2011

Improvement of measurement performance for self-mixing interferometry based displacement sensing system

Yuanlong Fan

University of Wollongong

Recommended Citation

NOTE

This online version of the thesis may have different page formatting and pagination from the paper copy held in the University of Wollongong Library.

UNIVERSITY OF WOLLONGONG

COPYRIGHT WARNING

You may print or download ONE copy of this document for the purpose of your own research or study. The University does not authorise you to copy, communicate or otherwise make available electronically to any other person any copyright material contained on this site. You are reminded of the following:

Copyright owners are entitled to take legal action against persons who infringe their copyright. A reproduction of material that is protected by copyright may be a copyright infringement. A court may impose penalties and award damages in relation to offences and infringements relating to copyright material. Higher penalties may apply, and higher damages may be awarded, for offences and infringements involving the conversion of material into digital or electronic form.
Improvement of Measurement Performance for Self-Mixing Interferometry based Displacement Sensing System

A thesis submitted in fulfillment of the requirements for award of the degree Master of Engineering by Research from UNIVERSITY OF WOLLONGONG

by Yuanlong Fan

School of Electrical, Computer and Telecommunications Engineering
March 2011
Dedicated to my family
Declaration

This is to certify the work reported in this thesis was done by the author, unless specified otherwise, and the no part of it has been submitted in a thesis to any other university or similar institution.

Signature:________________________

Yuanlong Fan

30 March 2011
Abstract

The core components of a Self-Mixing Interferometry (SMI) based displacement sensing system, consists of a Laser Diode (LD), a micro-lens and a moving target which forms an external cavity of the LD. Displacement information of the moving target is carried in the laser power emitted by the LD. The laser power is called a Self-Mixing Signal (SMS). The existing methods for extracting displacement from a SMS are not perfect, e.g. (or assuming a SMI with a constant optical feedback level, or not fully considering all SMS waveforms). By studying the features of SMSs at different cases, the thesis proposes an improved algorithm used for displacement reconstruction using a SMI sensing system. The algorithm can theoretically achieve an unbiased displacement measurement for all feedback cases. Meanwhile, considering the time-slow changing optical feedback level factor (denoted by \( C \)), the thesis presents a simple and fast estimation method of the \( C \) value. Real-time updating for \( C \) further improves the measurement performance for a SMI based displacement sensing system.

A SMI based experimental system is built to verify the proposed algorithm. Signal pre-processing methods for experimental SMSs in terms of filtering and normalization are also presented in this thesis. A commercial sensor is used for confirming the results from our experimental set-up. The comparison between the commercial sensor and
our SMI system shows that the proposed algorithm in the thesis can achieve accurate displacement reconstruction.

The studies in this thesis on SMSs build a solid foundation for developing a SMI based displacement sensor.
Acknowledgement

First and foremost, I would like to show my deepest gratitude to my supervisors, Dr Yanguang Yu and Professor Jiangtao Xi, two respectable, responsible and resourceful scholars, who have provided me with valuable guidance at every stage of the writing of this thesis. Without their enlightened instruction, impressive kindness and patience, I could not have completed my thesis. Their keen and vigorous academic observation enlightens me not only in this thesis but also in my future study.

Secondly, I shall extend my thanks to the staff in School of Electrical, Computer and Telecommunications Engineering (SECTE) and Information and Communication Technology Research (ICTR) Institute who have helped me to develop a fundamental and essential academic competence. My sincere appreciation also goes to the students from Optoelectronic Signal Processing Research (OSPR) Lab, who participated in this study with great cooperation.

Last but not least, my special gratitude goes to my parents for their loving support. They have sacrificed a lot for me due to my research abroad. Without their encouragement and understanding it would have been impossible for me to finish this work.
Publications


Yuanlong Fan, Yanguang Yu, Jiangtao Xi, Joe Chicharo, “Improvement of measurement performance for self-mixing interferometry based displacement sensing system”, manuscript was submitted to journal of IEEE Sensor.

Yuanlong Fan, Yanguang Yu, Jiangtao Xi, Joe Chicharo, "Normalization for self-mixing signals observed in a self-mixing interferometry system", it has been accepted by International Symposium on Photoelectronic Detection and Imaging (ISPDI/SPIE 2011) on 14.March.2011
Table of Contents

Chapter 1  Introduction and Background ................................................................. 1
  1.1  Background ................................................................................................... 2
       1.1.1  Mathematical Models ......................................................................... 4
  1.2  Literature Review .......................................................................................... 10
       1.2.1  SMI Based Displacement Sensing Method ..................................... 11
       1.2.2  Estimation Methods for $C$ ............................................................... 20
  1.3  Outstanding Issues and Contributions ......................................................... 23
       1.3.1  Outstanding Issues .......................................................................... 23
       1.3.2  Contributions .................................................................................. 25
  1.4  Thesis Organization ....................................................................................... 26

Chapter 2  SMI Based Displacement Sensing System ......................................... 29
  2.1  System Structure .......................................................................................... 29
  2.2  Performance of the Laser Diode ................................................................ 30
  2.3  SMI Based Displacement Sensing Principle ............................................... 32
  2.4  Signal Pre-processing .................................................................................. 35
       2.4.1  Filtering ........................................................................................... 38
       2.4.2  Normalization ................................................................................. 45
  2.5  Summary ....................................................................................................... 51
Chapter 3  Improvement for SMI Based Phase Unwrapping Algorithm .................53

3.1 Introduction ........................................................................................................53

3.2 Fringe Shape Analysis .......................................................................................54

3.3 Affection Factors on Displacement Reconstruction Accuracy .......................58

3.4 Improvement of Phase Unwrapping Algorithm ................................................59

3.4.1 Locating Reverse Points $R$ and Jumping Points $J$...............................59

3.4.2 Locating Peak Points $P$ and Valley Points $V$........................................61

3.4.3 Phase Unwrapping Rule .......................................................................61

3.5 Performance Evaluation ....................................................................................63

3.5.1 Simulation Verification ........................................................................63

3.5.2 Experimental Verification .....................................................................65

3.6 Summary .........................................................................................................67

Chapter 4  Real-time $C$ Estimation Method .......................................................69

4.1 Introduction .........................................................................................................69

4.2 Estimation of $C$ ...............................................................................................70

4.2.1 Affection of $C$ on Displacement Reconstruction Accuracy ...............70

4.2.2 Real-time Estimation of $C$ ..................................................................72

4.3 Performance Evaluation ...................................................................................75

4.3.1 Simulation Verification ........................................................................75

4.3.2 Experiment Verification ..........................................................................77

4.4 Summary .........................................................................................................78

Chapter 5  System Performance Test ....................................................................79
5.1 Overall Implementation of SMI Based Displacement Reconstruction Algorithm .............................................................. 79

5.2 Comparison with A Commercial Sensor ................................................................. 82

5.3 Summary ........................................................................................................ 85

Chapter 6 Conclusion and Future work .......................................................................... 87

6.1 Research Contributions ..................................................................................... 87

6.2 Suggested Future Research Topics ..................................................................... 88

Appendix A: Programming list .................................................................................... 91

Appendix B: Reference ............................................................................................... 115
**List of Figures**

**Figure 1-1:** Two mirror Fabry-Perot cavity model, where \( d \) is the laser cavity length, \( L \) is the external cavity length.

**Figure 1-2:** The relationship between \( \omega_0 \) and \( \omega_f \) with \( (C = 0.7, \alpha = 3) \). The dash-dot line represents a linear relation between \( \omega_0 \) and \( \omega_f \).

**Figure 1-3:** The relationship between \( \omega_0 \) and \( \omega_f \) with \( (C = 3, \alpha = 3) \). The dash-dot line represents a linear relation between \( \omega_0 \) and \( \omega_f \).

**Figure 1-4:** Blockdiagram of the feedback interferometer with semiconductor laser for displacement [2].

**Figure 1-5:** Experimental set-up for self-mixing interferometry with diffusive target [27].

**Figure 1-6:** Schematic diagram of the experimental setup [28].

**Figure 1-7:** Experimental set-up in [29].

**Figure 1-8:** Principle of signal processing for joint estimation of the target displacement, \( C \) and \( \alpha \) [32].

**Figure 2-1:** SMI based displacement sensing system.

**Figure 2-2:** A SMS under weak feedback level (with \( C = 0.7, \alpha = 3 \)).

**Figure 2-3:** A SMS under moderate feedback level (with \( C = 3, \alpha = 3 \)).

**Figure 2-4:** A SMS under strong feedback level (with \( C = 7, \alpha = 3 \)).
Figure 2-5: An experimental SMS data with additive white-like noise.

Figure 2-6: An experimental SMS data with sparkle-like impulsive disturbance.

Figure 2-7: An experimental SMS data with slow-time fluctuation.

Figure 2-8: A SMS with sparkle-like noise

Figure 2-9: Filtered result using a median filter (window length is five). (a) a SMS with sparkle-like noise, (b) filtered SMS.

Figure 2-10: Filtered result using a median filter (window length is twelve). (a) a SMS with sparkle-like noise, (b) filtered SMS.

Figure 2-11: Filtered result using a median filter (window length is thirty). (a) a SMS with sparkle-like noise, (b) filtered SMS.

Figure 2-12: (a) A distorted SMS when using a median filter with large window size (b) enlarged section of the filtered SMS.

Figure 2-13: Frequency responses of moving average filters with different $L$.

Figure 2-14: Filtered result using a moving average filter with $L = 20$. (a) a SMS with additive white-like noise, (b) the filtered SMS.

Figure 2-15: SMSs under different feedback levels.

Figure 2-16: Normalization for type I SMS. (a) a simulated SMS with additive white-like noise, (b) SMS after first normalization, (c) SMS after final normalization.

Figure 2-17: Normalization for type II SMS. (a) a simulated SMS with slow-time fluctuation envelope, (b) SMS after first normalization, (c) SMS after final normalization.

Figure 2-18: Normalization for type I SMS. (a) an experiment SMS with additive
white-like noise, (b) SMS after first normalization, (c) SMS after final normalization.

**Figure 2-19:** Normalization for type II SMS. (a) an experiment SMS with slow-time fluctuation envelope, (b) SMS after first normalization, (c) SMS after final normalization.

**Figure 3-1:** Relationship between displacement and a SMS. (a) Displacement $L(n)$ of target, (b) a SMS waveform with $C = 3, \alpha = 3$.

**Figure 3-2:** Classification of fringe shapes. (a) Displacement waveform $L(n)$, (b) sinusoidal-like fringe, named type 1 with $C = 0.5, \alpha = 3$, (c) approximate sawtooth-like fringe, named type 2 with $C = 2, \alpha = 3$, (d) nonlinear sawtooth fringe, named type 3 with $C = 2, \alpha = 1.3$.

**Figure 3-3:** Characteristic points on a SMS ($C = 2, \alpha = 1.3$).

**Figure 3-4:** Numerical simulation results when treating $P$ and $J$ as a same point. (a) displacement waveform $L(n)$, (b) a SMS with $C = 2, \alpha = 4$, (c) reconstructed displacement $\hat{L}(n)$ when treating $P$ and $J$ as a same point, (d) residual error between $L(n)$ and $\hat{L}(n)$.

**Figure 3-5:** Extracting reverse segments from a SMS. (a) a SMS with $C = 3, \alpha = 3$, (b) pulse train $E(n)$, (c) extracted reverse segments of the SMS.

**Figure 3-6:** Segmentation of a SMS with $C = 2, \alpha = 1.3$

**Figure 3-7:** Reconstruction results under a weak feedback level. (a) feedback phase $\phi_f(n)$, (b) simulated SMS with $C = 0.6, \alpha = 2$, (c) reconstructed $\hat{\phi}_f(n)$, (d) residual error between $\phi_f(n)$ and $\hat{\phi}_f(n)$. 
Figure 3-8: Reconstruction results under a moderate feedback level. (a) feedback phase $\phi_r(n)$, (b) simulated SMS with $C = 2, \alpha = 3$, (c) reconstructed $\hat{\phi}_r(n)$, (d) residual error between $\phi_r(n)$ and $\hat{\phi}_r(n)$.

Figure 3-9: Reconstruction results under a strong feedback level. (a) feedback phase $\phi_r(n)$, (b) simulated SMS with $C = 5, \alpha = 1.3$, (c) reconstructed $\hat{\phi}_r(n)$, (d) residual error between $\phi_r(n)$ and $\hat{\phi}_r(n)$.

Figure 3-10: Experiment verification for weak feedback level. (a) an experimental SMS data under weak feedback level, (b) reconstructed $\hat{\phi}_r(n)$.

Figure 3-11: Experiment verification for moderate feedback level. (a) an experimental SMS data under moderate feedback level, (b) reconstructed $\hat{\phi}_r(n)$.

Figure 3-12: Experiment verification for strong feedback level. (a) an experimental SMS data under strong feedback level, (b) reconstructed $\hat{\phi}_r(n)$.

Figure 4-1: Reconstruction result by using a pre-set $C (C = 2)$. (a) displacement $L(n)$, (b) a SMS with $C = 6, \alpha = 3$, (c) reconstructed displacement $\hat{L}(n)$ using the pre-set $C$, (d) residual error between $L(n)$ and $\hat{L}(n)$.

Figure 4-2: Reconstruction results with different estimation value of $\hat{C}$. (a) displacement waveform $L(n)$, (b) a SMS with $C = 3, \alpha = 4$, (c) reconstructed result $\hat{L}(n)$ using a over-estimated $\hat{C}$ ($\hat{C} > 3$), (d) reconstructed result $\hat{L}(n)$ using a down-estimated $\hat{C}$ ($\hat{C} < 3$).

Figure 4-3: Displacement reconstruction results and their corresponding fluctuation signals using deviated $\hat{C}$. (a) displacement reconstruction results $\hat{L}(n)$, (b) fluctuation signals $D_x(n)$.
Figure 4-4: Reconstruction result for an experimental SMS. (a) an experimental SMS, (b) estimation result $\hat{C}$, (c) reconstructed displacement by using $\hat{C}$ obtained in (b).

Figure 5-1: Block diagram of improved displacement reconstruction algorithm.

Figure 5-2: Displacement reconstruction results under weak feedback level. (a) an experimental SMS, (b) displacement reconstruction results cooperating estimated $\hat{C}$, (c) displacement reconstruction results with a pre-set $C$ ($C = 5$).

Figure 5-3: Displacement reconstruction results under moderate feedback level. (a) an experimental SMS, (b) displacement reconstruction results cooperating estimated $\hat{C}$, (c) displacement reconstruction results with a pre-set $C$ ($C = 5$).

Figure 5-4: Displacement reconstruction results under strong feedback level. (a) an experimental SMS, (b) displacement reconstruction results cooperating estimated $\hat{C}$, (c) displacement reconstruction results with a pre-set $C$ ($C = 5$).

Figure 5-5: LTC-025-04-SA: a commercial sensor provided by MTI instruments.

Figure 5-6: Displacement measurement results under weak feedback level. (a) a SMS acquired from our SMI system, (b) SMS after pre-processing, (c) displacement measured by the commercial sensor, (d) reconstructed displacement by the proposed method.

Figure 5-7: Displacement measurement results under moderate feedback level. (a) a SMS acquired from our SMI system, (b) SMS after pre-processing, (c) displacement measured by the commercial sensor, (d) reconstructed displacement by the
proposed method.

**Figure 5-8**: Displacement measurement results under strong feedback level. (a) a SMS acquired from our SMI system, (b) SMS after pre-processing, (c) displacement measured by the commercial sensor, (d) reconstructed displacement by the proposed method.
List of Tables

Table 1-1: Definition of the parameters and variables in (1.15)-(1.18) ..........................................10

Table 1-2: Existing approaches ........................................................................................................24

Table 2-1: Absolute maximum ratings ($T_c = 25^\circ C$ ) [42] .....................................................31

Table 2-2: Electrical characteristics ($T_c = 25^\circ C$ ) [42] .............................................................32

Table 4-1: $C$ estimation results for $\alpha = 1$ ..............................................................................75

Table 4-2: $C$ estimation results for $\alpha = 3$ .................................................................................76

Table 4-3: $C$ estimation results for $\alpha = 5$ .................................................................................76

Table 4-4: $C$ estimation results for noisy SMSs with $\alpha = 3$ .......................................................76
Chapter 1  Introduction and Background

The Self-Mixing Interferometry (SMI) based sensing system has been studied extensively in the last three decades. Since 1968, the first self-mixing interferometry laser based on He-Ne, was developed for measuring shift of a moving remote reflector [1]. SMI has seen applications in metrological quantities measurement such as displacement, velocity and absolute distance of a moving target. Its advantages include low-cost, stability and compactness. Their applications have also been reported in [2-4]. For example, SMI based velocimeters with optical fibers can be used for measuring blood flow [5-6] (a major medical application).

This thesis focuses on researching one of above-mentioned applications – displacement measurements. The existing SMI based displacement reconstruction methods are reviewed. We found that, in order to get the displacement information, the phase of a self-mixing signal (SMS) and a system parameter – feedback level factor (denoted as $C$) must be obtained accurately. This thesis fully studies the characteristics of SMSs. Based on the features of SMSs, an algorithm, which can obtain the displacement information effectively and accurately, is developed in this thesis. The proposed algorithm contains
three steps as outlined below:

- Step1: To obtain the phase information of a SMS accurately;
- Step2: To estimate $C$ value;
- Step3: To reconstruct displacement cooperating results obtained in step 1 and step2.

Also, we present pre-processing methods for experimental SMSs in terms of filtering and normalization to ensure better implementation of the proposed algorithm. To verify the correctness of our algorithm, comparison with a commercial sensor is also presented in this thesis.

