Optical Flow Measurement of Human Walking

by

Qingwen Liu

Submitted in partial fulfillment of the requirements for the degree of

Master of Science

in

Mechanical Engineering

TUFTS UNIVERSITY

August 2012

ADVISER: Jason Rife
Abstract

This thesis presents a method for using optical flow measurements to estimate stride length for pedestrian navigation applications. Optical flow sensors, such as the detectors used in an optical computer mouse, measure the velocity of visual features traversing an imaging array. The author considers the case in which the optical flow sensor is attached to the leg of a pedestrian and used to infer distance traveled. In this configuration, optical flow data are a projection of the velocity and angular velocity of the leg to which the sensor is attached; a dynamic motion model is needed to estimate leg states and to infer stride length from the optical flow data. In this thesis, a very simple dynamic walking model is introduced, called the Spring Loaded Inverted Pendulum (SLIP) model. In a hardware-based trial, the basic SLIP model estimated stride length with 10% error. The author anticipates that refinements to the basic SLIP model will enable more accurate stride-length estimation in the future.

Keywords: Optical Flow, Stride Length, Pedestrian Navigation
Acknowledgements

Thanks my honored advisor Professor Jason Rife for his help and support during my completion of Master of Science degree. Not only I have learned knowledge from him, but I do have learned from his attitude on research and life. I will never forget my two-year study under the guidance of my advisor: the time we discuss concepts, enjoy progress and analyze results. Thanks him again for his guidance and care.

Thank you all my laboratory members. They inspired me to pave on the research road. I have learned more from them from class. Thanks to their valuable suggestions and feedbacks, to my thesis and presentations, I can dig more into my research. Our friendship is what I treasure within a lifelong time.

Thank you my family and friends. Their support is the power to strength me. They are willing to listen to my complaints; they are also willing to check wording and grammars in my thesis. I feel I am always being supported and loved.
# Table of Contents

Abstract ............................................................................................................................... ii

Acknowledgements ............................................................................................................ iii

Chapter 1: Introduction ....................................................................................................... 1  
1.1 Motivation for Indoor Navigation ............................................................................. 1  
1.2 Existing Techniques .................................................................................................. 1  
1.3 Proposed Solution ..................................................................................................... 2  
1.4 Contributions ............................................................................................................ 4  
1.5 Thesis Overview ....................................................................................................... 5

Chapter 2 Optical Flow (OF) .............................................................................................. 6  
2.1 Background ............................................................................................................... 6  
2.2 Correlation-based Method ........................................................................................ 7  
2.2.1 Introduction ............................................................................................................ 7  
2.2.2 Algorithm Description ........................................................................................... 7  
2.4 Summary ................................................................................................................. 10

Chapter 3 Spring Loaded Inverted Pendulum (SLIP) ....................................................... 11  
3.1 Background ............................................................................................................. 11  
3.2 Method .................................................................................................................... 11  
3.2.1 Dynamic Model ................................................................................................... 12  
3.2.2 Kinematic Model ................................................................................................. 15
Control vector refers to initial conditions including initial velocities $x_0(0)$, initial angle $\delta_0$ and spring constant $k$. These control variables are the same in determining SLIP dynamics in section 3.2.1.
Figures

Fig. 1 Optical Flow Sensor for Computer Mouse ............................................................... 4
Fig. 5 Frame 1 ..................................................................................................................... 9
Fig. 6 Frame 2 ..................................................................................................................... 9
Fig. 9 SLIP Model Visualized at One Instant ................................................................. 12
Fig. 10 Leg Geometry .................................................................................................... 16
Fig. 11 Swing Leg Position at One Instant ................................................................. 21
Fig. 12 System Configuration ....................................................................................... 29
Fig. 13 Outdoor Test Area Surface .............................................................................. 34
Fig. 14 Optical Flow measurements over 9 seconds, Correlation-Based Method ....... 36
Fig. 15 Optical Flow Measurement and Simulation, Correlation-Based Method ....... 37
Fig. 16 Proportion of Stance Phase to Swing Phase of measurements and simulations ... 38

Tables

Table 1 Physical Parameters for the SLIP Model ............................................................. 35
Table 2 Initial Guess of States for SLIP ..................................................................... 35
Table 3 Stride Length Estimation ................................................................................ 39
Chapter 1: Introduction

1.1 Motivation for Indoor Navigation

Pedestrian navigation is of particular interest in indoor environments. Despite of its wide coverage outdoors, Global Positioning System (GPS) signals are not available indoors, and degrade significantly in deep city canyons. This has led researchers to focus on developing alternative navigation methods for pedestrian applications in indoor and other GPS-denied environments.

Many emerging applications demand high accuracy, indoor navigation for pedestrians [1]. Emergency responders, for example, are concerned with quickly locating their specific destinations in unfamiliar buildings, or escaping from buildings in poor visibility conditions during emergencies. Another interesting scenario is the consumer-grade navigation guidance, for example through a museum or shopping mall, where visitors would benefit from real-time navigation guidance. Navigation strategies designed to assist vision-impaired citizens bring up a different but compelling set of challenges [2].

1.2 Existing Techniques

Given the potentially large market for pedestrian navigation, many technologies have been proposed. Most proposed strategies fall into three categories: beacon-based, inertial-based and vision-based navigation.

Beacon-based methods rely on sensors such as infrared, ultrasound or Ultra-Wideband Impulse Radio (UWB-IR) sensors [3][4][5] and sometimes Wi-Fi hotspots [6]. They are based on principles similar to those underlying GPS navigation, measuring distance from
a receiver to transmitters with known locations. These strategies are highly accurate but require additional infrastructure.

Inertial-based methods oftentimes make use of low-cost Inertial Measurement Unit (IMU) to collect acceleration data [7]. The accuracy of these dead reckoning technologies depends largely on the quality of the IMU itself and that of the stride length estimation function. Besides IMU sensors drift rapidly; inertial-based methods are complementary to GPS positioning outdoors [8], or are applied in a vision-aid scenario [9].

Among the vision-based methods, numerous techniques have been proposed. Most approaches are based on feature tracking [10]. These vision-based techniques rely on capturing images at relatively high resolution, making the methods computationally intensive. Besides, clear vision is required to capture features.

The author aims at improving positioning accuracy indoors. Ideally, size, weight and power (SWAP) constrained applications are preferred to be implemented for real-time pedestrian navigation. By known limitations of existing techniques, (1) the author seeks to operate on a sensor with no other infrastructures; (2) it supposes to have smaller drift errors than those associated with an IMU; (3) this sensor can operate in a low-visibility environment such that it still works in an emergency environment.

1.3 Proposed Solution

The method proposed in this paper is an alternative that addresses some of the shortcomings in the previously introduced methods. Specifically, the author believes that an Optical Flow (OF) sensor can content requirements above. First, a small OF sensor can be attached to pedestrian’s body or be embedded in other portable devices, such as cellphone. OF-based navigation does not necessarily require transmitters or other
infrastructures. Second, an OF sensor can serve as an alternative to IMU, with a smaller drift error. This is because an OF sensor measures OF as a function of velocity, the first derivative of displacement; while an IMU sensor measures the acceleration, the second derivative of displacement. Ideally, an IMU sensor will incur larger drift errors by integration directly. Third, an OF sensor can work in low-vision environment because light source can be incorporate with an OF sensor.

To start with, I would like to briefly introduce OF and OF sensors. OF is the pattern describing motion of objects caused by the relative movement between the environment and the sensor. OF sensors are capable to capture visual motion and output OF in two dimensions. OF measurements can be obtained from conventional video cameras or from more compact sensors. These sensors do not require a clear vision with high light intensity. Moreover, light sources can be incorporated with OF sensors, similar to those designed for use in an optical mouse (see Fig. 1). Recently, compact, low-power OF sensors have also been custom-built for use in small Unmanned Aerial Vehicles [11], but they have not applied for indoor navigation. In order to navigate indoors with a low-vision, OF sensor is supposed to attach on the leg or mount on the foot, making it possible to detect features with little light intensity. OF data, as a function of sensor position, velocity and angular velocity, however, cannot be directly integrated to infer displacement.
In order to relate OF with pedestrian walking distance, method is proposed to use a model-based approach where a dynamic model, like Spring Loaded Inverted Pendulum (SLIP) model [12], helps relate the OF patterns that arise from a pedestrian walking with an OF sensor attached to the leg, to the corresponding distance traveled by the pedestrian. This idea is achieved by adjusting control parameters until OF simulations matches the OF measurements. It is thus possible to estimate corresponding distance traveled based on model.

1.4 Contributions

As identified in the previous section, a key challenge in applying an OF sensor to pedestrian navigation is to integrate sensor data using a dynamic walking model. The primary thesis contribution directly addresses this challenge, as detailed below:

Introduced a new fusion method that combines the SLIP model with OF measurements [13]. This method estimates stride length by using only one OF sensor. Estimation is computed by matching OF measurements from an OF sensor and OF simulations from a walking model as closely as possible. Such a combination of techniques has not previously been reported in the academic literature. In order to realize this vision, several innovations were required. These sub-contributions include the following:

a. Developed an algebraic kinematic model based on SLIP to specifically relate limb positions and velocities to a dynamic simulation of hip motion. The SLIP model was selected because the dynamic equations are fast to integrate as compared to more detailed dynamic models of leg segments. The SLIP model is
originally a dynamic model used to qualitatively analyze the human gait cycle, and specifically to compute the trajectory of the hip. By developing a kinematic model based on SLIP, more detailed information including limbs and knees positions and velocities can be derived. This kind of information is important to calculate OF simulation.

b. **Tested the new fusion method with experiment outdoors and presented results and analysis.** The new fusion method was tested with real data. (1) Pedestrian walking distanced estimate prototype based on SLIP was presented; (2) data was collected from a thigh-mounted conventional video camera and OF measurements and OF simulations are calculated. (3) OF measurements and OF simulations were compared; (4) results were compared with ground truth. This thesis showed relative error of 10%.

