

Spring 2012

# Mathematical modeling of pipeline features for robotic inspection

Yang Gao

*Louisiana Tech University*

Follow this and additional works at: <https://digitalcommons.latech.edu/dissertations>



Part of the [Applied Mathematics Commons](#), [Civil Engineering Commons](#), [Mathematics Commons](#), and the [Mechanical Engineering Commons](#)

---

## Recommended Citation

Gao, Yang, "" (2012). *Dissertation*. 372.

<https://digitalcommons.latech.edu/dissertations/372>

This Dissertation is brought to you for free and open access by the Graduate School at Louisiana Tech Digital Commons. It has been accepted for inclusion in Doctoral Dissertations by an authorized administrator of Louisiana Tech Digital Commons. For more information, please contact [digitalcommons@latech.edu](mailto:digitalcommons@latech.edu).

**MATHEMATICAL MODELING OF PIPELINE FEATURES FOR  
ROBOTIC INSPECTION**

by

Yang Gao, B.S.

A Dissertation Presented in Partial Fulfillment  
Of the Requirements for the Degree  
Doctor of Philosophy

COLLEGE OF ENGINEERING AND SCIENCE  
LOUISIANA TECH UNIVERSITY

March 2012

UMI Number: 3504533

All rights reserved

INFORMATION TO ALL USERS

The quality of this reproduction is dependent upon the quality of the copy submitted.

In the unlikely event that the author did not send a complete manuscript and there are missing pages, these will be noted. Also, if material had to be removed, a note will indicate the deletion.



UMI 3504533

Copyright 2012 by ProQuest LLC.

All rights reserved. This edition of the work is protected against unauthorized copying under Title 17, United States Code.



ProQuest LLC  
789 East Eisenhower Parkway  
P.O. Box 1346  
Ann Arbor, MI 48106-1346

LOUISIANA TECH UNIVERSITY

THE GRADUATE SCHOOL

12/09/2011

Date

We hereby recommend that the dissertation prepared under our supervision  
by Yang Gao

entitled MATHEMATICAL MODELING OF PIPELINE FEATURES FOR  
ROBOTIC INSPECTION

be accepted in partial fulfillment of the requirements for the Degree of  
Doctor of Philosophy



Supervisor of Dissertation Research



Head of Department

Mechanical Engineering

Department

Recommendation concurred in:



Advisory Committee



Approved:

  
Director of Graduate Studies

Approved:

  
Dean of the Graduate School  
Dean of the College

## ABSTRACT

Underground pipeline systems play an indispensable role in transporting liquids in both developed and developing countries. The associated social and economic cost to repair a pipe upon abrupt failure is often unacceptable. Regular inspection is a preventative action that aims to monitor pipe conditions, catch abnormalities and reduce the chance of undesirable surprises. Robots with CCTV video cameras have been used for decades to inspect pipelines, yielding only qualitative information. It is becoming necessary and preferable for municipalities, project managers and engineers to also quantify the 3-D geometry of underground pipe networks. Existing robots equipped specialized hardware and software algorithms are capable of scanning the interior geometry of pipelines. Improvement in the 3-D models created from the collected data is a prerequisite for true, quantitative assessment of underground pipelines to take hold.

Many issues regarding pipeline scanning and geometry modeling remain unaddressed or unsolved. The ultimate goal of this research is to target several prominent topics related to the robotic inspection and parametric modeling of pipe geometry, filling gaps in the literature needed for more quantitative pipeline assessment.

First, parametric models of a circular cylindrical pipe undergoing deformation are developed. Different shape patterns that develop for typical pipe deformation pathways can be mathematically expressed using a single parameter. This technique offers convenience in generating or fitting 3-D models of pipes whose cross sections vary along

the pipe length, where cross sections can consist of combinations of continuous and discontinuous circular and/or elliptical arcs. The parametric model is applied to the ASTM F1216 pipe liner design standard to improve the estimation of pipe ovality.

Second, the impact of robot length, wheel span and wheel radius on the offset between the pipe origin and the origin of the robotic measurement hardware is quantified; this is important because the interpretation of data collected from camera, radar systems and ultrasonic sensors depends on the location of the hardware inside the pipe. Geometry distortions resulting from the passage of a robot through a pipe bend are simulated to demonstrate errors that can arise in cross sectional pipe measurements.

Third, an algorithm is proposed to compute the pitch, yaw and roll of a robot as well as the major and minor axis of a pipe based on laser ring measurements taken from a single end of the robot. An enhanced version of an existing double-ended measurement algorithm is presented to reduce error when pitch, yaw and roll are large.

Fourth, the relationship between geometry measurement and image processing is explored. A template-guided lateral detection paradigm using homogeneous geometric transformations and the Discrete Fourier Transform is proposed and evaluated according to error arising from lateral size, camera position and camera orientation. Relatively large laterals resembling an eclipse are easier to detect than small ones.

Fifth, a new parametric model of the shape assumed by a flexible pipe liner encased in an elliptical host pipe is presented. This model overcomes deficiencies in existing models by correctly accounting for continuity in the slope and curvature of the liner profile.

## APPROVAL FOR SCHOLARLY DISSEMINATION

The author grants to the Prescott Memorial Library of Louisiana Tech University the right to reproduce, by appropriate methods, upon request, any or all portions of this Dissertation. It is understood that "proper request" consists of the agreement, on the part of the requesting party, that said reproduction is for his personal use and that subsequent reproduction will not occur without written approval of the author of this Dissertation. Further, any portions of the Dissertation used in books, papers, and other works must be appropriately referenced to this Dissertation.

Finally, the author of this Dissertation reserves the right to publish freely, in the literature, at any time, any or all portions of this Dissertation.

Author   
Date 12/09/2011

## TABLE OF CONTENTS

ABSTRACT.....	iii
LIST OF TABLES .....	xi
LIST OF FIGURES .....	xii
NOMENCLATURE.....	xviii
ACKNOWLEDGEMENTS.....	xxiii
CHAPTER 1 INTRODUCTION .....	1
1.1 Background and Research Need.....	2
1.2 Objectives and Scope.....	9
CHAPTER 2 LITERATURE REVIEW.....	11
2.1 Underground Pipeline Deteriorations .....	12
2.1.1 Cracks and Detection.....	12
2.1.2 Joints and Detection.....	14
2.1.3 Corrosion and Detection .....	15
2.1.4 Deformation of Pipes.....	16
2.1.5 Global Layout Shift of Pipes.....	18
2.2 Geometry Scanners .....	19
2.2.1 Contact Scanner .....	20
2.2.2 Time of Flight Laser Scanner.....	21
2.2.3 SONAR Scanner .....	22

2.2.4 Triangulation Scanner .....	22
2.2.5 Stereo Vision Scanner .....	23
2.2.6 Comparison and Selection .....	24
2.3 Scanner Carrying Robotic Platforms .....	28
2.4 Robotic Deviation .....	31
2.5 Inertial Navigation Systems .....	34
2.6 Geometry Data Study.....	36
2.6.1 2-D Geometry Data.....	36
2.6.2 3-D Geometry Data.....	38
2.6.3 Sampling, Noise and Storage.....	39
2.7 Camera Lens Models .....	42
2.8 Trenchless Polymeric Liners.....	44
2.9 Mathematical Principles and Techniques.....	48
2.9.1 Circle and Ellipse Fitting .....	48
2.9.2 DFT of Contours .....	54
2.9.3 Morphological Operations .....	58
2.10 Chapter Summary .....	59
CHAPTER 3 MATHEMATICAL MODELING OF PIPE GEOMETRIES .....	61
3.1 Elliptical Cross Section (Mode 1).....	61
3.2 Circular Hinged Cross Section (Mode 2).....	64
3.3 Composite Cross Sections.....	67
3.3.1 Composite Mode 3.....	68
3.3.2 Composite Mode 4.....	70

3.3.3 Composite Mode 5.....	73
3.3.4 Summary of Composite Modes .....	75
3.4 Geometry Modeling Extensions .....	77
3.4.1 Area Loss .....	77
3.4.2 3-D Pipe Segment .....	79
3.4.3 Accurate Ovality Calculator.....	80
3.5 Chapter Summary .....	88
CHAPTER 4 DISTORTED GEOMETRY SIMULATION .....	89
4.1 Measurement Origin Offset Relative to Pipe.....	90
4.1.1 Deformation Induced Offset .....	90
4.1.2 Pitch Induced Offset .....	95
4.2 Oblique Cross Sections of Bend .....	99
4.2.1 Geometry of Bend.....	99
4.2.2 Derivation .....	101
4.2.3 Plotting.....	104
4.2.4 Verification.....	108
4.3 Chapter Summary .....	110
CHAPTER 5 ROBOTIC ORIENTATION AND PIPE GEOMETRY CALCULATION	111
5.1 Oblique Shapes by Robotic Pitch or Yaw or Roll.....	112
5.2 Oblique Shapes by Robotic Pitch and Yaw.....	113
5.2.1 Derivation of Semi Axes.....	113
5.2.2 Derivation of Angle.....	120
5.2.3 Verification.....	123