This chapter gives an introduction and background for this thesis. The rest of this chapter is organized into the following sections. Section 1.1 gives a brief description of the self-mixing effect and merits of SMI. Also, in this section, the existing mathematical model for describing SMI has been presented. A literature review for SMI has been given in Section 1.2 in terms of displacement measurement and feedback level factor $C$ estimation. Based on the investigation of reviewed articles, the existing problems for SMI based displacement sensing systems are presented in Section 1.3. The contributions of this thesis are also given in Section 1.3. Section 1.4 gives the structure of this thesis.

1.1 Background

In the last decades, the various applications of the Self-Mixing Interferometry (SMI)
based sensing system have attracted the attention of researchers. SMI can be used for measuring metrological quantities, such as velocity, absolute distance, displacement and vibration [3]. The core part of a SMI consists of a Laser Diode (LD), a micro-lens and a target which forms an external cavity of the LD. A SMI works when a small portion of light is back-scattered or reflected by an external target and re-enters into the laser active cavity. The re-entered light modulates both amplitude and frequency of the emitted LD power [7]. This modulated power is called a Self-Mixing Signal (SMS) which can be used to detect metrological quantities associated to the external target. The advantage of using an SMI-based sensing system to perform measurement has been presented in [4]:

✧ No optical interferometer external to the source is required. This leads to a simple and compact set-up.

✧ No alignment is needed because the spatial mode that interacts with the cavity mode is filtered out spatially by the laser itself. It means that detection of diffusive target’s movement becomes possible.

✧ Sensitivity of the scheme is very high (sub-nm sensitivity).

Because of these advantages, SMI has been intensively investigated both theoretically and experimentally for exploring all kinds of sensing applications.
1.1.1 Mathematical Models

1.1.1.1 Three Mirror Model

A single-mode laser diode with external cavity can be considered as a two-mirror Fabry-Perot cavity [8](as shown in Figure 1-1). The back and front mirrors are defined by their reflection coefficient $r_1$ and $r_2$ respectively. $r_3$ represents the reflection coefficient of the external cavity. Generally, $|r_3| \ll |r_2|$, therefore the multiple reflection effect within the external cavity can be neglected [9].

![Two mirror Fabry-Perot cavity model](image)

Figure 1-1: Two mirror Fabry-Perot cavity model, where $d$ is the laser cavity length, $L$ is the external cavity length.

The electric field undergoing a round trip within the compound cavity can be described as follows:

$$E(t) = r_1r_2 \exp\left\{-j4\pi\nu\frac{nd}{c} + (g - \gamma)d\right\} E_0(t)$$

$$+ r_1 \left(1 - |r_2|^2\right) r_3 \exp\left\{-j4\pi\nu\frac{nd + L}{c} + (g - \gamma)d\right\} E_0(t)$$

Where $E_0(t)$ is the initial electric field, $\nu$ is the optical frequency, $n$ is the refractive index of the laser cavity material, $d$ is the laser cavity length, $L$ is the
external cavity length, \( c \) is the speed of light in a vacuum, \( g \) is the linear gain per unit length and \( \gamma \) is the optical loss per unit length within the cavity.

For a stationary stable laser oscillation, the amount of light amplified by the stimulated emission becomes equal to the total losses in the lasing system [9]. This effect results in the following equation:

\[
\begin{align*}
  r_1 \left[ r_2 + \left(1 - |r_2|^2 \right) \right] & \exp \left\{ -j4\pi \nu \frac{L}{c} \right\} \cdot \exp \left\{ -j4\pi \nu \frac{nd}{c} + (g - \gamma) \right\} = 1 \\
\end{align*}
\]

(1.2)

By solving (1.2), the excess required gain \( \Delta g \) and the additional phase \( \phi_a(\nu) \) for the laser system with feedback can be described in (1.3) and (1.4) respectively [9].

\[
\begin{align*}
  \Delta g &= g - g_0 = -\xi \cos \phi_{ext} \\
  \phi_a(\nu) &= \xi \sin \phi_{ext}
\end{align*}
\]

(1.3) and (1.4) are generally the mathematical expression used to describe a laser diode with external cavity. Early researchers accepted this model for SMI based sensing system.

### 1.1.1.2 Lang – Kobayashi Model

The theoretical basis of SMI based sensing system can also be described by Lang –
Kobayashi equation [10-13]. A widely accepted mathematical model for SMI is derived from this equation [10]. The Lang – Kobayashi model equations describe laser fields and their interaction [10]. The equations are rewritten below for the convenience of our description [2]:

\[
\frac{d}{dt} E_0(t) = \frac{1}{2} \left[ G_N \left( N(t) - N_0 \right) - 1/\tau_p \right] E_0(t) + \frac{\kappa}{\tau_L} E_0(t - \tau) \times \cos \left[ \omega_0 \tau + \phi(t) - \phi(t - \tau) \right] \tag{1.6}
\]

\[
\frac{d}{dt} \phi(t) = \frac{1}{2} \alpha G_N \left[ N(t) - N_T \right] - \frac{\kappa}{\tau_L} \frac{E_0(t - \tau)}{E_0(t)} \sin \left[ \omega_0 \tau + \phi(t) - \phi(t - \tau) \right] \tag{1.7}
\]

\[
\frac{d}{dt} N(t) = R_p - \frac{N(t)}{\tau_S} - G_N \left[ N(t) - N_0 \right] E_0^2(t) \tag{1.8}
\]

Where \(E_0(t)\) is the laser electric field, \(\omega_0\) is the angular frequency of the unperturbed laser, \(G_N\) is the model gain coefficient (typically \(G_N = 8 \cdot 10^{-13} m^3 s^{-1}\)), \(N(t)\) is the average carrier density in the active layer, \(N_0\) is the carrier density at transparency (typically \(N_0 = 1.4 \cdot 10^{24} m^{-3}\)), \(N_T\) is the carrier density at threshold for the unperturbed laser (typically \(N_T = 2.3 \cdot 10^{24} m^{-3}\)), \(\tau_p\), \(\tau_L\), \(\tau\) and \(\tau_S\) are photon lifetime, laser diode cavity round trip time, external cavity round trip time and carrier lifetime respectively, \(R_p\) is the electric pumping term and \(\kappa\) is the feedback parameter. \(\alpha\) is the linewidth enhancement factor (LEF) defined as \(\alpha = \left( \frac{\partial \chi_k}{\partial N} / \left( \frac{\partial \chi_i}{\partial N} \right) \right)\) with \(\chi = \chi_k - i \chi_i\) complex susceptibility. \(\alpha\) is a fundamental parameter of laser diode that characterizes the characteristics of laser diodes, such as the linewidth, the chirp, the injection lock range and the dynamic performance. Extensive research work has been done to study this parameter [14-17].
For stationary solutions of (1.6)-(1.8), $E_0(t)$ and $N(t)$ are considered as two constants which are $E_F$ and $N_F$ respectively [2]. Furthermore, the instantaneous optical frequency is $\omega(t) = \omega_0 + \left[ d\phi(t)/dt \right]$, so

$$\phi(t) = (\omega_F - \omega_0)t$$  \hspace{1cm} (1.9)

Where $\omega_F = \omega_F(\tau)$ is the angular frequency of the laser with external feedback. By substituting $E_F$, $N_F$ and (1.9) into (1.6), (1.10) can be obtained as follows:

$$N_F = N_F - \frac{2\kappa}{G_N\tau_L} \cos \omega_F \tau$$  \hspace{1cm} (1.10)

By substituting (1.10) into (1.7), the following relationship can be obtained:

$$\omega_0 \tau = \omega_F \tau + C \sin \left( \omega_F \tau + \arctan \alpha \right)$$  \hspace{1cm} (1.11)

$C$ is called feedback level factor, it can be used for indicating how strong the intensity of light reflected into the laser cavity.

$$C = \frac{\kappa \tau \sqrt{1 + \alpha^2}}{\tau_L}$$  \hspace{1cm} (1.12)

$C$ determines the possible solution numbers of (1.11) [12-13, 18-20]. For $0 < C < 1$, (1.11) gives a unique mapping between $\omega_F$ and $\omega_0$, that is, there is only a single solution in this situation (as shown in Figure 1-2). This range is also known as weak feedback level. For $C > 1$, there are more than two possible solutions in (1.10) (as shown in Figure 1-3). This range is referred to moderate or strong feedback level.
Figure 1-2: The relationship between $\omega_0$ and $\omega_F$ with $(C = 0.7, \alpha = 3)$. The dash-dot line represents a linear relation between $\omega_0$ and $\omega_F$.

Figure 1-3: The relationship between $\omega_0$ and $\omega_F$ with $(C = 3, \alpha = 3)$. The dash-dot line represents a linear relation between $\omega_0$ and $\omega_F$.

From (1.8) and (1.10), the output power $\Delta P$ due to feedback with respect to the unperturbed laser can be obtained:
\[ \Delta P \propto \tau_p \left( R_p - N_0 / \tau_L \right) \frac{2\kappa \tau_p}{\tau_L} \cos \omega \tau \]  \hspace{1cm} (1.13)

It can be better expressed, assuming that \( \kappa \) does not depend on the external cavity length [2], as:

\[ \Delta P = \Delta P_{\text{max}} \cos \omega \tau \]  \hspace{1cm} (1.14)

Based on the above mathematical derivation, equations (1.5), (1.11), (1.14) are used for describing SMI. We rewrite equations for the convenience of our description. Note that time \( \tau \) is replaced by discrete time index for simulation reason. A few signs are introduced to replace signs in equation (1.11) and (1.13) for simplifying SMI model by

\[ \phi_0(n) \rightarrow \phi_{\text{ext}}, \quad I(n) \rightarrow \frac{\Delta P}{\Delta P_{\text{max}}} . \]

\[ \phi_0(n) = 4\pi L(n) / \lambda_n \]  \hspace{1cm} (1.15)

\[ \phi_f(n) = \phi_0(n) - C \sin \left[ \phi_f(n) + \arctan(\alpha) \right] \]  \hspace{1cm} (1.16)

\[ I(n) = \cos(\phi_f(n)) \]  \hspace{1cm} (1.17)

\[ P(n) = P_0 \left[ 1 + bI(n) \right] \]  \hspace{1cm} (1.18)

The physical meaning of the parameters in (1.15)-(1.18) are described in Table 1-1.
Table 1-1: Definition of the parameters and variables in (1.15)-(1.18)

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>$n$</td>
<td>Discrete time index.</td>
</tr>
<tr>
<td>$\phi_n(n)$</td>
<td>Light phase without external optical feedback.</td>
</tr>
<tr>
<td>$L(n)$</td>
<td>Distance between the (LD) facet and the external target.</td>
</tr>
<tr>
<td>$\lambda_n$</td>
<td>Emitted laser wavelength without optical feedback.</td>
</tr>
<tr>
<td>$\phi_f(n)$</td>
<td>Light phase with external optical feedback.</td>
</tr>
<tr>
<td>$C$</td>
<td>Feedback Level Factor.</td>
</tr>
<tr>
<td>$\alpha$</td>
<td>Line-width Enhancement Factor (LEF).</td>
</tr>
<tr>
<td>$I(n)$</td>
<td>Interference function which indicates the influence of the self-mixing effect on the emitted intensity.</td>
</tr>
<tr>
<td>$P(n)$</td>
<td>Laser power emitted by LD with feedback from external cavity.</td>
</tr>
<tr>
<td>$P_0$</td>
<td>Laser power emitted by the free running LD.</td>
</tr>
<tr>
<td>$b$</td>
<td>Modulation index for the laser intensity (typically $b \approx 10^{-3}$).</td>
</tr>
</tbody>
</table>

1.2 Literature Review

In literature, metrology and estimation of physical quantities of a LD are two major applications of SMI. The applications include displacement, vibration, absolute distance, velocity, angle measurement and estimation of $\alpha$, $C$ [3-4]. In this section, we focus the literature review on displacement measurement and $C$ estimation method which are closely related to this thesis work. In the following part of this section, we firstly
review the existing methods for displacement measurement and \( C \) estimation. Based on the review, the existing problems and outstanding issues are identified.

### 1.2.1 SMI Based Displacement Sensing Method

- **AM and FM Demodulation**

  In 1978, Donati \textit{et al.} developed a compact self-aligning interferometry utilising a self-mixing effect to measure displacement [7]. This method is based on the fact that self-mixing effect results in both amplitude and frequency modulation of the laser cavity filed. The theoretical model of this method was developed from the work done by Spencer and Lamb [21] who showed that the two modulations are the in-quadrature interferometric signals \( \cos 2ks \) and \( \sin 2ks \) (where \( k \) is the wave number and \( s \) is the object distance). After demodulation on both amplitude and frequency of the laser cavity field, the object displacement information can be obtained.

- **Frequency Tuning**

  In 1987, Yoshino \textit{et al.} presented two types of interferometer (Twyman-Green and self-coupling interferometers) to measure displacement [22]. The measurement range is over a dynamic range of \( 8–9\mu m \) with a precision of \( 10–60nm \). The development of these two interferometers is based on the idea of locking the phase of the laser to a certain condition and consequently tuning the illumination optical frequency. Another advantage of the application of interferometry is that it can be used for stabilizing the interferometer pattern in 2-D interferometers. Although the experimental set-up is complex, it is helpful when using feedback interferometer to measure displacement by
single mode operation.

• **Fringe Counting Technique**

In 1995, Donati *et al.* proposed a fringe counting technique [2] based on the fact that each fringe on a SMS corresponds to a half-wavelength shift of a moving target. They used a GaAlAs laser diode to measure arbitrary displacements under moderate feedback level simply by means of the back reflection laser from the object surface. Their experimental setup is shown in Figure 1-4. They use an up-down counter to count the number of fringes. The displacement information can be obtained according to the relationship between fringe and displacement (each fringe corresponds to a half-wavelength displacement). This setup is a classical example of SMI based displacement sensing system. In this work, they also presented the mathematical theoretical model for their method which is based on Lang-Kobayashi equations [10]. Although the resolution (\(\lambda/2\), where \(\lambda\) is the wavelength of laser) is unsatisfactory, it provides a good solution for using a single interferometric channel to measure displacement. The measurement range of their interferometer is technically appealing. It can measure 1.2-m displacements on distances up to 2.5-m. However, the accuracy of this laser diode interferometry is limited by the emission wavelength which depends on the driving current and temperature.
In 1996, based on the work in [2], Donati et al. developed a novel, compact and low-cost interferometry, utilizing a PC to interface with a compact optical head which consists of the laser source, a collimating lens, a variable attenuator, a thermoelectric cooler, a temperature sensor and a preamplifier [23]. By using this optical head and a PC, the errors due to temperature fluctuations can be reduced with preprocessing software. The accuracy of this method is about $5 \mu m / m$ with a dynamic range of 2-m.

In 1998, a real-time displacement self-mixing laser diode sensor based on fringe counting technique was developed by Iwamoto et al. [24]. This sensor is compact and fast. It used a direction discrimination circuit to count fringes when the system is running under moderate feedback levels. And its analog output is proportional to an instantaneous displacement. The accuracy of measurement results mainly depends on two aspects:

1. How many bits the D/A converter has.
2. The width of a reset pulse they injected into the counter.

In this paper, the accuracy level was not discussed.

**Linear Interpolation Method**

In 1998, Servagent *et al.* [25] used linear interpolation on the inter-fringe of a sawtooth-like SMS under moderate and high feedback level to measure displacement. They found a linearity principle between a half-wavelength linear displacement and a sawtooth-like output optical power. By using linear interpolation on the output optical power, a resolution of $\lambda/12$ for displacement up to several micrometers can be achieved. The advantage of this method is that there is no need to know the value of feedback level factor $C$ which depends on the reflection coefficient of target. Due to its merit, this method can be applied to non-cooperative targets.

**Phase Shifting Method**

In 2000, Servagent *et al.* [26] first developed an SMI based sensor by using a lithium niobate crystal modulator to perform a phase shift of a SMS. A phase shift of $\pi$ induced a change of $\lambda/4$ (double resolution) in the position of a SMS. A counter was used in this technique. It increases or decreases by step of $\lambda/4$ whenever the SMS reaches zero. In the experiment, they selected three SMSs with different phase shifts to reconstruct displacement using double resolution algorithm. The maximum error is about $65\,nm$ (resolution of $\lambda/12$) for a displacement of $2.34\,\mu m$. However, this method suffers from noise as well as speckle effect and it is restricted in instances of weak feedback levels.

**Speckle Tracking Technique**
Later, in 2001, a laser speckle tracking technique performed on a diffusing object under moderate feedback level for measuring displacement was presented by Norgia et al. [27]. This technique is based on the analysis of probability distribution of amplitude of self-mixing signals on the object. In order to improve the amplitude, a pair of piezo-actuators is used to adjust the lens for tracking the bright spot (as shown in Figure 1-5). The operating range in this technique is over 0.5-1m with fraction-of-wavelength resolution. The significance of this method is that it is the milestone of using speckle-tracking based self-mixing interferometry to measure displacement. However, this method is restricted in moderate feedback level.

Figure 1-5: Experimental set-up for self-mixing interferometry with diffusive target [27].

- **Fourier Transform Method**

In 2001, Wang and Lai proposed a method to reconstruct displacement by sinusoidal phase modulation based SMI [28]. They modulated the phase of a SMS by intentionally changing the external cavity length. The generated SMS is processed by Fourier
transform from which zero and first order of spectra components can be obtained and used for calculating the displacement of the target. The experimental setup of their method is shown in Figure 1-6. In addition, they also gave details about the error sources in this paper. The error sources were summarized as follows:

1. The hysteresis of a piezoelectric transducer (PZT).
2. Fluctuations in bias current or temperature of the LD.
3. Other small sub-modes.

By using signal processing method, the error had been eliminated. The experimental result showed that the measurement error is approximately 8\,nm (resolution of \(\lambda/50\)).