1.5 Thesis Overview

The remainder of the thesis is organized as below.

In Chapter 2 the algorithm (the correlation method) for computing OF is described.

In Chapter 3 the mathematical representation of the SLIP model and the novel kinematic walking model are presented.

In Chapter 4 the fusion method is discussed and methods from Chapter 2 and Chapter 3 are incorporated into the fusion method.

In Chapter 5 the fusion method is tested based on a proof-of-concept trial.

In Chapter 6 contributions are reiterated and future works are summarized.
Chapter 2 Optical Flow (OF)

2.1 Background

OF refers to a class of vision-processing algorithms that identify the direction and the magnitude visual features move across an imaging array. OF is caused by the motion of features in the image plane due to motion of the sensor relative to a scene, which is generally assumed to be static and approximately planar. The computation of OF is to determine the 3-D movement of the sensor projected onto a 2-D image plane [14]. In the animal kingdom, intelligent creatures such as honeybees depend on their built-in OF field observation system for obstacle avoidance and route recognition [15]. Human beings are also thought to depend on OF for obstacle avoidance. OF has also been applied for obstacle avoidance in robot navigation [16].

A large number of OF algorithms have been proposed in the literature. The idea of OF is first introduced in Gibson’s book, The Perception of the Visual World, 1950 [17]. Afterwards researches have focused on developing varieties of algorithms to determine OF fields. Currently, three major categories of OF algorithms can be identified: gradient-based, correlation-based and spatiotemporal-based method. Typical OF algorithms extract only two components of motion: horizontal and vertical velocity along each axis of pixel array (in units of pixels for second). Extensions that infer angular velocity or three-dimensional velocities are possible in some cases for applications using large imaging arrays. In this thesis author focuses on a 2-D OF field in planar image plane so as to infer pedestrian movement on the targeted ground surface. Methods implemented are discussed in the following sections.
2.2 Correlation-based Method

2.2.1 Introduction

The correlation-based method is a robust image processing technique. It seeks to match image arrays in consecutive frames by finding the maximum cross-correlation. Image displacement (in pixels) is the distance between image arrays in corresponding frames. This method is straightforward, robust to noise, but computational expensive. This method requires global searching over images, in order to get the maximum correlation value of two image arrays. Another important parameter for computing OF is the size of the correlation windows chosen from frames. The larger the windows are, the more features will be contained therein. Thus, larger windows result in estimation accuracy increasing while computational workload also gets larger. In the literature, research focuses on in improving accuracy for real-time applications [18][19]. In this thesis OF is calculated off-line and accuracy is of our most important consideration, as the goal of the proof-of-concept is to determine the best possible performance of the method given high-quality OF measurements. Therefore comparatively large windows are used.

2.2.2 Algorithm Description

As discussed above, the idea of correlation-based methods is to match certain images patches in subsequent frames [20]. OF equals the displacement from one patches to another divided by the interval time. For example, consider an image sub-image array $I_1$ chosen from Frame 1 (see Fig. 2). It is matched to an sub-image array $I_2$ in Frame 2.

The correlation-based method scans to find the best match $I_2$. This is obtained by maximizing the cross-correlation $\Phi$ in (2.1):
\[
\Phi(\Delta x, \Delta y) = \sum_{m=0}^{row-1} \sum_{n=0}^{col-1} I_1(m, n) \cdot \text{conj}(I_2(m + \Delta x, n + \Delta y))
\] (2.1)

Here, the variables \textit{row} and \textit{col} symbolize the row and column of the image array respectively. Note that the correlation may be computed for all displacements within the following bounds:

\[
\begin{cases}
0 \leq \Delta x < \text{row} + \text{col} - 1; \\
0 \leq \Delta y < \text{row} + \text{col} - 1
\end{cases}
\] (2.2)

The time interval between these two frames is \(\Delta t\). The estimated 2-D displacements in the image plane are \(\Delta x\) and \(\Delta y\), along the horizontal and vertical axis respectively, as returned by the cross-correlation \(\Phi\).

\[
\begin{bmatrix}
\Delta x \\
\Delta y
\end{bmatrix} = \text{arg max}(\Phi)
\] (2.3)

Therefore, the OF values, in terms of pixel/second, are computed from the above equation as:

\[
\mathbf{F} = \begin{bmatrix}
\Delta x \\
\Delta y \\
\frac{\Delta x}{\Delta t} \\
\frac{\Delta y}{\Delta t}
\end{bmatrix}
\] (2.4)
Fig. 2 Frame 1

Fig. 3 Frame 2
Note \( u \) and \( v \) both cannot reach beyond \( \frac{(\text{row}+\text{col}-1)}{\Delta t} \). In this case, the OF is so fast that the true match \( I_2 \) is outside of Frame 2. Even though the cross-correlation function will still produce a maximum value, the OF calculation is nonsense and unpredictable errors appear. In order to avoid this phenomenon, it is preferable to increase the size of frames, or decrease the interval time \( \Delta t \). Increasing the frame size, however, will largely aggravate the computational workload; decreasing the interval time requires a more sensitive sensor [21]. In this thesis, \( I_1 \) and \( I_2 \) are 120×120 pixel\(^2\); Frame 1 and Frame 2 are 240×500 pixel\(^2\). Currently the author cares most about obtaining high accuracy.

2.4 Summary

In this thesis, the correlation-based method is applied to get OF measurements from frames captured from a conventional video camera. The correlation-based method is accurate and robust to noise yet computational workload is large. It exerts global searching in order to align sub-images in consecutive frames. At the stage of prototype demonstration we more concern about accuracy. Frames are chosen to be 240×500 pixel\(^2\); sub-images are 120×120 pixel\(^2\). For the future work it is important to incorporate a SWAP constrained OF sensor for pedestrian navigation.
Chapter 3 Spring Loaded Inverted Pendulum (SLIP)

3.1 Background

Understanding human walking patterns is still an open topic discussed in literature. Many different models have been proposed, including kinematic, dynamic and energy-related models [22][23]. Some researchers make certain assumptions and build robots mimicking the behavior of human walking. Two famous models, the passive walker [24] and the SLIP model [25], are well-studied.

The thesis focuses on the SLIP model to start with. This model is simple and straightforward, an important consideration in reducing computational complexity. The SLIP model is well-studied and it can be tuned to represent both walking and running. Robots based on SLIP models have been constructed in the past 10 years, and they have succeeded in mimicking the basic behavior of human walking and running [26]. A particular benefit of the SLIP model is that it can describe dynamic walking without any need to identify models for human muscle or for the control commands sent to those muscles.

3.2 Method

The SLIP model describes the motion of the hip during walking and running. The model is highly simplified in that all body mass is lumped at a single point (i.e., at the hip). Accordingly, the hip is also referred to as the center of mass G, as shown in Fig. 4. It should be noted that this placement of the center of mass of the SLIP model at the hip is an approximation (as the center of mass of most humans does not fall precisely at the hip). The SLIP model assigns no mass to the legs; nor is the leg geometry explicitly modeled.
Rather the leg is simply represented as a spring-like force that acts on the hip, so long as the foot is in contact with the ground.

![Fig. 4 SLIP Model Visualized at One Instant](image)

### 3.2.1 Dynamic Model

The particular form of the SLIP model used in this research, assumes that the leg force offsets gravity and provides an additional spring force $F_s$ as described in the following equation.

$$m\ddot{x}_G = F_s$$

(3.1)

Here $m$ is the lumped mass of the pedestrian and $x_G$ is the position vector describing the location of the hip in the sagittal plane. Forces exerted by muscles, tendons and bones within the leg are modeled with a spring force $F_s$. The spring force acts between the hip
and the heel contact point H. If the position vector to the heel is defined to be $x_H$, the spring equation has the following form, where $L_{\text{ref}}$ is the reference length of the spring.

$$
F_s = \begin{cases} 
  k \left( L_{\text{ref}} - L \right) \frac{x_G - x_H}{L}, & L \leq L_{\text{ref}} \\
  0, & \text{otherwise}
\end{cases}
$$

(3.2)

In this equation, the length $L$ is defined:

$$
L = \| x_G - x_H \| 
$$

(3.3)

The reference length $L_{\text{ref}}$ is equal to the total leg length, including the hip-to-knee (or thigh) length $L_{\text{th}}$ and the knee-to-heel (or shank) length $L_{\text{sh}}$:

$$
L_{\text{ref}} = L_{\text{th}} + L_{\text{sh}}
$$

(3.4)

The spring constant is $k$. Unlike $L_{\text{ref}}$, it cannot be directly measured and it affects the locomotion of hip.

This model initiates when leg is fully extended and the heel strike happens (the instance when one leg is about to hit the ground). If the initial length of spring is defined as $L_0$, it has the relationship:

$$
L_0 = L_{\text{ref}}
$$

(3.5)

Another condition is the initial angle $\delta_0$. It represents the angle between the extended to the vertical axis. Given $L_0$ and $\delta_0$ the starting position of the hip $x_G$ is predetermined.

In order to solve (3.2), initial velocity of hip $v_0$ should also been given to start with.
This model implicitly assumes that either one leg or the other is always in contact with the ground. The first leg is lifted off the ground (toe off) at the same instant that the second leg contacts the ground (heel strike). This simplified model does not specifically simulate double stance (in which both legs are in contact with the ground simultaneously) or double swing (in which neither leg is in contact with the ground). Because flight is not permitted in walking simulations, it is assumed that the pedestrian shifts from heel contact to toe contact if the length $L$ exceeds $L_{ref}$. Further description of human gaits is in the book [27].