5.3 Pipe Geometry Calculation with Known Robotic Orientation .....	126
5.4 Pipe Geometry Calculation with Unknown Robotic Orientation .....	130
5.5 Enhanced Algorithm for Double Ended Scanner .....	134
5.6 Chapter Summary .....	138
CHAPTER 6 IMAGE AND GEOMETRY FUSION FOR FEATURE DETECTION...	140
6.1 Image Based Joint and Lateral Detection .....	141
6.2 Lateral and Joint Geometry Modeling .....	144
6.3 Camera's Pose Modeling .....	145
6.4 Feature in Image by Different Camera Lenses .....	147
6.4.1 Laterals in Image.....	148
6.4.2 Joints in Image .....	152
6.4.3 Comparison with Real Image.....	153
6.5 Relationship between Image and Geometry .....	154
6.6 Template Based Lateral Image Detection .....	157
6.7 Chapter Summary .....	165
CHAPTER 7 MATHEMATICAL MODELING OF LINER GEOMETRIES .....	167
7.1 Quasi Elliptical Cross Section .....	168
7.2 Lobed Cross Section .....	170
7.2.1 Chicurel's Equation.....	170
7.2.2 Glock's Equation.....	175
7.3 Composite Cross Section .....	177
7.4 Chapter Summary .....	184
CHAPTER 8 CONCLUSIONS AND RECOMMENDATIONS .....	185

8.1 Research Conclusions .....	185
8.2 Recommendations for Future Work.....	187
APPENDIX A MATLAB® CODE FOR CHAPTER 3 .....	189
APPENDIX B MATLAB® CODE FOR CHAPTER 4.....	199
APPENDIX C MATLAB® CODE FOR CHAPTER 5.....	204
APPENDIX D MATLAB® CODE FOR CHAPTER 6 .....	209
APPENDIX E MATLAB® CODE FOR CHAPTER 7.....	215
BIBLIOGRAPHY.....	218

## LIST OF TABLES

Table 2.1 Scanning Technologies Qualitative Comparison .....	25
Table 2.2 Inspection Robotic Platforms for Water and Sewer Pipes .....	31
Table 3.1 Influences of Integral Methods on Mode 1 .....	83
Table 3.2 Influences of Integral Methods on Mode 2 .....	87
Table 3.3 Influences of Integral Methods on Mode 3 .....	87
Table 3.4 Influences of Integral Methods on Mode 4 .....	87
Table 3.5 Influences of Integral Methods on Mode 5 .....	88

## LIST OF FIGURES

Figure 1.1 Necessity of Pipe Interior Geometry at Different Stages .....	5
Figure 1.2 Pipe–Robot–Earth Relationships.....	6
Figure 1.3 Distorted Global Layouts of Pipeline at Different Times.....	7
Figure 1.4 Lab Pipe Scanner with Straight Traveling Path.....	8
Figure 2.1 Longitudinal and Circumferential Cracks .....	13
Figure 2.2 Crack Detection Algorithm Input and Output .....	14
Figure 2.3 Pipe Joint Dislocation.....	14
Figure 2.4 Circular Joint Image Detection.....	15
Figure 2.5 MFL Detectable Pitting Corrosion in Metal Pipe.....	16
Figure 2.6 UT Detectable Thickness Defect in Metal Pipe .....	16
Figure 2.7 Several Conceptual Cross sectional Deformations.....	17
Figure 2.8 Cross Sectional Deformation Progressions with Four Plastic Hinges.....	18
Figure 2.9 Flexible and Rigid Global Shift.....	19
Figure 2.10 Contact Scanner for Digitization of Portrait.....	20
Figure 2.11 Principle of Time-of-Flight Laser Scanner.....	21
Figure 2.12 Principle of SONAR Scanner .....	22
Figure 2.13 Principle of Optical Triangulation Laser Scanner .....	23
Figure 2.14 Principle of Stereo Vision Scanner .....	24
Figure 2.15 Laser Ring Scanning for Pipe with Water .....	26

Figure 2.16 SONAR Scanning for Pipe with Water .....	27
Figure 2.17 ROVVER 600 Inspection Robotic Platform for Water and Sewer Pipes.....	29
Figure 2.18 Rosen Geometry PIG for Oil and Gas Transmission Pipes.....	29
Figure 2.19 Cable Powered WellTec Hybrid Robotic for Oil and Gas Wells.....	29
Figure 2.20 Common Robotic Platform with Laser Rings .....	30
Figure 2.21 Pitch, Yaw and Roll .....	31
Figure 2.22 Robotic Yaw with Oblique Cross Sectional Scan.....	32
Figure 2.23 Robotic Pitch with Oblique Cross Sectional Scan .....	33
Figure 2.24 Wireframe Distortion Caused by Robotic Roll .....	33
Figure 2.25 Linear Acceleration and Angular Velocity .....	34
Figure 2.26 Two Dimensional Scanners with Measurements.....	37
Figure 2.27 Three Dimensional Scanner with Non-Uniform Measurement Density .....	39
Figure 2.28 Rectangular and Radial Grid Sampling.....	40
Figure 2.29 Cylindrical and Polar Coordinates System for Geometry Data Storage .....	41
Figure 2.30 Interior Surface of Pipe before and after CIPP Lining .....	45
Figure 2.31 FFP Liners before and after Deployment .....	45
Figure 2.32 Lining Defects after Installation.....	46
Figure 2.33 Elliptical and Lobed Cross Sections of Liners .....	47
Figure 2.34 Circle Fitting Outputs.....	52
Figure 2.35 Complex Coordinates of Contour.....	56
Figure 2.36 Common Spatial Variations of Contour .....	56
Figure 2.37 Morphological Dilation and Erosion.....	59
Figure 3.1 Cross Sectional View of Elliptical Deformation (Mode 1) .....	62

Figure 3.2 Cross Sectional View of Circular Hinged Deformation (Mode 2) .....	65
Figure 3.3 Composite Cross Section (Mode 3).....	68
Figure 3.4 Determination of Lower Limit for Elliptical Deformation Indicator .....	69
Figure 3.5 Relationship between Top and Bottom Deformation (Mode 3) .....	69
Figure 3.6 Composite Cross Section (Mode 4).....	70
Figure 3.7 Generalized Relationship between Top and Bottom Deformation (Mode 4)..	72
Figure 3.8 Composite Cross Section (Mode 5).....	73
Figure 3.9 General Relationship of Composite Cross Section (Mode 5, Case 1).....	74
Figure 3.10 Generalized Relationship of Composite Cross Section (Mode 5, Case 2)....	75
Figure 3.11 Deformation Pathway of Composite Cross Section (Case 1).....	76
Figure 3.12 Deformation Pathway of Composite Cross Section (Case 2).....	76
Figure 3.13 Reduced Area of Hinged Portion.....	78
Figure 3.14 Area Losses for Different Cross Section Modes .....	78
Figure 3.15 Three Dimensional RGB Plots of Cross Sectional Modes.....	79
Figure 3.16 Mean Diameter, Percent Ovality and Ovality Reduction Factor .....	81
Figure 3.17 Accurate Calculation of Critical Pressure using Elliptical Integral.....	81
Figure 3.18 Accurate Calculation of Liner Thickness using Elliptical Integral.....	82
Figure 4.1 Non Uniform Point Coverage Caused by Measurement Origin Offset.....	90
Figure 4.2 Different Configurations of Wheel Span.....	91
Figure 4.3 Cross Sectional Geometry of Relating Wheel Span and Pipe .....	92
Figure 4.4 Offset Induced by Pipe Deformation and Wheel Span.....	94
Figure 4.5 Geometry of Robot with Different Length and Wheel Radius.....	96
Figure 4.6 Offset Influenced by Robotic Length and Wheel Radius.....	98

Figure 4.7 Pipe Bend with Manufacturing Parameters .....	99
Figure 4.8 Three Dimensional Torus .....	100
Figure 4.9 Robot Traveling in Bend .....	101
Figure 4.10 Two Sequential Coordinates Transformations.....	103
Figure 4.11 Oblique Shapes in Sharp Bend Caused by Pitch or Yaw Only .....	106
Figure 4.12 Oblique Shapes in Smooth Bend Caused by Pitch or Yaw Only.....	106
Figure 4.13 Oblique Shapes in Sharp Bend Caused by Pitch and Yaw Combined.....	107
Figure 4.14 Oblique Shapes in Smooth Bend Caused by Pitch and Yaw Combined .....	108
Figure 4.15 Verification of Oblique Shapes in Bend by Laser Dots Simulation .....	110
Figure 5.1 Coordinates System Setup for Robotic Pitch and Yaw.....	114
Figure 5.2 Global Extremes Location of W Vector with Varying Pitch and Yaw.....	119
Figure 5.3 Implicit Roll by Direct and Stepwise Rotation.....	122
Figure 5.4 Verification of Theoretical Solutions for Oblique Ellipse .....	125
Figure 5.5 Determination of Solution Sign for Extreme w Vectors.....	129
Figure 5.6 Algorithm Performance (Four Unknowns, Wild Initial Guess).....	132
Figure 5.7 Algorithm Performance (Four Unknowns, Close Initial Guess) .....	132
Figure 5.8 Algorithm Performance (Close Initial Guess) .....	134
Figure 5.9 Double Ended Scanner .....	135
Figure 5.10 Coordinates System for Double Ended Scanner .....	136
Figure 5.11 Orientation Calculation and Enhancement .....	138
Figure 6.1 Joint and Lateral Detection with Different Edge Detectors .....	142
Figure 6.2 Joint and Lateral Detection with Morphological Operations .....	143
Figure 6.3 Coordinates System for Joint and Lateral .....	144