![Figure 1-6: Schematic diagram of the experimental setup [28].](image)

- **Sinusoidal Phase Modulating Method**

In 2005, Guo et al. [29] proposed a sinusoidal phase modulating method to reconstruct harmonic displacement by adding an electro-optic modulator (EOM) between the front mirror of the laser and the external target (as shown in Figure 1-7). This paper is based
on the work done in [28]. The EOM can provide a pure phase modulation in the optical length. By calculating the first and second order Fourier series of the modulated SMS, a wrapped phase of SMS can be obtained. Combining phase unwrapping, the displacement can be easily reconstructed based on the relationship between phase signal and the length of the external cavity. This method reduces the measurement error to 10nm (resolution of $\lambda/80$). However, this method only works for weak feedback level.

![Experimental set-up](image)

Figure 1-7: Experimental set-up in [29].

Five years later, in 2010, in order to carry real-time displacement measurements and improve measurement resolution, Guo and Wang introduced “integrating-bucket algorithm” [30] for reconstructing displacement using the same experimental set-up built in [29]. Integrating-bucket method integrates SMSs four times with the integration time of $T/4$ within a modulation time $T$. In this paper, they gave the error evaluations from 5 aspects and the standard deviation versus those 5 aspects. The accuracy of this method is in the order of nanometers with the measuring range up to a few micrometers. To our best knowledge, this work reported the highest measurement
accuracy for SMI based displacement measurements. However, it is still only suitable for weak feedback level.

- **Phase Unwrapping Method**

In addition to above-mentioned methods, phase unwrapping using advanced signal processing is a very promising technique for SMI based displacement sensing. It is based on the fact that a $2\pi$ phase gap corresponds to a fringe on a SMS waveform. This technique is more appealing because it does not require any extra optical or electrical elements to be added to the basic SMI sensing structure.

The phase unwrapping method was first mentioned in 1997 [31], in order to obtain the displacement of a moving object, Merlo and Donati used equations which are derived from Long-Kobayashi equations [10], to unwrap phase of a SMS. The reconstruction accuracy of this work is in the order of tens of nanometers, but this method needs a pre-estimation of $C$ value and is restricted in weak feedback level.

In 2006, Bes et al. [32] developed a systematic way to reconstruct displacement under moderate feedback level using phase unwrapping method. Since each fringe of a SMS corresponds to a $2\pi$ optical phase change (equivalent to $\lambda/2$ displacement), they first perform phase unwrapping by roughly adding $2\pi$ on the wrapped feedback phase $\phi_r(n)$ (see Table 1.1) whenever a fringe occurs. The unwrapped feedback phase $\phi_r(n)$ they obtained is inaccurate. In order to improve the phase unwrapping accuracy, they developed an optimization algorithm which jointly estimates $C$, line-width enhancement factor (LEF) $\alpha$ and displacement. Figure 1-8 shows the block diagram of their algorithm. The algorithm contains two parts: 1. Rough estimation of
the feedback phase $\phi_f$, 2. Joint estimation of $C_\alpha$ and displacement. The maximum error of the reconstructed displacement is $40 \text{ nm}$ for both harmonic and aleatory displacement. From the results shown in [32], it can be seen that the algorithm contains an inherent system error. The error source will be discussed in detail in Chapter 3. In addition, the optimization approach is sensitive to noise contained in a SMS because the criterion of optimization depends on the instantaneous power of the reconstructed signal discontinuities [33].

Figure 1-8: Principle of signal processing for joint estimation of the target displacement, $C_\alpha$ and $\alpha$ [32].

- **Evolutionary Algorithm**

Displacement reconstruction is based on the fact that each fringe of a SMS corresponds to a half wave-length displacement. If fringe loss occurs for some cases, e.g. strong feedback [34], the accuracy of reconstruction will be greatly degraded, especially in noisy environments. In 2007, in order to improve the robustness of the algorithm in
[32], Doncescu et al. [35] used differential evolution algorithm to detect transitions of a SMS. A coefficient named “Holder Coefficient” was introduced for detecting singularities (i.e. the SMS’s peaks where transition occurs). This method ensures the transition occurs on each fringe can be accurately detected even in a hostile environment. Although the work in [32] improved the measurement accuracy somehow, the inherent system error existed in [32] is still not eliminated.

- **Using GaN Laser**

In 2008, Kliese et al. [36] used a GaN laser as the laser source for an SMI based displacement sensing system. GaN laser produces a shorter wavelength than semiconductor lasers. This obviously can increase fringe resolution, and hence lead to a high reconstruction resolution. In the work of [36], they used three different lasers (IR, Red and GaN) with different wavelength as 780nm, 650nm and 405nm respectively. The resolution for using GaN laser based SMI is twice as high as other two lasers based SMI sensing. The disadvantage of using GaN laser is that the signal noise ratio (SNR) of a SMS is relatively low.

### 1.2.2 Estimation Methods for C

- **Calibration Method**

In 1997, Merlo et al. [31] gave a rough estimation method for $C$ value by measuring the increasing and decreasing time duration of a half-period of a SMS. This calibration method is based on the asymmetry feature of a SMS. However, this method is
restricted in weak feedback level and their experimental reconstruction result reveals ripple-like errors.

- **Gradient Based Optimization Algorithm**

In 2005, Xi *et al.* increased the accuracy of $C$ estimation method by using gradient optimization algorithm [37] to joint estimate $C$ and $\alpha$. This method is based on the data fitting technique which aims to find the value of parameters that can best fit the observed SMS samples. They performed gradient based algorithm on the cost function that they developed. When the cost function is minimized, those parameters are considered as optimal. This method is accurate and robust. The accuracy of $C$ value is 4.63% by using their method. But, this algorithm is restricted in weak feedback level.

In 2007, Yu *et al.* [38] gave an automatic measurement algorithm for $\alpha$. Meanwhile, $C$ value and displacement information can also be obtained. The SMS data can be automatically segmented into blocks based on the symmetrical feature of a SMS in weak feedback level. By studying the surface shape of the cost function, the gradient-based algorithm was performed for the minimization process. Comparing with the work done in [37], this approach does not need to know the exact displacement of the external target. Also, the error caused by the external target does not affect the estimation result of this algorithm.

The above three methods all consider the $C$ value as a constant during the measurements. However, in practical terms, it is difficult to keep a constant $C$ value [39]. This can be seen from the experiment results reported in [17] and [34]. In order to obtain real-time $C$ values, the following approaches were developed.
• **Signal Processing Method**

In 2006, Bes *et al.* [32] developed an optimization criterion, which depends on the instantaneous power of the reconstructed signal, for real-time jointly estimating $C$ and $\alpha$. The purpose of estimating $C$ value is to correct the rough unwrapped phase obtained in the first step of their algorithm. With the estimated $C$ value by using their method, the maximum error of reconstructed displacement is about 40 $nm$ for both harmonic and aleatory displacement. However, this method requires large computation of SMS samples and suffers from noise effect.

• **Derivative-less Optimization Algorithm**

In 2009, based on the work in [32], Zabit *et al.* [33] reduced the computation time by using a hybrid method, which is the combination of several optimization methods, such as Powell’s, Line search, Golden section and Quadratic interpolation method. They found that a wide variation of $\alpha$ is not characteristic for a Fabry Perot LD. So the optimization criterion in [32] becomes a 1-dimentioinal estimation of $C$. This method improves the calculation speed for estimating $C$ value. The minimum number of iterations of this algorithm is 65 with tolerance of $10^{-4}$. However, this method is still restricted in moderate feedback level and its noise resistant ability is unsatisfactory.
1.3 Outstanding Issues and Contributions

1.3.1 Outstanding Issues

Given above literature review, the outstanding research issues on SMI based displacement sensing can be drawn. Table 1-2 briefly describes the performance for the existing approaches on displacement measurement and $C$ estimation.

From the table, we can see the existing problems are

- All the existing methods can only work under a certain feedback level.
- Inherent system error exists in phase unwrapping method.
- An inaccurate $C$ is employed for displacement reconstruction.
Table 1-2: Existing approaches.

<table>
<thead>
<tr>
<th>Displacement reconstruction</th>
<th>Method</th>
<th>Feedback level</th>
<th>Accuracy (resolution)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>AM and FM Demodulation Method [7]</td>
<td>Moderate</td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td>Frequency Tuning Method [22]</td>
<td>Weak</td>
<td>10-60 nm</td>
</tr>
<tr>
<td></td>
<td>Fringe Counting Technique [2, 23-24]</td>
<td>Moderate</td>
<td>427 nm ($\lambda/2$)</td>
</tr>
<tr>
<td></td>
<td>Linear Interpolation Method [25]</td>
<td>Weak or Moderate</td>
<td>65 nm ($\lambda/12$)</td>
</tr>
<tr>
<td></td>
<td>Phase Shifting Method [26]</td>
<td>Weak</td>
<td>65 nm ($\lambda/12$)</td>
</tr>
<tr>
<td></td>
<td>Speckle Tracking Technique [27]</td>
<td>Moderate</td>
<td>Few part in $10^4$ (fraction-of-wavelength)</td>
</tr>
<tr>
<td></td>
<td>Fourier Transform Method [28]</td>
<td>Weak</td>
<td>8 nm ($\lambda/50$)</td>
</tr>
<tr>
<td></td>
<td>Sinusoidal Phase Modulating Method [29-30]</td>
<td>Weak</td>
<td>Less than 10 nm ($\lambda/80$)</td>
</tr>
<tr>
<td></td>
<td>Phase Unwrapping Method [31-32]</td>
<td>Weak or Moderate</td>
<td>Tens of nanometers</td>
</tr>
<tr>
<td></td>
<td>Evolutionary Algorithm [35]</td>
<td>Moderate</td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td>Using GaN Laser [36]</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>C estimation</td>
<td>Calibration method [31]</td>
<td>Weak</td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td>Gradient based optimization algorithm [37] [38]</td>
<td>Weak</td>
<td>4.63%</td>
</tr>
<tr>
<td></td>
<td>Signal processing method [32]</td>
<td>Moderate</td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td>Derivative-less optimization algorithm [33]</td>
<td>Moderate</td>
<td>N/A</td>
</tr>
</tbody>
</table>

In this thesis, we aim to develop an algorithm based on phase unwrapping method, which can resolve above problems. The thesis will firstly study the features of a SMS. We start from the analysis for the fringe shape, from which a set of phase unwrapping rules will be presented. Then we develop a new estimation method for $C$ value to achieve real-time measurement of $C$. 
1.3.2 Contributions

The significant contributions of this thesis are:

✧ A set of phase unwrapping rules is proposed in order to accurately obtain the feedback phase \( \phi_f(n) \) from a SMS. The phase unwrapping method can theoretically achieve an unbiased \( \phi_f(n) \) measurement for all feedback cases.

✧ A fast estimation method for \( C \) value is proposed. The employment of a real-time updating \( C \) can further improve the displacement reconstruction accuracy.

✧ A novel signal normalization method is proposed for pre-processing of experimental SMSs is proposed. The normalization method works well for all feedback cases. Meanwhile, the filtering performance of using a median filter or a mean filter for reducing the noise contained in experimental SMSs is discussed in detail.

✧ Comparison with a commercial sensor to test the accuracy of our displacement reconstruction result is presented. The results show that the displacement reconstructed by using our system is much better than using the commercial sensor.

The measurement results proved by above contributions have been compared and confirmed by a commercial displacement sensor (MTI instrument: LTC-025-04-SA).
1.4 Thesis Organization

This Thesis consists of six chapters:

Chapter 1 first presents the mathematical model for a SMI based sensing system. Then a comprehensive overview of history of displacement reconstruction and $C$ estimation is presented. This chapter also summarizes the outstanding issues and outlines the contributions of this thesis.

Chapter 2 presents our SMI based displacement sensing system and describe the sensing principle. We also give the pre-processing technique for obtaining good quality SMSs. The pre-processing technique includes filtering (a median filter and a moving average filter are chosen as the tool for filtering) and normalization. The normalization method we developed can normalize experimental SMSs into standard form of SMS.

Chapter 3 presents our proposed phase unwrapping method. This method is based on the analysis of fringe shapes of SMSs. It is found that, in order to acquire an accurate feedback phase, characteristic points on a SMS must be located accurately. This chapter gives the method for locating characteristic points and the phase unwrapping rule for obtaining feedback phase $\phi_f (n)$. Verification for the proposed method is also given in this chapter. The simulation results show that the proposed phase unwrapping method can achieve an unbiased feedback phase for all feedback cases.

Chapter 4 describes our method for estimating $C$ value. The reconstructed displacement waveforms are classified into two types when using different pre-set $\hat{C}$ values. Based on the features of two types waveforms, we develop a fast and robust
searching method to find true value of $C$.

Chapter 5 gives the overall implementation of our SMI based displacement reconstruction algorithm and its verification. At the end of this chapter, comparison with a commercial sensor is presented to verify our SMI based displacement reconstruction algorithm.

Chapter 6 summarizes the research activities in this thesis and gives the concluding remarks.
2.1 System Structure

The SMI based displacement sensing system is implemented and shown in Figure 2-1 mainly consists of a LD, a lens and an external target. The LD is controlled by a thermoelectric temperature controller and a LD controller simultaneously. The operation temperature is set as 25°C. The injection current of the LD is set as 90mA. The target is a loud speaker which is driven by a signal generator. Analog processing unit and digital processing unit are used for the processing of SMSs.

The temperature controller we use is TED 200 which is provided by Thorlabs GmbH. It is an extremely precise temperature controller for laser diode and detectors [40].

The TED200 is suited for:

- Wavelength stabilization of laser diode.
- Noise reduction of detectors
- Wavelength tuning by regulating the temperature.
- Modulation of wavelength by tuning the temperature.

The LD controller is the LDC2000 which is also provided by Thorlabs. The LDC2000 is used for controlling the LD and LEDs. With LDC2000 and TEC200, the laser current (or the optical output power) and the LD temperature can be precisely regulated to the required conditions [41].

The LDC2000 LD controller is well suited for:

- Safe and simple operation of all laser diodes up to 2A.
- Power stabilized light sources.
- Wavelength tuning by controlling the current.
- Wavelength modulation by current modulation.

![Diagram of displacement sensing system]

Figure 2-1: SMI based displacement sensing system.

### 2.2 Performance of the Laser Diode

In our SMI based displacement sensing system, the LD we choose is a HL7851G. The HL7851G is the product from the Thorlabs. It is a high power 0.78\(\mu\)m band GaAlAs LD.
with a multi-quantum well (MQW) structure. It is suitable as a light source for optical
disk memories, levelers and various other types of optical equipment.

The absolute maximum ratings and electrical characteristics of the HL7851G are shown
in Table 2-1 and Table 2-2 respectively [42].

<table>
<thead>
<tr>
<th>Item</th>
<th>Symbol</th>
<th>Rated Value</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Optical output power</td>
<td>$P_O$</td>
<td>50</td>
<td>mW</td>
</tr>
<tr>
<td>Pulse optical output power</td>
<td>$P_{O(pulse)}$</td>
<td>60</td>
<td>mW</td>
</tr>
<tr>
<td>LD reverse voltage</td>
<td>$V_{R(LD)}$</td>
<td>2</td>
<td>V</td>
</tr>
<tr>
<td>PD reverse voltage</td>
<td>$V_{R(PD)}$</td>
<td>30</td>
<td>V</td>
</tr>
<tr>
<td>Operating temperature</td>
<td>Topr</td>
<td>-10 to +60</td>
<td>°C</td>
</tr>
<tr>
<td>Storage temperature</td>
<td>Tstg</td>
<td>-40 to +85</td>
<td>°C</td>
</tr>
</tbody>
</table>
Table 2-2: Electrical characteristics (\( T_c = 25^\circ C \)) [42]

<table>
<thead>
<tr>
<th>Item</th>
<th>Symbol</th>
<th>Min</th>
<th>Typ</th>
<th>Max</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Optical output power</td>
<td>( P_o )</td>
<td>50</td>
<td>—</td>
<td>—</td>
<td>mW</td>
</tr>
<tr>
<td>Threshold current</td>
<td>( I_{th} )</td>
<td>—</td>
<td>45</td>
<td>70</td>
<td>mA</td>
</tr>
<tr>
<td>Slope efficiency</td>
<td>( \eta )</td>
<td>0.35</td>
<td>0.55</td>
<td>0.7</td>
<td>mW / mA</td>
</tr>
<tr>
<td>Operating current</td>
<td>( I_{op} )</td>
<td>—</td>
<td>140</td>
<td>170</td>
<td>mA</td>
</tr>
<tr>
<td>LD Operating voltage</td>
<td>( V_{op} )</td>
<td>—</td>
<td>2.3</td>
<td>2.7</td>
<td>V</td>
</tr>
<tr>
<td>Lasing wavelength</td>
<td>( \lambda_p )</td>
<td>775</td>
<td>785</td>
<td>795</td>
<td>nm</td>
</tr>
<tr>
<td>Beam divergence (parallel)</td>
<td>( \theta / )</td>
<td>8</td>
<td>9.5</td>
<td>12</td>
<td>deg</td>
</tr>
<tr>
<td>Beam divergence (perpendicular)</td>
<td>( \theta \perp )</td>
<td>18</td>
<td>23</td>
<td>28</td>
<td>deg</td>
</tr>
<tr>
<td>Monitor current</td>
<td>( I_s )</td>
<td>25</td>
<td>—</td>
<td>150</td>
<td>( \mu A )</td>
</tr>
<tr>
<td>Astigmatism</td>
<td>( A_s )</td>
<td>—</td>
<td>5</td>
<td>—</td>
<td>( \mu m )</td>
</tr>
</tbody>
</table>

### 2.3 SMI Based Displacement Sensing Principle

SMI based sensing system has been studied extensively [8, 43-44]. The mathematical model mentioned in section 1.1.1.2 (Equations (1.15)-(1.18)) is a widely accepted model for describing SMI. They are rewritten below for the convenience:

\[
\phi_0 (n) = \frac{4\pi L(n)}{\lambda_0}
\]  
\[
\phi_r (n) = \phi_0 (n) - C \sin \left[ \phi_r (n) + \arctan (\alpha) \right]
\]  
\[
I (n) = \cos \left( \phi_r (n) \right)
\]
\[ P(n) = P_0 [1 + b I(n)] \]  

\( \lambda_0 \) is the emitted laser wavelength without feedback and \( L(n) \) is the distance between the Laser Diode (LD) and the external target. \( n \) is the discrete time index. 

