this method, so the state vector \mathbf{x}_t does not include linear acceleration and angular velocity:

$$\mathbf{x}^T = [s_w^T \quad v_w^T \quad q_{sw}^T \quad b_s^{\omega T}] \quad (3.36)$$

The angular velocity data is still used, so the biases of the gyroscope are still in the state vector \mathbf{x}^T .

Like the method in Section 3.1.5, the time model assumes constant linear velocity and constant angular velocity because there is no linear acceleration data. The linear acceleration is not used in this method, so the linear acceleration is not used in time model like state vector. On the other hand, because angular velocity data is used as control input, it is not in the state vector but it is in the time model. Time model is:

$$x_t = \begin{bmatrix} s_{w,t} \\ v_{w,t} \\ q_{sw,t} \\ b_{s,t}^\omega \end{bmatrix} = \begin{bmatrix} s_{w,t-T} + T v_{w,t-T} + T \varepsilon_{w,t}^v \\ v_{w,t-T} + \varepsilon_{w,t}^v \\ \exp\left(-\frac{T}{2}\left((y_{s,t}^\omega - b_{s,t-T}^\omega) + \varepsilon_{s,t}^\omega\right)\right) \odot q_{sw,t-T} \\ b_{s,t-T}^\omega + \varepsilon_{s,t}^{b\omega} \end{bmatrix} \quad (3.37a)$$

$$u_t^T = y_{s,t}^\omega \quad (3.37b)$$

$$v_t^T = [\varepsilon_{w,t}^v \quad \varepsilon_{s,t}^\omega \quad \varepsilon_{s,t}^{b\omega}] \quad (3.37c)$$

The measurement model includes only the position of the feature points on the video frames:

$$y_t = m_t \quad (3.38a)$$

$$e_t^T = e_t^c \quad (3.38b)$$

The other equations of EKF used in this method are given below. Time model equations are:

$$\hat{x}_{t|t-T} = f(\hat{x}_{t-T|t-T}, y_{s,t}^w, 0, 0, 0) \quad (3.39a)$$

$$P_{t|t-T} = F_t P_{t-T|t-T} F_t^T + V_t Q_t V_t^T + V_m Q_m V_m^T \quad (3.39b)$$

Measurement model equations are:

$$z_t = y_t - h(\hat{x}_{t|t-T}, 0) \quad (3.40)$$

The linearization of the time model equations are:

$$F_t = \frac{\partial f}{\partial x} \Big|_{(\hat{x}_{t-T|t-T}, y_{s,t}^w, 0, 0, 0)} \quad (3.41a)$$

$$V_t = \frac{\partial f}{\partial v} \Big|_{(\hat{x}_{t-T|t-T}, y_{s,t}^w, 0, 0, 0)} \quad (3.41b)$$

$$Q_t = \text{diag}([\Sigma_{\varepsilon_w^v,t} \quad \Sigma_{\varepsilon_s^\omega,t} \quad \Sigma_{\varepsilon_s^{b\omega},t}]) \quad (3.41c)$$

$$V_m = \left. \frac{\partial f}{\partial y_{s,t}^\omega} \right|_{(\hat{x}_{t-T|t-T}, y_{s,t}^\omega, 0, 0, 0)} \quad (3.41d)$$

$$Q_m = \sigma_{e_{s,t}^\omega}^2 \quad (3.41e)$$

The linearization of the measurement model equations are:

$$H_t = \left. \frac{\partial h}{\partial x} \right|_{(\hat{x}_{t|t-T}, 0)} \quad (3.42a)$$

$$R_t = \text{diag}([\sigma_{e_{x,t}^c}^2 \quad \sigma_{e_{y,t}^c}^2]) \quad (3.42b)$$

3.1.7 Linear Acceleration Data Used As Measurement

In this method; the video frames and linear acceleration data are used as measurement. However, angular velocity data is not used in this method, so the state vector \mathbf{x}_t does not include angular velocity:

$$\mathbf{x}^T = [s_w^T \quad v_w^T \quad a_w^T \quad b_s^{aT} \quad q_{sw}^T] \quad (3.43)$$

The angular velocity data is not used, so the biases of the gyroscope are not in the state vector \mathbf{x}^T .

In this method, the time model assumes constant linear acceleration and constant angular difference because there is no angular velocity data. The angular velocity is not used in this method, so the angular velocity is not used in time model like state vector. Time model is:

$$x_t = \begin{bmatrix} s_{w,t} \\ v_{w,t} \\ a_{w,t} \\ b_{s,t}^a \\ q_{sw,t} \end{bmatrix} = \begin{bmatrix} s_{w,t-T} + T v_{w,t-T} + \frac{T}{2} a_{w,t-T} + \frac{T}{2} \varepsilon_{w,t}^a \\ v_{w,t-T} + T a_{w,t-T} + T \varepsilon_{w,t}^a \\ a_{w,t-T} + \varepsilon_{w,t}^a \\ b_{s,t-T}^a + \varepsilon_{s,t}^{b^a} \\ \exp\left(\frac{1}{2} \varepsilon_{s,t}^\theta\right) \odot q_{sw,t-T} \end{bmatrix} \quad (3.44a)$$

$$u_t^T = 0 \quad (3.44b)$$

$$v_t^T = [\varepsilon_{w,t}^a \quad \varepsilon_{s,t}^\theta \quad \varepsilon_{s,t}^{b^a}] \quad (3.44c)$$

where $\varepsilon_{s,t}^\theta$ denote time independent Gaussian process noise.

The measurement model includes the linear acceleration, and the position of the feature points on the video frames, respectively:

$$y_t = \begin{bmatrix} y_{s,t}^a \\ m_t \end{bmatrix} \quad (3.45a)$$

$$e_t^T = [e_{s,t}^a \quad e_t^c] \quad (3.45b)$$

The other equations of EKF used in this method are given below. Time model equations are:

$$\hat{x}_{t|t-T} = f(\hat{x}_{t-T|t-T}, 0, 0, 0) \quad (3.46a)$$

$$P_{t|t-T} = F_t P_{t-T|t-T} F_t^T + V_t Q_t V_t^T \quad (3.46b)$$

The linearization of the time model equations are:

$$F_t = \left. \frac{\partial f}{\partial x} \right|_{(\hat{x}_{t-T|t-T}, 0, 0, 0)} \quad (3.47a)$$

$$V_t = \left. \frac{\partial f}{\partial v} \right|_{(\hat{x}_{t-T|t-T}, 0, 0, 0)} \quad (3.47b)$$

$$Q_t = \text{diag}([\Sigma_{\varepsilon_w^a,t} \quad \Sigma_{\varepsilon_s^\theta,t} \quad \Sigma_{\varepsilon_s^{b^a}}]) \quad (3.47c)$$

The linearization of the measurement model equations are:

$$H_t = \left. \frac{\partial h}{\partial x} \right|_{(\hat{x}_{t|t-T}, 0, 0)} \quad (3.48a)$$

$$R_t = \text{diag}([\sigma_{e_{sx,t}^a}^2 \quad \sigma_{e_{sy,t}^a}^2 \quad \sigma_{e_{sz,t}^a}^2 \quad \sigma_{e_{x,t}^c}^2 \quad \sigma_{e_{y,t}^c}^2]) \quad (3.48b)$$

3.1.8 Linear Acceleration Data Used As Control Input

In this method; only the video frames are used as measurement. However, angular velocity data is not used, but linear acceleration data is used as control input in this method, so the state vector \mathbf{x}_t does not include linear acceleration and angular velocity:

$$\mathbf{x}^T = [s_w^T \quad v_w^T \quad b_s^{aT} \quad q_{sw}^T] \quad (3.49)$$

The linear acceleration data is still used, so the biases of the accelerometer are still in the state vector \mathbf{x}^T .

Like the method in Section 3.1.7, the time model assumes constant linear acceleration and constant angular difference because there is no angular velocity data. The angular velocity is not used in this method, so the angular velocity is not used in time model like state vector. On the other hand, because linear acceleration data is used as control input, it is not in the state vector but it is in the time model. Time model is:

$$\begin{bmatrix} s_{w,t} \\ v_{w,t} \\ b_{s,t}^a \\ q_{sw,t} \end{bmatrix} = \begin{bmatrix} s_{w,t-T} + T v_{w,t-T} + \frac{T^2}{2} \varepsilon_{w,t}^a + \frac{T^2}{2} Q_{ws,t-T} (y_{s,t}^a - b_{s,t-T}^a) - \frac{T^2}{2} g_w \\ v_{w,t-T} + T \varepsilon_{w,t}^a + T Q_{ws,t-T} (y_{s,t}^a - b_{s,t-T}^a) - T g_w \\ b_{s,t-T}^a + \varepsilon_{s,t}^a \\ \exp\left(\frac{1}{2} \varepsilon_{s,t}^\theta\right) \odot q_{sw,t-T} \end{bmatrix} \quad (3.50a)$$

$$u_t^T = y_{s,t}^a \quad (3.50b)$$

$$v_t^T = [\varepsilon_{w,t}^a \quad \varepsilon_{s,t}^\theta \quad \varepsilon_{s,t}^{b^a}] \quad (3.50c)$$

The measurement model includes only the position of the feature points on the video frames:

$$x_t = m_t \quad (3.51a)$$

$$e_t^T = e_t^c \quad (3.51b)$$

The other equations of EKF used in this method are given below. Time model equations are:

$$\hat{x}_{t|t-T} = f(\hat{x}_{t-T|t-T}, y_{s,t}^a, 0, 0, 0) \quad (3.52a)$$

$$P_{t|t-T} = F_t P_{t-T|t-T} F_t^T + V_t Q_t V_t^T + V_m Q_m V_m^T \quad (3.52b)$$

Measurement model equations are:

$$z_t = y_t - h(\hat{x}_{t|t-T}, 0) \quad (3.53)$$

The linearization of the time model equations are:

$$F_t = \left. \frac{\partial f}{\partial x} \right|_{(\hat{x}_{t-T|t-T}, y_{s,t}^a, 0, 0, 0)} \quad (3.54a)$$

$$V_t = \left. \frac{\partial f}{\partial v} \right|_{(\hat{x}_{t-T|t-T}, y_{s,t}^a, 0, 0, 0)} \quad (3.54b)$$

$$Q_t = \text{diag}([\Sigma_{\varepsilon_{w,t}^a} \quad \Sigma_{\varepsilon_{s,t}^\theta} \quad \Sigma_{\varepsilon_{s,t}^{b^a}}]) \quad (3.54c)$$

$$V_m = \left. \frac{\partial f}{\partial \sigma_{e_{s,t}^a}^2} \right|_{(\hat{x}_{t-T|t-T}, y_{s,t}^a, 0, 0, 0)} \quad (3.54d)$$

$$Q_m = \sigma_{e_{s,t}^a}^2 \quad (3.54e)$$

The linearization of the measurement model equations are:

$$H_t = \left. \frac{\partial h}{\partial x} \right|_{(\hat{x}_{t|t-T}, 0)} \quad (3.55a)$$

$$R_t = \text{diag} \left(\left[\sigma_{e_{x,t}^c}^2 \quad \sigma_{e_{y,t}^c}^2 \right] \right) \quad (3.55b)$$

3.1.9 Both Angular Velocity and Linear Acceleration Data Not Used

In this method; only the video frames are used as measurement. However, linear acceleration and angular velocity data are not used, so the state vector \mathbf{x}_t does not include linear acceleration and angular velocity:

$$\mathbf{x}^T = [s_w^T \quad v_w^T \quad q_{sw}^T] \quad (3.56)$$

In this method, the time model assumes constant linear velocity and constant angular difference because there is no linear acceleration and angular velocity data. The linear acceleration and angular velocity are not used in this method, so they are not used in time model like state vector. Time model is:

$$\mathbf{x}_t = \begin{bmatrix} s_{w,t} \\ v_{w,t} \\ q_{sw,t} \end{bmatrix} = \begin{bmatrix} s_{w,t-T} + T v_{w,t-T} + T \varepsilon_{w,t}^v \\ v_{w,t-T} + \varepsilon_{w,t}^v \\ \exp\left(-\frac{1}{2} \varepsilon_{s,t}^\theta\right) \odot q_{sw,t-T} \end{bmatrix} \quad (3.57a)$$

$$\mathbf{u}_t^T = 0 \quad (3.57b)$$

$$\mathbf{v}_t^T = [\varepsilon_{w,t}^v \quad \varepsilon_{s,t}^\theta] \quad (3.57c)$$

The measurement model includes only the position of the feature points on the video frames:

$$y_t = m_t \quad (3.58a)$$

$$e_t^T = e_t^c \quad (3.58b)$$

The other equations of EKF used in this method are given below. Time model equations are:

$$\hat{x}_{t|t-T} = f(x_{t-T|t-T}, 0, 0) \quad (3.59a)$$

$$P_{t|t-T} = F_t P_{t-T|t-T} F_t^T + V_t Q_t V_t^T \quad (3.59b)$$

Measurement model equations are:

$$z_t = y_t - h(\hat{x}_{t|t-T}, 0) \quad (3.60)$$

The linearization of the time model equations are:

$$F_t = \left. \frac{\partial f}{\partial x} \right|_{(x_{t-T|t-T}, 0, 0)} \quad (3.61a)$$

$$V_t = \left. \frac{\partial f}{\partial v} \right|_{(x_{t-T|t-T}, 0, 0)} \quad (3.61b)$$

$$Q_t = \text{diag}([\Sigma_{\varepsilon_w, t}^v \quad \Sigma_{\varepsilon_s, t}^\theta]) \quad (3.61c)$$

The linearization of the measurement model equations are:

$$H_t = \left. \frac{\partial h}{\partial x} \right|_{h(\hat{x}_{t|t-T}, 0)} \quad (3.62a)$$

$$R_t = \text{diag}([\sigma_{e_{x,t}}^2 \quad \sigma_{e_{y,t}}^2]) \quad (3.62b)$$

3.2 HYBRID FILTER DESIGN

9 different Hybrid Filter methods using video frames, accelerometer and gyroscope sensors are analyzed. Video frames are used in all models. In addition to video frames, angular velocity and linear acceleration data are used as measurement or control input in Hybrid Filter. These methods are:

1. Both angular velocity and linear acceleration data used as "measurement"
2. Angular velocity data used as "control input", linear acceleration data used as "measurement"
3. Angular velocity data used as "measurement", linear acceleration data used as "control input"
4. Both angular velocity and linear acceleration data used as "control input"
5. Angular velocity data used as "measurement"
6. Angular velocity data used as "control input"
7. Linear acceleration data used as "measurement"
8. Linear acceleration data used as "control input"
9. Both angular velocity and linear acceleration data "not" used

In hybrid filter, the feature points are not used in EKF filter directly. By using direct linear transformation, the pose of the camera can be estimated with these feature points. This estimated pose data is used in EKF with linear acceleration and angular velocity measurements to enhance the pose estimation. The difference of the hybrid filter design from EKF is the usage of the feature points.

3.2.1 Both Angular Velocity and Linear Acceleration Data Used As Measurement

In this method; all video frames, angular velocity, and linear acceleration data are used as measurement. The state vector \mathbf{x}_t comprises:

$$\mathbf{x}^T = [s_w^T \quad v_w^T \quad a_w^T \quad b_s^a{}^T \quad q_{sw}^T \quad \omega_s^T \quad b_s^\omega{}^T] \quad (3.63)$$

where s_w denotes the position of camera, v_w the linear velocity of camera, a_w the linear acceleration of camera, b_s^a the biases of accelerometer inertial sensor, q_{sw} the orientation quaternion, ω_s the angular velocity of camera, b_s^ω the biases of gyroscope inertial sensor. To simplify the time model, the position s_w , linear velocity v_w , linear acceleration a_w , and orientation q_{sw} of the inertial sensors with respect to the world frame are estimated in the state vector.

The time model of this method is same as the EKF design in Section 3.1.1.

The measurement model of the Hybrid filter includes the accelerometer data, gyroscope data, and the 3D position (pose and angle) data obtained from structure from motion by using the feature points:

$$y_t = \begin{bmatrix} y_{s,t}^a \\ y_{s,t}^\omega \\ y_{s,t}^q \\ y_{s,t}^s \end{bmatrix} \quad (3.64a)$$

$$e_t^T = [e_{s,t}^a \quad e_{s,t}^\omega \quad e_{s,t}^q \quad e_{s,t}^s] \quad (3.64b)$$

where, $y_{s,t}^q$ denotes the quaternion version of the rotation matrix, $y_{s,t}^s$ denotes the pose of the camera, $e_{s,t}^q$, and $e_{s,t}^s$ denote time independent Gaussian measurement noise.

The other equations of EKF used in this method are same as Section 3.1.1.

The linearization of the measurement model of this method:

$$H_t = \left. \frac{\partial h}{\partial x} \right|_{(\hat{x}_{t|t-T}, 0, 0, 0, 0)} \quad (3.65a)$$

$$R_t = \text{diag} \left(\left[\sigma_{e_{sx,t}}^2 \quad \sigma_{e_{sy,t}}^2 \quad \sigma_{e_{sz,t}}^2 \quad \sigma_{e_{s\omega}}^2 \quad \sigma_{e_{s\omega}}^2 \quad \sigma_{e_{s\omega}}^2 \quad \sigma_{e_{sq}}^2 \quad \sigma_{e_{sq}}^2 \quad \sigma_{e_{sq}}^2 \quad \sigma_{e_{sq}}^2 \quad \sigma_{e_{ss}}^2 \quad \sigma_{e_{ss}}^2 \quad \sigma_{e_{ss}}^2 \right] \right) \quad (3.65b)$$

3.2.2 Angular Velocity Data Used As Control Input, Linear Acceleration Data Used As Measurement

In this method; video frames, and linear acceleration data are used as measurement. However, angular velocity data is used as control input. Thus, the state vector \mathbf{x}_t does not include angular velocity:

$$\mathbf{x}^T = [s_w^T \quad v_w^T \quad a_w^T \quad b_s^{aT} \quad q_{sw}^T \quad b_s^{\omega T}] \quad (3.66)$$

The angular velocity data is still used, so the biases of the gyroscope are still in the state vector x^T .

The time model of this method is same as the EKF design in Section 3.1.2.

The measurement model of the Hybrid filter includes the accelerometer data, and the 3D position (pose and angle) data obtained from structure from motion by using the feature points:

$$y_t = \begin{bmatrix} y_{s,t}^a \\ y_{s,t}^q \\ y_{s,t}^s \end{bmatrix} \quad (3.67a)$$

$$e_t^T = [e_{s,t}^a \quad e_{s,t}^q \quad e_{s,t}^s] \quad (3.67b)$$

The other equations of EKF used in this method are same as Section 3.1.2.

The linearization of the measurement model of this method:

$$H_t = \left. \frac{\partial h}{\partial x} \right|_{(\hat{x}_{t|t-T}, 0, 0, 0)} \quad (3.68a)$$

$$R_t = \text{diag} \left(\left[\sigma_{e_{sx,t}}^2 \quad \sigma_{e_{sy,t}}^2 \quad \sigma_{e_{sz,t}}^2 \quad \sigma_{e_{sw,t}}^2 \quad \sigma_{e_{sx,t}}^2 \quad \sigma_{e_{sy,t}}^2 \quad \sigma_{e_{sz,t}}^2 \quad \sigma_{e_{sx,t}}^2 \quad \sigma_{e_{sy,t}}^2 \quad \sigma_{e_{sz,t}}^2 \right] \right) \quad (3.68b)$$

3.2.3 Angular Velocity Data Used As Measurement, Linear Acceleration Data Used As Control Input

In this method; video frames, and angular velocity data are used as measurement. However, linear acceleration data is used as control input. Thus, the state vector x_t does not include linear acceleration:

$$x^T = [s_w^T \quad v_w^T \quad b_s^{aT} \quad q_{sw}^T \quad \omega_s^T \quad b_s^{\omega T}] \quad (3.69)$$

The linear acceleration data is still used, so the biases of the accelerometer are still in the state vector x^T .

The time model of this method is same as the EKF design in Section 3.1.3.

The measurement model of the Hybrid filter includes the gyroscope data, and the 3D position (pose and angle) data obtained from structure from motion by using the feature points:

$$y_t = \begin{bmatrix} y_{s,t}^\omega \\ y_{s,t}^q \\ y_{s,t}^s \end{bmatrix} \quad (3.70a)$$

$$e_t^T = [e_{s,t}^\omega \quad e_{s,t}^q \quad e_{s,t}^s] \quad (3.70b)$$

The other equations of EKF used in this method are same as Section 3.1.3.

The linearization of the measurement model of this method:

$$H_t = \left. \frac{\partial h}{\partial x} \right|_{(\hat{x}_{t|t-T}, 0, 0, 0)} \quad (3.71a)$$

$$R_t = \text{diag} \left(\left[\sigma_{e_{sx,t}^\omega}^2 \quad \sigma_{e_{sy,t}^\omega}^2 \quad \sigma_{e_{sz,t}^\omega}^2 \quad \sigma_{e_{sw,t}^q}^2 \quad \sigma_{e_{sx,t}^q}^2 \quad \sigma_{e_{sy,t}^q}^2 \quad \sigma_{e_{sz,t}^q}^2 \quad \sigma_{e_{sx,t}^s}^2 \quad \sigma_{e_{sy,t}^s}^2 \quad \sigma_{e_{sz,t}^s}^2 \right] \right) \quad (3.71b)$$

3.2.4 Both Angular Velocity and Linear Acceleration Data Used As Control Input

In this method; only video frames are used as measurement. However, linear acceleration and angular velocity data are used as control input. Thus, the state vector x_t does not include linear acceleration and angular velocity:

$$x^T = [s_w^T \quad v_w^T \quad b_s^{aT} \quad q_{sw}^T \quad b_s^{\omega T}] \quad (3.72)$$

The linear acceleration and angular velocity data are still used, so the biases of the accelerometer and gyroscope are still in the state vector x^T .

The time model of this method is same as the EKF design in Section 3.1.4.

The measurement model of the Hybrid filter includes only the 3D position (pose and angle) data obtained from structure from motion by using the feature points:

$$y_t = \begin{bmatrix} y_{s,t}^q \\ y_{s,t}^s \end{bmatrix} \quad (3.73a)$$

$$e_t^T = [e_{s,t}^q \quad e_{s,t}^s] \quad (3.73b)$$

The other equations of EKF used in this method are same as Section 3.1.4.

The linearization of the measurement model of this method:

$$H_t = \left. \frac{\partial h}{\partial x} \right|_{(\hat{x}_{t|t-T}, 0, 0)} \quad (3.74a)$$

$$R_t = \text{diag} \left(\left[\sigma_{e_{sw,t}^q}^2 \quad \sigma_{e_{sx,t}^q}^2 \quad \sigma_{e_{sy,t}^q}^2 \quad \sigma_{e_{sz,t}^q}^2 \quad \sigma_{e_{sx,t}^s}^2 \quad \sigma_{e_{sy,t}^s}^2 \quad \sigma_{e_{sz,t}^s}^2 \right] \right) \quad (3.74b)$$

3.2.5 Angular Velocity Data Used As Measurement

In this method; the video frames and angular velocity data are used as measurement. However, linear acceleration data is not used in this method, so the state vector x_t does not include linear acceleration:

$$x^T = [s_w^T \quad v_w^T \quad q_{sw}^T \quad \omega_s^T \quad b_s^{\omega T}] \quad (3.75)$$

The linear acceleration data is not used, so the biases of the accelerometer are not in the state vector x^T .

The time model of this method is same as the EKF design in Section 3.1.5.

The measurement model of the Hybrid filter includes the gyroscope data, and the 3D position (pose and angle) data obtained from structure from motion by using the feature points:

$$y_t = \begin{bmatrix} y_{s,t}^\omega \\ y_{s,t}^q \\ y_{s,t}^s \end{bmatrix} \quad (3.76a)$$

$$e_t^T = [e_{s,t}^\omega \quad e_{s,t}^q \quad e_{s,t}^s] \quad (3.76b)$$

The other equations of EKF used in this method are same as Section 3.1.5.

The linearization of the measurement model of this method:

$$H_t = \left. \frac{\partial h}{\partial x} \right|_{(\hat{x}_{t|t-T}, 0, 0, 0)} \quad (3.77a)$$

$$R_t = \text{diag} \left(\left[\sigma_{e_{sx,t}}^2 \quad \sigma_{e_{sy,t}}^2 \quad \sigma_{e_{sz,t}}^2 \quad \sigma_{e_{sw,t}}^2 \quad \sigma_{e_{sx,t}}^2 \quad \sigma_{e_{sy,t}}^2 \quad \sigma_{e_{sz,t}}^2 \quad \sigma_{e_{sx,t}}^2 \quad \sigma_{e_{sy,t}}^2 \quad \sigma_{e_{sz,t}}^2 \right] \right) \quad (3.77b)$$

3.2.6 Angular Velocity Data Used As Control Input

In this method; only the video frames are used as measurement. However, linear acceleration data is not used, but angular velocity data is used as control input in this method, so the state vector x_t does not include linear acceleration and angular velocity:

$$x^T = [s_w^T \quad v_w^T \quad q_{sw}^T \quad b_s^{\omega T}] \quad (3.78)$$

The angular velocity data is still used, so the biases of the gyroscope are still in the state vector x^T .

The time model of this method is same as the EKF design in Section 3.1.6.

The measurement model of the Hybrid filter includes only the 3D position (pose and angle) data obtained from structure from motion by using the feature points:

$$y_t = \begin{bmatrix} y_{s,t}^q \\ y_{s,t}^s \end{bmatrix} \quad (3.79a)$$

$$e_t^T = [e_{s,t}^q \quad e_{s,t}^s] \quad (3.79b)$$

The other equations of EKF used in this method are same as Section 3.1.6.

The linearization of the measurement model of this method:

$$H_t = \left. \frac{\partial h}{\partial x} \right|_{(\hat{x}_{t|t-T}, 0, 0)} \quad (3.80a)$$

$$R_t = \text{diag} \left(\left[\sigma_{e_{sw,t}^q}^2 \quad \sigma_{e_{sx,t}^q}^2 \quad \sigma_{e_{sy,t}^q}^2 \quad \sigma_{e_{sz,t}^q}^2 \quad \sigma_{e_{sx,t}^s}^2 \quad \sigma_{e_{sy,t}^s}^2 \quad \sigma_{e_{sz,t}^s}^2 \right] \right) \quad (3.80b)$$

3.2.7 Linear Acceleration Data Used As Measurement

In this method; the video frames and linear acceleration data are used as measurement. However, angular velocity data is not used in this method, so the state vector x_t does not include angular velocity:

$$x^T = [s_w^T \quad v_w^T \quad a_w^T \quad b_s^{aT} \quad q_{sw}^T] \quad (3.81)$$

The angular velocity data is not used, so the biases of the gyroscope are not in the state vector x^T .

The time model of this method is same as the EKF design in Section 3.1.7.

The measurement model of the Hybrid filter includes the accelerometer data, and the 3D position (pose and angle) data obtained from structure from motion by using the feature points:

$$y_t = \begin{bmatrix} y_{s,t}^a \\ y_{s,t}^q \\ y_{s,t}^s \end{bmatrix} \quad (3.82a)$$

$$e_t^T = [e_{s,t}^a \quad e_{s,t}^q \quad e_{s,t}^s] \quad (3.82b)$$

The other equations of EKF used in this method are same as Section 3.1.7.

The linearization of the measurement model of this method:

$$H_t = \left. \frac{\partial h}{\partial x} \right|_{(\hat{x}_{t|t-T}, 0, 0, 0)} \quad (3.83a)$$

$$R_t = \text{diag} \left(\left[\sigma_{e_{sx,t}^a}^2 \quad \sigma_{e_{sy,t}^a}^2 \quad \sigma_{e_{sz,t}^a}^2 \quad \sigma_{e_{sw,t}^q}^2 \quad \sigma_{e_{sx,t}^q}^2 \quad \sigma_{e_{sy,t}^q}^2 \quad \sigma_{e_{sz,t}^q}^2 \quad \sigma_{e_{sx,t}^s}^2 \quad \sigma_{e_{sy,t}^s}^2 \quad \sigma_{e_{sz,t}^s}^2 \right] \right) \quad (3.83b)$$

3.2.8 Linear Acceleration Data Used As Control Input

In this method; only the video frames are used as measurement. However, angular velocity data is not used, but linear acceleration data is used as control input in this method, so the state vector x_t does not include linear acceleration and angular velocity:

$$x^T = [s_w^T \quad v_w^T \quad b_s^{aT} \quad q_{sw}^T] \quad (3.84)$$

The linear acceleration data is still used, so the biases of the accelerometer are still in the state vector x^T .

The time model of this method is same as the EKF design in Section 3.1.8.

The measurement model of the Hybrid filter includes only the 3D position (pose and angle) data obtained from structure from motion by using the feature points:

$$y_t = \begin{bmatrix} y_{s,t}^q \\ y_{s,t}^s \end{bmatrix} \quad (3.85a)$$

$$e_t^T = [e_{s,t}^q \quad e_{s,t}^s] \quad (3.85b)$$

The other equations of EKF used in this method are same as Section 3.1.8.

The linearization of the measurement model of this method:

$$H_t = \left. \frac{\partial h}{\partial x} \right|_{(\hat{x}_t|_{t-T}, 0, 0)} \quad (3.86a)$$

$$R_t = \text{diag} \left(\left[\sigma_{e_{sw,t}}^2 \quad \sigma_{e_{sx,t}}^2 \quad \sigma_{e_{sy,t}}^2 \quad \sigma_{e_{sz,t}}^2 \quad \sigma_{e_{sx,t}}^2 \quad \sigma_{e_{sy,t}}^2 \quad \sigma_{e_{sz,t}}^2 \right] \right) \quad (3.86b)$$

3.2.9 Both Angular Velocity and Linear Acceleration Data Not Used

In this method; only the video frames are used as measurement. However, linear acceleration and angular velocity data are not used, so the state vector x_t does not include linear acceleration and angular velocity:

$$x^T = [s_w^T \quad v_w^T \quad q_{sw}^T] \quad (3.87)$$

The angular velocity and linear acceleration data are not used, so the biases of both gyroscope and accelerometer are not in the state vector x^T .

The time model of this method is same as the EKF design in Section 3.1.9.

The measurement model of the Hybrid filter includes only the 3D position (pose and angle) data obtained from structure from motion by using the feature points:

$$y_t = \begin{bmatrix} y_{s,t}^q \\ y_{s,t}^s \end{bmatrix} \quad (3.88a)$$

$$e_t^T = [e_{s,t}^q \quad e_{s,t}^s] \quad (3.88b)$$

The other equations of EKF used in this method are same as Section 3.1.9.

The linearization of the measurement model of this method:

$$H_t = \left. \frac{\partial h}{\partial x} \right|_{(\hat{x}_{t|t-T}, 0, 0)} \quad (3.89a)$$

$$R_t = \text{diag} \left(\left[\sigma_{e_{sw,t}}^2 \quad \sigma_{e_{sx,t}}^2 \quad \sigma_{e_{sy,t}}^2 \quad \sigma_{e_{sz,t}}^2 \quad \sigma_{e_{sx,t}}^s \quad \sigma_{e_{sy,t}}^s \quad \sigma_{e_{sz,t}}^s \right] \right) \quad (3.89b)$$

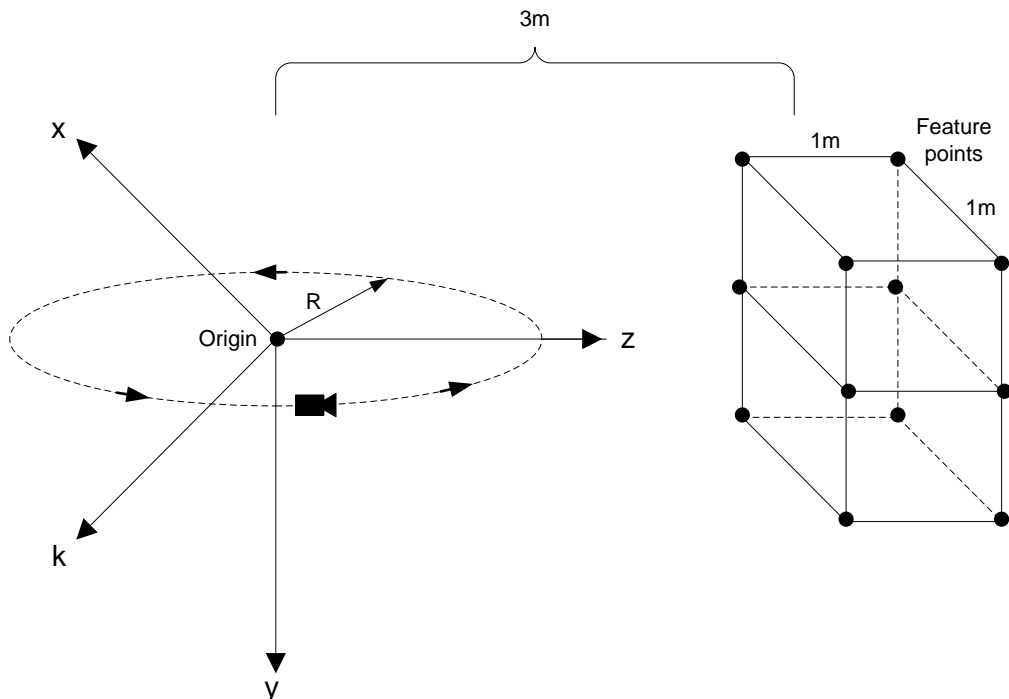
4. EXPERIMENTAL SETUP AND RESULTS

4.1 SIMULATION STRUCTURE

In this section, the simulation platform and performance evolution of the algorithms are provided. We compare the nine different EKF visual-inertial sensor tracking results. Also, we compare the EKF with mixed filter and pose estimation using only feature points (no EKF and no mixed filter).

For simulation, we used 3D camera motion including both 3D linear acceleration and 3D angular velocity. We used the edges of two cubes that are on top as the feature points (totally 12 points). In Figure 4.1, 3D linear camera motion and 3D feature points can be seen.

Figure 4.1: Translational movement



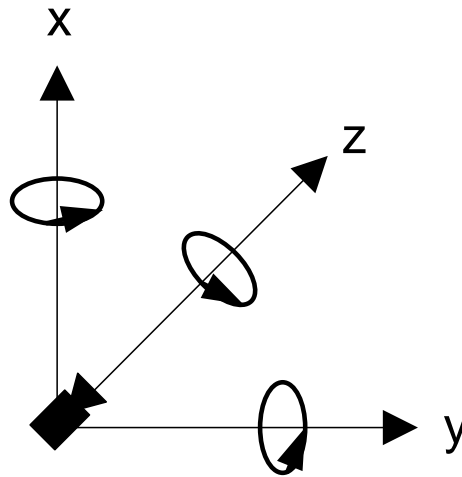
The equations of the linear camera motion are:

$$\left. \begin{aligned} x &= \frac{1}{\sqrt{2}}R\cos(\omega t) \\ y &= \frac{1}{\sqrt{2}}R\cos(\omega t) \\ z &= R\sin(\omega t) \end{aligned} \right\} \omega: \frac{2\pi}{T_{translation}} \text{ (frequency)} \quad (4.1)$$

where $T_{translation}$ denotes the translational movement period.

The camera rotational movement can be seen in Figure 4.2.

Figure 4.2: Rotational movement



The equations of the rotational camera motion are:

$$\left. \begin{aligned} \theta_x &= A_x \sin(\omega t) \\ \theta_y &= A_y \sin(\omega t) \\ \theta_z &= A_z \sin(\omega t) \end{aligned} \right\} \begin{aligned} A_x, A_y, A_z: \max(\theta_x), \max(\theta_y), \max(\theta_z) \\ \omega &= \frac{2\pi}{T_{rotation}} \end{aligned} \quad (4.2)$$

where $T_{rotation}$ denotes the rotational movement period, A_x , A_y , and A_z denote the maximum rotation angle of the camera in x , y , and z direction, respectively. In our simulation we take:

$$A_x = A_y = A_z = \frac{12\pi}{180} \quad (4.3)$$

We also examine how the results are affected when the maximum rotation angles of the camera are changed.

The camera motion used in the experiments is the combination of the translational and the rotational movements shown in Figure 4.1.1 and Figure 4.1.2, respectively. As a result, the camera movement chosen for simulation includes both 3D translational and 3D rotational movements.

In the experiments, the camera coordinate system and the sensor coordinate system are assumed to be same. As stated in previous chapter, the relations of the coordinate systems of the camera and inertial systems are fixed, so R_{cs} is identity matrix:

$$R_{cs} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (4.4)$$

4.2 NOISES

In the simulations, two kinds of noises are used:

1. Measurement Noises
2. Time Model Noises

4.2.1 Measurement Noises

For the inertial sensor measurement noise, we used the noise value written on the datasheet of the sensors.

We used LSM303DLH 3-axis accelerometer sensor for 3D linear acceleration and the linear acceleration noise density is $218 \mu g/\sqrt{Hz}$. In the algorithm, the linear acceleration noise is:

$$e_{s,t}^a = \sqrt{(218 \times 10^{-6})^2 \times f_s^a} \quad (4.5)$$

where f_s^a denotes the sampling rate of the accelerometer.

