

Jump Parameter Estimation with Low Cost MEMS Sensors and GPS for Action Sports Goggles

by

Fazle Sadi

B.Sc., Bangladesh University of Engineering and Technology, 2009

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF
THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF APPLIED SCIENCE

in

The College of Graduate Studies

(Electrical Engineering)

THE UNIVERSITY OF BRITISH COLUMBIA

(Okanagan)

August 2011

© Fazle Sadi 2011

Abstract

Algorithms to detect athletic jumps and to determine in real-time performance parameters such as jump air time (AT), horizontal distance, height and drop, are developed in this work. These algorithms are customized to be implemented onboard action sports goggles developed by Recon Instruments Ltd. These goggles are equipped with low cost micro-electro-mechanical inertial sensors and a single point GPS receiver which feed raw data to the algorithms. The micro-LCD display system in the goggles displays jump statistics to the user wearing the goggles. Two novel methods, namely WMCM (Windowed Mean Canceled Multiplication) and PFAD (Preceding and Following Acceleration Difference), are introduced for jump detection using accelerometer data. Four characteristic points in the resultant acceleration data are selected as the AT defining epochs for a jump. A novel threshold independent, probabilistic method using MADM (Multiple Attribute Decision Making) and the Closest Peak method are proposed to detect these characteristic points and determine the corresponding AT of a jump. A GPS/INS integration algorithm is developed to determine jump horizontal distance, height and drop. A novel sensor error compensation scheme is developed using sensor fusion and Linear Kalman Filters (LKF). The LKF parameters are varied to address the fluctuating dynamics of the athlete during a jump. The Extended Kalman Filter (EKF) used for GPS/INS integration has an observation vector augmented with sensor error measurements derived from sensor fusion.

The performance of the proposed algorithms was evaluated through experimental field tests. The proposed jump detection algorithm successfully detected 92% of the jumps performed by a snowboarder wearing the goggles whereas the current Recon algorithm only detects 60%. The AT determination algorithm exhibited an average error of 0.033 s (4.8%) which is well within the accuracy requirement of Recon, ± 0.1 s, and betters the current Recon algorithm which has an average error of 0.111 s (8.4%). For determination of jump horizontal distance, height and drop, the proposed algorithm has an error of 14.34 cm (5.55%), 1.56 cm (38.21%) and 6.71 cm (9.43%) respectively. The accuracy achieved is deemed to fulfill expectations of both recreational and professional athletes.

Table of Contents

Abstract	ii
Table of Contents	iii
List of Tables	vi
List of Figures	vii
Nomenclature	x
Notations and Abbreviations	xvii
1 Introduction	1
1.1 Context	1
1.2 Recon Action Sports Goggles	3
1.2.1 Key Performance Variables for Jumps	4
1.2.2 Jump Categories	7
1.3 Objective	9
1.4 Methodology	11
1.5 Thesis Outline	11
2 Literature Review	12
2.1 In Field Sports Performance Monitoring	12
2.1.1 General Strategies	12
2.1.2 Potential of Micro-Technology	13
2.2 Jump Detection and Air Time Calculation	14
2.2.1 Inertial Sensor Based Techniques	15
2.2.2 Non-Inertial Sensor Based Techniques	17
2.2.3 Application of Wavelets	18
2.2.4 Recon Application: Feasibility of Current Methods	19
2.3 Jump Horizontal Distance, Height and Drop Detection Strategies	21

Table of Contents

2.3.1	Imagery Based	21
2.3.2	Satellite Based	23
2.3.3	Satellite and Inertial Navigation Based	23
2.3.4	Alternative Techniques	24
2.4	Summary	24
3	Jump Detection and Air Time Determination	25
3.1	Sensor Selection	25
3.2	Accelerometer Data	25
3.3	Fast Fourier Transform	26
3.4	Sensor Correlation and Covariance	31
3.5	Windowed Mean Canceled Multiplication	36
3.6	Preceding and Following Acceleration Difference	37
3.7	Jump Detection	39
3.7.1	Detection through WMCM	40
3.7.2	Validity Check through PFAD	41
3.7.3	Jump Indication Categories	43
3.8	Air Time Determination	51
3.8.1	Characteristic Acceleration Points	51
3.8.2	Primary and Secondary Detection	53
3.8.3	Window Selection	55
3.8.4	Positive and Negative Peaks	55
3.8.5	Primary Detection	57
3.8.6	Secondary Detection	57
3.9	Sensor and Visual Air Time	65
3.10	Experimental Results and Comparison	66
3.10.1	Field Data Collection	66
3.10.2	Visual AT from SAT	66
3.10.3	Performance Comparison	68
3.11	Implementation Aspects for Other Devices/Applications	71
4	Trajectory Determination	75
4.1	Coordinate Frames	75
4.2	Rotation Matrix	76
4.3	Body Frame Attitude Computation	79
4.3.1	Rotation Matrix Update	79
4.3.2	Orientation Vector Computation	80
4.3.3	Coriolis Effect	81
4.4	Inertial Navigation Equations	82
4.5	Rotation Angle Computation	84

Table of Contents

4.5.1	Roll and Pitch from Accelerometer	84
4.5.2	Yaw from Magnetometer	87
4.5.3	Rotation Angles from Rotation Matrix	88
4.6	Sensor Errors	89
4.7	Proposed Algorithm	90
4.7.1	Acceleration Error Compensation	92
4.7.2	Angular Rate Error Compensation	94
4.7.3	INS Mechanization	97
4.7.4	Integration with GPS	98
4.7.5	Period Segmentation and Parameter Customization .	101
4.8	Field Experiments	104
4.9	Experimental Results	106
4.10	Summary	116
5	Conclusions and Recommendations	117
5.1	Objectives and Algorithm Performance	117
5.2	Methods and Key Findings	119
5.3	Recommendations	120
	Bibliography	122
	Appendix	128
	Appendix A Multiple Attribute Decision Making	128

List of Tables

3.1	Two sample t-Test for low impact JIs	49
3.2	Two sample t-Test for high impact JIs	50
3.3	Thresholds for secondary detection of JE_P and JE_N	60
3.4	Thresholds for secondary detection of JS_P and JS_N	60
3.5	Attribute type and weight for probabilistic secondary detection of JE_P and JE_N	63
3.6	Attribute type and weight for probabilistic secondary detection of JS_P and JS_N	63
3.7	Comparison of AT among Recon, Ripxx and the proposed algorithm.	72
3.8	Comparison of number of false processor wake up calls for Recon algorithm and the proposed algorithm.	73
3.9	Comparison of the number of false detected jumps + true undetected jumps among Recon, Ripxx and the proposed algorithm.	74
4.1	Quadrant information of roll from acceleration signs.	86
4.2	Quadrant information of pitch from acceleration signs.	87
4.3	Quadrant information of yaw from magnetic field component signs.	88
4.4	Comparison of jump horizontal distance determination performance between the proposed algorithm and Catapult GPS solution.	112
4.5	Comparison of jump height determination performance between the proposed algorithm and Catapult GPS solution.	113
4.6	Comparison of jump drop determination performance between the proposed algorithm and Catapult GPS solution.	114

List of Figures

1.1	Snowboard enthusiast catching some ‘big air’ [1].	1
1.2	Recon Goggle. Click on the image to see promotional video [1].	4
1.3	Micro-LCD display in Recon Action Sports Goggles [1]. . . .	5
1.4	Orientation of the three dimensional body/sensor frame. . . .	6
1.5	Real-time feedback in Recon goggles [1].	6
1.6	Categories of ski and snowboard jumping.	8
2.1	<i>StroMotion</i> highlights the essential moments in an athletic performance.	22
2.2	The movement and form of two performances is compared with <i>SimulCam</i>	22
3.1	Typical acceleration in the body frame.	27
3.2	Resultant acceleration a^b of the body frame during jump. . .	28
3.3	Disturbance and head motion in resultant acceleration a^b of the body frame during jump.	28
3.4	Sliding windowed FFT of a^b (resultant acceleration) near jump start region. Jump starts at 19.93 s.	29
3.5	Sliding windowed FFT of a^b (resultant acceleration) near jump end region. Jump ends at 21.62 s.	30
3.6	Sliding GT of a^b (resultant acceleration) near jump start region. Jump starts at 19.93 s.	31
3.7	Sliding GT of a^b (resultant acceleration) near jump end region. Jump ends at 21.62 s.	32
3.8	Cross-correlation at zero lag of windowed a_x^b and a_y^b	34
3.9	Covariance at zero lag of windowed a_x^b and a_y^b	35
3.10	WMCM of a_x^b , a_y^b and a_z^b	37
3.11	Windows for PFAD around a point of interest near take-off. .	38
3.12	PFAD value of the resultant acceleration a^b	39
3.13	Proposed jump detection method. Note: m_{xyz} is shifted by $-2 g^3$ for visual convenience.	42
3.14	High disturbance JI.	44

List of Figures

3.15 Low disturbance JI.	45
3.16 High impact JI.	46
3.17 Low impact JI.	47
3.18 Proposed jump detection method.	52
3.19 U-shaped region and characteristic points for an ideal jump. .	53
3.20 U-shaped region and characteristic points for a practical jump.	54
3.21 Window for characteristic acceleration points detection and potential positive peaks.	56
3.22 Primary detection.	58
3.23 (a) Potential positive peaks for JE_P . (b) Attribute value of the potential positive peaks.	59
3.24 Secondary detection of JE_P and JE_N with threshold based method.	61
3.25 D^*D^- plane for the secondary detection of JE_P	64
3.26 D_j^{OIRP} and rank of the top candidate peaks for JE_P	64
3.27 Secondary detection of JE_P and JE_N with probabilistic method.	65
3.28 Experimental video and captured sensor resultant acceleration.	67
3.29 Video AT vs SAT.	68
3.30 Error in AT comparison among the Recon, Ripxx and pro- posed algorithm.	70
4.1 Coordinate frames.	76
4.2 Coordinate frame before rotation.	78
4.3 Coordinate frame after rotations in ZYX sequence where roll = pitch = yaw = 90° . Click on the image to see sequential rotations.	78
4.4 Coordinate frame after rotations in ZXY sequence where roll = pitch = yaw = 90° . Click on the image to see sequential rotations.	79
4.5 Roll and pitch calculation from accelerometer data.	85
4.6 Error compensation block.	95
4.7 INS mechanization block.	95
4.8 Flow diagram of jump trajectory determination.	99
4.9 Period segmentation for trajectory determination.	102
4.10 Catapult Minimax unit affixed to the biker's helmet.	105
4.11 Catapult Minimax unit strapped to the Novatel GPS receiver antenna pole. [Photograph by Darren Handschuh]	106
4.12 Test jump performed by the athlete. [Photograph by Darren Handschuh]	107

List of Figures

4.13	Resultant acceleration of Catapult unit strapped to the antenna pole carried in backpack.	108
4.14	First example of jump trajectory determination with the proposed algorithm, with the Catapult GPS solution and with the Novatel receiver GPS solution.	109
4.15	Second example of jump trajectory determination with the proposed algorithm, with the Catapult GPS solution and with the Novatel receiver GPS solution.	110
4.16	Error comparison in KPV determination among Catapult GPS solution, GPS/INS integration excluding the error compensation block and proposed algorithm [Jump no. 1-7].	115
4.17	Error comparison in KPV determination among Catapult GPS solution, GPS/INS integration excluding the error compensation block and proposed algorithm [Jump no. 8-14].	115
A-1	TOPSIS method for MADM.	131
A-2	M-TOPSIS method for MADM.	132

Nomenclature

\bar{a}_{fw}^{bI}	Average of the signal a_b within the PFAD window following t_I
\bar{a}_{pw}^{bI}	Average of the signal a_b within the PFAD window preceding t_I
\bar{a}_{fw}^{bJI}	Average of the signal a_b within the PFAD window following t_{JI}
\bar{a}_{high}^{bJI}	Maximum between \bar{a}_{pw}^{bJI} and \bar{a}_{fw}^{bJI}
\bar{a}_{low}^{bJI}	Minimum between \bar{a}_{pw}^{bJI} and \bar{a}_{fw}^{bJI}
\bar{a}_{pw}^{bJI}	Average of the signal a_b within the PFAD window preceding t_{JI}
\bar{y}_1	Mean of the sample of \bar{a}_{high}^{bJI}
\bar{y}_2	Mean of the sample of \bar{a}_{low}^{bJI}
α	Integral of the angular rate vector
β^ω	Reciprocal of the process correlation time τ^ω
β^a	Reciprocal of the process correlation time τ^a
ω	Angular rate vector
ω_k	True angular rate of body frame with respect to navigation frame at epoch t_k
σ	Orientation vector
σ^ω	Amplitude of the angular rate bias power spectral density
σ^a	Amplitude of the acceleration bias power spectral density
τ^ω	Process correlation time of angular rate bias
τ^a	Process correlation time of acceleration bias
Δt	Sampling interval

Nomenclature

$\delta \mathbf{x}_{k+1}$	State vector of the Extended Kalman Filter (EKF) at epoch t_{k+1}
$\delta \mathbf{z}_{k+1}$	Observation vector of the EKF at epoch t_{k+1}
$\hat{\boldsymbol{\zeta}}_k$	estimated attitude error vector in Inertial Navigation Systems (INS) mechanization for update cycle starting at epoch t_K
$\hat{\mathbf{a}}_{k+1}$	Final estimate of the acceleration for update cycle starting at epoch t_{k+1}
$\hat{\mathbf{a}}_{k+1}^c$	Initial estimate of the acceleration for update cycle starting at epoch t_{k+1}
$\hat{\mathbf{b}}_k^{\omega E}$	Gyroscope bias error estimate from the EKF for GPS/INS integration in cycle starting at t_k
$\hat{\mathbf{b}}_k^{aE}$	Acceleration bias error estimate from the EKF for GPS/INS integration in cycle starting at t_k
$\hat{\mathbf{b}}_{k+1}^{\omega L}$	Gyroscope bias error estimate derived via sensor fusion in the cycle starting at t_{k+1}
$\hat{\mathbf{b}}_{k+1}^{aL}$	Acceleration bias error estimate derived via sensor fusion in cycle starting at t_{k+1}
$\hat{\mathbf{R}}_k$	final estimation of rotation matrix for update cycle starting at t_k
$\hat{\mathbf{r}}_{k+1}^n$	Estimated position vector in navigation frame from GPS/INS integration at epoch t_{k+1}
λ	Longitude
μ_1	Mean of the parameter \bar{a}_{high}^{bJI}
μ_2	Mean of the parameter \bar{a}_{low}^{bJI}
∇t_{ce}^l	Extension parameter of the characteristic window end
∇t_{cs}^l	Extension parameter of the characteristic window start
Ω_E	Earth's rotation rate
ϕ	Roll
ψ	Yaw

Nomenclature

\mathbf{a}^b	Acceleration of body frame observed from body frame
\mathbf{a}^b	The resultant acceleration in the body frame
\mathbf{a}^n	Acceleration of body frame observed from navigation frame
\mathbf{a}_k	True acceleration at epoch t_k
\mathbf{b}_k^ω	Bias in angular rate at epoch t_k
\mathbf{b}_k^a	Bias in acceleration at epoch t_k
\mathbf{g}^n	Normal gravity
\mathbf{g}_l^n	Plumb bob gravity
\mathbf{m}^b	Earth's magnetic field vector observed from body frame
\mathbf{q}_k^a	Process noise of acceleration with covariance matrix \mathbf{Q}_k^a
\mathbf{Q}_k^E	Covariance matrix of the EKF at epoch t_k
\mathbf{R}_b^n	Rotation matrix to convert from body frame to navigation frame
$\mathbf{R}_{\phi\theta}$	Rotation matrix to make the x-y plane of body frame coplanar with the x-y plane of local level frame
\mathbf{r}_{k+1}^n	True position vector in navigation frame at epoch t_{k+1}
\mathbf{r}_{k+1}^{nINS}	Position vector in navigation frame derived from INS equations at epoch t_{k+1}
\mathbf{u}_k^b	Incremental velocity of body frame observed from body frame for update interval starting at t_k
\mathbf{u}_k^n	Incremental velocity of body frame observed from navigation frame for update interval starting at t_k
\mathbf{v}^n	Velocity of the body frame with respect to the earth expressed in local level frame
\mathbf{v}_{k+1}^n	True velocity vector in navigation frame at epoch t_{k+1}
\mathbf{v}_{k+1}^{nINS}	Velocity vector in navigation derived from INS equations at epoch t_{k+1}

Nomenclature

\mathbf{w}_k^ω	Measured noise in angular rate at epoch t_k with covariance matrix \mathbf{R}_k^ω
\mathbf{w}_k^a	Measured noise in acceleration at epoch t_k with covariance matrix \mathbf{R}_k^a
$\mathbf{w}_{k+1}^{\omega L}$	Measurement noise of LKF ³ with covariance matrix $\mathbf{R}_{k+1}^{\omega L}$
\mathbf{w}_{k+1}^{aL}	Measurement noise of LKF ² with covariance matrix \mathbf{R}_{k+1}^{aL}
$\mathbf{z}\mathbf{a}_{k+1}^c$	Observation vector for LKF ¹
$\mathbf{z}\mathbf{b}_{k+1}^{aL}$	Observation vector for LKF ²
$\mathbf{z}\mathbf{w}_{k+1}^{\omega L}$	Observation vector for LKF ³
θ	Pitch
$\tilde{\boldsymbol{\omega}}_k$	Measured body frame angular rate with respect to navigation frame at epoch t_k
$\tilde{\boldsymbol{\omega}}_{ibk+1}^b$	Gyroscope reading for the update cycle starting at t_{k+1}
$\tilde{\mathbf{a}}_k$	Measured acceleration by accelerometer at epoch t_k
$\tilde{\mathbf{m}}_k$	Measured magnetometer readings at epoch t_k
$\tilde{\mathbf{R}}_k$	initial estimation of rotation matrix for update cycle starting at t_k
φ	Latitude
ζ	Significance level of the t -Test
A^*	Positive ideal solution of TOPSIS
A^-	Negative ideal solution of TOPSIS
A_j	j^{th} alternative of the MADM process
a_x^b	The forward/backward (anteroposterior) axis acceleration in the body frame
a_y^b	The right/left (mediolateral) axis acceleration in the body frame
a_z^b	The vertical (down/up) axis acceleration in the body frame
C_j^*	Similarities to positive ideal solution of j^{th} alternative

Nomenclature

$c_{xy}^w(kT)$	Covariance of windowed a_x^b and a_y^b with center epoch w and at lag k
D	Decision matrix
D_j^*	Distance from the positive ideal solution to j^{th} alternative
D_j^-	Distance from the negative ideal solution to j^{th} alternative
D_j^{OIRP}	Distance from the OIRP to j^{th} alternative
$da^b(t_I)$	PFAD of a^b at t_I
F	Sampling frequency
JE_N	Jump end negative peak
JE_P	Jump end positive peak
JS_N	Jump start negative peak
JS_P	Jump start positive peak
l	Extension counter index of the characteristic window
L_{fw}^I	Length of the PFAD window following t_I
L_{pw}^I	Length of the PFAD window preceding t_I
L_{fw}^{JI}	Length of the PFAD window following t_{JI}
L_{pw}^{JI}	Length of the PFAD window preceding t_{JI}
$m_{xyz}(t_s^j + nT)$	WMCM of a_x^b , a_y^b and a_z^b for a window starting at t_s^j
N	Number of samples in the data capture window
n_{max}^j	Index of the maximum WMCM value for the j^{th} window
n_1	Sample size of \bar{a}_{high}^{bJI}
n_2	Sample size of \bar{a}_{low}^{bJI}
NL_{fw}^I	Number of samples within a window of length L_{fw}^I
NL_{pw}^I	Number of samples within a window of length L_{pw}^I

Nomenclature

$r_{xy}^w(kT)$	Cross-correlation of windowed a_x^b and a_y^b with center epoch w and at lag k
$R_0(\varphi)$	Earth's radius at latitude φ
R_e	Earth's equatorial radius
R_p	Earth's polar radius
r_{ji}	Vector normalized decision matrix element for j^{th} alternative and i^{th} attribute
S_1^2	Sample variance of \bar{a}_{high}^{bJI}
S_2^2	Sample variance of \bar{a}_{low}^{bJI}
S_p^2	Estimate of the common variance of \bar{a}_{high}^{bJI} and \bar{a}_{low}^{bJI}
T	Sampling period
t_e^j	Ending epoch of the j^{th} data capture window
t_s^j	Beginning epoch of the j^{th} data capture window
t_c^N	Epochs of negative peak in characteristic window
t_c^P	Epochs of positive peak in characteristic window
t_e^{jump}	Jump end epoch
t_s^{AGP}	Start epoch of Acceleration Gain Period (AGP)
t_s^{jump}	Jump start epoch
t_{ce}	End epoch of the characteristic window
t_{cs}	Start epoch of the characteristic window
t_{JI}	Epoch of detected JI
th_m	Threshold for WMCM to detect JI
v_{ji}	Weighted normalized rating for j^{th} alternative and i^{th} attribute
w	Center epoch of the data capture window
w_i^c	Weight of i^{th} attribute

Nomenclature

X_h	Earth's magnetic field component along X -axis of the realigned body frame
x_{tilt}^b	The angle between horizontal plane and X^b
Y_h	Earth's magnetic field component along Y -axis of the realigned body frame
y_{tilt}^b	The angle between horizontal plane and Y^b
I	Identity matrix
t_{JE}^N	Epoch of JE_N
t_{JE}^P	Epoch of JE_P
t_{JS}^N	Epoch of JS_N
t_{JS}^P	Epoch of JS_P

Notations and Abbreviations

Convention

Matrices are represented by capital case bold letters and vectors are represented by lower case bold letters.

A superscript in a vector represents the particular frame in which the vector is represented.

Rotation matrices between two coordinate systems are defined with a superscript and a subscript denoting the two coordinate systems., e.g. \mathbf{R}_b^n transforms from body frame b to navigation frame n .

Angular rate between two frames is expressed by a angular rate vector, e.g. $\boldsymbol{\omega}_{ib}^b$ represents the angular rate between the inertial frame and the body frame observed from the body frame.

The skew symmetric matrix form of any vector $\boldsymbol{\omega} = [\omega_1 \omega_2 \omega_3]$ is represented as

$$[\boldsymbol{\omega} \times] = \begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix}. \quad (1)$$

Abbreviations

KPV Key Performance Variable

AT Air Time

TDR Total Degrees of Rotation

INS Inertial Navigation Systems

GPS Global Positioning System

Notations and Abbreviations

MEMS	Micro-Electro-Mechanical Systems
IMU	Inertial Measurement Unit
FFT	Fast Fourier Transform
PSD	Power Spectral Density
HMM	Hidden Markov Model
MADM	Multiple Attribute Decision Making
WMCM	Windowed Mean Canceled Multiplication
LKF	Linear Kalman Filter
EKF	Extended Kalman Filter
FFT	Fast Fourier Transform
PFAD	Preceding and Following Acceleration Difference
JI	Jump Indication
OIRP	Optimized Ideal Reference Point
SAT	Sensor Air Time
IAP	Initial Acceleration Period
AGP	Acceleration Gain Period
APP	Aero Phase Period

Chapter 1

Introduction

1.1 Context

Many sports enthusiasts crave the sensation of ‘catching the air’ while participating in such sports as snowboarding and skiing. ‘Catching air’ is a phrase often used to describe what happens when a snowboarder or skier travels fast enough over a jump to allow their board or skis to lift off the snow and into the air. Well executed and stylish routines consisting of complex aerial acrobatic maneuvers, as in Figure 1.1, are also highly appreciated in both sports. During competitions, judges often score competitors based upon the elevation, distance and time elapsed during the loft motion when ‘catching air’. The interest in witnessing and maneuvering complex jumps becomes apparent from the regular exclamation after a jump that he ‘caught’ some ‘big sky’, ‘big air’ or other assorted remarks when referring to the elevation, distance or time elapsed of the loft motion. Unfortunately, both the spectators and athletes themselves typically only have a qualitative sense as to the performance variables (e.g. jump height) without ever quantitatively knowing the statistics. However, many believe that objectivity in evaluation of these acrobatic performances could significantly enhance the experience for these athletes.

Enhancement of experience and entertainment is not the only motiva-



Figure 1.1: Snowboard enthusiast catching some ‘big air’ [1].

tion for finding objective information for evaluating sports performance. In competition environments, performances are also evaluated based upon subjective measures, referred to as ‘overall impression’, by a panel of judges [2]. During training sessions for these competitions, in the absence of appropriate measurement tools, athletes and coaches much rely largely upon evaluating performances based on their own experiences and feelings.

Recent analysis has revealed that sport specific Key Performance Variables (KPVs), e.g. loft duration and elevation for snowboard and ski jumping, strongly correlates with the subjectively judged scores in competition. Therefore, judges, coaches, and athletes could be greatly assisted by any complementary tools that objectively measure the sport specific KPVs. Even though video based assessment is widely used for training and evaluation, it has often proved to be misleading. Furthermore, video recording based performance analysis is generally time consuming and requires considerable ground infrastructure (e.g. video setup). It is also often very difficult to acquire and analyze information on the KPVs through the labor intensive post processing of the video data [3]. For instance, sophisticated video processing software is needed to determine the KPVs from the recorded data which is both resource and time consuming.

In more recent years, satellite based positioning, such as Global Positioning System (GPS), has been developed as a performance assessment technique to quantitatively measure sport specific variables. There are a number of commercial products available on the market, e.g. [4, 5], which provide performance parameters, such as distance, velocity and height profiles derived from satellite based position fixes. While these products may be effective for certain parameters, coaches and athletes are often also interested in other parameters such as orientation, degree of rotation, and epochs of take-off and landing which cannot be fully measured by satellite based monitoring systems. For example, in many instances the athlete’s environment may have naturally occurring obstructions that could block or attenuate satellite signals.

To over come many of these variables, which affect the ability to measure KPVs, Inertial Measurement Units (IMUs) are now being used along with GPS units for sports monitoring. Conventional GPS/INS equipment comprising dual-frequency GPS receivers and tactical grade INS, can provide highly accurate data (cm for positioning, cm/s for velocity and 1/100 degree for orientation). In the context of sports applications, however, such equipment is often not suitable because of adverse weight (a few kg) and expense (\geq €40,000). Recently, utilizing the advancements of micro technology, Micro-Electro-Mechanical Systems (MEMS) IMUs as well as low cost

L1 GPS receivers has entered the world of sports monitoring [6–9]. The benefit of such a GPS/INS system includes its global availability, light weight, low cost and fast setup. This system, therefore, diminishes the necessity for external infrastructure, such as video cameras [3].

The idea of measuring forces and position fix parameters with a GPS/INS integrated system to provide quantitative insight into athlete’s motion dynamics is academically sound. However, the assumption then declaration that this raw information will directly help the end users in functional evaluation and training seems to be an optimistic statement [2]. In addition, the interpretation of the system’s gathered information into something suitable for use by athletes and coaches constitutes a major component of any research focus. It is also imperative that consistency between the sporting communities expectations and the focus of scientific research be reached for the successful integration of any technological innovation into sports. Consequently, strong rationale has been provided for researchers to explore algorithms and systems that can bring cutting edge technology to the service of sports communities.

Recon Instruments, a Vancouver based company, has recently developed and is making Recon Action Sports Goggles, which are able to impart real-time information on KPVs to the individual wearing the goggles. A built-in MEMS GPS/INS unit is used to generate this data. The research conducted for this thesis is part of the Recon goggle development project, conducted in collaboration between The University of British Columbia and Recon Instruments. The goal of this collaborative study was to develop an innovative algorithm to derive sport specific KPVs in real-time onboard the goggles with the best achievable accuracy using the available sensors as well as to provide instantaneous feedback to individuals wearing the goggles.

1.2 Recon Action Sports Goggles

Recon Instruments has developed the world’s first GPS/INS enabled goggles with a head-mounted display system (Figure 1.2). The micro-LCD at the lower portion of the goggles, as shown in Figure 1.3, displays performance-enhancing data and statistics on KPVs. The graphics and optics of this direct-to-eye communication technology has made the display unobtrusive for front and peripheral vision. Furthermore, little interaction is needed during use to view the information making it applicable for use in fast-paced environments. The goggles can be used both in entertainment situations (for general interest during recreational skiing/snowboarding), or



Figure 1.2: Recon Goggle. Click on the image to see promotional video [1].

as a tool to aid in training for competitions.

The Recon goggles have sensors including a GPS receiver, a three axis MEMS accelerometer, a three axis MEMS gyroscope, a three axis MEMS magnetometer and an altimeter. The body frame coordinate system is shown in Figure 1.4, and the sensors axes are congruent with the body frame. The update rate of the GPS receiver position fixes is 1 Hz, and the data capture rate of the MEMS sensors is 100 Hz. The entire system is controlled with an ARM9TM processor.

Currently, these goggles provide real-time feedback including speed, latitude/longitude, altitude, vertical distance traveled, total distance traveled, chrono/stopwatch mode, a run-counter, temperature and time as shown in Figure 1.5. The goggles can be charged by USB, which can also be used to transfer data, along with the post-processing software included.

1.2.1 Key Performance Variables for Jumps

Recon Instruments has specially developed these goggles with sport performance quantifying capabilities specially relevant to the motions involved in snowboarding and skiing jumps. Four KPVs have been selected as the

1.2. Recon Action Sports Goggles



Figure 1.3: Micro-LCD display in Recon Action Sports Goggles [1].

1.2. Recon Action Sports Goggles

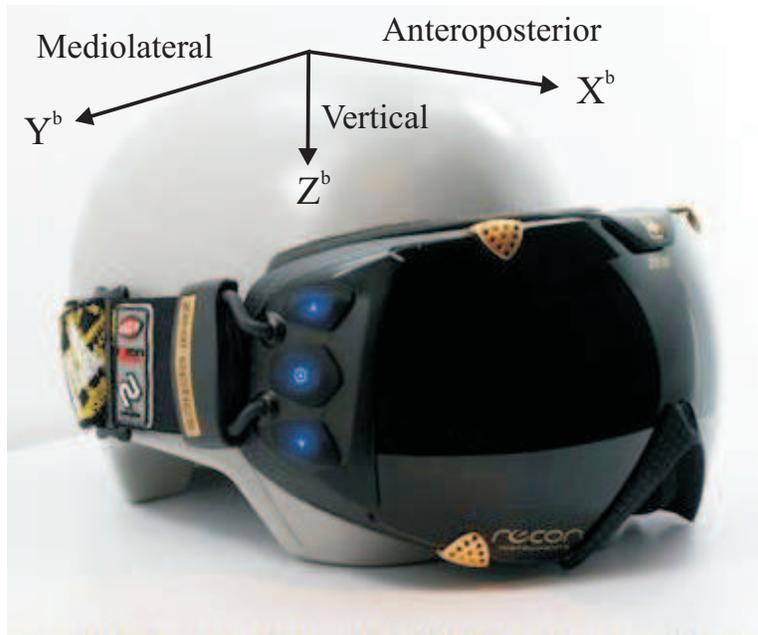


Figure 1.4: Orientation of the three dimensional body/sensor frame.



Figure 1.5: Real-time feedback in Recon goggles [1].

standard parameters of interest for the aerial acrobatic routines performed by skiers or snowboarders, especially during competitions. The KPVs are defined as follows:

1. **Air Time.** Total time traveled through the air from the time of take-off to the time of landing.
2. **Horizontal Distance.** The length over land traveled through the air from the time of take-off to the time of landing.
3. **Height.** Total vertical distance achieved from the point of take-off to the apex of the jump.
4. **Drop.** Total vertical distance achieved from the apex to the point of landing.

1.2.2 Jump Categories

Recon Instruments has also categorized the different types of ski and snowboard jumping into five major families. Each jump category possesses distinct characteristics with respect to the KPVs previously mentioned. The jump categories are illustrated in Figure 1.6 and defined as follows:

1. **Ollie.** Height and drop are equal, and the goggle wearer achieves distance and Air Time (AT).
2. **Step-up.** Height is larger than drop due to elevation change during the jump. AT and distance are both achieved, but AT is truncated.
3. **Standard.** This is the most common jump, where drop is larger than height due to elevation drop (common on jumps in mountain terrain). Longer AT and distances are achieved.
4. **Half-pipe.** AT is very large, and distance is very minimal. Drop is larger than height.
5. **Cliff-drop.** This is where there is very little to no height achieved, and drop is very large. AT is also very big, but distance is minimal.

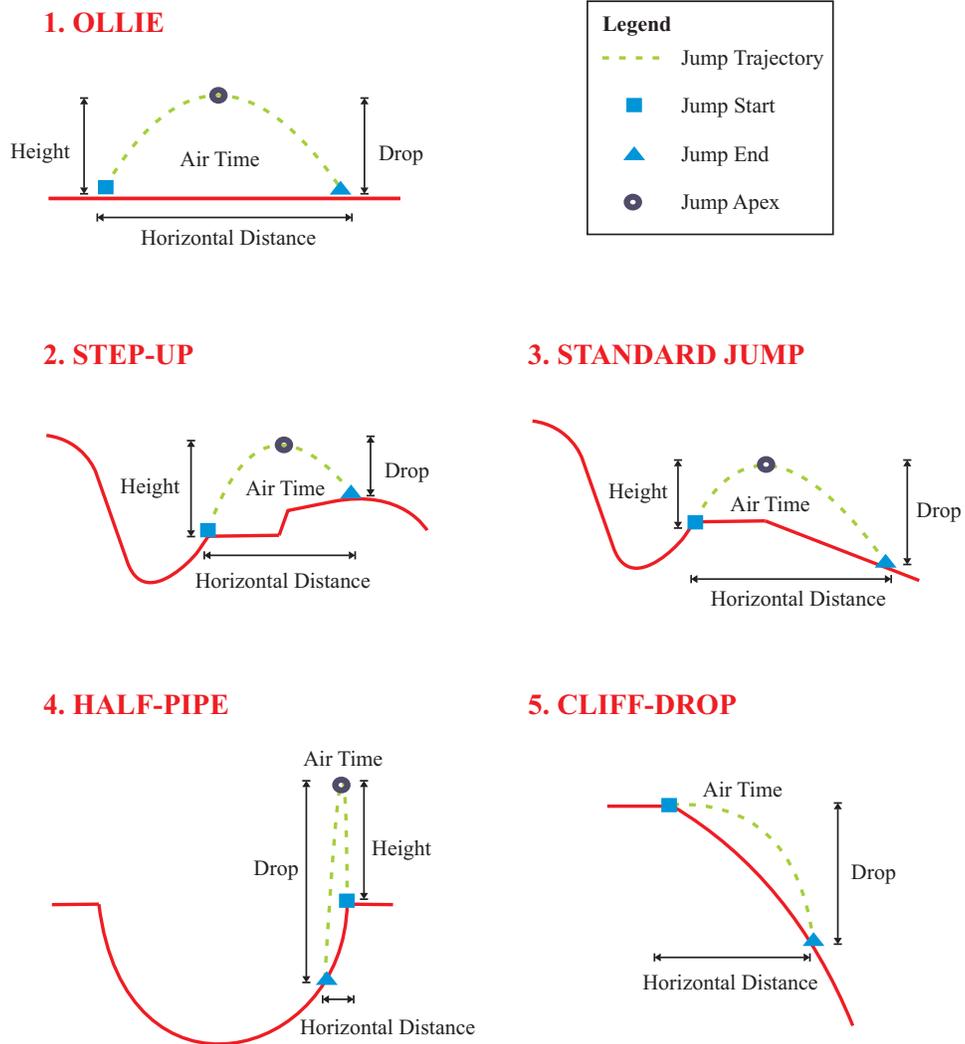


Figure 1.6: Categories of ski and snowboard jumping.

1.3 Objective

The objective of this research is to develop an algorithm for the on-board goggle processor to analyze all available sensor data. By doing so, the processor will then be able to successfully detect each of the above mentioned jump categories, with a minimum of false positives, and determine the standard KPVs as listed in Subsection 1.2.1. Recon has set the maximum allowable error in AT to be ± 0.1 s. For the other KPVs, Recon has set no specific requirements. However, errors should be negligible in magnitude in comparison with the KPVs measured such that the KPVs have practical application to the user. The key factors involved in this research are as follows:

- **Sensor Fusion.** The fundamental problem is how to combine the various sensor data, each of which has its own particular sensitivity, update rate, and noise, to achieve reliable conclusions. Whereas current work has concentrated on processing data from only two types of sensors, this project will combine data from five types of sensors - accelerometers, gyroscopes, magnetometers, GPS and altimeter. This task is complicated by the variability of situations a skier or snowboarder encounters over the course of a typical day with the goggles on. Therefore, the processing algorithm must cope with a large variety of dynamics.
- **Minimal Resources.** Since battery life is limited, power management is an important consideration in the goggle design. For instance, the minimum update rate from each the sensors required to reliably detect jumps must be determined. The lower the update rate, the greater the potential for battery and memory savings.
- **Online Processing.** The algorithm developed to process all sensor data and compute the parameters must be suitable for real-time operation and instantaneous feedback by the ARM9TM processor onboard the Recon goggles. The limited computing resources and battery life on the Recon goggles is a challenging factor in algorithm development.
- **Head Motion.** Since all sensors used are mounted within the goggles, these sensors will be subject to movement of the user's head. In other documented work, sensors are primarily mounted on the user's back, waist, legs or underfoot. Independent head motion could complicate (or aid) the interpretation of data from head-mounted sensors.

- **Unconventional Jumps.** Apart from the standard jumps, there are also many different ways that a skier or snowboarder can leave the ground that must be taken into account during jump detection. For instance, in some cases, the skier may never rise above the elevation of the take-off point. Flips, half-pipe moves, and other jump types each can result in radically different sensor readings.
- **Low Data Rate.** One of the major hardware limitations that the goggles have is the low rate (1 Hz) of position fixes from the GPS receiver whereas the typical ski jump duration is 0.5–2.0 s. Due to the scarcity of available data in such a brief operation time, conventional GPS dependent tracking algorithms are not suitable.
- **Noise.** MEMS sensors produce very noisy data. Depending upon the noise characteristics, denoising algorithms are also often computationally heavy. Therefore, focus should be given to the tradeoff between computational cost and denoising benefits.
- **Poor Sensitivity.** The on board altimeter is not sensitive enough to accurately capture typical jump height or drop of 0 – 2.0 m. The GPS receiver also does not have enough accuracy and sensitivity in the vertical axis. Consequently, none of the sensors can be used directly to compute the KPVs.
- **Accuracy.** The algorithm needs to satisfy the accuracy criteria demanded by sport professionals and recreational athletes. Recon specified that a ± 0.1 s error accuracy for AT must be met. Although Recon has not set any accuracy specification for jump horizontal distance, height and drop, error in the measurement should be constrained to what may render the estimated parameters to be useful.
- **Fast and Easy Operation.** Coaches and athletes often lack knowledge about emerging technologies. In addition, skiers and snowboarders often prefer to execute jumps without any interruptions. Therefore, it is important that the algorithm is independent of any tedious and repetitive routines such as manual sensor calibration, bias estimation or initial waiting period for filter convergence, which could interrupt one's performance.