\( C \) and \( \alpha \) are the optical Feedback Level Factor (FLF) and the Linewidth Enhancement Factor (LEF) respectively. \( \phi_F(n) \) is the laser phase when an external target exists and \( \phi_0(n) \) is the laser phase without feedback under free running conditions. \( I(n) \), which is called interference function, indicates the influence of the self-mixing effect on the emitted intensity. \( P(n) \) is the power emitted by a LD with optical feedback from its external cavity. \( P_0 \) is the power emitted by the free running LD and \( b \) is the modulation index for the laser intensity (typical \( b = 10^{-3} \)). 

Phase unwrapping based displacement reconstruction is to retrieve \( L(n) \) using a SMS \( I(n) \). The reconstruction procedure follows: \( I(n) \rightarrow \phi_F(n) \rightarrow \phi_0(n) \rightarrow L(n) \). 

Obtaining \( \phi_F(n) \) correctly from \( I(n) \) is the first important step (our method for accurately obtaining \( \phi_F(n) \) will be described in chapter 3). Due to complicated relationship between \( \phi_F(n) \) and \( \phi_0(n) \) described in (2.2), the fringe shape of a SMS exhibits variety for different optical feedback levels which are characterized by \( C \) values [34]. Algorithms for obtaining \( \phi_F(n) \) require an accurate analysis of a SMS’s shape. Figure 2-2 – Figure 2-4 show three SMSs under different feedback levels. For \( 0 < C < 1 \), known as weak feedback level, the waveform of a SMS is asymmetric sinusoidal like. For \( C > 1 \), known as moderate or strong feedback level, the waveform of a SMS reveals abrupt transitions with hysteresis.
Figure 2-2: A SMS under weak feedback level (with $C = 0.7, \alpha = 3$).

Figure 2-3: A SMS under moderate feedback level (with $C = 3, \alpha = 3$).

Figure 2-4: A SMS under strong feedback level (with $C = 7, \alpha = 3$).
Once $\phi_r(n)$ is obtained without any deviation, we can use (2.2) to calculate $\phi_i(n)$ and then displacement $L(n)$ can be reconstructed using (2.1). Obviously, parameter $\alpha$ and $C$ are required for this reconstruction. Inaccurate values of $\alpha$ and $C$ logically must cause a reconstruction error. The measurement for $\alpha$ and $C$ has been discussed in [16], [17], [19]. Generally, $\alpha$ is thought as constant for a laser diode. However, $C$ can be varying for a SMI system. This can be seen from the experiment results reported in [17], [34]. So a real-time $C$ estimation is a necessary step for achieving an accurate displacement reconstruction. Chapter 4 will give the method for real-time $C$ estimation method.

### 2.4 Signal Pre-processing

In practice, SMSs are often disturbed by noises. The contaminated SMSs used for reconstructing displacement will greatly degrade the accuracy of the final reconstruction result. However, from the literature, only few studies have researched the effects of de-noising SMSs. In 2007, Yu et al. [45] summarized noise or disturbance affecting SMSs into three types:

1. **Additive white-like noise.** This noise commonly exists in the acquired SMSs (as shown in Figure 2-5).

2. **Sparkle-like impulsive disturbance.** This disturbance occurs when the feedback level becomes stronger (usually in moderate feedback level). Figure 2-6 shows a SMS
that suffers from sparkle-like impulsive disturbance.

3. Slow-time fluctuation in the envelope of a SMS (as shown in Figure 2-7). The envelope is caused by a multiplicative noise which characteristics are unknown. This slow-time fluctuation occurs when a SMS is under moderate or strong feedback level.

Figure 2-5: An experimental SMS data with additive white-like noise.
In order to remove noise and disturbance effectively, Yu et al. [45] used a combination of two filters (Median filter and Kaiser filter) to achieve good performance in noise reduction of SMSs. In 2008, Zhang et al. [46] analyzed the spectrum of SMSs under
both weak and moderate feedback level. They found that $C$ value has significant influence to the bandwidth of SMSs under weak feedback levels and the SMSs are strictly band-limited. Based on this feature, they designed a band-pass filter to eliminate noise. However, at moderate feedback level, it is hard to utilize a band-pass filter because the spectrum spreads over a wider range at high frequency [46].

Both works in [45] and [46] are restricted in a certain feedback levels. In this section, in order to achieve a good performance of our SMI based displacement reconstruction algorithm, we pre-process SMS in terms of filtering and normalization. The purpose of filtering is to eliminate white-like noise and sparkle-like impulsive disturbance. Based on the work in [45], we used a median filter as the main filter to remove sparkle-like impulsive disturbance. To eliminate white-like noise, we used a moving average filter. The slow-time fluctuation in the envelope of SMSs is a low-frequency signal, it is hard to design a low-pass filter to remove the fluctuation because the main frequency of SMSs is very close to the frequency of the envelope. To get around this problem, in this section, we proposed a normalization method to remove the envelope of SMSs. Also, normalization is an essential step in the pre-processing of SMSs.

2.4.1 Filtering

2.4.1.1 Median Filter

Median filter is the nonlinear filter more used to remove the impulsive noise from signals [47-49]. It has been widely used in 2-D image processing to reduce ‘salt and
pepper’ noise. A median filter is effective when the goal is to simultaneously reduce noise and preserve edges. For 1-D signals, median filter applies a sliding window to a sequence. It replaces the centre value in the window with the median value of all the points within the window [50]. For example:

We have a 1-D data sequence input and use a window size of three to filter the sequence.

\[
\text{input} = [3, 90, 2, 4].
\]

So, the median filtered output signal will be:

\[
\text{output}[1] = \text{median}[3, 3, 90] = 3.
\]

\[
\text{output}[2] = \text{median}[3, 90, 2] = \text{median}[2, 3, 90] = 3.
\]

\[
\text{output}[3] = \text{median}[90, 2, 4] = \text{median}[2, 4, 90] = 4.
\]

\[
\text{output}[4] = \text{median}[2, 4, 4] = 4.
\]

The filtered output is

\[
\text{output} = [3, 3, 4, 4].
\]

It should be noted that for the above example, the first value is repeated as well as the last value. From this simple example, we can see that 90, which can be considered as a sparkle in the sequence, is removed.

For SMSs under moderate feedback level, sparkles like 90 in the above example always degrade the quality of SMSs. The zoomed sparkle appeared in Figure 2-6 is shown in Figure 2-8.
Based on the features of median filter discussed above, we use a median filter to remove the sparkle-like disturbance in SMSs. Figure 2-9 – Figure 2-11 show the filtered results of SMSs using median filters with different window size. From the results, we can see that a median filter with large window size can totally eliminate sparkles on a SMS. However, a large window can also distort the original shape of SMSs’ waveforms. This can be seen from the zoomed fringes of the filtered SMS (as shown in Figure 2-12) in Figure 2-11. Based on our experience, a median filter with a window size of 12 has the best performance on reducing sparkle-like noise while keep waveform nearby unchanged.
Figure 2-9: Filtered result using a median filter (window length is five). (a) a SMS with sparkle-like noise, (b) filtered SMS.

Figure 2-10: Filtered result using a median filter (window length is twelve). (a) a SMS with sparkle-like noise, (b) filtered SMS.
Figure 2-11: Filtered result using a median filter (window length is thirty). (a) a SMS with sparkle-like noise, (b) filtered SMS.

Figure 2-12: (a) A distorted SMS when using a median filter with large window size (b) enlarged section of the filtered SMS.

2.4.1.2 Moving Average Filter

To remove the additive white-like noise from SMSs, we choose a moving average filter
as our filtering tool. A moving average filter is a type of finite impulse response (FIR) filter which is commonly used with time series data to remove short-term fluctuations. Mathematically, it is quite similar to the low-pass filter used in signal processing [51]. The impulse response of a $L$ sample moving average is:

$$h(n) = \frac{1}{L}, \quad \text{for } n = 0, 1, ..., L-1 \quad (2.5)$$

$$h(n) = 0, \quad \text{otherwise} \quad (2.6)$$

So the frequency response $H(\omega)$ of the moving average filter is:

$$H(\omega) = \left(\frac{1}{L}\right) \sum_{m=0}^{L-1} e^{-j\omega m}, \quad (2.7)$$

We can use the well-known mathematical identity

$$\sum_{m=N}^{M} a^n = \frac{a^N - a^{M+1}}{1-a}, \quad (2.8)$$

to rewrite the frequency response as:

$$H(\omega) = \left(\frac{1}{L}\right) \frac{1 - e^{-j\omega L}}{1 - e^{-j\omega}}, \quad (2.9)$$

Where we have let $a = e^{-j\omega}$, $N = 0$ and $M = L-1$.

Figure 2-13 shows the frequency responses of moving filters with different $L$. 

43
In this thesis, according to the low-pass features of the moving average filter, we use a moving filter with \( L = 20 \) to remove the additive white-like noise from SMSs. Figure 4-10 shows the filtered result.

![Graph showing frequency responses of moving average filters with different \( L \).](image)

**Figure 2-13:** Frequency responses of moving average filters with different \( L \).

![Graphs showing filtered result using a moving average filter with \( L = 20 \).](image)

**Figure 2-14:** Filtered result using a moving average filter with \( L = 20 \). (a) a SMS with additive white-like noise, (b) the filtered SMS.
2.4.2 Normalization

Normalization is an essential step in the pre-processing of SMSs. In section 2.3 of this chapter, we have presented the mathematical model for a SMI system. From Equation (2.3), we can see that, in order to use inverse cosine function to obtain accurate phase information, a SMS $J(n)$ must be in the range of $[-1, 1]$ (as shown in Figure 2-15). However, in practice, SMSs acquired from the experimental set-up can hardly cover the range of $[-1, 1]$ (This can be seen from Figure 2-5 – Figure 2-7). So, to normalize SMSs is necessary before further measurements can be done. Also, as mentioned before, slow-time fluctuation in the envelope of a SMS is hardly removed by the conventional filtering method. Normalization can provide an alternative way to solve this problem by normalizing those deviated points to their real values.
2.4.2.1 Normalization Algorithm

In Figure 2-15, when increasing $C$ value, the amplitude of fringe becomes less and less. Based on this feature, we can classify SMSs into two types according to the peak to peak amplitude of fringe. One type SMS (type I) is that the peak to peak amplitude of each fringe covers the whole range of $[-1, 1]$ (as shown in Figure 2-15(a)). This type of SMS is under weak feedback level. Another type (type II) is that the peak to peak magnitude of each fringe is less than 2 (as shown in Figure 2-15(b)-(e)). In this type of SMS, peak points $P$ in right inclination segment reach 1, valley points $V$ in left inclination segment reach $-1$ (definition of left and right inclination will
presented in Chapter 3). Our normalization algorithm is developed based on these two types SMSs.

The basic procedure of our normalization algorithm can be described as follows:

- We first perform a rough normalization on a SMS $I(n)$. This step we call it as “first normalization”. First normalization contains three steps and it can be described as the following three consequent mathematical equations.

  $$I(n) = I(n) - \text{Mean}(I(n))$$ (2.10)

  $$I(n) = \frac{I(n)}{\max(\text{abs}(I(n)))}$$ (2.11)

  $$I(n) = \frac{I(n) - (A + B)/2}{A - B}$$ (2.12)

The purpose of (2.10) is to centre $I(n)$ at 0. (2.11) to ensure that the amplitude of $I(n)$ does not exceed the range of $[-1, 1]$. $\text{abs}$ in (2.11) means that to take absolute value of $I(n)$. (2.11) can limit $I(n)$ to a reasonable range. $A$ and $B$ in (2.12) are the maximum value and minimum value of $I(n)$ respectively.

- Then, we segment the rough normalized SMS into fringes and find peak points $P$ and valley points $V$ of the SMS (details of segmentation SMSs and find characteristic points of a SMS is described in chapter 3).

- After doing the segmentation of the SMS, we apply the same equations used in rough normalization on each segmented fringe. This step is called final normalization. For final normalization, type I and type II mentioned before should be normalized separately. $A$ and $B$ should be chosen as different values from first normalization. How to choose $A$ and $B$ in final normalization is presented as follows.
Since the peak to peak amplitude of each fringe covers the whole range of $[-1, 1]$ for type I SMSs, we simply set $A$ as 1 and $B$ as $-1$.

For type II SMSs, we can use the following equations to calculate $A$ and $B$.

In the right inclination segment of a type II SMS:

$$B = -1,$$

$$A = \text{mean}(I(P(j))),$$

(2.13)

(2.14)

In the left inclination segment of a type II SMS:

$$B = \text{mean}(I(V(j))),$$

(2.15)

$$A = 1,$$

(2.16)

$P(j)$ and $V(j)$ are two variables which are used for storing peak points $P$ and valley points $V$ of the SMS.

### 2.4.2.2 Simulation Verification for Normalization Method

To test our algorithm for normalization, we add an additive white-like noise to the type I SMS. For type II SMS, in order to simulate slow-time fluctuation of the envelop, we use a sinusoidal signal with a frequency close to the main frequency of SMS to modulate type II SMS. The sinusoidal signal is $S(n) = A_v + \Delta A \cos \left(2\pi \frac{f}{f_s}n\right)$, where $A_v$ is the DC component, $f$ is the vibration frequency, $f_s$ is the sampling frequency. Simulation parameters are chosen as: $A_v = 80, \Delta A = 20, f = 102Hz, f_s = 51200Hz$. The normalization result for type I and type II SMS are shown in Figure
2-16 and Figure 2-17 respectively.

Figure 2-16: Normalization for type I SMS. (a) a simulated SMS with additive white-like noise, (b) SMS after first normalization, (c) SMS after final normalization.

Figure 2-17: Normalization for type II SMS. (a) a simulated SMS with slow-time fluctuation envelope, (b) SMS after first normalization, (c) SMS after final normalization.

2.4.2.3 Experiment Verification for Normalization Method

We also tested the normalization algorithm for experimental data. Figure 2-18 – Figure
2-19 show the normalization result for type I and type II SMS respectively.

Figure 2-18: Normalization for type I SMS. (a) an experiment SMS with additive white-like noise, (b) SMS after first normalization, (c) SMS after final normalization.

Figure 2-19: Normalization for type II SMS. (a) an experiment SMS with slow-time fluctuation envelope, (b) SMS after first normalization, (c) SMS after final normalization.
2.5 **Summary**

In this chapter, we firstly present our SMI based displacement sensing system structure and describe core elements of the system. Then, the basic procedure of SMI phase unwrapping displacement reconstruction method is described. The pre-processing method for noise reduction of SMSs in terms of filtering and normalization is also presented in this chapter. Based on the features of noises, we choose a median filter and a moving average filter to remove sparkle-like noise and white-like noise respectively. In order to apply our SMI based displacement reconstruction algorithm, SMSs needs to be normalized. A novel normalization method is developed in this chapter. The normalization method we proposed is also effective for removing slow-time fluctuation of the envelope of a SMS. In order to let our normalization method is suitable for all feedback cases, we classified SMSs into two types based on the peak to peak amplitude of fringe. Both simulated data and experimental data are used for verifying our normalization method.
Chapter 3 Improvement for SMI Based Phase Unwrapping Algorithm

3.1 Introduction

SMI based displacement sensing technology has attracted significant research focus with early research utilizing fringe counting technique [2, 23]. It is based on the fact that each fringe on a SMS corresponds to a half-wavelength shift of a moving target. Although the resolution is unsatisfactory, it provides a good solution for using single interferometric channel to measure displacement. In order to improve the resolution and measurement range, some new approaches were reported [25, 27, 29, 31]. Among those approaches, a phase unwrapping technique using advanced signal processing is more attractive because this method does not require any extra optical or electrical elements to the basic SMI based sensing structure. In this thesis, we focus on phase unwrapping method to reconstruct displacement. However, the existing phase unwrapping methods only work in certain feedback cases. By carefully studying the existing phase unwrapping methods, we found that these methods cannot perfectly
reconstruct displacement information from a SMS. There is an inherited measurement error existing in these methods. This can be seen from the results presented in Figure 2 and Figure 3 in [31], Figure 3 and Figure 8 in [52], Figure 9 in [53]. Obviously, the reconstruction errors shown in [31], [52], [53] are deterministic and regular, which implies that the measurement error is caused by the measurement algorithm itself.

In this chapter, we firstly analyze what causes the measurement error for the phase unwrapping algorithm mentioned above. Then, we study the fringe phase information features of SMSs. The phase information contains displacement information, which can be obtained by carrying out an accurate fringe analysis on a SMS. Based on the features, we propose an improved phase unwrapping algorithm which can theoretically achieve an unbiased feedback phase for all feedback cases.

This chapter is organized as follows: In section 3.2 and section 3.3 the cause of measurement error in the existing method is presented by analyzing fringes’ shapes. The proposed phase unwrapping algorithm and its verification are given in section 3.4 and section 3.5 respectively. Section 3.6 summarizes this chapter.

### 3.2 Fringe Shape Analysis

The relationship between displacement and a SMS waveform is shown in Figure 3-1. Displacement $L(n)$ is a simple harmonic vibration (as shown in Figure 3-1(a)). The SMS plot in Figure 3-1(b) is generated using (2.1)-(2.4) with $C = 3, \alpha = 3$. We can see during the vibration period of increasing displacement (the target moves away from
the laser), each fringe is left-inclined. During the period of decreasing displacement (the target moves toward the laser), each fringe is right-inclined. Each fringe corresponds to a $2\pi$ phase change of $\phi_f(n)$ (equivalent to $\lambda/2$ change of the displacement).