We used LY330ALH yaw-rate gyroscope and LPR430AL dual axis pitch analog gyroscope for 3D angular velocity. The angular velocity noise density of the two gyroscopes are $0.014 \text{ dps}/\sqrt{Hz}$ and $0.018 \text{ dps}/\sqrt{Hz}$, respectively. In the algorithm, the angular velocity noise is:

$$e_{s,t}^w = \sqrt{\left(\frac{\pi}{180} \times 0.018\right)^2 \times f_s^w} \quad (4.6)$$

where f_s^w denotes the sampling rate of the gyroscope.

The measurement noise of the feature points on the camera frames is related to the translational and rotational movement of the camera. When the translational movement or rotational movement increases, the measurement noise of the feature points also increases (Ercan & Erdem 2011).

$$e_t^c = \sqrt{\left(\frac{2RF}{T_{translation}f_s^c}\right)^2 + \left(\frac{2\tan(A_x)F}{T_{rotation}f_s^c}\right)^2} + \alpha f_s^c \quad (4.7)$$

where F denotes the focal length of the camera, R denotes the radius of the translational movement, f_s^c denotes the sampling rate of the camera, α depends on the camera properties as well as external factors.

4.2.2 Time Model Noises

For time model noises of linear acceleration and angular velocity, we used the real 3D translational and rotational movements. We use the change value of the real 3D translational and rotational movements at the points where they match the predicted values from linear model at least. For instance, if the real linear acceleration on x axis is:

$$a = \sin\left(\frac{2\pi t}{T}\right) \quad (4.8)$$

the time model noise of the linear acceleration is:

$$v_{s,t}^a = \frac{2\pi}{T} \max(a) T_s \quad (4.9)$$

where T_s denotes the sampling period of the overall system.

4.3 EKF RESULTS

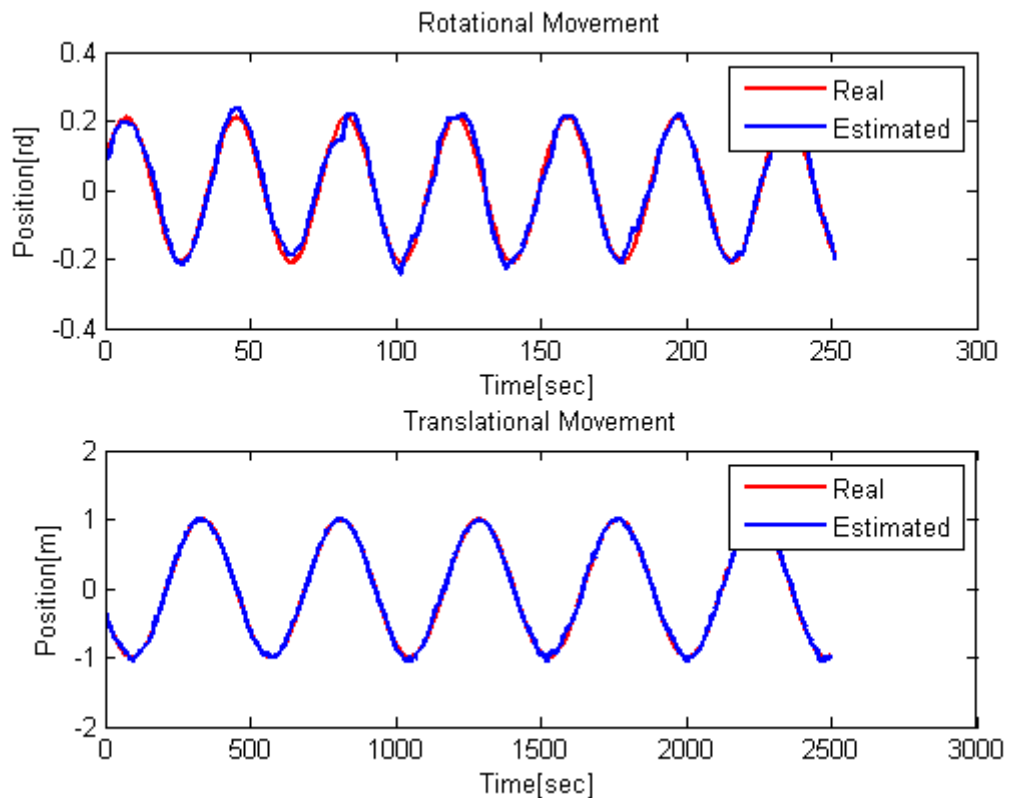
Our experiment results are showing in the following figures. We had 9 different EKF methods and implemented all of them. The result includes the individual results of these 9 methods, also comparison of them with each other.

The period of the translational movement of the camera in the figures below is 4 seconds, and the period of the rotational movement of the camera is 0.33 seconds. Measurement noises of the feature points, linear acceleration, and angular velocity are $1.55 \times 10^{-4} \frac{\text{pixel}}{\text{focal length}}$, $5.71 \times 10^{-4} \text{ m/s}^2$, and $4.48 \times 10^{-5} \frac{\text{rad}}{\text{s}}$, respectively.

4.3.1 Both Angular Velocity and Linear Acceleration Data Used As Measurement

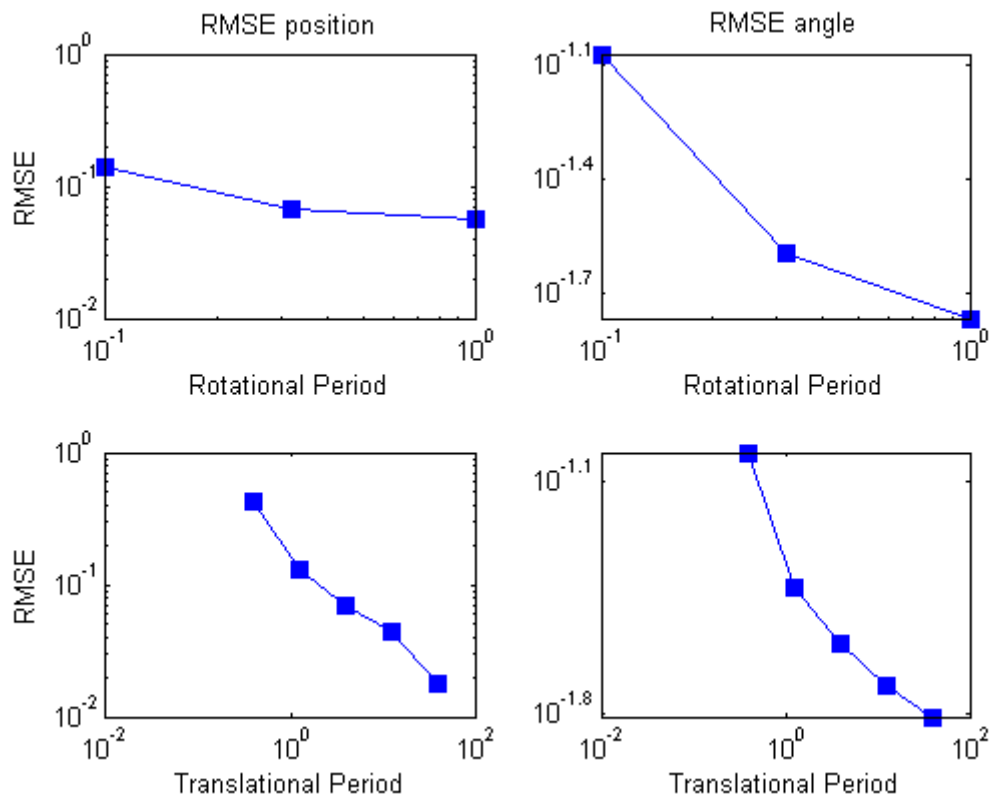
In Figure 4.3, the comparison of the real and estimated translational and rotational movement of the camera with $10^{0.6}$ s translational period and $10^{-0.5}$ s rotational period can be seen when all inertial sensor data and camera data are used as measurement. When both the accelerometer and the gyroscope data are used as measurement, performance of the camera tracking is good for both rotational and translation motion of the camera. So, true rotational and translational motion (red lines) and estimated rotational and translational motion (blue lines) of the camera are very close to each other in Figure 4.3.

Figure 4.3: Estimated and true motion of the camera when both angular velocity and linear acceleration data are used as measurement in EKF



Depending on the period of the camera motion, tracking performance varies. When the rotational or translational camera motion velocity increases, RMSE value between the real and estimated map increases. In Figure 4.4, the changes in RMSE values can be realized according to translational and rotational period in this case.

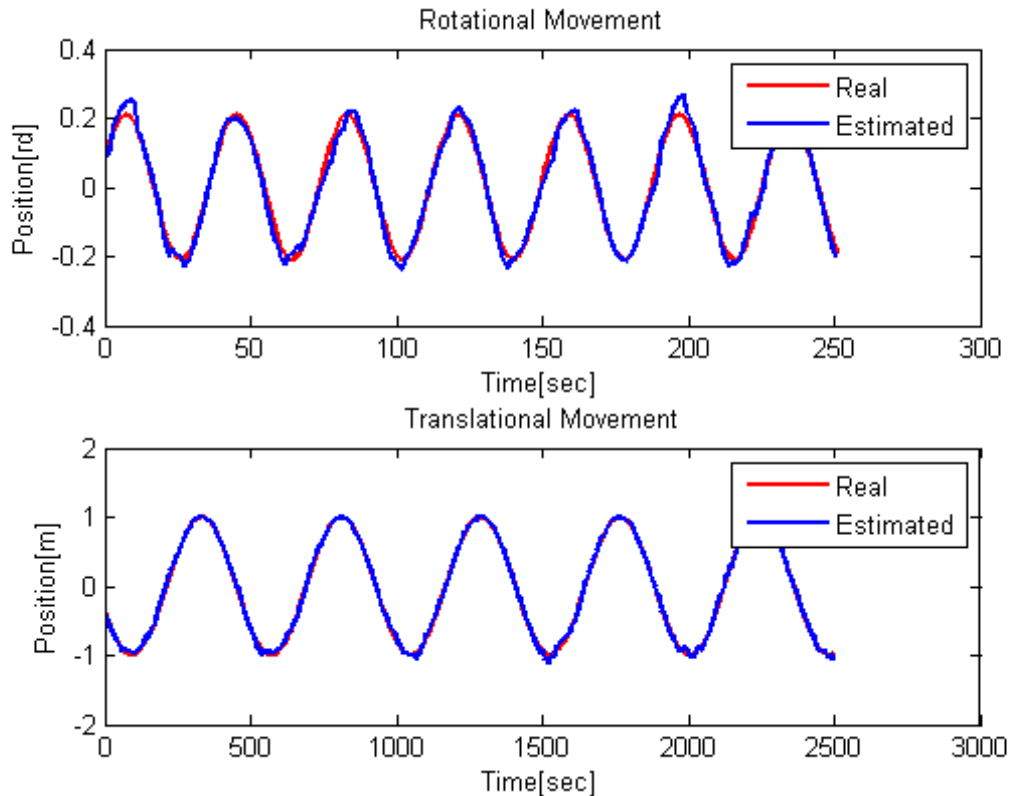
Figure 4.4: Change in RMSE of the position and orientation of the camera with rotational and translational period of the camera when both angular velocity and linear acceleration data are used as measurement in EKF



4.3.2 Angular Velocity Data Used As Control Input, Linear Acceleration Data Used As Measurement

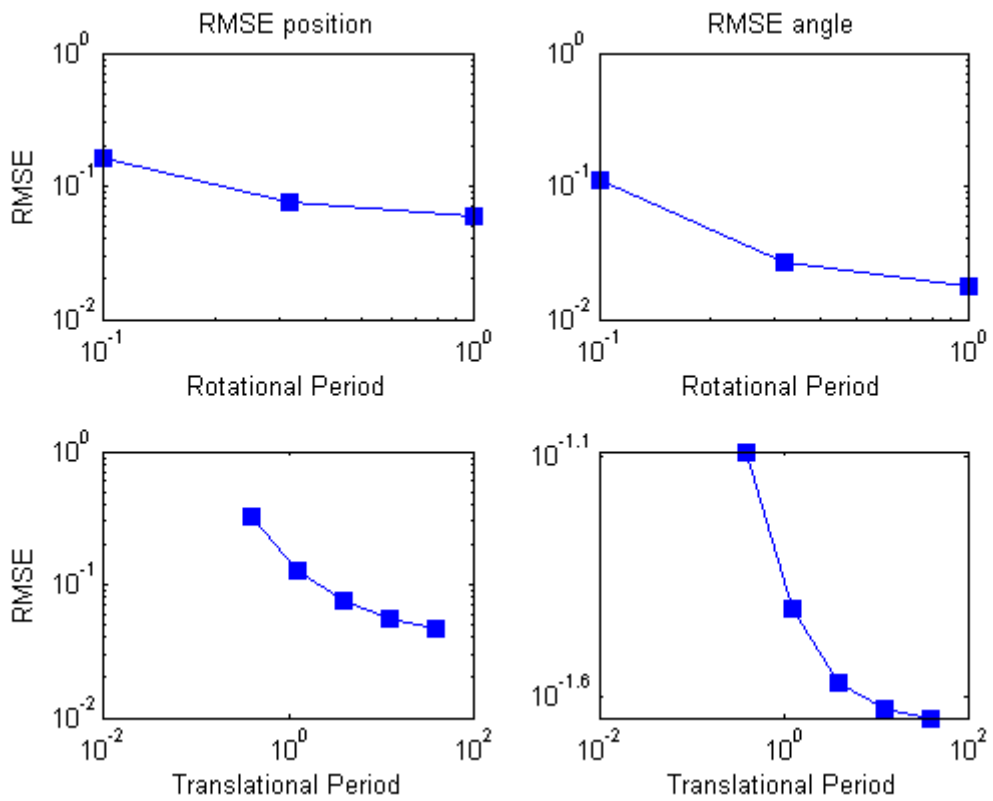
In Figure 4.5, the comparison of the real and estimated translational and rotational movement of the camera with $10^{0.6}$ s translational period and $10^{-0.5}$ s rotational period can be found when the linear acceleration data and camera data are used as measurement and the angular velocity data is used as control input. When the accelerometer is used as measurement and the gyroscope is used as control input instead of measurement does not affect the tracking performance significantly. The camera tracking performance is still good and the estimated motion of the camera for both translational and rotational directions is very close to true motion of the camera.

Figure 4.5: Estimated and true motion of the camera when angular velocity data is used as control input, linear acceleration data is used as measurement in EKF



In Figure 4.6, the changes in the RMSE values can be found according to translational and rotational period in this case. When the rotational or translational camera motion velocity increases, RMSE value between the real and estimated map increases.

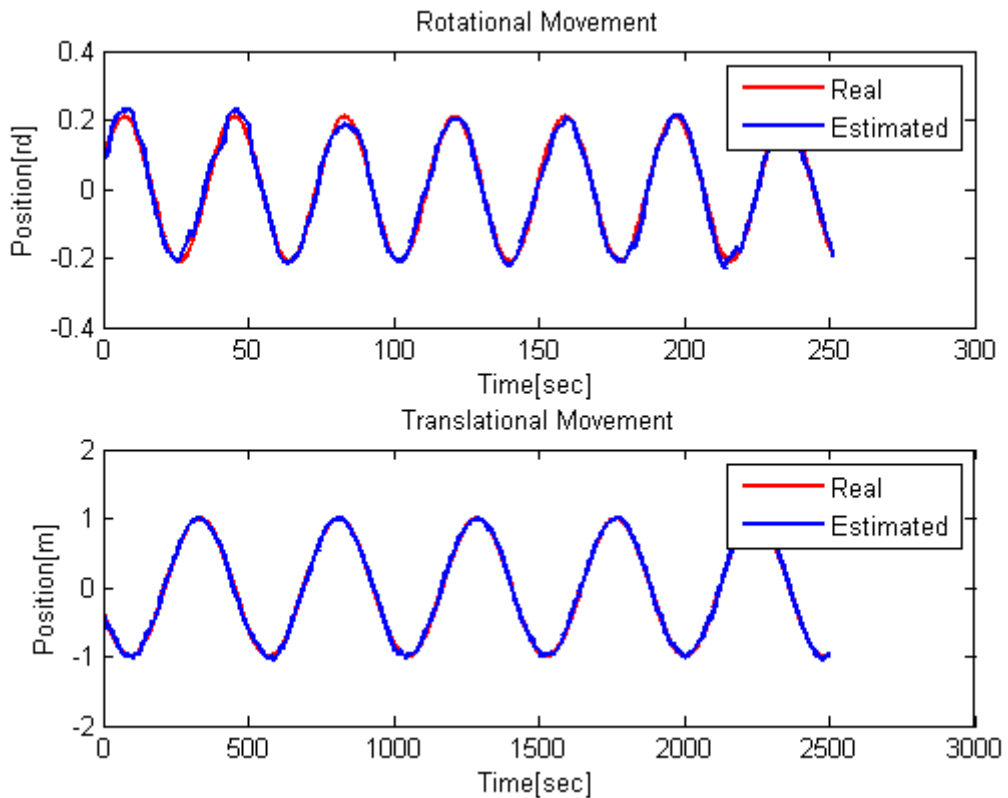
Figure 4.6: Change in RMSE of the position and orientation of the camera with rotational and translational period of the camera when angular velocity data is used as control input, linear acceleration data is used as measurement in EKF



4.3.3 Angular Velocity Data Used As Measurement, Linear Acceleration Data Used As Control Input

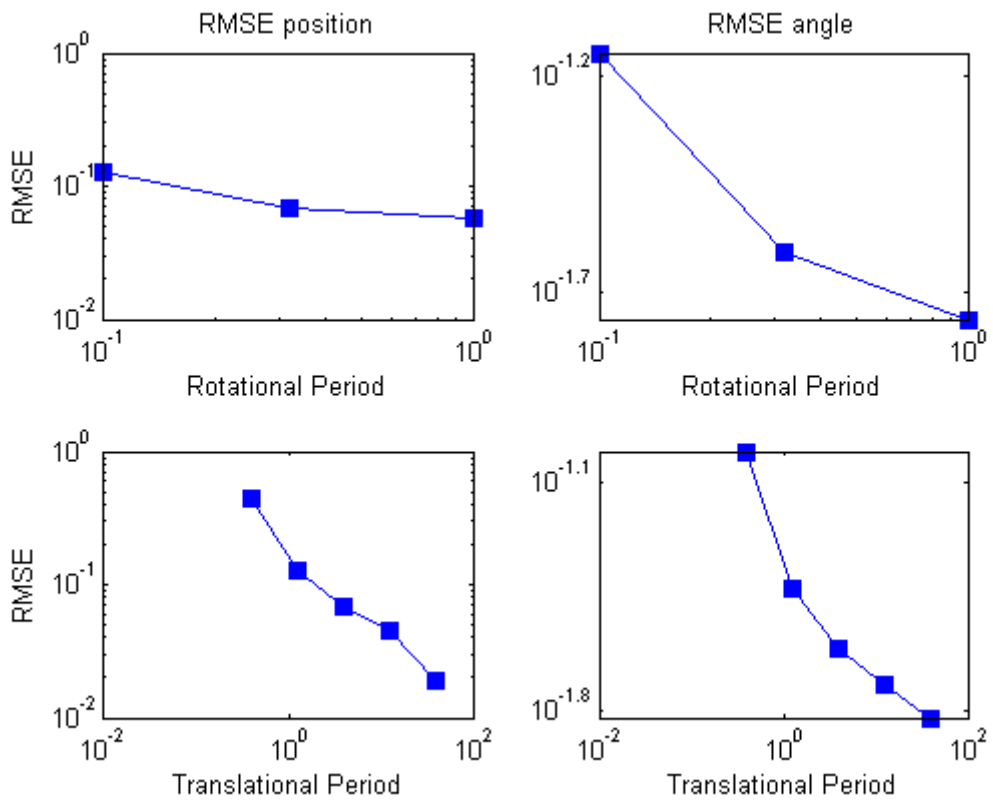
In Figure 4.7, the comparison of the real and estimated translational and rotational movement of the camera with $10^{0.6}$ s translational period and $10^{-0.5}$ s rotational period can be found when the angular velocity data and camera data are used as measurement and the linear acceleration data is used as control input. Similar to case 4.3.2, when the accelerometer is used as control input instead of measurement and the gyroscope is used as measurement does not affect the tracking performance significantly. The camera tracking performance is still good and the estimated motion of the camera for both translational and rotational directions is very close to true motion of the camera.

Figure 4.7: Estimated and true motion of the camera when angular velocity data is used as measurement, linear acceleration data is used as control input in EKF



In Figure 4.8, the changes in the RMSE values can be seen according to the translational and rotational period in this case. When the rotational or translational camera motion velocity increases, RMSE value between the real and estimated map increases.

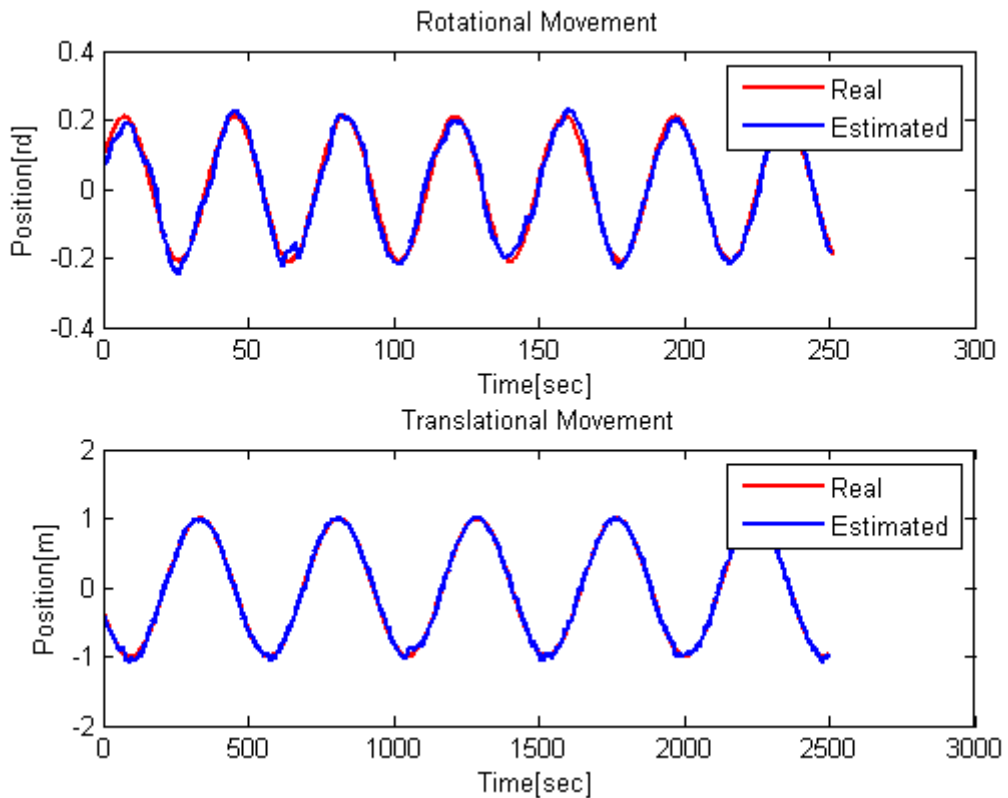
Figure 4.8: Change in RMSE of the position and orientation of the camera with rotational and translational period of the camera when angular velocity data is used as measurement, linear acceleration data is used as control input in EKF



4.3.4 Both Angular Velocity and Linear Acceleration Data Used As Control Input

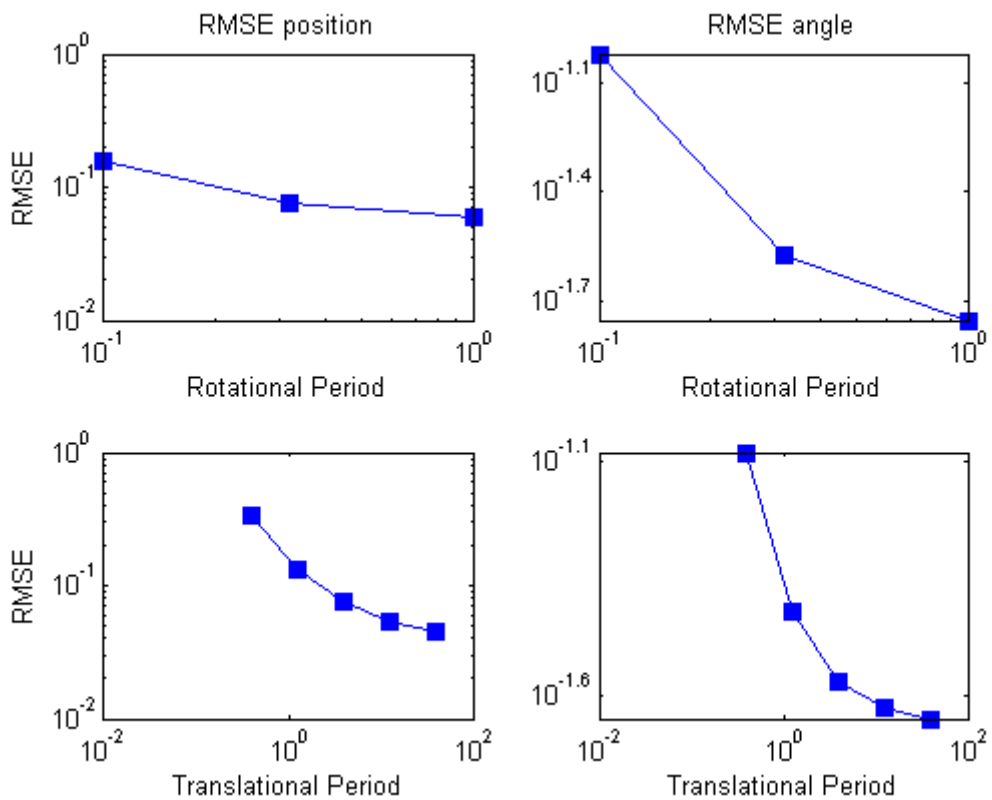
In Figure 4.9, the comparison of the real and the estimated translational and rotational movement of the camera with $10^{0.6}$ s translational period and $10^{-0.5}$ s rotational period can be found when all inertial sensor data is used as control input and only camera data are used as measurement. When both the accelerometer and gyroscope are used as control input instead of measurement does not affect the tracking performance significantly. The camera tracking performance is still good and the estimated motion of the camera in both translational and rotational directions is very close to true motion of the camera (Figure 4.9). However, if we compare the results with Figure 4.3, the rotational motion is more affected than translational motion.

Figure 4.9: Estimated and true motion of the camera when both angular velocity and linear acceleration data are used as control input in EKF



In Figure 4.10, the changes in the RMSE values can be found according to the translational and rotational period in this case. When the rotational or translational camera motion velocity increases, RMSE value between the real and estimated map increases.

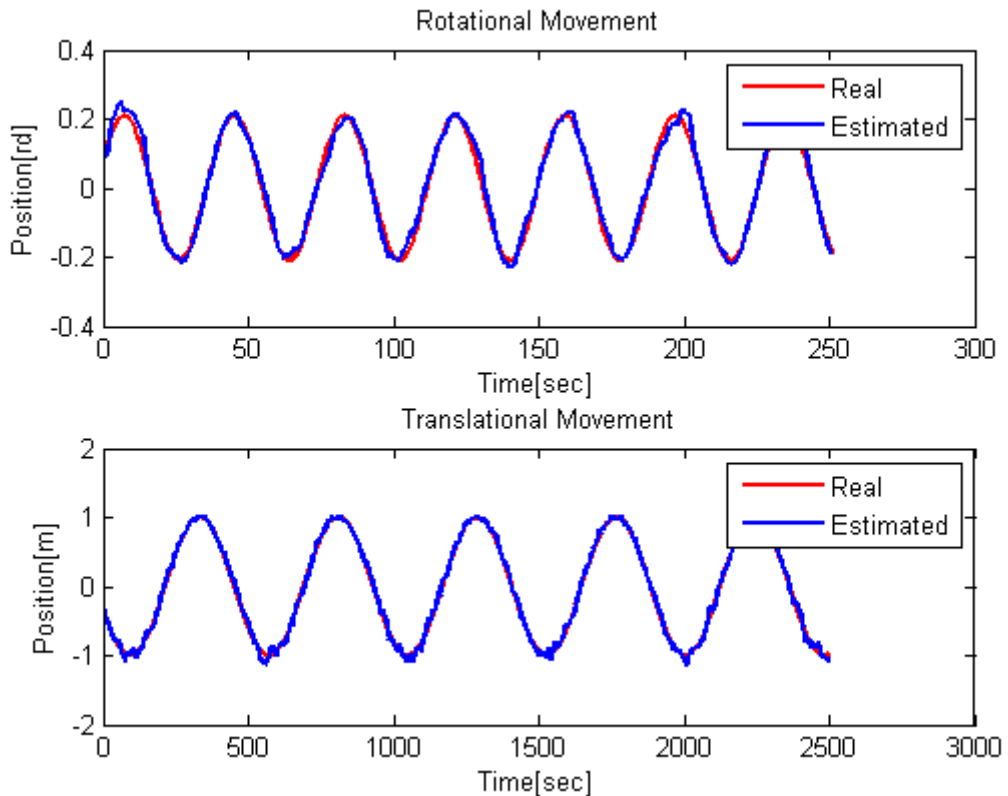
Figure 4.10: Change in RMSE of the position and orientation of the camera with rotational and translational period of the camera when both angular velocity and linear acceleration data are used as control input in EKF



4.3.5 Angular Velocity Data Used As Measurement

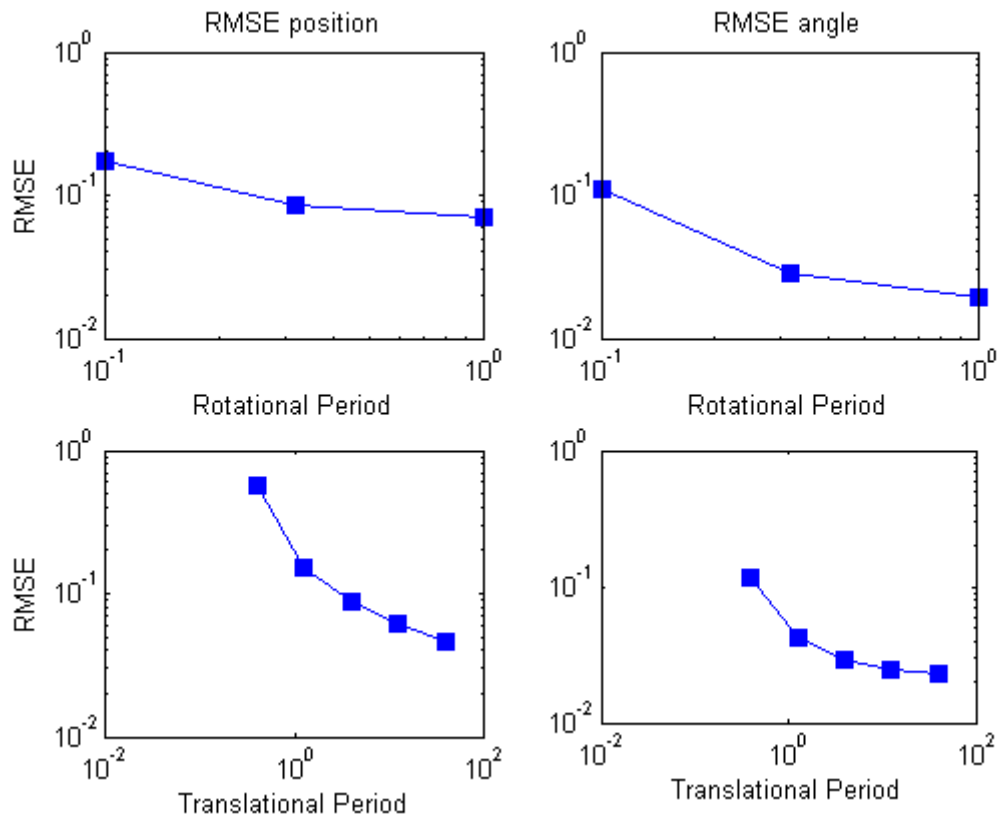
In Figure 4.11, the comparison of the real and estimated translational and rotational movement of the camera with $10^{0.6}$ s translational period and $10^{-0.5}$ s rotational period can be found when the angular velocity data and the camera data are used as measurement and the linear acceleration data is not used anymore. When the gyroscope is used as measurement and accelerometer is not used, the tracking performance of the camera decreases for both translational and rotational motion of the camera and the tracking is not good as the case 4.3.1 that both the accelerometer and the gyroscope are used as measurement (Figure 4.11). Not using accelerometer affects the tracking performance of the camera moving translational and rotational.

Figure 4.11: Estimated and true motion of the camera when angular velocity data is used as measurement in EKF



In Figure 4.12, the changes in the RMSE values change can be found according to the translational and rotational period in this case. When the rotational or translational camera motion velocity increases, RMSE value between the real and estimated map increases.

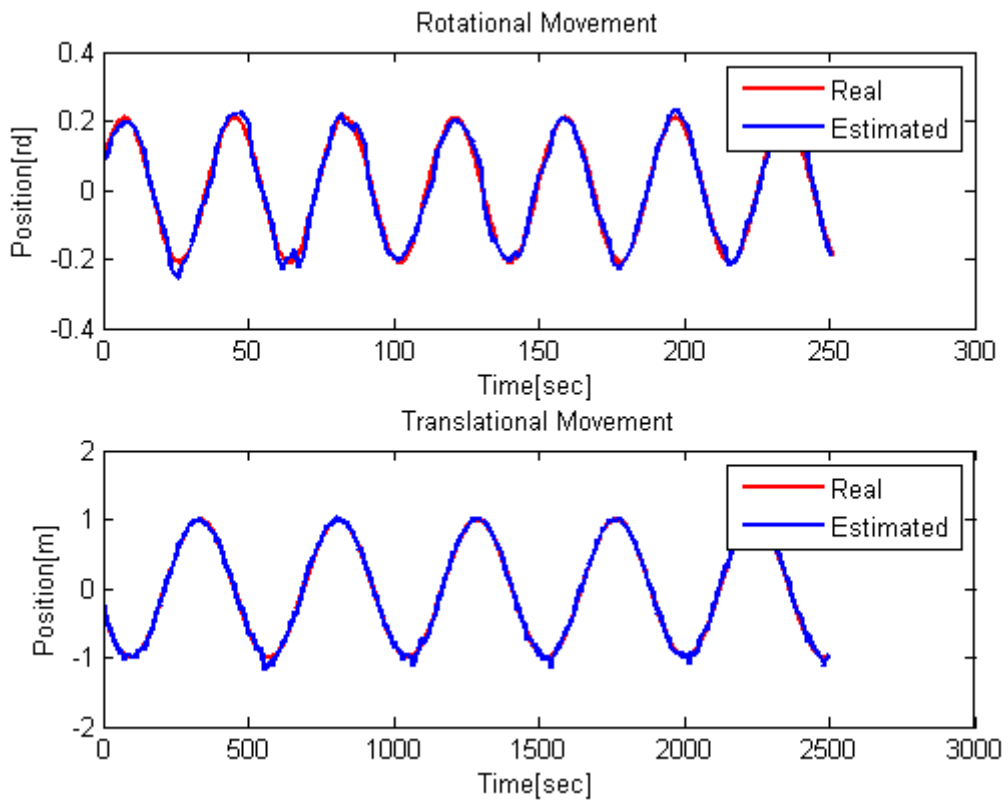
Figure 4.12: Change in RMSE of the position and orientation of the camera with rotational and translational period of the camera when angular velocity data is used as measurement in EKF



4.3.6 Angular Velocity Data Used As Control Input

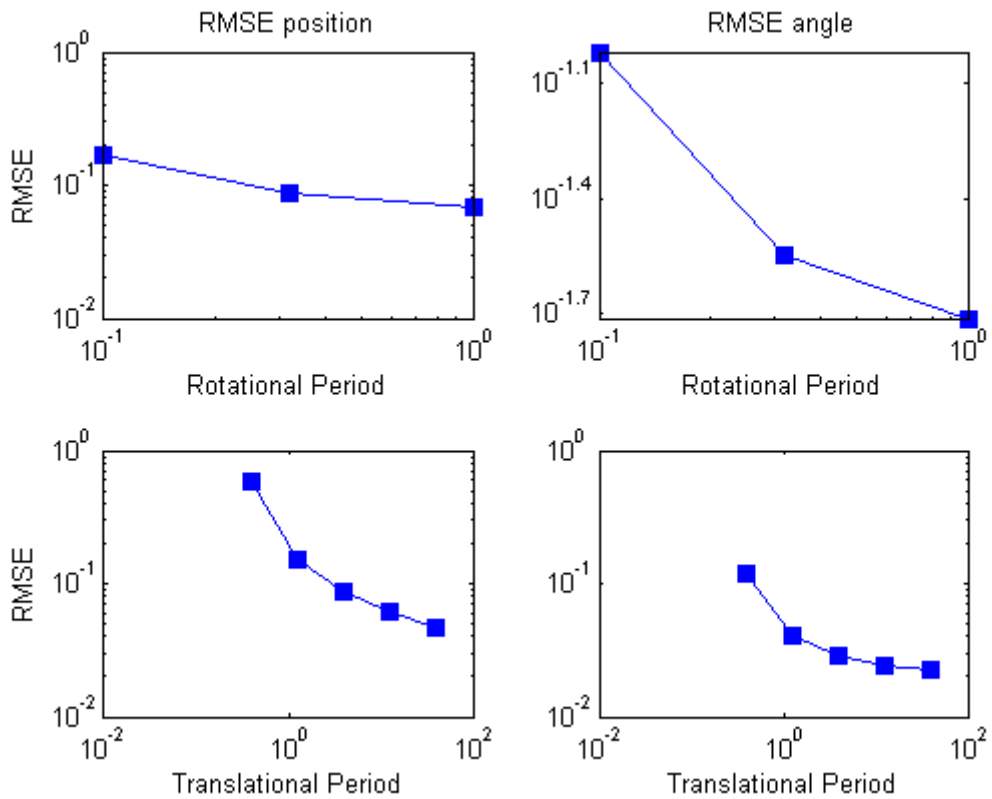
In Figure 4.13, the comparison of the real and estimated translational and rotational movement of the camera with $10^{0.6}$ s translational period and $10^{-0.5}$ s rotational period can be found when the angular velocity data is used as control input and the camera data is used as measurement and the linear acceleration data is not used anymore. When the gyroscope is used as control input and accelerometer is not used, the tracking performance of the camera is very similar to case 4.3.5. Using gyroscope as control input or measurement without accelerometer does not change the tracking performance.