The SLIP model implemented in this work makes no distinction between left and right feet. Step transitions between left foot and right foot contact are assumed to occur whenever the hip is descending and the hip passes through a critical height $h_{crit}$. The critical height strongly influences gait evolution. The critical height is set by assuming a leg angle (relative to the vertical) at the moment of heel strike; this control parameter is labeled $\delta_{hs}$.

$$h_{crit} = L_{ref} \cos \delta_{hs} \tag{3.6}$$

This equation assumes the leg is fully extended at the moment of heel strike.

Since only single stance (in which always only one leg contacts the ground) is considered in this thesis. $\delta_{hs}$ should equals to $\delta_0$.

$$\delta_{hs} = \delta_0 \tag{3.7}$$
At the moment when heel strike occurs, the heel contact point is updated as weight transitions to a new foot. An integer index $k$ is used to refer to each of these transitions. The heel contact point at the time of the transition is set by the following equation.

$$x_{hi}(k) = x_G + h_{crit} \left[ \begin{array}{c} \tan \delta_{hc} \\ -1 \end{array} \right]$$  \hspace{1cm} (3.8)

It is assumed that the odd indices $k$ refer to foot placement involving the foot on which the OF sensor is placed. Even indices correspond to the opposite foot.

This model starts and ends when *heel strike* of one chosen leg (the leg on which an OF sensor is attached) happens. The distance along the horizontal axis is the stride length $L_{stride}$.

$$\begin{bmatrix} L_{stride} \\ 0 \end{bmatrix} = x_G (2k + 1) - x_G (2k - 1)$$  \hspace{1cm} (3.9)

For pedestrian navigation, $L_{stride}$ is of my most consideration. In order to get $L_{strides}$, initial conditions such as $\delta_0$ and initial velocity of hip $V_0$ should be given.

### 3.2.2 Kinematic Model

Because SLIP captures only the dynamics of the lumped mass at the hip, it does not explicitly predict leg motion, as might a more detailed passive walker model [28]. As such, leg kinematics must be inferred from the hip states computed by the SLIP model. The notion of algebraically inferring the states of a complex model (also called an anchor) from a simple dynamic model (also called a template) has been studied extensively in the fields of comparative biology and biomimetic robotics [29].
In the implementation, the leg model consists of three rigid links, an upper leg (thigh), a lower leg (shank), and a foot. All three segments are illustrated in Fig. 5. The thigh extends a length $L_{th}$ from the hip $G$ to the knee $K$. The shank extends a length $L_{sh}$ from the knee $K$ to the heel $H$. The foot extends a length $L_{ft}$ from the heel $H$ to the toe $T$ (which might more accurately be labeled the ball of the foot). It is assumed that the heel remains on the ground as long as the hip to heel-contact-point distance $L$ does not exceed the combined length of the thigh and shank; when this limit is exceeded, the foot is assumed to rotate about the toe contact point $T$ to allow the hip to travel farther (and for $L$ to continue to increase).

Fig. 5 Leg Geometry
The focal point of the OF sensor (e.g., the video camera) is positioned along the shank at a distance $L_{cam}$ relative to the knee. It is assumed that the video camera is pointing along and rotates with the shank reference frame ($sh$ subscript). Reference frames attached to the thigh ($th$ subscript) and to the world frame ($w$ subscript) are also illustrated in Fig. 5. Unit vectors for each frame are indicated as the vectors $\hat{x}$ and $\hat{y}$, followed by appropriate subscripts to indicate their frame.

The position of the hip is known from simulation; so the position of the video camera can be computed if the coordinate system of the shank and thigh are known.

\[
x_C = x_G - L_{th}\hat{y}_{th} - L_{cam}\hat{y}_{sh}
\]  

(3.10)

In the same way, positions of heel $x_K$ and $x_H$ are:

\[
x_K = x_G - L_{th}\hat{y}_{th}
\]  

(3.11)

\[
x_H = x_G - L_{th}\hat{y}_{th} - L_{sh}\hat{y}_{sh}
\]  

(3.12)

The point on the ground at the center of the video camera optical axis is identified by the following equation, assuming the optical axis is aligned with ground. Here $D$ is distance from video camera focal point to the point $P$ where the camera optical axis intersects the ground plane.

\[
x_P = x_G - L_{th}\hat{y}_{th} - (L_{cam} + D)\hat{y}_{sh}
\]  

(3.13)

OF measurements are scaled by the distance $D$, which can be determined by dotting the above equation with the vertical unit vector $\hat{y}_w$ and assuming that the height of the ground plane is zero.
OF depends on the OF sensor’s velocity, which is:

\[ \mathbf{v}_c = \mathbf{v}_G + \omega_{th} \mathbf{t}_th + \omega_{sh} \mathbf{t}_sh \]  \hfill (3.15)

More specifically, OF measurements depend only on the component of this velocity perpendicular to the optical axis (e.g., perpendicular to \( \mathbf{t}_sh \)). Hence, the OF model will depend only on the component of velocity normal to the shank (e.g., on \( \mathbf{v}_c \cdot \mathbf{t}_sh \)).

In the equation above, \( \omega_{th} \) and \( \omega_{sh} \) are angular velocity of thigh and shank respectively. They depend on angles between \( \mathbf{t}_sh \) to \( \mathbf{t}_w \) and \( \mathbf{t}_th \) to \( \mathbf{t}_w \) respectively.

\[ \omega_{sh} = \frac{d}{dt} \left( \arccos \left( \mathbf{t}_sh \cdot \mathbf{t}_w \right) \right) \]  \hfill (3.16)

\[ \omega_{th} = \frac{d}{dt} \left( \arccos \left( \mathbf{t}_th \cdot \mathbf{t}_w \right) \right) \]  \hfill (3.17)

OF is also sensitive to shank rotation. The angular velocity, as well as the thigh and shank pointing vectors, must be treated separately for each of three configurations. Namely, different equations are needed to solve for the angular velocity and pointing vectors when (1) the heel contacts the ground (heel strike to toe off), (2) the toe contacts the ground (toe off to swing phase), and (3) the foot does not contact the ground (swing phase to the next heel strike of the same leg).

In the first configuration, when the heel is in contact with the ground, the thigh and shank rotation rates may be computed by solving the following equation for two angular
velocity terms, noting that the heel velocity $v_H$ is zero and that the hip velocity $v_G$ is known from the SLIP simulation.

$$v_H = 0 = v_G + \omega_{th} L_{th} \hat{y}_{th} + \omega_{sh} L_{sh} \hat{y}_{sh}$$  \hspace{1cm} (3.18)

In this configuration (heel in contact with the ground), the unit vectors $\hat{y}_{th}$ and $\hat{y}_{sh}$ are computed from the following equation.

$$\hat{y}_{H} = \hat{y}_{G} = -L_{th} \hat{y}_{th} - L_{sh} \hat{y}_{sh}$$  \hspace{1cm} (3.19)

Here $\hat{y}_{H}(k)$ is the current heel position, which is stationary during step $k$ (e.g., during stance phase). In order to compute the four unknown components of the two unit vectors in (3.12), it is necessary to invoke two additional constraint equations (namely, that the vectors are of unit length) in order to match the number of equations to the number of unknowns. The perpendicular unit vectors, $\hat{x}_{th}$ and $\hat{x}_{sh}$, can be obtained by a cross product of the computed $\hat{y}$ vectors with the vector pointing out of the plane.

In the second configuration, when only the toe touches the ground, (heel leaves the ground at this stage) angular velocities are computed with the following equation

$$v_T = 0 = v_G + \omega_{th} (L_{th} + L_{sh}) \hat{x}_{th} + \omega_{fl} L_{fl} \hat{y}_{sh}$$  \hspace{1cm} (3.20)

where it is assumed the thigh and shank rotate together such that their coordinate systems are aligned and their angular velocities are equal ($\omega_{th} = \omega_{sh}$). The unit vectors for each frame can be computed with a modified version of (3.12), where the contact point occurs at the toe:
\[ x_T(k) - x_G = -(L_{th} + L_{sh}) \hat{y}_{th} + L_{ft} \hat{x}_{sh} \]  

(3.21)

With

\[ x_T(k) = x_H(k) + L_{ft} x_w \]  

(3.22)

Additional assumptions must be made to compute the angular rotation rates during swing phase, when neither foot is on the ground, since the SLIP model provides no information about the leg during swing. In Fig. 6 Angle \( \theta \) is defined as the exterior angle between shank and line extended from thigh. \( \theta \) begins with \( \theta_{int} \), and will reach its maximum value \( \theta_{max} \) during the swing phase. Finally it ends up with zero in \( \theta_{end} \), meaning the completion of swing phase, which is also the end of one stride.
The initial angle when the leg hits swing is \( \theta_{\text{int}} \).

\[
\eta = \frac{\theta_{\text{max}}}{\theta_{\text{int}}} \tag{3.23}
\]

We introduce another control variable \( \eta \) representing the ratio between \( \theta_{\text{max}} \) and \( \theta_{\text{int}} \): 

\( \theta \) is a function of time, and is assumed to fit into a parabolic equation:

\[
\theta(t) = At^2 + Bt + C \tag{3.24}
\]
This equation can be solved by further assuming $\theta$ reaches $\theta_{\text{max}}$ in the middle of the swing of phase.