![Figure 3-1: Relationship between displacement and a SMS. (a) Displacement $L(n)$ of target, (b) a SMS waveform with $C = 3, \alpha = 3$.](image)

Using SMI model described by (2.1)-(2.4), extensive simulations for SMSs were conducted. The fringe shapes of SMSs can be classified into three types (as shown in Figure 3-2): 1. Sinusoidal-like fringes; 2. Approximate sawtooth-like fringes; and 3. Non-linear sawtooth fringes. Sinusoidal-like fringe occurs in weak feedback level. The second and third types happen in moderate or high feedback level with $C > 1$. Figure 3-2(b) shows a type 1 SMS with $C = 0.5, \alpha = 3$ and its one enlarged fringe. Figure 3-2(c) shows a type 2 SMS and its one enlarged fringe with $C = 2, \alpha = 3$. A type 3 SMS with $C = 2, \alpha = 1.3$ and its enlarged fringe are shown in Figure 3-2(d).
We only consider the fringe shape of type 3 for fringe analysis because type 1 and type 2 can be converted from type 3. For the convenience of description, re-plot Type 3 ($C=2, \alpha=1.3$) on Figure 3-3. Four characteristic points on a SMS (marked by $R$, $P$, $V$ and $J$) are used to describe displacement characteristics. $R$ is called ‘reverse point’ which indicates the position where the target changes its movement direction. $P$ and $V$, called ‘peak point’ and ‘valley point’ respectively, are the maximum and
minimum points on each fringe. ‘Jumping point’ denoted as $J$, is the point where a fringe value occurs sudden change, that is ‘discontinuity’. For type 1, there is no jumping points occur on fringes. For type 2, valley point and jumping point are the same point which means $V$ and $J$ are superposition.

![Diagram of characteristic points on a SMS ($C = 2, \alpha = 1.3$).](image)

In carefully observing the fringe shape in type 3 (shown in Figure 3-3), we can see that the fringe includes two monotonic regions marked by $VP$ and $PJ$. $\phi_r(n)$ should be calculated respectively in these two regions. If roughly treating $P$ and $J$ as a same point, the calculation of $\phi_r(n)$ must cause a reconstruction error for displacement measurement.
3.3 Affection Factors on Displacement Reconstruction Accuracy

Figure 3-4 shows the numerical simulation result when treating $P$ and $J$ as the same point. We can see that the residual error between $L(n)$ and $\hat{L}(n)$ is significant up to $0.5\mu m$. This residual error is similar to the result presented in [52] (Figure 8 in [52]). We can say that the algorithm in [52] uses an inaccurate $\phi_f(n)$ for displacement reconstruction.

Figure 3-4: Numerical simulation results when treating $P$ and $J$ as a same point. (a) displacement waveform $L(n)$, (b) a SMS with $C = 2, \alpha = 4$, (c) reconstructed displacement $\hat{L}(n)$ when treating $P$ and $J$ as a same point, (d) residual error between $L(n)$ and $\hat{L}(n)$. 

58
3.4 Improvement of Phase Unwrapping Algorithm

In this section, we present an improved algorithm in terms of obtaining an accurate $\phi_r(n)$. From the above analysis, we can see that the problem in the existing phase unwrapping displacement reconstruction methods is caused by the non-monotonicity of SMFs’ fringes. In order to distinguish two different monotonic regions, characteristic points must be accurately located. In the following section, we will give the method for finding these points and analytical expression of our phase unwrapping method to obtain $\phi_r(n)$.

To obtain $\phi_r(n)$ accurately, a SMS should be segmented correctly. The characteristic points described in section 3.2 are used for the segmentation of a SMS. Hence, the first important step is to locate these points from a SMS.

3.4.1 Locating Reverse Points $R$ and Jumping Points $J$

In order to locate reverse points, we first differentiate a SMS $I(n)$ (shown in Figure 3-5(a)) and then reshape the differentiated signal to obtain a pulse train $E(n)$ (shown in Figure 3-5(b)). $E(n)$ can indicate locations of jumping points $J$. We also use $E(n)$ to extract reverse segments marked by $R_{seg}$ (shown in Figure 3-5(c)). A reverse point is the maximum or minimum point of a reverse segment. Reverse points are stored in variable $R(m)$ where $m$ is the reverse segment index. The interval between two adjacent reverse points $R$ is called right or left inclination segment.
shown in Figure 3-5(a). The definition on right or left inclination can also be seen from Figure 3-1.

![Figure 3-5](image_url)

Figure 3-5: Extracting reverse segments from a SMS. (a) a SMS with \( C = 3, \alpha = 3 \), (b) pulse train \( E(n) \), (c) extracted reverse segments of the SMS.

From \( E(n) \) and \( R(m) \), we can obtain a flag signal named \( F(n) \) to indicate right or left inclination segment. \( F(n) \) is also used for the calculation of \( \phi_f(n) \). In order to obtain \( F(n) \), we calculate the mean value of \( E(n) \) between two adjacent reverse points \( R(m) \) and \( R(m+1) \). Those mean values are stored in a variable \( S(m) \). If \( S(m) > 0 \), we set \( F(n) = 1 \) for all instances between \( R(m) \) and \( R(m+1) \) to indicate right inclination segment. Otherwise, we set \( F(n) = 0 \) to indicate left inclination segment.
\[ F(n) = \begin{cases} 1 & \text{for right inclination segment} \\ 0 & \text{for left inclination segment} \end{cases} \quad (3.1) \]

### 3.4.2 Locating Peak Points \( P \) and Valley Points \( V \)

With the signal \( E(n) \) and \( F(n) \), a SMS can be further segmented into \( j \) fringes.

Figure 3-6 shows a segmented SMS. A peak point is the maximum point in each fringe. A valley point is minimum point between each adjacent peak points.

![Figure 3-6: Segmentation of a SMS with \( C = 2, \alpha = 1.3 \)](image)

#### 3.4.3 Phase Unwrapping Rule

We apply inverse cosine function onto a segmented SMS to get a wrapped phase denoted by \( \phi_{pw}(n) \) according to (2.3)

\[ \phi_{pw}(n) = \arccos (I(n)) \quad (3.2) \]
The, we locate characteristic points $P$, $V$ and $J$, and store their index in variable $P(j)$ and $V(j)$ and $J(j)$ respectively, $j$ is the fringe index which is a positive integer. Feedback phase $\phi_f(n)$ can be obtained by unwrapping $\phi_{fw}(n)$ as follows.

For weak feedback levels with $C < 1$:

$$\phi_f(n) = (-1)^{i_v} \phi_{fw}(n) + 2\pi i_v$$

(3.3)

For moderate and strong feedback levels:

When $F(n) = 0$:

$$\phi_f(n) = -\phi_{fw}(n) + 2\pi i_v$$

for $VP$ region (3.4)

and

$$\phi_f(n) = \phi_{fw}(n) + 2\pi i_j$$

for $PJ$ region (3.5)

When $F(n) = 1$:

$$\phi_f(n) = (-1)^{i_v} \phi_{fw}(n) + 2\pi (i_v + 1)$$

(3.6)

In (3.3) – (3.6), $i_{v,p}$, $i_v$ and $i_j$ are three counters. $i_{v,p}$ counts the times of sign change of $\phi_{fw}(n)$. $i_v$ and $i_j$ count the number of valley points and jumping points respectively.

The value of $i_{v,p}$, $i_v$ and $i_j$ are updated as follows:

$$i_{v,p} = i_{v,p} + 1, \quad n = P(j) \ or \ n = V(j)$$

(3.7)

$$i_v = i_v - 1, \quad n = V(j) \ & \ F(n) = 1$$

(3.8)

$$i_v = i_v + 1, \quad n = V(j) \ & \ F(n) = 0$$

(3.9)

$$i_v = i_v + 1, \quad n = J(j) \ & \ F(n) = 0$$

(3.10)

The initial values of $i_v$ and $i_j$ are set as 0. The initial value of $i_{v,p}$ can be determined by the difference of the first two samples in $\phi_{fw}(n)$ (denoted as $D_{1,2}$) and the difference of first two samples in $I(n)$ (denoted as $I_{1,2}$). When $D_{1,2}$ has the
same sign with \( I_{1,2} \), the initial value of \( i_{v,p} \) is set as 0. Otherwise, its initial value is set as 1.

### 3.5 Performance Evaluation

#### 3.5.1 Simulation Verification

SMS data was generated using (2.1)-(2.4) to verify the proposed algorithm, by using a simple harmonic vibration as the displacement. Where \( L(n) = L_0 + \Delta L \cos \left( \frac{2\pi f}{f_s} n \right) \), and where \( L_0 \) is the initial distance between the laser surface and the target, \( f \) is the vibration frequency, \( f_s \) is the sampling frequency. Simulation parameters are chosen as: \( L_0 = 0.24m \), \( \Delta L = 1.37\mu m \), \( f = 100Hz \) and \( f_s = 51200Hz \). In order to test the accuracy of \( \phi_f(n) \) calculation, three SMSs were generated and evaluated their corresponding feedback phase \( \phi_f(n) \) under different feedback levels (weak, moderate and strong). Figure 3-7 – Figure 3-9 show the reconstruction results by the proposed reconstruction algorithm, accordingly it can be seen that the reconstruction error of \( \phi_f(n) \) is zero for any feedback cases. Therefore, it can be concluded that the proposed algorithm is able to obtain \( \phi_f(n) \) accurately.
Figure 3-7: Reconstruction results under a weak feedback level. (a) feedback phase $\phi_f(n)$, (b) simulated SMS with $C = 0.6, \alpha = 2$, (c) reconstructed $\hat{\phi}_f(n)$, (d) residual error between $\phi_f(n)$ and $\hat{\phi}_f(n)$.

Figure 3-8: Reconstruction results under a moderate feedback level. (a) feedback phase $\phi_f(n)$, (b) simulated SMS with $C = 2, \alpha = 3$, (c) reconstructed $\hat{\phi}_f(n)$, (d) residual error between $\phi_f(n)$ and $\hat{\phi}_f(n)$. 

3.5.2 Experimental Verification

Using the experimental setup described in chapter 2 (section 2.1), three SMSs were acquired under three different feedback levels by adjusting the distance between target and LD. By firstly using the pre-processing method described in chapter 2 to pre-process SMSs, experiments were performed using the proposed phase unwrapping method on these SMSs. Figure 3-10 – Figure3-12 show the phase unwrapping results. The overall verification of our proposed SMI based displacement reconstruction algorithm (the rest part of the overall algorithm is given in chapter 4) will be presented in chapter 5. The displacement reconstructed by our algorithm is compared with a simulated SMS with $C = 5$, $\alpha = 1.3$. (c) reconstructed $\hat{\phi}_f(n)$, (d) residual error between $\phi_f(n)$ and $\hat{\phi}_f(n)$. 

Figure 3-9: Reconstruction results under a strong feedback level (a) feedback phase $\phi_f(n)$, (b) simulated SMS with $C = 5$, $\alpha = 1.3$, (c) reconstructed $\hat{\phi}_f(n)$, (d) residual error between $\phi_f(n)$ and $\hat{\phi}_f(n)$. 

commercial sensor (shown in chapter 5).

Figure 3-10: Experiment verification for weak feedback level. (a) an experimental SMS data under weak feedback level, (b) reconstructed \( \hat{\phi}_F(n) \).

Figure 3-11: Experiment verification for moderate feedback level. (a) an experimental SMS data under moderate feedback level, (b) reconstructed \( \hat{\phi}_F(n) \).
Figure 3-12: Experiment verification for strong feedback level. (a) an experimental SMS data under strong feedback level, (b) reconstructed $\hat{\phi}_F(n)$.

3.6 Summary

This chapter presented a phase unwrapping algorithm for obtaining the feedback phase $\phi_F(n)$ of a SMS. By studying the fringe shape of SMSs, it was found that the error introduced in the existing phase unwrapping method is caused by the non-monotonicity of SMSs’ fringes. The accuracy of this phase unwrapping method mainly depends on the location accuracy of characteristic points of a SMS. Both computer simulations and experiment is used to verify the proposed algorithm. The results show that the feedback phase $\phi_F(n)$ can be reconstructed by the proposed algorithm effectively and accurately. The proposed phase unwrapping algorithm builds a solid foundation for SMI based displacement reconstruction.

As mentioned before (in section 2.2.1), a fast real-time estimation of $C$ value is
important for displacement reconstruction. According to (2.2), once $\phi_r(n)$ is obtained accurately, we need a $C$ value to finally determine the displacement. The following chapter will detail the methodology for $C$ estimation.
Chapter 4  Real-time $C$ Estimation Method

4.1 Introduction

Based on the work of the previous chapter, a $C$ value needs to be known for displacement reconstruction using a SMS. The measurement method for $C$ has been reported in the literature [17, 31-33, 37]. In 1997, Merlo presented a calibration method to pre-calculate $C$ value [31]. In 2004, Yu [17] proposed a simple and practical method for measuring Line-width Enhancement Factor (LEF) $\alpha$, meanwhile, $C$ value can also be estimated. In 2005, Xi and Yu [37] proposed a gradient-based optimization algorithm to estimate both $C$ and $\alpha$. However, these methods are all restricted to a certain feedback regime, such as weak or moderate regime. And they consider $C$ value as a constant during the measurements. In practice, it is difficult to keep a constant $C$ value during the measurements [39]. In order to obtain real-time $C$ values, Bes [32] developed a signal processing method under moderate feedback regime to jointly estimate $C$, $\alpha$ and displacement using instantaneous power of the reconstructed signal discontinuities. However, the algorithm requires large
computation of SMS samples. In 2009, an improved method based on the work in [33] was proposed in [32]. Both methods in [32] and [33] suffer from noise contamination.

In this chapter, we propose a fast estimation method for $C$ values based on the analysis of the shape of reconstructed waveforms incorporating different pre-set $C$ values. Applying a derivative operation firstly and then following a high-pass filtering, a pulse train is obtained from the waveforms. The magnitude and direction of an impulse can be used to indicate the deviation between the incorporating and the true $C$.

Finally, a bisection searching based algorithm is employed for fast determination of the true $C$ values. The proposed real-time updating for $C$ can further improve the measurement performance for a SMI based displacement sensing system for all feedback cases.

The rest of this chapter is organized as follows: we first present the importance of a $C$ value in the displacement reconstruction in terms of reconstruction accuracy (in section 4.2.1). In section 4.2.2, our real-time $C$ estimation method is described. In section 4.3, we verify our real-time $C$ estimation method in terms of both simulation and experiment. Section 4.4 concludes this chapter.

4.2 Estimation of $C$

4.2.1 Affection of $C$ on Displacement Reconstruction Accuracy

Reconstruction of displacement requires an accurate estimation of parameters $C$
according to (2.2). \( C \) could be varied from 1 to 8 at moderate feedback for a practical SMI based sensing system [34]. Hence, a real-time estimation of \( C \) value is important. By using a pre-set \( C \) for displacement reconstruction, the reconstruction error must be also significant. This can be seen from Figure 4-1. Where a SMS \( (C = 6, \alpha = 3) \) is generated under moderate feedback regime and use a pre-set \( C \) to reconstruct the displacement, the maximum reconstruction error is up to \( 0.3 \mu m \) which is not acceptable in an accurate SMI based sensing.

![Figure 4-1: Reconstruction result by using a pre-set \( C \) \( (C = 2) \). (a) displacement \( L(n) \), (b) a SMS with \( C = 6, \alpha = 3 \), (c) reconstructed displacement \( \hat{L}(n) \) using the pre-set \( C \), (d) residual error between \( L(n) \) and \( \hat{L}(n) \).](image-url)
4.2.2 Real-time Estimation of $C$

According to (2.2), once $\phi_r(n)$ is obtained accurately, we need a $C$ value to reconstruct displacement. If we use an estimation value denoted by $\hat{C}$ to replace true $C$, fluctuations will occur on the reconstructed displacement $\hat{L}(n)$ (as shown in Figure 4-2). Figure 4-2(c) contains saw-tooth like fluctuations with $\hat{C}$ is greater than true $C$, Figure 4-2(d) has step-like fluctuations with $\hat{C}$ is less than true $C$.

![Figure 4-2: Reconstruction results with different estimation value of $\hat{C}$. (a) displacement waveform $L(n)$, (b) a SMS with $C = 3, \alpha = 4$, (c) reconstructed result $\hat{L}(n)$ using a over-estimated $\hat{C}$ ($\hat{C} > 3$), (d) reconstructed result $\hat{L}(n)$ using a down-estimated $\hat{C}$ ($\hat{C} < 3$).](image)

In order to minimize the fluctuations on the reconstruction result, we process the
reconstructed signal by the following steps:

1. Differentiate \( \hat{L}(n) \) to obtain a signal \( D(n) \).
2. Let \( D(n) \) pass a high-pass filter to obtain the fluctuation signal \( D_f(n) \).

Figure 4-3 shows reconstruction results and the fluctuation signals \( D_f(n) \) corresponding to different estimation \( \hat{C} \). From Figure 4-3, it can be seen that when \( \hat{C} \) is approaching the true value of \( C \), the magnitude of \( D_f(n) \) decreases. Also, we can see that signal \( D_f(n) \) for down-estimated case (\( \hat{C} < C \)) is the inverse version for the case of over-estimated (\( \hat{C} > C \)).