Figure 6.4 Camera's Pose in Earth's Coordinates System.....	145
Figure 6.5 Lateral Contours in Earth's and Camera's Coordinates System.....	147
Figure 6.6 Joint Contours in Earth's and Camera's Coordinates System .....	147
Figure 6.7 Small Laterals in Image Plane Acquired by Different Lenses.....	148
Figure 6.8 Large Laterals in Image Plane Acquired by Different Lens .....	149
Figure 6.9 Projection of Lateral onto Image Plane .....	150
Figure 6.10 Lateral Size in Image with Viewing Angle.....	151
Figure 6.11 Joints in Image Plane Acquired by Different Lens .....	152
Figure 6.12 Lateral Simulations by Intuition.....	153
Figure 6.13 Complementary Relationship between Geometry and Image .....	155
Figure 6.14 Distance Bias Correction for Double Camera System .....	156
Figure 6.15 Lateral Templates with Different Levels of Noise .....	158
Figure 6.16 DFT with Inaccurate Lateral Size Estimation .....	160
Figure 6.17 DFT with Inaccurate Camera Position Estimation (Large Lateral).....	161
Figure 6.18 DFT with Inaccurate Camera Position Estimation (Small Lateral).....	162
Figure 6.19 DFT with Inaccurate Camera Orientation Estimation (Large Lateral).....	164
Figure 6.20 DFT with Inaccurate Camera Orientation Estimation (Small Lateral).....	165
Figure 7.1 Possible Deformation Pathways for Restrained Circular Liners.....	168
Figure 7.2 Small Deflection Theory for Liner .....	168
Figure 7.3 Quasi Elliptical and Perfect Elliptical Shape for Liner .....	170
Figure 7.4 Chicurel's Coordinates System for Wavy Buckling Portion .....	171
Figure 7.5 Wavy Lobes Approximated by Chicurel's Equation .....	173
Figure 7.6 Comparison of Second Order Derivatives.....	175

Figure 7.7 Glock's Coordinates System for Wavy Buckling Portion .....	176
Figure 7.8 Post Buckling of Encased Liner .....	177
Figure 7.9 Polar Coordinates for Elliptical Host Pipe .....	178
Figure 7.10 Different Wavy Shapes with Varying Buckled Angles.....	181
Figure 7.11 Different Wavy Shapes with Varying Maximum Deflection.....	181
Figure 7.12 Different Wavy Shapes with Varying Powers.....	182
Figure 7.13 Top View of RGB Plots of Cross Sectional Modes.....	183

## NOMENCLATURE

$a, b, c$	vector components of robotic orientation
$a_c, b_c, c_c, d_c$	coefficients for circle equation
$a_e, b_e, c_e, d_e, e_e, f_e$	coefficients for ellipse equation
$c$	complex contour coordinate
$C$	ovality reduction factor
$C_f$	centroid of cross section in front laser plane
$C_r$	centroid of cross section in rear laser plane
$d_0$	small deflection
$d_r$	radial deflection
$d_t$	tangential deflection
$d_w$	radial deflection of lobe
$D$	linear distance
$D_m$	mean liner diameter
$D_o$	outside liner diameter
$E$	modulus of elasticity
$E_l$	long-term modulus of elasticity
$f, g$	complex Discrete Fourier Transform

$h$	lobe height
$H$	robotic height; bend height
$I$	moment of inertia
$k$	scaling factor of lens; ratio of lobe length to pipe diameter
$k_1, k_2, k_3$	coefficients for lobe equation
$K$	enhancement factor
$\mathbf{i}$	axis of rotation
$l$	dimensionless robotic length; lobe length
$L$	robotic length; bend length
$\mathbf{n}$	opposite camera viewing direction
$N$	safety factor; number of points
$o_f$	dimensionless origin offset
$O_f$	origin offset
$O_{fh}$	horizontal origin offset in front laser plane
$O_{fv}$	vertical origin offset in front laser plane
$O_h$	horizontal origin offset
$O_{rh}$	horizontal origin offset in rear laser plane
$O_{rv}$	vertical origin offset in rear laser plane
$O_v$	vertical origin offset
$p$	perimeter; power term of lobe equation
$P$	external pressure; origin in front laser plane

$P_{cr}$	critical pressure
$q$	percent ovality
$Q$	origin in rear laser plane
$r$	radius of pipe
$r_0$	polar distance of arc center
$r_1$	semi-major axis length of pipe
$r_2$	semi-minor axis length of pipe
$r_i$	radial distance in the images
$r_c$	polar distance of circle
$r_e, r_e', r_e''$	polar distance of ellipse
$r_l$	lateral radius
$r_{major}, r_{major\_est}$	semi-major axis length of oblique cross section
$r_{minor}, r_{minor\_est}$	semi-minor axis length of oblique cross section
$r_w$	polar distance of wavy lobe; dimensionless wheel radius
$\mathbf{R}$	rotation matrix
$R$	bend centerline radius
$R_w$	robotic wheel radius
$s$	deformation indicator; scaling factor of contour
$s_1, s_2$	deformation indicators
$s_{01}, s_{02}$	initial deformation indicators

$t$	complex translation of contour
$t$	liner thickness; radial distance from origin to pipe wall
$\mathbf{u}$	vector from left to right of camera
$u, v, w$	vector components in oblique cross section
$u_{major}, v_{major}, w_{major}$	major vector components in oblique cross section
$u_{minor}, v_{minor}, w_{minor}$	minor vector components in oblique cross section
$\mathbf{v}$	vector from bottom to top of camera
$\nu$	Poisson's ratio
$w$	dimensionless half wheel span; bend width
$W$	half wheel span; pipe wall thickness
$x$	longitudinal direction
$y$	vertical direction
$\bar{y}$	vertical centroid coordinate
$y_c$	vertical arc center coordinate
$z$	horizontal direction
$\bar{z}$	horizontal centroid coordinate
$z_c$	horizontal arc center coordinate
$\alpha$	half angle at hinge
$\beta, \beta_{est}$	counterclockwise angle of major axis of oblique cross section
$\delta$	longitudinal distance difference
$\Delta\theta_h, \Delta\theta'_h$	incremental yaw angles
$\Delta\theta_v, \Delta\theta'_v$	incremental pitch angles

$\Delta\theta_i, \Delta\theta_i'$	incremental roll angles
$\varepsilon$	infinitely small value
$\varphi$	half buckling angle
$\lambda_1, \lambda_2$	eigenvalues
$\mu$	bias correction coefficient
$\theta$	polar angle
$\theta_0$	polar angle of arc center
$\theta_a$	azimuth angle
$\theta_h$	yaw angle
$\theta_f$	half viewing angle of front camera
$\theta_r$	roll angle; half viewing angle of rear camera
$\theta_v$	pitch angle
$\theta_\backslash$	angle between robotic orientation and pipe centerline
$\theta_z$	zenith angle
$\rho$	polar distance in spherical coordinates system

## ACKNOWLEDGEMENTS

I feel deeply grateful to my advisor, Dr. David Hall, for his great guidance and support throughout my graduate studies; and for his generous efforts, patience, and motivations in the course of my research and dissertation writing. I want to express my appreciation to Dr. Erez Allouche for his insightful suggestions on the material organization, and for his experienced industry knowledge that broadens my vision in the relevant field. I want to sincerely thank Dr. Heath Tims for his helpful advice in the early phase of my research. My sincere appreciation extends to Dr. Jun-Ing Ker and Dr. Xingran Wang for their precious time and service on my committee.

I am truly indebted to my parents – the best friends I feel blessed to have. I cannot thank them enough for their inspiring encouragement when I was down in a valley, for their kind reminders when I am up at a peak, and for their unconditional love in this mountain journey that we travel together being half an earth away.

Thank you, God, for leading me through the academic problems with your wisdom, for stabilizing me in the knowledge voyage with abundant courage, and for marching me forth with faith and your amazing grace.

# **CHAPTER 1**

## **INTRODUCTION**

Many aspects of modern society are dependent on the complex network of water supply and sewer disposal systems. Underground water and sewer pipelines serve as an essential infrastructure element for liquid transport in civil and industrial applications. The age-related degradation of many underground pipelines in the United States is becoming increasingly prominent. As reliability requirements and concern for public safety and environmental impact increase, sudden failure of medium to large underground pipes with high flows or of pipes buried within a dense pipeline network becomes almost unacceptable.

Regular inspection of pipelines is a preventative measure that aims to shrink the time window for defect growth and locate defects by collecting and evaluating interior image and geometry information. Although inspection may disrupt the normal usage of a pipeline, such temporary disruptions still triumphs the time required to replace failed pipes. It is usually desirable to frequently repair small local defects rather than severe failures; only by regular inspection can defects be caught in an early stage. Many trenchless lining products are designed for a specified lifetime, such as 50 years; geometry inspection of the host pipes provides important design data for selecting the appropriate liner thickness, and

regular inspection of the liner system builds a database that is useful for remaining life predictions.

As an attempt to assess pipe conditions and automate this process, video-based inspection techniques involving CCTV (closed-circuit television) and SSET (sewer scanner evaluation technology) have been widely used. However, these traditional inspection methods mostly give qualitative information on local deterioration, mostly cracks, and leave the interior geometry untreated. Scanning robots built on various physical principles and equipped with advanced sensors have emerged over the past decade to allow for modeling the interior geometry of pipes. Geometry and non-geometry information that is available for interpretation is critical for appropriate decisions and actions on rehabilitation. Regulations on regular qualitative and quantitative scanning are being drafted, implemented, and enforced.

### **1.1 Background and Research Need**

Acquiring the correct interior geometry is very important for further studies of the pipe. First, during transportation and installation, a circular cylindrical pipe segment may deform into an elliptical cylindrical segment. Timely measurement of the interior geometry at this early stage not only discovers problems such as transportation damage, soil backfill quality or insufficient pipe design, but serves as an anchor dataset for future comparisons.