### 3.2.3 OF simulation

OF simulations are computed from projective geometry as described by [11]. The equation that simulates the OF $F_{\text{sim}}$ is:

$$F_{\text{sim}} = -f \left( \frac{v_c \cdot \dot{x}_{sh}}{D} + \omega_{sh} \right)$$ \hspace{1cm} (3.25)

This equation depends on the sensor focal length $f$ (unit in pixel), on the lateral sensor velocity $v_c \cdot \dot{x}_{sh}$, on the distance $D$ from the focal point to the ground plane, and on the angular velocity of the shank $\omega_{sh}$. $v_c$ is from equation (3.20); $D$ is from (3.14). $\omega_{sh}$ is different during each configuration.

### 3.3 Summary

This chapter specifically describes the SLIP model. The existing dynamic model is introduced to find the position of hip. For simplicity we only consider the single stance phase which means once one spring lifts off the ground the other one will instantaneously touch-down, entering stance phase. The dynamic equations governing CoM are specified.

What is new in this thesis is that a kinematic model to relate limb positions to center of gravity is developed. Both stance phase and swing phase are described within specific configurations. Position and velocity of knee and heel are calculated once those of hip are known.
OF simulation is derived based on algebraic information. The unit of OF simulation is also pixel/second, the same as OF measurement, since focal length is added. Ideally OF measurement and OF data are of the same magnitude, and they should have the same pattern during walking.
Chapter 4 Fusion Method

A preliminary demonstration was conducted to verify the concept for OF-based pedestrian navigation. Given the approximate nature of the SLIP model, the main goal of the verification study was to quantify how well the SLIP model describes bipedal walking and how accurately an estimator based on SLIP infers stride length for indoor navigation.

4.1 Background

Facing the complexity of human walking pattern, some researchers refer to adaptive Kalman Filters for recursive prediction. On foot IMU sensor is used, providing raw acceleration data. Several people were discouraged since Kalman linear Estimators work poorly for walking. States such as acceleration and velocity are weakly observable; the dynamic system for walking is unstable as the simulation errors aggravate rapidly. Thus, recursive Kalman Filters that try to introduce one measurement at a time is problematic, because weak observability results in large instantaneous errors over time, which grow more rapidly than the estimator can fix because the system dynamics are unstable. Some researchers, rather than recursively estimating step length using the Kalman Filter directly, they continuously estimate system parameters for each individual’s walking pattern, and then substitute these parameters into a step length estimation function along with step frequency and signal variance found by post digital signal processing. [30] This method, to some extent, increases the accuracy of estimation but requires additional works to predetermine walking system parameters at each step for each individual.
4.2 Method Description

Instead of implementing a recursive estimation method, the author proposed to apply a fusion method which infers states from many steps at once. (in this thesis, the author estimates two stride lengths, namely four step lengths together) In order to find states, this method involves two steps: (1) the definition of a cost function and (2) the minimization of the cost function. These states, which are system parameters in the walking model, are variables to determine stride length. Three major aspects of the fusion method are introduced in the following:

First, the accuracy of the fusion method does not highly depend on the walking model. For the recursive estimation states come directly from the walking model and measurements. These states can be problematic after a few steps because dynamics of a particular human walking model can be unstable, and errors accumulate rapidly as the walking displacement increases. For the fusion method states at one instance do not depend on previous states. States are estimated based on the minimization of the cost function. As a result, errors will not go into the next states estimated. Knowing the stride length estimation does not necessarily require a complex and highly accurate human walking model, the author chose a simple one in the literature such as SLIP to infer states.

Second, states are estimated iteratively so as to find the best match of walking model and real walking. Through continuously updating states, presumably the walking model will be closer to real walking. The differences between simulations from walking model and measurements from real walking are built into a cost function. Therefore the estimation
of states requires the determination and the minimization of the cost function; the states
optimized are thereafter applied into a stride length estimation function.

Third, stride length estimation function depends on states optimized. After finding states
they will be substituted into a stride length estimation function, and this function comes
from the walking model, SLIP. The reason why the author estimates the stride length in
this way is that this particular walking pattern from model is a proper simulation of real
walking, and states determine this model which should predict the stride length
approximately.

4.3 Mathematical Structure

The fusion method is similar to a state space control method. In the stride length
estimation problem, states \( x \) represents the position and velocity of center of gravity
located at hip; \( x_c(t) \) in (4.1) describes the motion of hip. Output from model \( y_{mod}(t) \) is OF
simulations \( \mathbf{F}_{\text{sim}}(t) \); \( y_{mod}(t) \) in (4.2) characterizes the motion of sensor.

\[
\begin{bmatrix}
\dot{x}_c(t) \\
\ddot{x}_c(t)
\end{bmatrix}
\]

(4.1)

\[
y_{mod}(t) = \mathbf{F}_{\text{sim}}(t)
\]

(4.2)

Control vector refers to initial conditions including initial velocities \( x_c(0) \), initial angle
\( \delta_0 \) and spring constant \( k \). These control variables are the same in determining SLIP
dynamics in section 3.2.1.

\[
\begin{bmatrix}
x_c(0) \\
\delta_0 \\
k
\end{bmatrix}
\]

(4.3)
States are calculated from SLIP kinematics as described by equation (3.1). By setting control vector SLIP dynamics are completely determined. The Dynamic equation in the form of state space format becomes:

\[
\dot{x}(t) = f(x(t); u)
\]  

(4.4)

The function \( f \) describes SLIP dynamics.

The outputs OF simulations are calculated from (3.25). (3.25) indicates \( F_{\text{sim}} \) depends on sensor velocity \( v_c \) in (3.15), angular velocity of shank \( \omega_{sh} \) in (3.16) and distance \( D \) in (3.14) from sensor to the targeted ground surface along the shank axis. These equations all indicate their dependence on position of hip \( x_G \). \( x_G \) are states and initiates at \( x_G(0) \) by setting initial angle \( \delta_0 \) and spring constant \( k \). Hence the observation equation comes from SLIP kinematics in section 3.2.2 and OF simulation in 3.2.3 presents:

\[
y_{\text{mod}}(t) = h(x(t); u)
\]  

(4.5)

In order to solve states and control vector, \( y_{\text{mod}}(t) \) is approximated by real measurement \( y(t) \), which is:

\[
y(t) = F_{\text{meas}}
\]  

(4.6)

Once states \( x \) and control vector \( u \) are solved the stride length is predicted based on the SLIP model. The \( k^{\text{th}} \) stride length \( L_{\text{stride}}(k) \) equals:

\[
L_{\text{stride}}(k) = g(x(t), u)
\]  

(4.7)
where function $g$ is the result of full simulation of SLIP through $k^{th}$ stride. $L_{\text{stride}}(k)$ is calculated base on (3.9).

The problem becomes how to solve states $x$ and control vector $u$ so as to calculate $L_{\text{stride}}(k)$. Here comes the fusion method.

Starting with a initial guess of control vector $u$, the fusion method tried to compare $y_{\text{mod}}(t)$ with $y(t)$ by continuously updating $u$. Finally an optimized control vector $u_{\text{opt}}$ is found that minimizing the differences between simulation and measurement within a certain time period. Thereafter $u$ is substituted into equation (4.4) to get states.

An overview the system configuration is in Fig. 7. In the methodology, stride length is estimated by an algorithm that compares sensor measurements to simulated data. OF sensor measurements are obtained by processing a series of video frames; OF simulations are generated based on the SLIP model. The post digital processing techniques are different (as indicated in the Fig. 7 that those two data processing blocks are different in color). Sum of Squared Differences (SSD) between OF simulations and OF measurements are calculated. Note in this thesis SSD is the cost function. An optimizer attempts to minimize the SSD by adjusting $u$. The optimizer belongs to an optimization routine that estimate states iteratively. When SSD converges to obtain the Least Sum of Squared Differences (LSSD), the corresponding stride length from SLIP is selected as the best estimate of the actual stride length. It is assumed that the SSD calculation is run on batch mode over an extended series of data, consisting of several strides (two strides in the following test). To conclude, the overall fusion method includes the determination of four stages: (1) OF measurements and OF simulations (discussed in Chapter 2 and
Chapter 3); (2) cost function; (3) optimization routine and (4) stride length estimation function \( g \). In the next sections the author will discuss (2) to (4) in detail.

4.2.1 Cost Function

Cost function in this thesis aims at quantification of differences between real walking and walking model. In particular, cost function is presumed to have its smaller value when measurements and simulations are closer. Measurements of real walking are OF measured from sensor (the video camera) called \( F_{\text{meas}} \); simulations is characterized by OF calculated from walking model (SLIP) called \( F_{\text{sim}} \). Therefore the cost function quantitatively presents the differences between \( F_{\text{meas}} \) and \( F_{\text{sim}} \).

The cost function in this thesis is the SSD between OF measurements \( F_{\text{meas}} \) from model and OF simulations \( F_{\text{sim}} \) from SLIP. The reason that the author applied SSD is because SSD is robust and straightforward indicating differences. It is widely used in digital signal processing. Cost function is defined as \( J_k \).

\[
J_k = \| y(t) - y_{\text{sim}}(t) \|_2 \approx \left( \sum_{k=0}^{K} (y(t_k) - y_{\text{sim}}(t_k))^2 \right)^{\frac{1}{2}} \tag{4.8}
\]
In (4.1), $J_k$ is the second norm of $F_{\text{meas}}$ and $F_{\text{sim}}$ within two strides; $\Delta t_k$ is the time duration for $k^{th}$ stride. Note two strides are estimated together by assuming these two are same in walking pattern and walking length.

$F_{\text{meas}}$ are function of time $t$ only. Raw data is collected by a video camera, and $F_{\text{meas}}$ are calculated for post signal processing using whether the interpolation method or correlation method. $F_{\text{sim}}$, however, is determined by SLIP, and SLIP is determined by initial conditions and system parameter. These serve as states in the cost function. For the SLIP model, it starts at the stage of heel strike at the end of the swing phase when one leg fully stretches and foot is about to land.