1.4 Methodology

- **Jump Detection.** For jump detection, accelerometer data has been used by the novel Windowed Mean Canceled Multiplication (WMCM) method as well as by the novel Preceding and Following Acceleration Difference (PFAD) method.
- **Air Time.** To determine the AT, two methods are proposed. One is a threshold based technique and another is a probabilistic approach that uses no threshold. The novel probabilistic approach uses Multiple Attribute Decision Making (MADM).
- **Horizontal Distance, Height and Drop.** To determine the jump horizontal distance, height and drop, GPS aided and GPS integrated INS algorithms have been proposed. The GPS aided INS algorithm uses Linear Kalman Filters (LKFs) for continuous recursive error correction. For, the integration of GPS with INS, the EKF has been explored. Both the aided and integrated algorithm have been developed with direct and indirect state estimation techniques.

1.5 Thesis Outline

The thesis is structured chronologically. Chapter 2 summarizes all the relevant current research and development in sports monitoring. A detailed discussion is presented on the feasibility of the current technology satisfying the requirements of the Recon goggles. In Chapter 3, the developed jump detection and AT determination algorithm is presented with practical results. Chapter 4 presents the detailed architecture of the algorithm developed for determining jump horizontal distance, height and drop with performance evaluation. Concluding remarks and potential future research is presented in Chapter 5.

Chapter 2

Literature Review

The research in determining KPVs for Recon Action Sports Goggles comprises several fields of study. For jump detection and Air Time (AT) determination, the main focus is on the dynamics of the signals during jump start, aero phase and jump end epochs available from the sensors. On the other hand, the research in determining horizontal distance, height and drop mostly involves the kinematics of the sensor data. These two fields of study are distinct in signal processing, tools of manipulation and denoising aspects. Therefore, the relevant previous and current research to determine the mentioned KPVs will be discussed in two main categories. Firstly, the prevailing jump detection and AT calculation strategies will be presented. Secondly, the shortcomings and feasibility of these current methods for the Recon goggles application will be discussed. Finally, the current work pertinent to horizontal distance, height and drop determination will be presented in this chapter. The application of INS and GPS is widely practiced in similar applications. Therefore, a priority will be given to the possibilities and implications of the present INS and GPS involved strategies to meet the Recon goggle feature requirements. Preceding these discussions, general monitoring strategies for acrobatic maneuvers and the advantages of MEMS sensors in this particular application will also be presented.

2.1 In Field Sports Performance Monitoring

2.1.1 General Strategies

Research has shown that subjectively judged scores for athletes doing acrobatic maneuvers in sports like skiing and snowboarding has strong correlation with the KPVs such as AT, Total Degrees of Rotation (TDR), jump height, drop, horizontal distance, etc. As laboratory performance assessment for these sports is not feasible, a number of attempts have been made to develop and use current technologies to quantify sports specific performance variables in the field. Recently, sports scientists from the Australian Institute of Sport (AIS) have undertaken video based analysis of KPVs as-

sociated with elite half-pipe snowboard competitions [10]. The video based analysis is currently the most readily available method for coaches, sports scientists, competition judges and recreational users to gather the most objective information from acrobatic maneuvers as possible and assess style and run execution. However, video based analysis poses intrinsic time delay in information feedback and is a labour intensive form of analysis. Due to the requirement for synchronization of cameras, exterior orientation and determination of the cameras' position, all video based systems require a large amount of infrastructure as well as considerable setup time. Video based analysis has, therefore, not been adapted for everyday recreational use as well as for training purposes, despite its ability to provide objective observations. Novel strategies have been evolving to allow sensors to be more capable of interacting with the physical world, and to provide sport statistics in a convenient way.

An electromagnetic tracking system has been introduced in [11] to quantify the kinetics and kinematics associated with snowboarding. The joint moment torque and angular displacement information related to snowboard turns is monitored by post-processing the data captured on snow by the electromagnetic tracking system. When the object is placed inside varying yet controlled magnetic fields, voltages are induced in the sensor coils. These induced voltages are used by the measurement system to calculate the position and orientation of the object [12]. Additional research has been conducted to measure the kinematics of the skier and snowboarder feet and legs during maneuvers using a similar electromagnetic motion tracking system [13]. A dynamometric platform has been capitalized for field based data acquisition while snowboarding [14]. Without any interaction with current snowboard binding technology this dynamometric platform measures all the load components transmitted between the boots and snowboard bindings. Therefore, this technology facilitates the capability to acquire quantitative information on the torques and forces imparted by the boot on the bindings.

2.1.2 **Potential of Micro-Technology**

The research in the previously mentioned field based assessment methods is a worthwhile goal for collecting information on the kinematics of technical skill based sports, such as skiing and snowboarding. Unfortunately, certain intrinsic drawbacks preclude its growth and the wide application of these methods. The bulky electromagnetic tracking system [11] attached to the athlete's leg and the 10 kg measurement system which is carried by the researcher and tethered to the athlete is quite an obtrusive measurement

2.2. *Jump Detection and Air Time Calculation*

system. The weight of this monitoring system on the subject alone would be enough to constrict and alter their normal movement patterns, removing any relevance to real life snowboarding or skiing techniques. Therefore, a heavy monitoring system with tethered requirements is unsuitable for any data collection for elite level athletes training or competing in aerial acrobatic events with elements of risk involved [2].

With the advent of small scale technologies, a promising alternative, MEMS has drawn major attention from the sports science community due to its light weight and unobtrusive nature. MEMS sensors are capable of measuring human motion thousands of times per seconds in multiple axes without the problems associated with size and weight. Micro sensors are easy to attach anywhere on the athlete and do not limit the natural mobility. Moreover, onboard data storage technology along with the MEMS sensors diminishes the necessity for tethering bulky equipment. Therefore, MEMS inertial sensors, such as triaxial accelerometer, triaxial gyroscope, triaxial magnetometer etc., possess enormous potential in monitoring natural, unaltered and unobstructed acrobatic maneuvers of athletes in high risk sports in a realistic manner. There are challenges involved in using the capabilities of MEMS IMUs to calculate reliable and accurate objective information relevant to the end user [2]. Furthermore, limited processing capability and battery power cause the challenges to increase for small size wearable integrated systems dedicated to online processing, such as the Recon goggles.

2.2 Jump Detection and Air Time Calculation

One of the primary features of the Recon goggles is the ability to detect jumps while skiing and snowboarding as well as during acrobatic maneuvers in similar sports. These features are also essential for determining the expected KPVs. Once the jump start and jump end epochs are determined, the duration of the aero phase of the athlete, i.e. AT, can be derived easily. A number of different types of sensors are being used to detect the occurrence of the jump and determine the associated AT. Among MEMS inertial sensors, the accelerometer is significantly more popular for this purpose due to its capability of capturing the changes in an athlete's dynamics related to jumps.

2.2.1 Inertial Sensor Based Techniques

A significant amount of work has been done by Australian sports engineers to provide sport specific KPV information automatically and immediately post run for skiers and snowboarders. These developed technologies allow coaches and competition judges to automatically and objectively assess athletic performance in the environment in which half-pipe snowboarding takes place [2, 10, 15, 16]. For example, a signal processing technique was developed to calculate the AT associated with each individual aerial acrobatic manoeuvre performed during all half-pipe snowboard runs detected in [2]. MEMS accelerometers were used as data capturing sensors. All accelerometer units underwent a static calibration in three axes (up/down, forward/backward and left/right) prior to each data collection session aligning each axes of sensitivity with and against the direction of gravity. As reported in [2], half-pipe snowboarding generates a very distinct raw triaxial accelerometer data trace. The forward/backward and up/down acceleration axes were reported to display elevated accelerations during the aero phase of aerial acrobatic maneuvers. As a result of the substantial increases and decreases in acceleration throughout the performance, the power levels were reported to rise during the aero phase. Thus, jump locations were detected and jump windows were extracted by analyzing the entire trace for elevated levels of acceleration and utilizing a sliding Fast Fourier Transform (FFT) window and subsequent power analysis of average power levels.

A threshold based algorithm was used to identify the jumps within a trace by evaluating each FFT window for frequencies ranging from 0.25 – 0.85 Hz (as aerial acrobatic maneuvers occur relatively rhythmically with a period of 1.2 – 4.0 s) [2]. For detecting the AT, a two pass method was proposed in this study. In the first pass, the accelerometer data was quantized into three levels/states, namely maximum, minimum and transition. In the second pass, a technique was applied to discard false jumps detected by focusing on the sport specific possible durations of aerial acrobatic maneuvers. The duration of any component between 0.8 – 2.2 s was considered to be a valid aerial acrobatic maneuver. It was claimed by Harding et al. that this range would cover most aerial acrobatic ATs completed by half-pipe snowboard athletes. However, the threshold was selected heuristically for the algorithm and no general threshold selection scheme was presented in [2].

In a similar study, a preliminary automated feedback system based upon MEMS sensors was designed to calculate objective information on these sport specific KPVs [10]. The authors, Harding et al., argued that the results

2.2. *Jump Detection and Air Time Calculation*

would provide practical benefits for elite half-pipe snowboard training and current subjective judging protocols.

According to the method for detecting flight phase (i.e. the aero phase of the jump) in ski jumping proposed in [17], acceleration data, sampled at 200 Hz was stored in the digital memory and a FFT analysis was conducted on the overlapping Hanning windows comprising 256 points. It was reported that the maximum amplitude of the acceleration at the initial flight phase was clearly large (over 2.5 G) in each local coordinate system. In this study, angular rate/velocity of the athlete was also used to detect jumps. The peak angular velocity, captured by MEMS gyroscopes, at the initial flight phase was reported to be produced 0.2 s later than the peak acceleration. The peak power spectrums for acceleration in three directions just after take-off were observed at 9.37 Hz. Peak power spectrums for angular velocity just after take-off were reported to be produced at the frequencies of 9.37 Hz in the anteroposterior axis of the trunk, and at 8.59 Hz in the transverse axis and the longitudinal axis. These peak power spectrums were granted as the evidence of jump or flight of the athletes.

In [18], an unorthodox way of sensing loft was proposed. Single and multiple accelerometers were used to sense the vibration created by a snowboard, skis or mountain bike, moving along a surface. While moving, the voltage output provided by the accelerometer generates a noisy spectrum over time. It was reported that the vibrational spectrum generated during the aero phase for the athlete is generally much smoother than the spectrum generated while in motion on the surface. This is because while in air, the sporting equipment (i.e. skis or snowboard) is not subjected to the random vibrations of the road or ski slope. Accordingly, this relatively smooth spectrum was discerned from the rest of the spectrum by the microprocessor subsystem and evaluated to generate the AT. The time difference between the epochs of the end and start of the smooth spectrum was deemed as the airtime by the authors. Because the vibrational spectrum is affected by the particular activity of a user, e.g. standing still, included with the microprocessor subsystem of the invention was a means for assessing boundary conditions of the spectrum. For excluding certain conditions from determining the air/loft time, a lower boundary of 500 ms and an upper boundary of 5 s were fixed. Any events with elapsed time duration outside this boundary condition are excluded from the determination of loft time.

2.2.2 Non-Inertial Sensor Based Techniques

A number of non-inertial sensor based techniques for jump and airtime detection were proposed in [18]. The authors, Fleantov et al., proposed loft sensors constructed with the following technologies:

- a microphone assembly that senses a noise spectrum
- a switch that is responsive to the weight of the user of the vehicle
- a voltage-resistance sensor that generates a voltage indicative of a vehicle's speed.

The loft sensors proposed in this study sense a spectrum of information, e.g. a vibrational or sound spectrum and the microprocessor subsystem determines the relative change in the spectrum of information. Furthermore, the microprocessor interprets this relative change in the spectrum to determine the loft time. For example, the microphone based embodiment detects sound waves and provides a voltage output accordingly. Similar to the accelerometer based system described earlier, the microphone senses the vibration of the vehicle, e.g. snowboard or skis, moving along the surface. From the smoother spectrum during aero phase, the loft sensor senses the jump and determines the AT.

Another embodiment of the loft sensing system that is proposed in [18] is a switch that rests below the boot of the skier and that senses the pressure caused by the weight of the skier. When the skier is on the ground, the boot squeezes the switch, thereby closing it. The closed switch sends a distinct input to the microprocessor subsystem. When the skier jumps into the air, the switch opens up and renders the microprocessor to detect the openness of the switch. When the skier lands on the ground the switch closes again. The duration of the switch state being open is determined by the subsystem to calculate the loft time.

In the same paper [18], Fleantov et al. proposed a pad that can be placed under the skier's boot and changes capacitance as a function of the change in applied pressure. The capacitance changing element is then connected to a circuit, incorporating a monostable multivibrator through a resistor. The value of the resistor is fixed while the variable capacitance causes the pulse train generated by the monostable multivibrator to change its frequency depending on the pressure exerted on the pad. When the pulse train repetition rate increases, the value of the capacitance decreases and the skier's boot applies less pressure on the pad. This event marks the start of a jump

and beginning of the AT measurement. When the pulse repetition rate decreases, meaning a increase in capacitance, this indicates that the boot is applying greater pressure on the pad. This event marks the end of the jump as well as the end of the AT measurement.

2.2.3 Application of Wavelets

The acceleration of a snowboarder or skier at the beginning of a jump (take-off for the flight) and at the end of the jump (landing of the flight), results in a sudden change in acceleration. The detection of this abrupt change in signals and/or systems is a classic problem in signal processing, especially in the vast research field of detecting singularities or non-stationarity in a stationary signal. Cusp or jump location identification techniques of any signal encompass a wide variety of signal processing tools and demand very customized and application specific solutions. Among the various signal processing tools, wavelet analysis has proven to be useful in irregularity detection of a signal for numerous areas of application. Wavelet coefficients generally exhibit high peaks near abrupt changes or cusps of the underlying signal. Therefore, even in the presence of additive noise, by checking the absolute value of wavelet coefficients it is possible to detect discontinuities in an otherwise smooth curve.

The problem of detecting a mean value jump of a stationary random process was addressed in [19] by means of the wavelet transform. A matched filter based approach was also proposed in this study for which an ideal knowledge of the Power Spectral Density (PSD) of the signal was needed. However, this is not generally the case. Therefore, a second approach was proposed using the classic pattern recognition principle relying on the correlation between the signal and the signature. Typically, wavelet based statistical signal processing techniques model the wavelet coefficients as independent or jointly Gaussian. To address the non-Gaussian statistics encountered in real world signals, a new framework for statistical signal processing based on the wavelet domain Hidden Markov Model (HMM) was proposed in [20]. The efficient expectation maximization algorithm was also developed in this study for fitting the HMM to observational signal data. The new framework was claimed to be suitable for a wide range of applications, including signal estimation, detection, classification, prediction, and even synthesis.

A method of detecting the cusp or an incident in the traffic flow was presented in [21]. Unlike the existing wavelet incident detection algorithm, where the wavelet technique was utilized to denoise data before the data was input into an algorithm, features that were extracted from traffic mea-

measurements by using wavelet transformation were directly utilized in detecting changes in traffic flow. Using the Peaks Over Threshold modeling of the noisy wavelet transformations and extreme value theory, a method for detecting an arbitrary number of discontinuities in an unknown function observed with noise was proposed in [22].

2.2.4 Recon Application: Feasibility of Current Methods

To develop the jump detection and AT calculation algorithm for the Recon goggles, extensive insight can be achieved from the prevailing relevant studies as previously discussed, even though most of the techniques were devoted to particular applications different from the application considered here. However, the majority of the inventions were developed using common platforms, e.g. FFT. Unfortunately, none of the current techniques meet Recon's requirements entirely. The significant characteristics of current methods which are not conducive with the application of Recon goggles are as follows:

1. The Recon goggles demand a online (real-time and onboard) algorithm to detect jumps which should be computationally light to save battery power and instantaneous feedback. An FFT algorithm for jump detection, as proposed in [2, 10, 15, 17], requires continuous heavy processing of the acceleration data. Therefore, FFT analysis is not likely to be a good option for any online algorithm required by Recon even if it is useful for offline processing .
2. The use of PSD extracted from a sliding FFT window [2, 10, 15, 17] for detecting jump and AT is heavily dependent upon the window size and its position relative to the abrupt change in acceleration. The expected high power may not even be captured if the window size is not precisely accurate. However, no criterion was proposed for choosing the proper window length in the related papers. Moreover, high power level, due to the sudden shift in acceleration, may continue for a number of consecutive FFT windows. In such cases, determination of the proper epoch for jump start or jump end becomes ambiguous. These phenomena will be discussed in detail in Chapter 3.
3. In most of the accelerometer data based jump detection and AT determination techniques, sensor calibration and bias correction is necessary. However, sensor calibration before using a device like the Recon goggles is very inconvenient for recreational users, even for coaches and professional athletes.

4. Almost all the current methods use stringent thresholds for the acceleration or post processed data, which causes the algorithm to lose generality. For fixed thresholds to work, bias correction and calibration is also necessary. This is not suitable for devices like the Recon goggles which are targeted for users in different environmental conditions with ease of operation. Some of the studies, e.g. [2], used a very low threshold so that no event is missed. However, implications such as too many false detections are there for these sort of strategies.
5. Most of MEMS accelerometer based techniques [2, 10, 15, 17] work with raw acceleration data captured with individual accelerometers aligned in specific directions with respect to the athlete's body. Attaching the accelerometers in exact alignment with direction is itself fastidious. Moreover, the orientation of the body part to which the sensors are attached is not the same for all body parts even for any particular acrobatic maneuver. The magnitude of the readings from the sensor may drastically change with a slight change from the expected alignment of the sensors. This phenomenon poses high risk of failure to detect events for the fixed threshold and/or individual axis acceleration based algorithms.
6. Strategies like quantization, as in [2], can also cause information loss when trying to find the exact epoch for jump start or end.
7. Certain boundary conditions, e.g. 0.8 – 2.2 s in [2], 0.5 – 5.0 s in [18], were set to distinguish jumps from other events, such as abrupt head movements. However, small jumps with AT less than these lower boundaries are reported frequently, especially for nonprofessional athletes. Therefore, the use of these boundary conditions will limit the use of the Recon goggles.
8. Examples shown in [2, 10, 15, 17, 18] are mostly for very well behaved data. However, it can be observed that low cost IMU sensors produce very noisy readings. It is quite common for the low cost IMU sensors to produce readings with true features buried under the noise. Movement artifacts also arise frequently from the rapid movements of athletes to maintain balance or to add an extra component of style to their acrobatic performance. All these undesired signal attributes should be accounted for to develop any robust algorithm.
9. In [18], it was assumed that the spectrum of acceleration is always smooth during the flight phase of the athlete. However, this is often

not the case for the jumps with complex acrobatic maneuvers such as spinning and flipping. The extra acrobatic components during the aero phase will render the IMU sensors to lose their spectral smoothness.

10. The biggest challenge in implementing any wavelet based algorithm is the computational demand of the wavelet decomposition. For online processing of the KPVs it is required that the computation tool be power efficient with very little time lag in producing feedback which is not always the case for wavelet analysis.

2.3 Jump Horizontal Distance, Height and Drop Detection Strategies

The task of determining the horizontal distance, height and drop of jumps for outdoor sports disciplines actually falls under the research for trajectory determination of the athlete during the loft period. Trajectory determination in sports application is relatively more explored than the AT detection discussed earlier. Even though every sport has its own method for performance analysis depending on the unique dynamics, ergonomic requirements, and trajectory determination techniques of the sport, most have developed with similar techniques. The general strategies of trajectory determination are discussed below.

2.3.1 Imagery Based

Dartfish [23] has developed a powerful tool using cameras to provide qualitative information about athletic trajectories. Their *StroMotion* technology is able to create trajectory video footage revealing the evolution of an athlete's movement, technique, execution and tactics. It is possible to superimpose one athletic movement over another for precise comparisons using the *SimulCam* technology developed by Dartfish. The developed system utilizes invariant image retrieval techniques, which detect the common features in the images captured for two different athletes, so that the images of the two different trajectories can then be superimposed and compared. Figure 2.1 and Figure 2.2 show examples of these two technologies. These methods are only useful for relative analysis as quantitative data can not be extracted. However, stereo-camera systems are able to provide highly accurate, quantitative information about the 3D trajectory and motion of athletes [3]. In [24], trajectory analysis was performed using single camera and stereo-camera systems. Codamotion has developed systems with

2.3. Jump Horizontal Distance, Height and Drop Detection Strategies



Figure 2.1: *StroMotion* highlights the essential moments in an athletic performance.



Figure 2.2: The movement and form of two performances is compared with *SimulCam*.

infrared LEDs which work as active markers that are detectable by the cameras [25]. Passive marker systems for optical or infrared cameras have also been developed [26]. Only the markers are visible on the image in an infrared system. An advanced markerless system has been developed by Organicmotion [27]. For this system, volume models are fitted specially to the athlete. Each of these unique systems (active marker, passive marker and markerless) [25–27], are able to capture the 3D motion of the subject which can be used to derive the movement and trajectory of the athlete.

2.3.2 Satellite Based

KPVs such as jump horizontal distance, height and drop are directly related to the position information of the athlete. Therefore, satellite based positioning has been widely used in sports applications. Many commercial products are available that can provide the satellite navigation tracking information and/or KPVs derived from satellite positioning [4, 5, 28–33]. GPS has been used for trajectory determination in multiple sports disciplines such as skiing, ski jumping and snowboarding [3, 34, 35], water sports [36, 37] and many others.

Unfortunately, GPS signal reception is not ensured for athletes performing maneuvers in adverse environments where satellites may get masked due to naturally occurring obstructions, such as cliffs, and hills. Therefore, in most sport disciplines, continuous observation of the athlete’s performance cannot be guaranteed. Moreover, the accuracy of low cost single point receivers does not always meet the requirements. For example, measuring jump height or drop with the Recon goggles is not possible because of the poor accuracy of the integrated GPS receiver along the vertical axis. Highly accurate dual-frequency GPS receivers are also not feasible for most sports applications because of their larger size and cost. In addition, the low update rate of the GPS receiver precludes the measurement of KPVs during relatively short acrobatic maneuvers in such sports as ski jumping or snowboard half-pipe. For example, the 1 Hz update rate of the single point GPS receiver available in the Recon goggles is not suitable for measuring the horizontal distances for jumps with small AT such as ~ 0.5 s. Even for long jumps, the low update rate of the GPS receiver is unable to guarantee to record position fixes near the jump start or the jump end epochs with acceptable accuracy.

2.3.3 Satellite and Inertial Navigation Based

Although a MEMS IMU cannot provide standalone navigation, INS with GPS aiding or GPS integration can provide continuous reliable tracking for sports applications. Integration of INS with GPS potentially bridges the GPS outages along with compensating for the systematic inertial sensor errors. The combination of the high short-term (relative) accuracy of the INS results and longterm (absolute) accuracy of GPS results smears the variation in positioning performance and provides data at a high rate including good orientation estimates [3]. Therefore, GPS/INS integration has been widely used for reliable tracking such as in [38–42]. However, GPS/INS in-

tegration incurs a heavy computational load on the microprocessor of the system which is not always affordable.

2.3.4 Alternative Techniques

There are a few radio technologies that can be used for position fixing instead of satellite navigation such as Ultra-Wide Band (UWB) [43, 44] and Wireless Local Area Networks (WLAN) [45]. These systems can perform well for indoor applications where the GPS signal is unavailable. Unfortunately, these technologies only work for a short range and the infrastructure needed for them is considerable. Moreover, shadowing by the human body and multipath propagation limit the use of these technologies for performance evaluation of athletes [3].

2.4 Summary

Upon reviewing the above-mentioned systems it becomes apparent that none of these systems or their methods can be seen as a complete solution for the Recon goggle requirements. Indeed, every system and strategy has its advantages and drawbacks for sport application. In terms of available resources, observed parameters and desired accuracies, however, a GPS aided inertial sensor based power efficient algorithm is the most suitable solution for the Recon goggles.

Chapter 3

Jump Detection and Air Time Determination

In this chapter, sensor signals are investigated to develop an algorithm to detect jumps and the corresponding Air Time (AT) involved. The commonly applied FFT is also explored to detect jump indicative changes in the sensor signals. A novel Windowed Mean Canceled Multiplication (WMCM) method to detect jump occurrence is proposed. Another novel method to determine the AT is presented in the later part of the chapter. Experimental results and comparative analysis among the proposed algorithm, the current Recon goggle algorithm and the Ripxx [4] algorithm are presented.

3.1 Sensor Selection

Among all the sensors available in the Recon goggles, the accelerometers and gyroscopes measure the dynamics involved in the motion of the athlete. Accelerometers capture the change in acceleration involved in jump take-off and jump landing. The change in acceleration is very consistent and significant from jump to jump. On the other hand, the gyroscopes are able to record the high rate of angular movement of the head that is likely to occur during the start and end of the loft motion. However, the high angular rate of head movement is not guaranteed for all jumps. Moreover, the peak angular rate tends to vary its relative position with respect to jump end or start from jump to jump. Therefore, the accelerometers are deemed to be the most suitable sensor for jump detection and AT determination.

3.2 Accelerometer Data

Ideally, in static conditions such as standing still on the ground, the sensor body frame experiences antigravity (vertically upward), which has the magnitude of 1 g, where $g = 9.81 \text{ ms}^{-2}$. While in free fall, no acceleration, including antigravity, is experienced by the sensor body frame. Therefore,

3.3. Fast Fourier Transform

in an ideal noiseless condition the resultant body frame acceleration should be 1 g in static condition and 0 g in free fall. These ideal scenarios will help understanding the accelerometer data recorded while ski jumping.

An example of typical acceleration data captured by the triaxial MEMS accelerometer during a ski jump is shown in Figure 3.1. This data was recorded with Recon goggles which were mounted on the skier's head. The acceleration along the forward/backward (anteroposterior), right/left (mediolateral) and down/up (vertical) axes in the body frame are termed as a_x^b , a_y^b and a_z^b respectively. It is significant in the figure that a_x^b and a_y^b exhibit a relatively low value (~ 0 g) during the aero phase, i.e. while the athlete is in the air. Except for the aero phase these sensors report significant accelerations which cause the resultant acceleration to be near 1 g. The sudden change in accelerations are also noticeable at the start and end of the aero phase. It should be noted that the accelerations exhibited by the three axes of the body frame depend entirely on the orientation of the body frame. It is not guaranteed that any specific body frame axis will capture the sudden change in acceleration during the loft motion. However, the abrupt change in acceleration is always captured in the total resultant acceleration a^b as shown in Figure 3.2, where $a^b = \sqrt{(a_x^b)^2 + (a_y^b)^2 + (a_z^b)^2}$.

The resultant acceleration falling near 0 g while the skier is in the air, provides an indication of the free fall state or, in other words, jump occurrence. However, due to the presence of noise, head motion and disturbance signals from other activities, the detection of this drop in acceleration is not straight forward. A typical signal due to disturbance and head motion is shown in Figure 3.3. The similarities among the signal patterns during the time of these unwanted motions and jump are clearly noticeable. A significant amount of noise is also present in these low cost MEMS sensor signals.

3.3 Fast Fourier Transform

The sliding windowed Fast Fourier Transform (FFT) is widely proposed to detect the abrupt change in acceleration as in [2, 10, 15, 17]. Figure 3.4 shows the sliding windowed FFT of the jump start region of the resultant acceleration shown in Figure 3.2. Here, the FFT window length is 0.6 s and each window overlaps with the following by 0.4 s. The sampling rate of the data is 100 Hz. The time stamp indicated above each plot represents the epoch at which the sliding window begins. The FFT for several windows preceding and following the jump start region are shown. Even though fluctuating spectrums at higher frequencies are noticeable in the windows

3.3. Fast Fourier Transform

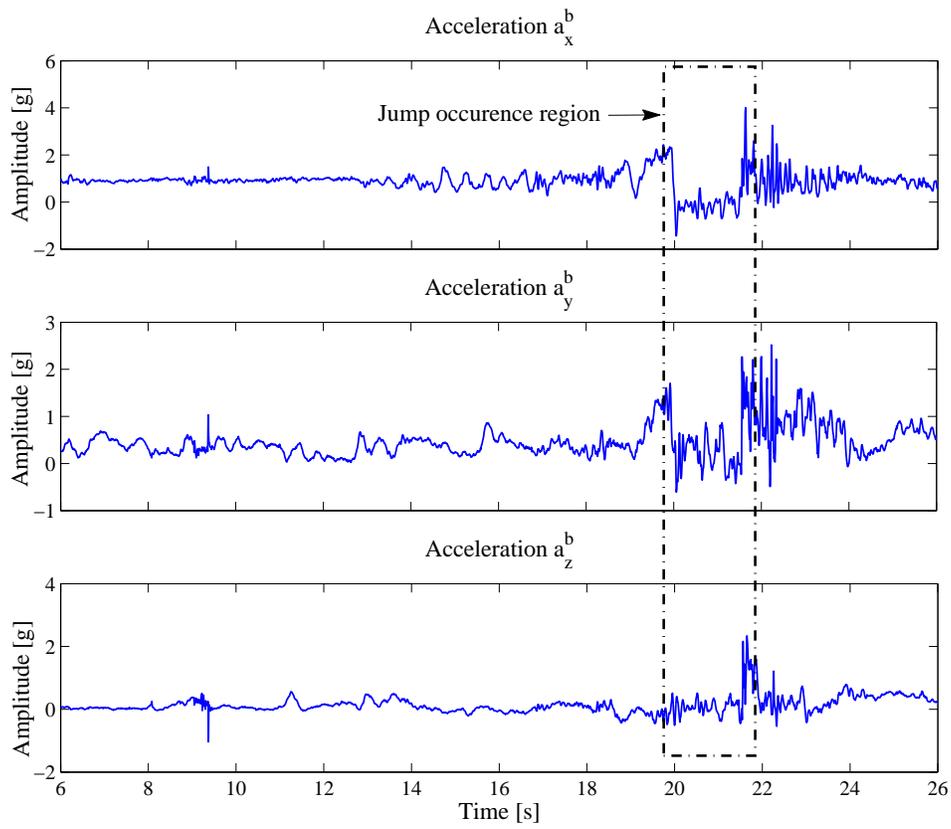


Figure 3.1: Typical acceleration in the body frame.

3.3. Fast Fourier Transform

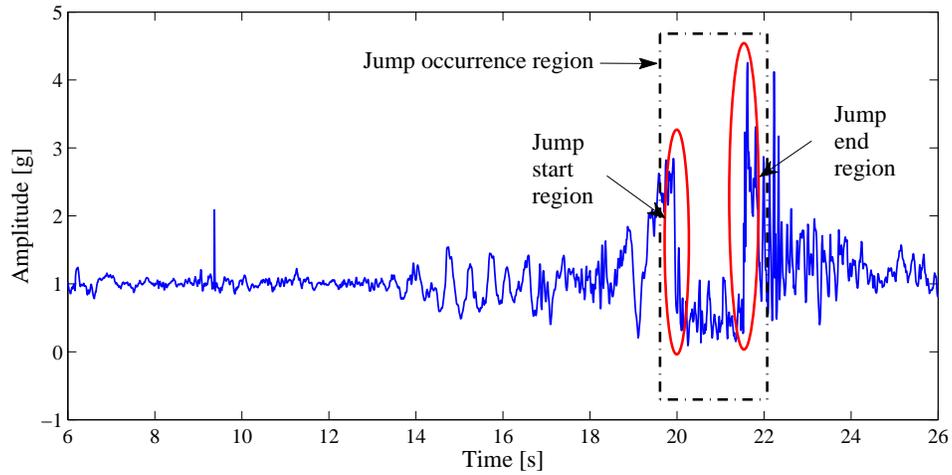


Figure 3.2: Resultant acceleration a^b of the body frame during jump.

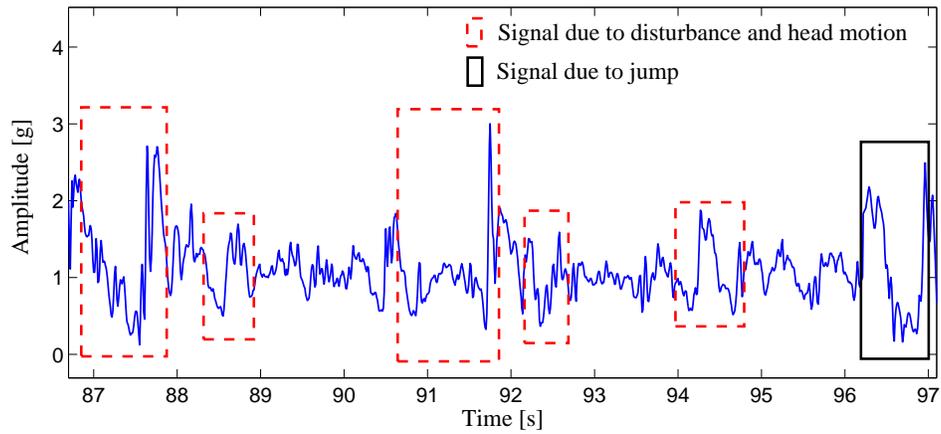


Figure 3.3: Disturbance and head motion in resultant acceleration a^b of the body frame during jump.

3.3. Fast Fourier Transform

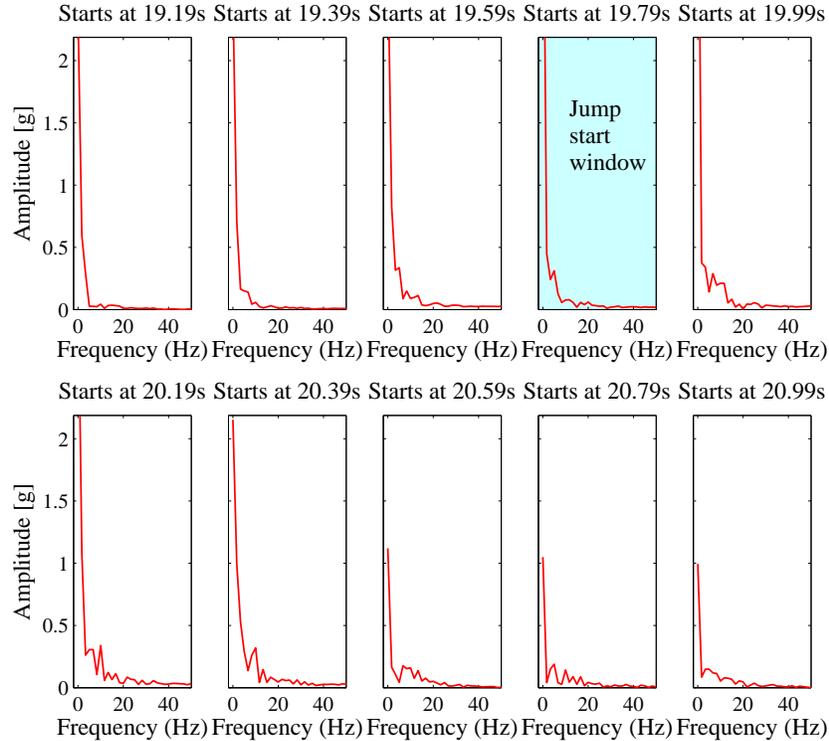


Figure 3.4: Sliding windowed FFT of a^b (resultant acceleration) near jump start region. Jump starts at 19.93 s.

neighboring that which contains the jump start, no distinct pattern in the spectrum can be observed for the jump start window. Similar observations can also be made for the sliding windowed FFT of the jump end region as shown in Figure 3.5.

The Gabor Transform (GT) [46] is explored as an alternative for the regular windowed FFT. In the GT, the data is multiplied with a Gaussian window before applying the FFT, whereas in the regular windowed FFT the data can be multiplied with a square window with the unity magnitude before applying the FFT. Figure 3.6 and 3.7 show the results of the sliding GT of the jump start and end region of the resultant acceleration shown in Figure 3.2. The spectrums are smoother than what is observed for the FFT. Still, the ripples at higher frequencies exhibit a random behavior near the jump start/end region without showing any consistent detectable pat-

3.3. Fast Fourier Transform

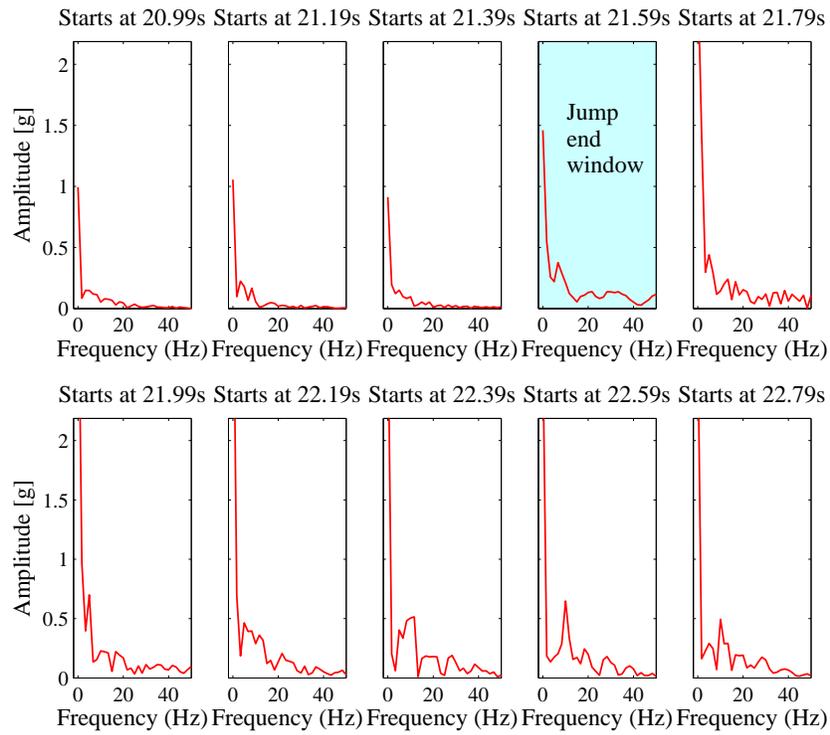


Figure 3.5: Sliding windowed FFT of a^b (resultant acceleration) near jump end region. Jump ends at 21.62 s.