Second, finite element software facilitates solving complex structural problems involving the stresses, strains and deformations of buried pipelines. The typical finite element procedure involves pre-processing, analyzing and post-processing. In the pre-processing step, primitive shapes such as spheres, cones, cubes, toroids, pyramids, and cylinders are chosen to describe the geometry of interest, and then the mesh grid is placed

onto the geometry to form discrete nodes. This methodology is reasonable if a pipe segment can indeed be approximated by primitive shapes. Otherwise, a point cloud taken from the measurements should be used to define the grid and nodes. Therefore, no matter whether it is to verify primitive shape assumptions or to capture shape details that cannot be described accurately by primitive shapes, correct interior measurement is critical, especially if structural integrity is of concern.

Third, when the deterioration of a pipeline has not yet reached the point of impending failure, trenchless lining may be an alternative rehabilitation method in place of the traditional dig-and-replace method. Currently, the most frequently used standard for CIPP (cured in place plastic) liner design is ASTM F1216 [1]. A design equation is established for partially deteriorated pipes as Eq. (1.1):

$$P_{cr} = \frac{2KE_L}{1-\nu^2} \cdot \frac{1}{(\text{SDR}-1)^3} \cdot \frac{C}{N}, \quad (1.1)$$

where  $C$  is the ovality reduction factor as in Eq. (1.2):

$$C = \left[ \frac{\left(1 - \frac{q}{100}\right)^3}{\left(1 + \frac{q}{100}\right)^2} \right], \quad (1.2)$$

$P_{cr}$  = groundwater load, psi (MPa),

$K$  = enhancement factor of the soil and existing pipe adjacent to the new pipe (a minimum value of 7 is recommended where there is full support of the existing pipe),

$\nu$  = Poisson's ratio (0.3 on average),

SDR = standard dimension ratio of CIPP =  $D_o/t$  = “outside liner diameter” / “average liner thickness”,

$q$  is the percent ovality (less than 10%) as in Eq. (1.3),

$$q = \frac{\text{MaxInsideDiameter} - \text{MeanInsideDiameter}}{\text{MeanInsideDiameter}} \times 100, \quad (1.3)$$

$N$  = factor of safety,

$E_L$  = long-term modulus of elasticity for CIPP, psi (MPa).

The ovality  $q$  of the host pipe is one quantity that the average liner thickness  $t$  is dependent on. Too small a value for  $t$  means the liner is not structurally adequate to support the critical groundwater load  $P_{cr}$ , whereas too large a value for  $t$  means excessive waste of raw materials and a reduction in flow capacity. Therefore, an accurate calculation of ovality  $q$  is important for both structural and economic reasons. The ASTM F1216 design equation implies a uniform pipe segment; to obtain the ovality, only the cross sectional points should be scanned or retrieved.

Last, there are few candid collections of data regarding the long term structural and hydraulic performance of pipe liners. Knowledge of the behavior of various lining products often stays at the laboratory level. Liners are relatively flexible and are encased in host pipes that are assumed to be perfectly rigid. These liners can exhibit a variety of deformation modes as they slowly deform over time in response to the applied groundwater pressure. Therefore, it is important that the evolution of liner shape be available as a liner system ages; this is accomplished by repeated scans of the liner system, as depicted in Figure 1.1. Accelerating changes in the measured profile of liners can alert system managers to initiate repairs.

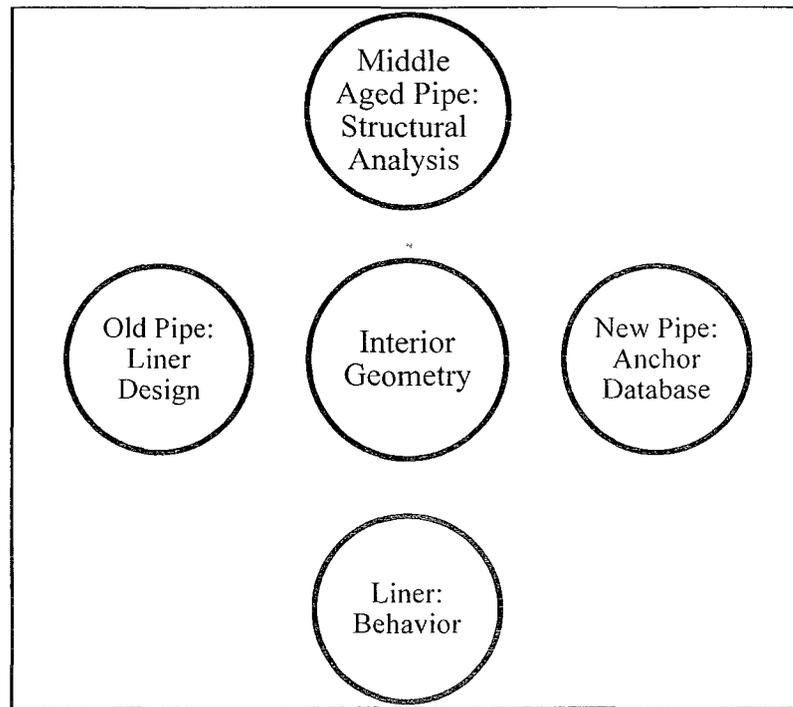


Figure 1.1 Necessity of Pipe Interior Geometry at Different Stages

As part of a pipeline integrity management program, knowledge of the global layout of pipelines is important for topographical mapping and for new pipeline layout designs. Underground pipelines can have global layout shifts relative to the earth as a result of earthquakes, traffic live loads, underground water level changes, and soil loss. Vertical sinking is a more common form of “damage” than horizontal movement.

In the ideal case, if a robot travels along a pipe’s centerline, then the robot–earth relationship is solely provided by the sensors. Unfortunately, at locations where the robot does not follow the centerline, it becomes necessary to use the robot–pipe relationship as an intermediate step to deduce the actual global layout, as shown in Figure 1.2 .

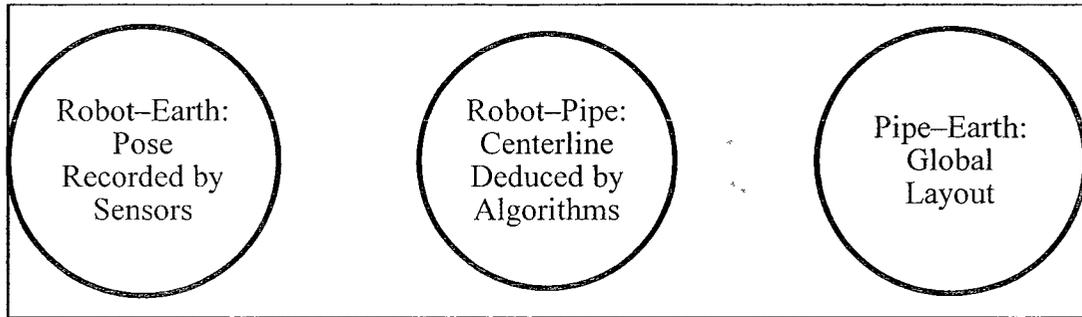


Figure 1.2 Pipe–Robot–Earth Relationships

On-board sensors usually diverge for a long pipeline, and measurement error is inevitable when different scanning technologies or configurations are implemented. Moreover, pipeline data referenced by inaccurate distance markers, undocumented pipeline rerouting, and location data with unknown accuracy and origin make different geometry datasets of the same pipeline incomparable without an effective algorithm. Figure 1.3 shows such a problem from a top view in the earth’s coordinate system. Solid and dashed lines represent two measured global layouts of a pipeline at different times. Stable features are depicted as circles with arrows connecting the corresponding pairs. Pipelines can undergo 3-D translation, rotation, and scaling. The actual global change is mixed with erroneous sensor readings and should be filtered out by aligning the corresponding features. Note that Figure 1.3 illustrates the horizontal movement; a side view of the same phenomenon can be constructed to show the pipeline sinking vertically.