![Figure 4-3: Displacement reconstruction results and their corresponding fluctuation signals using deviated \( \hat{C} \). (a) displacement reconstruction results \( \hat{L}(n) \), (b) fluctuation signals \( D_f(n) \).](image)

Given the above analysis, a fast searching algorithm for obtaining an optimal \( C \) (denoted by \( \hat{C}_0 \)) can be implemented as below:
Initiation: setting \( \hat{C} = \frac{(C_{\text{max}} + C_{\text{min}})}{2} \), where \( C_{\text{max}} \) and \( C_{\text{min}} \) are the maximum and minimum value of searching range respectively. By our experience, \( C \) values are mostly in range of \([0.5, 9]\). Calculating \( \phi_i(n) \) by (2.2) using the initial value of \( C \), from it to get \( \hat{L}(n) \) and \( D_F(n) \).

Iteration process:

Step 1: Calculating the mean value (denoted by \( M \)) of \( D_F(n) \) for each left or right inclination segment using (4.1).

\[
M = \frac{1}{R(m+1)-R(m)} \sum_{n=R(m)}^{R(m+1)} D_F(n) \tag{4.1}
\]

Step 2: Updating \( \hat{C} \) by (4.2) – (4.7). The estimation accuracy of \( C \) value is controlled by a pre-setting \( \varepsilon \).

For right inclination segments of a SMS:

\[
\hat{C} = \frac{(\hat{C} + C_{\text{min}})}{2} \quad \text{for} \quad M > \varepsilon \quad (4.2)
\]

\[
\hat{C} = \frac{(C_{\text{max}} + \hat{C})}{2} \quad \text{for} \quad M < \varepsilon \quad (4.3)
\]

\[
\hat{C}_0 = \hat{C} \quad \text{for} \quad |M| < \varepsilon \quad (4.4)
\]

For left inclination segments of a SMS:

\[
\hat{C} = \frac{(C_{\text{max}} + \hat{C})}{2} \quad \text{for} \quad M > \varepsilon \quad (4.5)
\]

\[
\hat{C} = \frac{(\hat{C} + C_{\text{min}})}{2} \quad \text{for} \quad M < \varepsilon \quad (4.6)
\]

\[
\hat{C}_0 = \hat{C} \quad \text{for} \quad |M| < \varepsilon \quad (4.7)
\]

Step 3: Updating \( \phi_i(n) \), \( \hat{L}(n) \) and \( D_F(n) \) using the updating \( C \) value in step 2.

The above iteration is terminated once \( |M| < \varepsilon \). The current value of \( \hat{C} \) is thought as an optimal estimation.
4.3 Performance Evaluation

4.3.1 Simulation Verification

After accurately obtaining the feedback phase \( \phi_r(n) \), we use \( \phi_r(n) \) to estimate \( C \) according to the method described in section 4.2.2, we set the searching range for bisection method as \([0.5, 9]\). The high pass filter is a 3-th high-pass Butterworth filter with cutoff frequency of 500Hz. We firstly test our \( C \) estimation method for pure SMSs. Table 4.1-Table 4.3 show the estimation result \( \hat{C} \) and the iteration steps of the method with different \( \alpha \) when \( \varepsilon = 0.0001 \). Then we test our method for noisy SMSs with \( \alpha = 3 \). We simulated a SMS 10 times with independent additive noise and use the average of the estimated \( \hat{C} \) values as the estimation result (as shown in Table 4.4). Table 4.1 -Table 4.4 show that the proposed estimation of \( C \) is able to produce a \( C \) value with better accuracy and less iteration steps.

<table>
<thead>
<tr>
<th>( C )</th>
<th>0.5</th>
<th>1.5</th>
<th>2.5</th>
<th>3.5</th>
<th>4.5</th>
<th>5.5</th>
<th>6.5</th>
<th>7.5</th>
<th>8.5</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \hat{C} )</td>
<td>0.4980</td>
<td>1.4985</td>
<td>2.4990</td>
<td>3.4980</td>
<td>4.4971</td>
<td>5.4990</td>
<td>6.4980</td>
<td>7.4883</td>
<td>8.4961</td>
</tr>
<tr>
<td>Error</td>
<td>0.0020</td>
<td>0.0015</td>
<td>0.0010</td>
<td>0.0020</td>
<td>0.0029</td>
<td>0.0010</td>
<td>0.0020</td>
<td>0.0117</td>
<td>0.0039</td>
</tr>
<tr>
<td>Iteration steps</td>
<td>9</td>
<td>11</td>
<td>10</td>
<td>7</td>
<td>10</td>
<td>9</td>
<td>8</td>
<td>8</td>
<td></td>
</tr>
</tbody>
</table>
Table 4-2: $C$ estimation results for $\alpha = 3$.

<table>
<thead>
<tr>
<th>$C$</th>
<th>0.5</th>
<th>1.5</th>
<th>2.5</th>
<th>3.5</th>
<th>4.5</th>
<th>5.5</th>
<th>6.5</th>
<th>7.5</th>
<th>8.5</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\hat{C}$</td>
<td>0.4995</td>
<td>1.4995</td>
<td>2.4961</td>
<td>3.4980</td>
<td>4.4971</td>
<td>5.4961</td>
<td>6.4922</td>
<td>7.4941</td>
<td>8.4961</td>
</tr>
<tr>
<td>Error</td>
<td>0.0005</td>
<td>0.0005</td>
<td>0.0039</td>
<td>0.0020</td>
<td>0.0029</td>
<td>0.0039</td>
<td>0.0078</td>
<td>0.0059</td>
<td>0.0039</td>
</tr>
<tr>
<td>Iteration steps</td>
<td>11</td>
<td>9</td>
<td>8</td>
<td>9</td>
<td>10</td>
<td>8</td>
<td>7</td>
<td>9</td>
<td>8</td>
</tr>
</tbody>
</table>

Table 4-3: $C$ estimation results for $\alpha = 5$.

<table>
<thead>
<tr>
<th>$C$</th>
<th>0.5</th>
<th>1.5</th>
<th>2.5</th>
<th>3.5</th>
<th>4.5</th>
<th>5.5</th>
<th>6.5</th>
<th>7.5</th>
<th>8.5</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\hat{C}$</td>
<td>0.5010</td>
<td>1.4985</td>
<td>2.4990</td>
<td>3.4980</td>
<td>4.4971</td>
<td>5.4961</td>
<td>6.4922</td>
<td>7.4883</td>
<td>8.4814</td>
</tr>
<tr>
<td>Error</td>
<td>0.0010</td>
<td>0.0015</td>
<td>0.0010</td>
<td>0.0020</td>
<td>0.0029</td>
<td>0.0039</td>
<td>0.0078</td>
<td>0.0117</td>
<td>0.0186</td>
</tr>
<tr>
<td>Iteration steps</td>
<td>10</td>
<td>11</td>
<td>10</td>
<td>9</td>
<td>10</td>
<td>8</td>
<td>7</td>
<td>8</td>
<td>10</td>
</tr>
</tbody>
</table>

Table 4-4: $C$ estimation results for noisy SMSs with $\alpha = 3$.

<table>
<thead>
<tr>
<th>Feedback level</th>
<th>Weak</th>
<th>Moderate</th>
<th>Strong</th>
</tr>
</thead>
<tbody>
<tr>
<td>$C$</td>
<td>0.5</td>
<td>0.7</td>
<td>1.4</td>
</tr>
<tr>
<td>$\hat{C}$</td>
<td>0.5197</td>
<td>0.6431</td>
<td>0.8590</td>
</tr>
<tr>
<td>Std Dev.%</td>
<td>0.97</td>
<td>1.1</td>
<td>1.92</td>
</tr>
</tbody>
</table>
4.3.2 Experiment Verification

Using the experimental set-up described in chapter 2, a SMS was acquired to test the $C$ estimation method. By firstly using the phase unwrapping method described in chapter 3 to obtain $\phi_p(n)$ and then perform real-time $C$ estimation method, Figure 4-4 shows the estimation result. It is shown that the fluctuations have been greatly reduced by using the estimated $\hat{C}$ values.

![Figure 4-4: Reconstruction result for an experimental SMS.](image)

Figure 4-4: Reconstruction result for an experimental SMS. (a) an experimental SMS, (b) estimation result $\hat{C}$, (c) reconstructed displacement by using $\hat{C}$ obtained in (b).
4.4 Summary

Considering time varying optical feedback level factor $C$ needed to be known to reconstruct displacement, in this chapter, real-time $C$ estimation method was presented, to ensure the final measurement performance. By using advanced signal processing method, we can estimate time-varying $C$ values rapidly based on the analysis of the features of reconstructed waveform incorporating different pre-set $\hat{C}$ values. Simulation results show that our method for estimating $C$ value is fast and robust for all feedback cases.
Chapter 5  System Performance Test

5.1 Overall Implementation of SMI Based Displacement Reconstruction Algorithm

Combining the phase unwrapping method described in Chapter 3 and the real-time $C$ estimation method (Chapter 4), the overall implementation of SMI based displacement reconstruction algorithm can be summarized using a block diagram shown in Figure 5-1. The diagram contains three parts.

1. To perform phase unwrapping algorithm on SMSs to obtain feedback phase $\phi_f(n)$.
2. Real-time $C$ estimation method.
3. Calculation of displacement using (2.1).
Figure 5-1: Block diagram of improved displacement reconstruction algorithm.

In order to test the effectiveness of our displacement reconstruction algorithm, we applied the method to the same SMSs which was used in section 3.5.2. Figure 5-2 – Figure 5-4 show the displacement reconstruction result. In each figure, reconstruction result with our real-time estimated $C$ values and with a pre-set $C$ value (pre-set $C=5$) are presented. From the figures, we can see that by using the proposed displacement reconstruction algorithm with real-time $C$ estimation, a smooth displacement $\hat{L}(n)$ can be achieved under three different feedback levels. However, if we use a pre-set $C$ value to reconstruct displacement, the reconstructed result $\hat{L}(n)$ contains fluctuations.
Figure 5-2: Displacement reconstruction results under weak feedback level. (a) an experimental SMS, (b) displacement reconstruction results cooperating estimated $\hat{C}$, (c) displacement reconstruction results with a pre-set $C$ ($C = 5$).

Figure 5-3: Displacement reconstruction results under moderate feedback level. (a) an experimental SMS, (b) displacement reconstruction results cooperating estimated $\hat{C}$, (c) displacement reconstruction results with a pre-set $C$ ($C = 5$).
Figure 5-4: Displacement reconstruction results under strong feedback level. (a) an experimental SMS, (b) displacement reconstruction results cooperating estimated \( \hat{C} \), (c) displacement reconstruction results with a pre-set \( C \) (\( C = 5 \)).

5.2 Comparison with A Commercial Sensor

To verify our proposed SMI based displacement reconstruction algorithm, we use a commercial sensor (as shown in Figure 5-5) which is provided by MTI instruments (LTC-025-04-SA) to calibrate our SMI sensing system.
The experimental setup for our SMI based sensing system is shown in Chapter 2 (Figure 2-1). The driving signal for the loudspeaker is a sinusoidal signal with a frequency of 75\textit{Hz}. To obtain SMSs under different feedback levels, we adjust the distance between target and laser facet. Before we perform our displacement reconstruction algorithm, we pre-process the acquired SMSs by using the method described in Chapter 2. After pre-processing, we apply our displacement reconstruction algorithm on the acquired SMSs. The second row of Figure 5-6 – Figure 5-8 is the SMS after pre-processing. The measurement results of displacement by using the commercial sensor and our SMI based sensing system are shown in the third and fourth row of Figure 5-6 – Figure 5-8 respectively.
Figure 5-6: Displacement measurement results under weak feedback level. (a) a SMS acquired from our SMI system, (b) SMS after pre-processing, (c) displacement measured by the commercial sensor, (d) reconstructed displacement by the proposed method.

Figure 5-7: Displacement measurement results under moderate feedback level. (a) a SMS acquired from our SMI system, (b) SMS after pre-processing, (c) displacement measured by the commercial
sensor, (d) reconstructed displacement by the proposed method.

Figure 5-8: Displacement measurement results under strong feedback level. (a) a SMS acquired from our SMI system, (b) SMS after pre-processing, (c) displacement measured by the commercial sensor, (d) reconstructed displacement by the proposed method.

From the results, it is clear to see that displacement measured by the commercial sensor is distorted and some parts of the sinusoidal waveform are truncated. However, the displacement reconstructed by using our algorithm well reveals the characteristics of the movement of target.

5.3 Summary

In this chapter, we described our overall SMI based displacement reconstruction...
algorithm in terms of a block diagram. To verify the displacement reconstruction algorithm, we applied our algorithm on three SMSs under different feedback levels. The results show that displacement reconstruction with real-time $C$ estimation method is better than using a pre-set $C$ value. In this chapter, we also complete the comparison by using a commercial sensor to verify our proposed displacement reconstruction algorithm.
Chapter 6  Conclusion and Future work

Self-Mixing Interferometry (SMI) based displacement sensing has been being an active research topic in the area of opto-electronic instrumentation and measurement. In this thesis, SMI based displacement reconstruction algorithm is proposed for achieving accurate displacement measurement. The algorithm consists of phase unwrapping and real-time $C$ estimation. To apply the proposed algorithm on practical a SMS, signal pre-processing method is required. The pre-processing includes filters and normalization. The effectiveness of the proposed approach is tested by a commercial sensor. In this chapter, the research contributions are summarized in section 6.1. Based on the studies in the thesis, future research topics are drawn and listed in section 6.2.

6.1 Research Contributions

According to extensive computer simulations and experimental results, the following contribution of this thesis can be stated:

- A novel phase unwrapping algorithm was developed which is suitable for all feedback cases. The proposed phase unwrapping algorithm can theoretically
achieve an unbiased feedback phase of a SMS.

- A real-time $C$ estimation method is proposed. The proposed estimation of $C$ is able to produce a $C$ value with better accuracy and less iteration steps. That means the proposed method is fast and robust.

- Both computer and experimental results show that sparkle-like noise and white-like noise can be eliminated from SMSs effectively by using a median filter and a moving average filter respectively.

- A novel normalization method for normalizing SMSs is proposed. This is the first time that a discussion on this issue is given. The proposed normalization method can normalize SMSs under all feedback levels. And it is also effective for removing slow-time fluctuation of the envelope of a SMS.

### 6.2 Suggested Future Research Topics

For further improving the performance of SMI based displacement sensing system, conducting the following research topics will be helpful:

- Development of an adaptive threshold algorithm for obtaining the pulse train $E(n)$ (seen section 3.4.1 in Chapter 3). A proper threshold can increase the counting accuracy for the pulse train.

- Development of an appropriate fringe interpolation method for displacement reconstruction. Doing so, the estimation of $C$ can be removed from the measuring procedure.
Development of better signal pre-processing method for a practical SMS. This includes:

- Removing slow fluctuation contained in a SMS.
- Increasing SNR.

Fringe loss phenomenon occurred in strong feedback level [34] should be considered when reconstructing displacement. This required to do:

- Identify the fringe loss case.
- Modify the phase unwrapping rules.
Appendix A: Programming list

A.1 Code for generating SMS

```matlab
function Gv=simulationG(NoiseLevel, LengthG, MainFrequency,samplingFrequency,vibrationAmp,initialLocation,alpha,feedbacklevel)
alpha =alpha;c =feedbacklevel;
am=vibrationAmp; am0=initialLocation;
ft = MainFrequency;
n=LengthG;
sfs =samplingFrequency;

%% -------- generating the ideal SMI signals
%% --------%y:phasef,x=phase0+constant, g:self-mixing
t1=[0:n-1]; % time index
phi_0=am0+am*sin(2*pi*ft/fs*t1); % the phase without external feedback
am0>am
```
\[ x = \phi_0 + \arctan(\alpha); \quad \% \text{introduce } x \text{ to simplify calculation} \]
\[ y = \text{zeros}(1,n); \quad \% \text{memory locating for solutions, where } y = \phi_f + \arctan(\alpha) \]

\[ \text{precision} = 10000; \quad \% \text{number of steps for searching within } y_{\text{range}} \]
\[ e = \text{zeros}(1, \text{precision} + 1); \]
\[ y y = e; \]
\[ y y y = \text{zeros}(1, 30); \quad \% \text{assume there are maximally 30 solutions for the phase equation} \]

\[ \% \text{this part determine the initial } y(1) \text{ by solving the phase equation when } x(1) \text{ is given} \]
\[ y y = x(1) - c + 2 \cdot c \cdot [1: \text{precision}] / \text{precision}; \quad \% \text{make the possible solutions within } [x(1) - c, x(1) + c] \]
\[ e = y y - x(1) + c \cdot \sin(y y); \quad \% y y = y_{\text{max}} \cdot [1: \text{precision}] / \text{precision}; \]
\[ e = \text{sign}(e); \quad \% \text{set } p \text{ and } N, \text{ determining the jump position} \]
\[ e = \text{diff}(e); \]
\[ e = \text{sign}(e); \]
\[ e = \text{abs}(e); \quad \% e \text{ is either 1 or 0, jump part is 1, no jump part is zero. by 1 determine the jumping position} \]
\[ i i i = 1; \]
\[ \text{for } i i = 1: \text{precision} - 1 \quad \% \text{check the values of } e(y) \]
\[ \quad \text{if } (1 + c \cdot \cos(y y(i i))) > 0; \quad \% \text{disgard the solutions with negative gradient} \]
\[ \quad \text{if } e(i i) == 1; y y y(i i i) = y y(i i); i i i = i i i + 1; \quad \% \text{solutions are these y y with zero } e(i i), \text{stable solutions r in y y y by increasing trend} \]
\[ \quad \text{end} \]
\[ \text{end} \]

\[ \% \text{number_of_roots} = i i i - 1; \]
\[ y(1) = y y y(1); \quad \% \text{choose the smallest solution as the initial } y(1). \text{ this will not affect the steady state behavior} \]