3.4. Sensor Correlation and Covariance

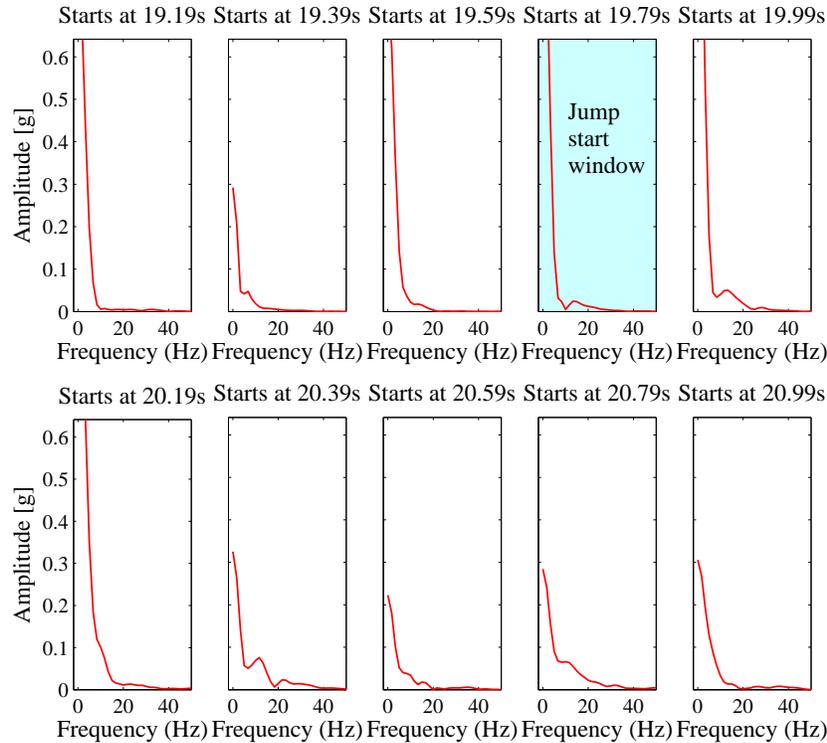


Figure 3.6: Sliding GT of a^b (resultant acceleration) near jump start region. Jump starts at 19.93 s.

tern. Even if the window for jump start/end is detected through the FFT or GT, the jump start/end epoch can only be detected with the very poor accuracy of the sliding window length, which, in this case is 0.6 s. The fluctuation spectrum patterns exhibited in and neighboring the jump start/end windows also vary significantly with the window size chosen and jump duration. Therefore, besides the heavy computational load, the inconsistent spectrum and poor accuracy provides enough rationale to discard the sliding windowed FFT or GT as a tool for jump detection for the Recon goggles.

3.4 Sensor Correlation and Covariance

In steady motion or static condition of the athlete the recorded accelerations in the three orthogonal axes of the body frame are understandably

3.4. Sensor Correlation and Covariance

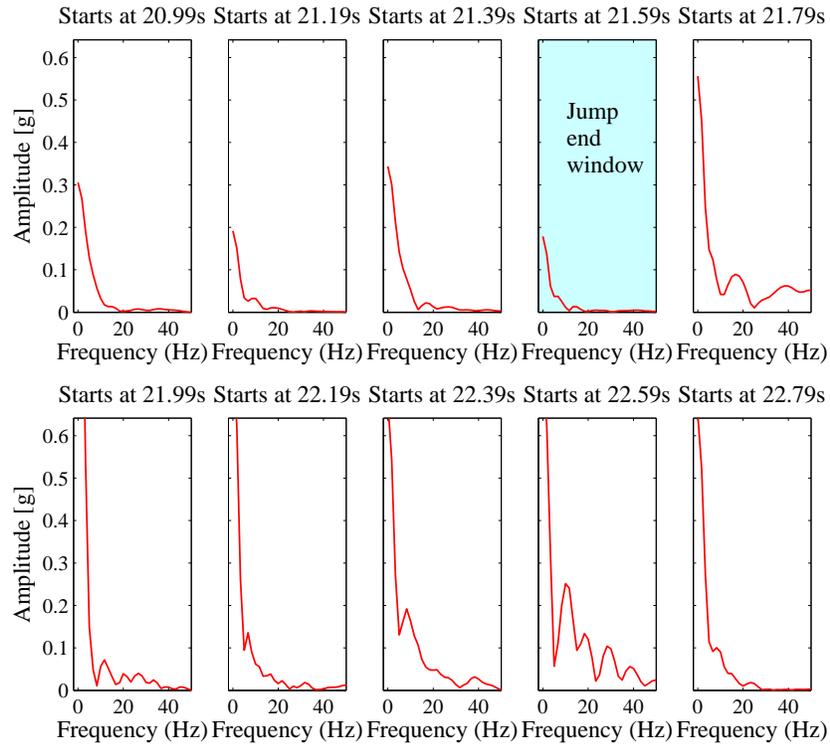


Figure 3.7: Sliding GT of a^b (resultant acceleration) near jump end region. Jump ends at 21.62 s.

3.4. Sensor Correlation and Covariance

uncorrelated due to the random nature of noise, head motion and fluctuation in the accelerations. However, it is virtually guaranteed that all three accelerometers in three orthogonal directions will experience a sudden impact if there is an abrupt change in motion dynamics. During the take-off for a jump, forces exerted by the athlete and the subsequent free fall state cause a rapid change in the athlete's dynamics. Hence, all the accelerometers are expected to experience a peak (positive or negative) value during this brief period of sudden change in acceleration. With similar reasoning, all the accelerometers are also expected to record a local maximum or minimum during the landing of the jump. Assuming this rationale to be valid, the following analysis of the accelerometer data is conducted.

The cross-correlation of two discrete time signals, e.g. a_x^b and a_y^b , can be defined as

$$r_{xy}(kT) = \sum_{n=-\infty}^{+\infty} a_x^b(nT)a_y^b(nT - kT), \quad k = 0, \pm 1, \pm 2, \dots \quad (3.1)$$

Here, the sampling period $T = \frac{1}{F}$ and the sampling frequency $F = 100$ Hz. The index k is the discrete lag parameter and kT is the time shift. For online signal processing, the data is captured as a window. Therefore, cross-correlation of the two signals captured in a window from t_s^j to t_e^j can be expressed as

$$r_{xy}^w(kT) = \sum_{n=-(N-1)}^{N-1} a_x^b(t_s^j + nT)a_y^b(t_s^j + nT - kT), \quad k = 0, \pm 1, \dots, \pm(N-1) \quad (3.2)$$

where

t_s^j = beginning epoch of the j^{th} data capture window, (integer multiple of T)

t_e^j = ending epoch of the j^{th} data capture window, (integer multiple of T)

N = number of samples in the data capture window
 $= (t_e^j - t_s^j)/T + 1$

w = center epoch of the window
 $= (t_e^j + t_s^j)/2$.

The purpose here is to capture the epoch of the highest correlation between any of the two accelerometer signals, given that one expects two or more accelerometers to be highly correlated at jump start and end. The

3.4. Sensor Correlation and Covariance

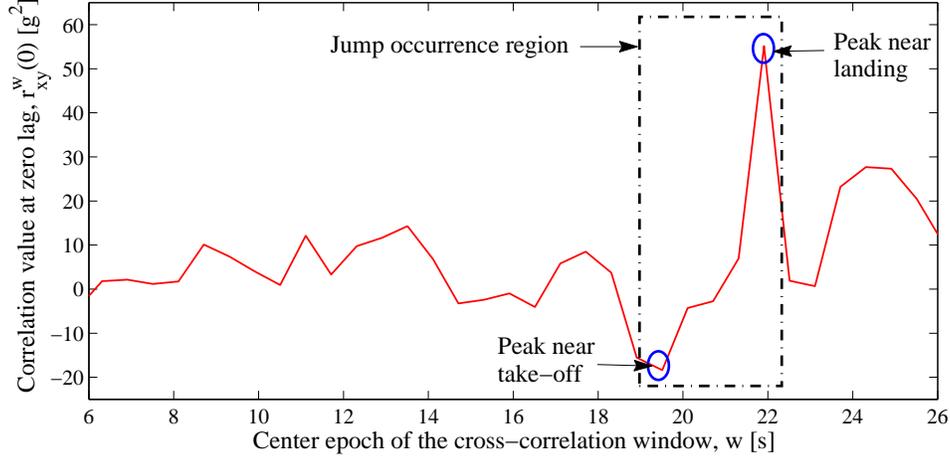


Figure 3.8: Cross-correlation at zero lag of windowed a_x^b and a_y^b .

region of simultaneous high correlation in all the acceleration signals is very brief in span. Therefore, the cross-correlation at zero lag, i.e. $r_{xy}^w(0)$, is the main focus of interest for each window. Figure 3.8 shows the crosscorrelation of a_x^b and a_y^b of the data shown in Figure 3.1. For convenience, the cross-correlation values are plotted against the center epoch of the pertinent window. The window size, i.e. $t_e^j - t_s^j$, is 0.6 s. The window size is empirically chosen to produce best results. The peak values at the jump start and end region are clearly noticeable in this cross-correlation plot. Hence it proves the hypothesis that the acceleration signals are highly correlated during the start and end of the aero phase due to the abrupt change in the dynamics involved.

In Figure 3.8, undesired fluctuations with considerable magnitude are present before and after the jump region. The nonzero average value of the sensor signal causes this high cross-correlation value. Moreover, as accelerometer values are very low during a jump, it is possible to miss a cross-correlation peak if one of the accelerometer values is essentially zero. To overcome these difficulties, covariance of the sensor signals may be used.

3.4. Sensor Correlation and Covariance

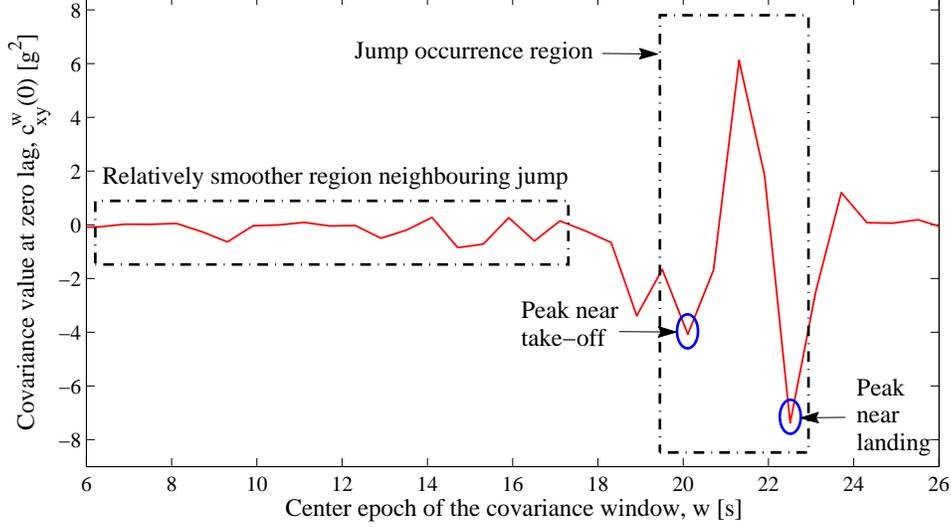


Figure 3.9: Covariance at zero lag of windowed a_x^b and a_y^b .

Covariance of the two discrete time signal a_x^b and a_y^b can be defined as

$$c_{xy}^w(kT) = \sum_{n=-(N-1)}^{N-1} \left\{ a_x^b(t_s^j + nT) - \frac{1}{N} \sum_{n=0}^{N-1} a_x^b(t_s^j + nT) \right\} \cdot \left\{ a_y^b(t_s^j + nT - kT) - \frac{1}{N} \sum_{n=0}^{N-1} a_y^b(t_s^j + nT) \right\}, \quad (3.3)$$

$$k = 0, \pm 1, \dots, \pm(N-1). \quad (3.4)$$

Figure 3.9 shows the covariance of a_x^b and a_y^b at zero lag, i.e. $c_{xy}^w(0)$ versus the center epoch of the window. Similarly, in this plot, the peaks near jump start and end are prominently distinguishable. Moreover, it is apparent that the region neighboring the jump is smoother than what is seen for the cross-correlation plot. Therefore, the covariance of the sensor signals would be a better tool for jump detection. The presence of a peak with high magnitude is also noticed after the peak near take-off. These scenarios may appear due to the two extremes (one minimum and one maximum) experienced by the accelerometers involved in jump take-off and landing.

The acceleration experienced by the body frame depends entirely on the orientation and direction of the forces involved in the jump. Consequently,

the covariance will differ significantly depending on the selection of two sensor signals for any particular jump. Moreover, using the covariance value, it is only possible to predict the jump start and end epochs with an accuracy of the covariance window length. Window length cannot be reduced, as it will diminish distinct features of covariance near the jump region. Exploiting the concept of covariance, a novel intuitive method is presented in the next section, which can subdue the above mentioned problems.

3.5 Windowed Mean Canceled Multiplication

From the previous discussion, it is understood that all three accelerometer signals should be involved in the jump detection algorithm to overcome the problem pertaining to the orientation uncertainty of the body frame. Another key factor is that the high peak value of the covariance is contributed to predominately by a very small number of epochs which fall within the brief time span of the event “jump start” or “jump end”. Based on these rationale, the novel Windowed Mean Canceled Multiplication (WMCM) method is proposed. WMCM of the three acceleration signals a_x^b , a_y^b and a_z^b captured in the j^{th} window spanning from t_s^j to t_e^j is defined as

$$\begin{aligned}
 m_{xyz}(t_s^j + nT) &= \left\{ a_x^b(t_s^j + nT) - \frac{1}{N} \sum_{p=0}^{N-1} a_x^b(t_s^j + pT) \right\} \cdot \\
 &\quad \left\{ a_y^b(t_s^j + nT) - \frac{1}{N} \sum_{p=0}^{N-1} a_y^b(t_s^j + pT) \right\} \cdot \\
 &\quad \left\{ a_z^b(t_s^j + nT) - \frac{1}{N} \sum_{p=0}^{N-1} a_z^b(t_s^j + pT) \right\}, \\
 n &= 0, 1, 2, \dots, N - 1.
 \end{aligned} \tag{3.5}$$

Since the multiplied values are not added together for a particular window in WMCM, the information on epochs with the most correlations is not lost. The involvement of all three axes accelerations also ensures that the change in dynamics is captured regardless of sensor orientation. Figure 3.10 shows the WMCM values of the acceleration data set shown in Figure 3.1. It is evident from the plot that near the take-off and landing regions the WMCM value exhibits a very large magnitude response. WMCM values other than in the jump region are very low in magnitude without any considerable fluctuation. Therefore, with a safe threshold for WMCM, jump

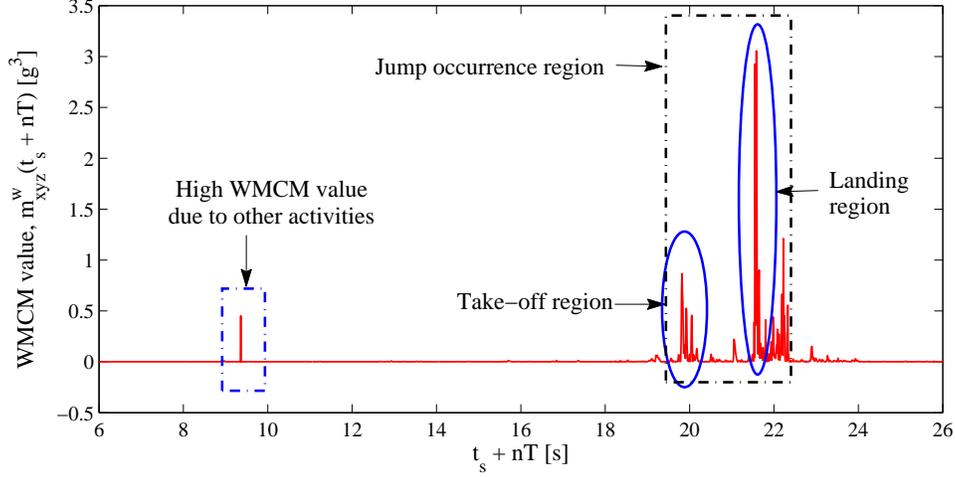


Figure 3.10: WCM of a_x^b , a_y^b and a_z^b .

occurrence can be detected with a good estimate of the jump start and end epochs. However, due to disturbance activities and abrupt head motion a high value in WCM can be observed as also shown in Figure 3.10. To verify the validity of any high value of WCM indicating jump occurrence, another complementary method is presented in the next section.

3.6 Preceding and Following Acceleration Difference

In a static state or in motion with little change in the dynamics, the resultant acceleration a^b of an athlete exhibits noisy characteristics with a steady mean. Therefore, the average values of the resultant acceleration over consecutive windows should not differ by much if there is no abrupt change in the resultant acceleration. However, at the jump occurrence region the average value of the resultant acceleration falls drastically. If a time epoch t_I is selected near the take-off or landing region for a^b , the neighboring accelerations preceding and following t_I are likely to differ significantly. This difference in the preceding and following accelerations should serve as an effective tool for jump detection.

Preceding and Following Acceleration Difference (PFAD) is computed considering a certain window around the point of interest. Figure 3.11 shows the PFAD windows of a^b for a point of interest at epoch t_I near the take-

3.6. Preceding and Following Acceleration Difference

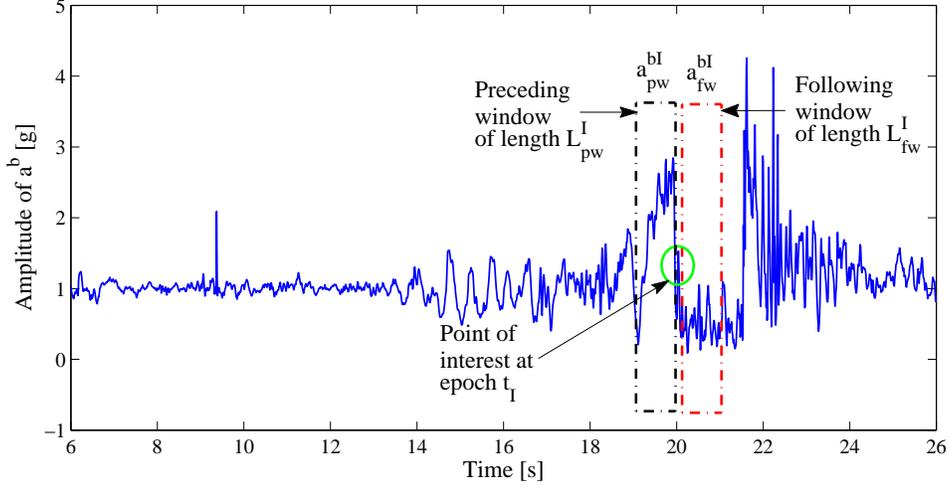


Figure 3.11: Windows for PFAD around a point of interest near take-off.

off region. L_{pw}^I and L_{fw}^I are the lengths of the windows preceding and following t_I respectively. The average resultant acceleration captured within the windows preceding and following t_I is termed \bar{a}_{pw}^{bI} and \bar{a}_{fw}^{bI} respectively. PFAD at t_I is the difference of these two averages. Therefore, the PFAD of a^b for any point of interest at epoch t_I , where t_I is an integer multiple of T , can be defined as

$$\begin{aligned} da^b(t_I) &= \frac{\sum_{n=1}^{NL_{pw}^I} a^b(t_I + nT)}{NL_{pw}^I} - \frac{\sum_{n=1}^{NL_{fw}^I} a^b(t_I - nT)}{NL_{fw}^I} \\ &= \bar{a}_{pw}^{bI} - \bar{a}_{fw}^{bI} \end{aligned} \quad (3.6)$$

where $NL_{pw}^I = L_{pw}^I/T$ and $NL_{fw}^I = L_{fw}^I/T$ are the number of samples within the windows of length L_{pw}^I and L_{fw}^I respectively.

It is clear from the captured acceleration magnitude that the average resultant acceleration within the preceding window \bar{a}_{pw}^{bI} , will be significantly greater than the average resultant acceleration within the following window \bar{a}_{fw}^{bI} . The opposite scenario will be observed if t_I is near the landing region. Therefore, $da^b(t_I)$ should be a maximum if t_I is near the take-off and a minimum if t_I is near the landing. At any other position except these two, the PFAD value $da^b(t_I)$ is not expected to exhibit any considerable magnitude. Figure 3.12 shows the PFAD values of the resultant acceleration

3.7. Jump Detection

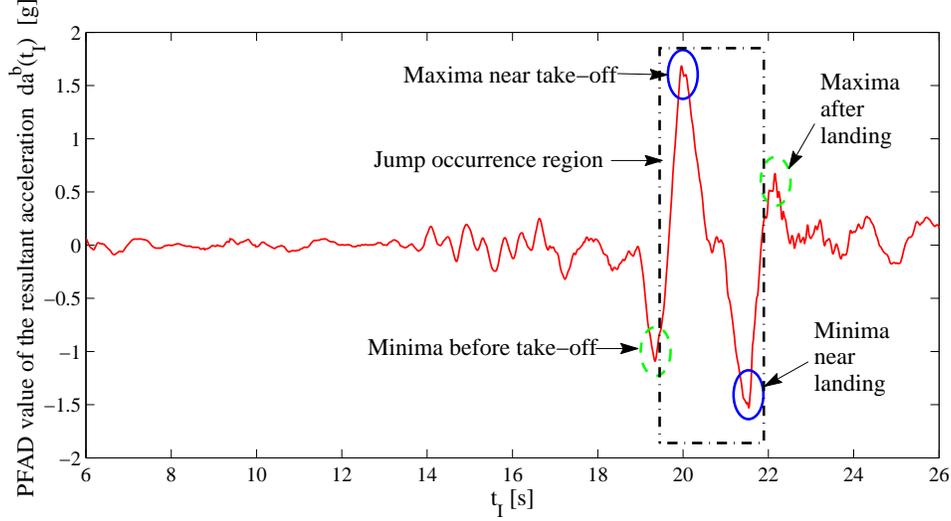


Figure 3.12: PFAD value of the resultant acceleration a^b .

shown in Figure 3.2. For this example, the window lengths L_{pw}^I and L_{fw}^I are set to 0.6 s. Selection of this window length depends on the jump category, which is explained in Subsection 3.7.3. A maximum and a minimum with large magnitude are evident near the jump start and the jump end respectively. Another minimum just before the take-off and another maximum just after the landing are also evident. These peak PFAD values appear due to the gain in acceleration (mostly vertically upward and forward) just before take-off and the gain in acceleration (mostly vertically upward and backward) just after landing. Apart from the jump occurrence region, the PFAD value $da^b(t_I)$ is generally small and varies little. Therefore, PFAD provides a powerful tool for detecting any jump occurrence. To be more specific, from the PFAD value it is possible to determine whether t_I is near the jump start or the jump end. If $da^b(t_I)$ is positive with a significant magnitude, then t_I is near the jump start or the take-off. On the other hand, if t_I is negative with a significant magnitude, then t_I can be determined to be near the jump end or the landing.

3.7 Jump Detection

Using the novel concepts of WCM and PFAD a jump detection algorithm is proposed in this section. The algorithm consists of two main

steps:

- **Step 1.** Primarily detect the jump occurrence by WMCM.
- **Step 2.** Check the validity of the jump detection with PFAD.

3.7.1 Detection through WMCM

Data from the accelerometers in three orthogonal axes of the body frame are captured in a window of $(t_e^j - t_s^j) = NT = 0.6$ s. For each window the WMCM value $m_{xyz}(t_s^j + nT)$ is calculated. Then for each window, the maximum value of $m_{xyz}(t_s^j + nT)$ is compared with a predefined threshold th_m . If, for a particular window, the maximum value of $m_{xyz}(t_s^j + nT)$ is greater than or equal to th_m , then it is considered that a Jump Indication (JI) has been detected at the corresponding time epoch $t_{JI} = t_s^j + n_{max}^j T$. Here, n_{max}^j is the index of the maximum WMCM value for the j^{th} window. To ensure that the maximum WMCM value in a potential locality is considered as JI, the maximum value of $m_{xyz}(t_s^{j+1} + nT)$, i.e. maximum WMCM value of the following window, is compared with the $m_{xyz}(t_{JI})$. If the maximum value of $m_{xyz}(t_s^{j+1} + nT)$ is greater than $m_{xyz}(t_{JI})$, the epoch of JI is updated to $t_{JI} = t_s^{j+1} + n_{max}^{j+1} T$. Otherwise the previous value for t_{JI} is preserved. Thus, indication of jump occurrence, i.e. JI, is detected primarily through WMCM and t_{JI} is passed to the next step for verification.

WMCM Threshold

It is important to note that detection of either jump start or jump end is adequate for successful detection of jump occurrence or JI. Depending on where the sensors are mounted, the relative magnitude of the WMCM value near the jump start and the jump end may vary. It has been empirically found that, for the Recon goggles where the sensors are mounted on the head, the WMCM value near the jump end is consistently greater than or equal to what is observed near the jump start. For example, the magnitude of the WMCM value near landing in Figure 3.10 is much higher than the magnitude observed near take-off. Therefore, to be more resistive to false JI detection and save power consumption, the threshold th_m is set to a higher value that only secures the detection of JI near the jump end or landing. Hence, any JI near the jump start is not guaranteed to be detected by this threshold. However, if any JI is detected near the jump start, the corresponding JI at the jump end is overlooked. The sign of the PFAD value at t_{JI} , $da_b(t_{JI})$, is used to determine whether the JI is near the jump start

3.7. Jump Detection

or end. In this research, the threshold th_m is set to a heuristically optimum value of 0.20 g^3 .

For the data captured with the Ripxx device [4], which is strapped on to the leg of an athlete, the scenario is opposite. That is, the WMCM value near the jump start is consistently greater than or equal to the WMCM value near the jump end. This is because the different body parts experience different changes in dynamics for maneuvers. Consequently, the acceleration sensors capture different accelerations depending on where the sensors are mounted. Therefore, the threshold for the WMCM value should be set considering the change in dynamics of the sensor attached to the specific body part.

3.7.2 Validity Check through PFAD

If any JI is detected by the WMCM procedure, the resultant acceleration a^b is calculated from the three axes acceleration data captured. Then the PFAD value of a^b at t_{JI} , $da^b t_{JI}$ is calculated and compared with a predefined threshold th_d . The selection procedure of th_d is demonstrated later in Section 3.7.3. If the magnitude of the PFAD value, $|da^b t_{JI}|$, is greater than th_d the detected JI is considered as a valid jump detection. Otherwise, the detected JI is discarded as a false jump indication. It is also possible to determine whether the JI is near take-off or landing from the sign of $da^b t_{JI}$ as mentioned earlier. This knowledge is important for the AT determination algorithm proposed later in Section 3.8.

Figure 3.13 summarizes the concept of the proposed jump detection algorithm. The plotted WMCM value m_{xyz} is shifted by -2 g^3 for visual convenience. The high WMCM value will primarily report the JIs at the jump start and jump end. Both the JIs will be validated by the high value of PFAD at corresponding t_{JI} . It should be noted that another JI will be reported in the first step due to the high WMCM value near 9.5 s. This detected JI will be discarded in the second step, as the PFAD value at the corresponding t_{JI} is negligible in magnitude. Thus, the proposed jump detection method will only detect true jumps by filtering the false indication in a two step method.

In Figure 3.13, L_{pw}^I and L_{fw}^I are selected to be 0.6 s. However, to efficiently address a wide range of jumps with different characteristics, the span of the PFAD windows can be varied. Depending on the acceleration variation characteristics at the neighboring epochs of the detected t_{JI} , JI can be divided into two categories, namely high disturbance JI and low disturbance JI. Both these categories can also be subdivided into two more categories, high impact JI and low impact JI. Depending on the JI category,

3.7. Jump Detection

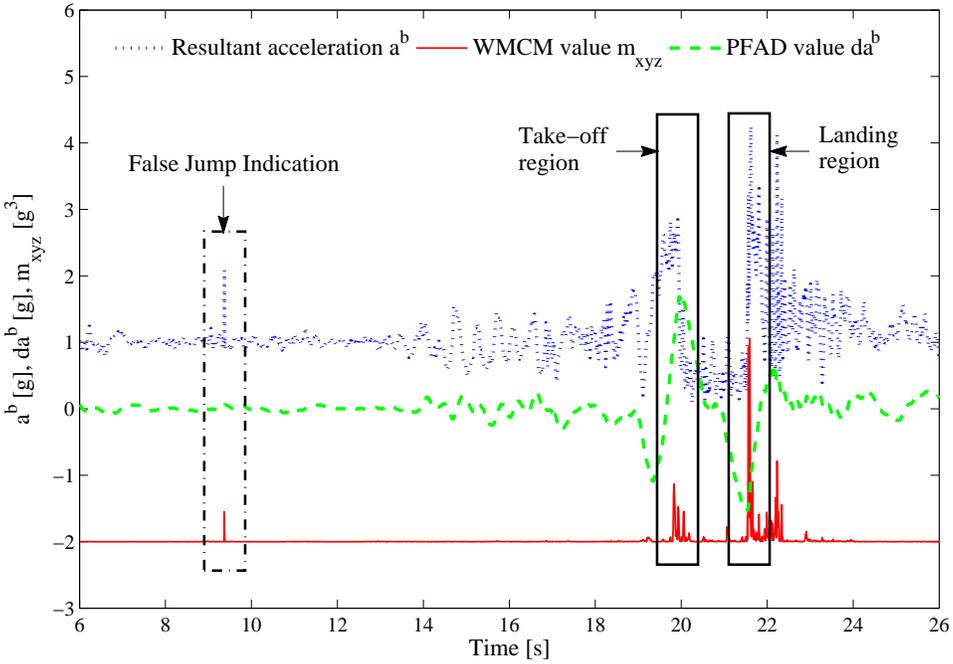


Figure 3.13: Proposed jump detection method. Note: m_{xyz} is shifted by -2 g^3 for visual convenience.

the threshold value, th_d , for the JI validation process is selected.

3.7.3 Jump Indication Categories

JIs are categorized into high disturbance, low disturbance, high impact and low impact. The high/low disturbance categories set the window length for the PFAD calculation and the high/low impact categories determine the threshold for the PFAD value.

High Disturbance JI. Generally for big jumps, where greater force is involved in take-off and stronger thrust is felt at landing, abrupt head and body motion cause high fluctuations in resultant acceleration near the jump start and end regions. In such cases, the detected JI is termed as high disturbance JI. Figure 3.14 shows an example of high disturbance JI. The trough in resultant acceleration following t_{JI} is due to the abrupt head and body motions associated with hard landings. Jumps associated with this type of JI generally have lengthy aero phase. It is understandable from the figure that the PFAD value at t_{JI} with narrow windows will not be significant in magnitude due to the trough. On the other hand, wide windows for PFAD calculations will lessen the effect of the trough and will produce a PFAD value with greater magnitude at t_{JI} . Therefore, relatively higher values for L_{pw}^{JI} and L_{fw}^{JI} should be selected for high disturbance JIs. For this research, L_{pw}^{JI} and L_{fw}^{JI} are heuristically selected to be 1 s for high disturbance JIs.

Low Disturbance JI. In many cases, athletes try to minimize the reaction force experienced by the body by conducting smooth take-offs and landings. In such cases, neighboring resultant accelerations of t_{JI} show less fluctuations. Therefore, JI detected for this type of take-off or landing is termed as low disturbance JI. An example of low disturbance JI is shown in Figure 3.15. From the figure, it can be observed that the narrow window for averaging is adequate to extract the high PFAD value at t_{JI} . It is important to note that the shallow trough is not deep enough to cause the PFAD value to be insignificant. In fact, wide windows may render the PFAD value to become less significant in magnitude. This is because low disturbance JIs are generally detected for short aero phase jumps. Hence, the span of wide windows around t_{JI} may exceed the jump occurrence region, causing higher resultant acceleration (as shown in dotted circle in Figure 3.15) to be included in the averaging process. Therefore, a relatively lower value for

3.7. Jump Detection

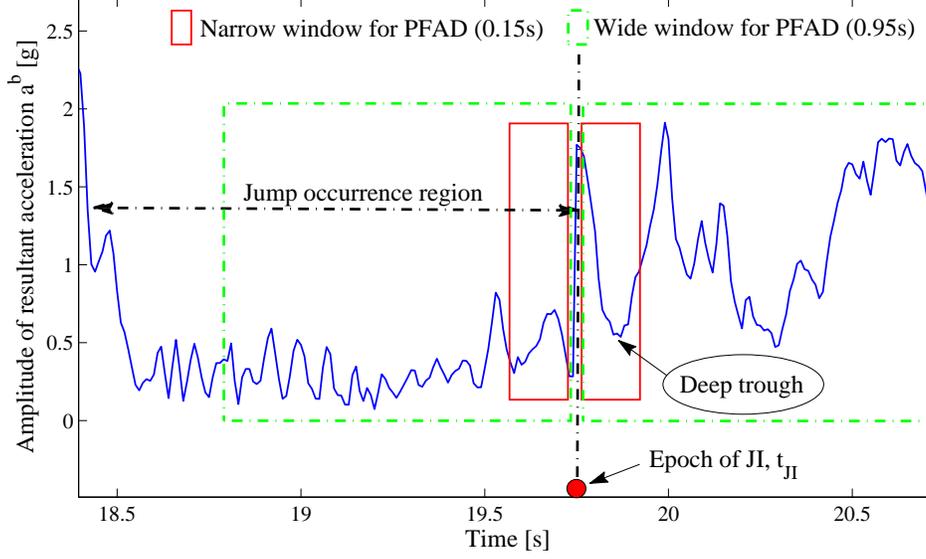


Figure 3.14: High disturbance JI.

L_{pw}^{JI} and L_{fw}^{JI} is chosen for low disturbance JIs, which is 0.30 s in this study.

Depending on the WMCM value at t_{JI} the above mentioned JI categories can also be subdivided into two subcategories:

High Impact JI. If the abrupt change in acceleration during take-off or landing occurs swiftly within a very brief time period, then the corresponding WMCM value tends to be higher in magnitude. This is because the mean cancellation process in WMCM causes the peak accelerations in the sensors to stand out from their neighbors. Hence the resultant multiplication at t_{JI} , $m_{xyz}(t_{JI})$, exhibits high magnitude. Therefore, JIs with high WMCM magnitude are defined as high impact JIs. In this study, any JI with $m_{xyz}(t_{JI}) \geq 0.50g^3$ is considered a high impact JI. Figure 3.16 shows a high impact JI along with the resultant acceleration and corresponding WMCM value. Understandably, the JI is detected at the highest peak of WMCM at 40.5 s. However, it can also be observed, as previously discussed, that the PFAD value for narrow windows might not achieve a high magnitude due to the presence of a high disturbance near t_{JI} . Similar scenarios are observed for general high impact JIs. Wide windows for PFAD will resolve

3.7. Jump Detection

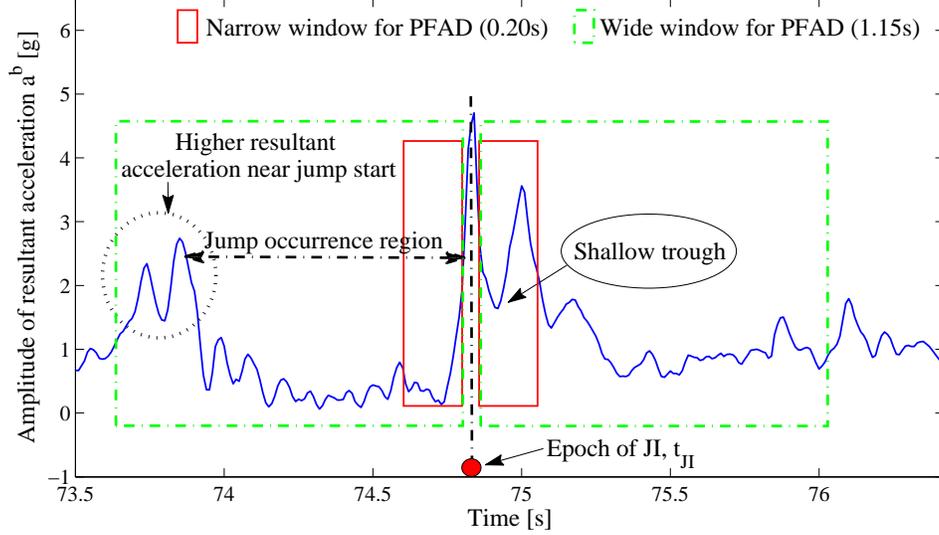


Figure 3.15: Low disturbance JI.

this issue for lengthy aero phase jumps such as in Figure 3.16. Also, as previously noted, wide windows will not work for jumps with short duration. Therefore, due to the unsteady nature of the resultant acceleration near JI, the threshold th_d is set at a relatively low value for validity check through PFAD. Fortunately, this is also logically suited to the fact that any JI with higher WMCM magnitude is a more eligible candidate than one with a lower magnitude. Consequently, it is justified to have a less stringent threshold for the validation procedure of the high impact JI primarily detected by the WMCM procedure.

Low Impact JI. Take-offs and landings are often observed where the change in resultant acceleration is steady and spans a prolonged period of time. Therefore, the acceleration at t_{JI} cannot excel itself significantly from its neighbors. This phenomenon results in a relatively low value of WMCM at t_{JI} . In this case, the JI with a relatively low WMCM value is termed as low impact JI. In this research, JIs with $0.20g^3 \leq m_{xyz}(t_{JI}) < 0.50g^3$ are considered low impact JIs. As mentioned earlier, JIs with a WMCM value less than $0.20g^3$ are not considered as a valid JI. Figure 3.17 shows an example of a low impact JI. It can be discerned from the figure that the PFAD value is likely to have a high magnitude due to the steady nature of

3.7. Jump Detection

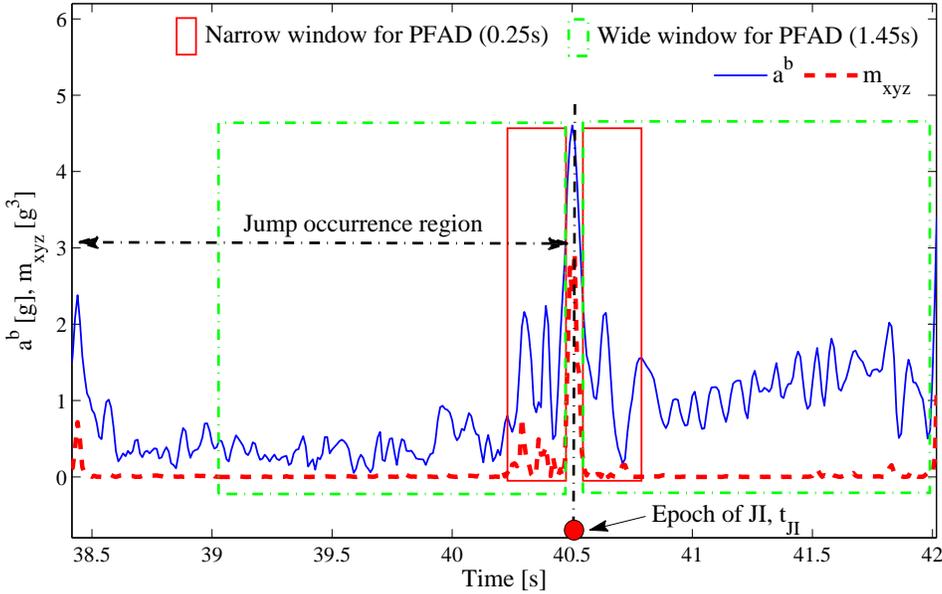


Figure 3.16: High impact JI.