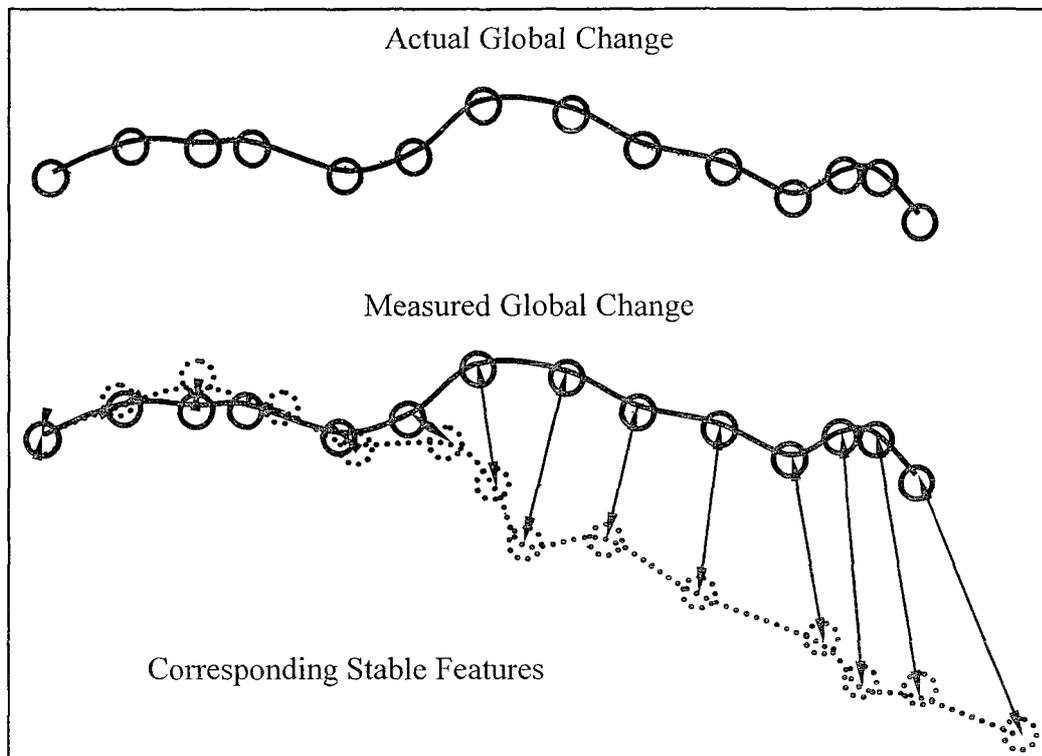


Figure 1.3 Distorted Global Layouts of Pipeline at Different Times

In a lab environment, pipes are usually straight and short in length, and measurements are acquired in a well-controlled setting. A scanner measures the distance from the pipe wall to a predefined reference point. The robot carrying the scanning hardware is forced to maintain a straight path parallel to the pipe's centerline. Very often the main purpose of such a strict setting is to prohibit the undesirable lateral movement of the robot and verify the correctness of the physical principle used by the scanner. A lab pipe scanner supported by a fixed track used by Salagundi [25] is shown in Figure 1.4. Therefore, for a 2-D scanner, sequential cross sectional scans can be placed along a straight line with intervals determined by the onboard odometer; for a 3-D scanner, sequential segmental scans can be merged the same way with redundancy removal.

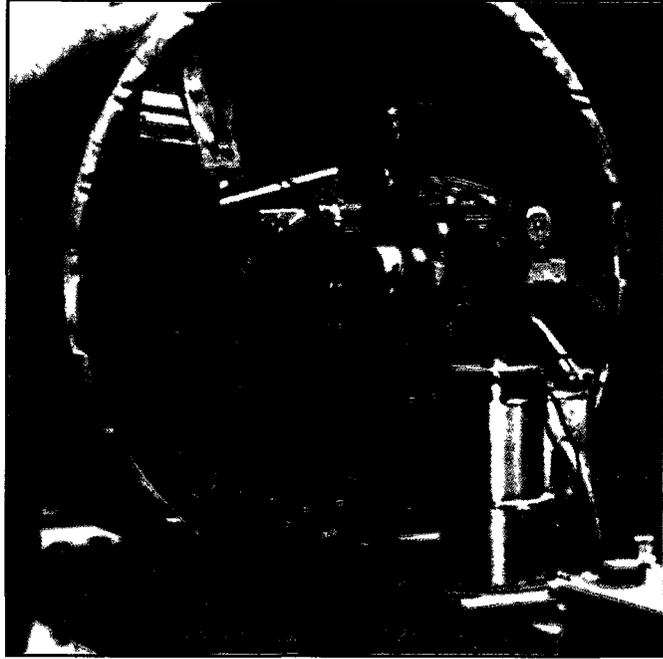


Figure 1.4 Lab Pipe Scanner with Straight Traveling Path

However, such an ideal setting can hardly be achieved in the field, especially for a long pipeline with bends and sediment that can make the robot deviate off the pipe's centerline. On-board sensors only give the robot–earth relationship; therefore, algorithms should be used to explore the robot–pipe relationship. Both interior geometry and global layout depend on how accurate the robot–pipe relationship is quantified.

A 2-D scanner measures points in a plane, perpendicular or oblique to the pipe's centerline. A 3-D scanner reserves the depth information, but the perpendicular cross section is not directly available. Algorithms are needed to fully utilize the depth information, to retrieve the robot–pipe relationship, and to determine the perpendicular cross sections.

In conclusion, underground pipelines subject to complex loadings can undergo significant changes in both interior geometry and global layout. A regular inspection plan

is desirable to monitor pipeline behavior through the service life. Correct interior geometry, global layout, and change quantification is needed for design, analysis and mapping purposes. However, it is inevitable that some erroneous sensor data, robotic deviation, calibration and sampling differences, noise, and vision occlusion will complicate the robot–earth and robot–pipe relationships. The huge amount of data generated by scanning robots can be misleading if left uncorrected, unregistered, and unprocessed. There has been little work exploring these issues.

## **1.2 Objectives and Scope**

One objective of this research is to develop a set of mathematical tools to solve engineering problems associated with mathematical modeling and robotic inspection of pipes and other buried conduits. This “toolbox” is used to parametrically establish the geometrical variants of an original cylindrical pipeline and to model and correct erroneous measurement and/or visual impressions resulting from deviation of the robotic platform as it travels through the conduit (i.e., correct the robot–pipe relationship) or changes in the pipe alignment (e.g., horizontal and vertical bends) that result in optical distortion of 2-D images.

This research explores the following topics to establish groundwork for several areas associated with pipeline geometry measurement by robotic inspection:

1. Model common cross sectional geometries parametrically. Several cross sectional deformation modes comprised of hinges and circular/elliptical arcs are mathematically described. The out of roundness (ovality), widely used in industry, is calculated for each of these modes.

2. Examine geometry distortions introduced by several sources, including pipe deformation, robotic deviation and pipe bends. Quantify and model the effects associated with each source.
3. Propose a new robotic orientation correction algorithm for a single-ended scanner in a uniform elliptical cylindrical pipe segment. Enhance the existing orientation estimation algorithm for a double-ended scanner.
4. Propose a lateral detection algorithm based on template modeling and the Discrete Fourier Transform. Explore the relationship between geometry measurements and image processing, effectively fusing the data from these two sensors for improved feature detection.
5. Review the restrictions of existing modeling equations for liners that deform into a lobed pattern when subjected to groundwater loading. Propose a new equation that satisfies the boundary conditions while enforcing continuity in slope and curvature.

## **CHAPTER 2**

### **LITERATURE REVIEW**

The following areas are reviewed as deemed relevant to the objectives of this research. With interior geometry and global layout being the focus, images are also discussed as valuable peripheral information.

1. Common deterioration modes that occur in underground pipes are reviewed and categorized according to their value to this research. Three types of trenchless liners are briefly discussed because of the unique structural behavior of polymeric materials.
2. Various scanning principles are introduced with respective advantages and disadvantages. Applications of these principles to underground water and sewer pipelines are discussed.
3. Several examples of commercially available robotic platforms that carry the scanners are examined. Robotic platform types and capabilities for different market sectors including water and sewer pipelines are summarized.
4. Geometry distortions caused by robotic deviation relative to the pipe are depicted.
5. Inertial navigation system sensors that record the robotic pose relative to the earth are introduced. Reasons for the inaccuracy and divergence of sensors are explained.

6. Detailed procedures of 2-D and 3-D laser scanners are described. Issues such as sampling, noise and data storage are discussed.
7. Five camera lens models that relate spatial and image locations are reviewed.
8. Relevant mathematics principles and techniques including shape fitting, Discrete Fourier Transforms, and morphological operations are introduced.

## **2.1 Underground Pipeline Deteriorations**

Underground pipes, especially middle-aged and old pipes, may have loading histories that evolve over time. Cracks, joint dislocations, corrosion, interior geometric deformation, global layout shifts, etc., with different causes and consequences, are common forms of deterioration found in aged pipes. Detection of cracks, joints dislocation, and corrosion provide valuable information for preventive maintenances and pipe integrity assessment. It is of primary interest to examine the interior geometric deformation and the global shift in the layout of the pipe system.

### 2.1.1 Cracks and Detection

Cracks can propagate along the longitudinal and circumferential directions. Longitudinal and circumferential cracks affect pipe integrity in slightly different ways. Longitudinal cracks weaken the structural strength along the length and give rise to hinged cross sectional deformation. Circumferential cracks do not deform the interior geometry as much along the length. When the severity of a circumferential crack reaches a certain level, it can be treated as a joint dislocation. Examples of both crack geometries are shown in Figure 2.1 (left: longitudinal; right: circumferential).

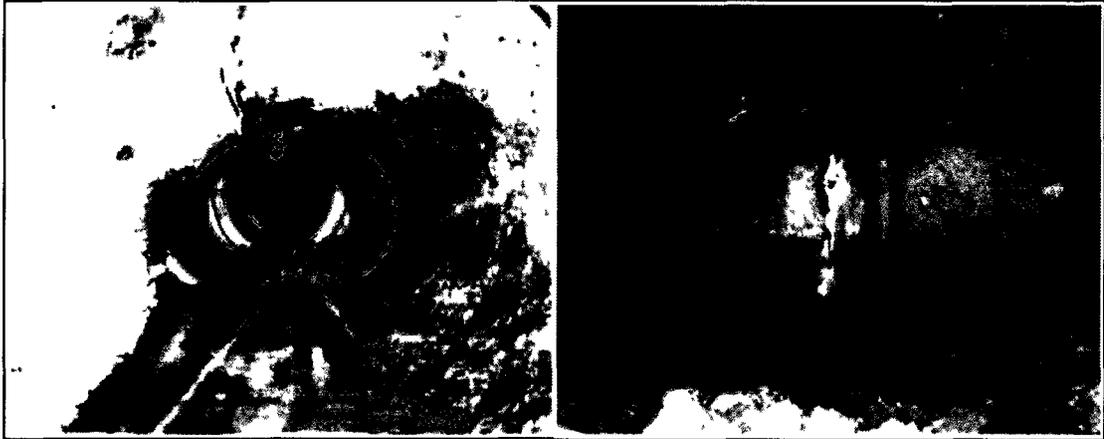


Figure 2.1 Longitudinal and Circumferential Cracks

Cracks do not result in significant geometric change at the early stage of appearance. However, because the adverse effects on the structural integrity of pipes can emerge and amplify if left untreated, cracks should be detected as a preventive action for a complete pipeline integrity management program.