Figure 4.13: Estimated and true motion of the camera when angular velocity data is used as control input in EKF



In Figure 4.14, the changes in the RMSE values can be seen according to translational and rotational period in this case. When the rotational or translational camera motion velocity increases, RMSE value between the real and estimated map increases.

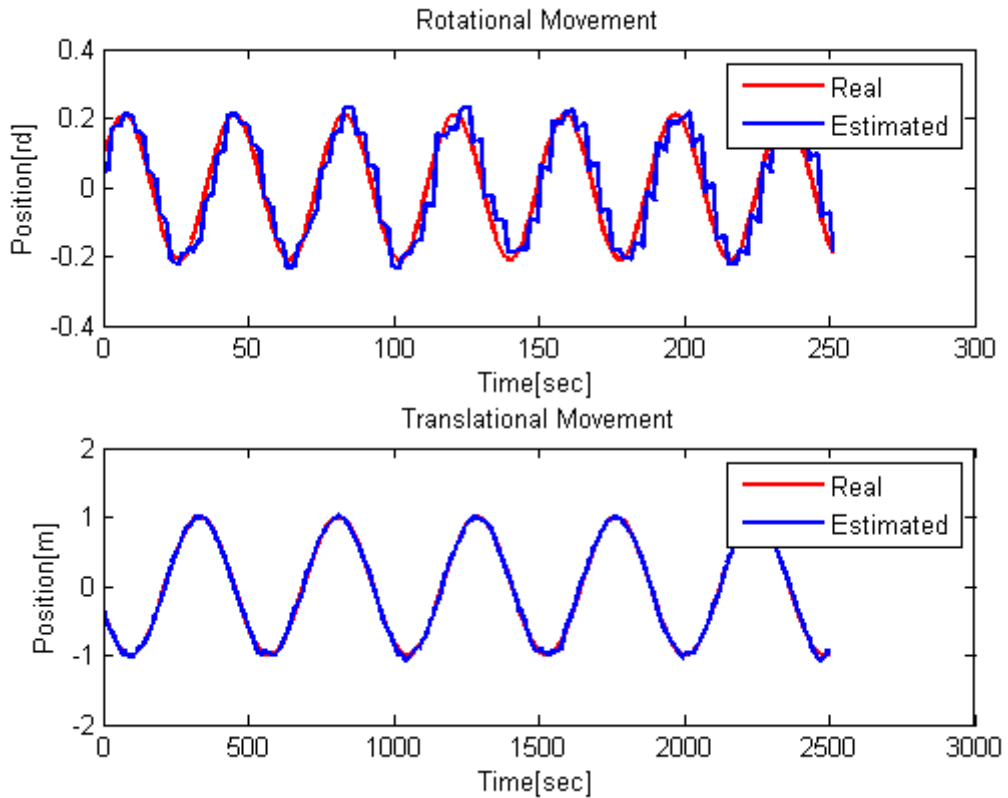
Figure 4.14: Change in RMSE of the position and orientation of the camera with rotational and translational period of the camera when angular velocity data is used as control input in EKF



4.3.7 Linear Acceleration Data Used As Measurement

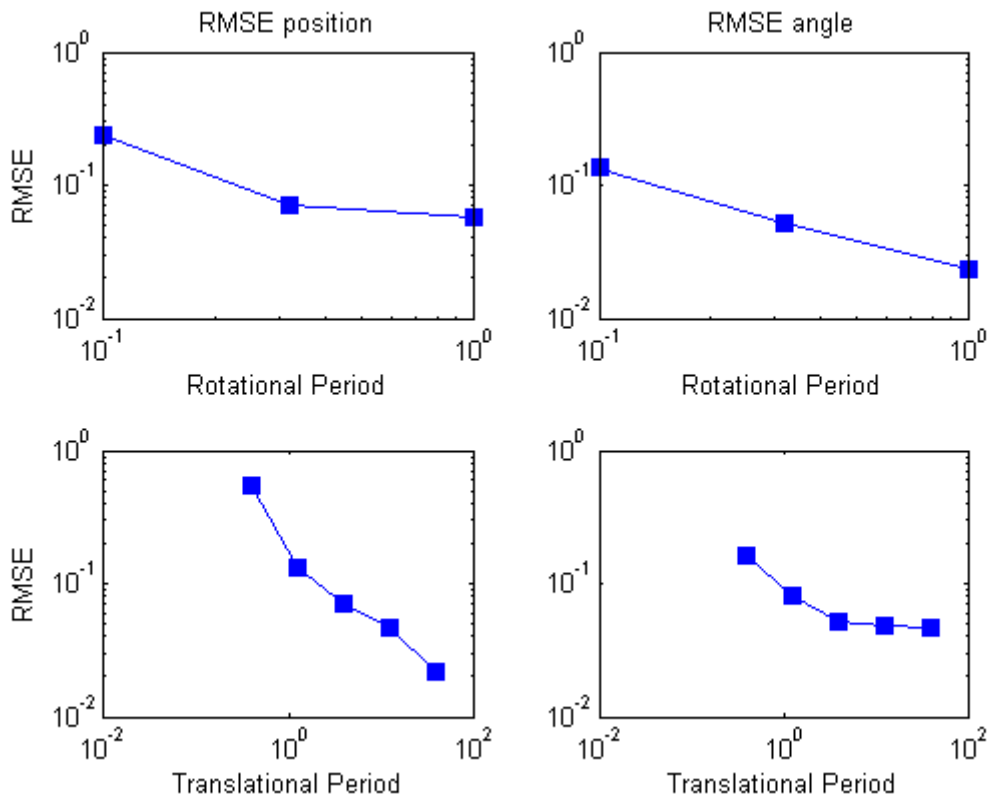
In Figure 4.15, the comparison of the real and estimated translational and rotational movement of the camera with $10^{0.6}$ s translational period and $10^{-0.5}$ s rotational period can be found when the linear acceleration data and the camera data are used as measurement and the angular velocity data is not used anymore. When the accelerometer is used as measurement and the gyroscope is not used, the camera tracking performance is affected significantly. Specially, the rotational motion is affected and the tracking performance in rotational direction is awful (Figure 4.15). Without gyroscope, accelerometer is not enough to track the camera for rotational motion. But the translational motion of the camera is better than case 4.3.5. Using accelerometer improve the tracking performance of the camera in translational motion.

Figure 4.15: Estimated and true motion of the camera when linear acceleration data is used as measurement in EKF



In Figure 4.16, the changes in the RMSE values can be seen according to translational and rotational period in this case. When the rotational or translational camera motion velocity increases, RMSE value between the real and estimated map increases.

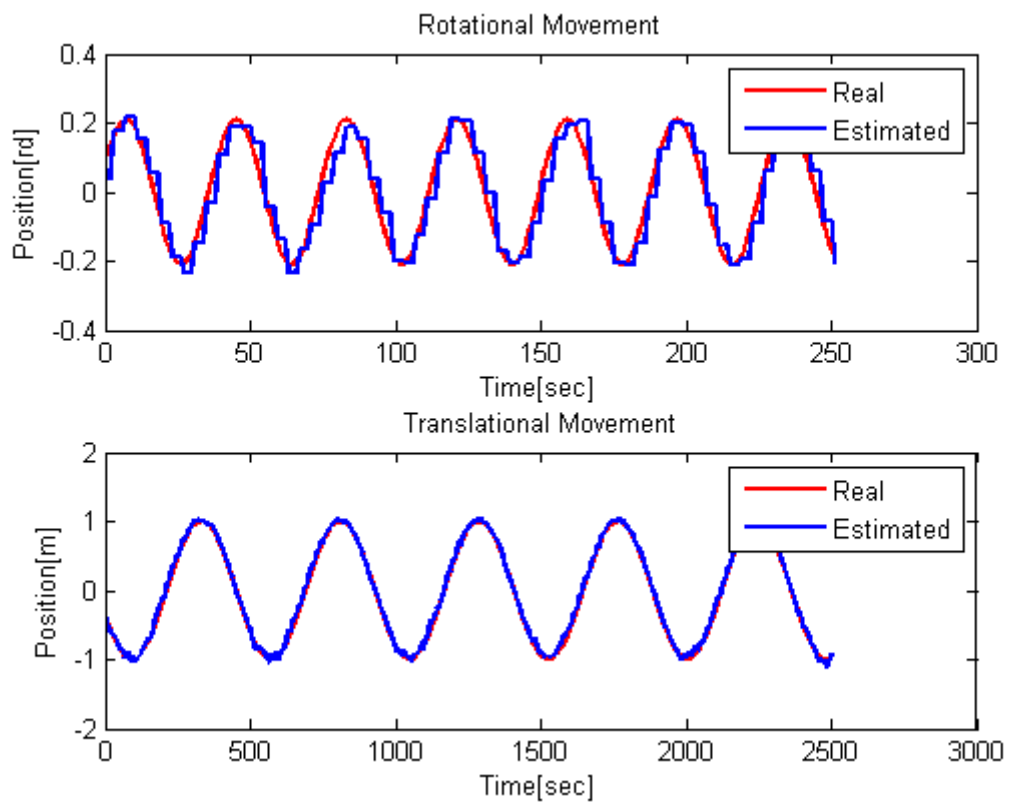
Figure 4.16: Change in RMSE of the position and orientation of the camera with rotational and translational period of the camera when linear acceleration data is used as measurement in EKF



4.3.8 Linear Acceleration Data Used As Control Input

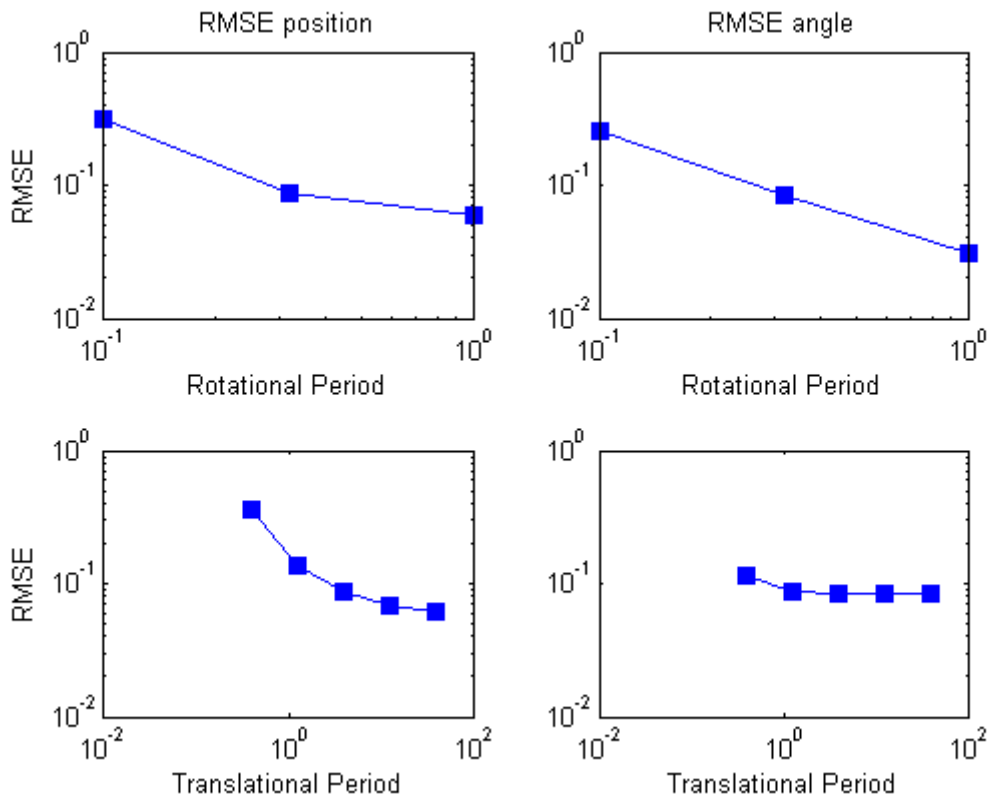
In Figure 4.17, the comparison of the real and estimated translational and rotational movement of the camera with $10^{0.6}$ s translational period and $10^{-0.5}$ s rotational period can be found when the linear acceleration data is used as control input, the camera data is used as measurement and the angular velocity is not used anymore. When the accelerometer is used as control and the gyroscope is not used, the camera tracking performance is very similar to case 4.3.7. Using accelerometer as control input reduces the translational tracking performance slightly.

Figure 4.17: Estimated and true motion of the camera when linear acceleration data is used as control input in EKF



In Figure 4.18, the changes in the RMSE values can be found according to translational and rotational period in this case. When the rotational or translational camera motion velocity increases, RMSE value between the real and estimated map increases.

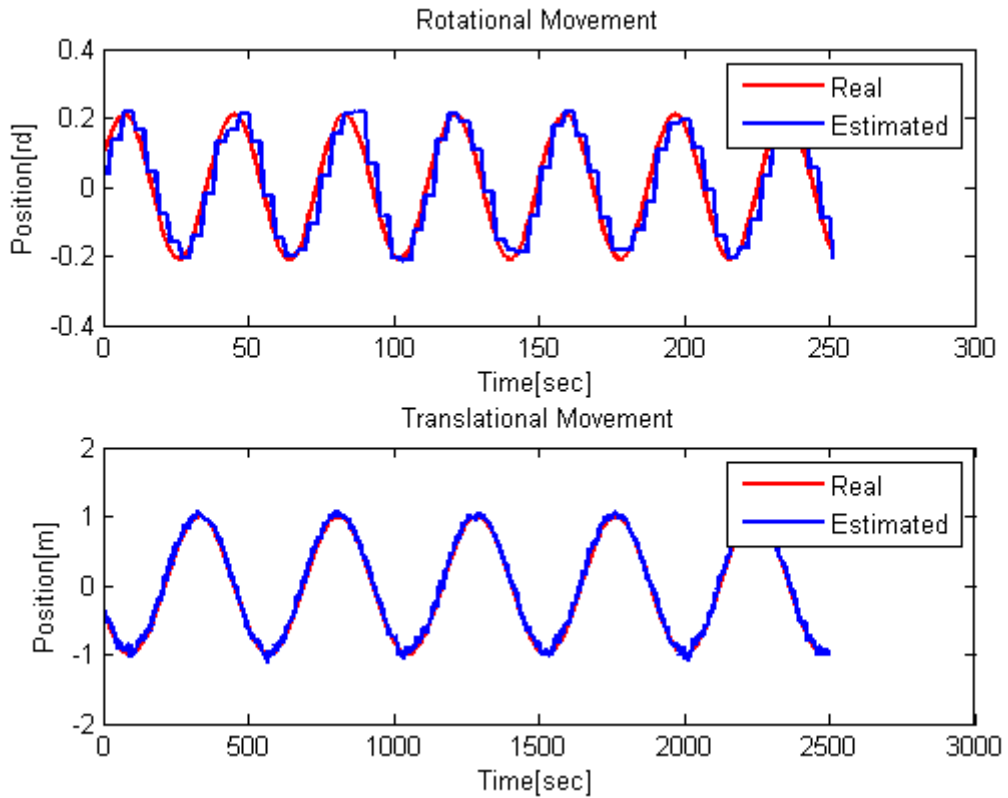
Figure 4.18: Change in RMSE of the position and orientation of the camera with rotational and translational period of the camera when linear acceleration data is used as control input in EKF



4.3.9 Both Angular Velocity and Linear Acceleration Data Not Used

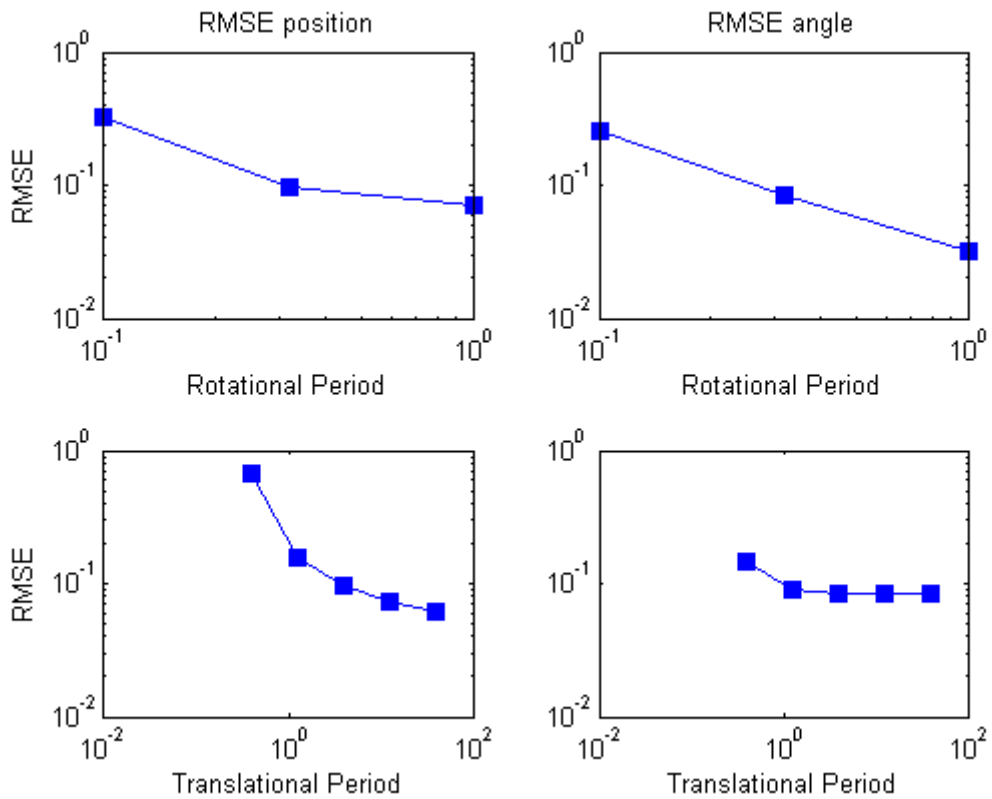
In Figure 4.19, the comparison of the real and estimated translational and rotational movement of the camera with $10^{0.6}$ s translational period and $10^{-0.5}$ s rotational period can be found when only the camera data is used as measurement, the angular velocity data and linear acceleration data are not used anymore. Tracking with only camera (without inertial sensors) gives the worst result. Both translational and rotational motion estimation is poor (Figure 4.19). Using inertial sensors with camera for tracking improves the performance of tracking. If the frequency of the camera is increased, the effects of the inertial sensors also will increase.

Figure 4.19: Estimated and true motion of the camera when both angular velocity and linear acceleration data are not used in EKF



In Figure 4.20, the changes in the RMSE values can be found according to translational and rotational period in this case. When the rotational or translational camera motion velocity increases, RMSE value between the real and estimated map increases.

Figure 4.20: Change in RMSE of the position and orientation of the camera with rotational and translational period of the camera when both angular velocity and linear acceleration data are not used in EKF



4.4 HYBRID FILTER RESULTS

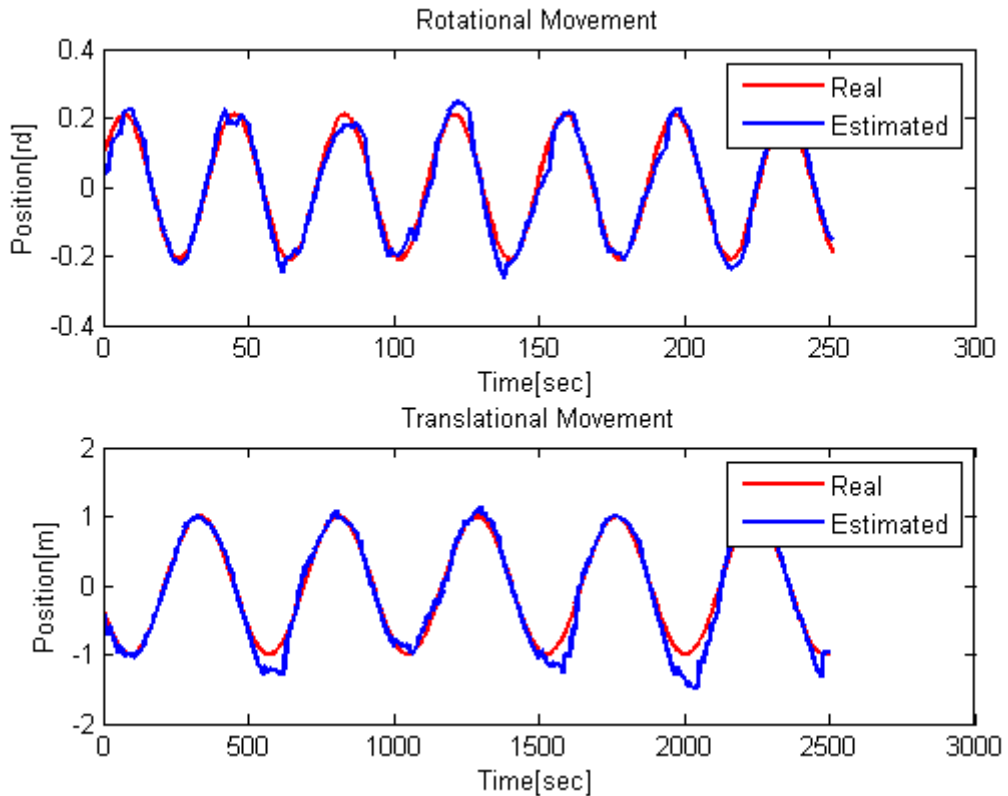
Our experiment results with Hybrid filter can be found in the following figures. We had 9 different Hybrid Filter methods and implemented all of them. The result includes all results of these 9 cases as well as their comparisons with each other.

The translational and rotational periods are the same as EKF Filter.

4.4.1 Both Angular Velocity and Linear Acceleration Data Used As Measurement

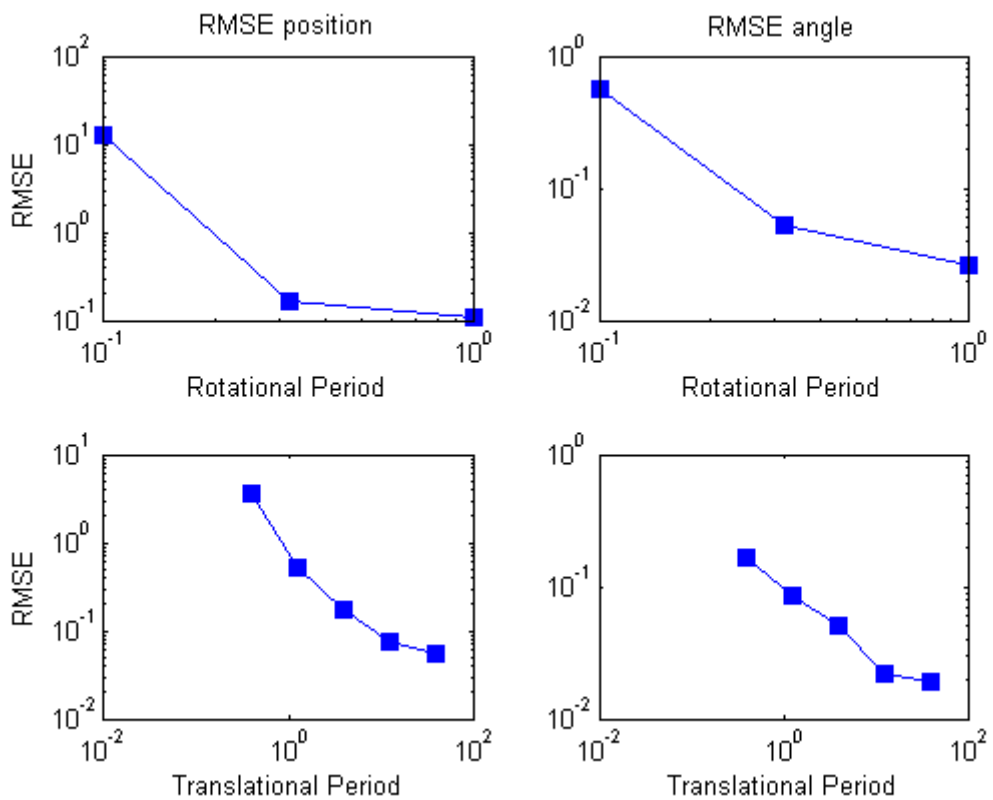
In Figure 4.21, the comparison of the real and estimated translational and rotational movement of the camera with $10^{0.6}$ s translational period and $10^{-0.5}$ s rotational period can be found when all inertial sensor data and camera data are used as measurement. When both the accelerometer and the gyroscope data are used as measurement, performance of the camera is good in both rotational and translation motions of the camera for hybrid filter. If we compare the results with EKF results, EKF gives better results than hybrid filter. The case that both the accelerometer and gyroscope are used as measurement for EKF is given in 4.3 and the results is better than the results given in Figure 4.21.

Figure 4.21: Estimated and true motion of the camera when both angular velocity and linear acceleration data are used as measurement in hybrid filter



Depending on the period of the camera motion, tracking performance varies. When the translational or rotational camera motion velocity increases, the RMSE value between the real map and estimated map increases. In Figure 4.22, the changes in the RMSE values can be found according to the translational and rotational period in this case.

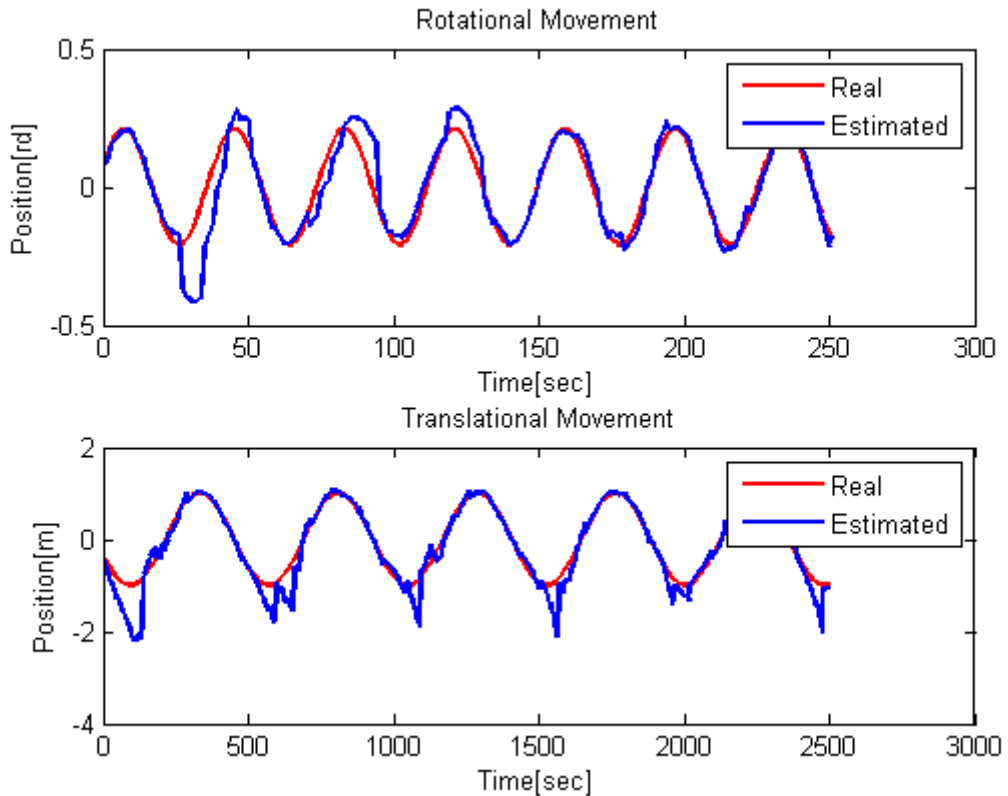
Figure 4.22: Change in RMSE of the position and orientation of the camera with rotational and translational period of the camera when both angular velocity and linear acceleration data are used as measurement in hybrid filter



4.4.2 Angular Velocity Data Used As Control Input, Linear Acceleration Data Used As Measurement

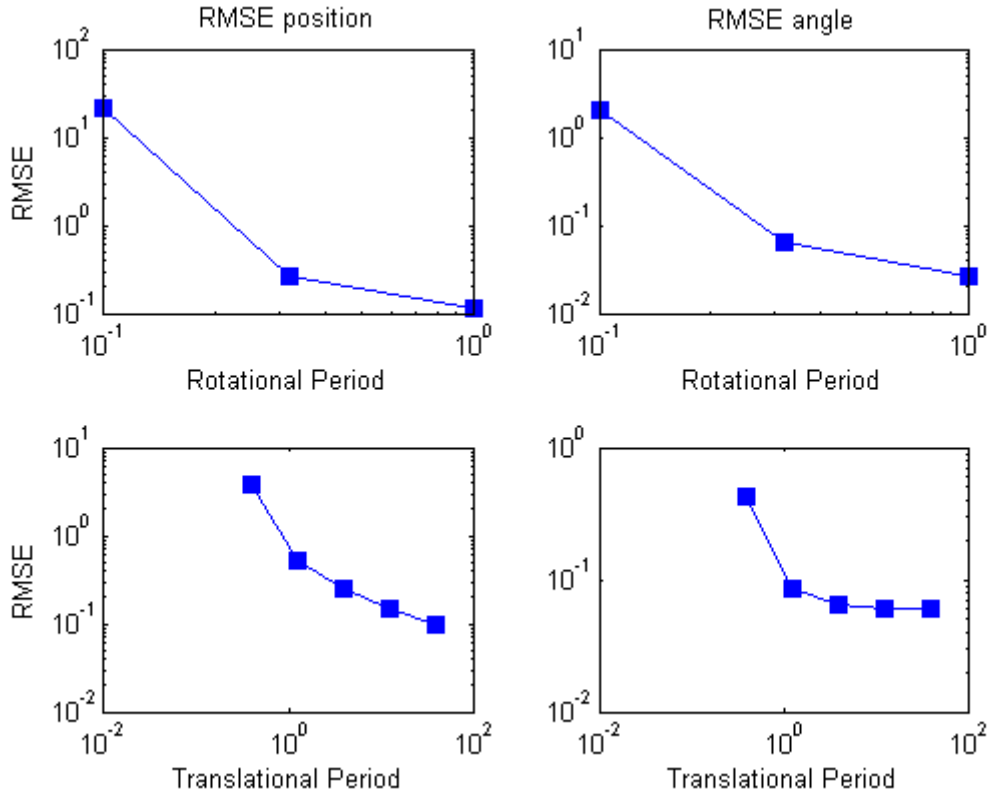
In Figure 4.23, the comparison of the real and estimated translational and rotational movement of the camera with $10^{0.6}$ s translational period and $10^{-0.5}$ s rotational period can be seen when the linear acceleration data and camera data are used as measurement and the angular velocity data is used as control input. Using angular velocity as control input instead of measurement and accelerometer as measurement reduces the tracking performance of the camera. As can be seen in Figure 4.23, the difference between the true and estimated rotational and translational motion of the camera is bigger than the case both accelerometer and gyroscope are used as measurement.

Figure 4.23: Estimated and true motion of the camera when angular velocity data is used as control input, linear acceleration data is used as measurement in hybrid filter



In Figure 4.24, the RMSE values change can be seen according to translational and rotational period in this case.

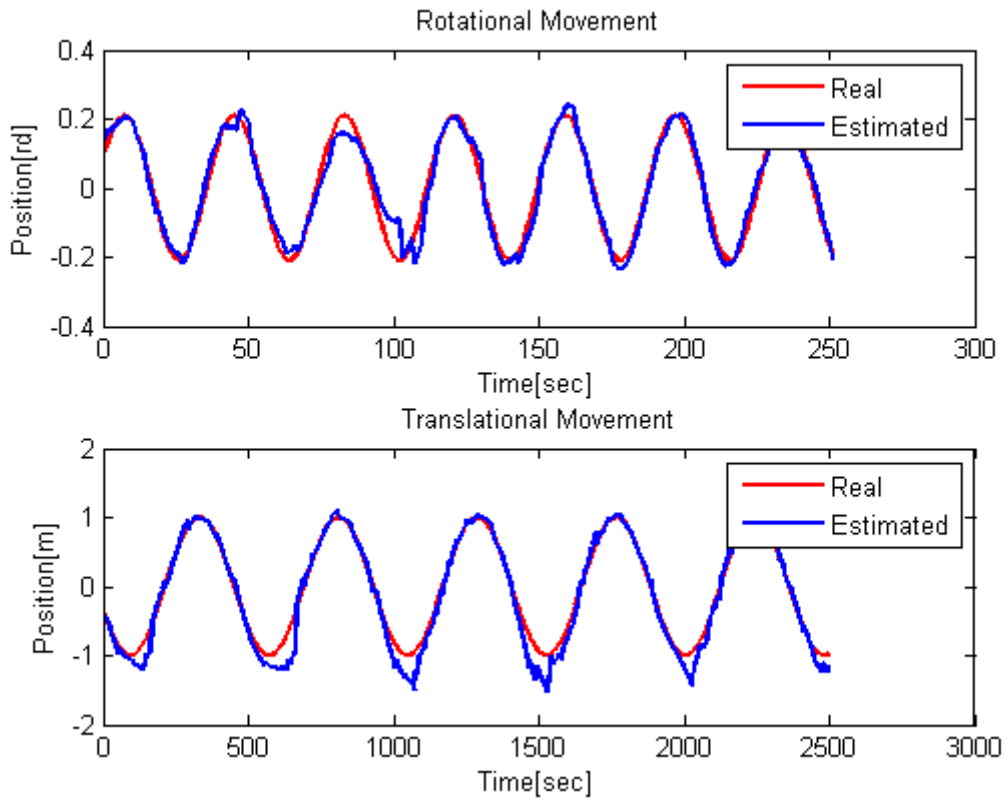
Figure 4.24: Change in RMSE of the position and orientation of the camera with rotational and translational period of the camera when angular velocity data is used as control input, linear acceleration data is used as measurement in hybrid filter



4.4.3 Angular Velocity Data Used As Measurement, Linear Acceleration Data Used As Control Input

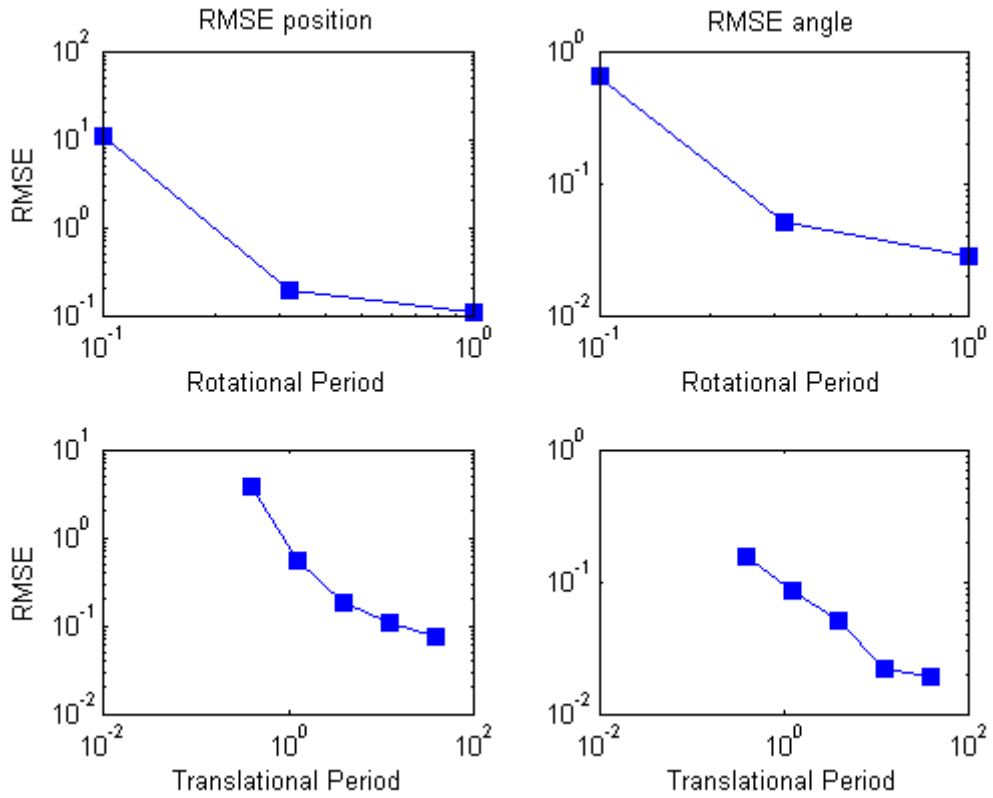
In Figure 4.25, the comparison of the real and estimated translational and rotational movement of the camera with $10^{0.6}$ s translational period and $10^{-0.5}$ s rotational period can be found when the angular velocity data and the camera data are used as measurement and the linear acceleration data is used as control input. When the angular velocity is used as measurement and the accelerometer is used as control input, the tracking performance of the camera does not change so much when compares the tracking performance of the case that both inertial sensors are used as measurement.