3.7. Jump Detection

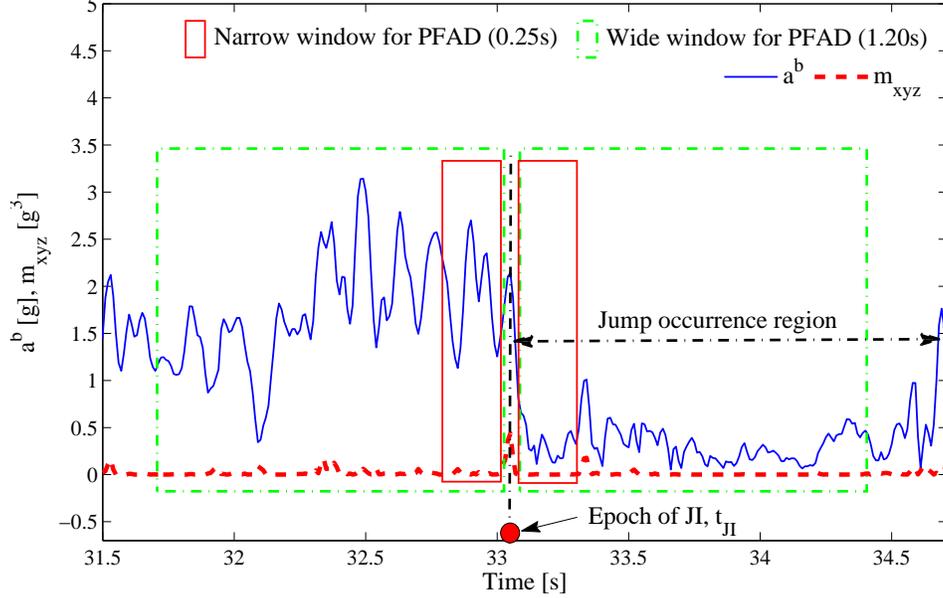


Figure 3.17: Low impact JI.

the resultant acceleration on opposite sides of t_{JI} for both wide and narrow windows. Therefore, for low impact JI the th_d can be set to a higher value relative to the case of high impact JI. Fortunately, this also makes logical sense, as a JI with low WMCM value is more prone to be a false indication of jump than a JI with high WMCM magnitude. Hence, a rigorous false detection procedure with high threshold is rational for a low impact JI.

PFAD Threshold Selection

The selection of the PFAD threshold th_d is done through a statistical analysis, namely a Two-Sample t -Test [47], for both high and low impact jumps. Acceleration data was collected with the Recon goggles for 27 ski jumps. Then JIs were detected through WMCM and corresponding \bar{a}_{pw}^{bJI} and \bar{a}_{fw}^{bJI} were calculated. For the purpose of this study, for jumps with aero phase duration well over 1 s, L_{pw}^{JI} and L_{fw}^{JI} are set at 1 s. For the others, the window lengths are set at 0.30 s. Depending on the WMCM value, the jumps are then divided into sets of high impact and low impact jumps. Since interest is focused on the difference in magnitude of \bar{a}_{pw}^{bJI} and \bar{a}_{fw}^{bJI} , the two

3.7. Jump Detection

sample t -Test is conducted with \bar{a}_{low}^{bJI} as one sample set and \bar{a}_{high}^{bJI} as another sample set. For any JI, \bar{a}_{low}^{bJI} and \bar{a}_{high}^{bJI} are the minimum and maximum value between \bar{a}_{pw}^{bJI} and \bar{a}_{fw}^{bJI} respectively.

In the two sample t -Test the test statistic is [47]

$$t_0 = \frac{\bar{y}_1 - \bar{y}_2}{S_p \sqrt{\frac{1}{n_1} + \frac{1}{n_2}}} \quad (3.7)$$

where \bar{y}_1 and \bar{y}_2 are the means of the samples of \bar{a}_{high}^{bJI} and \bar{a}_{low}^{bJI} , and n_1 and n_2 are the sample sizes. It is assumed that \bar{a}_{high}^{bJI} and \bar{a}_{low}^{bJI} have equal variances. S_p^2 is the estimate of the common variance computed from

$$S_p^2 = \frac{(n_1 - 1)S_1^2 + (n_2 - 1)S_2^2}{n_1 + n_2 - 2} \quad (3.8)$$

and S_1^2 and S_2^2 are two individual sample variances, i.e. variances of \bar{a}_{high}^{bJI} and \bar{a}_{low}^{bJI} respectively. The hypotheses being tested by the t -Test are

- **Null hypothesis H_0 :** $\mu_1 = \mu_2$
- **Alternative hypothesis H_1 :** $\mu_1 > \mu_2$

where μ_1 and μ_2 are the means of \bar{a}_{high}^{bJI} and \bar{a}_{low}^{bJI} . The observation models are: $\bar{a}_{high}^{bJI} = \mu_1 + error$ and $\bar{a}_{low}^{bJI} = \mu_2 + error$.

To determine whether to reject H_0 : $\mu_1 = \mu_2$, t_0 is compared with the t distribution with $n_1 + n_2 - 2$ degrees of freedom. If $t_0 \geq t_{\zeta, n_1+n_2-2}$, where t_{ζ, n_1+n_2-2} is the upper ζ percent point of the t distribution with $n_1 + n_2 - 2$ degrees of freedom, H_0 will be rejected. Here, ζ is the significance level, which represents the possibility of rejecting H_0 while it is in fact true. For this test, $\zeta = 0.01$ or 1%.

Table 3.1 lists the two samples collected for the low impact JIs. For a 1% significance level the calculated test statistic for low impact jumps is $t_0 = 23.7975 > t_{0.01, 17+17-2} = 2.4487$. As expected, the null hypothesis is rejected and it is confirmed that $\mu_1 > \mu_2$. The extremely low P -value of 3.2184×10^{-22} , where P -value is the smallest level of significance that would lead to the rejection of the null hypothesis [47], proves the strong rejection of the null hypothesis. However, the major goal of conducting this t -Test is to know the range within which $(\mu_1 - \mu_2)$ is expected to lie for the low impact JIs. Therefore, interest is focused on the confidence interval of the test, which gives the range of the difference in means of the parameters (\bar{a}_{high}^{bJI} ,

3.7. Jump Detection

Table 3.1: Two sample t-Test for low impact JIs

Jump No.	\bar{a}_{low}^{bJI}	\bar{a}_{high}^{bJI}
1	0.2939	1.8681
2	0.4915	1.8792
3	0.5255	1.5765
4	0.4384	1.8636
5	0.3136	1.6200
6	0.4369	1.7972
7	0.3926	1.4994
8	0.6478	1.2860
9	0.3205	1.6463
10	0.5037	1.8141
11	0.5072	1.6335
12	0.3574	1.6188
13	0.2552	1.5931
14	0.1568	1.6714
15	0.3652	1.6318
16	0.4095	1.6547
17	0.6945	2.0160

\bar{a}_{low}^{bJI}). The interval $1.1377 \leq \mu_1 - \mu_2 < \infty$ is the calculated $100(1 - \zeta) = 99\%$ confidence interval of the low impact JIs. In this regard, if a large number of $(\bar{a}_{high}^{bJI} - \bar{a}_{low}^{bJI})$, i.e. $|da^b(t_{JI})|$, is gathered, 99% of them will be ≥ 1.1377 .

The calculated confidence interval provides a reasonable estimate of the value for the threshold th_d for the low impact JI verification procedure. Since a value of $|da^b(t_{JI})| = 1.1377$ is copiously large for an acceleration difference, a safer lower value can be selected for th_d so that possibility of declaring any valid JI as invalid is minimized. Therefore, to validate numerous possible types of JIs, th_d is selected to be 0.80 g for low impact jumps.

A similar test is done for the samples of high impact JIs as shown in Table 3.2. Here, $t_0 = 6.2140 > t_{.01,10+10-2} = 2.5524$ and P -value is 3.6460×10^{-6} . As previously discussed, the null hypothesis is also rejected here. The 99% confidence interval is $0.4531 \leq \mu_1 - \mu_2 < \infty$. Again, the lower boundary of the confidence interval gives an estimate of the PFAD threshold. Due to reasoning similar to that in the case of the low impact JIs, th_d for high impact JIs is set to a relatively lower and safer value of 0.25 g for this project.

3.7. Jump Detection

Table 3.2: Two sample t-Test for high impact JIs

Jump No.	\bar{a}_{low}^{bJI}	\bar{a}_{high}^{bJI}
18	0.7547	2.2298
18	0.6225	1.2437
20	0.8521	1.3228
21	0.7997	1.2297
22	0.8587	1.3239
23	0.7796	1.6223
24	0.5607	1.6255
25	0.6478	1.6595
26	0.8489	1.0170
27	0.9048	1.9652

False JI Detection

After any JI is detected by the WMCM procedure, the validity checking steps through PFAD are as follows:

1. From $m_{xyz}(t_{JI})$ it is determined whether the JI is high impact or low impact and the corresponding value for th_d is set.
2. $da^b(t_{JI})$ is calculated considering L_{pw}^{JI} and L_{fw}^{JI} to be 1 s, i.e. assuming high disturbance JI, because high disturbance JI requires long windows.
3. If $da^b(t_{JI}) \geq th_d$ the JI is considered as valid and the AT calculation algorithm is invoked. Otherwise the assumption that the JI is high disturbance is discarded and one proceeds to the next validation step.
4. After the assumption that JI is categorized as high disturbance is discarded, L_{pw}^{JI} and L_{fw}^{JI} are set to 0.30 s and $da^b(t_{JI})$ is calculated again assuming JI to be low disturbance.
5. If $da^b(t_{JI}) \geq th_d$, the JI is considered as valid low disturbance JI and the AT calculation algorithm is used. Otherwise, the JI is completely discarded and considered to be a false detection.
6. After any false detection, the algorithm returns to the data acquisition mode and the WMCM value is calculated for the new data set.

The assumption that the detected JI is high disturbance is made earlier than the low disturbance assumption to reduce computation. Most of the regular jumps will comply with the high disturbance criteria and skip the low disturbance JI verification computation. On the other hand, jumps with small AT and disturbance activities will only be discarded in the first assumption and the algorithm for low disturbance JI verification will be invoked.

The flow-chart in Figure 3.18 summarizes the entire jump detection procedure.

3.8 Air Time Determination

3.8.1 Characteristic Acceleration Points

To determine the Air Time (AT) accurately, it is imperative to detect the precise epochs of the events ‘jump start’ or ‘take-off’ and ‘jump end’ or ‘landing’. ‘Take-off’ is defined as the event of momentary transition from ground to air. Similarly, ‘landing’ is the event of momentary transition from air to ground. In noiseless and undisturbed ideal conditions, the resultant acceleration pertaining to any jump should resemble a U-shaped pattern as shown in Figure 3.19. The characteristic points that define the U-shaped acceleration region are the highlighted points, namely jump start positive peak (JS_P), jump start negative peak (JS_N), jump end positive peak (JE_P), and jump end negative peak (JE_N). It should be noted that the local maxima of any cusp is termed as positive peak and the local minima of any notch is termed as negative peak.

As shown in Figure 3.19, JS_P is the latest point before aero phase when the reported acceleration indicates that the athlete is completely on the ground. JS_N is the first point in aero phase when the reported acceleration suggests that the athlete is completely in the air. Therefore, the event ‘take-off’ must have occurred in the time span between these two points. Similarly, ‘landing’ must have occurred between JE_N and JE_P . Depending on the specific sport maneuvers, the weighted average of the epochs of JS_P and JS_N , and JE_P and JE_N should theoretically produce the exact epochs of take-off and landing. Consequently, the difference between these epochs should produce exact AT. Since the time spans for both take-off and landing are very brief in time (~ 0.05 s), normal averages of the start and end points should also yield AT with a reasonably high accuracy.

The argument presented above describes why accurate AT determination is directly dependent on the precise detection of the four characteristic

3.8. Air Time Determination

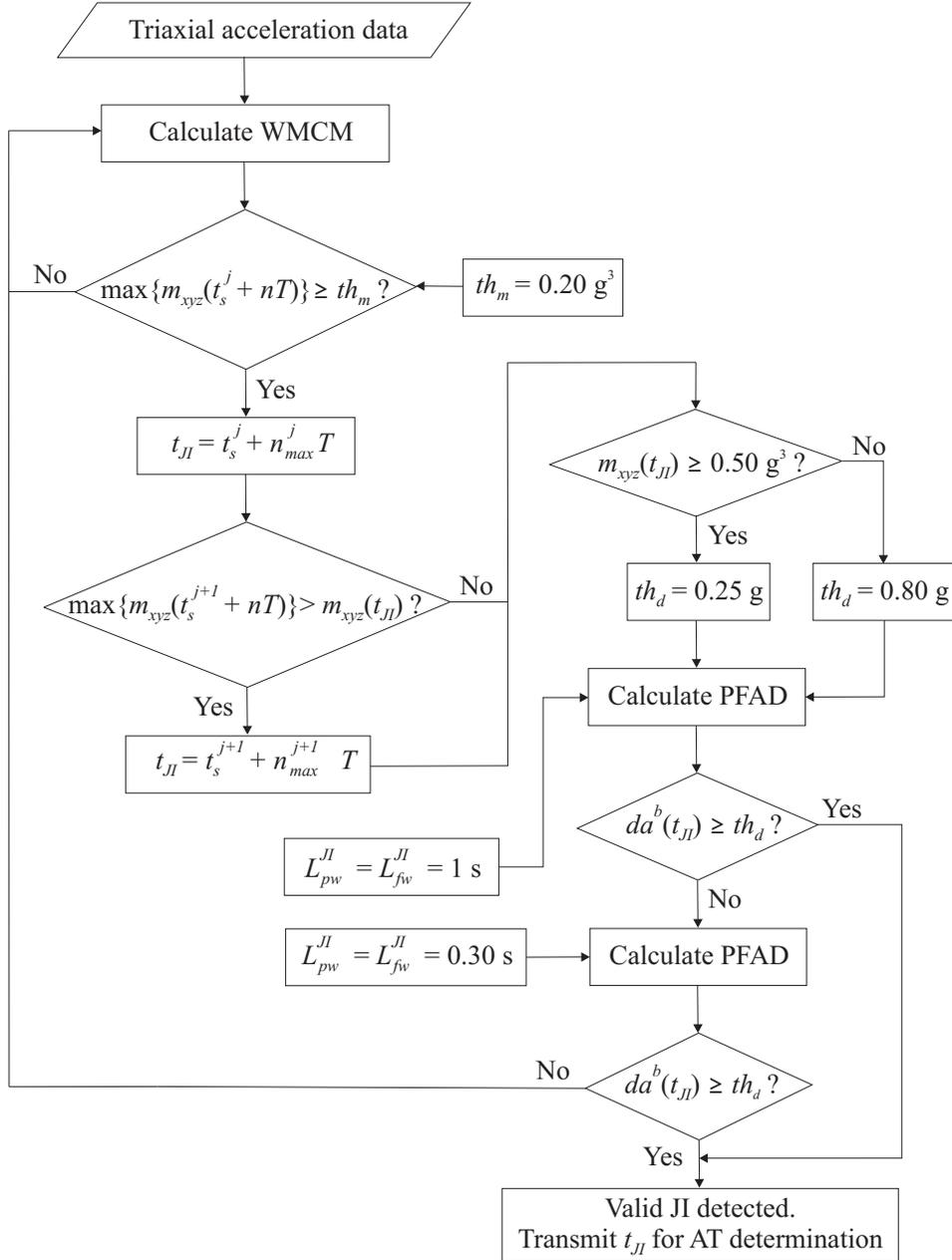


Figure 3.18: Proposed jump detection method.

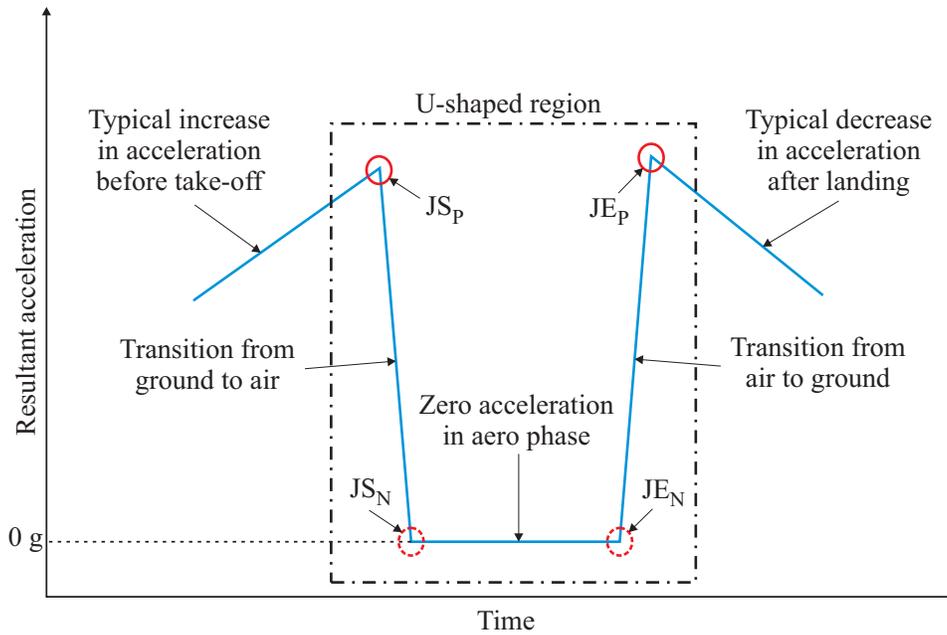


Figure 3.19: U-shaped region and characteristic points for an ideal jump.

points. Hence, the focus of the AT determination algorithm is to detect the four characteristic points as accurately as possible. Of course, for practical noisy jump acceleration signals the characteristic points are not as well defined as in the ideal case. The U-shaped region and the possible localities of JS_P , JS_N , JE_P and JE_N are shown in Figure 3.20 for a typical noisy data set. Due to noise and other disturbances, the transition periods as well as the characteristic points for the jump are not precisely obvious. Hence, in the proposed AT determination algorithm, positive or negative peaks possessing the most eligibility for the U-shape defining points are detected as the characteristic points.

3.8.2 Primary and Secondary Detection

Depending on the position of a valid JI, the detection techniques for the characteristic points differ. In this research, the detection of the two characteristic points near JI is termed Primary Detection and detection of the other two is termed Secondary Detection. The relevant peaks are consequently termed primary and secondary, positive/negative peaks. For example, if the JI is near take-off, then the detection of JS_P and JS_N is

3.8. Air Time Determination

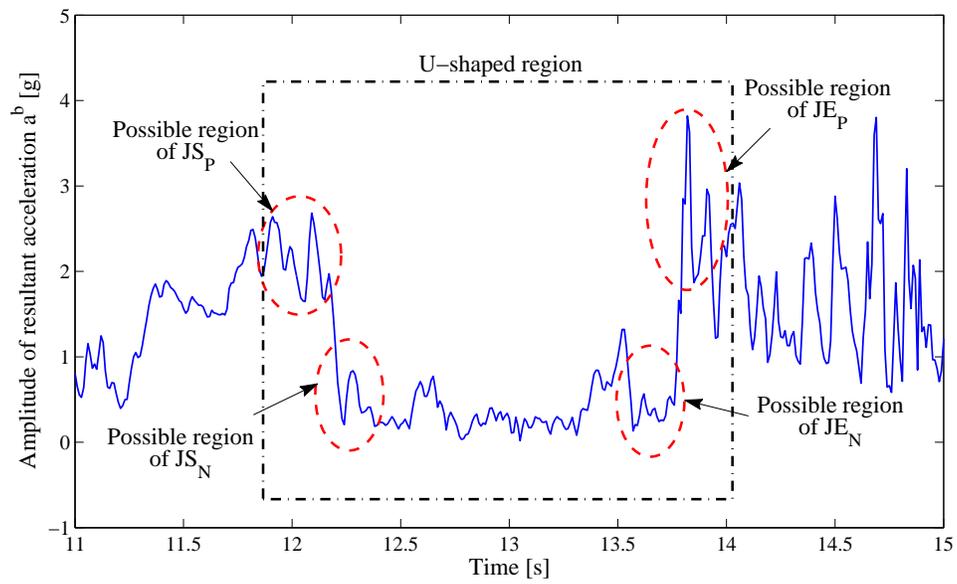


Figure 3.20: U-shaped region and characteristic points for a practical jump.

primary detection and the detection of JE_P and JE_N is secondary detection. In this case, primary positive and negative peaks are JS_P and JS_N respectively, and secondary positive and negative peaks are JE_P and JE_N respectively. Since the challenges in primary and secondary detection are dissimilar, distinct methods are proposed for each.

For the sake of convenience, methods for primary and secondary detection are demonstrated for a particular example. As the features of the jump start characteristic points are the reciprocal of those of the jump end characteristic points, the demonstration will not lose any generality.

3.8.3 Window Selection

After the detection and validation of any JI with WMCM and PFAD, a characteristic window is selected within which the AT determination algorithm is applied to detect all four characteristic points. This window spans from t_{cs} to t_{ce} where

$$t_{cs} = t_{JI} - \nabla t_{cs}^l, \quad l = 1, 2, \dots \quad (3.9)$$

$$t_{ce} = t_{JI} + \nabla t_{ce}^l, \quad l = 1, 2, \dots \quad (3.10)$$

Primarily, for any JI near the jump start, $\nabla t_{cs}^1 = 0.50$ s and $\nabla t_{ce}^1 = 3$ s. For a JI near the jump end these parameter values are reciprocated, i.e. $\nabla t_{cs}^1 = 3$ s and $\nabla t_{ce}^1 = 0.50$ s.

Since the jump aero phase can vary in length, the primarily selected window size may not prove to be adequate. On the other hand, selecting a very large window each time to accommodate the largest possible jump will excessively increase computation time and power. Therefore, window length is extended only if any of the characteristic points are not detected in the current window span. Each time, the window expansion parameter is increased following $\nabla t_{ce}^{l+1} = \nabla t_{ce}^l + 3$ s for JI near jump start and $\nabla t_{cs}^{l+1} = \nabla t_{cs}^l + 3$ s for JI near jump end. The window is expanded until $t_{ce}^l - t_{cs}^l > 9$ s. If any of the characteristic points are not detected within the maximum span of the window (9 s), the JI is discarded as invalid.

3.8.4 Positive and Negative Peaks

After selecting the characteristic window, the epochs of the positive and negative peaks within the window are determined. The positive peak epochs

3.8. Air Time Determination

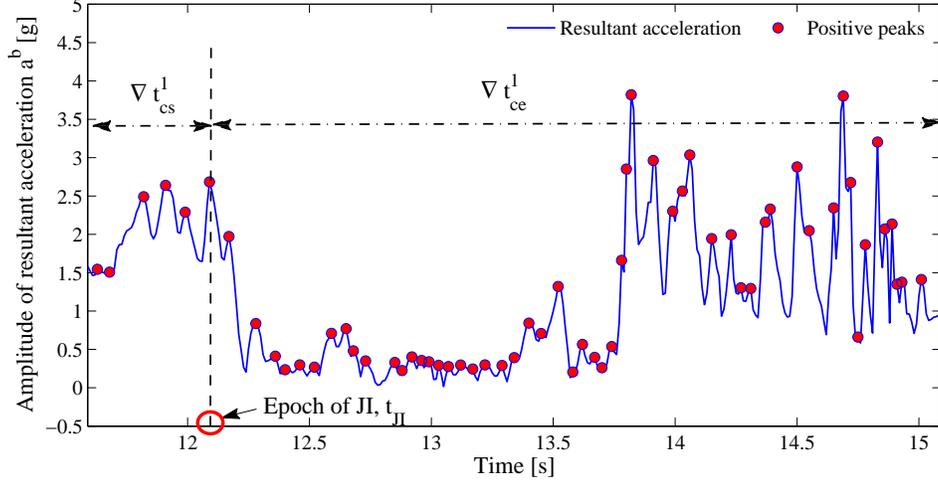


Figure 3.21: Window for characteristic acceleration points detection and potential positive peaks.

can be mathematically represented as

$$t_c^P = \arg_{t_{cs}+nT} \left\{ (a^b(t_{cs} + nT) - a^b(t_{cs} + nT - 1)) > 0 \right\} \cap \quad (3.11)$$

$$\arg_{t_{cs}+nT} \left\{ (a^b(t_{cs} + nT) - a^b(t_{cs} + nT + 1)) \geq 0 \right\} \quad (3.12)$$

where $\arg_{t_{cs}+nT} \{ \cdot \}$ is the set of epochs, $t_{cs} + nT$, for which the argument $\{ \cdot \}$ is valid. The operator for intersection of the set of epochs satisfying pertinent arguments is \cap . Similarly the epochs for negative peaks are

$$t_c^N = \arg_{t_{cs}+nT} \left\{ (a^b(t_{cs} + nT) - a^b(t_{cs} + nT - 1)) < 0 \right\} \cap \quad (3.13)$$

$$\arg_{t_{cs}+nT} \left\{ (a^b(t_{cs} + nT) - a^b(t_{cs} + nT + 1)) \leq 0 \right\}. \quad (3.14)$$

For the jump shown in Figure 3.20 the characteristic window and corresponding positive peaks are shown in Figure 3.21. The JI is detected at $t_{JI} = 12.09$ s and is validated afterwards by the jump detection algorithm. This example will be used for illustrating the primary and secondary detection procedures.

3.8.5 Primary Detection

From the PFAD value of the jump detection algorithm it is already known that the JI is near the jump start. Therefore, primary detection means the detection of JS_P and JS_N for this JI and subsequent detection of the pertinent epochs t_{JS}^P and t_{JS}^N . For the Recon goggles, primary detection is done by the straight-forward Closest Peak Method.

Closest Peak Method

In this method, the JI is exploited as a reference point. Within the characteristic window, the positive peak in the resultant acceleration nearest JI with a magnitude greater than or equal to 1 g is considered as JS_P . The nearest negative peak with a resultant acceleration less than 1 g is detected as JS_N . The time epochs for JS_P and JS_N can be expressed as

$$t_{JS}^P = \min_{t_I} \left\{ |t_{JI} - \arg\{a^b(t_I) \geq 1g\}| \right\}, \quad t_I \in t_c^P \quad (3.15)$$

$$t_{JS}^N = \min_{t_I} \left\{ |t_{JI} - \arg\{a^b(t_I) < 1g\}| \right\}, \quad t_I \in t_c^N \quad (3.16)$$

where $\min_{t_I}\{\cdot\}$ yields the epoch t_I which yields the minimum value of the argument $\{\cdot\}$.

Figure 3.22 shows the detected JS_P and JS_N . The detected peaks are at $t_{JS}^P = 12.09$ s and $t_{JS}^N = 12.24$ s. It can be observed from the figure that both the detected peaks reasonably estimate the two characteristic points on one side of the U-shaped region. The threshold of 1 g is selected theoretically since the ideal minimum resultant acceleration of the athlete is 1 g on the ground.

3.8.6 Secondary Detection

Each potential secondary peak possesses four characteristic attributes or criteria on which its eligibility can be assessed. The attributes for a peak at t_I are

1. Proximity to JI, $t_I - t_{JI}$
2. PFAD value, $da^b(t_I)$
3. Average magnitude of the preceding or following window, \bar{a}_{pw}^{bI} or \bar{a}_{fw}^{bI}
4. Self amplitude (resultant acceleration), $a^b(t_I)$.

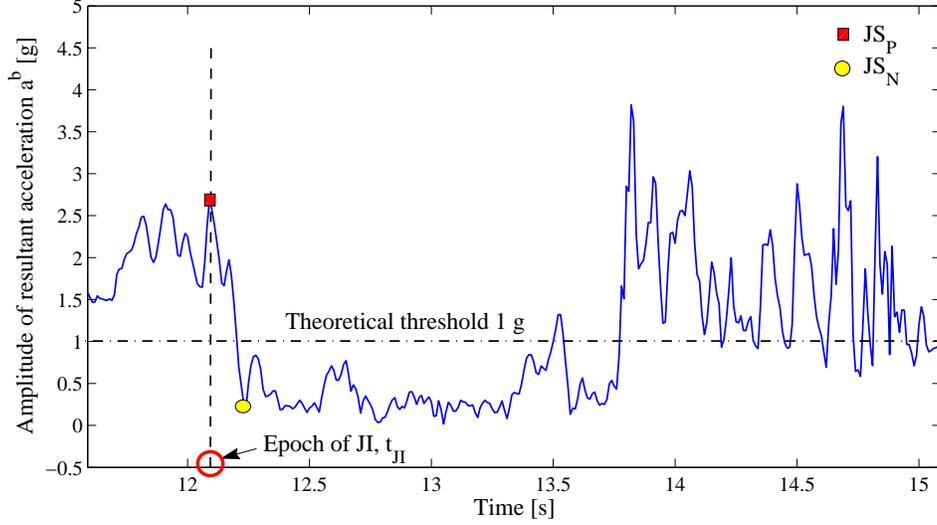


Figure 3.22: Primary detection.

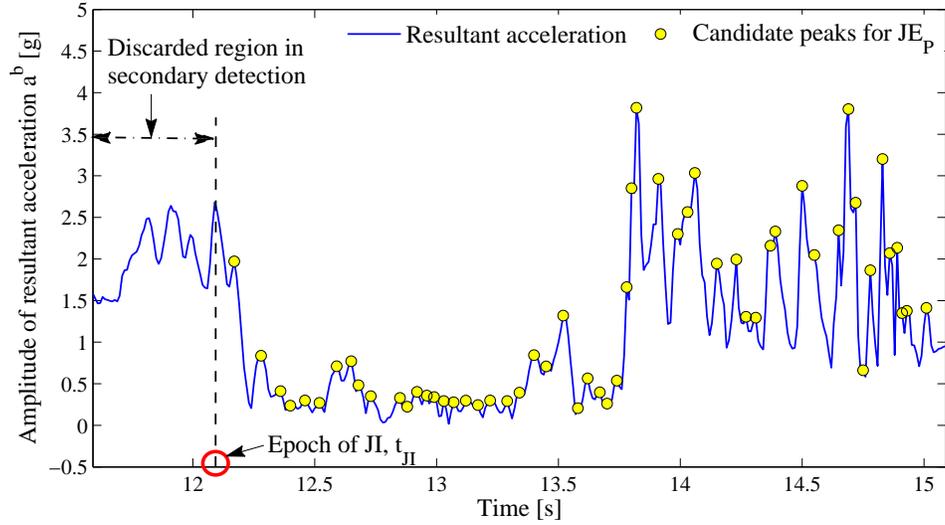
For each candidate peak, these four criteria values are calculated. To calculate the preceding and following window average \bar{a}_{pw}^{bI} and \bar{a}_{fw}^{bI} , the window spanning from t_{cs} to t_I and from t_I to t_{ce} is considered respectively. Two methods are proposed to assess the criteria/attributes and determine the most eligible positive and negative peak. The first method is a threshold based method and the second method is a probabilistic method, which is independent of any threshold. Both of these methods are demonstrated in the next section.

Similar to primary detection, JI serves as a reference for the secondary detection. For the JI of the given example, secondary detection means the detection of JE_P and JE_N and their epochs t_{JE}^P and t_{JE}^N . Figure 3.23a and 3.23b show the potential positive peaks for JE_P and the corresponding attribute values for each candidate. Note that for the detection of JE_P , all the positive peaks following the JI are the only viable candidates. A similar argument is valid for the negative peak candidates for JE_N .

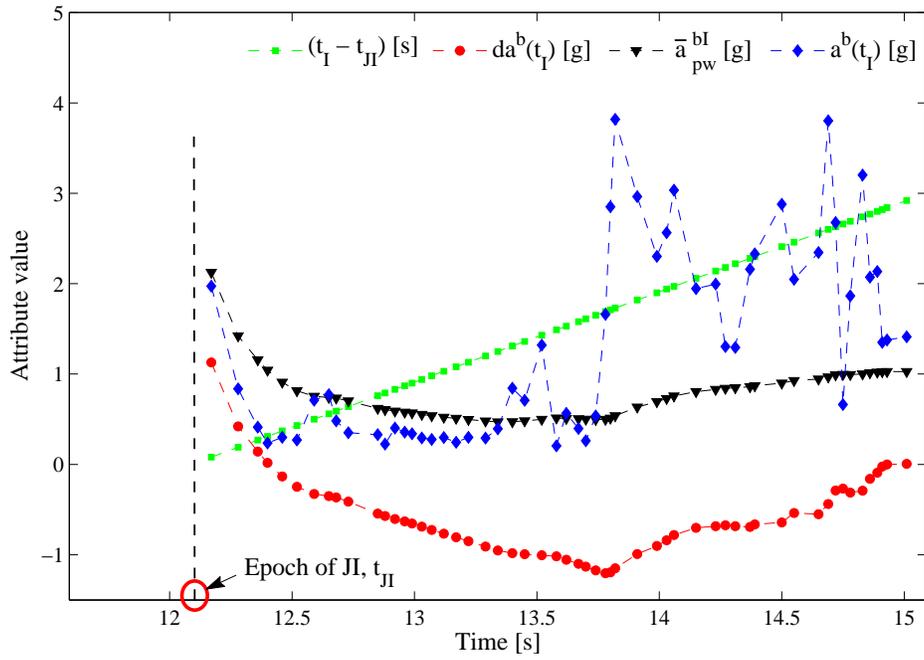
Threshold Based Method

As shown in Figure 3.23b, the attribute values for the candidate peaks for JE_P vary significantly. It can be understood that a potential candidate for JE_P at t_I should have a negative $da^b(t_I)$ with magnitude as high as possible,

3.8. Air Time Determination



(a)



(b)

Figure 3.23: (a) Potential positive peaks for JE_p . (b) Attribute value of the potential positive peaks.

3.8. Air Time Determination

\bar{a}_{pw}^{bI} should be as low as possible, and $a^b(t_I)$ should have a moderately high value. Therefore, three thresholds are set for these criteria. Among the set of candidate positive peaks that satisfies these thresholds, the peak nearest to JI is selected as JE_P . On the account of a reciprocal rationale, the farthest peak from JI is selected as JE_N from the set of the candidate negative peaks satisfying all the thresholds. The thresholds for the attribute values of the candidates for JE_P and JE_N at t_I are given in Table 3.3. Furthermore, Table 3.4 shows the thresholds if JS_P and JS_N are to be detected as secondary detection.

Table 3.3: Thresholds for secondary detection of JE_P and JE_N

Attribute	Threshold for JE_P	Threshold for JE_N
$t_I - t_{JI}$	Lowest	Highest
$da^b(t_I)$	< -0.25 g	< -0.25 g
\bar{a}_{pw}^{bI}	< 1 g	< 1 g
$a^b(t_I)$	> 1.50 g	< 0.50 g

Table 3.4: Thresholds for secondary detection of JS_P and JS_N

Attribute	Threshold for JS_P	Threshold for JS_N
$t_{JI} - t_I$	Lowest	Highest
$da^b(t_I)$	> 0.25 g	> 0.25 g
\bar{a}_{fw}^{bI}	< 1 g	< 1 g
$a^b(t_I)$	> 1.50 g	< 0.50 g

Using the predefined thresholds, detection of the epochs of JE_P and JE_N , i.e. t_{JE}^P and t_{JE}^N , can mathematically be represented as

$$t_{JE}^P = \min_{t_I} \left\{ \left\{ \arg_{t_I} \{ da^b(t_I) < -0.25\text{g} \} \cap \arg_{t_I} \{ \bar{a}_{pw}^{bI} < 1\text{g} \} \cap \arg_{t_I} \{ a^b(t_I) > 1.50 \} \right\} - t_{JI} \right\}, \quad t_I \in t_c^P \quad (3.17)$$

$$t_{JE}^N = \max_{t_I} \left\{ \left\{ \arg_{t_I} \{ da^b(t_I) < -0.25\text{g} \} \cap \arg_{t_I} \{ \bar{a}_{pw}^{bI} < 1\text{g} \} \cap \arg_{t_I} \{ a^b(t_I) < 0.50 \} \right\} - t_{JI} \right\}, \quad t_I \in t_c^N. \quad (3.18)$$

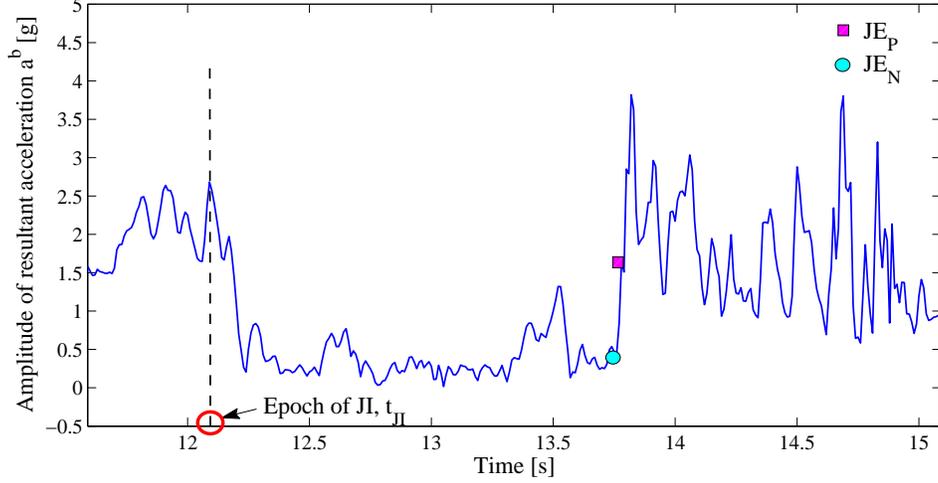


Figure 3.24: Secondary detection of JE_P and JE_N with threshold based method.

Similar expressions can be derived for t_{JS}^P and t_{JS}^N for the threshold based method. Figure 3.24 shows the detected JE_P at $t_{JE}^P = 13.78$ s and JE_N at $t_{JE}^N = 13.76$ s. It is evident from the figure that the detected peaks make reasonable sense as two of the characteristic points of the U-shaped region.

Probabilistic Method

The threshold based method is an effective and simple technique for secondary detection. However, the loss of generality cannot be avoided as is true for all threshold based techniques. Use of several thresholds makes the algorithm susceptible to error, especially for jumps with various aerial acrobatic maneuvers. As previously mentioned, a potential candidate for JE_P at t_I should have the highest $da^b(t_I)$ magnitude and the lowest \bar{a}_{pw}^{bI} . Unfortunately, a situation may occur whereby the candidate peaks possessing the highest $da^b(t_I)$ and the lowest \bar{a}_{pw}^{bI} are not the same. In such a case, the most eligible peak should be selected through a decision making procedure.

To accommodate all the above mentioned issues, a probabilistic method using MADM [48] is proposed. This method uses no thresholds and selects the best option among the alternatives, i.e. the candidate peaks. The idea of secondary detection by probabilistic means stems from the fact that it is possible to make an educated conjecture on the position of the characteristic

points only from the shape of the acceleration data. For example, without even looking at the y-axis magnitude, an educated eye should be able to detect the possible regions of the characteristic points as shown in Figure 3.20. Rather than directly discarding any point, the probabilistic method assesses and scores the possibility of all candidate peaks to be a characteristic point. Depending on the score, the optimum solution is selected. The most desirable advantage of this method is that it renders the characteristic point detection process to be very generalized and to be able to cope with a large variety of situations.