### 4.2.2 Optimization Routine

Note the optimization routine deals with a nonlinear system. It cannot ensure that a global minimum will be reached in a single run. Therefore the author intentionally selected 5 five initial guess of control vector $u$. The optimization routine fall into a typical problem solves the following:

\[
\begin{align*}
\min & J_i(u) ; \\
\text{s.t.} & \quad u_{\text{max},i} < u_i < u_{\text{min},i} \quad i = 1, 2, 3, 4, 5;
\end{align*}
\]

The solution is $u^*$ that minimize the cost function $J$

\[
\begin{align*}
(u^*) & = \arg \min(J^*) ; \\
J^* & = \min(J_i)
\end{align*}
\]

The optimization method the author used is a gradient-descent with fmincon in MATLAB. It performs a line search pattern within the search space. By given initial guess of those four states, it supposed to reach to a global minimum LSSD ideally.
However, human walking contains complex dynamics and even the SLIP kinematics and dynamics models include different stages such as stance phase and swing phase. It can be more complicated by adding double stance (this is not included in this thesis). The search space is non-convex; it is highly possible that fmincon finally reaches to a local minimum rather than a global one. In order to mitigate this problem, the author started at five distinct sets of initial guess, assuming optimized states are variables producing the least value after fmincon applied. This is a compromise made through the thesis. However it should produce certain accuracy.

4.2.3 Stride length estimation function

Once states given the best match SLIP, presumably the walking pattern at a time is identified. The pattern includes the trajectory of the hip, legs, feet and the sensor. So the navigation information such as stride length estimate \( L^* \) is function of those optimized control vector \( u^* \).

\[
L^* = g(u^*)
\] (4.11)

4.3 Summary

The fusion method belongs to a batch method which solves each stride at a time rather than estimates each stride based on previous stride. The advantage of the fusion method is that errors for one stride will not go into another stride, and this method does not require a sophisticated walking model.

The fusion method is composed of three basic parts: cost function, optimization routine and stride length estimation function. Cost function defines the differences between OF simulations and OF measurement; optimization routine seeks to find states and control vector; stride length estimation function infers stride length after states and control vector are reached.
This chapter also applies information from other chapters. OF measurements are calculated based on correlation method from Chapter 2; OF simulations are calculated based on SLIP dynamics and kinematics from Chapter 3.
Chapter 5 Test

In order to prove the concept of OF-based stride length estimation, tests are implemented both indoors and outdoors. Goals include (1) evaluating the model approximation (SLIP dynamics, kinematics, OF simulations), (2) assessing the quality of the processing used to obtain OF measurements. These are examined by comparing the final result, the stride length estimation to the ground truth. Note only one trial for the purpose of proof-of-concept is presented based on outdoor test.

5.1 Outdoor Test

The Outdoor test is also performed. Results are mainly discussed based on outdoor test. In outdoor area the vision is clearer and the light intensity is stronger in magnitude. This is important for determining OF simulations because it ensures good quality of frames.

As shown in Fig. 8 test is conducted in an open parking lot with a surface full of features. These features are small bulged rocks and fissures which are essential for computing OF. In order to reduce the blurring effect during collecting video frames, the author implement a high shuttle speed of 1/2000 second. Note for both indoor test and outdoor test, the ground truth is the same, 1.20 m per stride. Video is placed on the side of the shanks, pointed to the ground along the leg. The time duration is 9 seconds for each test.

The outdoor test is a preliminary test to verify the fusion method. Sources of errors are incurred in the process of tests. For example, the ground truth approximately equals to 1.20 m because the test engineer can not exactly follow the marker on the ground. Meanwhile the video camera is not entirely fixed so that the OF measurement might not be correctly predicted. These errors are system errors; it can be reduced by implementing a more comprehensive test. This belongs to the future. However the test itself still works
in presenting the following: (1) patterns of OF measurements and OF simulation; (2) stride length estimation.

5.2 Results

5.2.1 Parameters

Physical parameters of a test engineer. Parameters are summarized in Table 1 below.

These parameters are from one test engineering measurements. Length parameters were measured with a ruler; the mass parameter was measured with a scale. Thigh length $L_{th}$ is the distance from knee to heel; Shank Length $L_{sh}$ is the distance from hip to knee; The lump mass the total body mass $m$. These parameters are further adjusted based on study of human body from [31].
Table 1 Physical Parameters for the SLIP Model

<table>
<thead>
<tr>
<th>Body Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Thigh Length $L_{th}$</td>
<td>0.47±0.02 m</td>
</tr>
<tr>
<td>Shank Length $L_{sh}$</td>
<td>0.38±0.02 m</td>
</tr>
<tr>
<td>Foot Length $L_{ft}$</td>
<td>0.2±0.01 m</td>
</tr>
<tr>
<td>Body Mass $m$</td>
<td>50±2 kg</td>
</tr>
</tbody>
</table>

In the content of determination of cost function $OF_{meas}$ and $OF_{sim}$ are calculated separately. $OF_{meas}$ comes from post processing of a series of video frames within 9 seconds, six strides; $OF_{sim}$ comes from continuously adjusting control vector $u$ including initial conditions $v_{x_{int}}$ (m/s), $v_{y_{int}}$ (m/s), $\delta_{int}$ (°) and system parameter $K$ (kN/m).

The optimization routine involves start guess of states for SLIP to begin with. Note the author chose five sets of initial guess described in the following table:

Table 2 Initial Guess of States for SLIP

<table>
<thead>
<tr>
<th>Initial Guess of States</th>
<th>$u$</th>
</tr>
</thead>
<tbody>
<tr>
<td>First Guess of States</td>
<td>(1.6, -0.5, 10, 1.7)</td>
</tr>
<tr>
<td>Second Guess of States</td>
<td>(1.5, -0.5, 10, 1.6)</td>
</tr>
<tr>
<td>Third Guess of States</td>
<td>(1.6, -0.5, 10, 1.5)</td>
</tr>
<tr>
<td>Fourth Guess of States</td>
<td>(1.4, -0.5, 10, 1.7)</td>
</tr>
<tr>
<td>Fifth Guess of States</td>
<td>(1.2, -0.5, 10, 1.8)</td>
</tr>
</tbody>
</table>

Note the initial guess for $\delta_{int}$ and $K$ are 10° and 1.8 kN/m respectively. This is because to start with the initial angle can be approximately observed; the change of $y_{int}$ does not that much affect the stride length as $x_{int}$ and $K$ do. For simplicity, MATLAB optimization only concerns with two states $x_{int}$ and $K$. 

35
Finally, the stride length estimation is for two strides at a time. Once optimized states are confirmed, they will be re-substituted into SLIP again to predict stride length. The equation is (4.11).

5.2.2 Analysis

Correlation method was used for the outdoor test. A repeatable walking pattern (see Fig. 9) is observed up to the end of the test period. OF ranges from -1000 to 6000 pixel/second. Each stride is about 1.1 second in time. During the stance phase, OF is small; during the swing phase, OF jumps up to approximately 5000 pixel/second. The pattern of OF simulations and OF measurements during the swing phase will largely affect the final stride length estimation.

Fig. 9 Optical Flow measurements over 9 seconds, Correlation-Based Method
Fig. 10 is the zooming of the first two strides for both measurement and simulation. One good thing about measurements and simulations is that they both predict the same pattern of walking. They both have small values in the *stance phase* and large values in the *swing phase*. Two distinct differences is the match of magnitude and time scale for each stride. This is observed by looking at frames at each time directly. The maximum OF measurement in the first stride is 5410 pixel/second while the maximum OF simulation is 3131 pixel/second.

Fig. 11 describes the distribution of *stance phase* and *swing phase* in time. For OF simulations, the stride time durations are both 0.9374. The proportion is 67 to 33. This proportion is from [31]. For OF measurements, the first stride cost 1.2679 second; the
second stride cost 0.9009 second. The proportion for the both stride are approximately 33 to 67.

![Diagram showing proportion of stance phase to swing phase of measurements and simulations.](image)

**Fig. 11** Proportion of Stance Phase to Swing Phase of measurements and simulations

Discussion of possible error sources are in the following:

First, OF measurements are higher than OF simulations during the *swing phase*. This might be because in the SLIP dynamics, there are no specific equations determining the feet and legs motion when they are in the air.

Second, the time duration of each stride is different. This is might be caused by performing unnatural gait. The test engineer is required to follow markers on the ground so the stride length is fixed. This would force him to focus on aiming each marker during
swing phase so that more time is possible spending on this phase. However, according to [31] stance phase should be longer because of double stance, when both legs are on the ground. This different is caused in this trial.

However, the estimation of stride length is good start. For this trial, the true value of the stride length was 1.20 m. The stride length estimated using the SLIP model was 1.08 cm. Thus, the estimation error was 10.0%. Results are concluded in Table 3.