Figure 4.25: Estimated and true motion of the camera when angular velocity data is used as measurement, linear acceleration data is used as control input in hybrid filter



In Figure 4.26, the changes in the RMSE values can be seen according to translational and rotational period in this case.

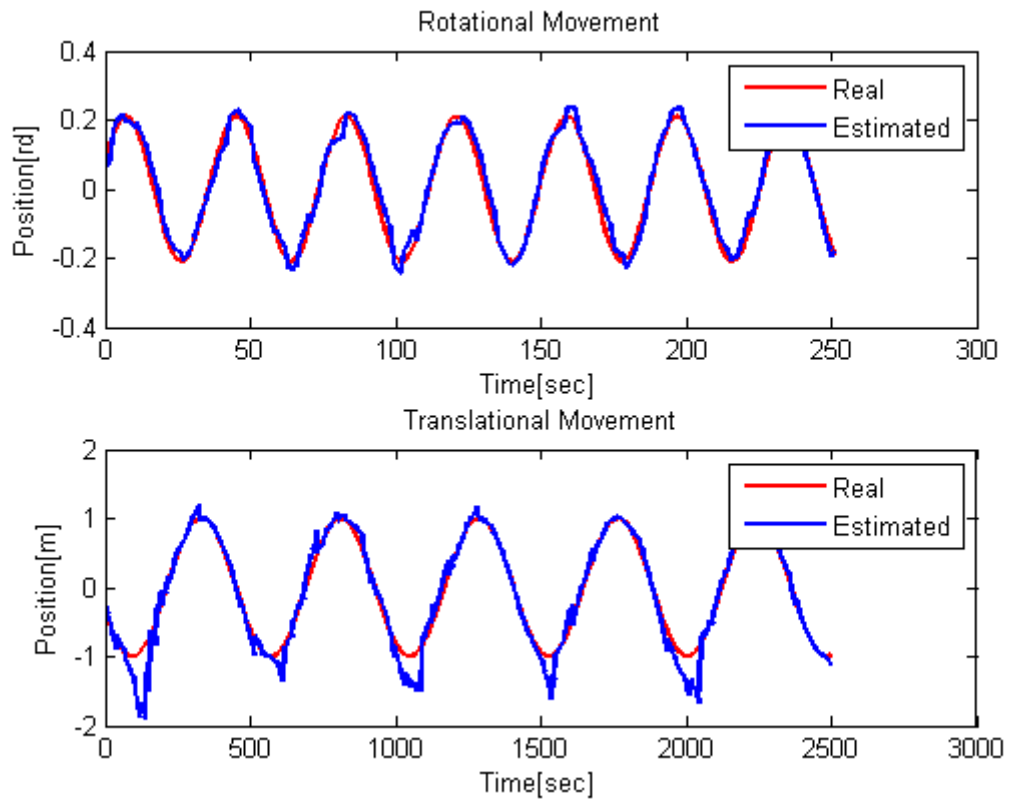
Figure 4.26: Change in RMSE of the position and orientation of the camera with rotational and translational period of the camera when angular velocity data is used as measurement, linear acceleration data is used as control input in hybrid filter



4.4.4 Both Angular Velocity And Linear Acceleration Data Used As Control Input

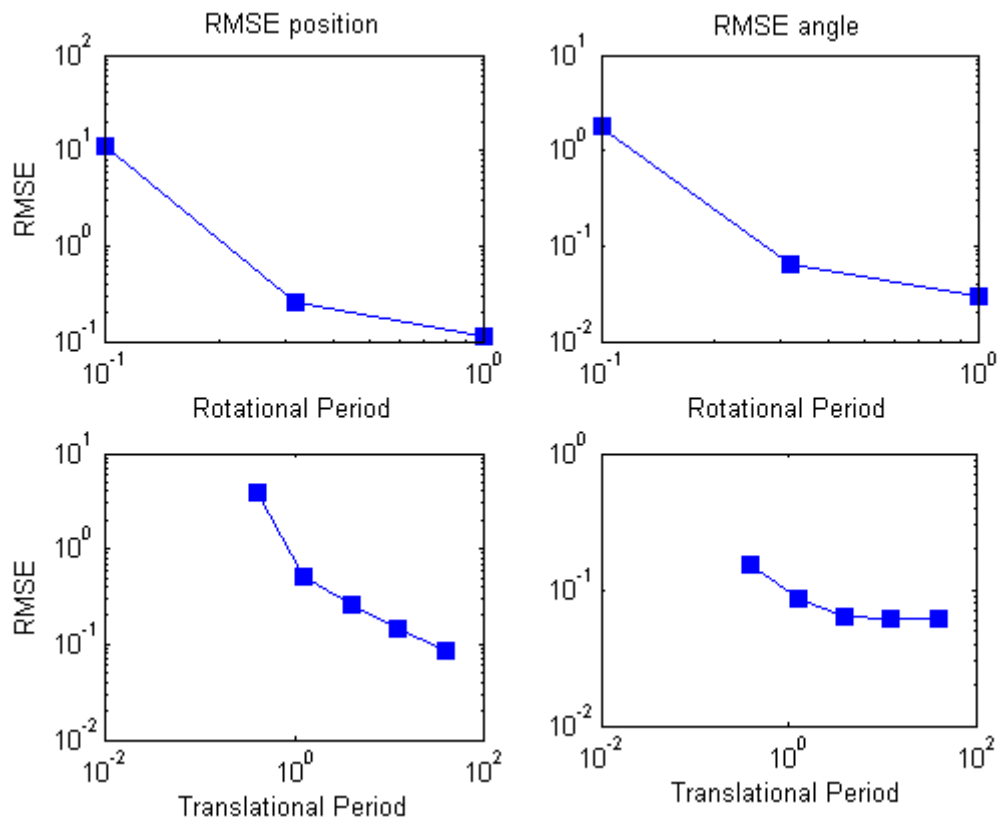
In Figure 4.27, the comparison of the real and estimated translational and rotational movement of the camera with $10^{0.6}$ s translational period and $10^{-0.5}$ s rotational period can be found when all inertial sensor data is used as control input and only camera data are used as measurement. When both inertial sensors are used as control input, tracking performance of the camera in translation decreases, but the tracking performance of the camera in rotation is not affected significantly.

Figure 4.27: Estimated and true motion of the camera when both angular velocity and linear acceleration data are used as control input in hybrid filter



In Figure 4.28, the changes in the RMSE values can be found according to translational and rotational period in this case.

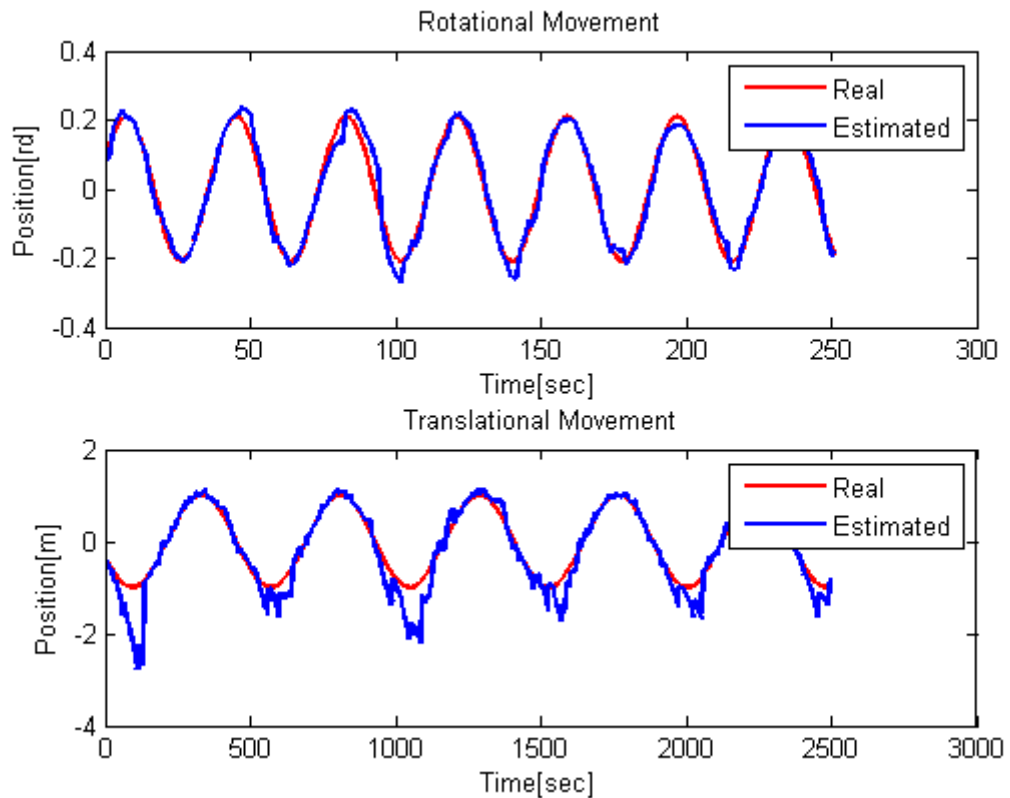
Figure 4.28: Change in RMSE of the position and orientation of the camera with rotational and translational period of the camera when both angular velocity and linear acceleration data are used as control input in hybrid filter



4.4.5 Angular Velocity Data Used As Measurement

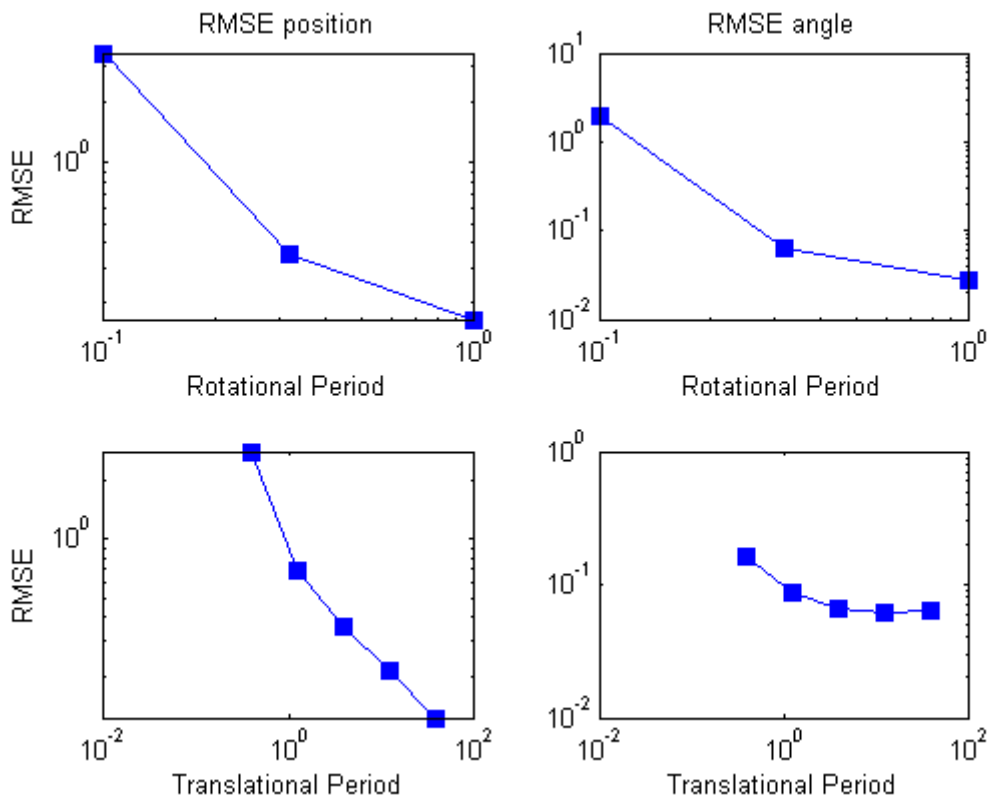
In Figure 4.29, the comparison of the real and estimated translational and rotational movement of the camera with $10^{0.6}$ s translational period and $10^{-0.5}$ s rotational period can be found when the angular velocity data and the camera data are used as measurement and the linear acceleration data is not used anymore. When we do not use accelerometer and use only gyroscope as measurement with camera, tracking performance in translation poor and tracking performance in rotation is good.

Figure 4.29: Estimated and true motion of the camera when angular velocity data is used as measurement



In Figure 4.30, the changes in the RMSE values can be found according to translational and rotational period in this case.

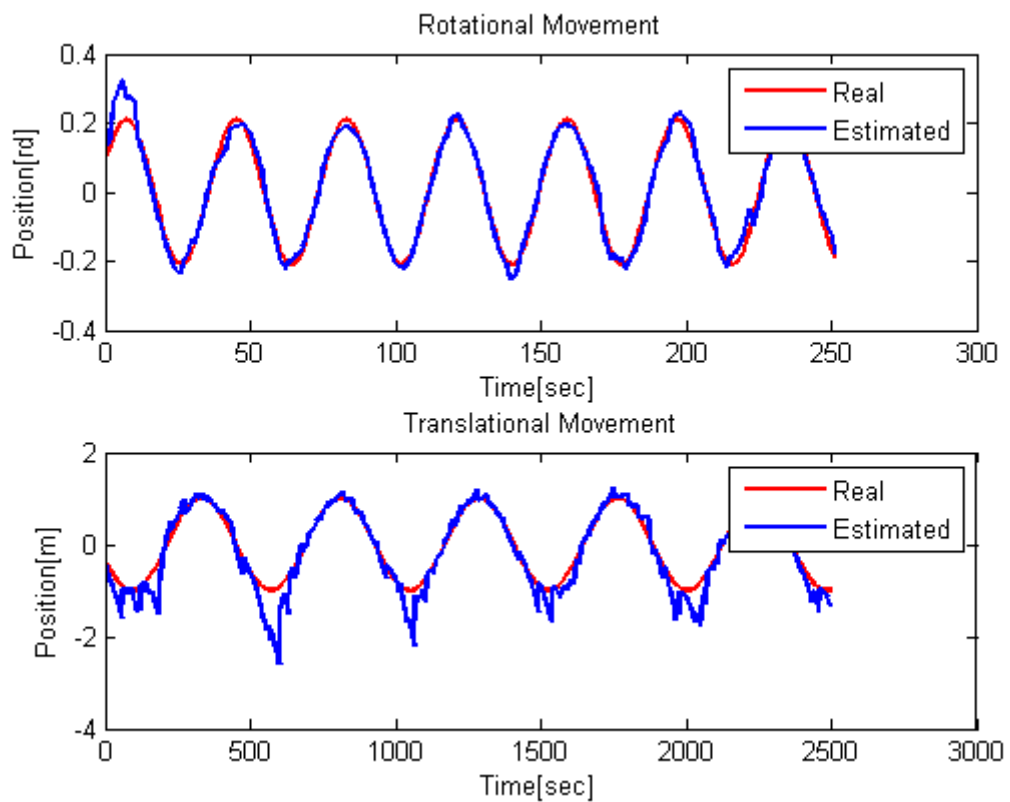
Figure 4.30: Change in RMSE of the position and orientation of the camera with rotational and translational period of the camera when angular velocity data is used as measurement



4.4.6 Angular Velocity Data Used As Control Input

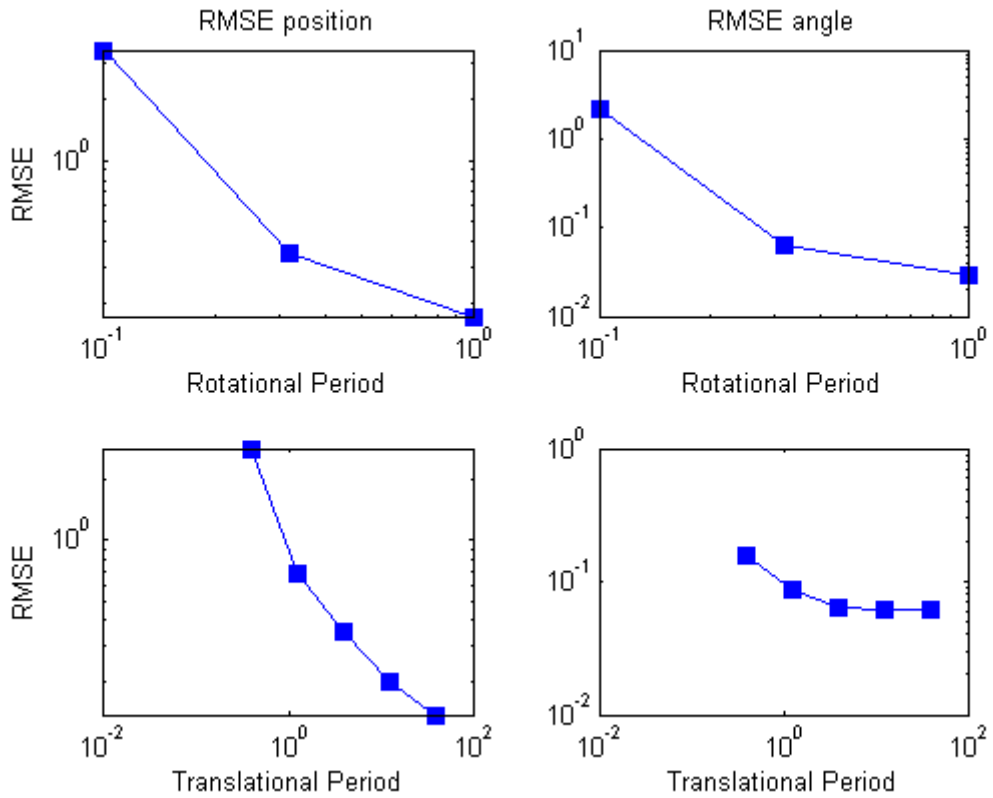
In Figure 4.31, the comparison of the real and estimated translational and rotational movement of the camera with $10^{0.6}$ s translational period and $10^{-0.5}$ s rotational period can be found when the angular velocity data is used as control input and the camera data is used as measurement and the linear acceleration data is not used anymore. Using only angular velocity as control input, tracking performance in translation is bad due to lack of accelerometer. On the other hand using gyroscope as control input instead of measurement does not change the tracking performance in rotation.

Figure 4.31: Estimated and true motion of the camera when angular velocity data is used as control input in hybrid filter



In Figure 4.32, the changes in the RMSE values can be found according to translational and rotational period in this case.

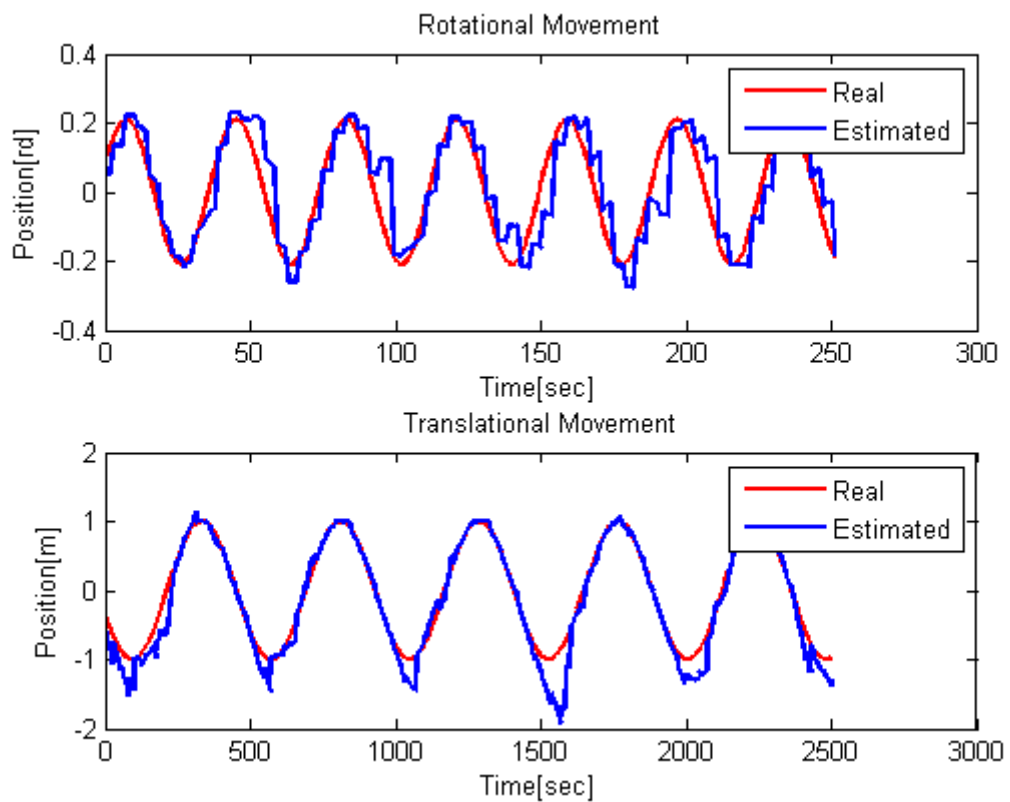
Figure 4.32: Change in RMSE of the position and orientation of the camera with rotational and translational period of the camera when angular velocity data is used as control input in hybrid filter



4.4.7 Linear Acceleration Data Used As Measurement

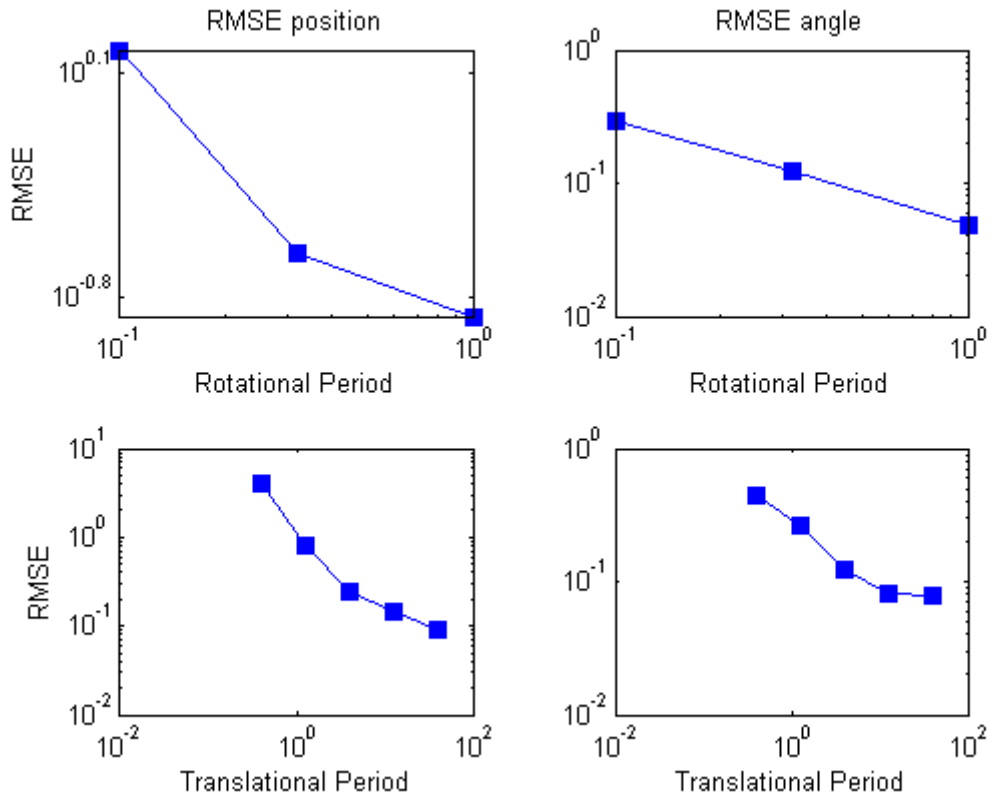
In Figure 4.33, the comparison of the real and estimated translational and rotational movement of the camera with $10^{0.6}$ s translational period and $10^{-0.5}$ s rotational period can be found when the linear acceleration data and the camera data are used as measurement and the angular velocity data is not used anymore. Using only accelerometer as measurement and not using gyroscope reduces the tracking performance of the camera in rotation. The tracking performance of the camera in translation reduces but not significantly.

Figure 4.33: Estimated and true motion of the camera when linear acceleration data is used as measurement in hybrid filter



In Figure 4.34, the changes in the RMSE values can be found according to translational and rotational period in this case.

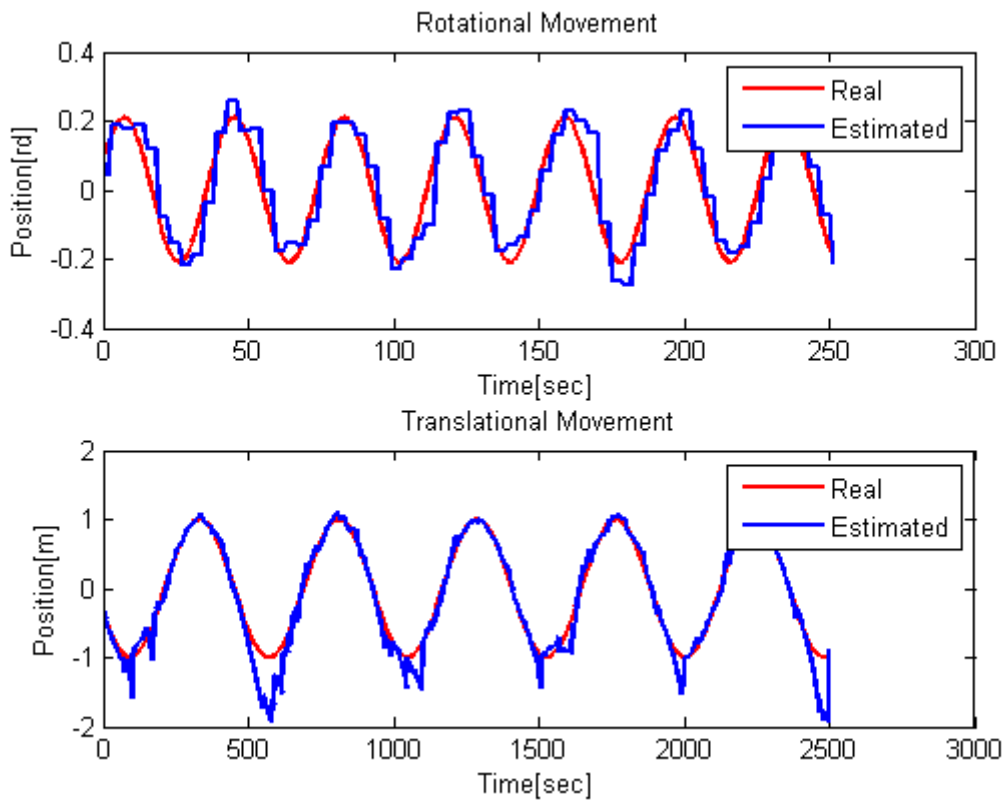
Figure 4.34: Change in RMSE of the position and orientation of the camera with rotational and translational period of the camera when linear acceleration data is used as measurement in hybrid filter



4.4.8 Linear Acceleration Data Used As Control Input

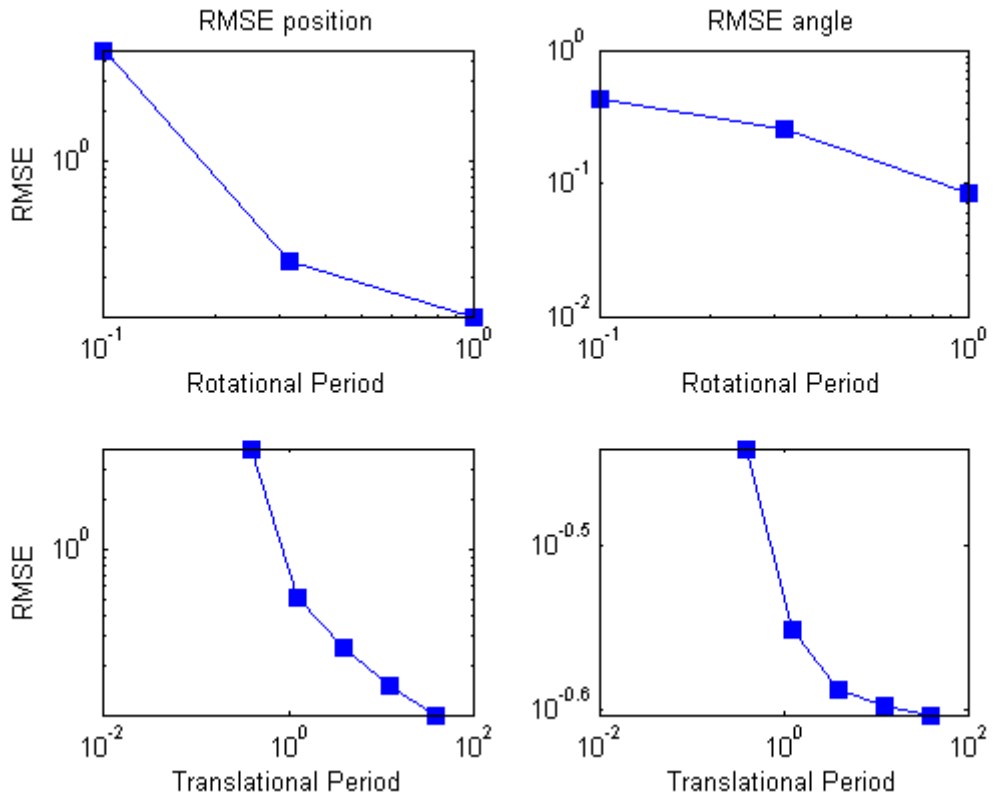
In Figure 4.35, the comparison of the real and estimated translational and rotational movement of the camera with $10^{0.6}$ s translational period and $10^{-0.5}$ s rotational period can be found when the linear acceleration data is used as control input, the camera data is used as measurement and the angular velocity is not used anymore. Using accelerometer as control input instead of measurement without gyroscope does not change the tracking performance in translation significantly. However the performance of the tracking in rotation is still poor due to lack of gyroscope.

Figure 4.35: Estimated and true motion of the camera when linear acceleration data is used as control input in hybrid filter



In Figure 4.36, the changes in the RMSE values can be found according to translational and rotational period in this case.

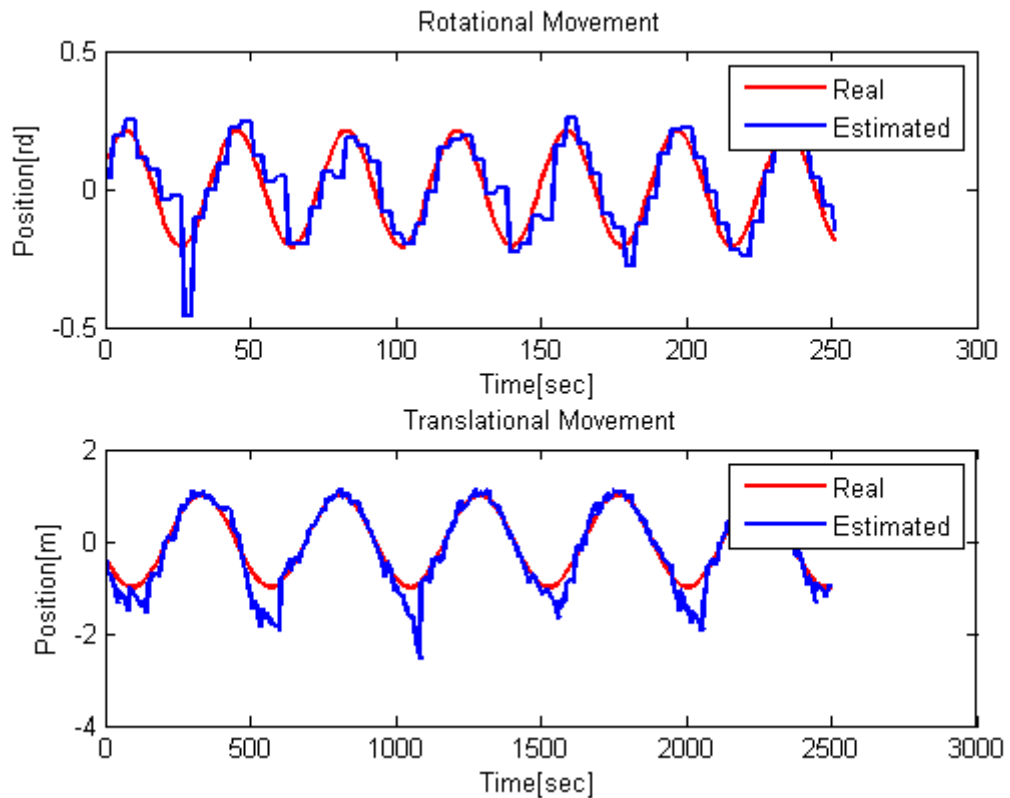
Figure 4.36: Change in RMSE of the position and orientation of the camera with rotational and translational period of the camera when linear acceleration data is used as control input in hybrid filter



4.4.9 Both Angular Velocity And Linear Acceleration Data Not Used

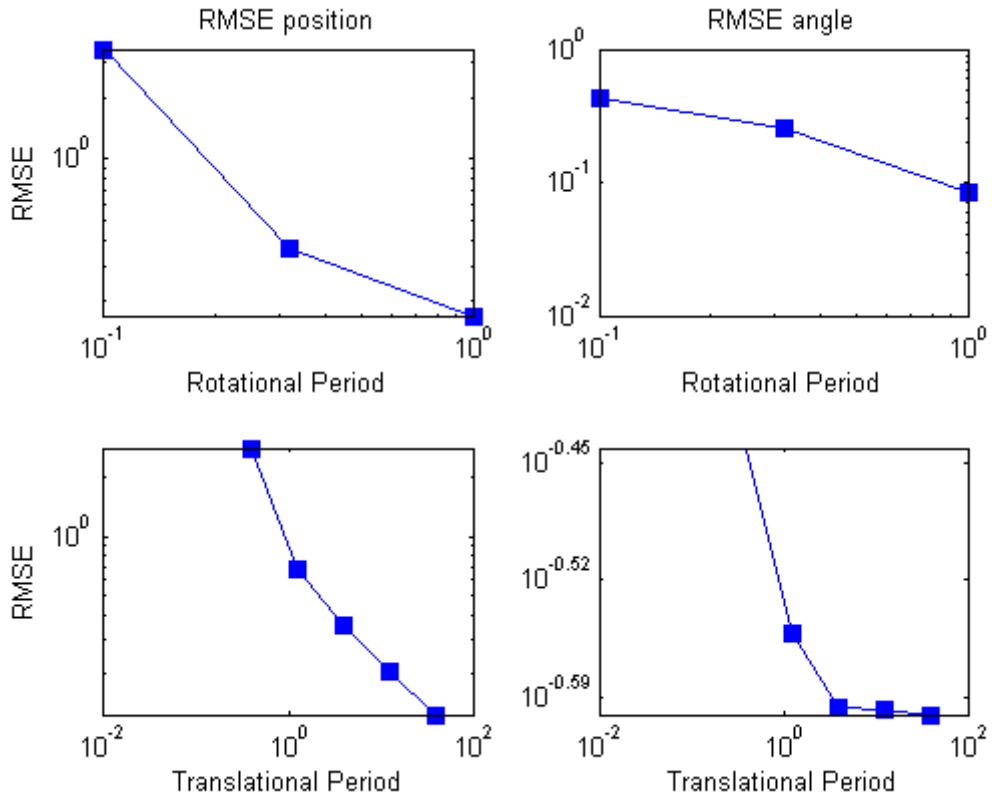
In Figure 4.37, the comparison of the real and estimated translational and rotational movement of the camera with $10^{0.6}$ s translational period and $10^{-0.5}$ s rotational period can be found when the only camera data is used as measurement, the angular velocity data and linear acceleration data are not used anymore. Using only camera for tracking gives the worst result in both rotation and translation. Difference between the true and estimated motion is high in Figure 4.37.