\[ \% \text{the following part for solving for } y \text{ for a given } x(i) \]
\[ y_r = \arccos(-1/c); \quad \% \text{the point where } y \text{ jumps when } x \text{ increases and reaches} \]
\[ x_r \]
\[ y_d = 2 \cdot \pi - \arccos(-1/c); \quad \% \text{the point where } x \text{ drops when } x \text{ decreases and reached} \]
\[ x_d \]
\[ x_r = y_r + \sqrt{c \cdot c - 1}; \]
\[ x_d = y_d + \sqrt{c \cdot c - 1}; \]
\[ \text{for } i = 2:n \]
\[ y y = x(i) - c + 2 \cdot c \cdot [1: \text{precision}] / \text{precision}; \quad \% y y \text{ -- } [x(i) - c, x(i) + c] \]
\[ e = yy - x(i) + c \cdot \sin(yy); \quad \text{\% } yy - [-y_{rang}/2, y_{rang}/2] \]
\[ e = \text{sign}(e); \]
\[ e = \text{diff}(e); \]
\[ e = \text{sign}(e); \]
\[ e = \text{abs}(e); \quad \text{\% } e \text{ is either 1 or 0} \]

\[ iii = 1; \]

\[ \text{for } ii = 1: \text{precision}-1 \quad \text{\% check the values of } e(y) \]
\[ \quad \text{if } (1 + c \cdot \cos(yy(ii))) > 0; \quad \text{\% discard the solutions with negative} \]
\[ \quad \quad \text{gradient} \]
\[ \quad \quad \text{if } e(ii) == 1; \quad yyy(iii) = yy(ii); \quad iii = iii + 1; \quad \text{end} \quad \text{\% solutions are saved in } yyy \]
\[ \quad \text{end} \]
\[ \text{end} \]
\[ \text{number_of_roots} = iii - 1; \]

\[ \text{\% the following part determine which solution is the true one based on} \]
\[ \text{linewidth mode selection principle} \]
\[ \text{\% the mode is selected based on the following rules:} \]
\[ \text{\% 1. } y \text{ varies nomotically with } x; \]
\[ \text{\% 2. } y \text{ tries to keep continuity whenever possible;} \]
\[ \text{\% 3. In case of jumping and drooping, the mode with least linewidth} \]
\[ \text{will be selected} \]

\[ ddx = x(i) - x(i-1); \]
\[ ddy = \text{abs}(yyy(1) - y(i-1)); \quad \text{\% distance of the smallest solution to previous} \]
\[ y(i-1), \quad yyy(1) \text{ is the smallest solution} \]
\[ y0 = yyy(1); \]
\[ \text{for } iii = 2: \text{number_of_roots}; \]
\[ \quad dddy = \text{abs}(yyy(iii) - y(i-1)); \quad \text{\% distance of } yyy(i) \text{ to } y(i) \]
\[ \quad \text{if } dddy < ddy; \quad y0 = yyy(iii); \quad ddy = dddy; \quad \text{end}; \]
\[ \text{end} \]
\[ \\text{\% Now } y0 \text{ is the solution closest to } y(i-1) \]

\[ \text{\% The following part determine the solution when jump or drop happens.} \]
\[ \text{\% The one with smallest distance to } x \text{ will be chosen} \]
\[ \text{if } \text{abs}(y0 - y(i-1)) > \pi/4; \quad \text{\% condition for detecting the jumping} \]
\[ \text{points, jump threshold.} \]
\[ dyx = \text{abs}(yyy(1) - x(i)); \]
\[ \text{for } iii = 1: \text{number_of_roots}; \]
\[ \quad ddyx = \text{abs}(yyy(iii) - x(i)); \]
\[ \quad \text{if } ddyx < dyx; \quad y0 = yyy(iii); \quad dyx = ddyx; \quad \text{end} \]
\[ \text{end} \]
y(i)=y0;
end;
y=y-atan(alfa);
g=cos(y);
Gv={g,y,phi_0}; %%%return GV three signals, they are g:self-mixing signal, y:phaseF and phi_0: phase_F

A.2 Phase unwrapping Algorithm

%-------------------------------------------------------------
% The following program is to implement
% our phase unwrapping algorithm
%-------------------------------------------------------------

close all; clear all;
clc;
% -- The generating of the simulating signal
NoiseLevel=0.;
alpha=3; feedbacklevel=3;
vibrationAmp=7*pi;
initialLocation=3.9*10^6;
MainFrequency=100;
samplingFrequency=25600;
fs_0=MainFrequency/samplingFrequency; % normalised vibration frequency
n0=1/fs_0;
LengthG=2*round(n0); % always get data length with two period
Gv=simulationG(NoiseLevel, LengthG, 
MainFrequency, samplingFrequency, vibrationAmp, initialLocation, alpha, feedbacklevel);
g=Gv(1);  % simulated Self-Mixing Signal
phase_F=Gv(2)-round(initialLocation/(2*pi))*2*pi; % simulated phase_F
phi_0=Gv(3)-round(initialLocation/(2*pi))*2*pi; % simulated phi_0

%% -- The gnerating of Reconstructed phase_F
%--------------------------------------parameters-------------------------------
% alpha ---- LEF
%%% g ---- Self-mixing signal
%%% Reconstructed_phase_F ---- reconstructed phase_F
%%% Reverse_points ---- the points where external target changes direction.
%%% m1 ---- either 1 or 0, most of cases, when C>2 m1=1;
%%% threshold1, threshold2 ---- to cut the differentiate signal to find fringe
%%%--------------------------------------------------------%
threshold1=0.5;
threshold2=0.5;
m1=1;
[Reconstructed_phase_F,Reverse_points]=Reconstruct_phase_F(g,threshold1,threshold2,m1);

%%%--------------------------------------------------------%
%% This is the function for phase unwrapping
%% Reconstruct_phase_F
%%--------------------------------------------------------%

function [Reconstructed_phase_F,Reverse_points]=Reconstruct_phase_F(g,threshold1,threshold2,m1)
%%%----------------------parameters-----------------------%
%%% alpha- LEF
%%% g - Self-mixing signal
%%% Reconstructed_phase_F - reconstructed phase_F
%%% Reverse_points - the points where external target changes direction.
%%% m1 - either 1 or 0, most of cases, when C>2 m1=1;
%%% threshold1, threshold2 - to cut the differentiate signal to find fringe
%%%--------------------------------------------------------%

ng=length(g);
h1=diff(g);
th1=threshold1*max(h1);
th2=threshold2*min(h1);
for i=1:length(h1)
    if h1(i)<th1 && h1(i)>th2
        h1(i)=0;
    end
end

control=zeros(1,ng); % control is a signal used to extract Reverse segment
I=find(h1);
for k=1:length(I)-1
    x=h1(I(k))*h1(I(k+1));
    if x<0
        for i=I(k)+1:I(k+1)
            control(i)=1;
        end
    else
        for i=I(k):I(k+1)
            control(i)=0;
        end
    end
end

x=find(control==1);
%%% temp is the front edge of the control signal
count=1;
for i=1:length(x)-1
    z=x(i+1)-x(i);
    if z==1
        count=count+1;
        temp(count)=x(i+1);
    end
end

y=find(control==0);
count=0;
for i=1:length(y)-1
    z=y(i+1)-y(i);
    if z==1
        count=count+1;
        a(count)=y(i+1)-y(i)-1;
        %%% a is the width of the control signal
    end
end

for i=1:length(control)
    result(i)=control(i).*g(i);  % result is the reverse segment
end

%---------------------------------------------------------------
% This part is to determine reverse points
% All reverse points are stored in reverse_point
for i=1:length(temp)
    if h1(temp(i)-1)<0
        [value,index1]=max(g(temp(i):(temp(i)+a(i)-1)));
index1=index1+temp(i)-1;
if abs(g(index1))>0.98
    [value,index1]=max(g(temp(i):(temp(i)+a(i)-1)));    
    index1=index1+temp(i)-1;
    [value,index2]=max(g(temp(i):index1-5));
    index2=index2+temp(i)-1;
    [value,index3]=max(g(index1+5:(temp(i)+a(i)-1)));    
    index3=index3+index1+5-1;
    if abs(index1-index2)>10
        peak1(i)=index2;
        peak2(i)=index1;
        [value,index4]=min(g(index2:index1));
        reverse_point(i)=index4+index2-1;
    else
        peak1(i)=index1;
        peak2(i)=index3;
        [value,index4]=min(g(index1:index3));
        reverse_point(i)=index4+index1-1;
    end
elseif abs(g(index1))<0.98
    [value,index1]=max(g(temp(i):temp(i)+a(i)-1));
    reverse_point(i)=index1+temp(i)-1;
    peak1=0;
    peak2=0;
elseif a(i)<30
    [value,index1]=max(g(temp(i):temp(i)+a(i)-1));
    reverse_point(i)=index1+temp(i)-1;
    peak1=0;
    peak2=0;
end
elseif h1(temp(i)-1)>0
    [value,index1]=min(g(temp(i):(temp(i)+a(i)-1)));    
    index1=index1+temp(i)-1;
    if abs(g(index1))>0.985
        [value,index1]=min(g(temp(i):(temp(i)+a(i)-1)));    
        index1=index1+temp(i)-1;
        [value,index2]=min(g(temp(i):index1-5));
        index2=index2+temp(i)-1;
        [value,index3]=min(g(index1+5:(temp(i)+a(i)-1)));    
        index3=index3+index1+5-1;
        if abs(index1-index2)>5
            bottom1(i)=index2;
            bottom2(i)=index1;
        end
    end
end
[value, index4] = max(g(index2:index1));
reverse_point(i) = index4 + index2 - 1;
else
  bottom1(i) = index1;
  bottom2(i) = index3;
  [value, index4] = max(g(index1:index3));
  reverse_point(i) = index4 + index1 - 1;
end

elseif abs(g(index1)) < 0.985
  [value, index1] = min(g(temp(i) + 5:temp(i) + a(i) - 1 - 5));
  reverse_point(i) = index1 + temp(i) + 5 - 1;
  bottom1 = 0;
  bottom2 = 0;
elseif a(i) < 30
  [value, index1] = max(g(temp(i):temp(i) + a(i) - 1));
  reverse_point(i) = index1 + temp(i) - 1;
  bottom1 = 0;
  bottom2 = 0;
end

end
end

peak = [peak1 peak2];
peak = sort(peak);
bottom = [bottom1 bottom2];
bottom = sort(bottom);
for i = 1:length(peak) - 1
  if peak(i + 1) - peak(i) == 0
    peak(i) = 0;
  else
    peak(i) = peak(i);
  end
end
q1 = find(peak == 0);
qq1 = peak(q1);
for i = 1:length(qq1)
  for j = 1:length(reverse_point)
    if qq1(i) == reverse_point(j)
      qq1(i) = 0;
    else
      qq1(i) = qq1(i);
    end
  end
end
end
q1=find(q1==0);
qqq1=qq1(q1);

for i=1:length(bottom)-1
    if bottom(i+1)-bottom(i)==0
        bottom(i)=0;
    else
        bottom(i)=bottom(i);
    end
end
q2=find(bottom==0);
qq2=bottom(q2);
for i=1:length(qq2)
    for j=1:length(reverse_point)
        if qq2(i)==reverse_point(j)
            qq2(i)=0;
        else
            qq2(i)=qq2(i);
        end
    end
end
q2=find(qq2==0);
qqq2=qq2(q2);

%--------------------------------------------------------------
% This part is to find all the peak points and valley points
% function Find_extreme_points is a sub-function used for
% locating characteristic points

n=length(reverse_point);
r=reverse_point;
I1=find(h1(1:r(1))~=0);
I2=find(h1(r(n-1):r(n))~=0);
I2=I2+r(n-1)-1;

wave=diff(sign(h1));
[zmax,zmin]=Find_extreme_points(h1,reverse_point,wave,g);
if c>1.7
    zmax=zmax;
else
    zmax=[zmax qqq1];
end
zmax=sort(zmax);
zmin=[zmin qqq2];
zmin=sort(zmin);
w=zeros(1,ng);

q1=find(zmax~=0);
zmax=zmax(q1);
q2=find(zmin~=0);
zmin=zmin(q2);

q1=find(zmax~=1);
zmax=zmax(q1);
q2=find(zmin~=1);
zmin=zmin(q2);

%------------------------------------------------------------------------%
% The following part is to perform our phase unwrapping algorithm

recover=abs(acos(g));
m2=0;

q1=zeros(1,length(g));
q2=zeros(1,length(g));
q3=zeros(1,length(g));
for j=1:2:n-1
    q1(r(j):r(j+1))=1;
end
q2(1:r(1))=1;
for j=2:2:n-1
    q2(r(j):r(j+1))=1;
end

if h1(I2)>0
    q3(r(n):length(g))=1;
else
    q3(r(n):length(g))=-1;
end

lmax=length(zmax);
for i=1:lmax-1
    if q2(zmax(i))==1 || q3(zmax(i))==1
        temp1=sqrt((recover(zmax(i))-recover(zmax(i)-1))^2+1);
        temp2=sqrt((recover(zmax(i))-recover(zmax(i)+1))^2+1);
        if temp1>temp2
            zmax(i)=zmax(i)+1;
end
end

lmin=length(zmin);
for i=1:lmin
    temp1=sqrt((recover(zmin(i))-recover(zmin(i)-1))^2+1);
    temp2=sqrt((recover(zmin(i))-recover(zmin(i)+1))^2+1);
    if temp1>temp2
        zmin(i)=zmin(i)+1;
    end
end

w(zmax)=-1;
w(zmin)=1;

for i=1:ng
    if h1(I1)<0
        if i==r
            recover(i)=((-1)^m1)*recover(i)+m2*2*pi;
        elseif w(i)==-1 && q2(i)==1
            m1=m1+1;
            recover(i)=((-1)^m1)*recover(i)+m2*2*pi;
        elseif w(i)==1 && q2(i)==1
            m1=m1+1;
            m2=m2+1;
            recover(i)=((-1)^m1)*recover(i)+m2*2*pi;
        elseif w(i)==-1 && q1(i)==1
            m1=m1+1;
            recover(i)=((-1)^m1)*recover(i)+m2*2*pi;
        elseif w(i)==1 && q1(i)==1
            m1=m1+1;
            m2=m2-1;
            recover(i)=((-1)^m1)*recover(i)+(m2+1)*2*pi;
        elseif w(i)==1 && q3(i)==1
            m1=m1+1;
            m2=m2+1;
            recover(i)=((-1)^m1)*recover(i)+m2*2*pi;
        elseif w(i)==-1 && q3(i)==1
            m1=m1+1;
            recover(i)=((-1)^m1)*recover(i)+m2*2*pi;
        elseif w(i)==-1 && q3(i)==-1
            m1=m1+2;
            m2=m2-1;
recover(i)=((-1)^m1)*recover(i)+m2*2*pi;
else
    recover(i)=((-1)^m1)*recover(i)+m2*2*pi;
end
else
    if i==r
        recover(i)=((-1)^m1)*recover(i)+m2*2*pi;
    elseif w(i)==-1 && q1(i)==1
        m1=m1+1;
        recover(i)=((-1)^m1)*recover(i)+m2*2*pi;
    elseif w(i)==1 && q1(i)==1
        m1=m1+1;
        m2=m2+1;
        recover(i)=((-1)^m1)*recover(i)+m2*2*pi;
    elseif w(i)==-1 && q2(i)==1
        m1=m1+2;
        m2=m2-1;
        recover(i)=((-1)^m1)*recover(i)+m2*2*pi;
    elseif w(i)==1 && q3(i)==1
        m1=m1+1;
        m2=m2+1;
        recover(i)=((-1)^m1)*recover(i)+m2*2*pi;
    elseif w(i)==-1 && q3(i)==1
        m1=m1+1;
        recover(i)=((-1)^m1)*recover(i)+m2*2*pi;
    elseif w(i)==-1 && q3(i)==-1
        m1=m1+2;
        m2=m2-1;
        recover(i)=((-1)^m1)*recover(i)+m2*2*pi;
    else
        recover(i)=((-1)^m1)*recover(i)+m2*2*pi;
    end
end
Reconstructed_phase_F=recover;