A decision making problem can be formulated for the secondary detection process considering the candidate peaks as alternatives to choose from. The decision making can be conducted using the four attribute values of each alternative. M-TOPSIS [49] is a modified version of the popular TOPSIS [48] method for MADM. This method effectively addresses the ‘Rank Inversion’ [49] problem of the TOPSIS method. Both the TOPSIS and M-TOPSIS methods, along with the rank inversion problem, are demonstrated in Appendix A. Due to the advantage over TOPSIS, M-TOPSIS is selected as the decision making technique for secondary detection.

To detect JE_P , the positive peaks shown in Figure 3.23a are fed to the MADM method as alternatives. The four attribute values for each alternative, as shown in Figure 3.23b, are used to construct the decision matrix. Preference information about the attributes, i.e. weights $w_i^c, i = 1, 2, 3, 4$, are predefined. The information regarding whether the attributes are cost type or benefit type is also provided to the MADM algorithm. Benefit attributes are those which are expected to be maximized and cost attributes are those which are expected to be minimized. Depending on the specific characteristic point, the attributes swap between benefit type and cost type. Table 3.5 and Table 3.6 lists all the attribute types and weights for secondary detection of JE_P, JE_N, JS_P and JS_N . Using the attribute type and weight information, the decision matrix is normalized and the weighted normalized ratings are calculated. The distances from the alternatives to the positive ideal solution (D_j^*) and to the negative ideal solution (D_j^-) are measured afterwards. Subsequently, the D^*D^- plane is constructed and the Optimized Ideal Reference Point (OIRP) is determined.

Figure 3.25 shows the $D_j^*D_j^-$ plane for JE_P detection. From this plane the distances between the OIRP and the alternatives (D_j^{OIRP}) are calculated. The color intensity of the alternatives shown in the figure is proportional to its distance from the OIRP. D_j^{OIRP} is sorted in ascending order and all the alternatives, i.e. candidate peaks, are ranked against it correspond-

3.8. Air Time Determination

ingly. The peak possessing the lowest D_j^{OIRP} is ranked 1 and considered as the optimal solution, i.e. JE_P , for the MADM problem. Figure 3.26 shows the top ranked peaks for secondary JE_P detection. Again, the color intensity of the peaks is proportional to D_j^{OIRP} . The epoch of the ranked 1 peak is determined as t_{JE}^P . In a similar manner, JE_N is detected and t_{JE}^N is determined. The result of secondary detection with the probabilistic method is shown in Figure 3.27. The detected $t_{JE}^P = 13.80$ s and $t_{JE}^N = 13.76$ s.

Table 3.5: Attribute type and weight for probabilistic secondary detection of JE_P and JE_N

For JE_P			For JE_N		
Attribute	Type	Weight	Attribute	Type	Weight
$t_I - t_{JI}$	Cost	0.05	$t_I - t_{JI}$	Benefit	0.08
$ da^b(t_I) $	Benefit	0.50	$ da^b(t_I) $	Benefit	0.50
\bar{a}_{pw}^{bI}	Cost	0.33	\bar{a}_{fw}^{bI}	Benefit	0.38
$a^b(t_I)$	Benefit	0.12	$a^b(t_I)$	Cost	0.04

Table 3.6: Attribute type and weight for probabilistic secondary detection of JS_P and JS_N

For JS_P			For JS_N		
Attribute	Type	Weight	Attribute	Type	Weight
$t_{JI} - t_I$	Cost	0.05	$t_{JI} - t_I$	Benefit	0.08
$ da^b(t_I) $	Benefit	0.50	$ da^b(t_I) $	Benefit	0.50
\bar{a}_{fw}^{bI}	Cost	0.33	\bar{a}_{pw}^{bI}	Benefit	0.38
$a^b(t_I)$	Benefit	0.12	$a^b(t_I)$	Cost	0.04

A very safe threshold can be useful from the computational point of view for the implementation of this MADM based probabilistic method even though it does not require a threshold. For example, a safe threshold of > 1 g for secondary JE_P or JS_P detection will eliminate a significant number of candidate positive peaks from being considered as alternatives for the decision making process. The reduced number of alternatives will cause reduced computation resulting in efficient use of resources.

3.8. Air Time Determination

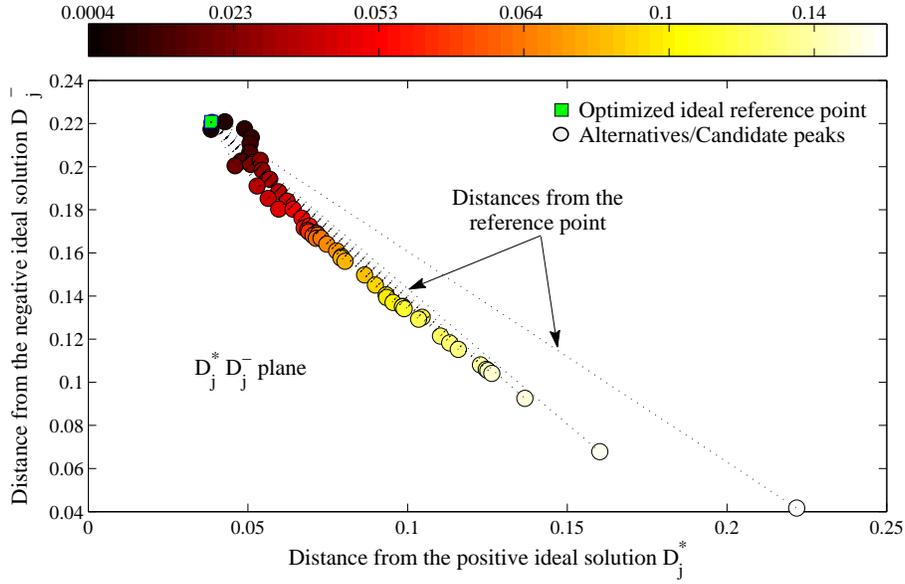


Figure 3.25: $D_j^* D_j^-$ plane for the secondary detection of JEP .

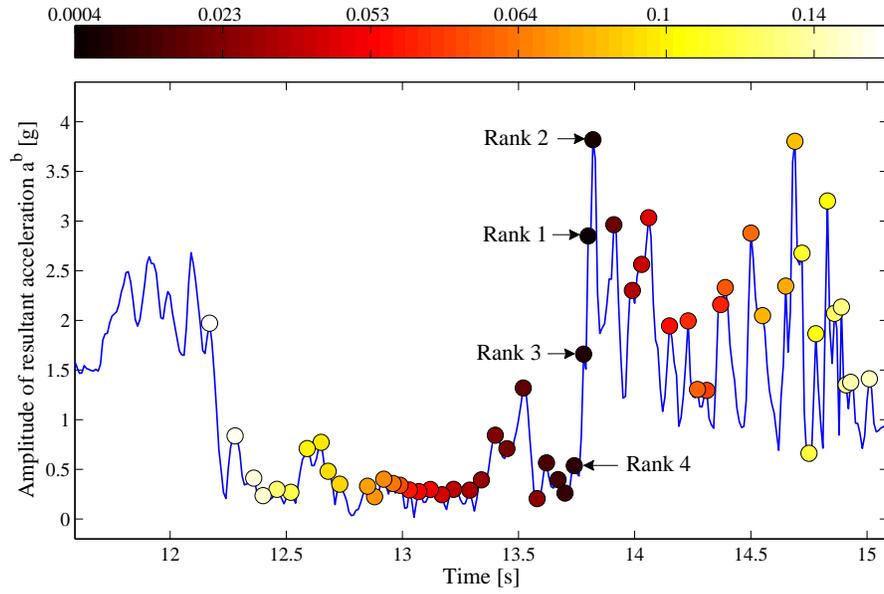


Figure 3.26: D_j^{OIRP} and rank of the top candidate peaks for JEP .

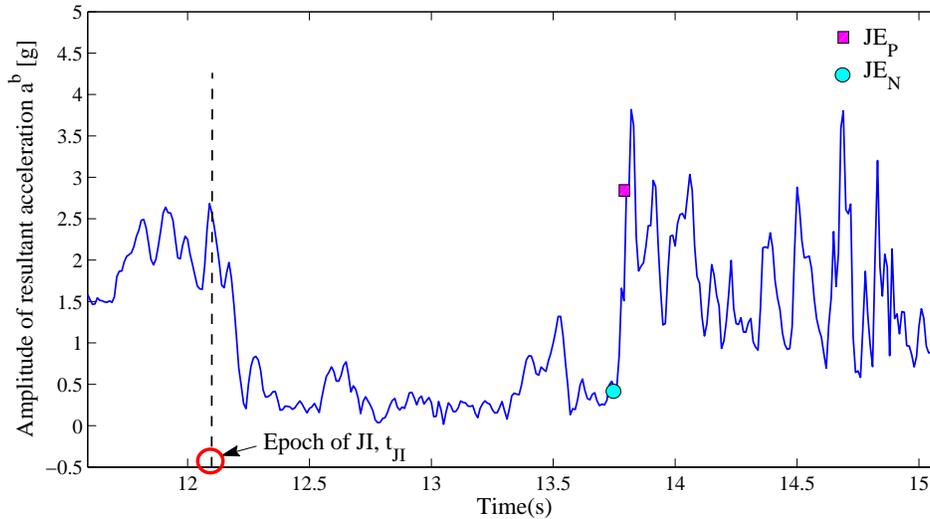


Figure 3.27: Secondary detection of JE_P and JE_N with probabilistic method.

3.9 Sensor and Visual Air Time

For snowboarding or skiing, Recon defines the jump start epoch as the moment when the skis/board are no longer in contact with the ground. Recon also defines the jump end epoch as the moment when any part of the skis/board come into contact with the ground again. The difference between the end and the start epochs defined in this manner leads to an AT which can be visually realized. The procedure for calculating AT from the JS_P , JS_N , JE_P and JE_N is described later in Subsection 3.10.2. However, up to this point, the jump start and end epochs have been discussed from the point of view of the sensor signals. The visual jump start and end epochs are not necessarily same as what the sensors detect as the jump start and end epochs. This is because the changes in the athlete's dynamics as detected by the sensors, does not always reflect what can be visually observed. For example, while landing, the athlete may have been observed to touch the ground even though no real impact was sensed by the sensors. Similarly, during take-off, a change in acceleration is typically sensed by the sensors prior to what is defined as the take-off by Recon. Logically, it is expected to have a constant bias in the visually realized AT compared to the AT determined from the sensor signal, which is termed as Sensor Air

Time (SAT). From the experimental data, a simple relationship is developed to convert the determined SAT into visual AT. For the rest of the discussion, AT will refer to what is visually observed unless otherwise stated.

3.10 Experimental Results and Comparison

3.10.1 Field Data Collection

Field data was collected to test the performance of the proposed jump detection and AT determination algorithm. The athlete had a Ripxx [4] unit strapped to his leg and the Recon goggles on his head. Both the devices were set to capture and log raw data from the available sensors. A video camera with 30 fps (frame per second) capture capability was set to record the jumps. Snowboard and ski jumps with different aero phase duration and including various aerial maneuvers, e.g. spinning, were conducted. The frame numbers of the skier leaving the ground and returning to the ground were determined from video playback. Since the frame rate is known, (difference in frame numbers / frame rate in fps) provides the AT in seconds as determined from the video. This video AT serves as the benchmark for comparing the performance of current Recon, Ripxx and proposed algorithms. However, the video AT is only accurate within a maximum error of $(0.033 \text{ s} + 0.033 \text{ s}) = 0.067 \text{ s}$. The error is twice ($1/\text{frame rate}$) since a single frame error may occur both at the jump start and end.

3.10.2 Visual AT from SAT

Data captured with the Recon goggles is fed to the proposed algorithm to detect the jump and determine the corresponding jump characteristic points. For secondary detection, the probabilistic method is used. The difference between t_{JS}^N and t_{JE}^N is used to determine the SAT, since JS_N/JE_N falls closest to the take-off/landing points defined by Recon. An example of the bias between visual AT and SAT is illustrated in Figure 3.28. The athlete's position and corresponding resultant acceleration are linked in the figure. The characteristic points detected by the proposed algorithm are also shown. Image 1 (frame 3409) is the first epoch when the accelerometers sense free fall. However, it is clear from the image that the snowboard is still in contact with the ground. Visually, the take-off is detected in image 2 (frame 3411). Hence, sensor take-off is at an earlier time than visual take-off. Similarly, images 4 and 5 exhibit the bias in sensor and visual landing. Therefore, it is evident that the visual AT would always be shorter than SAT.

3.10. Experimental Results and Comparison

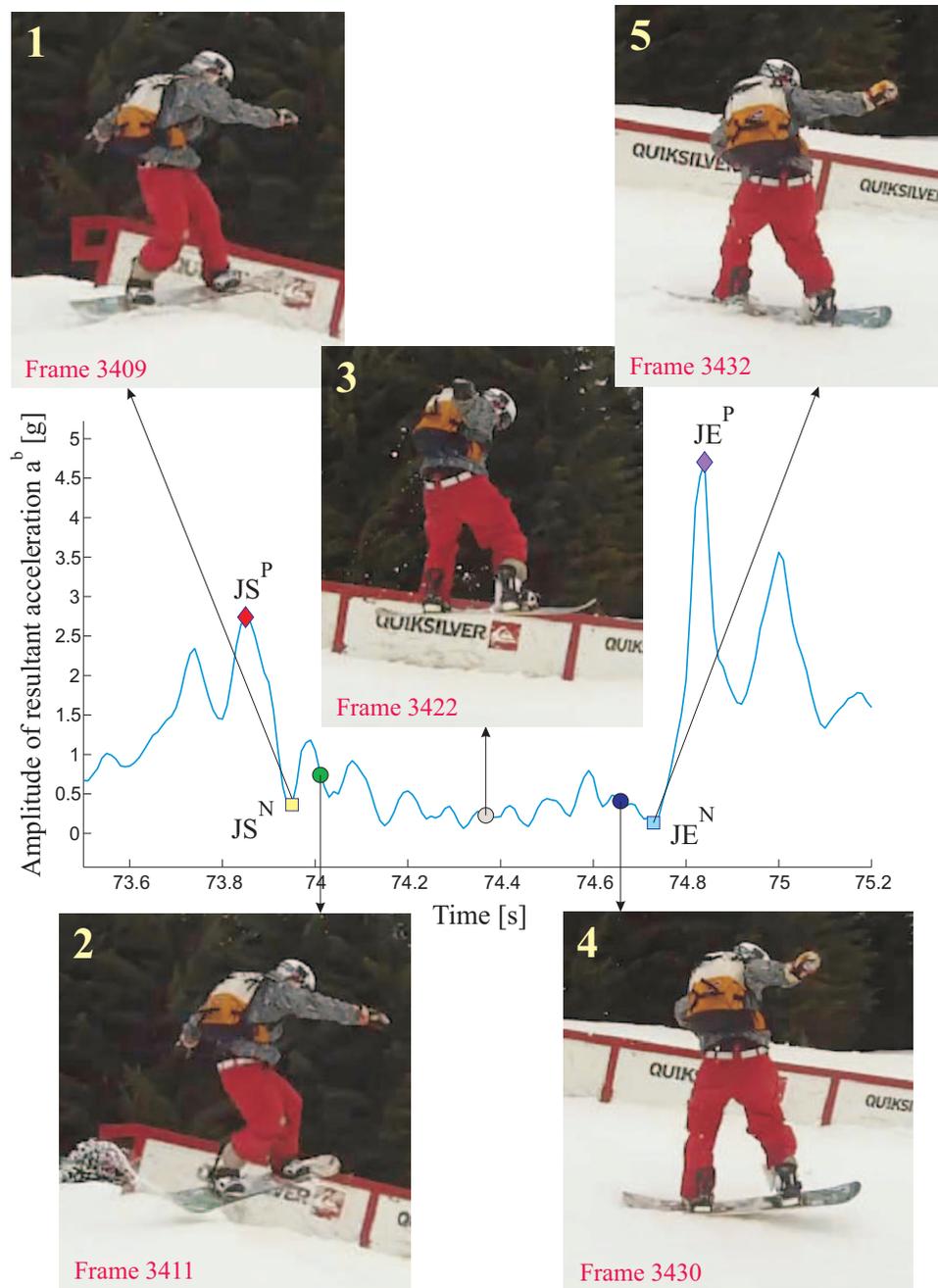


Figure 3.28: Experimental video and captured sensor resultant acceleration.

3.10. Experimental Results and Comparison

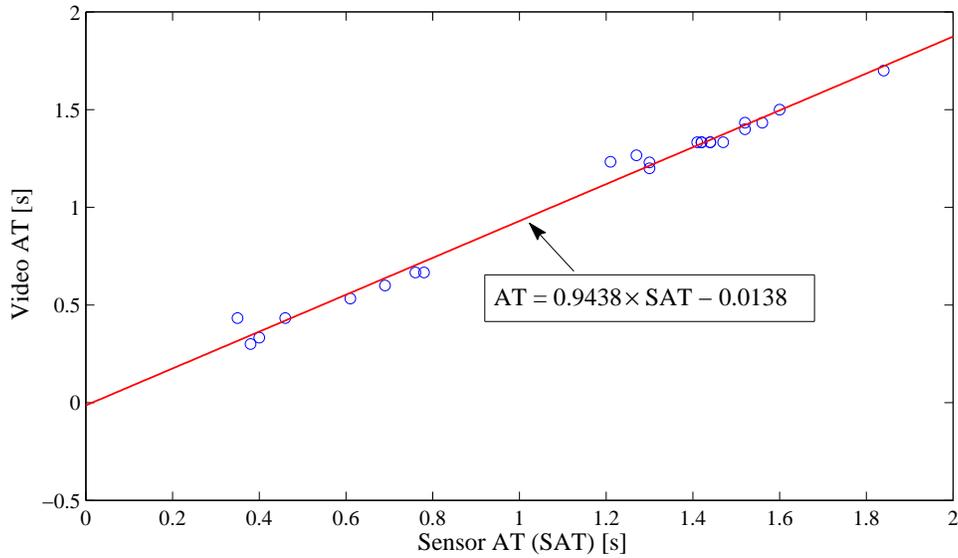


Figure 3.29: Video AT vs SAT.

To determine the AT from SAT, the determined SATs are plotted against the ATs from video. The plot is shown in Figure 3.29. A first order linear curve is fitted to develop a relation between the visual AT and SAT. The developed relation, $AT = 0.9438 \times SAT - 0.0138$ s, can be used to determine the visual AT from determined SAT. Of course, the relationship could be refined with acquisition of more jump data for any specific sport application. Furthermore, the interpretation or use of the characteristic points along with the SAT will vary with different definitions of take-off/landing. The developed relation between visual AT and SAT is most suitable for snowboard and ski jumps according to Recon's definition.

3.10.3 Performance Comparison

The algorithm for jump detection and AT detection are compared in different aspects. The most important aspect is the accuracy of the AT. Similar to the proposed algorithm, the Recon jump detection and AT determination algorithm, which is a threshold based method, was implemented in Matlab[®] for off-line processing. The Recon algorithm is proprietary and, therefore, cannot be described in detail here. The recorded accelerometer data sets collected with the Recon goggles are fed to both the Recon algorithm and the proposed algorithm to generate AT. ATs for the proposed

3.10. Experimental Results and Comparison

algorithm are determined by converting the SATs. AT for the test jumps are also available from the Ripxx log sheet. These three ATs are compared with the ATs deduced from the video. The error percentages with respect to the true AT (video) are also calculated. The results are shown in Table 3.7. Ripxx data was not available for the first two data sets. Note that an error of 100% indicates that a jump that in fact occurred, was not detected.

From Table 3.7 it can be seen that the proposed algorithm is more efficient in jump detection and has better accuracy in AT determination than the other two. Among the 25 jumps, the proposed algorithm successfully detected 23 of them (92% accuracy). On the other hand, the Recon algorithm detected only 15 jumps out of the 25 (60% accuracy). Among the available 23 jumps, the Ripxx algorithm was only able to successfully detect 10 of them (43.5% accuracy). Moreover, the Recon and Ripxx algorithms have shown an AT determination average error of 8.4% and 39.7% respectively where as the proposed algorithm has an average error of 4.8%. It is very important to note that in the average error calculations the undetected jumps for each algorithm were excluded. Therefore, the effect of undetected jumps does not reflect on the average error.

Figure 3.30 shows, for all jumps, the measured AT error in seconds for all three algorithms. Apart from very few exceptions, error for the proposed jump detection and AT determination algorithm is well within the maximum allowable error limit of ± 0.1 s, whereas the other two exhibit considerable error and often exceed the specification of ± 0.1 s.

The second important aspect when comparing algorithms is their use of resources. Depending on the jump detection method the number of false wake up calls to the processor varies from algorithm to algorithm. A false wake up means the realization of a false jump detection after most of the computation is conducted. For example, the proposed algorithm will cause a false wake up call if any of the characteristic points are not detected for the largest characteristic window. No information on Recon false wake up criteria can be provided in this study due to proprietary issues. For a given data set, if the number of false wake up calls is significantly large, more power will be unnecessarily drained, leading to a lower battery life. Therefore, the number of false wake up calls to the processor is also determined and used as an algorithm comparison criterion. The comparison is shown in Table 3.8. Since the Ripxx algorithm for AT detection is unknown, the comparison for Ripxx on this criterion could not be done. From Table 3.8 it is clear that the proposed algorithm generates far fewer false wake up calls as compared to the Recon algorithm. The Recon algorithm generates a total of 21 false wake up calls while operating on the 14 data sets containing 25 jumps in

3.10. Experimental Results and Comparison

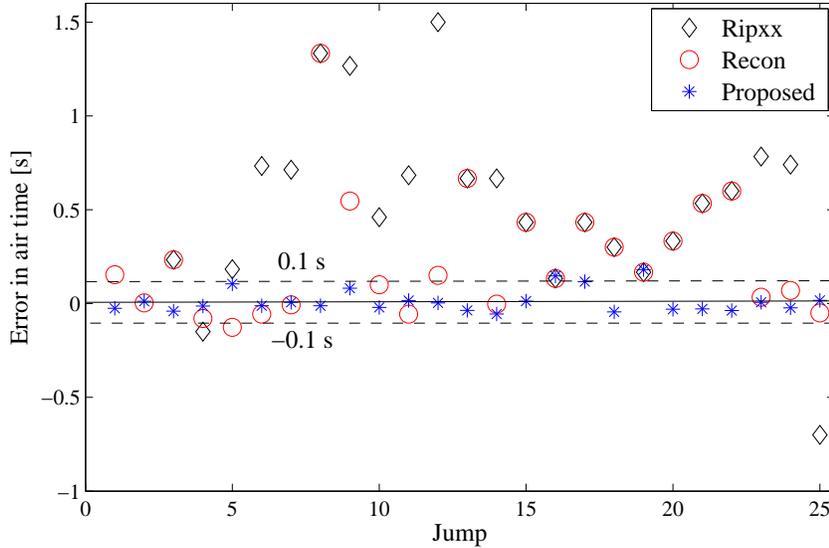


Figure 3.30: Error in AT comparison among the Recon, Ripxx and proposed algorithm.

total. Therefore, the current Recon jump detection algorithm unnecessarily wakes the processor $21 * 100 / (21 + 25) = 45.6\%$ of the time. The 11th data set contains 12 jumps which is far more than any of the other data sets. This is the reason for the large number of false wake up calls for this set for the Recon algorithm. On the other hand, the proposed algorithm generated no false wake up calls for any of the 14 data sets.

The third criterion for algorithm performance comparison is the number of false jumps detected and true jumps undetected. As shown in Table 3.7, any true jump undetected also results in a 100% error in the corresponding AT calculation. The results are shown in Table 3.9 for the collected data set. In the 14 data sets (containing 25 jumps), the Recon algorithm had 11 wrong detections (44%), i.e. true jumps undetected plus false jump detected. For Ripxx, the number of wrong detections is 13 (52%). On the other hand, the proposed algorithm has shown significantly better performance by only reporting 2 wrong detections, (8%) while operating over the 14 data sets.

From these results, it is evident that the proposed method has comprehensive supremacy in AT determination performance over the other two algorithms. The false wake up calls are also reduced significantly by the proposed method. Except for the 11th data set, the proposed method also

exhibits efficiency in avoiding false detection or missing true jumps. Moreover, the two missed jumps in the 11th set had extremely brief AT duration and are likely to be insignificant to the goggle user. Therefore, it can be concluded from the available experimental results that the proposed algorithm performs significantly better than the current algorithm Recon is using and the algorithm of the commercially available Ripxx unit.

3.11 Implementation Aspects for Other Devices/Applications

The proposed methods for primary and secondary detection can be altered according to the sensor signal characteristics. As mentioned earlier, the jump detection thresholds are specifically customized for Recon goggles only to guarantee the detection of landing. This is done to minimize the mandatory computation as well as the possibility of false detection. Therefore, the closest peak method for primary detection and the probabilistic method for secondary detection were proposed. However, if the scenario is such that both the JI near jump start and end are guaranteed to be detected, the closest peak method can be used for both primary and secondary detection. On the other hand, if scenarios are observed where the closest peak method is not as accurate as expected, the probabilistic method can be used for both primary and secondary detection. In such a case, after secondary detection, any of the detected secondary characteristic points can be selected as reference points to serve as the JI for the primary detection process.

3.11. Implementation Aspects for Other Devices/Applications

Table 3.7: Comparison of AT among Recon, Ripxx and the proposed algorithm.

Data set	Video AT(s)	Recon AT(s) [%error]	Ripxx AT(s) [%error]	Proposed AT(s) [%error]
1	1.4333	1.28 [10.6]	- [-]	1.4585 [1.8]
2	1.4333	1.43 [0.2]	- [-]	1.4207 [0.9]
3	1.3333	1.10 [17.5]	1.10 [17.5]	1.3735 [3.0]
4	1.2000	1.28 [6.7]	1.35 [12.5]	1.2131 [1.1]
5	1.2333	1.36 [10.3]	1.05 [14.9]	1.1281 [8.5]
6	1.3333	1.39 [4.2]	0.60 [55.0]	1.3452 [0.9]
7	1.3333	1.34 [0.5]	0.62 [53.5]	1.3263 [0.5]
8	1.3333	0.00 [100.0]	0.00 [100.0]	1.3452 [0.9]
9	1.2667	0.72 [43.1]	0.00 [100.0]	1.1848 [6.5]
10	1.4000	1.30 [7.1]	0.94 [32.8]	1.4207 [1.5]
11-a	1.3333	1.39 [4.2]	0.65 [51.2]	1.3169 [1.2]
11-b	1.5000	1.35 [10.0]	0.00 [100.0]	1.4962 [0.2]
11-c	0.6667	0.00 [100.0]	0.00 [100.0]	0.7034 [5.5]
11-d	0.6667	0.67 [0.5]	0.00 [100.0]	0.7223 [8.3]
11-e	0.4333	0.00 [100.0]	0.00 [100.0]	0.4203 [3.0]
11-f	0.1333	0.00 [100.0]	0.00 [100.0]	0.00 [100.0]
11-g	0.4333	0.00 [100.0]	0.00 [100.0]	0.3165 [26.9]
11-h	0.3000	0.00 [100.0]	0.00 [100.0]	0.3448 [14.9]
11-i	0.1667	0.00 [100.0]	0.00 [100.0]	0.0000 [100.0]
11-j	0.3333	0.00 [100.0]	0.00 [100.0]	0.3637 [9.1]
11-k	0.5333	0.00 [100.0]	0.00 [100.0]	0.5619 [5.4]
11-l	0.6000	0.00 [100.0]	0.00 [100.0]	0.6374 [6.2]
12	1.3333	1.30 [2.5]	0.55 [58.7]	1.3263 [0.5]
13	1.7000	1.63 [4.1]	0.96 [43.5]	1.7227 [1.3]
14	1.2300	1.28 [4.1]	1.93 [56.9]	1.2131 [1.4]
Avg. error(s) [%error]		0.111 [8.4]	0.538 [39.7]	0.033 [4.8]

3.11. Implementation Aspects for Other Devices/Applications

Table 3.8: Comparison of number of false processor wake up calls for Recon algorithm and the proposed algorithm.

Data set	Recon	Proposed
1	1	0
2	0	0
3	1	0
4	0	0
5	1	0
6	0	0
7	0	0
8	2	0
9	0	0
10	1	0
11	9	0
12	3	0
13	0	0
14	3	0
Total	21	0

3.11. Implementation Aspects for Other Devices/Applications

Table 3.9: Comparison of the number of false detected jumps + true undetected jumps among Recon, Ripxx and the proposed algorithm.

Data set	Recon	Ripxx	Proposed
1	0	-	0
2	0	-	0
3	0	1	0
4	0	0	0
5	0	0	0
6	0	0	0
7	0	0	0
8	1	1	0
9	0	1	0
10	0	0	0
11	9	10	2
12	0	0	0
13	0	0	0
14	1	0	0
Total	11	13	2

Chapter 4

Trajectory Determination

The determination of jump horizontal distance, height and drop corresponds to the task of determining the trajectory of the athlete during a jump. Once the trajectory is determined, the calculation of the desired parameters is straight forward. In this chapter a GPS/INS integrated algorithm is proposed for trajectory determination with the Recon goggles. First, the coordinate frames involved in the algorithm are presented. Afterwards, the basic INS algorithm and pertinent important factors are demonstrated. Finally, the proposed algorithm with detailed performance analysis is presented.

4.1 Coordinate Frames

The local level frame has been chosen as the navigation frame (index n). The local level frame is defined as the local North-East-Down (NED) axes. The north axis is labeled X , the east is labeled Y , and the down is labeled Z . As the reference frame for positioning, the Earth Centered Earth Fixed (ECEF) frame is used, which is indexed by e . The ECEF frame is defined as [50]:

- origin - at the mass center of the earth
- X axis - pointing towards the Greenwich meridian, in the equatorial plane
- Y axis - 90° east of Greenwich meridian, in the equatorial plane
- Z axis - axis of rotation of the reference ellipsoid.

Measurement from the gyroscope is done with respect to an inertial frame which is indexed by i . The inertial frame is considered to be rotationally fixed. Coordinates in the ECEF frame can be transformed to the inertial frame by a negative rotation about the Z axis of the frame by the amount of the Greenwich Mean Sidereal Time (GMST) [50]. The navigation frame, ECEF frame and inertial frame are depicted in Figure 4.1. The body frame is indexed by b and defined as shown in Figure 1.4. The triaxial sensors are

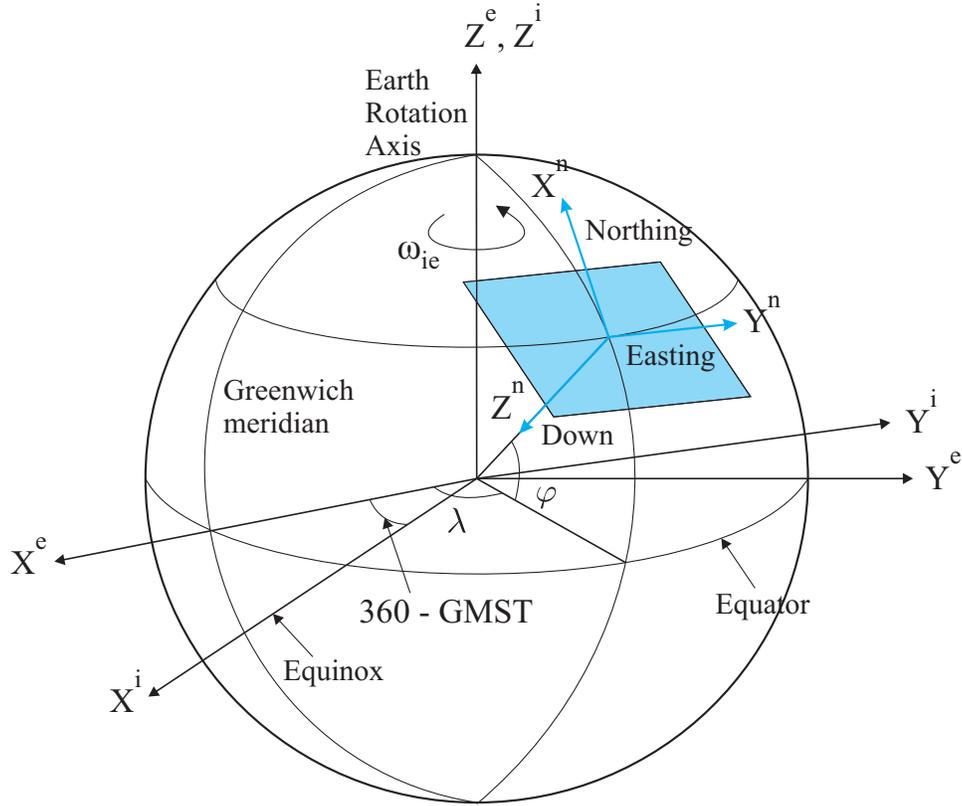


Figure 4.1: Coordinate frames.

aligned entirely with the body frame and the body frame is considered to be rigidly attached to the athlete. It is important to note that all of these frames follow right handedness.

4.2 Rotation Matrix

The most convenient way to represent the attitude of the body frame with respect to the navigation frame is through a set of three Euler angles [51]. These angles are commonly known as roll (ϕ), pitch (θ) and yaw (ψ) which represent rotation along the X , Y and Z axes accordingly. Any arbitrarily rotated body frame attitude with respect to the navigation frame can be represented as three consecutive coordinate rotations. The three coordinate rotations along the X , Y and Z axes can be represented in matrix

4.2. Rotation Matrix

form respectively as follows:

$$\mathbf{R}_X(\phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\phi) & \sin(\phi) \\ 0 & -\sin(\phi) & \cos(\phi) \end{bmatrix} \quad (4.1)$$

$$\mathbf{R}_Y(\theta) = \begin{bmatrix} \cos(\theta) & 0 & -\sin(\theta) \\ 0 & 1 & 0 \\ \sin(\theta) & 0 & \cos(\theta) \end{bmatrix} \quad (4.2)$$

$$\mathbf{R}_Z(\psi) = \begin{bmatrix} \cos(\psi) & \sin(\psi) & 0 \\ -\sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix}. \quad (4.3)$$

Multiplication of the matrices in Equation 4.1, 4.2 and 4.3 results in a rotation matrix. A rotation matrix is a matrix whereby the multiplication with a vector rotates the vector while preserving its length [51]. In this research a ZYX intrinsic rotation convention is followed. This convention means that rotation is applied along the Z , Y and X axes sequentially. Intrinsic rotation means that rotations are applied with respect to the rotated frame itself, not with respect to the reference frame. Therefore, the rotation matrix to convert any vector represented in the body frame into a vector represented in the navigation frame is derived as

$$\begin{aligned} \mathbf{R}_b^n &= \mathbf{R}_Z(\psi)\mathbf{R}_Y(\theta)\mathbf{R}_X(\phi) \\ &= \begin{bmatrix} c_\theta c_\psi & s_\phi s_\theta c_\psi - c_\phi s_\psi & c_\phi s_\theta c_\psi + s_\phi s_\psi \\ c_\theta s_\psi & s_\phi s_\theta s_\psi + c_\phi c_\psi & c_\phi s_\theta s_\psi - s_\phi c_\psi \\ -s_\theta & c_\theta s_\phi & c_\theta c_\phi \end{bmatrix} \end{aligned} \quad (4.4)$$

where $\cos(arg)$ and $\sin(arg)$ are represented as c_{arg} and s_{arg} .

It is very important to note that the matrix multiplication sequence is not commutative. Any fixed amount of roll, pitch and yaw can result in completely different results. For example, the coordinate frame shown in Figure 4.2 is rotated in ZYX and ZXY sequences both with 90° roll, pitch and yaw. The resultant attitudes are entirely different due to the interchange in the sequential rotations along X and Y axis as shown in Figure 4.3 and 4.4.

4.2. Rotation Matrix

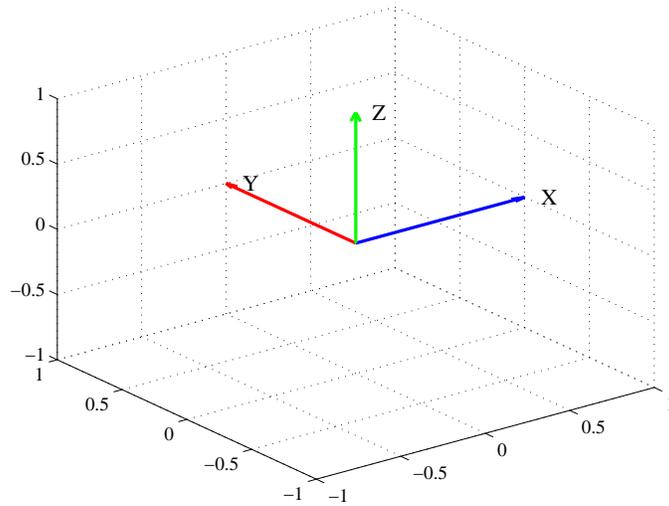


Figure 4.2: Coordinate frame before rotation.

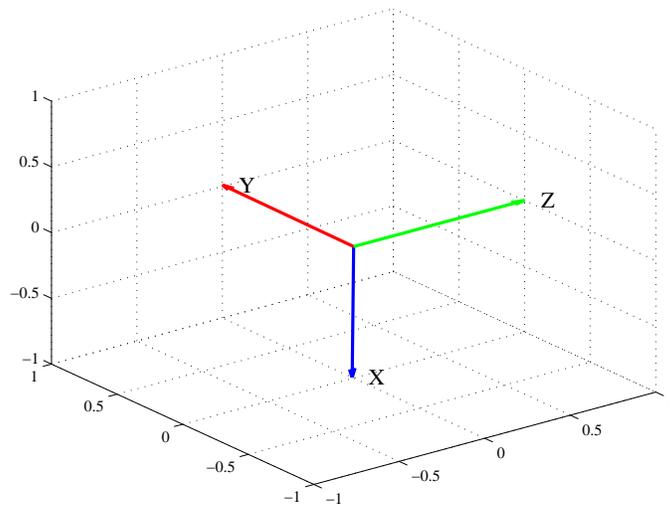


Figure 4.3: Coordinate frame after rotations in ZYX sequence where roll = pitch = yaw = 90° . Click on the image to see sequential rotations.