<table>
<thead>
<tr>
<th>Table 3 Stride Length Estimation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ground Truth</td>
</tr>
<tr>
<td>Stride length Estimation</td>
</tr>
<tr>
<td>Relative Error</td>
</tr>
</tbody>
</table>

5.2.3 Discussion

For the purpose of proof-of-concept, outdoor test based on correlation method is performed. By implementing this fusion method comparing OF measurements and OF simulation, the stride length estimation for each case has it relative error up to 10.0%, which is higher than expect but it is still a good start. These preliminary results suggest that OF sensors may provide useful information to support pedestrian navigation; however, the specific motion models used in this paper may not be accurate enough to support high-precision navigation. The author believes further research focusing on the double stance phase will lower the estimation errors.
5.3 Summary

This chapter specifically describes the stride length estimation based on the fusion method: minimizing SSD of OF measurements for an OF sensor (video camera in this thesis) and OF simulations from SLIP by tuning initial conditions (initial velocities and initial angle and spring constant). Both indoor tests and outdoor tests are conducted and results based on one trial for outdoor test is detailed discussed and results are listed. Errors are higher than expected but it is not that terrible. One conclusion might be that a higher precision walking model, such as a passive walking model is needed. However, the author hypothesizes that it is also possible to improve our results significantly by refining our SLIP model. In particular, current implementation of the SLIP model does not explicitly model double stance, which occurs when both feet are simultaneously on the ground. During the double stance phase, it would be most accurate to model the leg forces using two springs (rather than one as shown in Fig. 4). A refined kinematics model relating hip motion to shank motion might further improve the stride length estimation. To conclude, this fusion method is workable predicting stride length for navigation but it requires future work refining SLIP model such as the inclusion of double stance.
Chapter 6 Conclusion

5.1 Thesis Contribution

This thesis focuses on indoor navigation based on stride length estimation of human walking. The fusion method, which combines OF measurements from the OF sensor (in this thesis a video camera is used to collect video frames and OF are calculated off-line) with OF simulations from the SLIP model, is introduced to estimate stride length during walking. Currently the errors are noticeable but the author believes improvement can be reached by further refining the SLIP model. The contributions are summarized below.

Introduction of a novel method to estimate stride length by fusing OF information from measurement and SLIP. In this thesis, the author proposed to use an OF sensor for indoor pedestrian navigation. This is a problem not addressed previously in the research literature. Specifically I proposed a method to use a batch estimator (Chapter 4) to fuse correlation-based sensing (Chapter 2) with optical flow model obtain from a SLIP dynamic and kinematic simulation (Chapter 3). Two important aspects of this work are:

a. New processing (algebraic and kinematic equations) introduced to make batch solution possible. SLIP, a simple walking model, is implemented to derive position and velocity of the hip; algebraic equations, derived by the author based on SLIP, are presented to relate positioning information of the limbs and the sensor attached to the leg, to that of the hip. Position and velocity of the sensor are two variables to determine OF simulations.
b. **Proof-of-concept trial demonstrated to verify the fusion method.** An outdoor test is performed to compare the stride length estimation from the fusion method with the ground truth. Relative error is 10% for one trial.

### 5.2 Impact and Future Work

The work described in this thesis opens a novel way for indoor navigation. The author proposes to use an OF sensor to navigate a pedestrian indoors. One of the benefits of OF-based navigation is that it does not require other infrastructures preinstalled. The author also believes OF sensor based navigation is optimal for SWAP constrained application and is possible to have smaller cumulative errors compared with IMU-based navigation. Unlike IMU, OF sensor is not been detailed studied for indoor navigation. After proposed by the author, more emphasis could be put on OF-based indoor navigation.

The fusion method in this thesis describes one of the OF-based navigation strategies. It aims at estimating displacement using only one OF sensor. This is achieved by adding addition information from a simple walking model SLIP. Now it shows 10% of discrepancy between estimation and ground truth. By applying the fusion method the author opens a way to relate OF to pedestrian displacement, and I believe OF can be better translated into navigation information such as using better walking model or other estimation methods.

For the future work, the main task is to improve estimation accuracy. It is possible to achieve this by reducing modeling errors or using better walking model. SLIP is a simple walking model, and the author establishes the kinematic equations based on SLIP so as to relate hip to limbs and sensor. It is a good start for the fusion method but it could contain modeling errors, which will greatly affect the final estimation. The author believes better refinement on SLIP is highly possible to relief modeling errors. This could be done by
including *double stance* to SLIP, and better understanding of *swing phase* is of the same importance. Indeed, using a more complex walking model is another alternative.

Another suggestion is to do it many times, both indoors and outdoors. Now the author only claims 10% of accuracy based on one trial; more tests are necessary. It should be done many times on several testers in different environment. Meanwhile it would be better to add additional video cameras to capture the mode of walking from other views. Better fixing devices will be preferred for each test.
References


Appendix A: SLIP Model Simulation (Dynamics and Kinematics)

%%% direction function
function [lookfor stop direction]=slip_ph1(t,x)

global Lref
load ksi

lookfor=min(x(3)-(Lref*cos(ksi)+1e-9),x(3));
stop=1;
direction=1;

function [lookfor stop direction]=slip_ph2(t,x)

global Lref Xft
load ksi

lookfor=min(x(3)-(Lref*cos(ksi)+1e-9),x(3));
stop=1;
direction=-1;

%%% Dynamic constraint function
function dx=slip_M(t,x)

global Xft Lref m
load ksi load k

delx=Xft-x(1);
dely=x(3);

L=sqrt(delx^2+dely^2);

if L<Lref
    F=(k*(Lref-L));
else
    F=0;
end

dx=zeros(4,1);
dx(1)=x(2);
dx(2)=-1/m*(F*delx/L);
dx(3)=x(4);
dx(4)=1/m*(F*dely/L)-9.81;

% Update accumulated travel distance
%% main function
global Lref Xft m

Lref=1.1; % Nominal leg length (m)
Lmax=1.1; % Maximum leg extension (m)
Lft=0.2; % Feet length (m)
Lth=0.47; % Shank length (m)
Lsh=0.38; % Thigh length (m)
r=0.3175; % Distance from sensor to ground in sensor plane (m)
ksi=12/57.3; % Initial angle (rad)
save ksi ksi
% (ksi can be changed) state 1
k=18000; % Nominal leg stiffness (N/ma)
save k k
% (k can be changed) state 2
yt=1.06 % Minimum height (m)
% (yt can be changed) state 3
m=50; % Body mass (kg)
Cang=1.2; % Ratio of max angle to initial angle in swing phase for shank (1)
% (Cang can be changed) state 4

tin1=0; % Phase 1
Nth2=[0 1];
Xint=[0,1.5,Lref*cos(ksi),-1]; % Initial condition (x,xdot,y,ydot)
Xint1=Xint;
% (xdot,ydot can be changed) state 5 and state 6
TT=[];
XX=[];

for numsp=1:2
    Xft=Xint1(end,1)+Lref*sin(ksi) % Heal strike distance from origin
    XXft(numsp)=Xft;
    Xint1(2)=Xint(2); Xint1(4)=Xint(4);
    options1=odeset('Events',@slip_ph1); % Phase 1
    [th1,Xh1,te1,xe1,ie1]=ode45(@slip_M,[tin1,tin1+10],Xint1,options1);
    Xint2=Xh1(end,:);
tin2=th1(end);
Nph1(numsp)=numel(th1);
Nth1(1)=Nph2(2);
Nth1(2)=numel(th1)+Nth1(1);
TT(Nth1(1):Nth1(2)-1)=th1;
XX([Nth1(1):Nth1(2)-1],[1:4])=Xh1;

options2=odeset('Events',@slip_ph2);    % phase 2
[th2,Xh2,te2,xe2,ie2]=ode45(@slip_M,[tin2,tin2+10],Xint2,options2);
Xint1=Xh2(end,:);
tin1=th2(end);
Nth2(1)=Nph1(2);
Nth2(2)=numel(th2)+Nth2(1);
TT(Nth2(1):Nth2(2)-1)=th2;
XX([Nth2(1):Nth2(2)-1],[1:4])=Xh2;

for i=1:numel(th1)            % phase 1 plot
  X1=[Xft Xh1(i,1)];
  Y1=[0 Xh1(i,3)];
  plot(X1,Y1)
  axis equal
  pause(0.1)
  hold on
  if Xh1(i,2)<0  % fall
    break
  end
end
clear i Xh1 th1

for j=1:numel(th2)            % phase 2 plot
  X2=[Xft Xh2(j,1)];
  Y2=[0 Xh2(j,3)];
  plot(X2,Y2)
  axis equal
  pause(0.1)
  hold on
  if Xh2(j,2)<0  % fall
    break
  end
end
clear j Xh2 th2
end
Te=[TT(1:Nph1(1)) TT(Nph1(1)+2:Nth1(1)-1) TT(Nth1(1)+1:Nth2(1)-1) TT(Nth2(1)+1:end)];
clear TT
TT=Te;

Xe=[XX(1:Nph1(1),:); XX(Nph1(1)+2:Nth1(1)-1,:); XX(Nth1(1)+1:Nth2(1)-1,:); XX(Nth2(1)+1:end,:)];
clear XX
XX=Xe;

%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%   kinematic SLIP model simulation
N=numsp/2;
Nsp=2*N-1;

for z=1:numel(TT)
    if TT(z)>TT(end)*0.6
        break
    end
end
N_TT=z;

Nst=Nph1(1);
Ntr=N_TT;
if (Nst+1)>(Ntr-1),
    return
else
end

figure
Nst=Nph1(1);
Ntr=N_TT;

for j=1:Nst
    L=sqrt(XX(j,3)^2+(XX(j,1)-XXft(1))^2);
    ksi1(j)=asin((XXft(1)-XX(j,1))/L);
    tao1(j)=real(acos((L^2+Lsh^2-Lth^2)/(2*L*Lsh)));
    tao11(j)=pi/2+ksi1(j)-tao1(j);
    Xkn1(j)=XXft(1)+Lsh*cos(tao11(j));
    Ykn1(j)=Lsh*sin(tao11(j));
end

for j=1:Nst-1;
    om1(j)=(tao11(j+1)-tao11(j))/(TT(j+1)-TT(j));
end
clear j

for k=1:Nst
Xs=[XXft(1) Xkn1(k) XX(k,1)];  
Ys=[0 Ykn1(k) XX(k,3)];  
plot(Xs,Ys)  
axis equal  
pause(0.1)  
hold on  
end  
clear k  