%%%-------------------------------------------------------% %
%%% This is the function for finding characteristic points
%%% Find_extreme_points
%%%-------------------------------------------------------% %
function [zmax,zmin]=Find_extreme_points(h1,reverse_point,ave,g)
n=length(reverse_point);
r=reverse_point;
```matlab
I1 = find(h1(1:r(1)) ~= 0);
I2 = find(h1(r(n-1):r(n)) ~= 0);
I2 = I2 + r(n-1) - 1;

% main left and right inclination segment
if h1(I1) > 0
    a = 1; b = 2;
else
    a = 2; b = 1;
end
count = 0;
for j = a:2:n-1
    x = find(wave(r(j):r(j+1)) == -1);
    x = x + r(j) + 1;
    y = length(x);
    zmax1 = zeros(1, y);
    [value, zmax1(1)] = max(g(r(j):x(1)));
    zmax1(1) = zmax1(1) + r(j) - 1;
    for i = 2:y
        [value, zmax1(i)] = max(g(x(i-1)+5:x(i)));
        zmax1(i) = zmax1(i) + x(i-1) + 5 - 1;
    end
    zmin1 = zeros(1, y);
    for i = 1:y-1
        [value, zmin1(i)] = min(g(zmax1(i):zmax1(i+1)));
        zmin1(i) = zmin1(i) + zmax1(i) - 1;
    end
    [value, zmin1(y)] = min(g(zmax1(y):r(j+1)));
    zmin1(y) = zmin1(y) + zmax1(y) - 1;
    count = count + 1;
    z1(zmax1) = zmax1;
    q1 = find(z1 == 0);
    z2(zmin1) = zmin1;
    q2 = find(z2 == 0);
end

count = 0;
for j = b:2:n-1
    x = find(wave(r(j):r(j+1)) == 1);
    x = x + r(j) + 1;
    y = length(x);
    zmin3 = zeros(1, y);
    [value, zmin3(1)] = min(g(r(j):x(1)));
    zmin3(1) = zmin3(1) + r(j) - 1;
```
for i=2:y
    [value, zmin3(i)] = min(g(x(i-1)+5:x(i)));  
zmin3(i) = zmin3(i) + x(i-1) + 5 - 1;
end
zmax3 = zeros(1, y);
for i=1:y-1
    [value, zmax3(i)] = max(g(zmin3(i):zmin3(i+1)));  
zmax3(i) = zmax3(i) + zmin3(i) - 1;
end
[value, zmax3(y)] = max(g(zmin3(y):r(j+1)));  
zmax3(y) = zmax3(y) + zmin3(y) - 1;
count = count + 1;
z3(zmax3) = zmax3;
q3 = find(z3 == 0);
z4(zmin3) = zmin3;
q4 = find(z4 == 0);
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%front else part
if h1(I1) > 0
    x = find(wave(1:r(1)) == 1);
    x = x + 1 + 1;
y = length(x);
zmin2 = zeros(1, y);
    [value, zmin2(1)] = min(g(1:x(1)));  
zmin2(1) = zmin2(1) + 1 - 1;
for i=2:y
    [value, zmin2(i)] = min(g(x(i-1)+5:x(i)));  
zmin2(i) = zmin2(i) + x(i-1) + 5 - 1;
end
zmax2 = zeros(1, y);
for i=1:y-1
    [value, zmax2(i)] = max(g(zmin2(i):zmin2(i+1)));  
zmax2(i) = zmax2(i) + zmin2(i) - 1;
end
[value, zmax2(y)] = max(g(zmin2(y):r(1)));  
zmax2(y) = zmax2(y) + zmin2(y) - 1;
else
    x = find(wave(1:r(1)) == -1);
    x = x + 1 + 1;
y = length(x);
zmax2 = zeros(1, y);
    [value, zmax2(1)] = max(g(1:x(1)));  
zmax2(1) = zmax2(1) + 1 - 1;
for i=2:y
    [value, zmax2(i)] = max(g(x(i-1)+5:x(i)));  
zmax2(i) = zmax2(i) + x(i-1) + 5 - 1;
end
```matlab
[value, zmax2(i)] = max(g(x(i-1)+5:x(i))); 
zmax2(i) = zmax2(i) + x(i-1) + 5 - 1;
end
if zmax2(1) == 1;
    zmax2(1) = 0;
    m1 = find(zmax2 == 0);
    zmax2 = zmax2(m1);
y = y - 1;
end
zmin2 = zeros(1, y+1);
for i = 1:y-1
    [value, zmin2(i)] = min(g(zmax2(i):zmax2(i+1))); 
zmin2(i) = zmin2(i) + zmax2(i) - 1;
end
[value, zmin2(y)] = min(g(zmax2(y):r(1))); 
zmin2(y) = zmin2(y) + zmax2(y) - 1;
if zmin2(1) <= 5
    m1 = find(zmin2 == 0);
    zmin2 = zmin2(m1);
else
    [value, zmin2(y+1)] = min(g(1:zmin2(1)-5)); 
zmin2(y+1) = zmin2(y+1) + 1 - 1;
    if abs(zmin2(y+1) - zmin2(1)) > 10
        zmin2(y+1) = zmin2(y+1);
    else
        zmin2(y+1) = 0;
    end
    m1 = find(zmin2 == 0);
    zmin2 = zmin2(m1);
end
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% back else part
if h1(I2) > 0
    x = find(wave(r(n):length(wave)) == -1);
    x = x + r(n) + 1;
    y = length(x);
    zmax4 = zeros(1, y+1);
    [value, zmax4(1)] = max(g(r(n):x(1))); 
zmax4(1) = zmax4(1) + r(n) - 1;
    for i = 2:y
        [value, zmax4(i)] = max(g(x(i-1)+5:x(i))); 
zmax4(i) = zmax4(i) + x(i-1) + 5 - 1;
    end
```
if length(g)-zmax4(y)<=5
    m1=find(zmax4==0);
    zmax4=zmax4(m1);
else
    [value,zmax4(y+1)]=max(g(zmax4(y)+5:length(g)));
    zmax4(y+1)=zmax4(y+1)+zmax4(y)+5-1;
    if abs(zmax4(y+1)-zmax4(y))>10
        zmax4(y+1)=zmax4(y+1);
    else
        zmax4(y+1)=0;
    end
    m2=find(zmax4~=0);
    zmax4=zmax4(m2);
end

zmin4=zeros(1,y);
for i=1:y-1
    [value,zmin4(i)]=min(g(zmax4(i):zmax4(i+1)));
    zmin4(i)=zmin4(i)+zmax4(i)-1;
end
[value,zmin4(y)]=min(g(zmax4(y):length(wave)));
else
    x=find(wave(r(n):length(wave))==1);
    x=x+r(n)+1;
    y=length(x);
    zmin4=zeros(1,y);
    [value,zmin4(1)]=min(g(r(n):x(1)));
    zmin4(1)=zmin4(1)+r(n)-1;
    for i=2:y
        [value,zmin4(i)]=min(g(x(i-1)+5:x(i)));
        zmin4(i)=zmin4(i)+x(i-1)+5-1;
    end
    zmax4=zeros(1,y+1);
    for i=1:y-1
        [value,zmax4(i)]=max(g(zmin4(i):zmin4(i+1)));
        zmax4(i)=zmax4(i)+zmin4(i)-1;
    end
    [value,zmax4(y)]=max(g(zmin4(y):length(wave)));
    zmax4(y)=zmin4(y)+2;
end

zmax=[q1,q3,zmax2,zmax4];
zmax=sort(zmax);
zmin=[q2,q4,zmin2,zmin4];
zmin=sort(zmin);

A.3 Feedback level factor C estimation

% -- The generating of Reconstructed phi_0 and estimated C value
%%%----------------------parameters-----------------------%
%%% a , b ---- boundary of the searching range
%%% alpha ---- LEF
%%% samplingFrequency ---- used to determine the cutoff frequency of high
%%% pass filter
%%% Reconstructed_phase_F ---- reconstructed phase_F
%%% Reverse_points ---- the points where external target changes direction.
%%% Estimated_C ---- the C value estimated by bisection method
%%% Reconstructed_phi_0 ---- reconstructed phi_0 using the estimated C
%%% value
%%%--------------------------------------------------------%
a=0; b=10;
[Reconstructed_phi_0,Estimated_C]=Estimate_C_value(a,b,samplingFrequency,Reconstructed_phase_F,alpha,Reverse_points);

%---------------------------------------------%
% This is the matlab function for C estimation
% Estimate_C_value
%---------------------------------------------%
function [Reconstructed_phi_0,Estimated_C]=Estimate_C_value(a,b,samplingFrequency,Reconstructed_phase_F,alpha,Reverse_points)
c=(a+b)./2;
r=Reverse_points;
ng=length(Reconstructed_phase_F);
[B,A]=butter(3,500/(samplingFrequency/2),'high');
d=round(ng*0.04);

for i=0:20
    phi=Reconstructed_phase_F+c*sin(Reconstructed_phase_F+atan(alpha));
    %z=smooth(phi);
    h=diff(phi);
```matlab
%h=medfilt1(h,5);
y=filter(B,A,h);
threshold1=max(y(r(1)+d:r(2)-d));
I1=find(y(r(1)+d:r(2)-d)>0.2*threshold1);
I1=I1+r(1)+d-1;
peak=mean(y(I1));
threshold2=min(y(r(1)+d:r(2)-d));
I2=find(y(r(1)+d:r(2)-d)<0.2*threshold2);
I2=I2+r(1)+d-1;
bottom=mean(y(I2));

if abs(peak)<abs(bottom)
a=c;
c=(a+b)/2;
else
b=c;
c=(a+b)/2;
end
end
Estimated_C=c;
Reconstructed_phi_0=Reconstructed_phase_F+Estimated_C*sin(Reconstructed_phase_F+atan(alpha));

%%%---------------------------------------------%
%%% This is the generating plot of simulated and
%%% reconstructed phi_0 and phase_F
%%%---------------------------------------------%

%%% --The generating plot of simulated and reconstructed phi_0 and phase_F
subplot(511)
plot(g)   % plot the self-mixing signal
h=title('The Self-Mixing Signal g(t)');
set(h,'fontsize',14)
axis([1 length(g)-1 -1.2 1.2])

subplot(512)
plot(phi_0)   % plot the simulated phi_0
h=title('The Simulated phi_0');
set(h,'fontsize',14)
axis([1 length(phi_0)-1 min(phi_0)-5 max(phi_0)+5])
```

```matlab
subplot(513)
```
A.4 Normalization

%%%%---------------------------------------------%
%%%% This program is to normalize Type 1 SMS
%%%% parameters: h1 - E(n)
%%%%             zmax - peak points
%%%%             zmin - valley points
%%%%             g - SMS
%%%%---------------------------------------------%

%%%%--------rough normalization---------%
g=g-mean(g);  
g=g/max(abs(g));  
gmax=max(g);  
gmin=min(g);  
g=(g-(gmax+gmin)/2)/(gmax-gmin)*2;

r=reverse_point;  
I1=find(h1(1:r(1))~=0);  
I2=find(h1(r(n-1):r(n))~=0);  
I2=I2+r(n-1)-1;

%%%%--------detailed normalization---------%
if h1(I1)>0  
a=1;b=2;
else
    a=2;b=1;
end

for j=b:2:n-1
    q=find(zmax>r(j) & zmax<r(j+1));
    for i=1:length(q)-1
        [g(zmax(q(i))):zmax(q(i+1))),ps]=mapminmax(g(zmax(q(i))):zmax(q(i+1))));
    end
end

% for j=b:2:n-1
%     q=find(zmax>r(j) & zmax<r(j+1));
%     [y,index]=min(abs(zmin-zmax(q(length(q)))))
%     [g(zmax(q(length(q))):zmin(index+1)),ps]=mapminmax(g(zmax(q(length(q))
%    ):zmin(index+1)));
% end

for j=a:2:n-1
    q=find(zmin>r(j) & zmin<r(j+1));
    % [g(r(j):zmin(q(1))),ps]=mapminmax(g(r(j):zmin(q(1))'),g(r(j)),1)
    for i=1:length(q)-1
        [g(zmin(q(i))):zmin(q(i+1))),ps]=mapminmax(g(zmin(q(i))):zmin(q(i+1))));
    end
end

% for j=a:2:n-1
%     q=find(zmin>r(j) & zmin<r(j+1));
%     [y,index]=min(abs(zmax-zmin(q(length(q)))))
%     [g(zmin(q(length(q))):zmax(index+1)),ps]=mapminmax(g(zmin(q(length(q))
%    ):zmax(index+1)));
% end
if h1(I1)>0
    q=find(zmax<r(1));
    for i=1:length(q)-1
        [g(zmax(q(i))):zmax(q(i+1))),ps]=mapminmax(g(zmax(q(i))):zmax(q(i+1))));
end

% [g(1:zmax(1)), ps] = mapminmax(g(1:zmax(1)));
else
  q = find(zmin < r(1));
  for i = 1:length(q) - 1
    [g(zmin(q(i)):zmin(q(i+1))), ps] = mapminmax(g(zmin(q(i)):zmin(q(i+1))));
  end
  % [g(1:zmin(1)), ps] = mapminmax(g(1:zmin(1)));
end

if h1(I2) < 0
  q = find(zmax > r(n)+1);
  for i = 1:length(q) - 1
    [g(zmax(q(i)):zmax(q(i+1))), ps] = mapminmax(g(zmax(q(i)):zmax(q(i+1))));
  end
  [g(zmax(q(i+1)):length(g)), ps] = mapminmax(g(1:(zmax(q(i+1)):length(g)));
else
  q = find(zmin > r(n)+1);
  for i = 1:length(q) - 1
    [g(zmin(q(i)):zmin(q(i+1))), ps] = mapminmax(g(zmin(q(i)):zmin(q(i+1))));
  end
  [g(zmin(q(i+1)):length(g)), ps] = mapminmax(g(zmin(q(i+1)):length(g)));
end

%%%---------------------------------------------%
%%% This program is to normalize Type 2 SMS
%%% parameters: h1 - E(n)
%%% zmax - peak points
%%% zmin - valley points
%%% g - SMS
%%%---------------------------------------------%

%%%-----------------rough normalization-------------% 
g = g-mean(g);
g = g/max(abs(g));
gmax = max(g); gmin = min(g);
g = (g - (gmax + gmin)/2)/(gmax - gmin)*2;
r = reverse_point;
I1 = find(h1(1:r(1)) ~= 0);
I2 = find(h1(r(n-1):r(n)) ~= 0);
I2 = I2 + r(n-1) - 1;

%%% --------------- detailed normalization -----------------%
if h1(I1) > 0
    a = 1; b = 2;
else
    a = 2; b = 1;
end

for j = b:2:n-1
    q = find(zmax > r(j) & zmax < r(j+1));
    meanmax = max(g(zmax(q)));
    [g(zmax(q(length(q))):r(j+1)), ps] = mapminmax(g(zmax(q(length(q))):r(j+1)), g(r(j+1)), meanmax);
    for i = 1:length(q)-1
        [g(zmax(q(i)):zmax(q(i+1))), ps] = mapminmax(g(zmax(q(i)):zmax(q(i+1))), -1, meanmax);
    end
end

% for j = b:2:n-1
%    q = find(zmax > r(j) & zmax < r(j+1));
%    [y, index] = min(abs(zmin - zmax(q(length(q)))));
%    [g(zmax(q(length(q))):zmin(index+1)), ps] = mapminmax(g(zmax(q(length(q))):zmin(index+1)), -1, mean(g(zmax(q))));
% end

for j = a:2:n-1
    q = find(zmin > r(j) & zmin < r(j+1));
    [g(r(j):zmin(q(1))), ps] = mapminmax(g(r(j):zmin(q(1))), g(r(j)), 1);
    meanmin = min(g(zmin(q)));
    for i = 1:length(q)-1
        [g(zmin(q(i)):zmin(q(i+1))), ps] = mapminmax(g(zmin(q(i)):zmin(q(i+1))), meanmin, 1);
    end
end
% % for j = a:2:n-1
%    q = find(zmin > r(j) & zmin < r(j+1));
%   [y, index] = min(abs(zmax - zmin(q(length(q)))));
%
[g(zmin(q(length(q))):zmax(index+1)), ps] = mapminmax(g(zmin(q(length(q))):
   zmax(index+1)), mean(g(zmin(q))), 1);
% end

if h1(I1) > 0
    q = find(zmax < r(1));
    meanmax = max(g(zmax(q)));
    for i = 1:length(q) - 1
        [g(zmax(q(i)):zmax(q(i+1))), ps] = mapminmax(g(zmax(q(i)):
            zmax(q(i+1))), -1, meanmax);
    end
    [g(1:zmax(1)), ps] = mapminmax(g(1:zmax(1)), -1, meanmax);
else
    q = find(zmin < r(1));
    meanmin = min(g(zmin(q)));
    for i = 1:length(q) - 1
        [g(zmin(q(i)):zmin(q(i+1))), ps] = mapminmax(g(zmin(q(i)):
            zmin(q(i+1))), meanmin, 1);
    end
    [g(1:zmin(1)), ps] = mapminmax(g(1:zmin(1)), meanmin, 1);
end

% if h1(I1) < 0
%   q = find(zmin < r(1));
%   [y, index] = min(abs(zmax - zmin(q(length(q)))));
%   [g(zmin(q(length(q))):zmax(index+1)), ps] = mapminmax(g(zmin(q(length(q))):
%       zmax(index+1)));
% else
%   q = find(zmax < r(1));
%   [y, index] = min(abs(zmin - zmax(q(length(q)))));
%   [g(zmax(q(length(q))):zmin(index+1)), ps] = mapminmax(g(zmax(q(length(q))):
%       zmin(index+1)));
% end

if h1(I2) < 0
    q = find(zmax > r(n) + 1);
    for i = 1:length(q) - 1
        [g(zmax(q(i)):zmax(q(i+1))), ps] = mapminmax(g(zmax(q(i)):
            zmax(q(i+1))), -1, meanmax);
    end
    [g(1:zmax(1)), ps] = mapminmax(g(1:zmax(1)), -1, meanmax);
else
    q = find(zmin > r(n) + 1);
    [g(zmin(q(length(q))):zmin(index+1)), ps] = mapminmax(g(zmin(q(length(q))):
        zmin(index+1)));
\begin{verbatim}
[g(zmax(q(i)) : zmax(q(i+1))), ps] = mapminmax(g(zmax(q(i)) : zmax(q(i+1))), -1, mean(g(zmax(q(i))));
  end
  \% [g(zmax(q(i+1)) : length(g)), ps] = mapminmax(g(1 : (zmax(q(i+1)) : length(g))), -1, g(zmax(q(i+1))));
else
  q = find(zmin > r(n) + 1);
  [g(r(n) : zmin(q(1))), ps] = mapminmax(g(r(n) : zmin(q(1))), g(r(n)), 1);
  meanmin = min(g(zmin(q)));
  for i = 1:length(q) - 1
    [g(zmin(q(i)) : zmin(q(i+1))), ps] = mapminmax(g(zmin(q(i)) : zmin(q(i+1))), meanmin, 1);
  end
  \% [g(zmin(q(i+1)) : length(g)), ps] = mapminmax(g(zmin(q(i+1)) : length(g)), g(zmin(q(i+1))));
end
\end{verbatim}
Appendix B: Reference


36. R. Kliese, L. Yah Leng, K. Bertling, A. A. A. Bakar, T. Bosch, and A. D. Rakic,


45. Y. Yu, J. Xi, and J. F. Chicharo, "Improving the Performance in an Optical


51. "[http://en.wikipedia.org/wiki/Moving_average](http://en.wikipedia.org/wiki/Moving_average)."