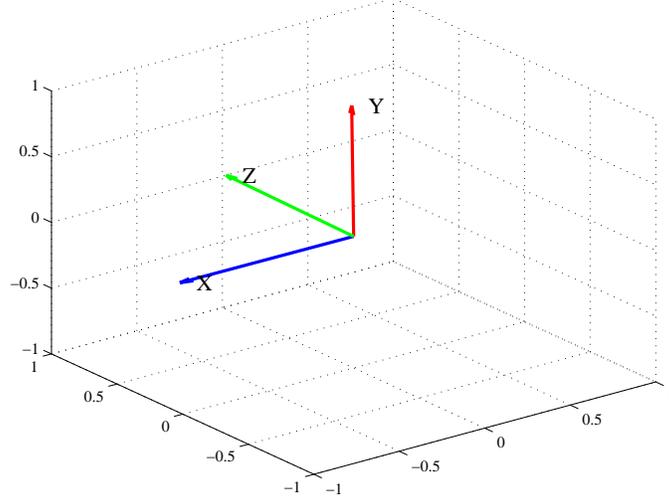


Figure 4.4: Coordinate frame after rotations in ZXY sequence where roll = pitch = yaw = 90° . Click on the image to see sequential rotations.

4.3 Body Frame Attitude Computation

4.3.1 Rotation Matrix Update

To compute the rotating body frame attitude with respect to the navigation frame from the measured angular rate vector $\boldsymbol{\omega}$, the rotation matrix must be updated. In order to update the rotation matrix a differential equation of the following form [52] needs to be solved with

$$\dot{\mathbf{R}} = \mathbf{R}[\boldsymbol{\omega} \times] \quad (4.5)$$

where $\boldsymbol{\omega} \times$ is the skew-symmetric form of the angular rate vector. Note that the subscript and superscript of the rotation matrix \mathbf{R}_b^n are left out for convenience. Over an update period from time t_k to t_{k+1} , the solution of the equation can be written as

$$\mathbf{R}_{k+1} = \mathbf{R}_k \exp\left(\int_{t_k}^{t_{k+1}} [\boldsymbol{\omega} \times] dt\right). \quad (4.6)$$

If it is assumed that the direction of the angular rate vector $\boldsymbol{\omega}$ remains fixed over the update interval, the integral of the skew symmetric matrix of the angular rate is written as

$$[\boldsymbol{\sigma} \times] = \int_{t_k}^{t_{k+1}} [\boldsymbol{\omega} \times] dt. \quad (4.7)$$

4.3. Body Frame Attitude Computation

where $\boldsymbol{\sigma}$ is the orientation vector. Hence

$$\mathbf{R}_{k+1} = \mathbf{R}_k \exp[\boldsymbol{\sigma} \times] \quad (4.8)$$

$$= \mathbf{R}_k \mathbf{A}_k. \quad (4.9)$$

Therefore, \mathbf{A}_k is the rotation matrix that transforms a vector in the body frame at the epoch t_{k+1} into the body frame at epoch t_k . Here, t_k and t_{k+1} are the times associated with the start and end of the computation cycle respectively. The exponential term in Equation 4.8 is expanded afterwards, which gives

$$\begin{aligned} \mathbf{A}_k &= \mathbf{I} + [\boldsymbol{\sigma} \times] + \frac{1}{2!} [\boldsymbol{\sigma} \times]^2 + \frac{1}{3!} [\boldsymbol{\sigma} \times]^3 + \frac{1}{4!} [\boldsymbol{\sigma} \times]^4 \dots \\ &= \mathbf{I} + [\boldsymbol{\sigma} \times] + \frac{1}{2!} [\boldsymbol{\sigma} \times]^2 - \frac{1}{3!} \sigma^2 [\boldsymbol{\sigma} \times] - \frac{1}{4!} \sigma^2 [\boldsymbol{\sigma} \times]^2 \dots \\ &= \mathbf{I} + \left(1 - \frac{1}{3!} \sigma^2 + \frac{1}{5!} \sigma^4 - \dots\right) [\boldsymbol{\sigma} \times] + \left(\frac{1}{2!} - \frac{1}{4!} \sigma^2 + \frac{1}{6!} \sigma^4 - \dots\right) [\boldsymbol{\sigma} \times]^2 \end{aligned} \quad (4.10)$$

$$= \mathbf{I} + \frac{1}{\sigma} \sin(\sigma) [\boldsymbol{\sigma} \times] + \frac{1}{\sigma^2} \{1 - \cos(\sigma)\} [\boldsymbol{\sigma} \times]^2 \quad (4.11)$$

where \mathbf{I} is the identity matrix. The exact representation of the rotation matrix, which relates body frame attitude at times t_k and t_{k+1} , is provided in Equation 4.11. However, this exact form can not be directly implemented onboard the goggles' processor due to the sine and cosine functions. Therefore, Equation 4.11 is approximated by Equation 4.10 where only the first three terms of the infinite series are used.

4.3.2 Orientation Vector Computation

In the previous section, it was assumed that the direction of the angular vector $\boldsymbol{\omega}$ remains constant during the update cycle. However, the direction of $\boldsymbol{\omega}$ does not remain fixed in space as the body frame is rotating. According to Bortz [53] the rate of change of the orientation vector $\dot{\boldsymbol{\sigma}}$ is

$$\dot{\boldsymbol{\sigma}} = \boldsymbol{\omega} + \dot{\boldsymbol{\varepsilon}}. \quad (4.12)$$

Here $\dot{\boldsymbol{\varepsilon}}$ (non-commutativity rate vector) is a component of $\dot{\boldsymbol{\sigma}}$ due to non-inertially measurable angular motion. The angular rate vector $\boldsymbol{\omega}$ is provided by the triaxial gyroscope which has its input axes coincident with the axes of the rotating body frame.

4.3. Body Frame Attitude Computation

Differentiating Equation 4.11, where $\dot{\mathbf{A}}_k = \mathbf{A}_k[\boldsymbol{\omega} \times]$, and manipulating vectors gives [52]

$$\dot{\boldsymbol{\sigma}} = \boldsymbol{\omega} + \frac{1}{2}\boldsymbol{\sigma} \times \boldsymbol{\omega} + \frac{1}{\sigma^2}\left[1 - \frac{\sigma \sin(\sigma)}{2(1 - \cos(\sigma))}\right]\boldsymbol{\sigma} \times \boldsymbol{\sigma} \times \boldsymbol{\omega}. \quad (4.13)$$

Together, the second and third terms on the right hand side of Equation 4.13 account for the non-commutativity vector. After a series of rotations, as a result of the non-commutativity terms, the final orientation is dependent on both the individual rotations and the order in which they have occurred [52]. Ignagni in [54] has demonstrated that for practical applications, where the rotation matrix is computed in an iterative manner and integration of the orientation vector rate is conducted over a brief period of time, Equation 4.13 can be simplified to

$$\dot{\boldsymbol{\sigma}} = \boldsymbol{\omega} + \frac{1}{2}\boldsymbol{\alpha} \times \boldsymbol{\omega} \quad (4.14)$$

where

$$\boldsymbol{\alpha} = \int_{t_k}^{t_{k+1}} \boldsymbol{\omega} dt. \quad (4.15)$$

Ignagni has also shown that the orientation vector can be found from direct integration of Equation 4.14 rather than solving the differential equation in Equation 4.13. The second term in Equation 4.14 is the simplified form of the non-commutativity rate vector.

4.3.3 Coriolis Effect

In inertial navigation calculations, the effects of the earth's rotation with respect to the inertial frame and the rotation of the navigation frame with respect to the earth, should be accounted for. These effects are known as the Coriolis effects. The earth's angular velocity with respect to the inertial frame and as observed from the navigation frame can be represented as

$$\boldsymbol{\omega}_{ie}^n = \begin{bmatrix} \Omega_E \cos(\varphi) \\ 0 \\ -\Omega_E \sin(\varphi) \end{bmatrix} \quad (4.16)$$

and the angular velocity of the navigation frame with respect to the earth and as observed from the navigation frame is

$$\boldsymbol{\omega}_{en}^n = \begin{bmatrix} \dot{\lambda} \cos(\varphi) \\ \dot{\varphi} \\ -\dot{\lambda} \sin(\varphi) \end{bmatrix} \quad (4.17)$$

where

- φ = latitude
- λ = longitude
- Ω_E = earth's rotation rate 7.292115×10^{-5} rad/s.

Therefore, the angular velocity of the navigation frame with respect to the inertial frame is

$$\boldsymbol{\omega}_{in}^n = \boldsymbol{\omega}_{ie}^n + \boldsymbol{\omega}_{en}^n. \quad (4.18)$$

This angular velocity $\boldsymbol{\omega}_{in}^n$ is important to account for because the gyroscope sensors measure the angular rate of the body frame with respect to the inertial frame rather than the navigation frame or body frame.

4.4 Inertial Navigation Equations

The rate of change in velocity of the body frame with respect to the earth expressed in the navigation frame is represented as [52]

$$\dot{\mathbf{v}}^n = \mathbf{a}^n - (2\boldsymbol{\omega}_{ie}^n + \boldsymbol{\omega}_{en}^n) \times \mathbf{v}^n + \mathbf{g}_l^n \quad (4.19)$$

where

- \mathbf{v}^n = velocity of the body frame with respect to the earth expressed in local level frame
 $= [v_N \ v_E \ v_D]^T$
- \mathbf{a}^n = acceleration of body frame observed from navigation frame
 $= [f_N \ f_E \ f_D]^T$
- \mathbf{g}_l^n = plumb bob gravity.

Plumb bob gravity is defined as

$$\mathbf{g}_l^n = \mathbf{g}^n - \frac{\Omega_E^2 R_0(\varphi)}{2} [\sin(2\varphi) \ 0 \ \cos(2\varphi)]^T. \quad (4.20)$$

Here

- \mathbf{g}^n = normal gravity = $[0 \ 0 \ 9.81]^T$ m/s²
- $R_0(\varphi)$ = earth's radius at latitude φ
 $= (\sqrt{(R_e^2 \cos(\varphi))^2 + (R_p^2 \sin(\varphi))^2}) / (\sqrt{(R_e \cos(\varphi))^2 + (R_p \sin(\varphi))^2})$
- R_e = earth's equatorial radius (6378137 m)
- R_p = earth's polar radius (6356752.3142 m).

4.4. Inertial Navigation Equations

The change in velocity \mathbf{v}^n during the update interval between t_k and t_{k+1} can be calculated from integrating Equation 4.19 which is represented as

$$\Delta \mathbf{v}_{k+1}^n = \int_{t_k}^{t_{k+1}} \mathbf{a}^n dt - \int_{t_k}^{t_{k+1}} (2\boldsymbol{\omega}_{ie}^n + \boldsymbol{\omega}_{en}^n) \times \mathbf{v}_k^n dt + \int_{t_k}^{t_{k+1}} \mathbf{g}_l^n dt. \quad (4.21)$$

To solve Equation 4.21, the three integral terms need to be calculated. The first integral term calculation needs further derivation, whereas the integral of the second and the third term can be calculated directly.

The acceleration vector of the body frame observed from the body frame can be transformed to an acceleration vector observed from the navigation frame by multiplication with the rotation matrix, i.e.

$$\mathbf{a}^n = \mathbf{R}_b^n \mathbf{a}^b \quad (4.22)$$

where $\mathbf{a}^b = [a_x^b \ a_y^b \ a_z^b]^T$ is the acceleration vector of the body frame observed from the body frame. Now, the incremental velocity change of the body frame as observed from the navigation frame is defined as

$$\begin{aligned} \mathbf{u}_k^n &= \int_k^{k+1} \mathbf{R}_b^n \mathbf{a}^b dt \\ &= \int_k^{k+1} \mathbf{R}_k \mathbf{A}_k \mathbf{a}^b dt. \end{aligned} \quad (4.23)$$

As the rotation matrix \mathbf{R}_b^n varies continuously with time over the update interval, it can be written in terms of the matrices \mathbf{R}_k and \mathbf{A}_k . The matrix \mathbf{R}_k is the value of \mathbf{R}_b^n at time t_k and \mathbf{A}_k is a matrix representing the transformation of the body frame at time t_{k+1} to the body frame at the start of the update interval t_k . Equation 4.23 can be revised as

$$\mathbf{u}_k^n = \mathbf{R}_k \int_k^{k+1} \mathbf{A}_k \mathbf{a}^b dt. \quad (4.24)$$

Utilizing Equation 4.10 and ignoring second and higher order terms

$$\mathbf{u}_k^n = \mathbf{R}_k \left(\int_k^{k+1} \mathbf{a}^b dt + \int_k^{k+1} [\boldsymbol{\sigma} \times] \mathbf{a}^b dt \right). \quad (4.25)$$

The incremental velocity change of the body frame as observed from the body frame is

$$\mathbf{u}_k^b = \int_k^{k+1} \mathbf{a}^b dt. \quad (4.26)$$

4.5. Rotation Angle Computation

Using integration by parts, Equation 4.25 and Equation 4.26 can be used to write

$$\mathbf{u}_k^n = \mathbf{R}_k(\mathbf{u}_k^b + \frac{1}{2}[\boldsymbol{\sigma} \times] \mathbf{u}_k^b + \frac{1}{2} \int_k^{k+1} ([\boldsymbol{\sigma} \times] \mathbf{a}^b - [\dot{\boldsymbol{\sigma}} \times] \mathbf{u}_k^b) dt). \quad (4.27)$$

In this study, it is assumed that \mathbf{a}^b and $\boldsymbol{\sigma}$ remain constant for the update interval. Therefore, as shown in [52], the integral term in Equation 4.27 becomes identically zero.

After solving the integral terms of Equation 4.27, the velocity vector of the body frame as observed in the navigation frame at time t_{k+1} can be written as

$$\mathbf{v}_{k+1}^n = \mathbf{v}_k^n + \mathbf{u}_k^n + \mathbf{g}_l^n(t_{k+1} - t_k). \quad (4.28)$$

In general the contribution of the Coriolis terms in Equation 4.19 is small compared to the other terms in the equation [52]. Therefore, it is sufficient to include the Coriolis corrections in Equation 4.19 at a relatively lower rate than the rotation matrix computation rate. Moreover, in this study, $\boldsymbol{\omega}_{en}^n$ is considered negligible due to the low velocity of the athlete and very short duration (aero phase of the jump only) of the trajectory determination.

Finally, the position can be found by integrating the velocity. By applying the fourth order Runge-Kutta integration as in [52], the position at time t_{k+1} can be derived as

$$\mathbf{r}_{k+1}^n = \mathbf{r}_k^n + \frac{\mathbf{v}_{k-1}^n + 4\mathbf{v}_k^n + \mathbf{v}_{k+1}^n}{6}(t_{k+1} - t_k). \quad (4.29)$$

4.5 Rotation Angle Computation

In the proposed trajectory determination algorithm, as will be described in Section 4.7, rotation angles, i.e. roll, pitch and yaw, will be computed from accelerometer and magnetometer data. Rotation angles can also be derived from the rotation matrix, which is determined from the gyroscope data. In this section, the procedures for computing the rotation angles are demonstrated.

4.5.1 Roll and Pitch from Accelerometer

If the body frame is in steady motion, i.e. no specific acceleration on the body frame, then the triaxial accelerometer experiences antigravity only. In such a condition, the triaxial readings represent the vertically upward antigravity axis with respect to the body frame axes. Therefore, from the

4.5. Rotation Angle Computation

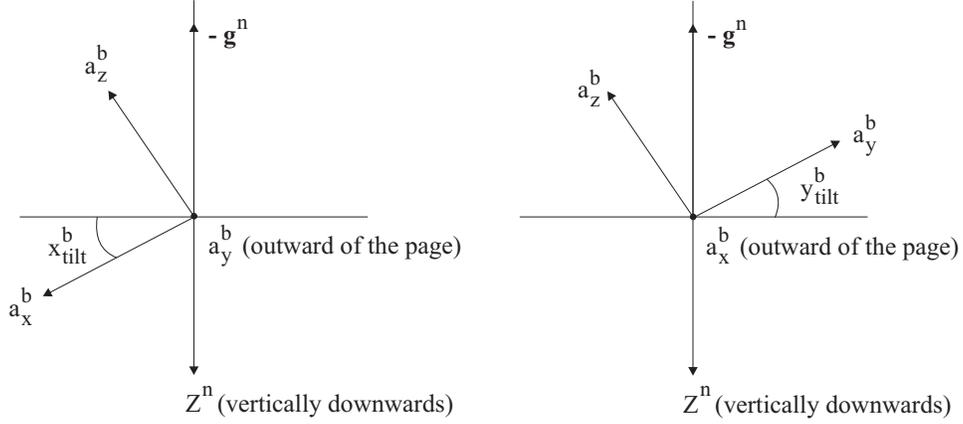


Figure 4.5: Roll and pitch calculation from accelerometer data.

normalized accelerometer readings, it is possible to find the opposite unit vector of the Z -axis of the navigation frame. On the other hand, the third row of the rotation matrix in Equation 4.4 represents the unit vector along the Z -axis of the navigation frame. Exploiting these facts, it is possible to find the roll and pitch angle from the accelerometer data as follows.

Figure 4.5 shows the body frame acceleration in the presence of anti-gravity only. From the first element of the third row of the rotation matrix in Equation 4.4 and the normalized X -axis body frame acceleration, pitch can be computed as follows:

$$\sin(x_{tilt}^b) = \frac{a_x^b}{|\mathbf{g}_l^n|} = -(-\sin(\theta)) \quad (4.30)$$

Therefore,

$$x_{tilt}^b = \sin^{-1}\left(\frac{a_x^b}{|\mathbf{g}_l^n|}\right) \quad (4.31)$$

and

$$\theta = x_{tilt}^b. \quad (4.32)$$

Here x_{tilt}^b is the angle between the horizontal plane and X^b . In [55], it has been proven that the sensitivity of X^b of the body frame decreases as it comes closer to the vertical axis. Therefore, to increase the sensitivity of the x_{tilt}^b calculation, the tangent function is used instead of the sine function in Equation 4.31 as below:

$$x_{tilt}^b = \tan^{-1}\left(\frac{a_x^b}{\sqrt{a_y^{b2} + a_z^{b2}}}\right). \quad (4.33)$$

4.5. Rotation Angle Computation

From the second element of the third row of the rotation matrix in Equation 4.4 and the normalized Y -axis body frame acceleration, roll can be computed as follows:

$$\sin(y_{tilt}^b) = \frac{a_y^b}{-|\mathbf{g}_l^n|} = \cos(\theta) \sin(\phi). \quad (4.34)$$

Therefore,

$$y_{tilt}^b = \sin^{-1}\left(\frac{a_y^b}{-|\mathbf{g}_l^n|}\right) \quad (4.35)$$

and

$$\phi = \sin^{-1}\left(\frac{\sin(y_{tilt}^b)}{\cos(\theta)}\right). \quad (4.36)$$

Here y_{tilt}^b is the angle between the horizontal plane and Y^b . For reasoning similar to that in [55], the tangent function is used instead of the sine function in Equation 4.35 as follows:

$$y_{tilt}^b = -\tan^{-1}\left(\frac{a_y^b}{\sqrt{a_x^{b2} + a_z^{b2}}}\right). \quad (4.37)$$

Sign Ambiguity in Roll and Pitch

From Equation 4.36 and Equation 4.32 it is not possible to find the quadrant in which the roll and pitch fall. Therefore, lookup tables are constructed where acceleration signs along orthogonal axes are used to extract the quadrant information of the rotation angles. Table 4.1 and Table 4.2 are lookup tables required to resolve the rotation angle ambiguity that occurs while computing roll and pitch from acceleration data.

Table 4.1: Quadrant information of roll from acceleration signs.

Roll quadrant	Sign of a_y^b	Sign of a_z^b
1	-	-
2	-	+
3	+	+
4	+	-

4.5. Rotation Angle Computation

Table 4.2: Quadrant information of pitch from acceleration signs.

Pitch quadrant	Sign of a_x^b	Sign of a_z^b
1	+	-
2	+	+
3	-	+
4	-	-

4.5.2 Yaw from Magnetometer

The triaxial magnetometer gives the magnetic component information vector of the earth's magnetic field in the body frame. This vector is represented as $\mathbf{m}^b = [m_x^b \ m_y^b \ m_z^b]^T$. From this vector, it is possible to deduce the yaw of the body frame with respect to the navigation frame. Even though the earth's magnetic north pole and geographic north pole do not entirely coincide, the error in yaw due to this phenomenon is negligible for the purpose of this study.

The magnetometer sensitivity decreases with an increase in x_{tilt}^b and y_{tilt}^b angles [56]. Therefore, it is important to realign the body frame Z -axis with the navigation frame Z -axis before computing yaw. For this purpose, rotation must be applied to first remove the roll angle followed by a second rotation that removes the pitch angle. The rotation matrix required is

$$\begin{aligned}
 \mathbf{R}_{\phi\theta} &= \begin{bmatrix} \cos(\theta) & 0 & -\sin(\theta) \\ 0 & 1 & 0 \\ \sin(\theta) & 0 & \cos(\theta) \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\phi) & \sin(\phi) \\ 0 & -\sin(\phi) & \cos(\phi) \end{bmatrix} \\
 &= \begin{bmatrix} \cos(\theta) & \sin(\phi)\sin(\theta) & -\cos(\phi)\cos(\theta) \\ 0 & \cos(\phi) & \sin(\phi) \\ \sin(\theta) & -\sin(\phi)\cos(\theta) & \cos(\phi)\cos(\theta) \end{bmatrix}. \quad (4.38)
 \end{aligned}$$

The order of the matrix multiplication is important, as matrix multiplication is not commutative.

From the magnetic field readings and $\mathbf{R}_{\phi\theta}$ the magnetic field components along the X and Y axes of the realigned body frame can be computed as

$$X_h = m_x^b \cos(\theta) + m_y^b \sin(\phi) \sin(\theta) - m_z^b \cos(\phi) \sin(\theta) \quad (4.39)$$

$$Y_h = m_y^b \cos(\phi) + m_z^b \sin(\phi). \quad (4.40)$$

Finally, the yaw angle can be found as

$$\psi = \tan^{-1}\left(\frac{Y_h}{X_h}\right). \quad (4.41)$$

4.5. Rotation Angle Computation

Similar to the roll and pitch angle, the quadrant information should be resolved from the signs of X_h and Y_h . The necessary lookup table is given in Table 4.3.

Table 4.3: Quadrant information of yaw from magnetic field component signs.

For	ϕ
$X_h < 0$	$\pi - \tan^{-1}\left(\frac{Y_h}{X_h}\right)$
$X_h > 0, Y_h < 0$	$-\tan^{-1}\left(\frac{Y_h}{X_h}\right)$
$X_h > 0, Y_h > 0$	$2\pi - \tan^{-1}\left(\frac{Y_h}{X_h}\right)$
$X_h = 0, Y_h < 0$	$\frac{\pi}{2}$
$X_h = 0, Y_h > 0$	$\frac{3}{2}\pi$

4.5.3 Rotation Angles from Rotation Matrix

As previously discussed, in an INS algorithm the rotation matrix is updated in each cycle from the angular rate readings of the triaxial gyroscope. For the proposed trajectory determination algorithm, which is described later in Section 4.7, rotation angles need to be computed directly from the rotation matrix in each update cycle. For a rotation matrix

$$\mathbf{R}_b^n = \begin{bmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{bmatrix} \quad (4.42)$$

the roll, pitch and yaw angles can be computed as

$$\phi = \tan^{-1}\left(\frac{R_{32}}{R_{33}}\right) \quad (4.43)$$

$$\theta = -\sin^{-1}(R_{31}) \quad (4.44)$$

$$\psi = \tan^{-1}\left(\frac{R_{21}}{R_{11}}\right). \quad (4.45)$$

$$(4.46)$$

As it will later be shown, the difference between the computed rotation angles from the rotation matrix and the computed rotation angles from the accelerometer and magnetometer serve as the observation vector for a particular Linear Kalman Filter (LKF). Therefore, to make both sets of

rotation angles coherent with each other, quadrant ambiguities are resolved for the set computed from the rotation matrix using Tables 4.1, 4.2 and 4.3 as well.

4.6 Sensor Errors

For the implementation of the INS algorithm, it is important to mitigate the error generated by the noise present in the accelerometer and gyroscope readings. Among the number of error sources, such as misalignment, scaling, bias, drift, temperature effects, etc., the accelerometer and gyroscope biases are the most significant and should be accounted for [57]. Accelerometer bias has a quadratic effect on the position derived from the INS algorithm. The gyroscope bias, which is even more common, has an equally large impact on the position estimate. Due to the bias in the gyroscope, the inertial measurement unit is under the impression that it is rotating and the INS equations will not account for gravity correctly. This can lead to a false movement calculation due to an acceleration of maximum 9.81 ms^{-2} depending on how much more the body frame is rotated due to the bias. Generally, for MEMS sensors the gyroscope bias is more prominent than that of the accelerometers and it also tends to drift significantly with time. Therefore, the gyroscope bias contributes to the majority of the errors in the INS algorithm position estimate.

According to the research conducted by Waegli et al., it was revealed that for given MEMS sensor characteristics the misalignment, scaling, drift and bias errors cannot be decorrelated efficiently for limited integration periods [38]. Therefore, in this research, a simplified error model is considered where the bias terms are only taken into account and modeled as 1st order Gauss-Markov processes as in [38].

The inertial accelerometer raw readings in discrete form can be modeled as

$$\tilde{\mathbf{a}}_k = \mathbf{a}_k + \mathbf{b}_k^a + \mathbf{w}_k^a \quad (4.47)$$

where $\tilde{\mathbf{a}}_k$ is the measured acceleration, \mathbf{a}_k is the true body frame acceleration, \mathbf{b}_k^a is the bias in acceleration and \mathbf{w}_k^a is the measurement noise at epoch t_k . Note that the superscript notion of the body frame is excluded for the ease of representation. According to the 1st order Gauss-Markov model, the bias can be represented with a discrete differential equation [58] as

$$\mathbf{b}_{k+1}^a = (\mathbf{1} - \beta^a \Delta t) \mathbf{b}_k^a + \sqrt{2\beta^a \sigma^{a2} \Delta t} \mathbf{w}_k^a \quad (4.48)$$

4.7. Proposed Algorithm

where σ^{a2} is the square of the amplitude of the acceleration bias power spectral density, β^a is the reciprocal of the acceleration process correlation time τ^a and Δt is the sampling interval. These parameters can be calculated from the widely practiced autocorrelation analysis, which is described in [58].

Similarly, the measured angular rate and the bias in angular rate are modeled as

$$\tilde{\omega}_k = \omega_k + \mathbf{b}_k^\omega + \mathbf{w}_k^\omega \quad (4.49)$$

$$\mathbf{b}_{k+1}^\omega = (\mathbf{1} - \beta^\omega \Delta t) \mathbf{b}_k^\omega + \sqrt{2\beta^\omega \sigma^{\omega 2} \Delta t} \mathbf{w}_k^\omega \quad (4.50)$$

where $\tilde{\omega}_k$ is the measured body frame angular rate with respect to the navigation frame, ω_k is the true body frame angular rate, \mathbf{b}_k^ω is the bias in angular rate and \mathbf{w}_k^ω is the measurement noise at epoch t_k . The bias model parameter $\sigma^{\omega 2}$ is the square of the amplitude of the angular rate bias power spectral density and β^ω is the reciprocal of the angular rate process correlation time τ^ω .

The errors in magnetometer readings only affect the computation of the yaw angle of the body frame with respect to the navigation frame. These errors are not related to the critical gravity cancelation process. Hence, magnetometer errors do not impose any large error in the position estimation for a short duration. Even if there is a considerable bias in the magnetometer reading, it can be safely ignored as the goal of this research is to find the relative jump trajectory rather than the absolute trajectory. Relative jump trajectory means the evolving position estimates of the athlete with respect to a reference point, e.g. jump start position, whereas the absolute trajectory represents the evolving position estimates in global positioning terms, e.g. latitude and longitude. Therefore, magnetometer error estimation is ignored in this work and measured magnetometer readings, i.e., $\tilde{\mathbf{m}}_k$ are directly used for yaw computation to reduce the computational load.

4.7 Proposed Algorithm

In this section, the proposed algorithm for jump trajectory determination is demonstrated. A loosely coupled GPS/INS integrated algorithm is proposed. A flow diagram of the overall proposed algorithm is given in Figure 4.8. Three Linear Kalman Filters (LKF) are used to correct the sensor data prior to computation of the position estimates with the INS algorithm. Later, an Extended Kalman Filter (EKF) is implemented to correct the trajectory from the INS with the position updates from the GPS. Even

4.7. Proposed Algorithm

though, only the trajectory during the jump is required, the algorithm is applied starting at an epoch prior to the jump start up to the epoch of the jump end. This is done in order to capture the dynamics undertaken as a preparation for the jump. The key and novel aspects of the proposed trajectory determination algorithm are as follows:

1. In general, the INS or GPS/INS integrated algorithm corrects the accelerometer and gyroscope readings with a bias estimate from the previous update interval, as in [59], prior to resolving the INS equations. This is because the sensor error estimates for any update interval are only available once the INS or GPS/INS navigation equations are solved. Computing in this manner might not address the bias drift correctly. Moreover, misalignment error and temperature effects are also not decorrelated from the bias error, which may cause the sensor error to drift rapidly. In the proposed algorithm, sensor data fusion is exploited to estimate the bias in the readings with the sensor data itself. LKF is used for the sensor fusion and to estimate the bias in one sensor with the help of another prior to resolving the INS equations. These estimated bias terms for the current update cycle are used to correct the sensor readings, which results in a better position estimate from the INS algorithm.
2. The sensor bias error estimate from the sensor signal itself also provides leverage in the GPS/INS integrated algorithm. In GPS/INS integration with augmented states, i.e. where the sensor errors are also incorporated in the state vector, typically there are no sensor error terms in the observation vector as in [60]. However, the bias estimates made prior to the INS algorithm may serve as the observations for sensor biases in the EKF for the GPS/INS integration. Therefore, two separate sensor error estimations are conducted in a particular update cycle. The estimated error from the EKF for an update cycle is also used in the next update cycle to make primary corrections in the accelerometer and gyroscope signal. In this way, the error estimates from the EKF contribute a better initial estimate of the sensor bias in the next update cycle. Therefore, the augmented observations for GPS/INS integration leads to more accurate overall sensor error estimation.
3. In previous work where acceleration is used to correct the inclination estimate, e.g. [59], the body frame is assumed to be entirely stationary and the relevant Kalman filter parameters also remain static. This is

never the case for an athlete performing a jump. High specific acceleration, i.e. non gravitational acceleration, is particularly involved before the take-off and after the landing. On the other hand, during the aero phase the body frame accelerometers feel no acceleration. Therefore, the captured sensor data during and neighboring jumps are segmented into different portions depending on the athletes dynamics. The LKF parameters are customized for each segment and varied over time to properly address the different acceleration scenarios.

4. Even though the error estimation processes for accelerometer and gyroscope through sensor fusion are not entirely independent from each other, certain assumptions are made to decouple the states of the accelerometer and gyroscope biases. This is done to reduce computational load as the state vectors for the accelerometer and gyroscope errors are updated separately using transition matrices with smaller dimensions (3×3) rather than using a single matrix with large dimension (6×6).

The proposed algorithm is explained chronologically below. The basic equations for the implementation of LKF and EKF are not explained in this work. A detailed demonstration of the LKF and EKF can be found in [61] and [52].

4.7.1 Acceleration Error Compensation

For an update cycle, an initial estimate of the body frame acceleration is done with a LKF, which is termed as LKF¹. The bias error estimated from the GPS/INS integration in the previous cycle is used to correct the accelerometer readings. This initial estimate is the observation vector for the LKF¹. As the acceleration of an athlete changes gradually with time, acceleration of the current cycle can be estimated from the previous cycle with a rotation matrix. The rotation matrix rotates the acceleration in the previous cycle to reflect into the body frame attitude of the current update cycle. To adjust to the changing dynamics, enough uncertainty is injected into the process through considerable process noise.

The state and measurement equations of the LKF¹ in discrete form are

$$\mathbf{a}_{k+1}^c = \mathbf{A}_k^T \hat{\mathbf{a}}_k + \mathbf{q}_k^a \tag{4.51}$$

$$\mathbf{z}\mathbf{a}_{k+1}^c = \mathbf{a}_{k+1}^c + \mathbf{w}_{k+1}^a \tag{4.52}$$

where

4.7. Proposed Algorithm

- \mathbf{a}_{k+1}^c = initial corrected acceleration for the current update cycle
- \mathbf{A}_k^T = state transition matrix, transpose of \mathbf{A}_k
- $\hat{\mathbf{a}}_k$ = final estimate of the acceleration for the previous update cycle starting at epoch t_k
- \mathbf{q}_k^a = process noise of acceleration with covariance matrix \mathbf{Q}_k^a
- $\mathbf{z}\mathbf{a}_{k+1}^c$ = observation vector for LKF¹
- \mathbf{w}_{k+1}^a = measurement noise of acceleration with covariance matrix \mathbf{R}_{k+1}^a .

Note that, the transition matrix \mathbf{A}_k^T is the transpose of the rotation matrix derived in Equation 4.10. This is because the acceleration is reflected forward in time in the state equation, whereas \mathbf{A}_k reflects the vector backwards in time.

The measurement vector $\mathbf{z}\mathbf{a}_{k+1}^c$ for the current cycle is computed as

$$\mathbf{z}\mathbf{a}_{k+1}^c = \tilde{\mathbf{a}}_{k+1} - \hat{\mathbf{b}}_k^{aE}. \quad (4.53)$$

Here, $\tilde{\mathbf{a}}_{k+1}$ is the accelerometer reading for the current update cycle starting at t_{k+1} and $\hat{\mathbf{b}}_k^{aE}$ is the acceleration error estimate from the EKF for GPS/INS integration in the previous cycle starting at t_k .

With the initial acceleration estimate $\hat{\mathbf{a}}_{k+1}^c$ from the LKF¹, the algorithm proceeds to estimate the acceleration bias for the current cycle derived via sensor fusion. Another LKF, namely LKF², is applied for this purpose. According to the 1st order Gauss-Markov model in Equation 4.48, the state and measurement equations of LKF² are represented as

$$\mathbf{b}_{k+1}^{aL} = (\mathbf{1} - \beta^a \Delta t) \mathbf{b}_k^{aL} + \sqrt{2\beta^a \sigma^{a2}} \Delta t \mathbf{w}_k^{aL} \quad (4.54)$$

$$\mathbf{z}\mathbf{b}_{k+1}^{aL} = \mathbf{b}_{k+1}^{aL} + \mathbf{w}_{k+1}^{aL} \quad (4.55)$$

where

- \mathbf{b}_{k+1}^{aL} = bias error in the acceleration derived via sensor fusion for the current update cycle
- $\mathbf{z}\mathbf{b}_{k+1}^{aL}$ = observation vector for LKF².
- \mathbf{w}_{k+1}^{aL} = measurement noise of acceleration bias with covariance matrix \mathbf{R}_{k+1}^{aL} .

The observation vector is found from the initial acceleration estimate and the sensor readings as follows:

$$\mathbf{z}\mathbf{b}_{k+1}^{aL} = \tilde{\mathbf{a}}_{k+1} - \hat{\mathbf{a}}_{k+1}^c. \quad (4.56)$$

The acceleration bias estimated from LKF², $\hat{\mathbf{b}}_{k+1}^{aL}$, is claimed to be derived via sensor fusion as the gyroscope readings are used in LKF¹ to calculate

$\hat{\mathbf{a}}_{k+1}^c$. Moreover, the bias estimate from the EKF in the previous cycle indirectly plays a role in $\hat{\mathbf{b}}_{k+1}^{aL}$ calculation as $\hat{\mathbf{b}}_k^{aE}$ is used to derive $\hat{\mathbf{a}}_{k+1}^c$. Here it is assumed that the gyroscope errors are decoupled from the acceleration errors. The rationale behind this assumption is that the gyroscope readings are recursively corrected with a separate LKF, as will be later discussed, for each update cycle. Therefore, the estimated error in the accelerometer should become independent from the gyroscope error with time in an iterative manner.

Finally, with the estimation of the acceleration error from LKF², $\hat{\mathbf{b}}_{k+1}^{aL}$, the accelerometer readings are corrected for the current update cycle as

$$\hat{\mathbf{a}}_{k+1} = \tilde{\mathbf{a}}_{k+1} - \hat{\mathbf{b}}_{k+1}^{aL}. \quad (4.57)$$

This final acceleration estimate $\hat{\mathbf{a}}_{k+1}$ is used for the rest of the algorithm including in the gyroscope error estimation and in the INS equations for the current update cycle.

4.7.2 Angular Rate Error Compensation

The gyroscope readings provide the angular rate of the body frame with respect to the inertial frame. The Coriolis effects are accounted for to find the angular rate of the body frame relative to the navigation frame following the equation

$$\tilde{\boldsymbol{\omega}}_{k+1} = \tilde{\boldsymbol{\omega}}_{ibk+1}^b - (\mathbf{I} + [\hat{\boldsymbol{\zeta}}_k \times])^T \hat{\mathbf{R}}_k^T \boldsymbol{\omega}_{ie}^n \quad (4.58)$$

where

- $\tilde{\boldsymbol{\omega}}_{ibk+1}^b$ = gyroscope reading for the update cycle starting at t_{k+1}
- $[\hat{\boldsymbol{\zeta}}_k \times]$ = skew symmetric matrix of the estimated attitude error vector $\hat{\boldsymbol{\zeta}}_k$ in INS mechanization for previous update cycle starting at t_k
- $\hat{\mathbf{R}}_k$ = estimated rotation matrix for previous update cycle starting at t_k
- $\boldsymbol{\omega}_{ie}^n$ = earth's angular velocity with respect to the inertial frame and as observed from the navigation frame.

It should be noted that the angular rate of the navigation frame relative to the earth is ignored. This is because the athlete's velocity relative to the earth is insignificant enough to have no impact on the rotation matrix. The attitude error vector is computed from the state estimation with EKF, which is discussed in Subsection 4.7.4.

4.7. Proposed Algorithm

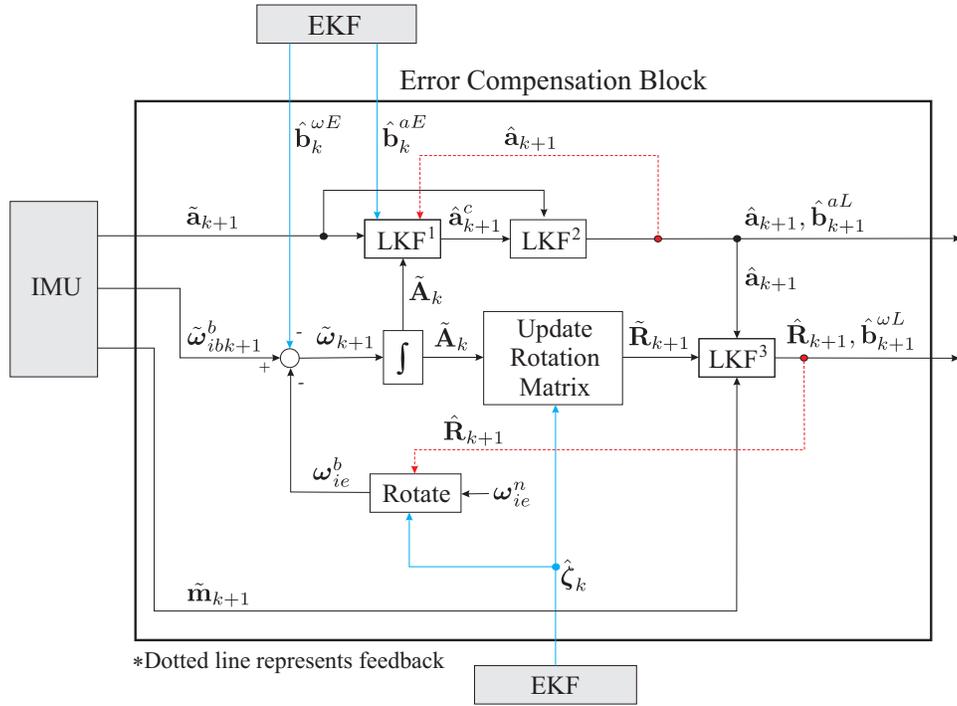


Figure 4.6: Error compensation block.