%% Stance phase, transition  
for j=Nst+1:Ntr  
XXft1=XXft(1)+Lft;  
L=sqrt(XX(j,3)^2+(XX(j,1)-XXft1)^2);  
Lsh2=sqrt(Lft^2+Lsh^2);  
ksi2(j)=asin(-(XXft1-XX(j,1))/L);  
tao2(j)=real(acos((L^2+Lsh2^2-Lth^2)/2/L/Lsh2));  
tao22(j)=pi/2-ksi2(j)-tao2(j);  
Xkn2(j)=XXft1+Lsh2*cos(tao22(j));  
Ykn2(j)=Lsh2*sin(tao22(j));  
beta=acos(Lft/Lsh2);  
alpha(j)=pi-beta-tao22(j);  
Xhl(j)=XXft1-Lft*cos(alpha(j));  
Yhl(j)=Lft*sin(alpha(j));  
end  
for j=Nst+1:Ntr-1  
om2(j)=(alpha(j+1)-alpha(j))/(TT(j+1)-TT(j));  
end  
clear j  
for k=Nst+1:Ntr  
X1=[XXft1 Xhl(k) Xkn2(k) XX(k,1)];  
Y1=[0 Yhl(k) Ykn2(k) XX(k,3)];  
plot(X1,Y1,'-r')  
axis equal  
pause(0.1)  
hold on  
end  
clear k  
Feet=[Xhl(end);Yhl(end)];  
Knee=[Xkn2(end);Ykn2(end)];  

%% Swing phase  
Ns_int=N_TT;  % beginning time of swing phase  
Ns_end=numel(TT);  % ending time of swing phase
L1=Lref-Lsh;
L2=Lsh-r;

Ang_int1=atan(real((Knee(1)-XX(Ns_int,1)))/real((XX(Ns_int,3)-Knee(2))));
Ang_end1=ksi;

omega1=(Ang_end1-Ang_int1)/(TT(Ns_end)-TT(Ns_int));

for p=Ns_int+1:Ns_end
    Stheta1(p)=omega1*(TT(p)-TT(Ns_int))+Ang_int1;
    SXknee(p)=Lth*sin(Stheta1(p))+XX(p,1);
    SYknee(p)=-Lth*cos(Stheta1(p))+XX(p,3);
end

Ang_int2=atan(real((Feet(1)-Knee(1)))/real(Knee(2)-Feet(2)));
Ang_end2=ksi;
Max_Ang=Cang*Ang_int2;
Ns_mid=ceil((Ns_end+Ns_int)/2);
Coeff=inv([TT(Ns_int)^2 TT(Ns_int) 1;
            TT(Ns_mid)^2 TT(Ns_mid) 1;
            TT(Ns_end)^2 TT(Ns_end) 1])*[Ang_int2;Max_Ang;Ang_end2];
C1=Coeff(1); C2=Coeff(2); C3=Coeff(3);

Stheta2(Ns_int)=Ang_int2;
for y=Ns_int+1:Ns_end
    Stheta2(y)=C1*TT(y)^2+C2*TT(y)+C3;
end

clear y
for y=Ns_int+1:Ns_end-1
    omega2(y)=(Stheta2(y+1)-Stheta2(y))/(TT(y+1)-TT(y));
end

for q=Ns_int+1:Ns_end
    SXheel(q)=Lsh*sin(Stheta2(q))+SXknee(q);
    SYheel(q)=-Lsh*cos(Stheta2(q))+SYknee(q);
    Ds(q)=(XX(q,3)-L1*cos(Stheta1(q))-L2*cos(Stheta2(q)))/(cos(Stheta1(q)+Stheta2(q)));
end

clear q

for num=Ns_int+1:Ns_end
    if SYheel(num)<0
        Lsd=0; SSD=1e+10;
        return
    end
end

for q=Ns_int+1:Ns_end-1
    theta4(q)=Stheta1(q)-Stheta2(q);
    v0x(q)=(XX(q+1,1)-XX(q,1))/(TT(q+1)-TT(q));
\[ v_0y(q) = \frac{XX(q+1,3) - XX(q,3)}{TT(q+1) - TT(q)}; \]

\[ \theta_5(q) = \theta_1(q) + \theta_4(q); \]

\[ V_1(q) = -\cos(\theta_5(q))v_0x(q) - \sin(\theta_5(q))v_0y(q) - \]

\[ \cos(\theta_4(q))L_1 + \cos(\theta_5(q))v_0x(q) - \cos(\theta_5(q))v_0y(q) - \]

\[ \sin(\theta_4(q))L_2*(\omega_1 + \omega_2(q)); \]

\[ V_2(q) = -\sin(\theta_5(q))v_0x(q) - \cos(\theta_5(q))v_0y(q) + \sin(\theta_4(q))L_1*(\omega_1 + \omega_2(q)); \]

\[ \%
\]

\[ for\ c=N_s\ int+1:N_s\ end\]

\[ SX = [XX(c,1) SXknee(c) SXheel(c)]; \]

\[ SY = [XX(c,3) SYknee(c) SYheel(c)]; \]

\[ plot(SX,SY,'-k'); \]

\[ axis\ equal; \]

\[ pause(0.1); \]

\[ hold\ on; \]

\[ end; \]

\[ clear; \]
Appendix B: OF measurements, Correlation-Based Calculation

```matlab
%% Correlation-Based Calculation
mov=aviread('OF_raw');
fn=num=size(mov,2);

gaussianFilter = fspecial('gaussian');
for i=1:fnum
    frame(:,:,i)=mov(i).cdata;
    Frame(:,:,i)=rgb2gray(frame(:,:,i));
    Frame(:,:,i)=imfilter(Frame(:,:,i), gaussianFilter, 'symmetric', 'conv');
end
F=double(Frame);
time=[0:1/29.97:1/29.97*(fnum-2)];

%% patch in the middle 40*40 pixel^2
row=ones(1,20)*40; col=[40-9:40+10];
for z=1:fnum-1
    for x=1:3
        for y=1:3
            Zone(:,:,z)=F([141+(x-2)*40:340+(x-2)*40],[221+(y-2)*40:500+(y-2)*40],z+1);
            Patch(:,:,z)=F([221+(x-2)*40:260+(x-2)*40],[341+(y-2)*40:380+(y-2)*40],z);
            Patch_m(:,:,z)=mean(mean(Patch(:,:,z)));
            Zone_m(:,:,z)=mean(mean(Zone(:,:,z)));
            Patch_d(:,:,z)=Patch(:,:,z)-Patch_m(:,:,z);
            Zone_d(:,:,z)=Zone(:,:,z)-Zone_m(:,:,z);
            Cor(:,:,z)=normxcorr2(Patch_d(:,:,z),Zone_d(:,:,z));
            Cor1(:,:,z)=Cor([41:200],[41:280],z);
            MaxC(x,y,z)=max(max((Cor1(:,:,z))));
            [a(x,y,z),b(x,y,z)]=find(Cor1(:,:,z)==MaxC(x,y,z));
        end
    end
end
for x=1:3
    for y=1:3
        for z=1:fnum-1
            if MaxC(x,y,z)<0.5;
                MaxC(x,y,z)=0.01;
            end
        end
    end
end
for z=1:fnum-1
    A(z)=sum(sum(a(:,:,z).*MaxC(:,:,z)))./sum(sum(MaxC(:,:,z)));
    B(z)=sum(sum(b(:,:,z).*MaxC(:,:,z)))./sum(sum(MaxC(:,:,z)));
    examine(z)=max(max((MaxC(:,:,z))));
end
```

end

plot(time,examine)

[row, col] = size(Cor1(:, :, 1));
row = row / 2; col = col / 2;

for z = 1: fnum - 1
    ofx(z) = col - B(z);
    ofyy(z) = row - A(z);
end
figure
plot(time, -29.97*x)
Appendix C: the fusion Method, SLIP Model Based Estimation Function

% SSD between OF simulations and OF measurement
function SSD=OF_SSD(fx)

global Lref Xft m CTe COFe

Lref=1.1;          % leg length (m)
Lft=0.2;           % Foot length (m)
Lth=0.47;           % shank length (m)
Lsh=0.38;           % thigh length (m)
r=0.32;            % distance from sensor to ground along optical axis (m)
fl=1340;           % focal length
ksi=fx(1)/57.3;       % Initial angle (rad)
save ksi ksi
% (ksi can be changed) state 1
k=18000;           % Nominal leg stiffness (N/ma)
save k k
% (k can be changed) state 2
yt=Lref*cos(ksi)-0.02;            % minimum height (m)
% (yt can be changed) state 3
m=50;              % Body mass (kg)
Cang=1.2;          % ratio of max angle to initial angle in swing phase for shank (1)
% (Cang can be changed) state 4
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
tin1=0;
Nth2=[0 1];

Xint=[0,fx(2),Lref*cos(ksi),0];    % Initial condition (x,xdot,y,ydot)
Xint1=Xint;
% (xdot,ydot can be changed) state 5 and state 6

TT=[];
XX=[];

for numsp=1:2

    Xft=Xint1(end,1)+Lref*sin(ksi);   % Heal strike distance from origin
    XXft(numsp)=Xft;
    Xint1(2)=Xint(2); Xint1(4)=Xint(4);

    options1=odeset('Events',@slip_ph1);    % phase 1
    [th1,Xh1,te1,xe1,ie1]=ode45(@slip_M,[tin1,tin1+2],Xint1,options1);
    Xint2=Xh1(end,:);
    tin2=th1(end);

end
Nph1(numsp)=numel(th1);
Nth1(1)=Nth2(2);
Nth1(2)=numel(th1)+Nth1(1);