Cracks show different intensity characteristics from the neighborhood. Iyer and Sinha [16] noticed that many cracks exhibit a tree-like shape and developed an approach for automatic detection and segmentation of cracks. Cracks were defined as darkest in the image, locally linear and branching in a piecewise fashion. Contrast enhancement, morphological treatment, curvature evaluation in the cross direction and alternating filters were sequentially applied to the image to produce the final segmented binary crack map. The input and output with intermediate steps from their work are shown in Figure 2.2 (top left: raw image; top right: after thresholding; bottom left: after edge detector; bottom right: final result). Their approach required very good illumination and high contrast in the image, which is sometimes difficult to achieve in the CCTV video footage or images according to the current inspection standard.

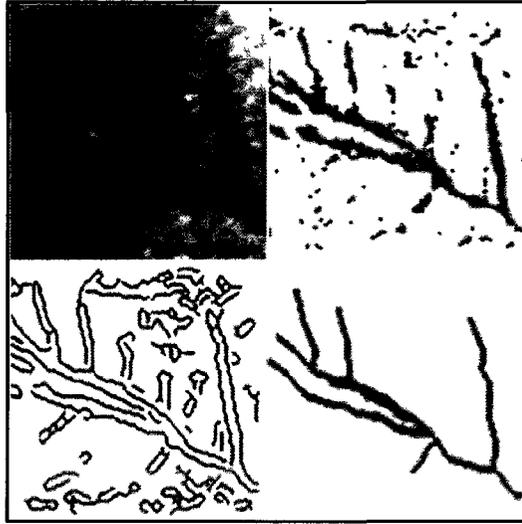


Figure 2.2 Crack Detection Algorithm Input and Output

### 2.1.2 Joints and Detection

Joints dislocation can be caused by traffic loadings, hydro-geological changes, earthquakes and leaks. The consequences can be loss of functionality due to gradient reverse, breaking of adjacent pipes, maintenance budget increase, etc. Figure 2.3 shows a typical joint dislocation.

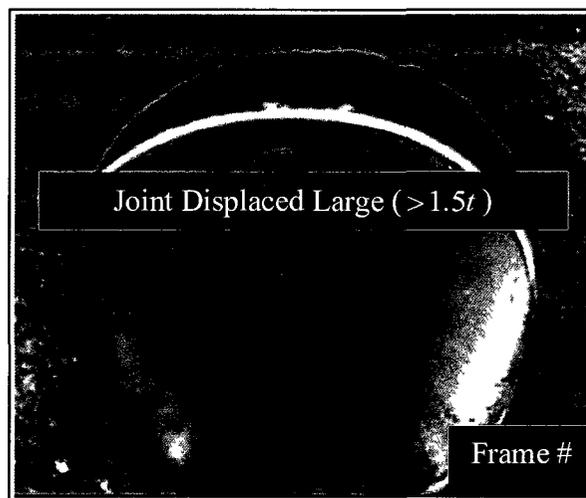


Figure 2.3 Pipe Joint Dislocation

Pan, Clarke and Ellis [21] worked on the detection of pipe joints in low illumination and noisy images. Assuming each image has at least one joint, a circle or connected arcs were fitted in the image to reveal the location of the joint. A prior knowledge of the configuration of the camera and the pipe was required to model the characteristics of the images. Figure 2.4 shows an end result of their algorithm, which unfortunately does not generalize to non-circular joints. Joint features may not exist in all images due to joint distance and vision occlusion, so exhaustive analyses of every image is either unnecessary or leads to algorithm failure. Aids from other information are needed to reduce the search window.

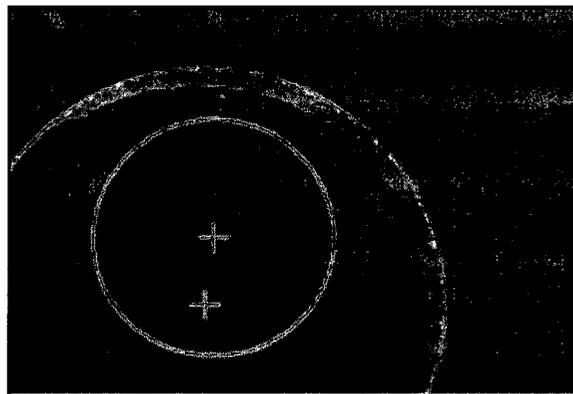


Figure 2.4 Circular Joint Image Detection

### 2.1.3 Corrosion and Detection

Rare in polymeric pipes but common in metallic pipes, corrosion, which usually results from electro-chemical interference reaction, can reduce the pipe wall thickness and eventually lead to cracks, collapse, and leaks. There are two popular technologies in the industry to detect corrosion according to Beuker [4], magnetic flux leakage (MFL) and ultrasonic testing (UT).

MFL involves saturating the pipeline wall with a magnetic field, which is interrupted at locations of corrosion and cracking, creating a magnetic flux leakage. Figure 2.5 shows pitting corrosions in a steel pipe that can be detected by MFL.

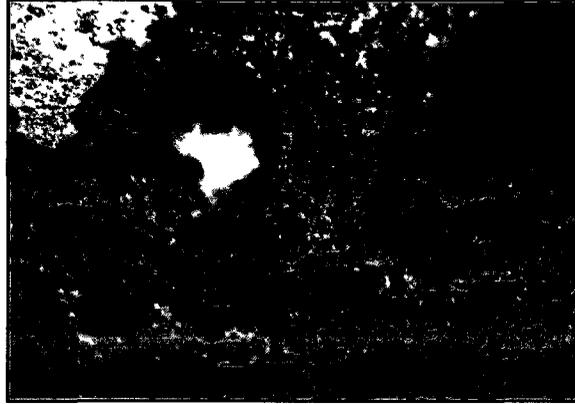


Figure 2.5 MFL Detectable Pitting Corrosion in Metal Pipe

UT derives pipe wall thickness measurements by relating the travel-time of the ultrasonic wave to the travel distance in the pipe material. Defect dimensions can be filtered out from dense grid measurements of the local pipe wall thickness. Figure 2.6 shows metal loss in a large scale that can be detected by UT.



Figure 2.6 UT Detectable Thickness Defect in Metal Pipe

#### 2.1.4 Deformation of Pipes

Both flexible and rigid pipes are subject to cross sectional deformations. Normally, a polymeric pipe or liner often exhibits a continuous shape because of its good ductility to

deform to find a new equilibrium state under loadings. On the other side, a rigid pipe made of steel, iron, or clay lacks of flexibility for the stress redistribution, so fractures often propagate from some initially high stress concentration area and the fracture lines represent discontinuities in the shape. Whether a pipe exhibits flexible or rigid behavior is determined a number of factors including the material property, cross sectional shape, the pipe–soil system interaction, temperature and humidity. Under certain circumstances, a polymeric pipe can be “rigid” and yield discontinuity in its shape, and a metallic pipe can exhibit a fair amount of ovality.

Rogers [24] performed a cross sectional geometry analysis of underground flexible pipes, gave an overview of qualitative cross sectional deformation evaluation, and proposed some conceptual and continuous cross sectional shapes as shown in Figure 2.7 (from left to right: ellipse, heart, inverted heart, and filleted rectangle).

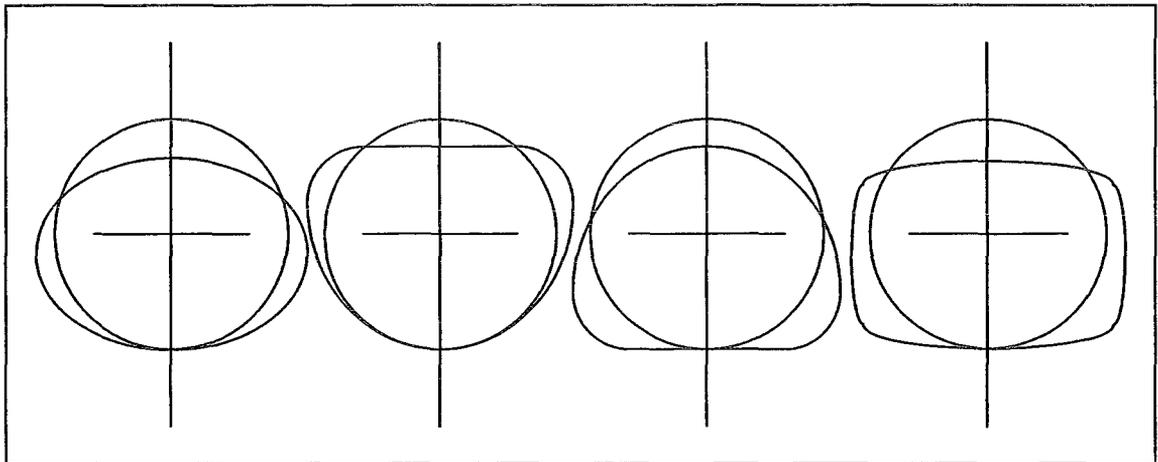


Figure 2.7 Several Conceptual Cross sectional Deformations

It was stated in [34] that when a thin pipe buckled, the elastic effects became less significant and the pipe material can be considered perfectly plastic. It was assumed that deformation occurred only at the hinges but nowhere else, as shown in Figure 2.8.