Figure 4.37: Estimated and true motion of the camera when both angular velocity and linear acceleration data are not used in hybrid filter



In Figure 4.38, the changes in the RMSE values can be found according to translational and rotational period in this case.

Figure 4.38: Change in RMSE of the position and orientation of the camera with rotational and translational period of the camera when both angular velocity and linear acceleration data are not used in hybrid filter



4.5 COMPARISON OF EKF AND HYBRID FILTER

Both EKF and Hybrid filters are Bayesian filters. The only difference of these two filters is the usage of the camera measurements. EKF uses the feature points on image plane in filter directly. On the other hand, Hybrid filter uses the structure (pose and angle) obtained from structure from motion by using feature points. In Table 4.1, the difference of two filters, and their RMSE values calculated by using estimated and the real 3D position and angle data.

Furthermore, the usage of the inertial sensors effects can be found in Table 4.1. In the table, both pose and angle RMSE values of the each dimension x, y, z are given. Also, Both EKF and Hybrid filter solutions for 9 different cases are given in Table 4.1.

Table 4.1: RMSE values from EKF and hybrid filter simulations

filter	acc	gyr	cam	x	y	z	θ_x	θ_y	θ_z
EKF	m	m	m	0.04445	0.04523	0.02722	0.01365	0.01393	0.01683
	ci	m	m	0.04847	0.04924	0.03024	0.01470	0.01503	0.01692
	m	ci	m	0.04473	0.04489	0.02683	0.01295	0.01330	0.01616
	ci	ci	m	0.04896	0.04913	0.03020	0.01477	0.01517	0.01640
	x	m	m	0.05516	0.05548	0.03701	0.01609	0.01654	0.01701
	x	ci	m	0.05486	0.05539	0.03710	0.01620	0.01656	0.01652
	m	x	m	0.04468	0.04555	0.02829	0.01523	0.01566	0.04697
	ci	x	m	0.05044	0.05021	0.04871	0.04838	0.04855	0.04973
	x	x	m	0.05599	0.05578	0.05412	0.04897	0.04908	0.04950
HYBRID	m	m	m	0.03564	0.03697	0.16186	0.03062	0.03513	0.02165
	ci	m	m	0.04158	0.04102	0.24941	0.03750	0.04668	0.02260
	m	ci	m	0.03635	0.03676	0.17335	0.02957	0.03510	0.02161
	ci	ci	m	0.04128	0.04057	0.25836	0.03874	0.04606	0.02283
	x	m	m	0.05597	0.05503	0.34731	0.03930	0.04653	0.02823
	x	ci	m	0.05710	0.05535	0.34651	0.03737	0.04583	0.02276
	m	x	m	0.03975	0.03714	0.23360	0.06484	0.07679	0.07365
	ci	x	m	0.04716	0.04222	0.24839	0.15569	0.18127	0.097457
	x	x	m	0.05660	0.05575	0.34727	0.15045	0.18183	0.09344

In the table, *acc*, *gyr*, and *cam* denotes the accelerometer, gyroscope, and camera, respectively. The expression under these three columns *m*, *ci*, and *x* denotes the data used as measurement, data used as control input, and data not used, respectively.

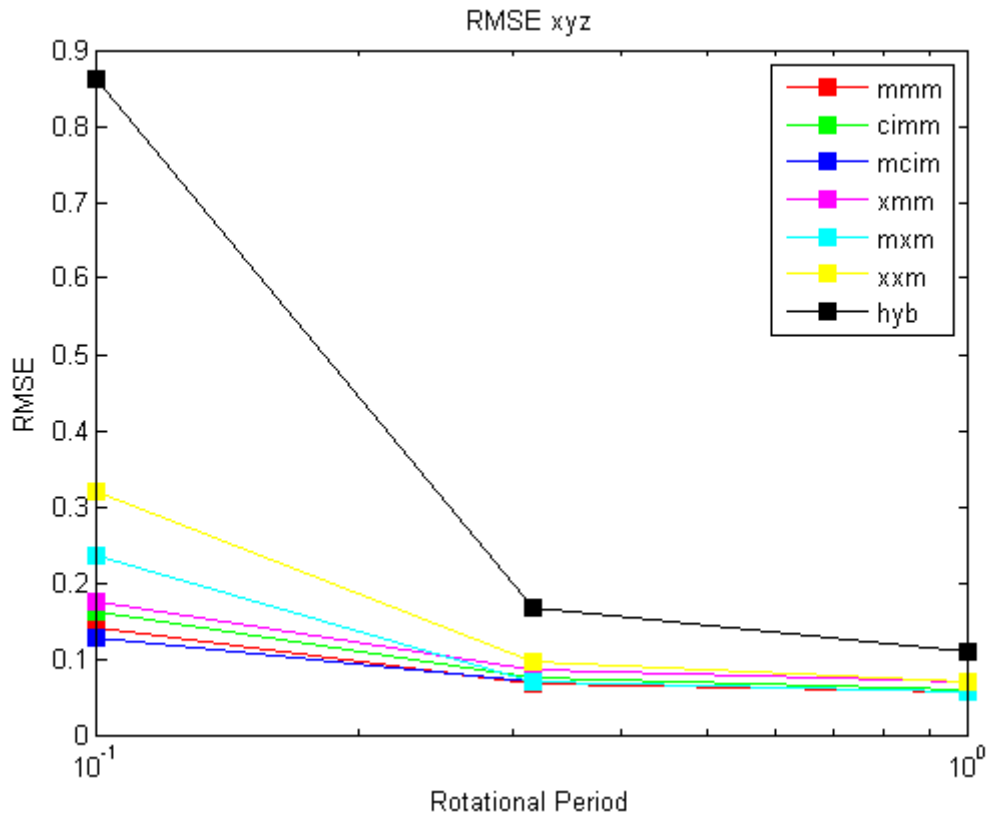
The simulations are run for 300 seconds and 5 times. The amount of noise is changed randomly each time. So, the standard deviations of these 5 simulations are shown in Table 4.2.

Table 4.2: Standard deviation values from EKF and hybrid filter simulations

filter	acc	gyr	cam	x	y	z	θ_x	θ_y	θ_z
EKF	m	m	m	0.00104	0.00057	0.00078	0.00010	0.00014	0.00003
	ci	m	m	0.00130	0.00134	0.00037	0.00042	0.00062	0.00056
	m	ci	m	0.00152	0.00131	0.00036	0.00034	0.00030	0.00012
	ci	ci	m	0.00095	0.00095	0.00031	0.00021	0.00021	0.00019
	x	m	m	0.00093	0.00094	0.00037	0.00021	0.00019	0.00026
	x	ci	m	0.00057	0.00113	0.00045	0.00018	0.00006	0.00013
	m	x	m	0.00052	0.00126	0.00060	0.00026	0.00013	0.00010
	ci	x	m	0.00124	0.00176	0.00025	0.00017	0.00011	0.00011
	x	x	m	0.00077	0.00131	0.00034	0.00026	0.00023	0.00008
HYBRID	m	m	m	0.00171	0.00200	0.01468	0.00091	0.00050	0.00042
	ci	m	m	0.00245	0.00288	0.02273	0.00070	0.00063	0.00029
	m	ci	m	0.00133	0.00228	0.00763	0.00030	0.00098	0.00020
	ci	ci	m	0.00185	0.00231	0.00688	0.00088	0.00074	0.00041
	x	m	m	0.00093	0.00258	0.01250	0.00268	0.00177	0.00946
	x	ci	m	0.00196	0.00170	0.01874	0.00031	0.00060	0.00039
	m	x	m	0.00115	0.00101	0.01069	0.00376	0.00196	0.00155
	ci	x	m	0.00125	0.00230	0.02130	0.00726	0.00301	0.00560
	x	x	m	0.00319	0.00158	0.01129	0.00364	0.00252	0.00340

Some EKF cases and Hybrid filter comparisons in different periods can be found in the figures below. In Figure 4.39, RMSE values are calculated by using estimated and the real 3D position data for Case 1, 2, 3, 5, 7, 9 for EKF and Case 1 for Hybrid filter in different rotational period.

Figure 4.39: RMSE (position) comparison of different usage of EKF and hybrid



In Figure 4.40, RMSE values are calculated by using the estimated and real 3D position data for Case 1, 2, 3, 5, 7, 9 for EKF and Case 1 for Hybrid filter in different rotational period.

Figure 4.40: RMSE (orientation) comparison of different usage of EKF and hybrid

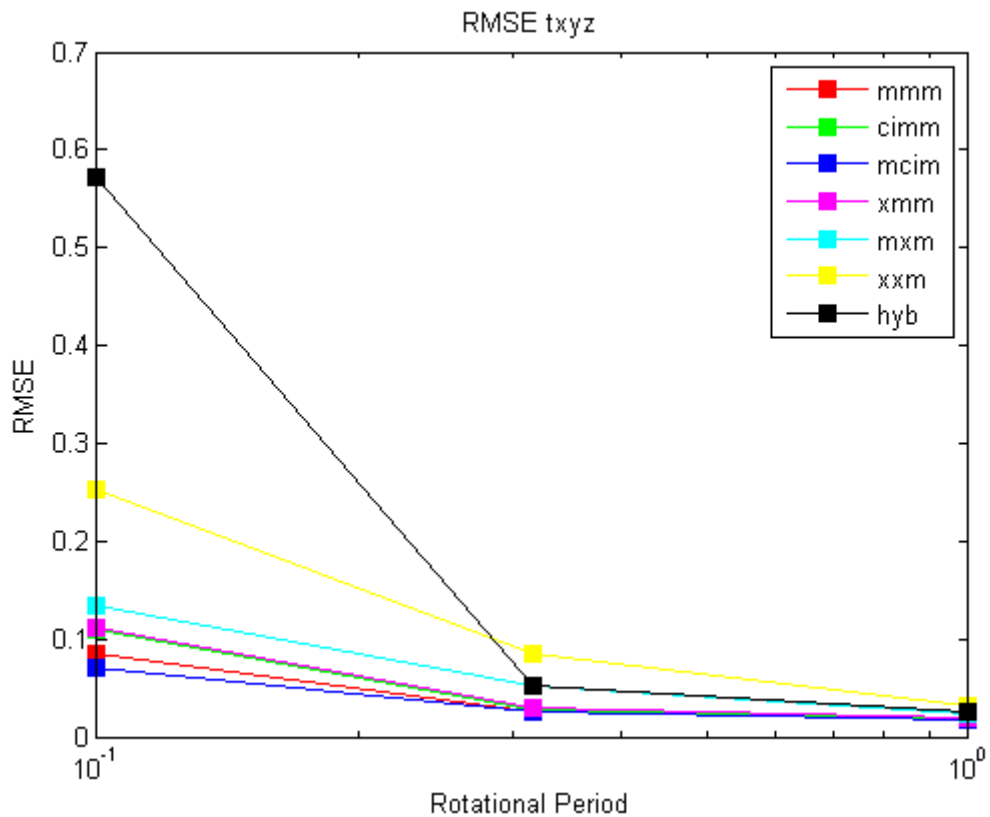
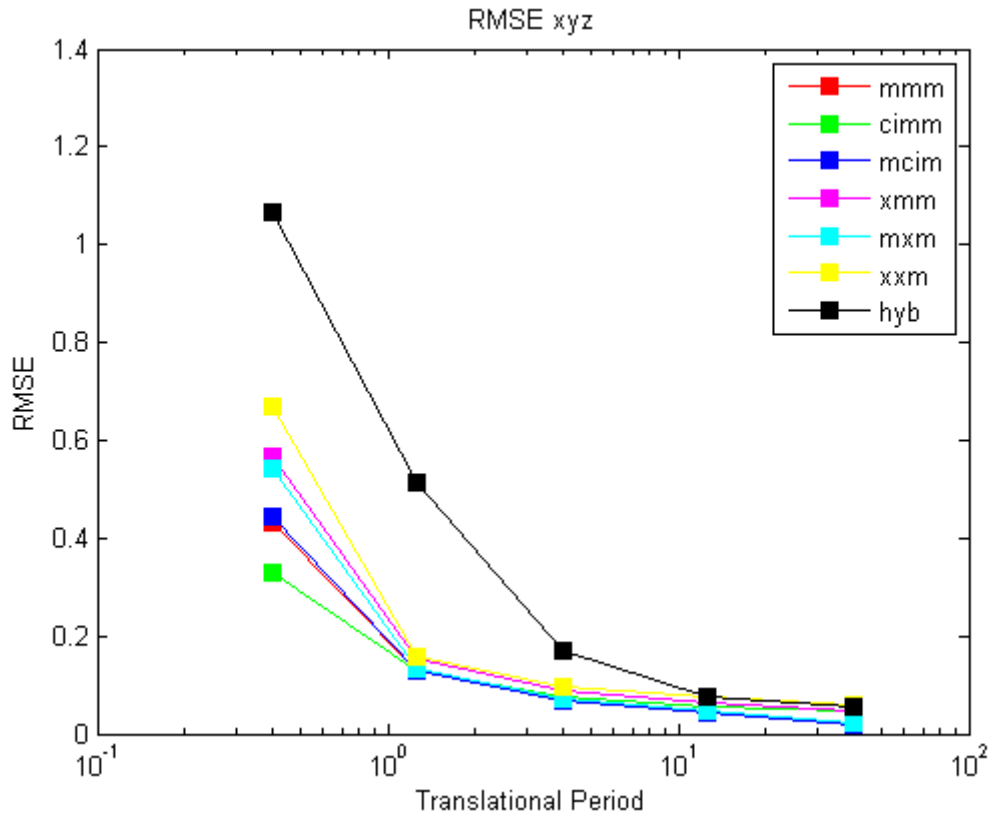


Figure 4.39 and 4.40 show that when the period of the camera in rotation is low, using inertial sensors as control input gives slightly better results, also using inertial sensors for tracking as measurement or control input improves the tracking performance. However, when the period increases, the effects of the inertial sensors decrease because as we said before the camera gives better tracking results in slow motion (blue, red and green lines). The camera measurement will be enough when the period of the camera motion. Moreover, hybrid filter does not give good results as EKF at high speed, but when the motion decreases hybrid filter performance is close to EKF performance (black line).

In Figure 4.41, RMSE values are calculated by using the estimated and real 3D position data for Case 1, 2, 3, 5, 7, 9 for EKF and Case 1 for Hybrid filter in different translational period.

Figure 4.41: RMSE (position) comparison of different usage of EKF and hybrid



In Figure 4.42, RMSE values are calculated by using the estimated and real 3D angle data for Case 1, 2, 3, 5, 7, 9 for EKF and Case 1 for Hybrid filter in different translational period.

Figure 4.42: RMSE (orientation) comparison of different usage of EKF and hybrid

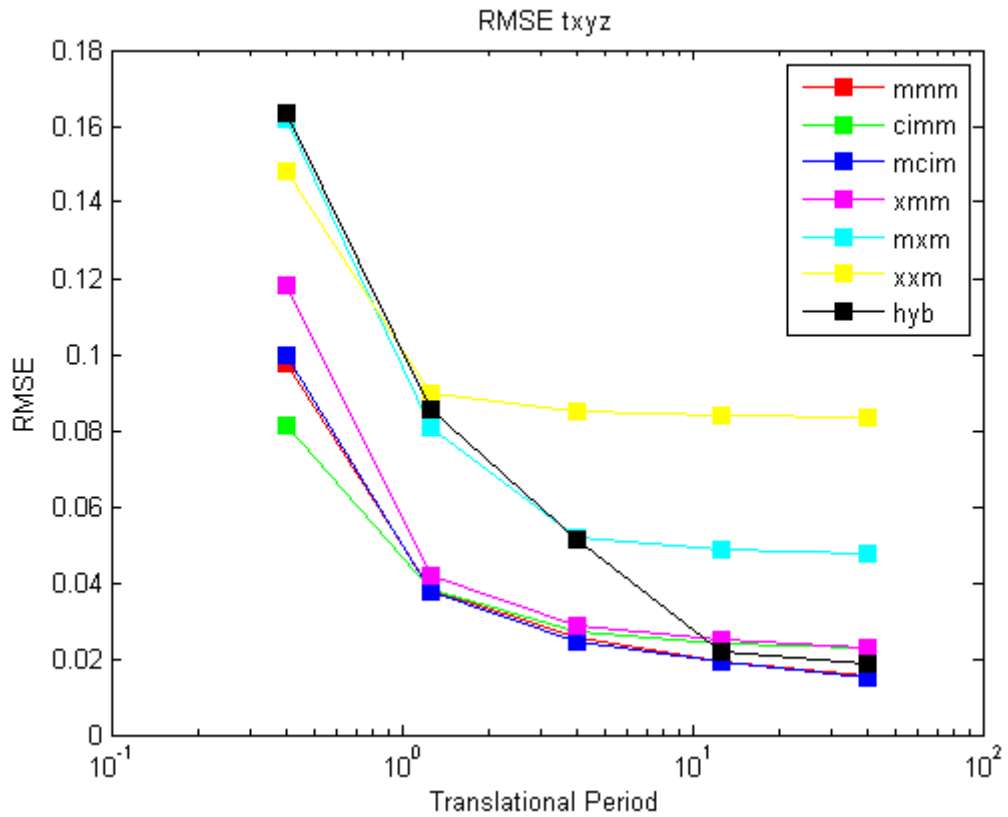
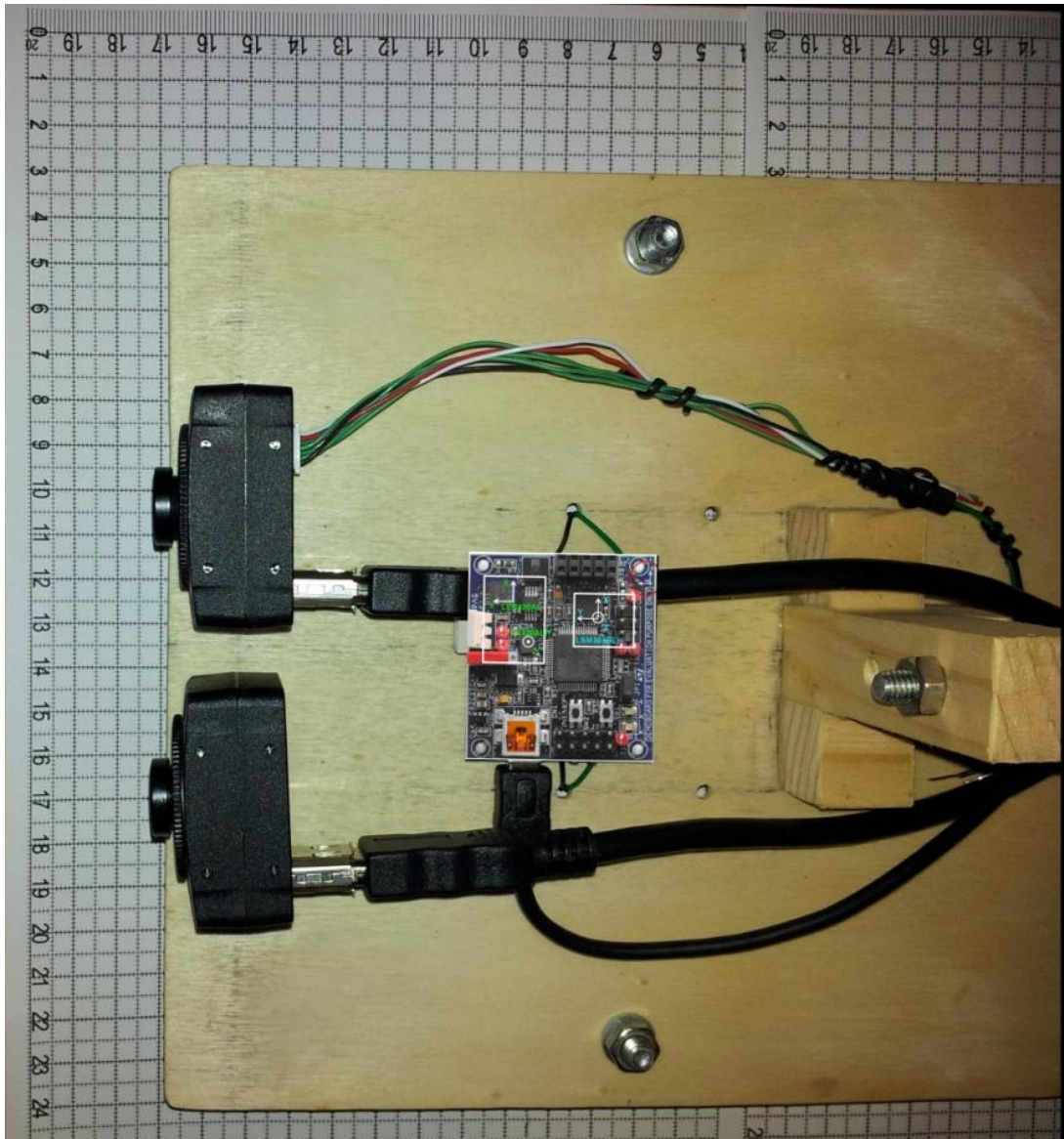


Figure 4.41 and 4.42 show that when the period of the camera in translation is low, using inertial sensors as control input gives slightly better results. When the period increases, the tracking performance does not increase significantly for the cases not using gyroscope (yellow and cyan lines). Not using accelerometer or using as measurement or control input to track the rotational motion of the camera does not affect the tracking performance (pink, green and red lines). Moreover, hybrid filter does not give good results as EKF at high speed, but when the motion decreases hybrid filter performance is close to EKF performance (black line).

4.6 ACTUAL EXPERIMENT SETUP

In this section, the real world simulation platform and the performance evaluation of the algorithms are provided. For platform, we prepared a setup including a camera, two gyroscopes, and an accelerometer (Figure 4.43).

Figure 4.43: Simulation setup



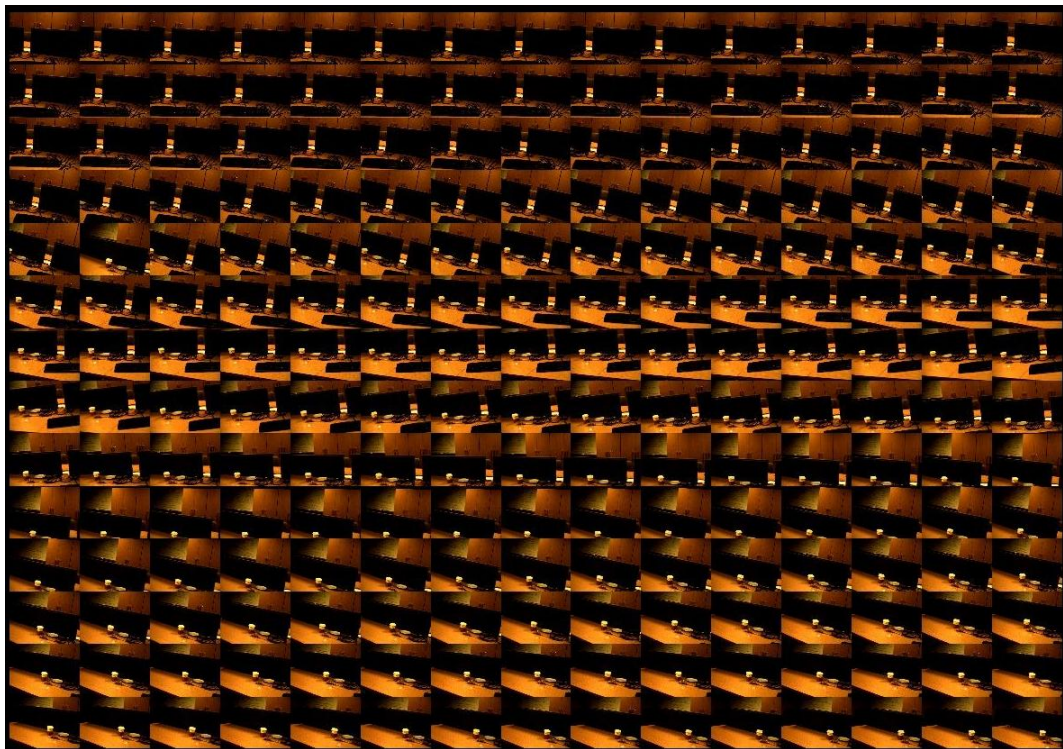
With this setup, we moved in the lab and captured a video and sensor data from accelerometer and gyroscope. There are two cameras on the image, but we use only one

of them for now. However; theboth of the camera can be used for the next step of the project.

Camera:

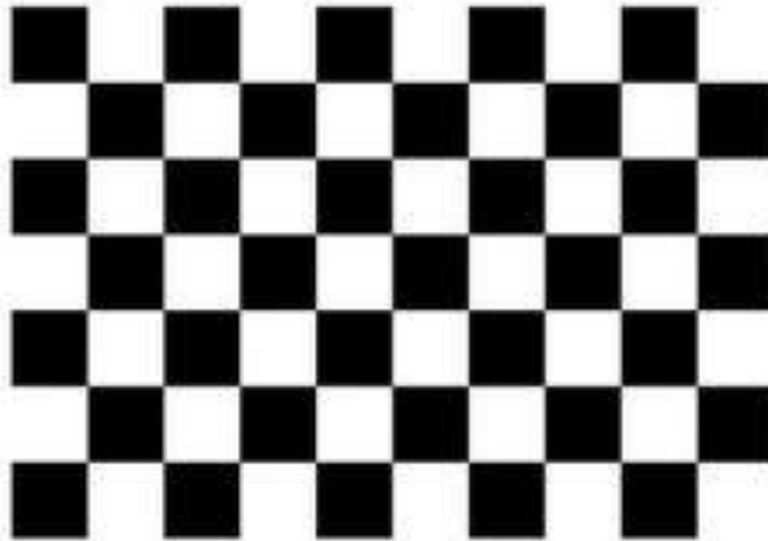
The camera on the setup has 640x480 resolution and capture RGB value image in 30Hz. To get information about 2D/3D correspondence, 3D map of the lab should be found in offline. For this purpose, we find the feature points of each frame captured by the camera using SIFT. Then Bundler Adjustment algorithm is used to find the 3D information of the feature points. Since SIFT finds many feature points, we eliminated some of them. Then, we found the 3D map of the lab before starting to track (Figure 4.44).

Figure 4.44: Image sequences used to find 3D map of the scene



Also, we found the camera intrinsic parameters by using calibration patterns (Figure 4.45).

Figure 4.45: Camera calibration pattern



Accelerometer:

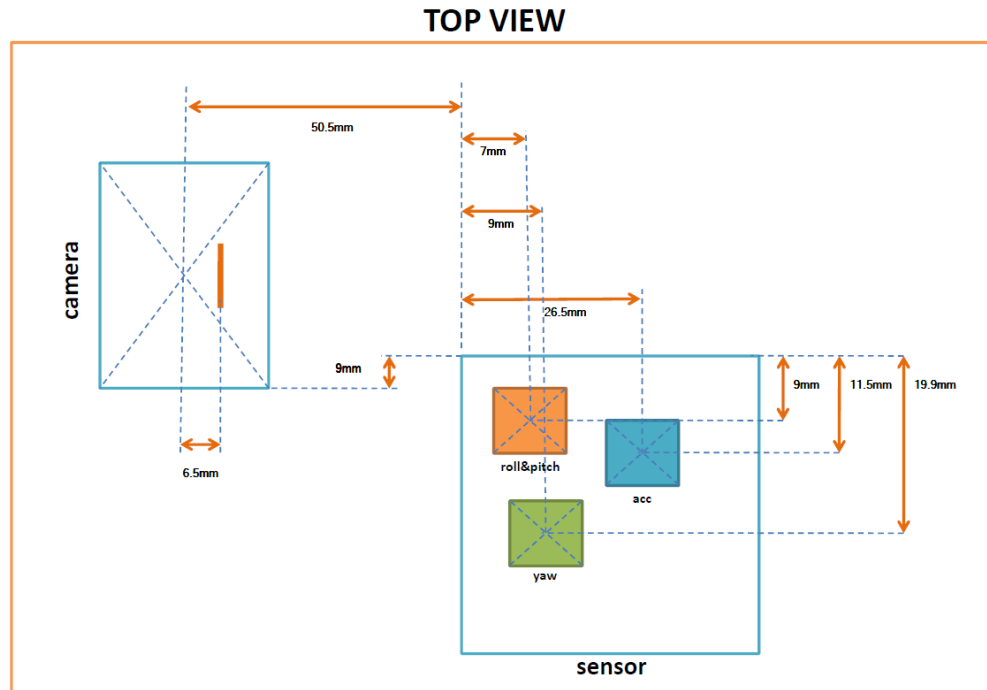
In this setup, LSM303DHL type 3D accelerometer is used. The measurement noise obtained from its data sheet. When accelerometer measures the linear acceleration, it also measures the gravity force on z direction. The gravity force of the space where camera tracking is made should be found and subtracted from the measurements of the accelerometer. To calculate the gravity force, we left the accelerometer stable and wait measurements. Because accelerometer is not moved, the measurement values obtained from gyroscope is related to gravity force. We made 100 trials and calculate the average of all measurement and gravity force is calculated. In the lab, we found the gravity force to be $[0.03238 \quad -0.12443 \quad -10.02318]$.

Gyroscope:

We used two gyroscopes in the setup. LPR430AL type gyroscope obtains x and y coordinates of the rotational movement. The other gyroscope LY330ALH type measures the z coordinate of the rotational movement.

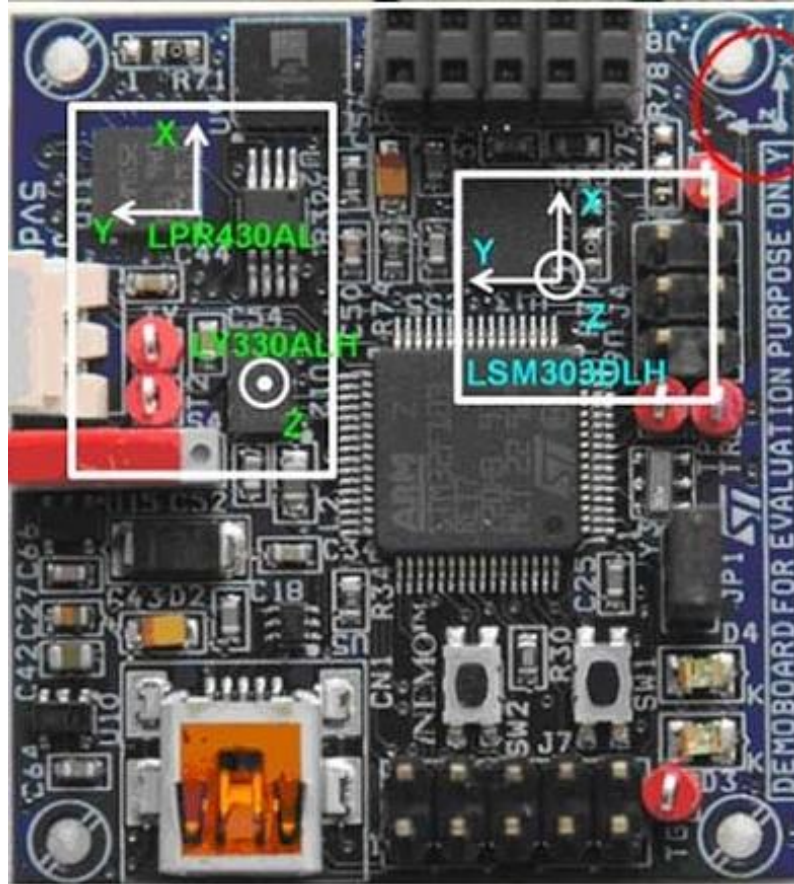
The coordinate systems of the camera, gyroscope and accelerometer are found. The positions of the inertial sensors and camera on the board are found by using metrical paper (Figure 4.46).

Figure 4.46: Positions of the inertial sensors and camera on the setup



The devices, i.e., camera, accelerometer and gyroscope on the board have their own coordinate system. To find the relationships of the coordinate system, we have to find their exact position and the difference of their centers from each others. On the datasheets of the devices, we found their coordinate systems specifications and by using metric paper to find the distances of the centers (Figure 4.47).

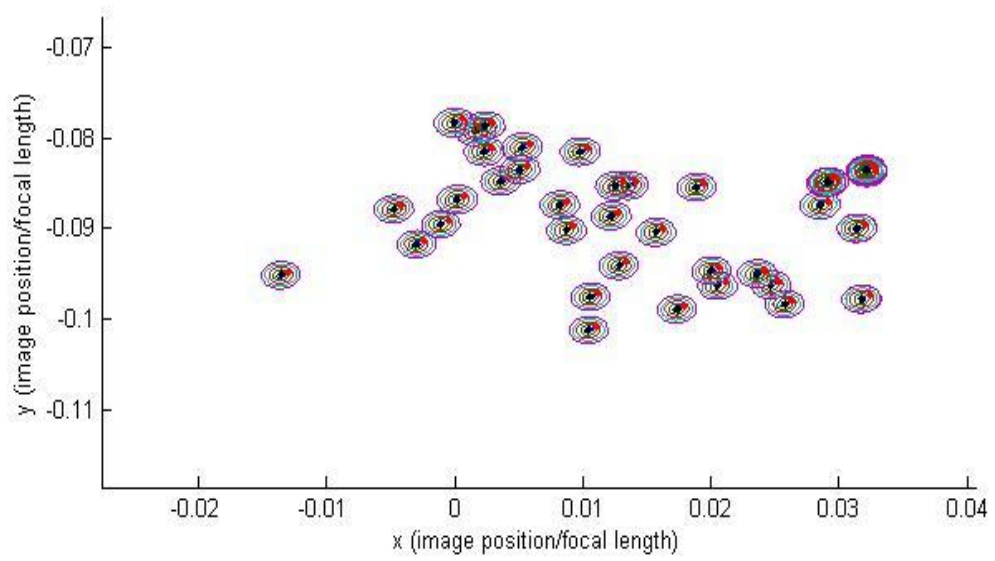
Figure 4.47: Relations of the inertial sensors' coordinate systems



4.7 ACTUAL EXPERIMENT RESULTS

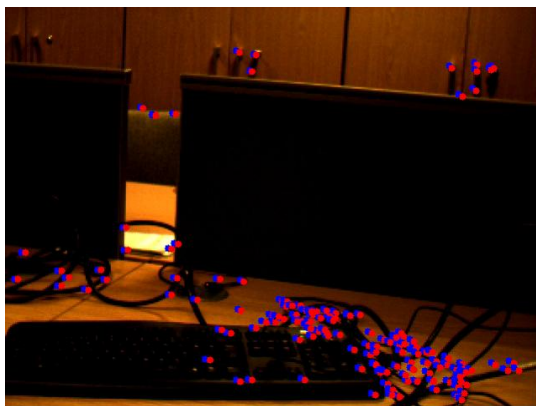
Figure 4.49 shows the matches between the feature points projected from 3D map by using estimated camera position and orientation matrix (red points) and the feature points found by using SIFT feature detection algorithms (blue points). We used the camera, accelerometer and gyroscope as measurement in this experiment. Firstly, we projected the 3D points in our map on the current image by using the predicted camera pose, and we found the feature points for current frame by using SIFT. Then, we tried to match the feature points coming from projection and SIFT. We found the best match of a feature point from projection in a elliptical region around its position (Figure 4.48).

Figure 4.48: Feature point search for best match in an ellipse

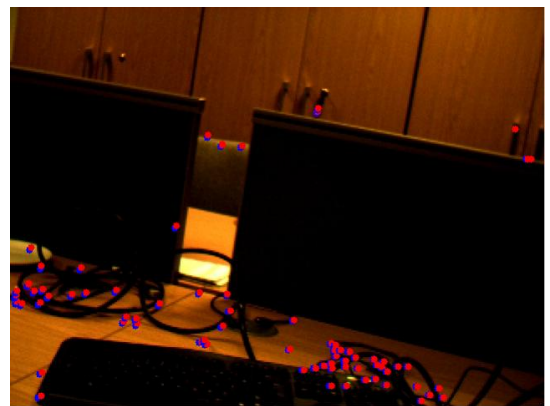


Then we updated the camera pose by using accelerometer and gyroscope measurements. In Figure 4.49, the match can be seen after the 3D points are projected on the image by using updated camera position and orientation matrix.

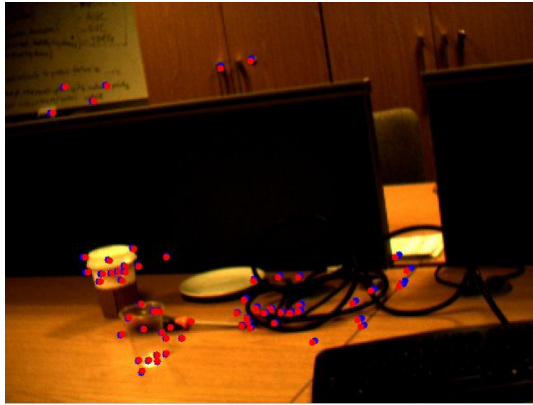
Figure 4.49: Projected and detected feature points



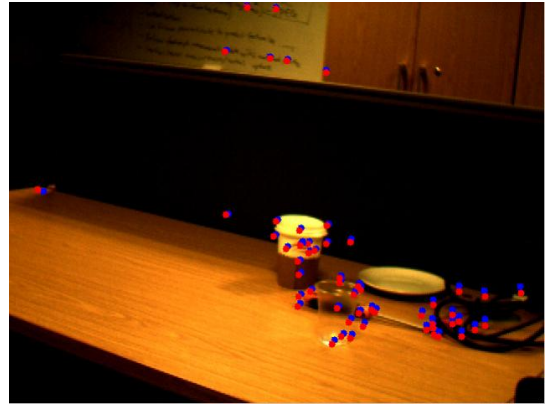
Frame 17



Frame 45



Frame 111



Frame 191

In this experiment, we used 211 figures to built 3D map by using Bundler adjustment. We found 888 3D points and we used these 3D points and their projection for camera measurement.