TT(Nth1(1):Nth1(2)-1)=th1;
XX([Nth1(1):Nth1(2)-1],[1:4])=Xh1;

options2=odeset('Events',@slip_ph2); % phase 2
[th2,Xh2,te2,xe2,ie2]=ode45(@slip_M,[tin2,tin2+2],Xint2,options2);
Xint1=Xh2(end,:);
tin1=th2(end);
Nth2(1)=Nth1(2);
Nth2(2)=numel(th2)+Nth2(1);

TT(Nth2(1):Nth2(2)-1)=th2;
XX([Nth2(1):Nth2(2)-1],[1:4])=Xh2;

clear j Xh2 th2

end

Te=[TT(1:Nph1(1)) TT(Nph1(1)+2:Nth1(1)-1) TT(Nth1(1)+1:Nth2(1)-1)
TT(Nth2(1)+1:end) ];
clear TT
TT=Te;

Xe=[XX(1:Nph1(1),:); XX(Nph1(1)+2:Nth1(1)-1,:); XX(Nth1(1)+1:Nth2(1)-1,:);
XX(Nth2(1)+1:end,:)];
clear XX
XX=Xe;

% slip walking model simulation
N=numsp/2;
Nsp=2*N-1;

for z=1:numel(TT)
    if TT(z)>TT(end)*0.6
        break
    end
end
N_TT=z;

% for z=1:numel(TT)
%    if (XX(z,3)<yt&(XX(z,1)-XXft(Nsp))>=(Lref*sin(ksi)+0.0005))
%        break
%    end
%end
%z=Nth1(1);
N_TT=z;

%%%%%%%%%%%%%%%%%%%%%%%%
Nst=Nph1(1);
Ntr=N_TT;

if (Nst+1)==(Ntr-1),
  return
else
end

%%%%%%%%%%%%%%%%%%%%%%%%
Nst=Nph1(1);
Ntr=N_TT;

for j=1:Nst
  L=sqrt(XX(j,3)^2+(XX(j,1)-XXft(1))^2);
  ksi1(j)=asin((XX(j,1)-XXft(1))/L);
  tao1(j)=real(acos((L^2+Lsh^2-Lth^2)/(2*L*Lsh)));
  tao11(j)=pi/2+ksi1(j)-tao1(j);

  Xkn1(j)=XXft(1)+Lsh*cos(tao11(j));
  Ykn1(j)=Lsh*sin(tao11(j));
end

for j=1:Nst-1;
  om1(j)=(tao11(j+1)-tao11(j))/(TT(j+1)-TT(j));
end
OF1=-om1;

clear j

%% Stance phase, transition
for j=Nst+1:Ntr
  XXft1=XXft(1)+Lft;
  L=sqrt(XX(j,3)^2+(XX(j,1)-XXft(1))^2);
  Lsh2=sqrt(Lft^2+Lsh^2);
  ksi2(j)=asin(-(XXft1-XX(j,1))/L);
  tao2(j)=acos((L^2+Lsh2^2-Lth^2)/2/L/Lsh2);
  tao22(j)=pi/2-ksi2(j)-tao2(j);

  Xkn2(j)=XXft1+Lsh2*cos(tao22(j));
  Ykn2(j)=Lsh2*sin(tao22(j));

  beta(j)=acos((Lft^2+Lsh2^2-Lsh^2)/(2*Lft*Lsh2));
  %beta=acos(Lft/Lsh2);
  alpha(j)=pi-beta(j)-tao22(j);

  Xhl(j)=XXft1-Lft*cos(alpha(j));
  Yhl(j)=Lft*sin(alpha(j));

end
end

for j=Nst+1:Ntr-1
    \[\text{om2}(j) = \frac{\alpha(j+1) - \alpha(j)}{T_T(j+1) - T_T(j)};\]
    \[\text{OF2}(j) = \frac{(L_f \tan(\alpha(j)) + r) \cdot r \cdot \text{om2}(j) + L_f \cdot r \cdot \text{om2}(j)^2 \cdot (T_T(j+1) - T_T(j))}{(L_f \tan(\alpha(j)) + r)^2};\]
end
clear j

Feet=[Xhl(end);Yhl(end)];
Knee=[Xkn2(end);Ykn2(end)];

%%% Swing phase
Ns_int=N_TT;  % beginning time of swing phase
Ns_end=numel(TT); % ending time of swing phase

L1=Lref-Lsh;
L2=Lsh-r;

Ang_int1=atan(real((Knee(1)-XX(Ns_int,1)))/real((XX(Ns_int,3)-Knee(2))));
Ang_end1=ksi;

\[\omega_1 = \frac{\text{Ang_end1} - \text{Ang_int1}}{T_T(Ns_end) - T_T(Ns_int)};\]

for p=Ns_int+1:Ns_end
    Stheta1(p)=\[\omega_1 \times (T_T(p) - T_T(Ns_int)) + \text{Ang_int1};\]
    SXknee(p)=Lth*sin(Stheta1(p))+XX(p,1);
    SYknee(p)=-Lth*cos(Stheta1(p))+XX(p,3);
end

Ang_int2=atan(real((Feet(1)-Knee(1)))/real(Knee(2)-Feet(2)));
Ang_end2=ksi;
Max_Ang=Cang*Ang_int2;

kk=Ns_int;
while TT(kk)<1*(TT(Ns_end)-TT(Ns_int))/4+TT(Ns_int)
    kk=kk+1;
end
Ns_mid=kk; % Ns_mid=ceil((Ns_end+Ns_int)/2);

Coeff=inv([TT(Ns_int)^2 TT(Ns_int) 1;\nTT(Ns_mid)^2 TT(Ns_mid) 1;\nTT(Ns_end)^2 TT(Ns_end) 1])*[Ang_int2;Max_Ang;Ang_end2];
C1=Coeff(1); C2=Coeff(2); C3=Coeff(3);

Stheta2(Ns_int)=Ang_int2;
for y=Ns_int+1:Ns_end
    Stheta2(y)=C1*TT(y)^2+C2*TT(y)+C3;
end
clear y
for y=Ns_int+1:Ns_end-1
    omega2(y)=(Stheta2(y+1)-Stheta2(y))/(TT(y+1)-TT(y));
end
%omega2=(Ang_end2-Ang_int2)/(TT(Ns_end)-TT(Ns_int));

for q=Ns_int+1:Ns_end
    SXheel(q)=Lsh*sin(Stheta2(q))+SXknee(q);
    SYheel(q)=-Lsh*cos(Stheta2(q))+SYknee(q);
    Ds(q)=(XX(q,3)-L1*cos(Stheta1(q))-L2*cos(Stheta2(q)))/(cos(Stheta1(q)+Stheta2(q)));
    Xs(q)=SXknee(q)-L2*(SXknee(q)-SXheel(q))/Lsh;
end

clear q
%%%%%%%%%%%%%%%%%%%%%%%%%%%
for num=Ns_int+1:Ns_end
    if SYheel(num)<-0.05
        Lsd=0; SSD=1e+10;
        return
    else
        end
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%
for q=Ns_int+1:Ns_end-1
    theta4(q)=Stheta1(q)-Stheta2(q);
    v0x(q)=(XX(q+1,1)-XX(q,1))/(TT(q+1)-TT(q));
    v0y(q)=(XX(q+1,3)-XX(q,3))/(TT(q+1)-TT(q));
    theta5(q)=Stheta1(q)+theta4(q);
    V1(q)=-(cos(theta5(q))*v0x(q)-sin(theta5(q))*v0y(q)-
    (cos(theta4(q))*L1+L2)*(omega1+omega2(q));
    V2(q)=sin(theta5(q))*v0x(q)-
    cos(theta5(q))*v0y(q)+sin(theta4(q))*L1*(omega1+omega2(q));%OF3(q)=cos(Stheta2(q))*(Xs(q+1)-Xs(q))/Ds(q)/(TT(q+1)-TT(q))+(omega1+omega2(q));
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%
for c=1:Nst-1
    OF(c)=OF1(c);
end
for c=Nst+1:Ntr-1
    OF(c)=OF2(c);
end
for c=Ntr+1:Ns_end-1
    OF(c)=OF3(c);
end
OF(Nst)=(OF(Nst-1)+OF(Nst+1))/2; OF(Ntr)=(OF(Ntr-1)+OF(Nst+1))/2;
OF(Ns_end)=(OF(11)+OF(Ns_end-1))/2;

%%%%%%%%%%%%%%%%%%%%%%%%%%%
COF=[OF(11:Ns_end) OF(11:Ns_end) OF(11:Ns_end)];
COF=[COF COF];
%COF=fl*COF;

for cc=1:6
    BfTT(:,:,cc)=TT(:,11:Ns_end)+(cc-1)*TT(Ns_end);
end
CTT=[BfTT(:,:,1) BfTT(:,:,2) BfTT(:,:,3) BfTT(:,:,4) BfTT(:,:,5) BfTT(:,:,6)];

load CTe COFe   % OF measurements from test

buff=1;
cof(1)=COF(1);
for i=2:numel(CTe)
    j=buff;
    while CTe(i)>CTT(j)
        j=j+1;
    end
    buff=j;
    cof(i)=((CTe(i)-CTT(buff-1))*COF(buff)+(CTT(buff)-CTe(i))*COF(buff-1))/(CTT(buff)-CTT(buff-1));
end

SSD=sum((cof-COFe).^2);

% main function, optimization routine
% This is simplified by optimizing only two states, initial angle and initial horizontal
gain function OF_SSD
A=[1 0 ;
   -1 0 ;
   0 1 ;
   0 -1 ;]
B=[18;-12;2;-1.3];
x0=[18;1.5];

[x,fval]=fmincon(@FSSD2,x0,A,B)