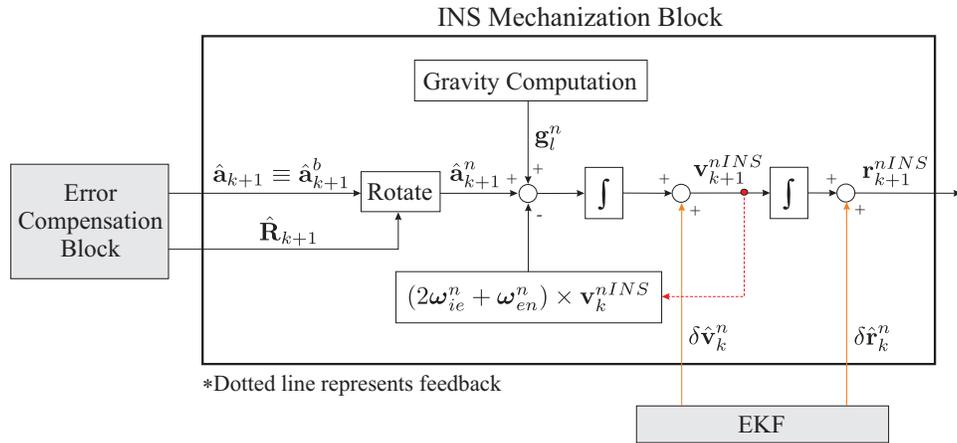


Figure 4.7: INS mechanization block.

4.7. Proposed Algorithm

The rotation matrix for the current cycle is computed initially using the gyroscope error estimation from the previous cycle

$$\tilde{\mathbf{R}}_{k+1} = \hat{\mathbf{R}}_k(\mathbf{I} + [\hat{\boldsymbol{\zeta}}_k \times])\tilde{\mathbf{A}}_k \quad (4.59)$$

where $\tilde{\mathbf{A}}_k$ is computed following Equation 4.10 using the angular rate vector $(\tilde{\boldsymbol{\omega}}_{k+1} - \hat{\mathbf{b}}_k^{\omega E})$. The estimated error in the gyroscope computed via EKF in the previous cycle is $\hat{\mathbf{b}}_k^{\omega E}$. It is important to note that the transpose of the matrix $\tilde{\mathbf{A}}_k$ is actually used as the transition matrix in Equation 4.51, i.e. \mathbf{A}_k^T .

From $\tilde{\mathbf{R}}_{k+1}$, roll, pitch and yaw angles are computed using the method described in Subsection 4.5.3. The computed set of angles are represented as

$$\tilde{\boldsymbol{\gamma}}_{k+1} = [\tilde{\phi}_{k+1} \quad \tilde{\theta}_{k+1} \quad \tilde{\psi}_{k+1}]^T \quad (4.60)$$

Another separate set of rotation angles can be derived from the accelerometer and magnetometer data. The method described in Subsection 4.5.1 is used to calculate the roll and pitch angles from the acceleration data. The normalized estimated acceleration for the current update cycle, $\hat{\mathbf{a}}_{k+1}/|\hat{\mathbf{a}}_{k+1}|$, is used as the acceleration signal for this purpose. The acceleration is normalized to ensure that the amplitude of the acceleration vector is 1 g. Subsection 4.5.2 is used with the available magnetometer data to find the yaw angle. The set of rotation angles derived from the accelerometer and magnetometer is represented as

$$\mathbf{z}\boldsymbol{\gamma}_{k+1} = [z\phi_{k+1} \quad z\theta_{k+1} \quad z\psi_{k+1}]^T. \quad (4.61)$$

The difference between the two separate estimates of the rotation angle sets is used to derive the observation vector of the LKF, namely LKF³, for the gyroscope error estimation. The state and measurement equations for LKF³ are

$$\mathbf{b}_{k+1}^{\omega L} = (\mathbf{1} - \beta^\omega \Delta t)\mathbf{b}_k^{\omega L} + \sqrt{2\beta^\omega \sigma^{\omega 2} \Delta t} \mathbf{w}_k^{\omega L} \quad (4.62)$$

$$\mathbf{z}\mathbf{b}_{k+1}^{\omega L} = \mathbf{b}_{k+1}^{\omega L} + \mathbf{w}_{k+1}^{\omega L} \quad (4.63)$$

where

$\mathbf{b}_{k+1}^{\omega L}$ = bias error in the gyroscope derived via sensor fusion for the current update cycle

$\mathbf{z}\mathbf{b}_{k+1}^{\omega L}$ = observation vector for LKF³.

$\mathbf{w}_{k+1}^{\omega L}$ = measurement noise in angular rate with covariance matrix $\mathbf{R}_{k+1}^{\omega L}$.

4.7. Proposed Algorithm

The observation vector of LKF³ is computed as

$$\mathbf{z}\mathbf{b}_{k+1}^{\omega L} = \frac{\tilde{\gamma}_{k+1} - \mathbf{z}\gamma_{k+1}}{\Delta t}. \quad (4.64)$$

The estimated error in angular rate using sensor fusion, $\hat{\mathbf{b}}_{k+1}^{\omega L}$, is found as the result of LKF³. This error estimate is used to correct the rotation angle derived from the rotation matrix. Therefore, the estimated rotation angle set is

$$\hat{\gamma}_{k+1} = \tilde{\gamma}_{k+1} - \hat{\mathbf{b}}_{k+1}^{\omega L} \Delta t. \quad (4.65)$$

Henceforth, the estimated rotation matrix for the current update cycle $\hat{\mathbf{R}}_{k+1}$ is computed from the set of corrected rotation angles $\hat{\gamma}_{k+1}$ using Equation 4.4.

Figure 4.6 shows the flow diagram of the proposed error compensation technique for a particular update cycle starting at epoch t_{k+1} . The Inertial Measurement Unit (IMU) represents the set of body frame sensors, which provides raw data to the algorithm. Among the four, two of the final outputs after the error compensation, the estimated acceleration $\hat{\mathbf{a}}_{k+1}$ and the estimated rotation matrix $\hat{\mathbf{R}}_{k+1}$, are fed to the INS mechanization equations. The other two final outputs, estimated acceleration error and estimated angular rate error from sensor fusion, namely $\hat{\mathbf{b}}_{k+1}^{aL}$ and $\hat{\mathbf{b}}_{k+1}^{\omega L}$, are fed to the EKF for the GPS/INS integration as a part of the observation vector.

4.7.3 INS Mechanization

The estimated body frame acceleration $\hat{\mathbf{a}}_{k+1}$ and rotation matrix $\hat{\mathbf{R}}_{k+1}$ are used to solve the INS equations derived in Section 4.4. The initial velocity of the athlete required for the INS equations can be directly achieved from the GPS receiver data. Finally, the position update at epoch t_{k+1} in relation to the navigation frame is derived from the INS equations, which are termed as the vector $\mathbf{r}_{k+1}^{nINS} = [\gamma_{xk+1}^{nINS} \quad \gamma_{yk+1}^{nINS} \quad \gamma_{zk+1}^{nINS}]^T$. Here, the subscripts x, y and z denote the directions toward north, east and vertically downwards respectively. The pertinent velocity vector is represented as $\mathbf{v}_{k+1}^{nINS} = [v_{xk+1}^{nINS} \quad v_{yk+1}^{nINS} \quad v_{zk+1}^{nINS}]^T$.

Figure 4.7 depicts the flow diagram of INS mechanization steps. The final output, i.e. the position vector \mathbf{r}_{k+1}^{nINS} , is fed forward to the GPS/INS integration.

4.7.4 Integration with GPS

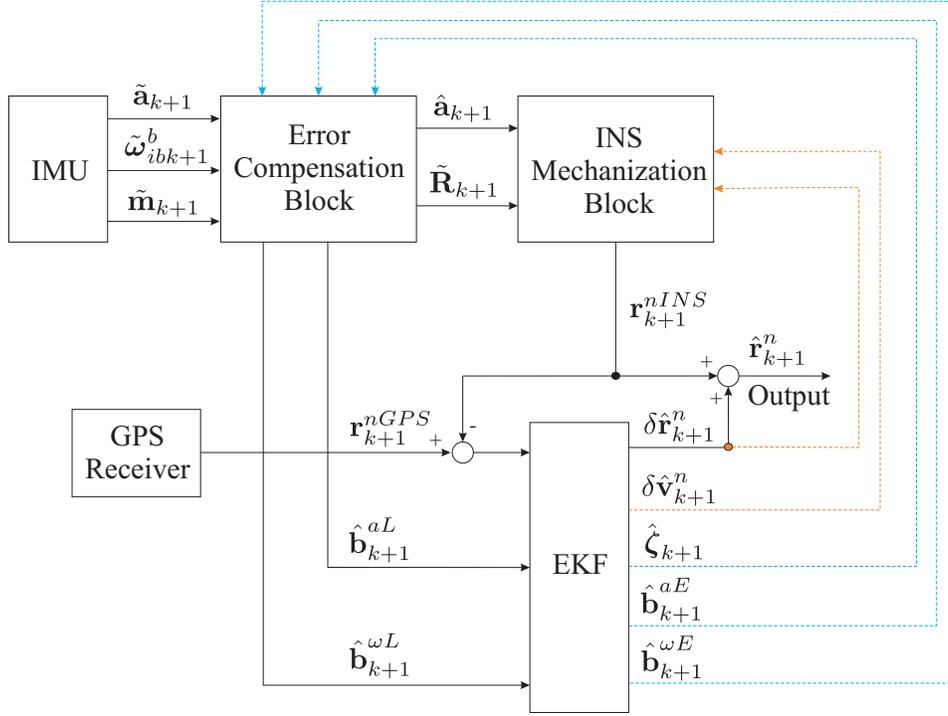
To estimate the optimum position updates, the output of the INS needs to be integrated with the position updates from the GPS with a suitable estimator. The Extended Kalman Filter (EKF) is one of the most popular estimators for such integrations [62]. The EKF is an adaptation of the LKF to nonlinear functions, such as the INS perturbation model provided later in Equation 4.66-4.68, that is approximated by linearization. A detailed description of the EKF can be found in [52, 61].

The type of integration, where GPS position and/or velocity updates are directly used to correct the INS output, is called a loosely coupled system. In this study, a loosely coupled system is implemented for integrating GPS position updates. The integration strategy can also be either open looped or close looped. In an open looped system, the INS mechanization operates without being concerned about the presence of the estimator or external data. The estimated state vectors are used to correct the INS output only. However, in an open loop system, the error in INS mechanization with low cost MEMS sensors can grow unbounded and propagate significantly large error within a short period of time. As a result, unbounded error observations are delivered to the estimator, such as the Kalman filter. This causes problems in the linear filter since only small errors are allowed due to the linearization process [63]. Therefore, a closed loop system is required to continuously compensate for the error in INS mechanization. In a closed loop system, error estimates from the estimator are fed back to correct the sensor outputs and other mechanization parameters.

In a closed loop loosely coupled system, the GPS/INS integration is typically done by estimating the error states in the INS mechanization. The basic estimator consists of nine navigation error states consisting of three position, three velocity and three attitude error states [41]. However, due to the error in sensors, the state vector is augmented by incorporating acceleration and angular rate error states resulting in a fifteen element state vector.

The INS error state models are obtained by the perturbation of the mechanization equations [41]. The perturbation analysis is not within the scope of this research and, therefore, is not demonstrated here. Several well documented perturbation analyses are available as in [64] and [65]. As given in [66], the obtained model is expressed as a series of differential

4.7. Proposed Algorithm



*Dotted line represents feedback

Figure 4.8: Flow diagram of jump trajectory determination.

equations as follows:

$$\delta \dot{\mathbf{r}}^n = \delta \mathbf{v}^n \quad (4.66)$$

$$\delta \dot{\mathbf{v}}^n = -2[\boldsymbol{\omega}_{ie}^n \times] \delta \mathbf{v}^n - \mathbf{a}^n \boldsymbol{\zeta}^n + \mathbf{R}_b^n \mathbf{b}^{aE} \quad (4.67)$$

$$\dot{\boldsymbol{\zeta}}^n = -[\boldsymbol{\omega}_{ie}^n \times] \boldsymbol{\zeta}^n + \mathbf{R}_b^n \mathbf{b}^{\omega E} \quad (4.68)$$

where the “dot”s denote time derivatives and

- \mathbf{r}^n = position error state vector
- \mathbf{v}^n = velocity error state vector
- $\boldsymbol{\zeta}^n$ = attitude error state vector
- \mathbf{b}^{aE} = accelerometer error state
- $\mathbf{b}^{\omega E}$ = gyroscope error state.

Following the above continuous perturbation model and the simplified discrete time approximated model demonstrated in [63], the augmented dis-

4.7. Proposed Algorithm

crete time fifteen state model can be represented as

$$\begin{aligned}
 \delta \mathbf{x}_{k+1} &= \begin{bmatrix} \delta \mathbf{r}_{k+1}^n \\ \delta \mathbf{v}_{k+1}^n \\ \zeta_{k+1}^n \\ \mathbf{b}_{k+1}^{aE} \\ \mathbf{b}_{k+1}^{\omega E} \end{bmatrix} \\
 &= \begin{bmatrix} \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & -2[\boldsymbol{\omega}_{ie}^n \times] & -\mathbf{a}^n & \mathbf{R}_{bk+1}^n & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & -[\boldsymbol{\omega}_{ie}^n \times] & \mathbf{0} & \mathbf{R}_{bk+1}^n \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & (1/\Delta t - \beta^a) & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & (1/\Delta t - \beta^\omega) \end{bmatrix} \begin{bmatrix} \delta \dot{\mathbf{r}}_k^n \\ \delta \dot{\mathbf{v}}_k^n \\ \dot{\zeta}_k^n \\ \mathbf{b}_k^{aE} \\ \mathbf{b}_k^{\omega E} \end{bmatrix} \Delta t + \mathbf{q}_k^E
 \end{aligned} \tag{4.69}$$

where \mathbf{q}_k^E is the driven response at t_{k+1} due to the presence of the process white noise during the time interval $t_k - t_{k+1}$ [63]. The corresponding covariance matrix is set as [62]

$$\mathbf{Q}_k^E = \begin{bmatrix} \mathbf{Q}_k^r & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{Q}_k^v & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{Q}_k^\zeta & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{Q}_k^{aE} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{Q}_k^{\omega E} \end{bmatrix} \tag{4.70}$$

where

- \mathbf{Q}_k^r = covariance matrix for positioning error
- \mathbf{Q}_k^v = covariance matrix for velocity error
- \mathbf{Q}_k^ζ = covariance matrix for attitude error
- \mathbf{Q}_k^{aE} = covariance matrix for acceleration error
- $\mathbf{Q}_k^{\omega E}$ = covariance matrix for angular rate error.

Here, the bias errors are modeled as a 1st order Gauss-Markov process as given in Equation 4.48 and 4.50.

The observation vector for the EKF is derived from the difference between GPS and INS position estimates, and augmented by the two estimated errors from the error compensation block. Therefore, the nine element observation vector is

$$\delta \mathbf{z}_{k+1} = \begin{bmatrix} \mathbf{r}_{k+1}^{nGPS} - \mathbf{r}_{k+1}^{nINS} \\ \hat{\mathbf{b}}_{k+1}^{aL} \\ \hat{\mathbf{b}}_{k+1}^{\omega L} \end{bmatrix} \tag{4.71}$$

where \mathbf{r}_{k+1}^{nGPS} is the position update from GPS at epoch t_{k+1} . As GPS position updates occur at lower frequency than the frequency of the INS position updates, the GPS positions are interpolated to the INS update rate within the time period selected for the jump trajectory determination.

Finally, the estimated states are fed back to the error compensation block and the INS mechanization block for error correction. The estimated position error is used to correct the position output from the INS. Therefore, the final estimated position update is given as

$$\hat{\mathbf{r}}_{k+1}^n = \mathbf{r}_{k+1}^{nINS} + \delta\hat{\mathbf{r}}_{k+1}^n. \quad (4.72)$$

The flow diagram of the entire proposed algorithm is shown in Figure 4.8. In the figure, dotted lines represent feedback. The proposed technique is depicted for a particular update cycle and the index of the update cycle is explicitly mentioned as a subscript in the variable names.

4.7.5 Period Segmentation and Parameter Customization

For jump trajectory determination, it is important to capture all the abrupt changes in dynamics undertaken by the athlete in preparation for the jump. Failure to accommodate any small portion of significant change in acceleration prior to the jump may cause a large error in KPV determination, i.e. jump horizontal distance, height and drop. Therefore, the trajectory determination algorithm should be conducted on the data from a time epoch preceding jump start. However, the computation capability of the onboard processor is limited as well as the battery power. Moreover, computation should be completed shortly after the jump to provide real-time feedback on the KPVs to the goggle user. Therefore, due to resource constraints, it is also not feasible to operate the trajectory determination algorithm starting from far behind the jump start epoch to safely accommodate all of the jump preparation dynamics.

Another significant difficulty in jump trajectory determination using the Kalman filter is that the rate of change in dynamics varies abruptly within a very short period of time immediately before and after a jump. The jump's dynamic characteristics also vary a great deal for a wide range of different types of jumps. Therefore, the Kalman filter variables involved in predicting an athlete's dynamics should be customized and fine-tuned to address highly fluctuating scenarios.

To address the above difficulties, an automated algorithm is implemented to select the total trajectory determination period and segment it depending on the acceleration characteristics. As shown in Figure 4.9, the jump

4.7. Proposed Algorithm

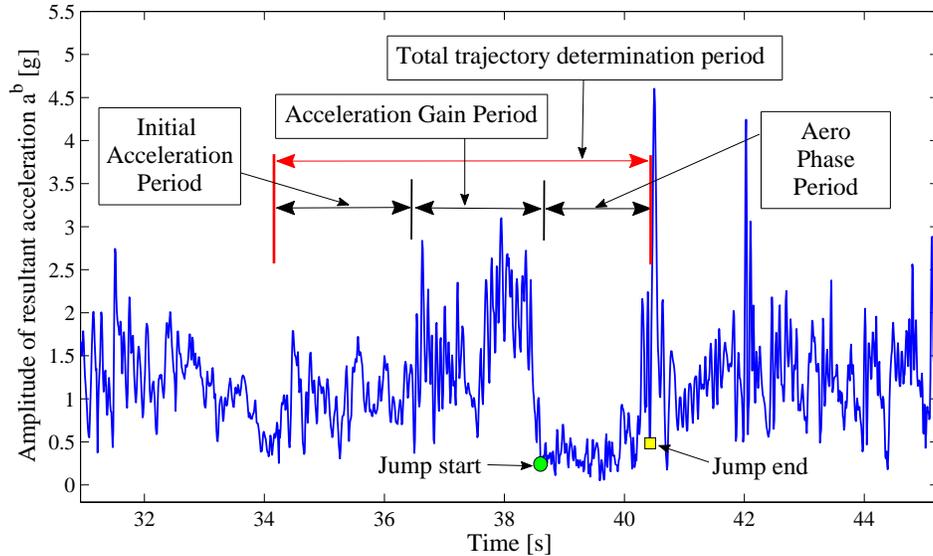


Figure 4.9: Period segmentation for trajectory determination.

preparation period, including the jump, has been segmented into three portions depending on the resultant acceleration profile, namely Initial Acceleration Period (IAP), Acceleration Gain Period (AGP) and Aero Phase Period (APP). In IAP, the resultant acceleration is steady and there is no abrupt change in the acceleration. In this period, the athlete has only started to prepare for the jump and the average acceleration slowly increases. This period also provides enough time for the Kalman filters to converge. In AGP, the majority of the gain in acceleration (mostly in upward and forward direction) occurs. Rapid fluctuations in the acceleration also take place during the preparation for take-off. This segment is of the most importance for the KPV calculation. The last segment, i.e. APP, is the time period when the athlete is in the air. During APP, the acceleration falls to near zero. Any algorithm parameter that uses gravity sensed by the accelerometer should be carefully handled during AGP and APP. This is because during AGP the accelerometer senses specific acceleration in magnitude much higher than gravity, whereas during APP the accelerometer senses no gravity at all. The jump trajectory determination algorithm is operated from the start of IAP to the end of APP.

The IAP and AGP are automatically selected by the algorithm from the resultant acceleration. APP is predetermined from the jump start epoch

4.7. Proposed Algorithm

to jump end epoch, which are determined by the algorithm described in Chapter 3. A threshold th^{AGP} is selected to extract the information about the gain in acceleration. If the resultant acceleration exceeds th^{AGP} at any epoch within a considerable period, which is set to 10 s in this study, prior to the jump start, then an indication of the start of AGP is achieved. Another 2 s of safety period is annexed before this epoch to make sure that the rise of acceleration is incorporated into AGP. Therefore, the start of AGP is calculated as,

$$t_s^{AGP} = \min_t \left\{ |\mathbf{a}^b(t)| > th^{AGP} \right\} - 2 \text{ s}, \quad t_s^{jump} - 10 \text{ s} < t < t_s^{jump} \quad (4.73)$$

where

$$\begin{aligned} t_s^{AGP} &= \text{start epoch of AGP} \\ \mathbf{a}^b(t) &= \text{resultant acceleration} \\ t_s^{jump} &= \text{jump start epoch.} \end{aligned}$$

Hence the AGP spans from t_s^{AGP} to t_s^{jump} . Furthermore, the IAP is empirically selected to span from $t_s^{AGP} - 3 \text{ s}$ to t_s^{AGP} and, as mentioned earlier, APP spans from t_s^{jump} to t_e^{jump} . Here, t_e^{jump} denotes the jump end epoch.

Two parameters of the LKFs implemented in the error compensation block are directly dependent on the dynamics of the athlete and, therefore, need to be varied over different segments. One of the parameters is the process noise covariance matrix, \mathbf{Q}_k^a , of LKF¹. Due to the very abrupt nature of the acceleration during AGP, \mathbf{Q}_k^a is set to a higher value during AGP relative to what it is set to for IAP and APP.

The second parameter, and also the more important one, which needs to be tuned during different segments is the observation noise covariance matrix, $\mathbf{R}_{k+1}^{\omega L}$, in LKF³. As previously noted, maintaining alignment, i.e. determination of the roll and pitch of the body frame, is important because of its direct relation to gravity cancelation. Therefore, acceleration measurements are used to correct the estimate of the body frame attitude found from the gyroscope. For accurate estimation of body frame attitude, the accelerometer should only sense vertically directed acceleration, e.g. anti gravity. However, during AGP, high specific acceleration is involved, which renders the acceleration sensed by the accelerometer in any random direction. On the other hand, during APP there is practically no acceleration to calculate the attitude from. Therefore, the reliability of the rotation angles calculated from the accelerometer data varies amongst the segmented periods. As the measurement vector of LKF³ directly depends on the measured rotation angles from the accelerometer, the measurement noise varies

significantly during the trajectory determination period. Therefore, the measurement noise covariance vector $\mathbf{R}_{k+1}^{\omega L}$ is considered a tuning parameter for proper estimation of the angular rate error with LKF³. The covariance matrix is tuned to a high value during AGP relative to what it is tuned to for IAP. This is done to decrease the reliability of the measurement vector during AGP. The reliability of the observation for LKF³ becomes even worse during APP due to an absence of any acceleration and, therefore, $\mathbf{R}_{k+1}^{\omega L}$ is set to a higher value to completely rely on the data achieved from the gyroscopes.

4.8 Field Experiments

The version of the Recon goggles, which are equipped with all the sensors needed for trajectory determination, was not yet ready for field data collection when this research was conducted. Therefore, jump data was collected with another data capturing device with similar sensors, named ‘Minimax’, developed by Catapult [31]. As ski resorts were closed during the time of this research, data was collected for jumps performed on a mountain bike as this closely resembles ski jumps. For benchmark data, a Novatel DL-V3 GPS receiver was carried by the mountain biker while executing the jumps. The Novatel receiver position updates were further refined with ‘CSRS - Precise Point Positioning’ provided by Natural Resources Canada [67]. The Novatel GPS receiver has a 5 Hz update rate while the Minimax GPS receiver gives position updates at 10 Hz. All the inertial sensors in the Catapult unit have an update rate of 100 Hz.

The Catapult unit affixed to the side of the biker’s helmet, as shown in Figure 4.10, while executing jumps to mimic the similar dynamic environment felt by the Recon goggle sensors. The Novatel receiver was carried in a backpack with its antenna projected outside. Synchronization of the data achieved from these two devices is a critical issue. Even though the GPS time stamps of the data captured with the Novatel receiver are reliable, the Catapult GPS time stamps suffer from a bias, which drifts with time. This is due to a varying time delay that occurs when the Catapult microprocessor writes the time stamps. This is a common issue for microsystems such as in Recon goggles, Minimax etc.

Therefore, data sets from the different devices are synchronized by calculating the minimum Root Mean Square (RMS) value of the difference in positional data readings (latitude and longitude) from different devices. However, this synchronization strategy renders an unavoidable implication.



Figure 4.10: Catapult Minimax unit affixed to the biker’s helmet.

Due to the spatial distance between the head mounted Catapult unit and the Novatel GPS antenna in the backpack, the derived jump start and end epochs from the acceleration captured by the Catapult unit’s accelerometer will not represent the same jump start and end epochs for the Novatel receiver. Therefore, to eliminate the spatial difference, the Catapult is strapped to the antenna pole of the Novatel receiver. The entire setup and a test jump performed by the athlete is shown in Figure 4.11 and Figure 4.12.

Carrying the Catapult unit in the backpack also poses some negative implications. First of all, the collected acceleration sensor data is more noisy than the data collected with the Catapult affixed to the helmet. This is because the antenna pole is not rigidly attached to the backpack and the backpack itself generates noise from movement due to the flexible coupling with the athlete. Secondly, as the catapult unit is not firmly coupled with the athlete, the detected jump from the Catapult acceleration represents the jump of the Catapult unit itself, not necessarily the jump of the athlete. Experimental data has shown that the athlete’s jump occurrence is delayed in time relative to the jump detected by the Catapult unit. The probable reason for this phenomenon is that the backpack settles on the athlete’s shoulder while he is still in the air. However, this should not cause any problem in the performance analysis of the trajectory determination algorithm as all the interest is focused on the trajectory determination, not the detection of jump start or end epochs. The jump start and end are detected with the accelerometer data as described in Chapter 3 and then the detected epochs are used to find the jump region, i.e. the aero phase

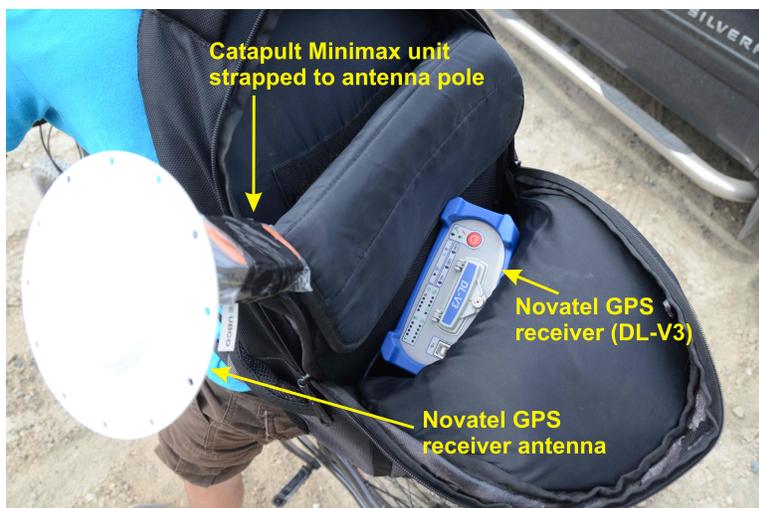


Figure 4.11: Catapult Minimax unit strapped to the Novatel GPS receiver antenna pole. [Photograph by Darren Handschuh]

region for the Catapult unit. The detected jump region is moved forward suitably to resemble the athlete's jump and relevant KPVs without any loss of generality. An example of noisy resultant acceleration of the Catapult unit strapped to the antenna pole is shown in Figure 4.13.

4.9 Experimental Results

Fourteen jumps were executed for which the Catapult MEMS inertial sensor and GPS receiver data, along with the Novatel receiver data, were collected. The Catapult data was fed to the proposed trajectory determination algorithm and the jump trajectory was determined. The trajectory determination begins from the start of the IAP and stops at the jump end. The trajectory was also determined from the GPS position updates of the Catapult unit. Afterwards, the relevant KPVs, namely jump horizontal distance, height and drop, were calculated easily. Similarly, KPVs are found from the Novatel receiver GPS position updates which are used as the benchmark KPVs. The Novatel receiver GPS receiver updates with precise point positioning have a horizontal positioning accuracy of ~ 2 cm and vertical positioning accuracy of ~ 7 cm. Furthermore, the Novatel GPS position updates are interpolated to the inertial sensor data rate for KPV determination, which is another potential error source for the benchmark data. Hence,

4.9. *Experimental Results*



Figure 4.12: Test jump performed by the athlete. [Photograph by Darren Handschuh]

4.9. Experimental Results

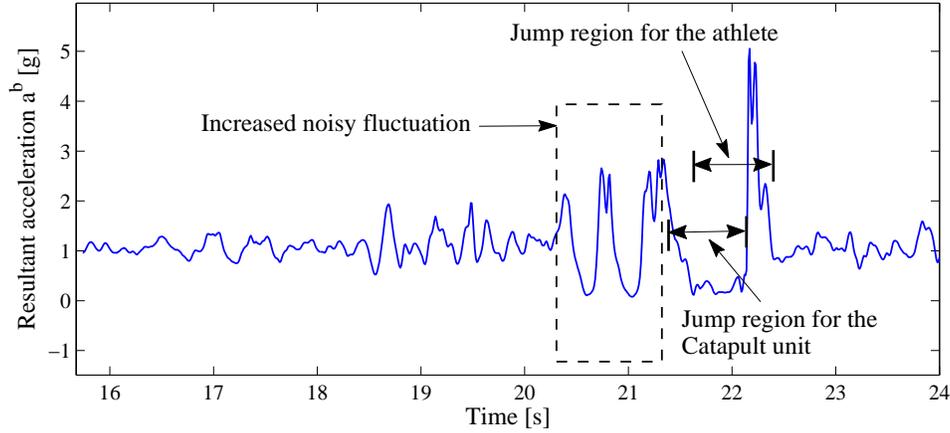


Figure 4.13: Resultant acceleration of Catapult unit strapped to the antenna pole carried in backpack.

the possibility of error in the benchmark KPV value should be considered when it is used for performance evaluation.

Figure 4.14 and Figure 4.15 show two examples of jump trajectory determination with the proposed algorithm, with the Catapult GPS solution and with the Novatel receiver GPS solution. The jump start and end epochs for the athlete were found as described in the previous section. It is noticeable in both figures that the trajectory from the Catapult GPS solution has very poor performance in vertical positioning updates. Not being entirely certain about the Catapult algorithm, it is speculated by looking at the data from the Catapult unit that the height profile is derived from both the altimeter and GPS solution. Therefore, the poor accuracy of the altimeter and the single point GPS receiver causes the unsatisfactory trajectory determination with the Catapult GPS solution. On the other hand, the trajectory determined by the proposed algorithm is very similar to the benchmark trajectory from the Novatel receiver GPS solution. This is due to the successful integration of the INS with the GPS updates. The fluctuation at the beginning of the trajectory for the proposed algorithm is due to the time required for filter convergence. However, there is a distance (mostly in vertical direction) between the benchmark trajectory and the trajectory derived from the proposed algorithm. This is because the proposed algorithm depends heavily on the INS for vertical position estimates and, as seen earlier, the up-

4.9. Experimental Results

dates for vertical positioning from the Catapult height profile are also very poor in accuracy. Therefore, the noise in low cost MEMS sensors causes a drift from the benchmark in the trajectory determined by the proposed algorithm, mostly in the vertical direction. However, this should not cause considerable error as the interest is focused on relative measurements for a brief period of time.

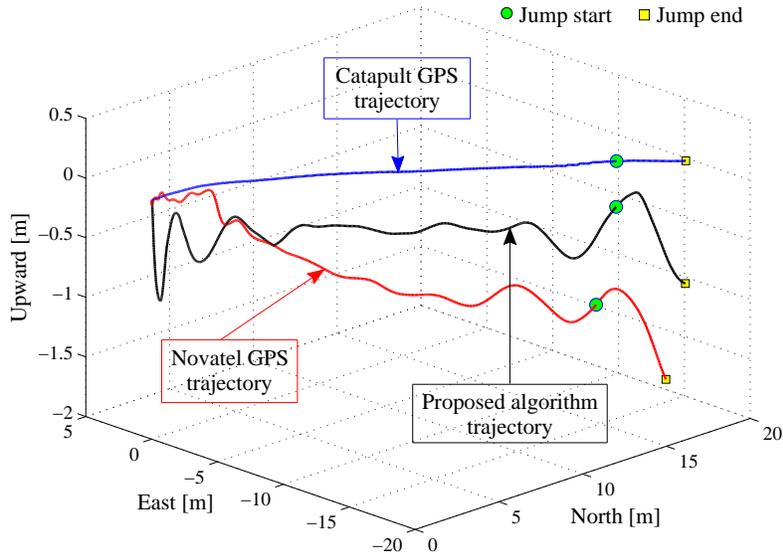


Figure 4.14: First example of jump trajectory determination with the proposed algorithm, with the Catapult GPS solution and with the Novatel receiver GPS solution.

Tables 4.4, 4.5 and 4.6 lists KPV values (horizontal distance, height, and drop respectively) for all the jumps along with the benchmark values deduced from Novatel trajectory. For the calculation of average errors, error in KPV for each jump was first determined and then the errors for all the jumps were averaged. Similarly, the percentage errors were also calculated. To show the advantage of the augmented error observation, provided by the error compensation block in the GPS/INS integration, two results have been shown for the proposed algorithm. One set of KPVs is calculated by the application of the proposed algorithm excluding the augmenting sensor error observations in the EKF, whereas, the second set of KPVs is calculated

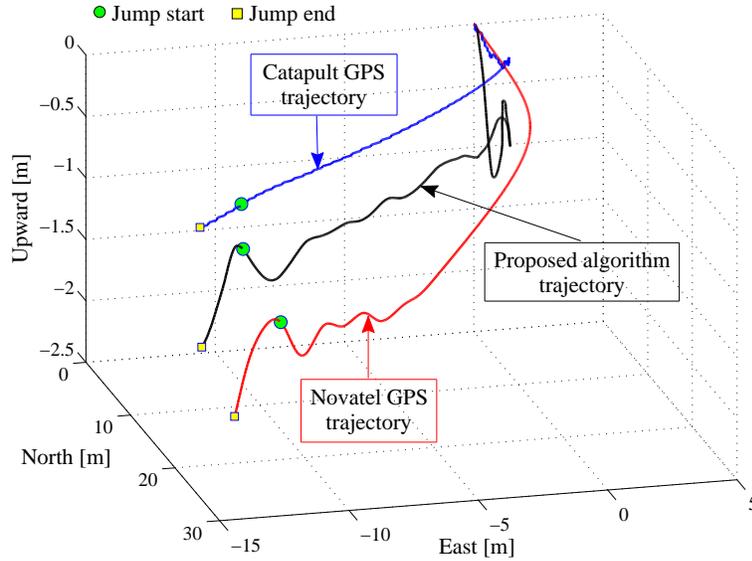


Figure 4.15: Second example of jump trajectory determination with the proposed algorithm, with the Catapult GPS solution and with the Novatel receiver GPS solution.

by the complete application of the proposed algorithm, i.e. including the augmenting sensor error observations.

As shown Table 4.4, the proposed algorithm provides minor improvements in the horizontal distance calculation. The proposed algorithm provides 1.99 % and 1.71 % improvement over the Catapult GPS solution. The percentage error improvements are calculated with respect to the Catapult unit errors in KPV determination. The reason behind this slight improvement in horizontal distance measurement is because horizontal distance measurement depends greatly on the initial velocity of the body frame. With the inertial sensors, it is not possible to derive the initial velocity at the start of the trajectory determination period. Hence, the velocity from the Catapult GPS receiver is used for the proposed algorithm. Therefore, the error in the Catapult GPS solution also propagates through the proposed algorithm for horizontal distance measurements. It is postulated that the Catapult GPS solution provides velocity readings lower than the true velocity. Therefore, the horizontal distance measurements from the proposed

4.9. Experimental Results

algorithm are generally lower in magnitude than the benchmark. The error in horizontal distance by the Catapult GPS solution is due to an internal mechanism error of the unit, which is unknown.

On the other hand, the vertical velocity of the athlete is generally very small at the start of the trajectory determination period and, therefore, error in the Catapult velocity has little effect on the proposed algorithm. As a result, great improvements are reflected in the height and drop calculations for the proposed algorithm as shown in Table 4.5 and Table 4.6. In height, the complete application of the proposed algorithm provides 60.51 % improvement over the Catapult GPS solution, whereas the improvement is even greater (88.73 %) for drop determination. In Table 4.5, it is noticeable that there is no jump height for jumps no. 7, 10 and 12. This is due to the characteristic of the jump itself where the athlete achieved no elevation from the jump start point. For the application of the proposed algorithm without the sensor error observations in the EKF, the improvement for height and drop relative to the Catapult GPS solution are 55.70 % and 85.40 % accordingly. The smaller improvement in height and drop determination for the proposed algorithm with no sensor error observation provides the rationale behind augmenting the observation vector in the EKF for GPS/INS integration with the estimated sensor errors from the error compensation block. Of course, the addition of the augmented measurements will increase the computation load. However, this is the price to pay for the increased accuracy in KPV determination.

To show the effectiveness of the proposed novel error compensation scheme, GPS/INS integration was also done but excluding the entire error compensation block depicted in Figure 4.6 as opposed to just excluding the use of the augmented observation vector in the EKF, the results of which are shown in Tables 4.4, 4.5 and 4.6. However, the error estimates from the EKF are still fed back to correct the sensor readings and INS mechanization parameters. The bar plots shown in Figure 4.16 and Figure 4.17 summarize the error comparison in KPV determination produced by the catapult GPS solution, GPS/INS integration without the error compensation block, proposed algorithm without the augmented observation in EKF, and the complete proposed algorithm. Each block is colored differently to mark the error contribution of each KPV calculation to the total error of all the KPV determinations. The height of each block represents the total error in the jump horizontal distance, height and drop calculation for the applied method to the particular jump. It is evident from the bar plot that the error compensation block has a major impact on the accuracy of the INS mechanization as well as the GPS/INS integration.