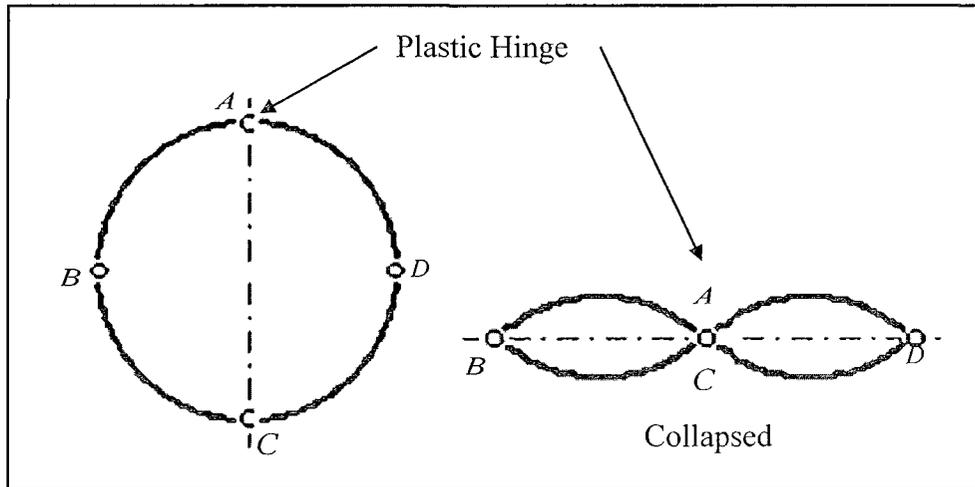


Figure 2.8 Cross Sectional Deformation Progressions with Four Plastic Hinges

In this research, elliptical and hinged cross sections are chosen to be the two basic shapes for pipes as they are mostly observed in the field. Non-uniform cross sectional deformations along the pipeline can occur if the loadings distribution and material properties are not uniform. For non-uniform deformations, it is assumed that the shape mode does not change over a short distance and only the severity of deformation changes. In other words, elliptical and hinged cross sections do not occur adjacent to each other.

#### 2.1.5 Global Layout Shift of Pipes

Global layout shift can be divided into two groups: rigid-body movement caused by loose joints and flexible deflection as a hollow beam under loadings, determined by joint strength together with pipe material, shown in Figure 2.9. Note that in order for the pipe to connect to the manhole while still being continuous, longitudinal elongation shall be present for both cases. Global layout shift of a continuous rigid pipeline is usually small compared to the pipeline length. However, for cases where pipes are relatively flexible and longitudinal elongation is not negligible, global layout shift can be obvious, mostly in the form of sinking.

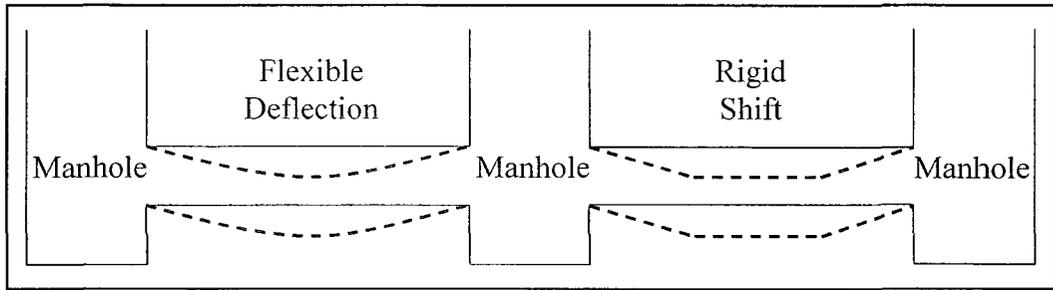


Figure 2.9 Flexible and Rigid Global Shift

## 2.2 Geometry Scanners

Governmental Accounting Standards Board (GASB) published Statement 34 and placed more responsibility in governments in the public assets management. Under the regulation of GASB 34, governments have to emphasize the need for long term maintenance, which is be accounted for in financial reports. GASB 34 requires a full water and sewer assets assessment at an interval of three years or less. Traditional techniques like closed-circuit television (CCTV) and Sewer Scanner & Evaluation Technology (SSET), though still dominant in industry, only yield qualitative information. Therefore, the assessment is totally dependent on the operator's expertise and experience, which is tedious and prone to errors. Advanced quantitative scanning technology is needed for municipalities in order to comply with GASB 34. It is necessary to take a review of the current geometry scanning technology because the form of the acquired data is closely dependent on the specific technology used. Factors such as noises, vision occlusion, resolution, etc. associated with each technology are discussed in this section.

A scanner in this research is a device that analyzes objects or environment by collecting geometry data, possibly together with images. The collected geometry data and images can then be used to construct digital, 3-D models useful for a wide variety of

applications. Such devices are used extensively by the entertainment industry in the production of movies and video games. Other common applications of this technology include industrial design, orthotics and prosthetics, reverse engineering and prototyping, quality control and inspection, and documentation of cultural artifacts. Many different technologies can be used to build the scanning devices; each technology comes with its own limitations, advantages and costs. Scanners can be categorized as contact and non-contact. Contact scanners mechanically touches the surface of an object, whereas non-contact scanners usually emit some kind of radiation or light and detect or analyze its reflection or pattern using physics laws in order to measure an object. Time-of-flight, SONAR, optical triangulation, and stereo vision are popular techniques used in non-contact scanners.

### 2.2.1 Contact Scanner

Contact scanners measure the object by physical touch. A hand driven touch probe used to digitize a clay model in the computer animation industry is shown in Figure 2.10.



Figure 2.10 Contact Scanner for Digitization of Portrait

A contact scanner is usually very precise by taking a direct measurement without a following signal processing step as commonly used in a non-contact scanner. One disadvantage of the contact scanners is that the actual touch from the probe may modify or damage the object, so they are usually prohibited from measuring delicate or valuable objects such as historical artifacts, or highly flexible objects such as sponges and jelly. Another disadvantage is that they are relatively slow compared to other scanning technologies. Extending and retrieving the arm that the probe is mounted on can be very slow and the fastest coordinate measuring machine can only operate on a few hundred hertz. In contrast, an optical system like a laser scanner can operate from 10 to 500 kHz.

### 2.2.2 Time of Flight Laser Scanner

A time-of-flight scanner, as shown in Figure 2.11 [32], finds the distance to a surface by timing the round-trip time of a pulse of light. A laser is used to emit a pulse of light and a sensor is used to detect the reflected light. Since the speed of light  $c$  is known, the round-trip time  $t$  determines the traveling distance of the light, which is twice the linear distance  $D$  between the scanner and the surface, i.e.,  $D = ct/2$ .

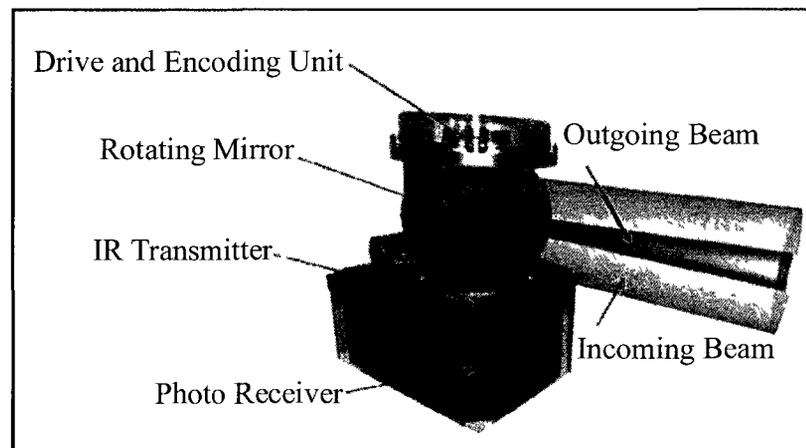


Figure 2.11 Principle of Time-of-Flight Laser Scanner

Because of the large magnitude of light speed, an error on the order of millisecond in  $t$  can lead to a significantly different  $D$ , therefore making the time-of-flight scanner more suitable for long-range application such as land surveying.

### 2.2.3 SONAR Scanner

A Sound Navigation and Ranging (SONAR) scanner employs a similar principle as a time-of-flight laser scanner. It determines the distance of a surface by measuring the time required for an emitted sound wave to be reflected by the surface and returned to the SONAR unit, shown in Figure 2.12 [32]. Unlike laser, SONAR can also be used for underwater measurement because the sound wave penetrates through the water.