5. CONCLUSIONS AND FUTURE WORKS

We investigate the hybrid algorithm using inertial sensor and camera together for 3D camera tracking. When using only camera without inertial sensors, 3D tracking is hard when the velocity of the camera is high due to motion blur. When motion blur increases in an image, finding feature points and matching them becomes difficult. The inertial sensors are used to track motion when the motion of the camera is fast, because inertial sensors give more accurate results at high velocities due to their specifications. The data coming from accelerometer and gyroscope and data coming from camera video are fused in a Bayesian filter. We use two kinds of Bayesian filter in our algorithm: EKF and hybrid filter. We compare the performance of two filters. In addition to comparison of filters, we analyze how the tracking performance changes when the measurements of inertial sensors, accelerometer and gyroscope, are used as control input instead of measurements. One of the inertial sensors can also be used as control input, as both of them can be used as control input. Furthermore, we examine the tracking performance changes when the rotational and translational motion of the camera varies in time.

Simulation results show that the tracking performance using EKF gives more accurate results than using hybrid filter for all cases. It is also observed that using inertial sensors with camera in filter gives more accurate results than using only camera in filter to track the camera for both filters. Moreover, the results confirms that using accelerometer or gyroscope as control input in filter does not affect the tracking performance instead of using accelerometer or gyroscope as measurement. Using inertial sensor as control input reduce the filter computational complexity; however performance does not decrease.

The simulation results depict that when the period of the camera decreases, the tracking performance increases. Because the camera measurement will be better due to less motion blur. It is also observed that using inertial sensors as control input does not degrade the performance for all periods. Additionally, the hybrid filter performance converges the EKF performance with the increase of the camera translational period. The last observation from the figure is that using or not using gyroscope does not affect the translational performance of the tracking. As we know that gyroscope measure the

rotation of the object, so it does not affect the any translational movement. On the other, using accelerometer does not affect the performance of the camera rotational tracking. Since accelerometer measure the linear acceleration of the object, using or not using accelerometer does not affect the performance of rotational camera tracking.

Future research plan is 3D camera tracking in scene whose 3D information is not known previously. Also previously captured 3D scene information will be refined during the head tracking.

REFERENCES

Books

Fox, D. & Thrun, S. and Burgard, W., 2005. *Probabilistic robotics*. Cambridge: MA: MIT Press.

Luong, QT & Faugeras, OD., 2001. *The Geometry of Multiple Images*. Boston: MIT Press.

Periodicals

- Akatsuka, Y. & Bekey, G.A., 1998. Compensation for End to End Delays in a VR System, *Virtual Reality Annual International Symposium*. pp.156-159.
- Armesto, L. & Tornero, J. & Vincze, M., 2007. Fast Ego-Motion Estimation With Multi-Rate Fusion of Inertial and Vision. *The International Journal of Robotics Research*. **26** (6), pp.577-589.
- Azuma, R., 1997. A Survey of Augmented Reality. *Presence: Teleoperators and Virtual Environments*. **6** (4), pp.355-385.
- Azuma, R. & Bailiot, Y. & Behringer, R. & Feiner, S. & Julier, S. & MacIntyre, B., 2001. Recent Advances in Augmented Reality. *Computer Graphics and Applications, IEEE*. **21** (6), pp.34-47.
- Azuma, R. & Neely, H. & Daily, M. & Leonard, J., 2006. Performance Analysis of an Outdoor Augmented Reality Tracking System that Relies Upon a Few Mobile Beacon, *IEEE/ACM International Symposium on Mixed and Augmented Reality (ISMAR 2006)*. pp.101-104.
- Bleser, G. & Stricker, D., 2009. Advanced Tracking Through Efficient Image Processing and Visual-Inertial Sensor Fusion. *Computers & Graphics*. **33** (1), pp.59-72.
- Chai, L. & Nguyen, K. & Hoff, B. & Vincent, T., 1999. An Adaptive Estimator for Registration in Augmented Reality, *2nd IEEE and ACM International Workshop on Augmented Reality (IWAR'99)*. pp.23-32.
- Chekhlov, D. & Pupilli, M. & Mayol-Cuevas, W. & Calway, A., 2006. Real-Time and Robust Monocular Slam Using Predictive Multi-Resolution Descriptors, *Advances in Visual Computing*. pp.276-285.
- Cho Y. & Lee J. & Neumann U., 1998. A Multi-Ring Fiducial System and an Intensity-Invariant Detection Method for Scalable Augmented Reality, *Proc. Int'l Workshop Augmented Reality (IWAR)*. pp.147-165.
- Corke, P. & Lobo, J. & Dias, J., 2007. An introduction to inertial and visual sensing. *The International Journal of Robotics Research*. **26** (6), pp.519.
- Davison, A.J. & Reid, I.D. & Molton, N.D. & Stasse, O., 2007. MonoSLAM: Real-Time Single Camera SLAM. *Pattern Analysis and Machine Intelligence*. **29** (6), pp.1052-1067.
- DiVerdi, S. & Hollerer, T., 2007. GroundCam: A Tracking Modality for Mobile Mixed Reality, *Virtual Reality Conference (VR'07)*. pp.75-82.
- Durrant-Whyte, H. & Bailey, T., 2006. Simultaneous Localization & Mapping: Part I. *IEEE*. **13** (2), pp.99-110.

- Ercan, A.O. & Erdem, A.T., 2011. On Sensor Fusion for Head Tracking in Augmented Reality Applications, *American Control Conference (ACC)*. pp.1286-1291. San Francisco, CA, USA: IEEE.
- Fischler, M.A. & Bolles, R.C., 1981. Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. *Communications of the ACM*. **24** (6), pp.381-395.
- Gemeiner, P. & Einramhof, P. & Vincze, M., 2007. Simultaneous Motion and Structure Estimation by Fusion of Inertial and Vision Data. *The International Journal of Robotics Research*. **26** (6), pp.591-605.
- Guivant, J.E. & Nebot, E.M., 2001. Optimization of the Simultaneous Localization and Map-Building Algorithm for Real-Time Implementation. *Robotics and Automation*. **17** (3), pp.242-257.
- Kim, J. & Sukkarieh, S., 2007. Real-Time Implementation of Airborne Inertial-SLAM. *Robotics and Autonomous Systems*. **55** (1), pp.62-71.
- Kleeman, L., 1996. Understanding and Applying Kalman Filtering, *Proceedings of the Second Workshop on Perceptive Systems*. Perth Western Australia.
- Klein, G. & Murray, D., 2008. Improving the Agility of Keyframe-Based SLAM, *Computer Vision (ECCV 2008)*. pp.802-815.
- Klein, G. & Murray, D., 2007. Parallel Tracking and Mapping for Small AR Workspaces, *6th IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR 2007)*. pp.225-234.
- Konolige, K. & Agrawal, M., 2008. FrameSLAM: From Bundle Adjustment to Realtime Visual Mapping. *Robotics*. **24** (5), pp.1066-1077.
- Lowe, D., 2004. Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision*. **60** (2), pp.91-110.
- Newman, J. & Schall, G. & Barakonyi, I. & Schürzinger, A. & Schmalstieg, D., 2006, Wide-Area Tracking Tools for Augmented Reality. *Advances in Pervasive Computing*.
- Ohta, Y. & Sugaya, Y. & Igarashi, H. & Ohtsuki, T. & Taguchi, K., 2002. Share-Z: Client/Server Depth Sensing for See-Through Head-Mounted Displays. *Presence: Teleoperators & Virtual Environments*. **11** (2), pp.176-188.
- Papagiannakis, G. & Magnenat-Thalmann, N., 2007. Mobile Augmented Heritage: Enabling Human Life in ancient Pompeii. *International Journal of Architectural Computing*. **5** (2), pp.396-415.
- Papagiannakis, G. & Schertenleib, S. & O'Kennedy, B. & Arevalo-Poizat, M. & Magnenat-Thalmann, N. & Stoddart, A. & Thalmann, D., 2005. Mixing Virtual

- and Real Scenes in the Site of Ancient Pompeii. *Computer Animation and Virtual Worlds*. **16** (1), pp.11-24.
- Papagiannakis, G. & Singh, G. & Magnenat-Thalmann, N., 2008. A Survey of Mobile and Wireless Technologies for Augmented Reality Systems. *Computer Animation and Virtual Worlds*. **19** (1), pp.3-22.
- Pentland, A., 1987. A New Sense for Depth of Field, *Pattern Analysis and Machine Intelligence*. pp.523-531.
- Schmeil, A. & Broll, W., 2006. MARA: An Augmented Personal Assistant and Companion, *ACM SIGGRAPH 2006 Sketches*. New York, NY, USA: ACM, pp.141.
- Schon, TB & Karlsson, R. & Tornqvist, D. & Gustafsson, F., 2007. A Framework for Simultaneous Localization and Mapping Utilizing Model Structure, *10th International Conference on Information Fusion*. pp.1-8.
- Shi, J. & Tomasi, C., 1994. Good Features to Track, *Computer Vision and Pattern Recognition*. pp.593-600.
- Strasdat, H. & Montiel, JMM & Davison, A.J., 2010. Real-time Monocular SLAM: Why Filter? *IEEE International Conference on Robotics and Automation (ICRA 2010)*. pp.2657-2664.
- Steggles, P. & Gschwind, S., 2005. The Ubisense Smart Space Platform. *Adjunct Proceedings of the Third International Conference on Pervasive Computing*. **191**, pp.73-76.
- Yokokohji, Y. & Sugawara, Y. & Yoshikawa, T., 2000. Accurate Image Overlay on Video See-through HMDs Using Vision and Accelerometers, *Virtual Reality, 2000*. pp.247-254.
- You, S. & Neumann, U. & Azuma, R., 1999. Hybrid Inertial and Vision Tracking for Augmented Reality Registration, *Virtual Reality*. pp.260-267.

Other Publications

Lourakis M., 2010, A Generic Sparse Bundle Adjustment C/C++ Package Based on the Levenberg-Marquardt Algorithm [online]. Forth - Institute of Computer Science, <http://www.ics.forth.gr/~lourakis/sba/> [accessed May 01, 2010]

SensorWiki.org. 2009. <http://sensorwiki.org> [accessed May 01, 2011]

Appendix A. CAMERA MEASUREMENTS

2D/3D correspondence equations of the feature points detected on the image plane are given below:

$$p_t = \begin{bmatrix} p_{x,t} \\ p_{y,t} \\ p_{z,t} \end{bmatrix} = Q_{cs}(Q_{sw,t}(m_{w,t} - s_{w,t}) - c_s) \quad (\text{A.1})$$

where $m_{w,t}$ denotes the 3D position of the feature point in world coordinate system, $s_{w,t}$ denotes the camera position in world coordinate system, c_s denotes the center difference between camera and sensor coordinate systems, $Q_{sw,t}$ denotes the rotation matrix from world to sensor coordinate system, $Q_{cs,t}$ denotes the rotation matrix from sensor to camera coordinate system.

$$m_t = \begin{bmatrix} m_{x,t} \\ m_{y,t} \end{bmatrix} = \begin{bmatrix} \frac{p_{x,t}}{p_{z,t}} + e_{x,t}^c \\ \frac{p_{y,t}}{p_{z,t}} + e_{y,t}^c \end{bmatrix} \quad (\text{A.2})$$

where $e_{x,t}^c$ and $e_{y,t}^c$ denote time independent Gaussian camera measurement noise, $m_{x,t}$ and $m_{y,t}$ denote the 2D position of the feature point in image plane.

Using the equations given above in an extended Kalman filter in Section 3.1 requires the computation of the first derivative (Jacobian) of m_t with respect to $s_{w,t}$, $v_{w,t}$, $a_{w,t}$, $b_{s,t}^a$, $q_{sw,t}$, $\omega_{s,t}$, $b_{s,t}^\omega$.

The Jacobian of m_t with respect to $s_{w,t}$ is:

$$\frac{\partial m_t}{\partial s_{w,t}} = \begin{bmatrix} \frac{\partial m_{x,t}}{\partial s_{w,t}} \\ \frac{\partial m_{y,t}}{\partial s_{w,t}} \end{bmatrix} = \begin{bmatrix} \frac{\partial m_{x,t}}{\partial p_t} \frac{\partial p_t}{\partial s_{w,t}} \\ \frac{\partial m_{y,t}}{\partial p_t} \frac{\partial p_t}{\partial s_{w,t}} \end{bmatrix} \quad (\text{A.3})$$

where

$$\frac{\partial m_{x,t}}{\partial p_t} = \begin{bmatrix} \frac{\partial m_{x,t}}{\partial p_{x,t}} & \frac{\partial m_{x,t}}{\partial p_{y,t}} & \frac{\partial m_{x,t}}{\partial p_{z,t}} \end{bmatrix} \quad (\text{A.4a})$$

$$\frac{\partial m_{y,t}}{\partial p_t} = \begin{bmatrix} \frac{\partial m_{y,t}}{\partial p_{x,t}} & \frac{\partial m_{y,t}}{\partial p_{y,t}} & \frac{\partial m_{y,t}}{\partial p_{z,t}} \end{bmatrix} \quad (\text{A.4b})$$

$$\frac{\partial p_t}{\partial s_{w,t}} = \left(Q_{cs}(-Q_{sw,t}) \right) = \begin{bmatrix} \frac{\partial p_{x,t}}{\partial s_{wx,t}} & \frac{\partial p_{x,t}}{\partial s_{wy,t}} & \frac{\partial p_{x,t}}{\partial s_{wz,t}} \\ \frac{\partial p_{y,t}}{\partial s_{wx,t}} & \frac{\partial p_{y,t}}{\partial s_{wy,t}} & \frac{\partial p_{y,t}}{\partial s_{wz,t}} \\ \frac{\partial p_{z,t}}{\partial s_{wx,t}} & \frac{\partial p_{z,t}}{\partial s_{wy,t}} & \frac{\partial p_{z,t}}{\partial s_{wz,t}} \end{bmatrix} \quad (\text{A.4c})$$

The derivations of the above equations are:

$$\frac{\partial m_{x,t}}{\partial p_{x,t}} = \frac{1}{p_{z,t}}, \quad \frac{\partial m_{x,t}}{\partial p_{y,t}} = 0, \quad \frac{\partial m_{x,t}}{\partial p_{z,t}} = -\frac{p_{x,t}}{p_{z,t}^2} \quad (\text{A.5a})$$

$$\frac{\partial m_{y,t}}{\partial p_{x,t}} = 0, \quad \frac{\partial m_{y,t}}{\partial p_{y,t}} = \frac{1}{p_{z,t}}, \quad \frac{\partial m_{y,t}}{\partial p_{z,t}} = -\frac{p_{y,t}}{p_{z,t}^2} \quad (\text{A.5b})$$

and

$$\frac{\partial m_{x,t}}{\partial s_{wx,t}} = \frac{\partial m_{x,t}}{\partial p_{x,t}} \frac{\partial p_{x,t}}{\partial s_{wx,t}} + \underbrace{\frac{\partial m_{x,t}}{\partial p_{y,t}} \frac{\partial p_{y,t}}{\partial s_{wx,t}}}_0 + \frac{\partial m_{x,t}}{\partial p_{z,t}} \frac{\partial p_{z,t}}{\partial s_{wx,t}} = \frac{\partial m_{x,t}}{\partial p_{x,t}} \frac{\partial p_{x,t}}{\partial s_{wx,t}} + \frac{\partial m_{x,t}}{\partial p_{z,t}} \frac{\partial p_{z,t}}{\partial s_{wx,t}} \quad (\text{A.6a})$$

$$\frac{\partial m_{x,t}}{\partial s_{wy,t}} = \frac{\partial m_{x,t}}{\partial p_{x,t}} \frac{\partial p_{x,t}}{\partial s_{wy,t}} + \underbrace{\frac{\partial m_{x,t}}{\partial p_{y,t}} \frac{\partial p_{y,t}}{\partial s_{wy,t}}}_0 + \frac{\partial m_{x,t}}{\partial p_{z,t}} \frac{\partial p_{z,t}}{\partial s_{wy,t}} = \frac{\partial m_{x,t}}{\partial p_{x,t}} \frac{\partial p_{x,t}}{\partial s_{wy,t}} + \frac{\partial m_{x,t}}{\partial p_{z,t}} \frac{\partial p_{z,t}}{\partial s_{wy,t}} \quad (\text{A.6b})$$

$$\frac{\partial m_{x,t}}{\partial s_{wz,t}} = \frac{\partial m_{x,t}}{\partial p_{x,t}} \frac{\partial p_{x,t}}{\partial s_{wz,t}} + \underbrace{\frac{\partial m_{x,t}}{\partial p_{y,t}} \frac{\partial p_{y,t}}{\partial s_{wz,t}}}_0 + \frac{\partial m_{x,t}}{\partial p_{z,t}} \frac{\partial p_{z,t}}{\partial s_{wz,t}} = \frac{\partial m_{x,t}}{\partial p_{x,t}} \frac{\partial p_{x,t}}{\partial s_{wz,t}} + \frac{\partial m_{x,t}}{\partial p_{z,t}} \frac{\partial p_{z,t}}{\partial s_{wz,t}} \quad (\text{A.6c})$$

and

$$\frac{\partial m_{y,t}}{\partial s_{wx,t}} = \frac{\partial m_{y,t}}{\partial p_{x,t}} \frac{\partial p_{x,t}}{\partial s_{wx,t}} + \underbrace{\frac{\partial m_{y,t}}{\partial p_{y,t}} \frac{\partial p_{y,t}}{\partial s_{wx,t}}}_0 + \frac{\partial m_{y,t}}{\partial p_{z,t}} \frac{\partial p_{z,t}}{\partial s_{wx,t}} = \frac{\partial m_{y,t}}{\partial p_{y,t}} \frac{\partial p_{y,t}}{\partial s_{wx,t}} + \frac{\partial m_{y,t}}{\partial p_{z,t}} \frac{\partial p_{z,t}}{\partial s_{wx,t}} \quad (\text{A.7a})$$

$$\frac{\partial m_{y,t}}{\partial s_{wy,t}} = \underbrace{\frac{\partial m_{y,t}}{\partial p_{x,t}} \frac{\partial p_{x,t}}{\partial s_{wy,t}}}_0 + \frac{\partial m_{y,t}}{\partial p_{y,t}} \frac{\partial p_{y,t}}{\partial s_{wy,t}} + \frac{\partial m_{y,t}}{\partial p_{z,t}} \frac{\partial p_{z,t}}{\partial s_{wy,t}} = \frac{\partial m_{y,t}}{\partial p_{y,t}} \frac{\partial p_{y,t}}{\partial s_{wy,t}} + \frac{\partial m_{y,t}}{\partial p_{z,t}} \frac{\partial p_{z,t}}{\partial s_{wy,t}} \quad (\text{A.7b})$$

$$\frac{\partial m_{y,t}}{\partial s_{wz,t}} = \underbrace{\frac{\partial m_{y,t}}{\partial p_{x,t}} \frac{\partial p_{x,t}}{\partial s_{wz,t}}}_0 + \frac{\partial m_{y,t}}{\partial p_{y,t}} \frac{\partial p_{y,t}}{\partial s_{wz,t}} + \frac{\partial m_{y,t}}{\partial p_{z,t}} \frac{\partial p_{z,t}}{\partial s_{wz,t}} = \frac{\partial m_{y,t}}{\partial p_{y,t}} \frac{\partial p_{y,t}}{\partial s_{wz,t}} + \frac{\partial m_{y,t}}{\partial p_{z,t}} \frac{\partial p_{z,t}}{\partial s_{wz,t}} \quad (\text{A.7c})$$

As a result the final version of the Jacobian is:

$$\frac{\partial m_{x,t}}{\partial s_{w,t}} = \begin{bmatrix} \frac{1}{p_{z,t}} & 0 & -\frac{p_{x,t}}{p_{z,t}^2} \end{bmatrix} (Q_{cs}(-Q_{sw,t})) \quad (\text{A.8a})$$

$$\frac{\partial m_{y,t}}{\partial s_{w,t}} = \begin{bmatrix} 0 & \frac{1}{p_{z,t}} & -\frac{p_{y,t}}{p_{z,t}^2} \end{bmatrix} (Q_{cs}(-Q_{sw,t})) \quad (\text{A.8b})$$

The Jacobian of m_t with respect to $v_{w,t}$ is:

$$\frac{\partial m_t}{\partial v_{w,t}} = \begin{bmatrix} \frac{\partial m_{x,t}}{\partial v_{w,t}} \\ \frac{\partial m_{y,t}}{\partial v_{w,t}} \end{bmatrix} = \begin{bmatrix} \frac{\partial m_{x,t}}{\partial p_t} \frac{\partial p_t}{\partial v_{w,t}} \\ \frac{\partial m_{y,t}}{\partial p_t} \frac{\partial p_t}{\partial v_{w,t}} \end{bmatrix} \quad (\text{A.9})$$

where

$$\frac{\partial p_t}{\partial v_{w,t}} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} \frac{\partial p_{x,t}}{\partial v_{wx,t}} & \frac{\partial p_{x,t}}{\partial v_{wy,t}} & \frac{\partial p_{x,t}}{\partial v_{wz,t}} \\ \frac{\partial p_{y,t}}{\partial v_{wx,t}} & \frac{\partial p_{y,t}}{\partial v_{wy,t}} & \frac{\partial p_{y,t}}{\partial v_{wz,t}} \\ \frac{\partial p_{z,t}}{\partial v_{wx,t}} & \frac{\partial p_{z,t}}{\partial v_{wy,t}} & \frac{\partial p_{z,t}}{\partial v_{wz,t}} \end{bmatrix} \quad (\text{A.10})$$

The derivations of the above equations are:

$$\frac{\partial m_{x,t}}{\partial v_{wx,t}} = \frac{\partial m_{x,t}}{\partial p_{x,t}} \underbrace{\frac{\partial p_{x,t}}{\partial v_{wx,t}}}_0 + \frac{\partial m_{x,t}}{\partial p_{y,t}} \underbrace{\frac{\partial p_{y,t}}{\partial v_{wx,t}}}_0 + \frac{\partial m_{x,t}}{\partial p_{z,t}} \underbrace{\frac{\partial p_{z,t}}{\partial v_{wx,t}}}_0 = 0 \quad (\text{A.11a})$$

$$\frac{\partial m_{x,t}}{\partial v_{wy,t}} = \frac{\partial m_{x,t}}{\partial p_{x,t}} \underbrace{\frac{\partial p_{x,t}}{\partial v_{wy,t}}}_0 + \frac{\partial m_{x,t}}{\partial p_{y,t}} \underbrace{\frac{\partial p_{y,t}}{\partial v_{wy,t}}}_0 + \frac{\partial m_{x,t}}{\partial p_{z,t}} \underbrace{\frac{\partial p_{z,t}}{\partial v_{wy,t}}}_0 = 0 \quad (\text{A.11b})$$

$$\frac{\partial m_{x,t}}{\partial v_{wz,t}} = \frac{\partial m_{x,t}}{\partial p_{x,t}} \underbrace{\frac{\partial p_{x,t}}{\partial v_{wz,t}}}_0 + \frac{\partial m_{x,t}}{\partial p_{y,t}} \underbrace{\frac{\partial p_{y,t}}{\partial v_{wz,t}}}_0 + \frac{\partial m_{x,t}}{\partial p_{z,t}} \underbrace{\frac{\partial p_{z,t}}{\partial v_{wz,t}}}_0 = 0 \quad (\text{A.11c})$$

and

$$\frac{\partial m_{y,t}}{\partial v_{wx,t}} = \frac{\partial m_{y,t}}{\partial p_{x,t}} \underbrace{\frac{\partial p_{x,t}}{\partial v_{wx,t}}}_0 + \frac{\partial m_{y,t}}{\partial p_{y,t}} \underbrace{\frac{\partial p_{y,t}}{\partial v_{wx,t}}}_0 + \frac{\partial m_{y,t}}{\partial p_{z,t}} \underbrace{\frac{\partial p_{z,t}}{\partial v_{wx,t}}}_0 = 0 \quad (\text{A.12a})$$

$$\frac{\partial m_{y,t}}{\partial v_{wy,t}} = \frac{\partial m_{y,t}}{\partial p_{x,t}} \underbrace{\frac{\partial p_{x,t}}{\partial v_{wy,t}}}_0 + \frac{\partial m_{y,t}}{\partial p_{y,t}} \underbrace{\frac{\partial p_{y,t}}{\partial v_{wy,t}}}_0 + \frac{\partial m_{y,t}}{\partial p_{z,t}} \underbrace{\frac{\partial p_{z,t}}{\partial v_{wy,t}}}_0 = 0 \quad (\text{A.12b})$$

$$\frac{\partial m_{y,t}}{\partial v_{wz,t}} = \frac{\partial m_{y,t}}{\partial p_{x,t}} \underbrace{\frac{\partial p_{x,t}}{\partial v_{wz,t}}}_0 + \frac{\partial m_{y,t}}{\partial p_{y,t}} \underbrace{\frac{\partial p_{y,t}}{\partial v_{wz,t}}}_0 + \frac{\partial m_{y,t}}{\partial p_{z,t}} \underbrace{\frac{\partial p_{z,t}}{\partial v_{wz,t}}}_0 = 0 \quad (\text{A.12c})$$

As a result the final version of the Jacobian is:

$$\frac{\partial m_{x,t}}{\partial v_{w,t}} = \begin{bmatrix} \frac{1}{p_{z,t}} & 0 & -\frac{p_{x,t}}{p_{z,t}^2} \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} = [0 \quad 0 \quad 0] \quad (\text{A.13a})$$

$$\frac{\partial m_{y,t}}{\partial v_{w,t}} = \begin{bmatrix} 0 & \frac{1}{p_{z,t}} & -\frac{p_{y,t}}{p_{z,t}^2} \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} = [0 \quad 0 \quad 0] \quad (\text{A.13b})$$

The Jacobian of m_t with respect to $a_{w,t}$ is:

$$\frac{\partial m_t}{\partial v_{w,t}} = \begin{bmatrix} \frac{\partial m_{x,t}}{\partial a_{w,t}} \\ \frac{\partial m_{y,t}}{\partial a_{w,t}} \end{bmatrix} = \begin{bmatrix} \frac{\partial m_{x,t}}{\partial p_t} & \frac{\partial p_t}{\partial a_{w,t}} \\ \frac{\partial m_{y,t}}{\partial p_t} & \frac{\partial p_t}{\partial a_{w,t}} \end{bmatrix} \quad (\text{A.14})$$

where

$$\frac{\partial p_t}{\partial a_{w,t}} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} \frac{\partial p_{x,t}}{\partial a_{wx,t}} & \frac{\partial p_{x,t}}{\partial a_{wy,t}} & \frac{\partial p_{x,t}}{\partial a_{wz,t}} \\ \frac{\partial p_{y,t}}{\partial a_{wx,t}} & \frac{\partial p_{y,t}}{\partial a_{wy,t}} & \frac{\partial p_{y,t}}{\partial a_{wz,t}} \\ \frac{\partial p_{z,t}}{\partial a_{wx,t}} & \frac{\partial p_{z,t}}{\partial a_{wy,t}} & \frac{\partial p_{z,t}}{\partial a_{wz,t}} \end{bmatrix} \quad (\text{A.15})$$

The derivations of the above equations are:

$$\frac{\partial m_{x,t}}{\partial a_{wx,t}} = \frac{\partial m_{x,t}}{\partial p_{x,t}} \underbrace{\frac{\partial p_{x,t}}{\partial a_{wx,t}}}_0 + \frac{\partial m_{x,t}}{\partial p_{y,t}} \underbrace{\frac{\partial p_{y,t}}{\partial a_{wx,t}}}_0 + \frac{\partial m_{x,t}}{\partial p_{z,t}} \underbrace{\frac{\partial p_{z,t}}{\partial a_{wx,t}}}_0 = 0 \quad (\text{A.16a})$$

$$\frac{\partial m_{x,t}}{\partial a_{wy,t}} = \frac{\partial m_{x,t}}{\partial p_{x,t}} \underbrace{\frac{\partial p_{x,t}}{\partial a_{wy,t}}}_0 + \frac{\partial m_{x,t}}{\partial p_{y,t}} \underbrace{\frac{\partial p_{y,t}}{\partial a_{wy,t}}}_0 + \frac{\partial m_{x,t}}{\partial p_{z,t}} \underbrace{\frac{\partial p_{z,t}}{\partial a_{wy,t}}}_0 = 0 \quad (\text{A.16b})$$

$$\frac{\partial m_{x,t}}{\partial a_{wz,t}} = \frac{\partial m_{x,t}}{\partial p_{x,t}} \underbrace{\frac{\partial p_{x,t}}{\partial a_{wz,t}}}_0 + \frac{\partial m_{x,t}}{\partial p_{y,t}} \underbrace{\frac{\partial p_{y,t}}{\partial a_{wz,t}}}_0 + \frac{\partial m_{x,t}}{\partial p_{z,t}} \underbrace{\frac{\partial p_{z,t}}{\partial a_{wz,t}}}_0 = 0 \quad (\text{A.16c})$$

and

$$\frac{\partial m_{y,t}}{\partial a_{wx,t}} = \frac{\partial m_{y,t}}{\partial p_{x,t}} \underbrace{\frac{\partial p_{x,t}}{\partial a_{wx,t}}}_0 + \frac{\partial m_{y,t}}{\partial p_{y,t}} \underbrace{\frac{\partial p_{y,t}}{\partial a_{wx,t}}}_0 + \frac{\partial m_{y,t}}{\partial p_{z,t}} \underbrace{\frac{\partial p_{z,t}}{\partial a_{wx,t}}}_0 = 0 \quad (\text{A.17a})$$

$$\frac{\partial m_{y,t}}{\partial a_{wy,t}} = \frac{\partial m_{y,t}}{\partial p_{x,t}} \underbrace{\frac{\partial p_{x,t}}{\partial a_{wy,t}}}_0 + \frac{\partial m_{y,t}}{\partial p_{y,t}} \underbrace{\frac{\partial p_{y,t}}{\partial a_{wy,t}}}_0 + \frac{\partial m_{y,t}}{\partial p_{z,t}} \underbrace{\frac{\partial p_{z,t}}{\partial a_{wy,t}}}_0 = 0 \quad (\text{A.17b})$$

$$\frac{\partial m_{y,t}}{\partial a_{wz,t}} = \frac{\partial m_{y,t}}{\partial p_{x,t}} \underbrace{\frac{\partial p_{x,t}}{\partial a_{wz,t}}}_0 + \frac{\partial m_{y,t}}{\partial p_{y,t}} \underbrace{\frac{\partial p_{y,t}}{\partial a_{wz,t}}}_0 + \frac{\partial m_{y,t}}{\partial p_{z,t}} \underbrace{\frac{\partial p_{z,t}}{\partial a_{wz,t}}}_0 = 0 \quad (\text{A.17c})$$

As a result the final version of the Jacobian is:

$$\frac{\partial m_{x,t}}{\partial a_{w,t}} = \begin{bmatrix} 1 \\ p_{z,t} & 0 & -\frac{p_{x,t}}{p_{z,t}^2} \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} = [0 \quad 0 \quad 0] \quad (\text{A.18a})$$

$$\frac{\partial m_{y,t}}{\partial a_{w,t}} = \begin{bmatrix} 0 & \frac{1}{p_{z,t}} & -\frac{p_{y,t}}{p_{z,t}^2} \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} = [0 \quad 0 \quad 0] \quad (\text{A.18b})$$

The Jacobian of m_t with respect to $b_{s,t}^a$ is:

$$\frac{\partial m_t}{\partial b_{s,t}^a} = \begin{bmatrix} \frac{\partial m_{x,t}}{\partial b_{s,t}^a} \\ \frac{\partial m_{y,t}}{\partial b_{s,t}^a} \\ \frac{\partial m_{z,t}}{\partial b_{s,t}^a} \end{bmatrix} = \begin{bmatrix} \frac{\partial m_{x,t}}{\partial p_t} \frac{\partial p_t}{\partial b_{s,t}^a} \\ \frac{\partial m_{y,t}}{\partial p_t} \frac{\partial p_t}{\partial b_{s,t}^a} \\ \frac{\partial m_{z,t}}{\partial p_t} \frac{\partial p_t}{\partial b_{s,t}^a} \end{bmatrix} \quad (\text{A.19})$$

where

$$\frac{\partial p_t}{\partial b_{s,t}^a} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} \frac{\partial p_{x,t}}{\partial b_{sx,t}^a} & \frac{\partial p_{x,t}}{\partial b_{sy,t}^a} & \frac{\partial p_{x,t}}{\partial b_{sz,t}^a} \\ \frac{\partial p_{y,t}}{\partial b_{sx,t}^a} & \frac{\partial p_{y,t}}{\partial b_{sy,t}^a} & \frac{\partial p_{y,t}}{\partial b_{sz,t}^a} \\ \frac{\partial p_{z,t}}{\partial b_{sx,t}^a} & \frac{\partial p_{z,t}}{\partial b_{sy,t}^a} & \frac{\partial p_{z,t}}{\partial b_{sz,t}^a} \end{bmatrix} \quad (\text{A.20})$$

The derivations of the above equations are:

$$\frac{\partial m_{x,t}}{\partial b_{sx,t}^a} = \frac{\partial m_{x,t}}{\partial p_{x,t}} \underbrace{\frac{\partial p_{x,t}}{\partial b_{sx,t}^a}}_0 + \frac{\partial m_{x,t}}{\partial p_{y,t}} \underbrace{\frac{\partial p_{y,t}}{\partial b_{sx,t}^a}}_0 + \frac{\partial m_{x,t}}{\partial p_{z,t}} \underbrace{\frac{\partial p_{z,t}}{\partial b_{sx,t}^a}}_0 = 0 \quad (\text{A.21a})$$

$$\frac{\partial m_{x,t}}{\partial b_{sy,t}^a} = \frac{\partial m_{x,t}}{\partial p_{x,t}} \underbrace{\frac{\partial p_{x,t}}{\partial b_{sy,t}^a}}_0 + \frac{\partial m_{x,t}}{\partial p_{y,t}} \underbrace{\frac{\partial p_{y,t}}{\partial b_{sy,t}^a}}_0 + \frac{\partial m_{x,t}}{\partial p_{z,t}} \underbrace{\frac{\partial p_{z,t}}{\partial b_{sy,t}^a}}_0 = 0 \quad (\text{A.21b})$$

$$\frac{\partial m_{x,t}}{\partial b_{sz,t}^a} = \frac{\partial m_{x,t}}{\partial p_{x,t}} \underbrace{\frac{\partial p_{x,t}}{\partial b_{sz,t}^a}}_0 + \frac{\partial m_{x,t}}{\partial p_{y,t}} \underbrace{\frac{\partial p_{y,t}}{\partial b_{sz,t}^a}}_0 + \frac{\partial m_{x,t}}{\partial p_{z,t}} \underbrace{\frac{\partial p_{z,t}}{\partial b_{sz,t}^a}}_0 = 0 \quad (\text{A.21c})$$

and

$$\frac{\partial m_{y,t}}{\partial b_{sx,t}^a} = \frac{\partial m_{y,t}}{\partial p_{x,t}} \underbrace{\frac{\partial p_{x,t}}{\partial b_{sx,t}^a}}_0 + \frac{\partial m_{y,t}}{\partial p_{y,t}} \underbrace{\frac{\partial p_{y,t}}{\partial b_{sx,t}^a}}_0 + \frac{\partial m_{y,t}}{\partial p_{z,t}} \underbrace{\frac{\partial p_{z,t}}{\partial b_{sx,t}^a}}_0 = 0 \quad (\text{A.22a})$$

$$\frac{\partial m_{y,t}}{\partial b_{s_y,t}^a} = \underbrace{\frac{\partial m_{y,t}}{\partial p_{x,t}}}_{0} \underbrace{\frac{\partial p_{x,t}}{\partial b_{s_y,t}^a}}_0 + \underbrace{\frac{\partial m_{y,t}}{\partial p_{y,t}}}_{0} \frac{\partial p_{y,t}}{\partial b_{s_y,t}^a} + \underbrace{\frac{\partial m_{y,t}}{\partial p_{z,t}}}_{0} \frac{\partial p_{z,t}}{\partial b_{s_y,t}^a} = 0 \quad (\text{A.22b})$$

$$\frac{\partial m_{y,t}}{\partial b_{s_z,t}^a} = \underbrace{\frac{\partial m_{y,t}}{\partial p_{x,t}}}_{0} \underbrace{\frac{\partial p_{x,t}}{\partial b_{s_z,t}^a}}_0 + \underbrace{\frac{\partial m_{y,t}}{\partial p_{y,t}}}_{0} \frac{\partial p_{y,t}}{\partial b_{s_z,t}^a} + \underbrace{\frac{\partial m_{y,t}}{\partial p_{z,t}}}_{0} \frac{\partial p_{z,t}}{\partial b_{s_z,t}^a} = 0 \quad (\text{A.22c})$$