4.9. Experimental Results

Table 4.4: Comparison of jump horizontal distance determination performance between the proposed algorithm and Catapult GPS solution.

Jump no.	Horizontal Distance [cm]			
	Novatel GPS	Catapult GPS	Proposed algo.(without augmented observation)	Proposed algo.(with augmented observation)
1	206.25	189.46	189.40	189.37
2	215.81	198.68	198.26	198.24
3	339.44	332.82	333.17	333.17
4	290.47	273.39	274.07	274.13
5	251.52	200.45	201.39	201.27
6	313.86	296.44	296.92	296.74
7	197.96	180.57	180.62	180.67
8	231.89	225.01	225.19	225.21
9	237.45	224.27	225.07	225.07
10	206.10	200.08	200.09	199.95
11	289.50	278.61	279.01	278.79
12	238.07	221.39	221.79	221.62
13	333.62	332.79	333.14	333.16
14	262.59	256.24	256.32	256.32
Average error in cm [error%]		14.59 [5.65%]	14.30 [5.53%]	14.34 [5.55%]

4.9. Experimental Results

Table 4.5: Comparison of jump height determination performance between the proposed algorithm and Catapult GPS solution.

Jump no.	Height [cm]			
	Novatel GPS	Catapult GPS	Proposed algo.(without augmented observation)	Proposed algo.(with augmented observation)
1	0.66	0.00	1.98	1.47
2	7.13	0.00	3.37	3.37
3	13.85	0.30	10.98	11.95
4	0.21	0.00	0.00	0.32
5	2.01	1.50	4.86	4.21
6	8.02	0.00	7.84	7.42
7	0.00	0.00	0.00	0.00
8	8.80	0.00	4.14	4.18
9	6.54	0.00	7.95	7.68
10	0.00	0.00	0.00	0.00
11	2.80	0.00	3.92	3.88
12	0.00	0.00	0.09	0.10
13	3.33	0.00	8.56	8.06
14	3.81	0.00	4.57	4.53
Average error in cm [error%]		3.95 [96.85%]	1.75 [42.85%]	1.56 [38.21%]

4.9. Experimental Results

Table 4.6: Comparison of jump drop determination performance between the proposed algorithm and Catapult GPS solution.

Jump no.	Drop [cm]			
	Novatel GPS	Catapult GPS	Proposed algo.(without augmented observation)	Proposed algo.(with augmented observation)
1	55.72	0.70	43.14	54.61
2	30.87	1.90	60.95	59.58
3	75.49	0.00	70.68	76.20
4	106.23	60.50	110.14	107.60
5	91.53	0.00	74.67	73.98
6	83.94	7.20	82.38	82.06
7	49.03	3.20	49.71	55.09
8	62.51	4.90	78.64	74.90
9	46.19	5.00	48.87	51.12
10	64.54	6.20	63.73	64.20
11	75.68	16.70	68.68	72.54
12	90.27	9.50	78.28	78.23
13	94.10	38.00	98.13	94.18
14	70.02	8.40	78.68	73.73
Average error in cm [error%]		59.57 [83.72%]	8.70 [12.22%]	6.71 [9.43%]

4.9. Experimental Results

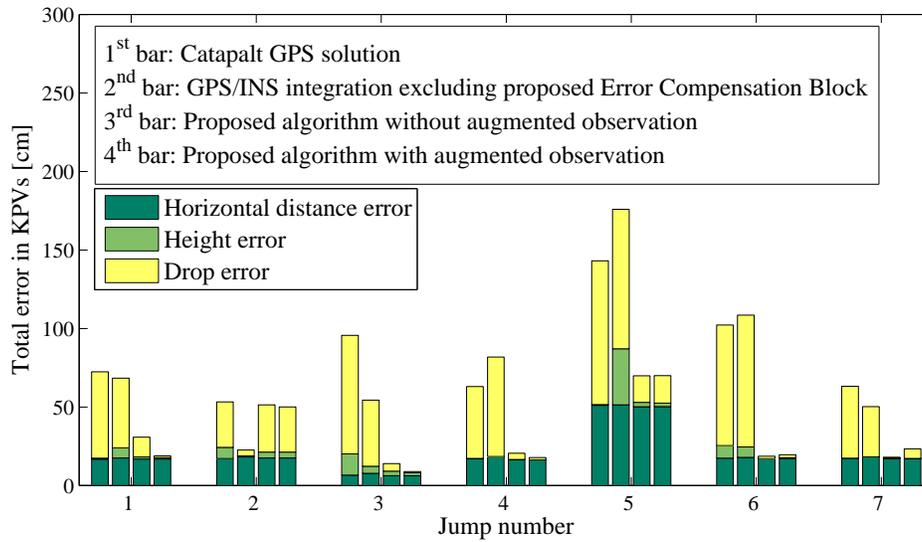


Figure 4.16: Error comparison in KPV determination among Catapult GPS solution, GPS/INS integration excluding the error compensation block and proposed algorithm [Jump no. 1-7].

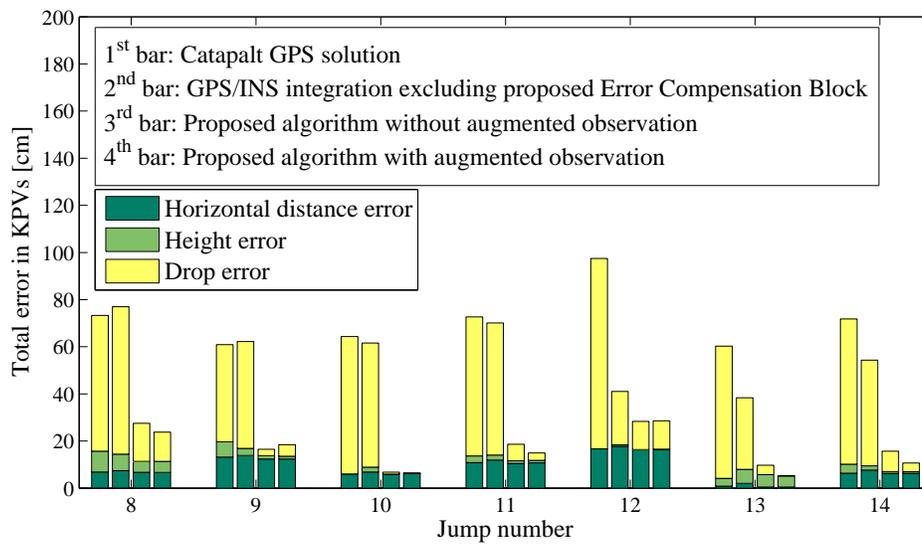


Figure 4.17: Error comparison in KPV determination among Catapult GPS solution, GPS/INS integration excluding the error compensation block and proposed algorithm [Jump no. 8-14].

4.10 Summary

In this chapter, a jump trajectory determination scheme was presented using inertial sensors and GPS updates. A novel sensor fusion scheme was proposed to mitigate the error in the inertial sensor readings. The sensor fusion scheme also provides leverage to augment the observation vector for GPS/INS integration. Experimental results have shown that the proposed algorithm calculates the jump horizontal distance, height and drop with an average accuracy of 14.34 cm, 1.56 cm and 6.71 cm respectively. Even though Recon has provided no accuracy requirement for these KPVs, the proposed algorithm accuracy should be acceptable for recreational users for whom the magnitude of jump horizontal distance, height and drop are typically small but higher accuracy is not required. For training purposes of serious athletes, the jump will typically be larger and, therefore, errors in the range of 2 to 14 cm, as demonstrated here, will be a small percentage of the true jump horizontal distance, height and drop.

Chapter 5

Conclusions and Recommendations

The objective of this research was to develop algorithms for Recon action sports goggles to determine the Key Performance Variables (KPVs) relevant to sport maneuvers. Dedicated algorithms were developed to determine the KPVs taking into account the onboard resources available in the goggles. In this chapter, the research objectives and the performance of the proposed algorithms will be highlighted. Then, the key findings of the research will be summarized. Finally, future work will be discussed and recommendations will be made on the expansion of the current work.

5.1 Objectives and Algorithm Performance

Objective

The objective of this research was to develop online algorithms to detect ski or snowboard jumps and to determine four KPVs associated with the jump, namely Air Time (AT), horizontal distance, height and drop. The algorithm should provide real-time feed back and should be developed based on the available low cost MEMS inertial sensors and single point GPS receiver contained in the Recon goggles. Recon provided an AT accuracy requirement of ± 0.1 s. Although Recon set no specific accuracy requirement for jump horizontal distance, height and drop, the algorithm is expected to produce results with sufficient accuracy to meet user's expectations. For instance, for a recreational user, an accuracy of 10 – 20 cm should be sufficient. A professional user would likely desire better accuracy. However, a professional user would typically perform larger jumps than a recreational user and, therefore, errors no more than 10 – 20 cm will be insignificant compared to the size of the jump itself.

Jump Detection

For jump detection, a novel algorithm was developed using Windowed Mean Canceled Multiplication (WMCM) and Preceding and Following Acceleration Difference (PFAD), two novel methods developed by the author. Experimental results showed that, the proposed jump detection algorithm successfully detects 92% of the jumps performed by a snowboarder whereas the current Recon jump detection algorithm and the commercial Ripxx unit accurately detect only 60% and 43.50% of the jumps respectively. The false wake up call rate of the proposed jump detection algorithm was found to be 0% for the experimental jumps. On the other hand, the current Recon jump detection algorithm exhibits a false wake up call rate of 45.60%. Considering the number of wrong detections, i.e. the combined number of true jumps undetected and false jumps detected, the Recon and Ripxx algorithms possess an error rate of 44% and 52% for all jump detections respectively. Unlike these two, the proposed algorithm has a wrong detection rate of only 8%.

Air Time Determination

To determine AT, a threshold independent probabilistic method using Multiple Attribute Decision Making was developed. The proposed algorithm exhibits an average error of 0.033 s (4.8%), which is well within the accuracy requirement of Recon. On the other hand, the current Recon and Ripxx AT determination algorithms have respective errors of 0.111 s (8.4%) and 0.538 s (39.7%) in AT determination.

Jump Horizontal Distance, Height and Drop

A GPS/INS integrated algorithm using a novel sensor error correction scheme and augmented measurement vector was developed for the determination of jump horizontal distance, height and drop. From experimental results it was found that the proposed algorithm has an error of 14.34 cm (5.55%), 1.56 cm (38.21%) and 6.71 cm (9.43%) in the determination of jump horizontal distance, height and drop respectively, whereas, the Catapult unit GPS solution has an error of 14.59 cm (5.65%), 3.95 cm (96.85%) and 59.57 cm (83.72%) accordingly. The proposed algorithm accuracy should be sufficient for the recreational user, where magnitude of the KPVs is small, and higher accuracy is not strictly required. On the other hand, for the training purposes of serious athletes, the jumps are typically large and, therefore, errors in the range of 2 to 14 cm will be insignificant with respect to the true

jump horizontal distance, height and drop. Hence, the proposed algorithm should also meet the requirements of professional athletes.

5.2 Methods and Key Findings

Among the available inertial sensors, namely accelerometer, gyroscope and magnetometer, the accelerometer is the most useful for jump detection as the change in acceleration during a jump is most consistent. Even though gyroscope readings change significantly as a result of the head movements associated with a jump, these changes are not consistent. The magnetometer is not usable for jump detection as this sensor is not related to jump dynamics. Therefore, the proposed tools for jump detection, i.e. the WMCM and PFAD methods, only exploit the accelerometer signals. Use of redundant accelerometers will increase the accuracy and reliability of the WMCM method. This is because the WMCM method extracts jump information from each individual accelerometer oriented in different directions. With an increased number of accelerometer sensors oriented in other directions, the WMCM method will have more information about true jump occurrence. Therefore, jump detection performance will increase with the addition of more accelerometers. However, the addition of more sensors will increase the power consumption and cost of the device.

Four characteristic points in resultant acceleration relevant to jump occurrence are detected for AT determination. For detecting two of the points, the closest peak method is proposed. For detecting the other two characteristic points, two methods are developed, one being threshold based and the other being a threshold independent probabilistic method using MADM. The threshold dependent method incurs a loss of generality of the algorithm. Therefore, the probabilistic method is promoted even though it causes a minor increase in the computational burden. M-TOPSIS, a modified version of the TOPSIS method, is used for the implementation of MADM. From the detection of these characteristic points, Sensor Air Time (SAT) is derived and related to the visual AT later on. It is of utmost importance to detect precisely the four characteristic points in order to determine the AT with high accuracy. Depending on the accuracy requirement and the definition of jump take-off and landing, detection of two characteristic points may also suffice.

A GPS/INS integrated algorithm was developed for jump horizontal distance, height and drop measurement and it incorporates a novel sensor error compensation scheme. Three Linear Kalman Filters (LKFs) were used to

exploit sensor fusion to compensate for the errors in the accelerometer and gyroscope. It was found to be critically important to vary the Kalman filter parameters over the relevant jump period as the athlete's dynamics significantly change before, during, and after a jump. As results have shown, the error compensation scheme is very important for the detection of the KPVs. The Extended Kalman Filter (EKF) used for GPS/INS integration is executed with an observation vector augmented with the sensor error observations calculated by the error compensation block, which results in better estimation of the sensor errors. Of course, the improvement in the accuracy of the state estimation comes at the cost of an increased computational load due to the expansion in vector size. As mentioned earlier, the proposed algorithm with the augmented observation vector has an error of 14.34 cm (5.55%), 1.56 cm (38.21%) and 6.71 cm (9.43%) in the determination of jump horizontal distance, height and drop respectively, whereas, the proposed algorithm without the augmented observation vector has an error of 14.30 cm (5.53%), 1.75 cm (42.85%) and 8.70 cm (12.22%) accordingly.

5.3 Recommendations

During the course of this research, new challenges important and worthy of investigation were encountered.

Quaternion. The body frame to navigation frame rotation matrix implemented with Euler angles may become singular at times, which can be addressed by quaternion algebra. With the application of quaternion algebra the rotation matrix calculation will also be decreased as only four elements are calculated with quaternion algebra as opposed to nine elements with Euler angles.

Kalman Filter Delay. The delay in the Kalman filter should be investigated for the proposed algorithms. If the error due to this phenomenon is significant, recovery measures should be undertaken for compensation of the filter delay.

Tight Coupling. GPS/INS integration tight coupling should also be investigated and the performance should be analyzed. The increase in computational complexity and the gain in KPV determination accuracy should be evaluated in case of the adoption of a tightly coupled GPS/INS integrated system.

Algorithm Application. Even though the algorithms are developed specially for ski or snowboard jumping, the developed techniques might have application aspects in a variety of fields with modifications and different tuning. As seen in Chapter 4, the athlete's jump dynamics in skiing/snowboarding and mountain biking are very similar. Hence, this research can also be applied to sports where maneuvering jumps plays a key role. For instance, these developed algorithms can be directly implemented in BMX biking where similar KPVs are used to judge the aerial stunts executed by professional athletes. Therefore, the scope of the developed algorithms should be widely explored, not only in outdoor sports, but also in other potential sectors such as video gaming, bio-medical engineering, etc.

Bibliography

- [1] R. Instruments, “<http://www.reconinstruments.com/>,” 2011.
- [2] J. W. Harding, J. W. Small, and D. A. James, “Feature extraction of performance variables in elite half-pipe snowboarding using body mounted inertial sensors,” in *BioMEMS and Nanotechnology III, Proc. of SPIE*, vol. 6799, 2007, pp. 679 917–1–679 917–12.
- [3] A. Waegli, “Trajectory determination and analysis in sports by satellite and inertial navigation,” Ph.D. dissertation, Department of Geomatics Engineering, University of Calgary, 2009.
- [4] Ripxx, “<http://ripxx.com/>,” 2011.
- [5] SHADOWBOX, “<http://www.shadowboxlive.com/>,” 2011.
- [6] N. El-Sheimy and X. Niu, “The promise of MEMS to the navigation community,” *InsideGNSS*, March/April 2007.
- [7] S. Godha and M. Cannon, “GPS/MEMS INS integrated system for navigation in urban areas,” *GPS Solutions*, vol. 11, no. 3, pp. 193–203, 2007.
- [8] Y. Li, A. Dempster, B. Li, J. Wang, and C. Rizos, “A low-cost attitude heading reference system by combination of GPS and magnetometers and MEMS inertial sensors for mobile applications,” *Journal of Global Positioning Systems*, vol. 5, no. 1-2, pp. 90–97, 2006.
- [9] Y. Li, J. Wang, C. Rizos, P. Mumford, and W. Ding, “Low-cost tightly coupled GPS/INS integration based on a nonlinear kalman filtering design,” in *Proceedings of the ION NTM*, 2006.
- [10] J. W. Harding, K. Toohey, D. T. Martin, C. G. Mackintosh, A. Lindh, and D. A. James, “Automated inertial feedback for half-pipe snowboard competition and the community perception,” in *The Impact of Technology on Sport II*, 2007, pp. 845–850.

Bibliography

- [11] H. Doki, T. Yamada, C. Nagai, and M. Horaki, "Development of a measurement system for snowboarding turn analysis," *In: Subic A, Ujihashi S, eds. The Impact of Technology in Sport*, pp. 324–325, 2005.
- [12] NDI, "<http://www.ndigital.com/medical/technology-em.php>," 2011.
- [13] L. Ren, Y. Zhang, Y. Wang, and Z. Sun, "Kinematics of the ankle joint complex in snowboarding," *Applied Mathematics Research eXpress*, 2007.
- [14] L. Bianchi, N. Petrone, and M. Marchiori, "A dynamometric platform for load data acquisition in snowboarding: Design and field analysis," *The Engineering of Sport 5, M. Hubbard, R. D. Mehta, J. M. Pallis, J.M.*, vol. 2, pp. 187–193, 2004.
- [15] J. W. Harding, C. G. Mackintosh, A. G. Hahn, and D. A. James, "Classification of aerial acrobatics in elite half-pipe snowboarding using body mounted inertial sensors," in *Proceedings of 7th ISEA CONFERENCE*, June 2008.
- [16] J. W. Harding, K. Toohey, D. T. Martin, A. G. Hahn, and D. A. James, "Technology and half-pipe snowboard competition-insight from elite-level judges," in *Proceedings of 7th ISEA Conference*.
- [17] S. Tsutomu, I. Kazutoyo, T. Kazuhiko, H. Hiroshi, M. Shosuke, K. Takayuki, and O. Manabu, "Development of acceleration measuring system with a 3-D gyroscope sensor during the flight phase in ski jumping," in *12th Annual Congress of the ECSS*, July 2007.
- [18] B. P. Fleantov, D. D. M. Darcy, and S. C. A. Vock, "Apparatus and methods for determining loft time and speed," Patent 5 636 146, June, 1997.
- [19] F. Castanie and A. Denjean, "Mean value jump detection using wavelet decomposition," in *Time-Frequency and Time-Scale Analysis, 1992., Proceedings of the IEEE-SP International Symposium*, oct 1992, pp. 181–184.
- [20] M. Crouse, R. Nowak, and R. Baraniuk, "Wavelet-based statistical signal processing using hidden markov models," *Signal Processing, IEEE Transactionson*, vol. 46, no. 4, pp. 886–902, apr 1998.

- [21] H. Teng and Y. Qi, “Application of wavelet technique to freeway incident detection,” *Transportation Research Part C: Emerging Technologies*, vol. 11, no. 3-4, pp. 289–308, 2003, traffic Detection and Estimation. [Online]. Available: <http://www.sciencedirect.com/science/article/B6VGJ-492707R-2/2/ced7d3a53aa7787eaf87c82c01bc9184>
- [22] M. Raimondo and N. Tajvidi, “A peaks over threshold model for change-point detection by wavelets,” *Statistica Sinica*, vol. 14, pp. 395–412, 2004.
- [23] DARTFISH, “<http://www.dartfish.com/en/index.htm>,” 2011.
- [24] C. Nicolas and M. Tavernier, “Contribution to the analysis of kinematics and energy and skating cross country skiing. application to the comparative study of three nordic disciplines: cross country skiing, biathlon and nordic combined. comparative analysis of two 3d motion analysis system using a stick and a human model volumic model and application to human case studies in cross-country skiing,” Ph.D. dissertation.
- [25] CODAMOTION, “<http://www.codamotion.com/>,” 2011.
- [26] VICON, “<http://www.vicon.com/>,” 2011.
- [27] organicmotion, “<http://www.organicmotion.com/>,” 2011.
- [28] POLAR, “<http://www.polarusa.com/us-en/>,” 2011.
- [29] GARMIN, “<http://www.garmin.com/>,” 2011.
- [30] FRWD, “<http://www.frwd.fi/>,” 2011.
- [31] CATAPULT, “<http://www.catapultinnovations.com/index.html>,” 2011.
- [32] Sportvision, “<http://www.sportvision.com/>,” 2011.
- [33] TracTrac, “<http://www.tractrac.com/>,” 2011.
- [34] J. Skaloud and P. Limpach, “Synergy of CP-DGPS, accelerometry and magnetic sensors for precise trajectography in ski racing,” in *In Proceedings of the ION GPS/GNSS 2003*.
- [35] S. Ducret, P. Ribot, R. Vargiolu, J. Lawrence, and A. Midol, “Analysis of downhill ski performance using GPS and grounding force recording,” *Science and Skiing III*.

- [36] K. Zhang, R. Deakin, R. Grenfell, Y. Li, J. Zhang, W. N. Cameron, and D. M. Silcock, "GNSS for sports-sailing and rowing perspectives," *Journal of Global Positioning Systems*, vol. 3, no. 1-2.
- [37] K. Zhang, R. Grenfell, R. Deakin, Y. Li, Z. Jason, A. Hahn, C. Gore, and T. Rice, "Towards a low-cost, high output rate, real-time GPS rowing coaching and training system," in *In Proceedings of the ION GPS/GNSS 2003*.
- [38] A. Waegli and J. Skaloud, "Optimization of two GPS/MEMS-IMU integration strategies with application to sports," *GPS Solutions*, vol. 13, pp. 315–326, 2009, 10.1007/s10291-009-0124-5. [Online]. Available: <http://dx.doi.org/10.1007/s10291-009-0124-5>
- [39] J. Skaloud, "Optimizing georeferencing of airborne survey systems by INS/DGPS," no. UCGE Report 20126.
- [40] I. Skog and P. Hndel, "A low-cost GPS aided inertial navigation system for vehicle applications," in *Proc. EUSIPCO*.
- [41] S. Godha, "Performance evaluation of low cost MEMS-based IMU integrated with GPS for land vehicle navigation application," Master's thesis, Department of Geomatics Engineering, University of Calgary, February 2006.
- [42] H. Qi and J. Moore, "Direct kalman filtering approach for GPS/INS integration," *Aerospace and Electronic Systems, IEEE Transactions on*, vol. 38, no. 2, pp. 687–693, apr 2002.
- [43] I. Oppermann, L. Stoica, A. Rabbachin, Z. Shelby, and J. Haapola, "Uwb wireless sensor networks: UWEN-a practical example," *Communications Magazine, IEEE*, vol. 42, no. 12, pp. S27–S32, dec. 2004.
- [44] G. Opshaug and P. Enge, "GPS and UWB for indoor navigation," in *Proceedings of the ION GPS/GNSS*, 2001.
- [45] T. Kitasuka, T. Nakanishi, and A. Fukuda, "Design of WiPS: WLAN-based indoor positioning system," *Korea Multimedia Society*, vol. 7, no. 4, pp. 15–29, 2003.
- [46] R. T. Ogden, *Essential Wavelets for Statistical Applications and Data Analysis*. Birkhauser Boston, 1997.

Bibliography

- [47] D. C. Montgomery, *Design and Analysis of Experiments*, 5th ed. John Wiley & Sons (Asia) Pte. Ltd., 2006.
- [48] K. P. Yoon and C. L. Hwang, *Multiple Attribute decision making: An Introduction*. Sage University Paper series on Quantitative Applications in the Social Sciences, 07-104. Thousand Oaks, CA: Sage, 1995.
- [49] L. Ren, Y. Zhang, Y. Wang, and Z. Sun, "Comparative analysis of a novel M-TOPSIS method and TOPSIS," *Applied Mathematics Research eXpress*, 2007.
- [50] R. V. Wong, "Development of a RLG strapdown inertial survey system," Ph.D. dissertation.
- [51] J. Diebel, "Representing attitude: Euler angles, unit quaternions, and rotation vectors," Stanford University, Stanford, California 94301-9010, Tech. Rep., October 2006.
- [52] D. H. Titterton and J. L. Weston, *Strapdown Inertial Navigation Technology*, 2nd ed. The Institution of Electrical Engineers and The American Institute of Aeronautics and Astronautics, 2004.
- [53] J. Bortz, "A new mathematical formulation for strapdown inertial navigation," *Aerospace and Electronic Systems, IEEE Transactions on*, vol. AES-7, no. 1, pp. 61 –66, jan. 1971.
- [54] M. Ignagni, "On the orientation vector differential equation in strapdown inertial systems," *Aerospace and Electronic Systems, IEEE Transactions on*, vol. 30, no. 4, pp. 1076 –1081, oct 1994.
- [55] Kionix, "Tilt sensing with kionix MEMS accelerometers," Tech. Rep.
- [56] C. Konvalin, "Calculating heading, elevation and bank angle," MEM-SENSE, Tech. Rep. MTD-0801.
- [57] K. J. Walchko and D. P. A. C. Mason, "Inertial navigation," in *Florida Conference on Recent Advances in Robotics*, 2002.
- [58] S. Nassar, "Improving the inertial navigation system (INS) error model for INS and INS/DGPS applications," Ph.D. dissertation, Department of Geomatics Engineering, University of Calgary, 2003.
- [59] H. Luinge and P. Veltink, "Measuring orientation of human body segments using miniature gyroscopes and accelerometers,"

Bibliography

- Medical and Biological Engineering and Computing*, vol. 43, pp. 273–282, 2005, 10.1007/BF02345966. [Online]. Available: <http://dx.doi.org/10.1007/BF02345966>
- [60] E. Nebot and H. Durrant-Whyte, “Initial calibration and alignment of low-cost inertial navigation units for land vehicle applications,” *Journal of Robotic Systems*, vol. 2, no. 16, pp. 82–92, 1999.
- [61] S. M. Kay, *Fundamentals of Statistical Signal Processing: Estimation Theory*. Prentice Hall PTR, Upper Saddle River, 1993.
- [62] N. El-Sheimy, “The potential of partial IMUs for land vehicle navigation,” *InsideGNSS*, Spring 2008.
- [63] E.-H. Shin, “Accuracy improvement of low cost INS/GPS for land applications,” Master’s thesis, Department of Geomatics Engineering, University of Calgary, Canada, December 2001.
- [64] P. G. Savage, “Strapdown analytics,” *Strapdown Associates, Inc*, vol. 1, 2000.
- [65] R. M. Rogers, *Applied Mathematics in Integrated Navigation Systems*. AIAA Education Series, 2000.
- [66] N. El-Sheimy, “Inertial techniques and INS/DGPS integration,” *ENGO 623 Course Notes*, 2004.
- [67] N. R. Canada, “<http://www.geod.nrcan.gc.ca>,” 2011.
- [68] M. Ehrgott, *Multicriteria Optimization*, 2nd ed. Springer-Berlin Heidelberg New York.
- [69] O. Bandte, “A probabilistic multi-criteria decision making technique for conceptual and preliminary aerospace systems design,” Ph.D. dissertation, Georgia Institute of Technology, September 2000.
- [70] A. Davey and D. Olson, “Multiple criteria decision making models in group decision support,” *Group Decision and Negotiation*, vol. 7, pp. 55–75, 1998.

Appendix

Appendix A Multiple Attribute Decision Making

In general, a Multiple Attribute Decision Making (MADM) problem can be described as [68, 69]

$$\max_{x_1, x_2, \dots, x_p} \{F_1(x_1, x_2, \dots, x_p), F_2(x_1, x_2, \dots, x_p) \dots, F_P(x_1, x_2, \dots, x_p)\} \quad (\text{A-1})$$

subject to a set of constraints

$$g_r(x_1, x_2, \dots, x_p) \leq 0, \quad r = 1, 2, \dots, L \quad (\text{A-2})$$

where $F_i, i = 1, 2, \dots, P$ are the criteria and $x_i, i = 1, 2, \dots, n$ are the design variables which the criteria and constraints depend on. Any set $X = \{x_1, x_2, \dots, x_p | g(x_1, x_2, \dots, x_p) \leq 0\}$ is the set of feasible design variables which complies with the design constraints. For each point in X there is an associated set $\{F_1, F_2, \dots, F_P\}$ such that X is mapped into a set

$$S = \{F_1, F_2, \dots, F_P | x_1, x_2, \dots, x_p \in X\} \quad (\text{A-3})$$

in the criteria space. Further, $\{x_1^*, x_2^*, \dots, x_p^*\}$ is called the optimal solution if and only if $\{x_1^*, x_2^*, \dots, x_p^*\} \in X$ and $F_i\{x_1^*, x_2^*, \dots, x_p^*\} \geq F_i\{x_1, x_2, \dots, x_p\}, i = 1, 2, \dots, P$.

TOPSIS

The Technique for Order Preference by Similarity to Ideal Solution (TOPSIS) is one of the most popular compromising methods among the compensatory techniques, utilizing preference information provided in the form of weights $w_i^c, i = 1, 2, \dots, P$ for each criterion. TOPSIS defines an index called similarity (or relative closeness) to the positive-ideal solution by combining the proximity to the positive-ideal solution and remoteness from the negative-ideal solution [48]. The positive-ideal solution simultaneously optimizes each objective. The positive-ideal solution to a multi-criteria problem

A. Multiple Attribute Decision Making

is generally not feasible, but can serve as a standard to the alternatives [70]. Assuming that the greatest utility can be achieved from the positive-ideal solution, closeness to the positive-ideal solution can be considered analogous to maximizing utility.

The concept of the decision matrix D is applicable to all other MADM techniques and can be expressed as

$$D = \begin{bmatrix} f_{11} & f_{12} & \cdots & f_{1i} & \cdots & f_{1P} \\ f_{21} & f_{22} & \cdots & f_{2i} & \cdots & f_{2P} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ f_{j1} & f_{j2} & \cdots & f_{ji} & \cdots & f_{jP} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ f_{M1} & f_{M2} & \cdots & f_{Mi} & \cdots & f_{MP} \end{bmatrix}. \quad (\text{A-4})$$

D is an M-by-P-matrix with elements f_{ji} that indicate the value of the criterion F_i with respect to the alternative A_j . The initial step of most MADM problems is to calculate the normalized rating [48], r_{ji} , for each element of D . The vector normalization procedure can be given as

$$r_{ji} = \frac{f_{ji}}{\sqrt{\sum_{j=1}^M x_{ji}^2}}, \quad j = 1, \dots, M; \quad i = 1, \dots, P. \quad (\text{A-5})$$

Hence the weighted normalized ratings are calculated as

$$v_{ji} = w_i^c r_{ji}, \quad j = 1, \dots, M; \quad i = 1, \dots, P. \quad (\text{A-6})$$

In such a context, the positive ideal solution can be denoted as

$$\begin{aligned} A^* &= \{v_1^*, \dots, v_i^*, \dots, v_P^*\} \\ &= \{(\max_j v_{ji} | i \in I_1), \\ &\quad (\min_j v_{ji} | i \in I_2) | j = 1, \dots, M\} \end{aligned} \quad (\text{A-7})$$

where v_i^* is the best value for the i th criterion among all available alternatives. I_1 is the set of benefit attributes and I_2 is the set of cost attributes. Benefit attributes are those which are expected to be maximized and cost attributes are those which are expected to be minimized. Consequently, the

negative-ideal solution can be denoted as

$$\begin{aligned} A^- &= \{v_1^-, \dots, v_i^-, \dots, v_P^-\} \\ &= \{(\min_j v_{ji} | i \in I_1), \\ &\quad (\max_j v_{ji} | i \in I_2) | j = 1, \dots, M\} \end{aligned} \quad (\text{A-8})$$

where v_i^- is the worst value for the i th criterion among all alternatives. The separation of each alternative from the positive ideal solution, A^* , can then be given by

$$D_j^* = \sqrt{\sum_{i=1}^P (v_{ji} - v_i^*)^2}, \quad j = 1, \dots, M. \quad (\text{A-9})$$

Similarly, the separation from the negative ideal solution, A^- , can be given by

$$D_j^- = \sqrt{\sum_{i=1}^P (v_{ji} - v_i^-)^2}, \quad j = 1, \dots, M. \quad (\text{A-10})$$

Afterwards, similarities to the positive ideal solution are calculated as

$$C_j^* = \frac{D_j^-}{D_j^* + D_j^-}, \quad j = 1, \dots, M. \quad (\text{A-11})$$

Note that $0 \leq C_j^* \leq 1$, where $C_j^* = 0$ when $A_j = A^-$, and $C_j^* = 1$ when $A_j = A^*$. Finally, all the alternatives are ranked against C_i^* in descending order and the alternative with the maximum C_i^* (rank 1) can be chosen as the optimal solution for the system. For a two dimensional case, i.e. for MADM for two criteria, and six alternatives, the TOPSIS method is illustrated in Figure A-1.

M-TOPSIS

The TOPSIS method prone to ‘Rank Inversion’ problem which can be resolved by the Modified TOPSIS (M-TOPSIS) method [49]. Rank inversion means the change in relative ranking among any of the two alternatives in the event of exclusion or inclusion of any other alternative.

In M-TOPSIS, the $D_j^*D_j^-$ plane is constructed and a new Optimized Ideal Reference Point (OIRP) is constructed. In the $D_j^*D_j^-$ plane, D^* is the x -axis and D^- is the y -axis. (D_j^*, D_j^-) represents each alternative and

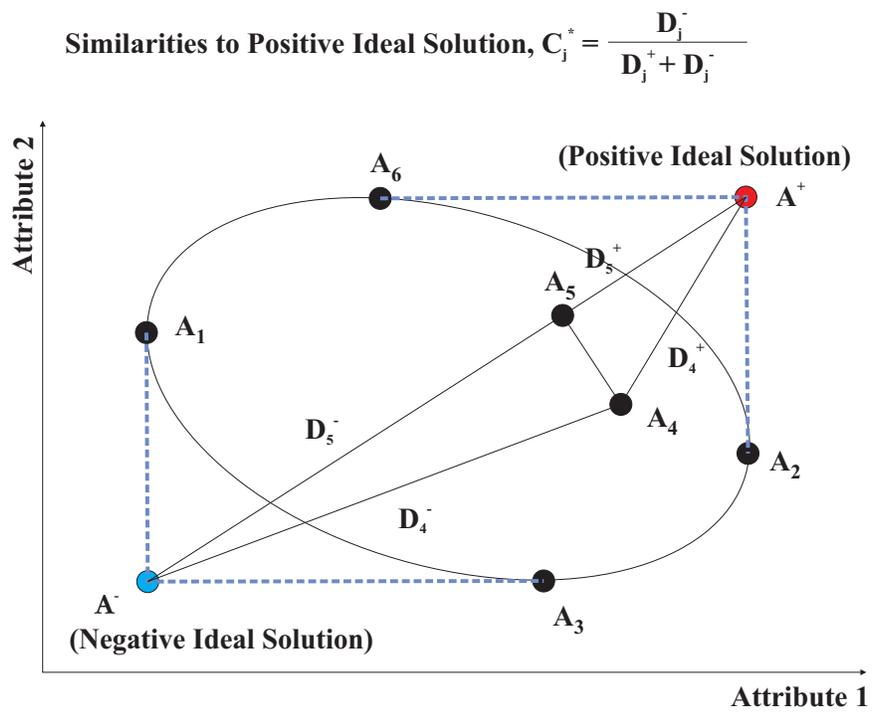


Figure A-1: TOPSIS method for MADM.

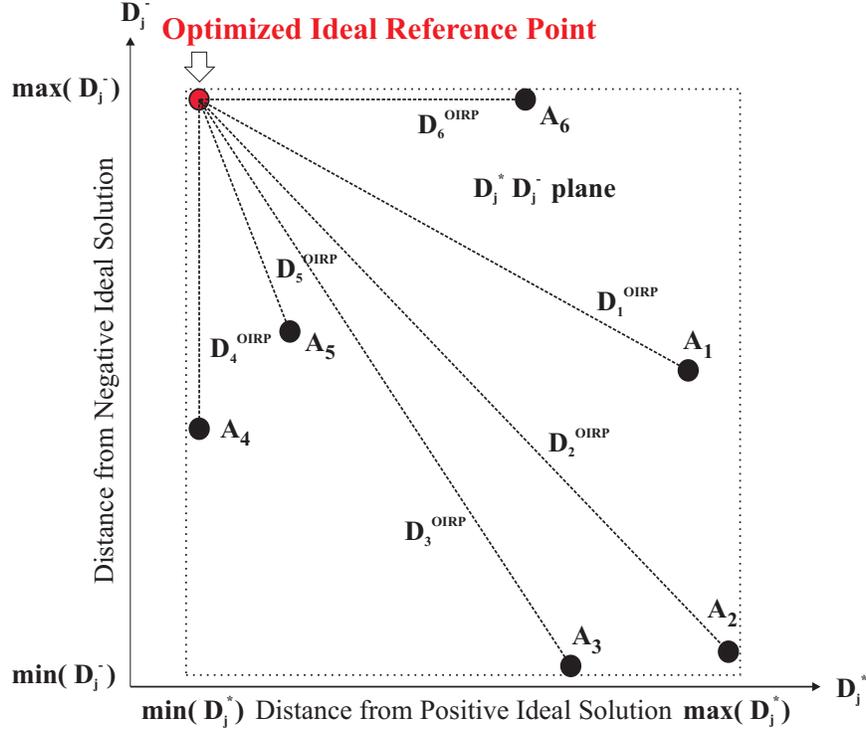


Figure A-2: M-TOPSIS method for MADM.

the OIRP is the point $(\min(D_j^*), (\max)(D_j^-))$. Then the distances from the OIRP to the alternatives are calculated as

$$D_j^{OIRP} = \sqrt{\{\min(D_j^*) - D_j^*\}^2 + \{\min(D_j^-) - D_j^-\}^2}, \quad j = 1, \dots, M. \quad (\text{A-12})$$

Afterwards, D_j^{OIRP} is sorted in ascending order and all the alternatives are ranked against its' closeness to the OIRP. The alternative with minimum D_j^{OIRP} (rank 1) is considered as the optimal choice for the MADM problem. The $D_j^* D_j^-$ plane and the distances between OIRP and the alternatives are depicted in Figure A-2.