As a result the final version of the Jacobian is:

$$\frac{\partial m_{x,t}}{\partial b_{s_t}^a} = \begin{bmatrix} 1 & 0 & -\frac{p_{x,t}}{p_{z,t}^2} \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} = [0 \quad 0 \quad 0] \quad (\text{A.23a})$$

$$\frac{\partial m_{y,t}}{\partial b_{s_t}^a} = \begin{bmatrix} 0 & 1 & -\frac{p_{y,t}}{p_{z,t}^2} \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} = [0 \quad 0 \quad 0] \quad (\text{A.23b})$$

The Jacobian of m_t with respect to $q_{sw,t}$ is:

$$\frac{\partial m_t}{\partial q_{sw,t}} = \begin{bmatrix} \frac{\partial m_{x,t}}{\partial q_{sw,t}} \\ \frac{\partial m_{y,t}}{\partial q_{sw,t}} \end{bmatrix} = \begin{bmatrix} \frac{\partial m_{x,t}}{\partial p_t} & \frac{\partial p_t}{\partial q_{sw,t}} \\ \frac{\partial m_{y,t}}{\partial p_t} & \frac{\partial p_t}{\partial q_{sw,t}} \end{bmatrix} \quad (\text{A.24})$$

where

$$\frac{\partial p_t}{\partial q_{sw,t}} = \left(Q_{cs} \frac{\partial Q_{sw,t}(m_{w,t}-s_{w,t})}{\partial q_{sw,t}} \right) = \begin{bmatrix} \frac{\partial p_{x,t}}{\partial q_{w,t}} & \frac{\partial p_{x,t}}{\partial q_{x,t}} & \frac{\partial p_{x,t}}{\partial q_{y,t}} & \frac{\partial p_{x,t}}{\partial q_{z,t}} \\ \frac{\partial p_{y,t}}{\partial q_{w,t}} & \frac{\partial p_{y,t}}{\partial q_{x,t}} & \frac{\partial p_{y,t}}{\partial q_{y,t}} & \frac{\partial p_{y,t}}{\partial q_{z,t}} \\ \frac{\partial p_{z,t}}{\partial q_{w,t}} & \frac{\partial p_{z,t}}{\partial q_{x,t}} & \frac{\partial p_{z,t}}{\partial q_{y,t}} & \frac{\partial p_{z,t}}{\partial q_{z,t}} \end{bmatrix} \quad (\text{A.25a})$$

$$\frac{\partial Q(m)}{\partial q} = \begin{bmatrix} d_0 & d_1 & d_2 & d_3 \\ -d_3 & -d_2 & d_1 & d_0 \\ d_2 & -d_3 & -d_0 & d_1 \end{bmatrix} \quad \begin{aligned} d_0 &= 2(q_w m_x - q_z m_y + q_y m_z) \\ d_1 &= 2(q_x m_x + q_y m_y + q_z m_z) \\ d_2 &= 2(-q_y m_x + q_x m_y + q_w m_z) \\ d_3 &= 2(-q_z m_x - q_w m_y + q_x m_z) \end{aligned} \quad (\text{A.25b})$$

The derivations of the above equations are:

$$\frac{\partial m_{x,t}}{\partial q_{w,t}} = \frac{\partial m_{x,t}}{\partial p_{x,t}} \frac{\partial p_{x,t}}{\partial q_{w,t}} + \underbrace{\frac{\partial m_{x,t}}{\partial p_{y,t}} \frac{\partial p_{y,t}}{\partial q_{w,t}}}_0 + \frac{\partial m_{x,t}}{\partial p_{z,t}} \frac{\partial p_{z,t}}{\partial q_{w,t}} = \frac{\partial m_{x,t}}{\partial p_{x,t}} \frac{\partial p_{x,t}}{\partial q_{w,t}} + \frac{\partial m_{x,t}}{\partial p_{z,t}} \frac{\partial p_{z,t}}{\partial q_{w,t}} \quad (\text{A.26a})$$

$$\frac{\partial m_{x,t}}{\partial q_{x,t}} = \frac{\partial m_{x,t}}{\partial p_{x,t}} \frac{\partial p_{x,t}}{\partial q_{x,t}} + \underbrace{\frac{\partial m_{x,t}}{\partial p_{y,t}} \frac{\partial p_{y,t}}{\partial q_{x,t}}}_0 + \frac{\partial m_{x,t}}{\partial p_{z,t}} \frac{\partial p_{z,t}}{\partial q_{x,t}} = \frac{\partial m_{x,t}}{\partial p_{x,t}} \frac{\partial p_{x,t}}{\partial q_{x,t}} + \frac{\partial m_{x,t}}{\partial p_{z,t}} \frac{\partial p_{z,t}}{\partial q_{x,t}} \quad (\text{A.26b})$$

$$\frac{\partial m_{x,t}}{\partial q_{y,t}} = \frac{\partial m_{x,t}}{\partial p_{x,t}} \frac{\partial p_{x,t}}{\partial q_{y,t}} + \underbrace{\frac{\partial m_{x,t}}{\partial p_{y,t}} \frac{\partial p_{y,t}}{\partial q_{y,t}}}_0 + \frac{\partial m_{x,t}}{\partial p_{z,t}} \frac{\partial p_{z,t}}{\partial q_{y,t}} = \frac{\partial m_{x,t}}{\partial p_{x,t}} \frac{\partial p_{x,t}}{\partial q_{y,t}} + \frac{\partial m_{x,t}}{\partial p_{z,t}} \frac{\partial p_{z,t}}{\partial q_{y,t}} \quad (\text{A.26c})$$

$$\frac{\partial m_{x,t}}{\partial q_{z,t}} = \frac{\partial m_{x,t}}{\partial p_{x,t}} \frac{\partial p_{x,t}}{\partial q_{z,t}} + \underbrace{\frac{\partial m_{x,t}}{\partial p_{y,t}} \frac{\partial p_{y,t}}{\partial q_{z,t}}}_0 + \frac{\partial m_{x,t}}{\partial p_{z,t}} \frac{\partial p_{z,t}}{\partial q_{z,t}} = \frac{\partial m_{x,t}}{\partial p_{x,t}} \frac{\partial p_{x,t}}{\partial q_{z,t}} + \frac{\partial m_{x,t}}{\partial p_{z,t}} \frac{\partial p_{z,t}}{\partial q_{z,t}} \quad (\text{A.26d})$$

and

$$\frac{\partial m_{y,t}}{\partial q_{w,t}} = \frac{\partial m_{y,t}}{\partial p_{x,t}} \frac{\partial p_{x,t}}{\partial q_{w,t}} + \frac{\partial m_{y,t}}{\partial p_{y,t}} \frac{\partial p_{y,t}}{\partial q_{w,t}} + \frac{\partial m_{y,t}}{\partial p_{z,t}} \frac{\partial p_{z,t}}{\partial q_{w,t}} = \frac{\partial m_{y,t}}{\partial p_{x,t}} \frac{\partial p_{x,t}}{\partial q_{w,t}} + \frac{\partial m_{y,t}}{\partial p_{z,t}} \frac{\partial p_{z,t}}{\partial q_{w,t}} \quad (\text{A.27a})$$

$$\frac{\partial m_{y,t}}{\partial q_{x,t}} = \frac{\partial m_{y,t}}{\partial p_{x,t}} \frac{\partial p_{x,t}}{\partial q_{x,t}} + \frac{\partial m_{y,t}}{\partial p_{y,t}} \frac{\partial p_{y,t}}{\partial q_{x,t}} + \frac{\partial m_{y,t}}{\partial p_{z,t}} \frac{\partial p_{z,t}}{\partial q_{x,t}} = \frac{\partial m_{y,t}}{\partial p_{x,t}} \frac{\partial p_{x,t}}{\partial q_{x,t}} + \frac{\partial m_{y,t}}{\partial p_{z,t}} \frac{\partial p_{z,t}}{\partial q_{x,t}} \quad (\text{A.27b})$$

$$\frac{\partial m_{y,t}}{\partial q_{y,t}} = \frac{\partial m_{y,t}}{\partial p_{x,t}} \frac{\partial p_{x,t}}{\partial q_{y,t}} + \frac{\partial m_{y,t}}{\partial p_{y,t}} \frac{\partial p_{y,t}}{\partial q_{y,t}} + \frac{\partial m_{y,t}}{\partial p_{z,t}} \frac{\partial p_{z,t}}{\partial q_{y,t}} = \frac{\partial m_{y,t}}{\partial p_{x,t}} \frac{\partial p_{x,t}}{\partial q_{y,t}} + \frac{\partial m_{y,t}}{\partial p_{z,t}} \frac{\partial p_{z,t}}{\partial q_{y,t}} \quad (\text{A.27c})$$

$$\frac{\partial m_{y,t}}{\partial q_{z,t}} = \frac{\partial m_{y,t}}{\partial p_{x,t}} \frac{\partial p_{x,t}}{\partial q_{z,t}} + \frac{\partial m_{y,t}}{\partial p_{y,t}} \frac{\partial p_{y,t}}{\partial q_{z,t}} + \frac{\partial m_{y,t}}{\partial p_{z,t}} \frac{\partial p_{z,t}}{\partial q_{z,t}} = \frac{\partial m_{y,t}}{\partial p_{x,t}} \frac{\partial p_{x,t}}{\partial q_{z,t}} + \frac{\partial m_{y,t}}{\partial p_{z,t}} \frac{\partial p_{z,t}}{\partial q_{z,t}} \quad (\text{A.27d})$$

As a result the final version of the Jacobian is:

$$\frac{\partial m_{x,t}}{\partial q_{w,t}} = \begin{bmatrix} \frac{1}{p_z} & 0 & -\frac{p_x}{p_z^2} \end{bmatrix} \left(Q_{cs} \frac{\partial Q_{sw,t}(m_{w,t}-s_{w,t})}{\partial q_{sw,t}} \right) \quad (\text{A.28a})$$

$$\frac{\partial m_{ny,t}}{\partial q_{w,t}} = \begin{bmatrix} 0 & \frac{1}{p_z} & -\frac{p_y}{p_z^2} \end{bmatrix} \left(Q_{cs} \frac{\partial Q_{sw,t}(m_{w,t}-s_{w,t})}{\partial q_{sw,t}} \right) \quad (\text{A.28b})$$

The Jacobian of m_t with respect to $\omega_{s,t}$ is:

$$\frac{\partial m_t}{\partial \omega_{s,t}} = \begin{bmatrix} \frac{\partial m_{x,t}}{\partial \omega_{s,t}} \\ \frac{\partial m_{y,t}}{\partial \omega_{s,t}} \end{bmatrix} = \begin{bmatrix} \frac{\partial m_{x,t}}{\partial p_t} \frac{\partial p_t}{\partial \omega_{s,t}} \\ \frac{\partial m_{y,t}}{\partial p_t} \frac{\partial p_t}{\partial \omega_{s,t}} \end{bmatrix} \quad (\text{A.29})$$

where

$$\frac{\partial p_t}{\partial \omega_{s,t}} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} \frac{\partial p_{x,t}}{\partial \omega_{sx,t}} & \frac{\partial p_{x,t}}{\partial \omega_{sy,t}} & \frac{\partial p_{x,t}}{\partial \omega_{sz,t}} \\ \frac{\partial p_{y,t}}{\partial \omega_{sx,t}} & \frac{\partial p_{y,t}}{\partial \omega_{sy,t}} & \frac{\partial p_{y,t}}{\partial \omega_{sz,t}} \\ \frac{\partial p_{z,t}}{\partial \omega_{sx,t}} & \frac{\partial p_{z,t}}{\partial \omega_{sy,t}} & \frac{\partial p_{z,t}}{\partial \omega_{sz,t}} \end{bmatrix} \quad (\text{A.30})$$

The derivations of the above equations are:

$$\frac{\partial m_{x,t}}{\partial \omega_{sx,t}} = \frac{\partial m_{x,t}}{\partial p_{x,t}} \underbrace{\frac{\partial p_{x,t}}{\partial \omega_{sx,t}}}_0 + \frac{\partial m_{x,t}}{\partial p_{y,t}} \underbrace{\frac{\partial p_{y,t}}{\partial \omega_{sx,t}}}_0 + \frac{\partial m_{x,t}}{\partial p_{z,t}} \underbrace{\frac{\partial p_{z,t}}{\partial \omega_{sx,t}}}_0 = 0 \quad (\text{A.31a})$$

$$\frac{\partial m_{x,t}}{\partial \omega_{sy,t}} = \frac{\partial m_{x,t}}{\partial p_{x,t}} \underbrace{\frac{\partial p_{x,t}}{\partial \omega_{sy,t}}}_0 + \frac{\partial m_{x,t}}{\partial p_{y,t}} \underbrace{\frac{\partial p_{y,t}}{\partial \omega_{sy,t}}}_0 + \frac{\partial m_{x,t}}{\partial p_{z,t}} \underbrace{\frac{\partial p_{z,t}}{\partial \omega_{sy,t}}}_0 = 0 \quad (\text{A.31b})$$

$$\frac{\partial m_{x,t}}{\partial \omega_{sz,t}} = \frac{\partial m_{x,t}}{\partial p_{x,t}} \underbrace{\frac{\partial p_{x,t}}{\partial \omega_{sz,t}}}_0 + \frac{\partial m_{x,t}}{\partial p_{y,t}} \underbrace{\frac{\partial p_{y,t}}{\partial \omega_{sz,t}}}_0 + \frac{\partial m_{x,t}}{\partial p_{z,t}} \underbrace{\frac{\partial p_{z,t}}{\partial \omega_{sz,t}}}_0 = 0 \quad (\text{A.31c})$$

and

$$\frac{\partial m_{y,t}}{\partial \omega_{sx,t}} = \frac{\partial m_{y,t}}{\partial p_{x,t}} \underbrace{\frac{\partial p_{x,t}}{\partial \omega_{sx,t}}}_0 + \frac{\partial m_{y,t}}{\partial p_{y,t}} \underbrace{\frac{\partial p_{y,t}}{\partial \omega_{sx,t}}}_0 + \frac{\partial m_{y,t}}{\partial p_{z,t}} \underbrace{\frac{\partial p_{z,t}}{\partial \omega_{sx,t}}}_0 = 0 \quad (\text{A.32a})$$

$$\frac{\partial m_{y,t}}{\partial \omega_{sy,t}} = \frac{\partial m_{y,t}}{\partial p_{x,t}} \underbrace{\frac{\partial p_{x,t}}{\partial \omega_{sy,t}}}_0 + \frac{\partial m_{y,t}}{\partial p_{y,t}} \underbrace{\frac{\partial p_{y,t}}{\partial \omega_{sy,t}}}_0 + \frac{\partial m_{y,t}}{\partial p_{z,t}} \underbrace{\frac{\partial p_{z,t}}{\partial \omega_{sy,t}}}_0 = 0 \quad (\text{A.32b})$$

$$\frac{\partial m_{y,t}}{\partial \omega_{sz,t}} = \frac{\partial m_{y,t}}{\partial p_{x,t}} \underbrace{\frac{\partial p_{x,t}}{\partial \omega_{sz,t}}}_0 + \frac{\partial m_{y,t}}{\partial p_{y,t}} \underbrace{\frac{\partial p_{y,t}}{\partial \omega_{sz,t}}}_0 + \frac{\partial m_{y,t}}{\partial p_{z,t}} \underbrace{\frac{\partial p_{z,t}}{\partial \omega_{sz,t}}}_0 = 0 \quad (\text{A.32c})$$

As a result the final version of the Jacobian is:

$$\frac{\partial m_{x,t}}{\partial \omega_{s,t}} = \begin{bmatrix} \frac{1}{p_{z,t}} & 0 & -\frac{p_{x,t}}{p_{z,t}^2} \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} = [0 \quad 0 \quad 0] \quad (\text{A.33a})$$

$$\frac{\partial m_{y,t}}{\partial \omega_{s,t}} = \begin{bmatrix} 0 & \frac{1}{p_{z,t}} & -\frac{p_{y,t}}{p_{z,t}^2} \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} = [0 \quad 0 \quad 0] \quad (\text{A.33b})$$

The Jacobian of m_t with respect to $b_{s,t}^\omega$ is:

$$\frac{\partial m_t}{\partial b_{s,t}^\omega} = \begin{bmatrix} \frac{\partial m_{x,t}}{\partial b_{s,t}^\omega} \\ \frac{\partial m_{y,t}}{\partial b_{s,t}^\omega} \end{bmatrix} = \begin{bmatrix} \frac{\partial m_{x,t}}{\partial p_t} & \frac{\partial p_t}{\partial b_{s,t}^\omega} \\ \frac{\partial m_{y,t}}{\partial p_t} & \frac{\partial p_t}{\partial b_{s,t}^\omega} \end{bmatrix} \quad (\text{A.34})$$

where

$$\frac{\partial p_t}{\partial b_{s,t}^\omega} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} \frac{\partial p_{x,t}}{\partial b_{sx,t}^\omega} & \frac{\partial p_{x,t}}{\partial b_{sy,t}^\omega} & \frac{\partial p_{x,t}}{\partial b_{sz,t}^\omega} \\ \frac{\partial p_{y,t}}{\partial b_{sx,t}^\omega} & \frac{\partial p_{y,t}}{\partial b_{sy,t}^\omega} & \frac{\partial p_{y,t}}{\partial b_{sz,t}^\omega} \\ \frac{\partial p_{z,t}}{\partial b_{sx,t}^\omega} & \frac{\partial p_{z,t}}{\partial b_{sy,t}^\omega} & \frac{\partial p_{z,t}}{\partial b_{sz,t}^\omega} \end{bmatrix} \quad (\text{A.35})$$

The derivations of the above equations are:

$$\frac{\partial m_{x,t}}{\partial b_{sx,t}^\omega} = \frac{\partial m_{x,t}}{\partial p_{x,t}} \underbrace{\frac{\partial p_{x,t}}{\partial b_{sx,t}^\omega}}_0 + \frac{\partial m_{x,t}}{\partial p_{y,t}} \underbrace{\frac{\partial p_{y,t}}{\partial b_{sx,t}^\omega}}_0 + \frac{\partial m_{x,t}}{\partial p_{z,t}} \underbrace{\frac{\partial p_{z,t}}{\partial b_{sx,t}^\omega}}_0 = 0 \quad (\text{A.36a})$$

$$\frac{\partial m_{x,t}}{\partial b_{sy,t}^\omega} = \frac{\partial m_{x,t}}{\partial p_{x,t}} \underbrace{\frac{\partial p_{x,t}}{\partial b_{sy,t}^\omega}}_0 + \frac{\partial m_{x,t}}{\partial p_{y,t}} \underbrace{\frac{\partial p_{y,t}}{\partial b_{sy,t}^\omega}}_0 + \frac{\partial m_{x,t}}{\partial p_{z,t}} \underbrace{\frac{\partial p_{z,t}}{\partial b_{sy,t}^\omega}}_0 = 0 \quad (\text{A.36b})$$

$$\frac{\partial m_{x,t}}{\partial b_{sz,t}^\omega} = \frac{\partial m_{x,t}}{\partial p_{x,t}} \underbrace{\frac{\partial p_{x,t}}{\partial b_{sz,t}^\omega}}_0 + \frac{\partial m_{x,t}}{\partial p_{y,t}} \underbrace{\frac{\partial p_{y,t}}{\partial b_{sz,t}^\omega}}_0 + \frac{\partial m_{x,t}}{\partial p_{z,t}} \underbrace{\frac{\partial p_{z,t}}{\partial b_{sz,t}^\omega}}_0 = 0 \quad (\text{A.36c})$$

and

$$\frac{\partial m_{y,t}}{\partial b_{sx,t}^\omega} = \underbrace{\frac{\partial m_{y,t}}{\partial p_{x,t}}}_{0} \underbrace{\frac{\partial p_{x,t}}{\partial b_{sx,t}^\omega}}_0 + \underbrace{\frac{\partial m_{y,t}}{\partial p_{y,t}}}_{0} \frac{\partial p_{y,t}}{\partial b_{sx,t}^\omega} + \underbrace{\frac{\partial m_{y,t}}{\partial p_{z,t}}}_{0} \frac{\partial p_{z,t}}{\partial b_{sx,t}^\omega} = 0 \quad (\text{A.37a})$$

$$\frac{\partial m_{y,t}}{\partial b_{sy,t}^\omega} = \underbrace{\frac{\partial m_{y,t}}{\partial p_{x,t}}}_{0} \underbrace{\frac{\partial p_{x,t}}{\partial b_{sy,t}^\omega}}_0 + \underbrace{\frac{\partial m_{y,t}}{\partial p_{y,t}}}_{0} \frac{\partial p_{y,t}}{\partial b_{sy,t}^\omega} + \underbrace{\frac{\partial m_{y,t}}{\partial p_{z,t}}}_{0} \frac{\partial p_{z,t}}{\partial b_{sy,t}^\omega} = 0 \quad (\text{A.37b})$$

$$\frac{\partial m_{y,t}}{\partial b_{sz,t}^\omega} = \underbrace{\frac{\partial m_{y,t}}{\partial p_{x,t}}}_{0} \underbrace{\frac{\partial p_{x,t}}{\partial b_{sz,t}^\omega}}_0 + \underbrace{\frac{\partial m_{y,t}}{\partial p_{y,t}}}_{0} \frac{\partial p_{y,t}}{\partial b_{sz,t}^\omega} + \underbrace{\frac{\partial m_{y,t}}{\partial p_{z,t}}}_{0} \frac{\partial p_{z,t}}{\partial b_{sz,t}^\omega} = 0 \quad (\text{A.37c})$$

As a result the final version of the Jacobian is:

$$\frac{\partial m_{x,t}}{\partial b_{s,t}^\omega} = \begin{bmatrix} \frac{1}{p_{z,t}} & 0 & -\frac{p_{x,t}}{p_{z,t}^2} \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} = [0 \quad 0 \quad 0] \quad (\text{A.38a})$$

$$\frac{\partial m_{y,t}}{\partial b_{s,t}^\omega} = \begin{bmatrix} 0 & \frac{1}{p_{z,t}} & -\frac{p_{y,t}}{p_{z,t}^2} \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} = [0 \quad 0 \quad 0] \quad (\text{A.38b})$$

Appendix B. BAYESIAN FILTER LINEARIZATION

The time and the measurement model of Bayesian filters used in Section 3 are nonlinear and they have to be linearized in order to use EKF and Hybrid filters. The open versions of the linearized equations given in Section 3 are given in this part.

1. Both Angular Velocity and Linear Acceleration Data Used As Measurement

Time Model (EKF & Hybrid Filter):

$$F_t = \frac{\partial f}{\partial x} \Big|_{(\hat{x}_{t|t-T}, 0, 0, 0, 0)} = \begin{bmatrix} I_3 & TI_3 & \frac{T^2}{2}I_3 & 0_{3 \times 3} & 0_{3 \times 4} & 0_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & I_3 & TI_3 & 0_{3 \times 3} & 0_{3 \times 4} & 0_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & 0_{3 \times 3} & I_3 & 0_{3 \times 3} & 0_{3 \times 4} & 0_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & I_3 & 0_{3 \times 4} & 0_{3 \times 3} & 0_{3 \times 3} \\ 0_{4 \times 3} & 0_{4 \times 3} & 0_{4 \times 3} & 0_{4 \times 3} & A & -\frac{T}{2}B & 0_{4 \times 3} \\ 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 4} & I_3 & 0_{3 \times 3} \\ 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 4} & 0_{3 \times 3} & I_3 \end{bmatrix} \quad (\text{B.1a})$$

$$A = \frac{\partial(\exp(v) \odot q)}{\partial q} \Big|_{\substack{v = -\frac{T}{2}(\hat{\omega}_{s,t-T}) \\ q = \hat{q}_{sw,t-T}}} \quad (\text{B.1b})$$

$$B = \frac{\partial(\exp(v) \odot q)}{\partial v} \Big|_{\substack{v = -\frac{T}{2}(\hat{\omega}_{s,t-T}) \\ q = \hat{q}_{sw,t-T}}} \quad (\text{B.1c})$$

$$V_t = \frac{\partial f}{\partial v} \Big|_{(\hat{x}_{t|t-T}, 0, 0, 0, 0)} = \begin{bmatrix} \frac{T^2}{2}I_3 & 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} \\ TI_3 & 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} \\ I_3 & 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & 0_{3 \times 3} & I_3 & 0_{3 \times 3} \\ 0_{4 \times 3} & -\frac{T}{2}C & 0_{4 \times 3} & 0_{4 \times 3} \\ 0_{3 \times 3} & I_3 & 0_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & I_3 \end{bmatrix} \quad (\text{B.1d})$$

$$C = \frac{\partial(\exp(v) \odot q)}{\partial v} \Big|_{\substack{v = -\frac{T}{2}(\hat{\omega}_{s,t-T}) \\ q = \hat{q}_{sw,t-T}}} \quad (\text{B.1e})$$

Measurement Model (EKF):

$$H_t = \frac{\partial h}{\partial x} = \begin{bmatrix} 0_{3 \times 3} & 0_{3 \times 3} & Q_{sw,t} & I_3 & D & 0_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 4} & I_3 & I_3 \\ \frac{\partial m_{x,t}}{\partial s_{w,t}} & 0_{1 \times 3} & 0_{1 \times 3} & 0_{1 \times 3} & \frac{\partial m_{x,t}}{\partial q_{sw,t}} & 0_{1 \times 3} & 0_{1 \times 3} \\ \frac{\partial m_{y,t}}{\partial s_{w,t}} & 0_{1 \times 3} & 0_{1 \times 3} & 0_{1 \times 3} & \frac{\partial m_{y,t}}{\partial q_{sw,t}} & 0_{1 \times 3} & 0_{1 \times 3} \end{bmatrix} \quad (\text{B.2a})$$

$$D = \frac{\partial Q(q)m}{\partial q} \Big|_{\substack{m = \hat{a}_{w,t} - g_w \\ q = \hat{q}_{sw,t}}} \quad (\text{B.2b})$$

Measurement Model (Hybrid Filter):

$$H_t = \frac{\partial h}{\partial x} = \begin{bmatrix} 0_{3 \times 3} & 0_{3 \times 3} & Q_{sw,t} & I_3 & D & 0_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 4} & I_3 & I_3 \\ 0_{4 \times 3} & 0_{4 \times 3} & 0_{4 \times 3} & 0_{4 \times 3} & I_4 & 0_{4 \times 3} & 0_{4 \times 3} \\ I_3 & 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 4} & 0_{3 \times 3} & 0_{3 \times 3} \end{bmatrix} \quad (\text{B.3a})$$

$$D = \frac{\partial Q(q)m}{\partial q} \Big|_{\substack{m = \hat{a}_{w,t} - g_w \\ q = \hat{q}_{sw,t}}} \quad (\text{B.3b})$$

2. Angular Velocity Data Used As Control Input, Linear Acceleration Data Used As Measurement

Time Model (EKF & Hybrid Filter):

$$F_t = \frac{\partial f}{\partial x} \Big|_{(\hat{x}_{t|t-T}, 0, 0, 0, 0)} = \begin{bmatrix} I_3 & T I_3 & \frac{T^2}{2} I_3 & 0_{3 \times 3} & 0_{3 \times 4} & 0_{3 \times 3} \\ 0_{3 \times 3} & I_3 & T I_3 & 0_{3 \times 3} & 0_{3 \times 4} & 0_{3 \times 3} \\ 0_{3 \times 3} & 0_{3 \times 3} & I_3 & 0_{3 \times 3} & 0_{3 \times 4} & 0_{3 \times 3} \\ 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & I_3 & 0_{3 \times 4} & 0_{3 \times 3} \\ 0_{4 \times 3} & 0_{4 \times 3} & 0_{4 \times 3} & 0_{4 \times 3} & A & \frac{T}{2} B \\ 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 4} & I_3 \end{bmatrix} \quad (\text{B.4a})$$

$$A = \frac{\partial(\exp(v)\odot q)}{\partial q} \Big|_{\substack{v=-\frac{T}{2}(y_{s,t}^\omega - \hat{b}_{s,t-T}^\omega) \\ q=\hat{q}_{sw,t-T}}} \quad (\text{B.4b})$$

$$B = \frac{\partial(\exp(v)\odot q)}{\partial v} \Big|_{\substack{v=-\frac{T}{2}(y_{s,t}^\omega - \hat{b}_{s,t-T}^\omega) \\ q=\hat{q}_{sw,t-T}}} \quad (\text{B.4c})$$

$$V_t = \frac{\partial f}{\partial v} \Big|_{(\hat{x}_{t|t-T}, y_{s,t}^\omega, 0, 0, 0, 0)} = \begin{bmatrix} \frac{T^2}{2} I_3 & 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} \\ T I_3 & 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} \\ I_3 & 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & 0_{3 \times 3} & I_3 & 0_{3 \times 3} \\ 0_{4 \times 3} & -\frac{T}{2} C & 0_{4 \times 3} & 0_{4 \times 3} \\ 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & I_3 \end{bmatrix} \quad (\text{B.4d})$$

$$C = \frac{\partial(\exp(v)\odot q)}{\partial v} \Big|_{\substack{v=-\frac{T}{2}(y_{s,t}^\omega - \hat{b}_{s,t-T}^\omega) \\ q=\hat{q}_{sw,t-T}}} \quad (\text{B.4e})$$

$$V_m = \frac{\partial f}{\partial y_{s,t}^\omega} \Big|_{(\hat{x}_{t|t-T}, y_{s,t}^\omega, 0, 0, 0, 0)} = \begin{bmatrix} 0_{3 \times 3} \\ 0_{3 \times 3} \\ 0_{3 \times 3} \\ 0_{3 \times 3} \\ -\frac{T}{2} D \\ 0_{3 \times 3} \end{bmatrix} \quad (\text{B.4f})$$

$$D = \frac{\partial(\exp(v)\odot q)}{\partial v} \Big|_{\substack{v=-\frac{T}{2}(y_{s,t}^\omega - \hat{b}_{s,t-T}^\omega) \\ q=\hat{q}_{sw,t-T}}} \quad (\text{B.4g})$$

Measurement Model (EKF):

$$H_t = \frac{\partial h}{\partial x} = \begin{bmatrix} 0_{3 \times 3} & 0_{3 \times 3} & Q_{sw,t} & I_3 & E & 0_{3 \times 3} \\ \frac{\partial m_{x,t}}{\partial s_{w,t}} & 0_{1 \times 3} & 0_{1 \times 3} & 0_{1 \times 3} & \frac{\partial m_{x,t}}{\partial q_{sw,t}} & 0_{1 \times 3} \\ \frac{\partial m_{y,t}}{\partial s_{w,t}} & 0_{1 \times 3} & 0_{1 \times 3} & 0_{1 \times 3} & \frac{\partial m_{y,t}}{\partial q_{sw,t}} & 0_{1 \times 3} \end{bmatrix} \quad (\text{B.5a})$$

$$E = \frac{\partial Q(q)m}{\partial q} \Big|_{\substack{m=m_{w,t} - \hat{s}_{w,t} \\ q=\hat{q}_{sw,t}}} \quad (\text{B.5b})$$

Measurement Model (Hybrid Filter):

$$H_t = \frac{\partial h}{\partial x} = \begin{bmatrix} 0_{3 \times 3} & 0_{3 \times 3} & Q_{sw,t} & I_3 & E & 0_{3 \times 3} \\ 0_{4 \times 3} & 0_{4 \times 3} & 0_{4 \times 3} & 0_{4 \times 3} & I_4 & 0_{4 \times 3} \\ I_3 & 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 4} & 0_{3 \times 3} \end{bmatrix} \quad (\text{B.6a})$$

$$E = \left. \frac{\partial Q(q)m}{\partial q} \right|_{\substack{m=\hat{a}_{w,t}-g_w \\ q=\hat{q}_{sw,t}}} \quad (\text{B.6b})$$

3. Angular Velocity Data Used As Measurement, Linear Acceleration Data Used As Control Input

Time Model (EKF & Hybrid Filter):

$$F_t = \left. \frac{\partial f}{\partial x} \right|_{(\hat{x}_{t|t-T}, 0, 0, 0, 0)} = \begin{bmatrix} I_3 & T I_3 & -\frac{T^2}{2} Q_{ws,t-T} & \frac{T^2}{2} AB & 0_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & I_3 & -T Q_{ws,t-T} & TAB & 0_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & 0_{3 \times 3} & I_3 & 0_{3 \times 4} & 0_{3 \times 3} & 0_{3 \times 3} \\ 0_{4 \times 3} & 0_{4 \times 3} & 0_{4 \times 3} & C & -\frac{T}{2} D & 0_{4 \times 3} \\ 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & I_3 & 0_{3 \times 3} \\ 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 4} & 0_{3 \times 3} & I_3 \end{bmatrix} \quad (\text{B.7a})$$

$$A = \left. \frac{\partial Q(q)m}{\partial q} \right|_{\substack{m=(y_{s,t}^a - \hat{b}_{s,t-T}^a) \\ q=\hat{q}_{ws,t-T}}} \quad (\text{B.7b})$$

$$B = \left. \frac{\partial q^{-1}}{\partial q} \right|_{q=\hat{q}_{sw,t-T}} \quad (\text{B.7c})$$

$$C = \left. \frac{\partial (\exp(v) \odot q)}{\partial q} \right|_{\substack{v=-\frac{T}{2}(\hat{\omega}_{s,t-T}) \\ q=\hat{q}_{sw,t-T}}} \quad (\text{B.7d})$$

$$D = \left. \frac{\partial (\exp(v) \odot q)}{\partial v} \right|_{\substack{v=-\frac{T}{2}(\hat{\omega}_{s,t-T}) \\ q=\hat{q}_{sw,t-T}}} \quad (\text{B.7e})$$

$$V_t = \frac{\partial f}{\partial v} \Big|_{(\hat{x}_{t|t-T}, y_{s,t}^a, 0, 0, 0, 0)} = \begin{bmatrix} \frac{T^2}{2} I_3 & 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} \\ T I_3 & 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & 0_{3 \times 3} & I_3 & 0_{3 \times 3} \\ 0_{4 \times 3} & -\frac{T}{2} E & 0_{4 \times 3} & 0_{4 \times 3} \\ 0_{3 \times 3} & I_3 & 0_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & I_3 \end{bmatrix} \quad (\text{B.7f})$$

$$E = \frac{\partial(\exp(v) \odot q)}{\partial v} \Big|_{\substack{v = -\frac{T}{2}(\hat{\omega}_{s,t-T}) \\ q = \hat{q}_{sw,t-T}}} \quad (\text{B.7g})$$

$$V_m = \frac{\partial f}{\partial y_{s,t}^a} \Big|_{(\hat{x}_{t|t-T}, y_{s,t}^a, 0, 0, 0, 0)} = \begin{bmatrix} \frac{T^2}{2} Q_{ws,t-T} \\ T Q_{ws,t-T} \\ 0_{3 \times 3} \\ 0_{4 \times 3} \\ 0_{3 \times 3} \\ 0_{3 \times 3} \end{bmatrix} \quad (\text{B.7h})$$

Measurement Model (EKF):

$$H_t = \frac{\partial h}{\partial x} = \begin{bmatrix} 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 4} & I_3 & I_3 \\ \frac{\partial m_{x,t}}{\partial s_{w,t}} & 0_{1 \times 3} & 0_{1 \times 3} & \frac{\partial m_{x,t}}{\partial q_{sw,t}} & 0_{1 \times 3} & 0_{1 \times 3} \\ \frac{\partial m_{y,t}}{\partial s_{w,t}} & 0_{1 \times 3} & 0_{1 \times 3} & \frac{\partial m_{y,t}}{\partial q_{sw,t}} & 0_{1 \times 3} & 0_{1 \times 3} \end{bmatrix} \quad (\text{B.8})$$

Measurement Model (Hybrid Filter):

$$H_t = \frac{\partial h}{\partial x} = \begin{bmatrix} 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 4} & I_3 & I_3 \\ 0_{4 \times 3} & 0_{4 \times 3} & 0_{4 \times 3} & I_4 & 0_{4 \times 3} & 0_{4 \times 3} \\ I_3 & 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 4} & 0_{3 \times 3} & 0_{3 \times 3} \end{bmatrix} \quad (\text{B.9})$$

4. Both Angular Velocity And Linear Acceleration Data Used As Control Input

Time Model (EKF & Hybrid Filter):

$$F_t = \frac{\partial f}{\partial x} \Big|_{(\hat{x}_{t|t-T}, y_{s,t}^a, y_{s,t}^\omega, 0, 0, 0, 0)} = \begin{bmatrix} I_3 & TI_3 & -\frac{T^2}{2} Q_{ws,t-T} & \frac{T^2}{2} AB & 0_{3 \times 3} \\ 0_{3 \times 3} & I_3 & -T Q_{ws,t-T} & TAB & 0_{3 \times 3} \\ 0_{3 \times 3} & 0_{3 \times 3} & I_3 & 0_{3 \times 4} & 0_{3 \times 3} \\ 0_{4 \times 3} & 0_{4 \times 3} & 0_{4 \times 3} & C & \frac{T}{2} D \\ 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 4} & I_3 \end{bmatrix} \quad (\text{B.10a})$$

$$A = \frac{\partial Q(q)m}{\partial q} \Big|_{\substack{m=(y_{s,t}^a - \hat{b}_{s,t-T}^a) \\ q=\hat{q}_{ws,t-T}}} \quad (\text{B.10b})$$

$$B = \frac{\partial q^{-1}}{\partial q} \Big|_{q=\hat{q}_{sw,t-T}} \quad (\text{B.10c})$$

$$C = \frac{\partial (\exp(v) \odot q)}{\partial q} \Big|_{\substack{v=-\frac{T}{2}(y_{s,t}^\omega - \hat{b}_{s,t-T}^\omega) \\ q=\hat{q}_{sw,t-T}}} \quad (\text{B.10d})$$

$$D = \frac{\partial (\exp(v) \odot q)}{\partial v} \Big|_{\substack{v=-\frac{T}{2}(y_{s,t}^\omega - \hat{b}_{s,t-T}^\omega) \\ q=\hat{q}_{sw,t-T}}} \quad (\text{B.10e})$$

$$V_t = \frac{\partial f}{\partial v} \Big|_{(\hat{x}_{t|t-T}, y_{s,t}^a, y_{s,t}^\omega, 0, 0, 0, 0)} = \begin{bmatrix} \frac{T^2}{2} I_3 & 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} \\ TI_3 & 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & 0_{3 \times 3} & I_3 & 0_{3 \times 3} \\ 0_{4 \times 3} & -\frac{T}{2} E & 0_{4 \times 3} & 0_{4 \times 3} \\ 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & I_3 \end{bmatrix} \quad (\text{B.10f})$$

$$E = \frac{\partial (\exp(v) \odot q)}{\partial v} \Big|_{\substack{v=-\frac{T}{2}(y_{s,t}^\omega - \hat{b}_{s,t-T}^\omega) \\ q=\hat{q}_{sw,t-T}}} \quad (\text{B.10g})$$

$$V_m = \frac{\partial f}{\partial y_{s,t}} \Big|_{(\hat{x}_{t|t-T}, y_{s,t}^a, y_{s,t}^\omega, 0, 0, 0, 0)} = \begin{bmatrix} \frac{T^2}{2} Q_{ws,t-T} & 0_{3 \times 3} \\ T Q_{ws,t-T} & 0_{3 \times 3} \\ 0_{3 \times 3} & 0_{3 \times 3} \\ 0_{4 \times 3} & -\frac{T}{2} F \\ 0_{3 \times 3} & 0_{3 \times 3} \end{bmatrix} \quad (\text{B.10h})$$

$$F = \frac{\partial(\exp(v) \odot q)}{\partial v} \Big|_{\substack{v = -\frac{T}{2}(y_{s,t}^\omega - \hat{b}_{s,t-T}^\omega) \\ q = \hat{q}_{sw,t-T}}} \quad (\text{B.10i})$$

Measurement Model (EKF):

$$H_t = \frac{\partial h}{\partial x} = \begin{bmatrix} \frac{\partial m_{x,t}}{\partial s_{w,t}} & 0_{1 \times 3} & 0_{1 \times 3} & \frac{\partial m_{x,t}}{\partial q_{sw,t}} & 0_{1 \times 3} \\ \frac{\partial m_{y,t}}{\partial s_{w,t}} & 0_{1 \times 3} & 0_{1 \times 3} & \frac{\partial m_{y,t}}{\partial q_{sw,t}} & 0_{1 \times 3} \end{bmatrix} \quad (\text{B.11})$$

Measurement Model (Hybrid Filter):

$$H_t = \frac{\partial h}{\partial x} = \begin{bmatrix} 0_{4 \times 3} & 0_{4 \times 3} & 0_{4 \times 3} & I_4 & 0_{4 \times 3} \\ I_3 & 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 4} & 0_{3 \times 3} \end{bmatrix} \quad (\text{B.12})$$

5. Angular Velocity Data Used As Measurement

Time Model (EKF & Hybrid Filter):

$$F_t = \frac{\partial f}{\partial x} \Big|_{(\hat{x}_{t|t-T}, 0, 0, 0)} = \begin{bmatrix} I_3 & T I_3 & 0_{3 \times 4} & 0_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & I_3 & 0_{3 \times 4} & 0_{3 \times 3} & 0_{3 \times 3} \\ 0_{4 \times 3} & 0_{4 \times 3} & A & -\frac{T}{2} B & 0_{4 \times 3} \\ 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 4} & I_3 & 0_{3 \times 3} \\ 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 4} & 0_{3 \times 3} & I_3 \end{bmatrix} \quad (\text{B.13a})$$

$$A = \frac{\partial(\exp(v) \odot q)}{\partial q} \Big|_{\substack{v = -\frac{T}{2}(\hat{\omega}_{s,t-T}) \\ q = \hat{q}_{sw,t-T}}} \quad (\text{B.13b})$$

$$B = \frac{\partial(\exp(v) \odot q)}{\partial v} \Big|_{\substack{v = -\frac{T}{2}(\hat{\omega}_{s,t-T}) \\ q = \hat{q}_{sw,t-T}}} \quad (\text{B.13c})$$

$$V_t = \frac{\partial f}{\partial v} \Big|_{(\hat{x}_{t|t-T}, 0, 0, 0)} = \begin{bmatrix} TI_3 & 0_{3 \times 3} & 0_{3 \times 3} \\ I_3 & 0_{3 \times 3} & 0_{3 \times 3} \\ 0_{4 \times 3} & -\frac{T}{2}C & 0_{4 \times 3} \\ 0_{3 \times 3} & I_3 & 0_{3 \times 3} \\ 0_{3 \times 3} & 0_{3 \times 3} & I_3 \end{bmatrix} \quad (\text{B.13d})$$

$$C = \frac{\partial(\exp(v) \odot q)}{\partial v} \Big|_{\substack{v = -\frac{T}{2}(\hat{\omega}_{s,t-T}) \\ q = \hat{q}_{sw,t-T}}} \quad (\text{B.13e})$$

Measurement Model (EKF):

$$H_t = \frac{\partial h}{\partial x} = \begin{bmatrix} 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 4} & I_3 & I_3 \\ \frac{\partial m_{x,t}}{\partial s_{w,t}} & 0_{1 \times 3} & \frac{\partial m_{x,t}}{\partial q_{sw,t}} & 0_{1 \times 3} & 0_{1 \times 3} \\ \frac{\partial m_{y,t}}{\partial s_{w,t}} & 0_{1 \times 3} & \frac{\partial m_{y,t}}{\partial q_{sw,t}} & 0_{1 \times 3} & 0_{1 \times 3} \end{bmatrix} \quad (\text{B.14})$$

Measurement Model (Hybrid Filter):

$$H_t = \frac{\partial h}{\partial x} = \begin{bmatrix} 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 4} & I_3 & I_3 \\ 0_{4 \times 3} & 0_{4 \times 3} & I_4 & 0_{4 \times 3} & 0_{4 \times 3} \\ I_3 & 0_{3 \times 3} & 0_{3 \times 4} & 0_{3 \times 3} & 0_{3 \times 3} \end{bmatrix} \quad (\text{B.15})$$

6. Angular Velocity Data Used As Control Input

Time Model (EKF & Hybrid Filter):

$$F_t = \frac{\partial f}{\partial x} \Big|_{(x_{t-T}, y_{s,t}^\omega, 0, 0, 0)} = \begin{bmatrix} I_3 & TI_3 & 0_{3 \times 4} & 0_{3 \times 3} \\ 0_{3 \times 3} & I_3 & 0_{3 \times 4} & 0_{3 \times 3} \\ 0_{4 \times 3} & 0_{4 \times 3} & A & \frac{T}{2}B \\ 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 4} & I_3 \end{bmatrix} \quad (\text{B.16a})$$

$$A = \frac{\partial(\exp(v) \odot q)}{\partial q} \Big|_{\substack{v = -\frac{T}{2}(y_{s,t}^\omega - \hat{b}_{s,t-T}^\omega) \\ q = \hat{q}_{sw,t-T}}} \quad (\text{B.16b})$$

$$B = \frac{\partial(\exp(v)\odot q)}{\partial v} \Big|_{\substack{v=-\frac{T}{2}(y_{s,t}^\omega - \hat{b}_{s,t-T}^\omega) \\ q=\hat{q}_{sw,t-T}}} \quad (\text{B.16c})$$

$$V_t = \frac{\partial f}{\partial v} \Big|_{(x_{t-T}, y_{s,t}^w, 0, 0, 0)} = \begin{bmatrix} TI_3 & 0_{3 \times 3} & 0_{3 \times 3} \\ I_3 & 0_{3 \times 3} & 0_{3 \times 3} \\ 0_{4 \times 3} & -\frac{T}{2}C & 0_{4 \times 3} \\ 0_{3 \times 3} & 0_{3 \times 3} & I_3 \end{bmatrix} \quad (\text{B.16d})$$

$$C = \frac{\partial(\exp(v)\odot q)}{\partial v} \Big|_{\substack{v=-\frac{T}{2}(y_{s,t}^\omega - \hat{b}_{s,t-T}^\omega) \\ q=\hat{q}_{sw,t-T}}} \quad (\text{B.16e})$$

$$\hat{x}_{t|t-T} = f(x_{t-T}, y_{s,t}^\omega, 0, 0, 0) \quad (\text{B.16f})$$

$$V_m = \frac{\partial f}{\partial y_{s,t}^w} \Big|_{(x_{t-T}, y_{s,t}^\omega, 0, 0, 0)} = \begin{bmatrix} 0_{3 \times 3} \\ 0_{3 \times 3} \\ -\frac{T}{2}D \\ 0_{3 \times 3} \end{bmatrix} \quad (\text{B.16g})$$

$$D = \frac{\partial(\exp(v)\odot q)}{\partial v} \Big|_{\substack{v=-\frac{T}{2}(y_{s,t}^\omega - \hat{b}_{s,t-T}^\omega) \\ q=\hat{q}_{sw,t-T}}} \quad (\text{B.16h})$$

Measurement Model (EKF):

$$H_t = \frac{\partial h}{\partial x} = \begin{bmatrix} \frac{\partial m_{x,t}}{\partial s_{w,t}} & 0_{1 \times 3} & \frac{\partial m_{x,t}}{\partial q_{sw,t}} & 0_{1 \times 3} \\ \frac{\partial m_{y,t}}{\partial s_{w,t}} & 0_{1 \times 3} & \frac{\partial m_{y,t}}{\partial q_{sw,t}} & 0_{1 \times 3} \end{bmatrix} \quad (\text{B.17})$$

Measurement Model (Hybrid Filter):

$$H_t = \frac{\partial h}{\partial x} = \begin{bmatrix} 0_{4 \times 3} & 0_{4 \times 3} & I_4 & 0_{4 \times 3} \\ I_3 & 0_{3 \times 3} & 0_{3 \times 4} & 0_{3 \times 3} \end{bmatrix} \quad (\text{B.18})$$

7. Linear Acceleration Data Used As Measurement

Time Model (EKF & Hybrid Filter):

$$F_t = \frac{\partial f}{\partial x} \Big|_{(\hat{x}_{t|t-T}, 0, 0, 0)} = \begin{bmatrix} I_3 & TI_3 & \frac{T^2}{2}I_3 & 0_{3 \times 3} & 0_{3 \times 4} \\ 0_{3 \times 3} & I_3 & TI_3 & 0_{3 \times 3} & 0_{3 \times 4} \\ 0_{3 \times 3} & 0_{3 \times 3} & I_3 & 0_{3 \times 3} & 0_{3 \times 4} \\ 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & I_3 & 0_{3 \times 4} \\ 0_{4 \times 3} & 0_{4 \times 3} & 0_{4 \times 3} & 0_{4 \times 3} & I_4 \end{bmatrix} \quad (\text{B.19a})$$

$$V_t = \frac{\partial f}{\partial v} \Big|_{(\hat{x}_{t|t-T}, 0, 0, 0)} = \begin{bmatrix} \frac{T^2}{2}I_3 & 0_{3 \times 3} & 0_{3 \times 3} \\ TI_3 & 0_{3 \times 3} & 0_{3 \times 3} \\ I_3 & 0_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & 0_{3 \times 3} & I_3 \\ 0_{4 \times 3} & A & 0_{4 \times 3} \end{bmatrix} \quad (\text{B.19b})$$

$$A = \frac{\partial q}{\partial v} \Big|_{\substack{q = \hat{q}_{sw,t|t-T} \\ v = \varepsilon_{s,t}}} \quad (\text{B.19c})$$

Measurement Model (EKF):

$$H_t = \frac{\partial h}{\partial x} = \begin{bmatrix} 0_{3 \times 3} & 0_{3 \times 3} & Q_{sw,t} & I_3 & B \\ \frac{\partial m_{x,t}}{\partial s_{w,t}} & 0_{1 \times 3} & 0_{1 \times 3} & 0_{1 \times 3} & \frac{\partial m_{x,t}}{\partial q_{sw,t}} \\ \frac{\partial m_{y,t}}{\partial s_{w,t}} & 0_{1 \times 3} & 0_{1 \times 3} & 0_{1 \times 3} & \frac{\partial m_{y,t}}{\partial q_{sw,t}} \end{bmatrix} \quad (\text{B.20a})$$

$$A = \frac{\partial Q(m)}{\partial q} \Big|_{\substack{Q = Q_{sw,t} \\ m = m_{w,t} - \hat{s}_{w,t} \\ q = \hat{q}_{sw,t}}} \quad (\text{B.20b})$$

Measurement Model (Hybrid Filter):

$$H_t = \frac{\partial h}{\partial x} = \begin{bmatrix} 0_{3 \times 3} & 0_{3 \times 3} & Q_{sw,t} & I_3 & B \\ 0_{4 \times 3} & 0_{4 \times 3} & 0_{4 \times 3} & 0_{4 \times 3} & I_4 \\ I_3 & 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 4} \end{bmatrix} \quad (\text{B.21a})$$

$$B = \frac{\partial Q(q)m}{\partial q} \Big|_{\substack{m=\hat{a}_{w,t}-g_w \\ q=\hat{q}_{sw,t}}} \quad (\text{B.21b})$$

8. Linear Acceleration Data Used As Control Input

Time Model (EKF & Hybrid Filter):

$$F_t = \frac{\partial f}{\partial x} \Big|_{(\hat{x}_{t|t-T}, 0, 0, 0)} = \begin{bmatrix} I_3 & TI_3 & -\frac{T^2}{2} Q_{ws,t-T} & \frac{T^2}{2} AB \\ 0_{3 \times 3} & I_3 & -TQ_{ws,t-T} & TAB \\ 0_{3 \times 3} & 0_{3 \times 3} & I_3 & 0_{3 \times 4} \\ 0_{4 \times 3} & 0_{4 \times 3} & 0_{4 \times 3} & I_4 \end{bmatrix} \quad (\text{B.22a})$$

$$A = \frac{\partial Q(q)m}{\partial q} \Big|_{\substack{m=(y_{s,t}^a - \hat{b}_{s,t-T}^a) \\ q=\hat{q}_{ws,t-T}}} \quad (\text{B.22b})$$

$$B = \frac{\partial q^{-1}}{\partial q} \Big|_{q=\hat{q}_{sw,t-T}} \quad (\text{B.22c})$$

$$V_t = \frac{\partial f}{\partial v} \Big|_{(x_{t-T}, y_{s,t}^a, 0, 0, 0)} = \begin{bmatrix} \frac{T^2}{2} I_3 & 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} \\ TI_3 & 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & 0_{3 \times 3} & I_3 & 0_{3 \times 3} \\ 0_{4 \times 3} & C & 0_{4 \times 3} & 0_{4 \times 3} \end{bmatrix} \quad (\text{B.22d})$$

$$B = \frac{\partial q}{\partial v} \Big|_{\substack{q=\hat{q}_{sw,t|t-T} \\ v=\varepsilon_{s,t}^\theta}} \quad (\text{B.22e})$$

$$V_m = \frac{\partial f}{\partial \sigma_{e_{s,t}^a}^2} \Big|_{(x_{t-T}, y_{s,t}^a, 0, 0, 0)} = \begin{bmatrix} \frac{T^2}{2} Q_{ws,t-T} \\ TQ_{ws,t-T} \\ 0_{3 \times 3} \\ 0_{4 \times 3} \end{bmatrix} \quad (\text{B.22f})$$

Measurement Model (EKF):

$$H_t = \frac{\partial h}{\partial x} = \begin{bmatrix} \frac{\partial m_{x,t}}{\partial s_{w,t}} & 0_{1 \times 3} & 0_{1 \times 3} & \frac{\partial m_{x,t}}{\partial q_{sw,t}} \\ \frac{\partial m_{y,t}}{\partial s_{w,t}} & 0_{1 \times 3} & 0_{1 \times 3} & \frac{\partial m_{y,t}}{\partial q_{sw,t}} \end{bmatrix} \quad (\text{B.23})$$

Measurement Model (Hybrid Filter):

$$H_t = \frac{\partial h}{\partial x} = \begin{bmatrix} 0_{4 \times 3} & 0_{4 \times 3} & 0_{4 \times 3} & I_4 \\ I_3 & 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 4} \end{bmatrix} \quad (\text{B.24})$$

9. Both Angular Velocity and Linear Acceleration Data Not Used

Time Model (EKF & Hybrid Filter):

$$F_t = \frac{\partial f}{\partial x} \Big|_{(\hat{x}_{t|t-T}, 0, 0)} = \begin{bmatrix} I_3 & TI_3 & 0 \\ 0 & I_3 & 0 \\ 0 & 0 & A \end{bmatrix} \quad (\text{B.25a})$$

$$A = \frac{\partial(\exp(v) \odot q)}{\partial q} \Big|_{\substack{v=0 \\ q=\hat{q}_{sw,t-T}}} \quad (\text{B.25b})$$

$$V_t = \frac{\partial f}{\partial v} \Big|_{(\hat{x}_{t|t-T}, 0, 0)} = \begin{bmatrix} TI_3 & 0 \\ I_3 & 0 \\ 0 & B \end{bmatrix} \quad (\text{B.25c})$$

$$B = \frac{\partial(\exp(v) \odot q)}{\partial \theta} \Big|_{\substack{q=\hat{q}_{sw,t|t-T} \\ v=\frac{1}{2}\xi_{s,t}^\theta}} \quad (\text{B.25d})$$

Measurement Model (EKF):

$$H_t = \frac{\partial h}{\partial x} = \begin{bmatrix} \frac{\partial m_{x,t}}{\partial s_{w,t}} & 0_{1 \times 3} & \frac{\partial m_{x,t}}{\partial q_{sw,t}} \\ \frac{\partial m_{y,t}}{\partial s_{w,t}} & 0_{1 \times 3} & \frac{\partial m_{y,t}}{\partial q_{sw,t}} \end{bmatrix} \quad (\text{B.26})$$

Measurement Model (Hybrid Filter):

$$H_t = \frac{\partial h}{\partial x} = \begin{bmatrix} 0_{4 \times 3} & 0_{4 \times 3} & I_4 \\ I_3 & 0_{3 \times 3} & 0_{3 \times 4} \end{bmatrix} \quad (\text{B.27})$$

Appendix C. QUATERNIONS

Unit quaternion provides a convenient mathematical notation for representing orientations and rotations of the objects in 3Ds. Compared to Euler angles they are simpler to compose and avoid problem of gimbal lock. Gimbal lock is the loss of one degree of freedom in a 3D space that occurs when the axes of two of the three gimbals are driven into a parallel configuration, “locking” the system into rotation in a degenerate 2D space. Compared to rotation matrices they are more numerically stable and may be more efficient. A quaternion $q = [q_w, q_x, q_y, q_z]^T$ is defined as a four-dimensional vector.

A unit quaternion represents the rotation of the 3D vector around \hat{n} with angle θ .

$$q = \begin{bmatrix} \cos \frac{\theta}{2} \\ \hat{n} \sin \frac{\theta}{2} \end{bmatrix} = \begin{bmatrix} \cos \frac{\theta}{2} \\ n_x \sin \frac{\theta}{2} \\ n_y \sin \frac{\theta}{2} \\ n_z \sin \frac{\theta}{2} \end{bmatrix} \quad (\text{C.1})$$

Quaternion Conjugate:

The conjugate q^* of quaternion q is defined as:

$$q^* = [q_w, -q_x, -q_y, -q_z]^T \quad (\text{C.2})$$

Quaternion Sum:

The sum of quaternions is the same as the vector sum:

$$q + p = \begin{bmatrix} q_w + p_w \\ q_x + p_x \\ q_y + p_y \\ q_z + p_z \end{bmatrix} \quad (\text{C.3})$$

Quaternion Product:

The product \odot of two quaternions q and p is:

$$p \odot q = \begin{bmatrix} p_w q_w - p_v \cdot q_v \\ p_w q_v + q_w p_v + p_v \times q_v \end{bmatrix} = \begin{bmatrix} p_w q_w - p_x q_x - p_y q_y - p_z q_z \\ p_w q_x + p_x q_w + p_y q_z - p_z q_y \\ p_w q_y + p_y q_w + p_z q_x - p_x q_z \\ p_w q_z + p_z q_w + p_x q_y - p_y q_x \end{bmatrix} \quad (\text{C.4})$$

where $p_v \cdot q_v$ denotes the scalar (inner) product and $p_v \times q_v$ denotes the cross (outer) product of two 3D vectors:

$$p_v \times q_v = \begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix} \times \begin{bmatrix} q_x \\ q_y \\ q_z \end{bmatrix} = \begin{bmatrix} p_y q_z - p_z q_y \\ p_z q_x - p_x q_z \\ p_x q_y - p_y q_x \end{bmatrix} \quad (\text{C.5})$$

This can be derived by writing the quaternions in extended complex form and expanding the product. The quaternion product is not commutative:

$$p \odot q \neq q \odot p \quad (\text{C.6})$$

Matrix representation of quaternion product is:

$$p \odot q = Q(q)p = \begin{bmatrix} q_w & -q_x & -q_y & -q_z \\ q_x & q_w & q_z & -q_y \\ q_y & -q_z & q_w & q_x \\ q_z & q_y & -q_x & q_w \end{bmatrix} \begin{bmatrix} p_w \\ p_x \\ p_y \\ p_z \end{bmatrix} \quad (\text{C.7})$$

or equivalently as

$$p \odot q = Q(p)q = \begin{bmatrix} p_w & -p_x & -p_y & -p_z \\ p_x & p_w & -p_z & p_y \\ p_y & p_z & p_w & -p_x \\ p_z & -p_y & p_x & p_w \end{bmatrix} \begin{bmatrix} q_w \\ q_x \\ q_y \\ q_z \end{bmatrix} \quad (\text{C.8})$$

The Jacobian of (C.7) with respect to p :

$$\frac{\partial(p \odot q)}{\partial p} = \begin{bmatrix} q_w & -q_x & -q_y & -q_z \\ q_x & q_w & q_z & -q_y \\ q_y & -q_z & q_w & q_x \\ q_z & q_y & -q_x & q_w \end{bmatrix} \quad (\text{C.9})$$

The Jacobian of (C.8) with respect to q :

$$\frac{\partial(p \odot q)}{\partial q} = \begin{bmatrix} p_w & -p_x & -p_y & -p_z \\ p_x & p_w & -p_z & p_y \\ p_y & p_z & p_w & -p_x \\ p_z & -p_y & p_x & p_w \end{bmatrix} \quad (\text{C.10})$$

Quaternion length:

The length of a quaternion is the same as the norm of the corresponding vector:

$$\|q\| = \sqrt{q \odot q^*} = \sqrt{q_w^2 + q_x^2 + q_y^2 + q_z^2} \quad (\text{C.11})$$

Quaternion Inverse:

The inverse of quaternion q with respect to the quaternion product is given as:

$$q^{-1} = \frac{q^*}{\|q\|^2} \quad (\text{C.12})$$

The Jacobian of (C.12) with respect to q :

$$\frac{\partial q^{-1}}{\partial q} = \text{diag}([1 \quad -1 \quad -1 \quad -1]) \quad (\text{C.13})$$

Quaternion Conversion to Rotation Matrix:

A unit quaternion can be converted into equivalent rotation matrix as follows:

$$Q(q) = \begin{bmatrix} (q_w^2 + q_x^2 - q_y^2 - q_z^2) & 2(q_x q_y - q_z q_w) & 2(q_z q_x + q_w q_y) \\ 2(q_x q_y + q_w q_z) & (q_w^2 - q_x^2 + q_y^2 - q_z^2) & 2(q_y q_z - q_w q_x) \\ 2(q_z q_x - q_w q_y) & 2(q_y q_z + q_w q_x) & (q_w^2 - q_x^2 - q_y^2 + q_z^2) \end{bmatrix} \quad (\text{C.14})$$

The rotation of a point m :

$$Q(q)m = \begin{bmatrix} (q_w^2 + q_x^2 - q_y^2 - q_z^2) & 2(q_x q_y - q_z q_w) & 2(q_z q_x + q_w q_y) \\ 2(q_x q_y + q_w q_z) & (q_w^2 - q_x^2 + q_y^2 - q_z^2) & 2(q_y q_z - q_w q_x) \\ 2(q_z q_x - q_w q_y) & 2(q_y q_z + q_w q_x) & (q_w^2 - q_x^2 - q_y^2 + q_z^2) \end{bmatrix} \begin{bmatrix} m_x \\ m_y \\ m_z \end{bmatrix} \quad (\text{C.15})$$

The compact version of (C.15) is:

$$Q(q)m = \begin{bmatrix} (q_w^2 + q_x^2 - q_y^2 - q_z^2)m_x + 2(q_x q_y - q_z q_w)m_y + 2(q_z q_x + q_w q_y)m_z \\ 2(q_x q_y + q_w q_z)m_x + (q_w^2 - q_x^2 + q_y^2 - q_z^2)m_y + 2(q_y q_z - q_w q_x)m_z \\ 2(q_z q_x - q_w q_y)m_x + 2(q_y q_z + q_w q_x)m_y + (q_w^2 - q_x^2 - q_y^2 + q_z^2)m_z \end{bmatrix} \quad (\text{C.16})$$

The Jacobian of (C.16) with respect to q is given by the 3x4 matrix:

$$\frac{\partial Q(q)m}{\partial q} = \begin{bmatrix} d_0 & d_1 & d_2 & d_3 \\ -d_3 & -d_2 & d_1 & d_0 \\ d_2 & -d_3 & -d_0 & d_1 \end{bmatrix} \quad \begin{aligned} d_0 &= 2(q_w m_x - q_z m_y + q_y m_z) \\ d_1 &= 2(q_x m_x + q_y m_y + q_z m_z) \\ d_2 &= 2(-q_y m_x + q_x m_y + q_w m_z) \\ d_3 &= 2(-q_z m_x - q_w m_y + q_x m_z) \end{aligned} \quad (\text{C.17})$$

Quaternions and Angular Velocities:

The time evolution of time varying quaternion with angular velocity is given by the differential equation:

$$\frac{dq_{ab,t}}{dt} = \frac{1}{2} w_{a,t}^{ab} \odot q_{ab,t} \quad (\text{C.18})$$

where $w_{a,t}^{ab}$ is the quaternion defined as the instantaneous angular velocity from coordinate frame b to coordinate frame a expressed in coordinate frame a :

$$w_{a,t}^{ab} = [0 \quad w_{ax,t}^{ab} \quad w_{ay,t}^{ab} \quad w_{az,t}^{ab}]^T \quad (\text{C.19})$$

The equation (C.18) can be also written in form:

$$\frac{dq_{ab,t}}{dt} = Q(w)q_{ab,t} \quad (\text{C.20})$$

where

$$Q(w) = \frac{1}{2} \begin{bmatrix} 0 & -w_{ax,t}^{ab} & -w_{ay,t}^{ab} & -w_{az,t}^{ab} \\ w_{ax,t}^{ab} & 0 & -w_{az,t}^{ab} & w_{ay,t}^{ab} \\ w_{ay,t}^{ab} & w_{az,t}^{ab} & 0 & -w_{ax,t}^{ab} \\ w_{az,t}^{ab} & -w_{ay,t}^{ab} & w_{ax,t}^{ab} & 0 \end{bmatrix} \quad (\text{C.21})$$

If the angular velocity $w_{a,t}^{ab}$ is time independent constant, then the solution the equation (C.18) with given initial conditions $q_{ab,t}$ can be written as

$$q_{ab,t+T} = \exp(T Q(w))q_{ab,t} \quad (\text{C.22})$$

where

$$\exp(T Q(w)) = \cos \left\| w \frac{T}{2} \right\| I + 2 \frac{\sin \left\| w \frac{T}{2} \right\|}{\|w\|} Q(w) \quad (\text{C.23})$$

The solution (C.22) can be expressed as the quaternion product

$$q_{ab,t+T} = \exp\left(\frac{T}{2} w_{a,t}^{ab}\right) \odot q_{ab,t} \quad (\text{C.24})$$

where

$$\exp(v)^T = \left[\cos\|v\| \quad \frac{v^T}{\|v\|} \sin\|v\| \right] \quad (\text{C.25})$$

Defining $v = \frac{T}{2} w_{a,t}^{ab}$, the Jacobian of (C.24) with respect to v is given:

$$\frac{\partial(\exp(v)\odot q)}{\partial v} = \frac{\partial(\exp(v)\odot q)}{\partial \exp(v)} \frac{\partial \exp(v)}{\partial v} \quad (\text{C.26})$$

$$\begin{bmatrix} q_w & -q_x & -q_y & -q_z \\ q_x & q_w & q_z & -q_y \\ q_y & -q_z & q_w & q_x \\ q_z & q_y & -q_x & q_w \end{bmatrix} \begin{bmatrix} \frac{1}{\|v\|} [I_3 & -\frac{vv^T}{\|v\|^2}] \sin(\|v\|) + \frac{vv^T}{\|v\|^2} \cos(\|v\|) \end{bmatrix}$$

finally

$$\frac{\partial(\exp(v)\odot q)}{\partial v} = \begin{bmatrix} q_w & -q_x & -q_y & -q_z \\ q_x & q_w & q_z & -q_y \\ q_y & -q_z & q_w & q_x \\ q_z & q_y & -q_x & q_w \end{bmatrix} \begin{bmatrix} -\frac{v^T}{\|v\|} \sin(\|v\|) \\ \frac{1}{\|v\|} [I_3 & -\frac{vv^T}{\|v\|^2}] \sin(\|v\|) + \frac{vv^T}{\|v\|^2} \cos(\|v\|) \end{bmatrix} \quad (\text{C.27})$$

Quaternion from Euler Angles:

By combining the quaternion representations of the Euler rotations:

$$q = \begin{bmatrix} \cos(\varphi/2)\cos(\theta/2)\cos(\psi/2) + \sin(\varphi/2)\sin(\theta/2)\sin(\psi/2) \\ \sin(\varphi/2)\cos(\theta/2)\cos(\psi/2) - \cos(\varphi/2)\sin(\theta/2)\sin(\psi/2) \\ \cos(\varphi/2)\sin(\theta/2)\cos(\psi/2) + \sin(\varphi/2)\cos(\theta/2)\sin(\psi/2) \\ \cos(\varphi/2)\cos(\theta/2)\sin(\psi/2) - \sin(\varphi/2)\sin(\theta/2)\cos(\psi/2) \end{bmatrix} \quad (\text{C.28})$$

Defining $v = [\varphi \quad \theta \quad \psi]^T$, the Jacobian of (C.28) with respect to v is given:

$$\frac{\partial q}{\partial v} = \begin{bmatrix} A_\varphi & A_\theta & A_\psi \\ B_\varphi & B_\theta & B_\psi \\ C_\varphi & C_\theta & C_\psi \\ D_\varphi & D_\theta & D_\psi \end{bmatrix} \quad (\text{C.29})$$

with

$$A_\varphi = 1/2 (-\sin(\varphi/2)\cos(\theta/2)\cos(\psi/2) + \cos(\varphi/2)\sin(\theta/2)\sin(\psi/2)) \quad (\text{C.30a})$$

$$A_\theta = 1/2 (-\cos(\varphi/2)\sin(\theta/2)\cos(\psi/2) + \sin(\varphi/2)\cos(\theta/2)\sin(\psi/2)) \quad (\text{C.30b})$$

$$A_\psi = 1/2 (-\cos(\varphi/2)\cos(\theta/2)\sin(\psi/2) + \sin(\varphi/2)\sin(\theta/2)\cos(\psi/2)) \quad (\text{C.30c})$$

and

$$B_\varphi = 1/2 (\cos(\varphi/2)\cos(\theta/2)\cos(\psi/2) + \sin(\varphi/2)\sin(\theta/2)\sin(\psi/2)) \quad (\text{C.31a})$$

$$B_\theta = 1/2 (-\sin(\varphi/2)\sin(\theta/2)\cos(\psi/2) - \cos(\varphi/2)\cos(\theta/2)\sin(\psi/2)) \quad (\text{C.31b})$$

$$B_\psi = 1/2 (-\sin(\varphi/2)\cos(\theta/2)\sin(\psi/2) - \cos(\varphi/2)\sin(\theta/2)\cos(\psi/2)) \quad (\text{C.31c})$$

and

$$C_\varphi = 1/2 (-\sin(\varphi/2)\sin(\theta/2)\cos(\psi/2) + \cos(\varphi/2)\cos(\theta/2)\sin(\psi/2)) \quad (\text{C.32a})$$

$$C_\theta = 1/2 (\cos(\varphi/2)\cos(\theta/2)\cos(\psi/2) - \sin(\varphi/2)\sin(\theta/2)\sin(\psi/2)) \quad (\text{C.32b})$$

$$C_\psi = 1/2 (-\cos(\varphi/2)\sin(\theta/2)\sin(\psi/2) + \sin(\varphi/2)\cos(\theta/2)\cos(\psi/2)) \quad (\text{C.32c})$$

and

$$D_\varphi = 1/2 (-\sin(\varphi/2)\cos(\theta/2)\sin(\psi/2) - \cos(\varphi/2)\sin(\theta/2)\cos(\psi/2)) \quad (\text{C.33a})$$

$$D_\theta = 1/2 (-\cos(\varphi/2)\sin(\theta/2)\sin(\psi/2) - \sin(\varphi/2)\cos(\theta/2)\cos(\psi/2)) \quad (\text{C.33b})$$

$$D_\psi = 1/2 (\cos(\varphi/2)\cos(\theta/2)\cos(\psi/2) + \sin(\varphi/2)\sin(\theta/2)\sin(\psi/2)) \quad (\text{C.33c})$$

CURRICULUM VITAE

Full Name : Nuri Özer

Address : Rumelifeneri Yolu, Medyakent Sitesi, B/3, D:12, Sarıyer/İSTANBUL

Birth Place / Date : Emet / 01.04.1986

Foreign Languages : English (Advanced)

High School : Kütahya Anatolian Teacher Training High School, 2003

B. Sc : Electrical & Electronics Engineering, Koç University, 2008

M. Sc : Electrical & Electronics Engineering, Bahçeşehir University, 2011

Name Of Institute : Natural And Applied Sciences

Name Of Program : Electrical And Electronics Engineering,
(Embedded Video Systems - Video Track)

Work Experience :

January 2011-Present

Vestek R&D Corp., Istanbul, Full-time Senior Design Engineer at Vestek R&D Corp.

June 2008-January 2011

Vestek R&D Corp., Istanbul, Full-time Design Engineer at Vestek R&D Corp.

September 2007-June2008

Vestek R&D Corp., Istanbul, Part-time Design Engineer at Vestek R&D Corp.

Fall 2007

Koç University, Istanbul, Undergraduate Senior Design Project on Super Resolution in Corporate Project with Prof. Dr. A. Murat Tekalp and TÜBİTAK

Summer 2007

Vestek R&D Corp., Istanbul, Internship at Vestek R&D Corp.

Fall 2006-Summer 2008

Koç University Multimedia, Vision & Graphics Laboratory, Istanbul, Undergraduate Assistant Student of Prof. Dr. A. Murat TEKALP, Engineering Department

Summer 2006

Aselsan A.Ş., Ankara, Internship at Aselsan Inc. Microelectronics, Guidance and Electro-Optics Division

Fall 2005-Spring 2006

Koç University Optical Microsystems Laboratory, Istanbul, Undergraduate Assistant Student of Assist. Prof. Hakan ÜREY, Engineering Department

Fall 2004-Spring 2005

Computer Information Technology (CIT), Istanbul. Assistant Student

Fall 2003-Spring 2004

Suna Kıraç Library, Istanbul, Assistant Student, Science & Math Department