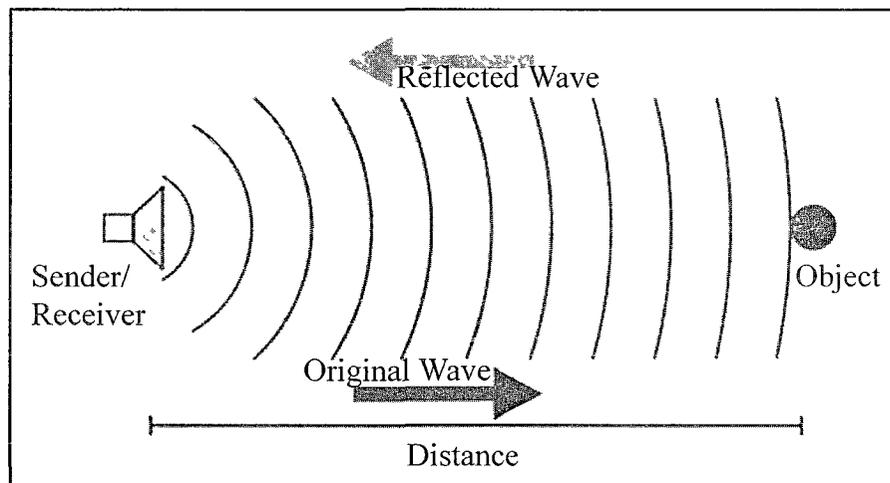


Figure 2.12 Principle of SONAR Scanner

### 2.2.4 Triangulation Scanner

An optical triangulation laser scanner projects a laser onto the object and exploits a camera to take a picture and analyze to find the location of the laser dot in the image. Varying with the distance between laser emitter and object surface, the laser dot appears at different places in the image plane. A triangle is formed by the laser emitter, the laser dot,

and the camera. Relationship can be established between  $\Delta x$  and  $\Delta x_1$  in Figure 2.13 [32] using trigonometry. Because  $\Delta x$  is known at the laser dot searching step, the distance between laser emitter and object surface can be calculated. In order to speed up the data acquisition process, a laser stripe or grid pattern is projected to sweep across the object.

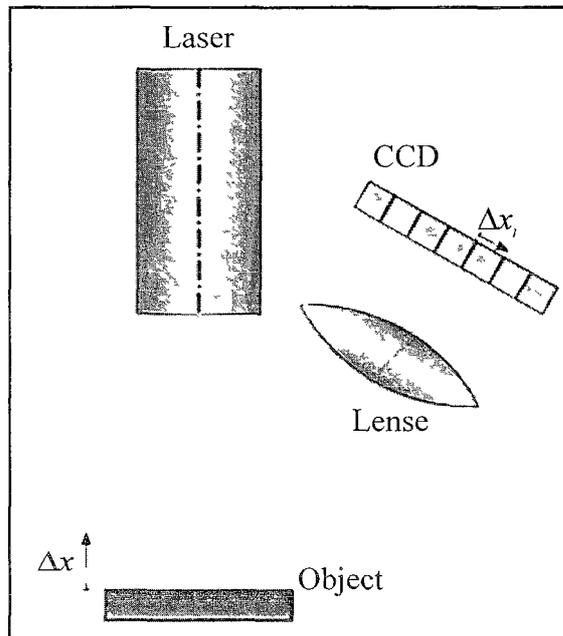


Figure 2.13 Principle of Optical Triangulation Laser Scanner

The potential of optical-laser triangulation scanning for its speediness, high resolution and non-destructiveness makes it an attractive alternative to contact scanners. However, the performance of this type of scanner is dependent on the object surface under measurement. Shiny, mirroring, transparent and rough objects pose many problems to optical scanners.

### 2.2.5 Stereo Vision Scanner

Unlike a single camera used in optical-laser system, stereo vision system usually employs two cameras, slightly apart, looking at the same scene. By analyzing the slight

differences between the images seen by each camera, it is possible to determine the distance at each point in the images using triangulation, since the extended lines of  $O_L X_L$  and  $O_R X_R$  shall intersect at  $X$ , as shown in Figure 2.14 [32]. Stereo vision scanners require heavy image processing computations and ample illumination to be feasible.

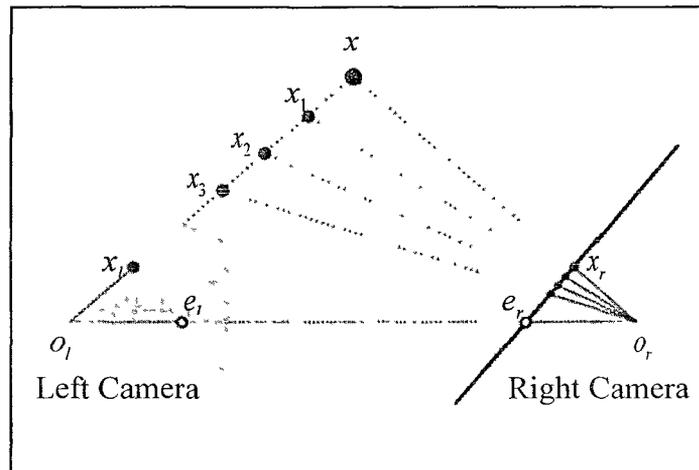


Figure 2.14 Principle of Stereo Vision Scanner

### 2.2.6 Comparison and Selection

To summarize, Table 2.1 lists several qualitative criteria for selection of scanning technologies for pipe inspection applications (not just water and sewer pipes). No single technology is superior to the others in all circumstances. Note that cutting-edge technologies are emerging; therefore, Table 2.1 only reveals a basic idea of pros and cons based on the average practice in the terms of scanning technology. Nevertheless, it shall serve as a general guideline and a quick reference to aid the selection of measurement principles. The code designations are as follow: T.O.F: Time of Flight, O.T.: Optical Triangulation, S.V.: Stereo Vision, L: Low, M: Medium, H: High, S: Small, L: Large, Y: Yes, N: No, P: Possible, E: Easy, and C: Complex.

Table 2.1 Scanning Technologies Qualitative Comparison

Technique	Contact	T.O.F.	O.T.	SONAR	S.V.
Accuracy	H	L	L-H	M-H	M-H
Algorithm	E	E	M	E	C
Cost	L	H	M	H	M
Destruction	P	N	N	N	N
Diameter	S-M	L	S-M	M	S
Light	N	N	N	N	Y
Resolution	M	H	H	M	H
Speed	L	H	M-H	M-H	M-H
Surface	L	L	H	L	M
Water	N	Y	Y	N	Y

Based on Table 2.1, the scanning technology selection for underground water and sewer pipe inspection is discussed next. First, time-of-flight scanner is not suitable for water and sewer pipe inspection application because of the range limitation. A small error in round-time detection can lead to a very large distance difference due to the high speed of light. For an underground pipe whose diameter usually ranges between several inches to a few feet, the measurement error can be comparable to or even exceed pipe diameter, therefore making the resulting data very unreliable. Although some new time-of-flight camera emerges recently with lower range limit about a few feet, the complexity of the semiconductor process and the expensive cost of such a device make it still unpopular in the market.

Second, water and sewer pipes are usually rigid enough to endure the forces exerted by probes. The low cost makes contact scanners attractive to contractors when abundant inspection time is allowed. Contact scanners are most appealing when measuring a pipe with water running inside or with rough surface where non-contact scanners fail or encounter a great amount of noises. However, because of their slowness, contact scanners

are not preferred in a water and sewer pipe inspection when a tight time frame is specified by the contract with municipalities.

Third, stereo vision scanner captures, analyzes, and extracts useful information out of images, so only a single image collection action is needed and it saves the online time. Complex offline image processing is involved when converting intensity information to geometric coordinates. Also, a light source needs to be integrated with the scanner because there is not much illumination underground. Even if a light source with enough power is installed with the scanner, the computational and storage burden make it impractical to scan a pipe with its length on the order of miles.

Fourth, optical-laser scanners offer a good trade-off between speed and accuracy. Underground pipes have a dark ambience with little or no other light source, which is a desired characteristic for laser to reach a maximum contrast value. However, accuracy is be greatly reduced at the occurrence of rough surfaces and water because of reflection scattering and refraction. Figure 2.15 depicts a laser ring scattered by the water at the bottom, according to Bennett [3].

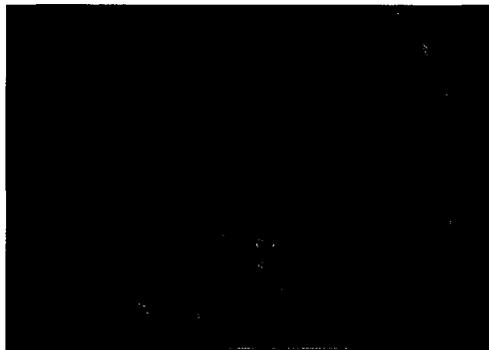


Figure 2.15 Laser Ring Scanning for Pipe with Water

Fifth, SONAR, on the other hand, overcomes the biggest challenge to laser at the existence of water. Water levels fluctuate depending on factors such as the time of day and storm events. This is particularly true with larger diameter pipes, where it is common to have a percentage of the pipe always underwater. Figure 2.16 [31] shows a full circumferential SONAR scan with a clear evidence of water at the bottom.

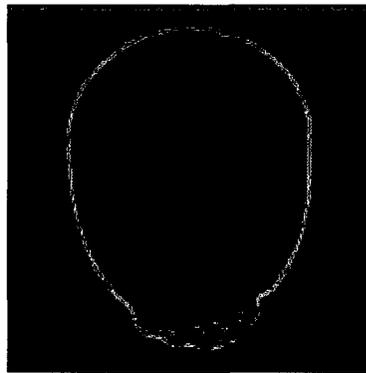


Figure 2.16 SONAR Scanning for Pipe with Water

However, studies showed that air sonar used for measurement above the flow line was not sufficiently accurate over the larger distances involved in the large diameter pipes to allow valid condition assessment. Moreover, SONAR cannot be operated in both air and water simultaneously because different electronics and transducers are needed. It records only the part of the pipe that is not under water, or records only the part that is below water level.

In conclusion, for water and sewer pipes inspection, laser combined with SONAR emerges as a promising technology for pipe geometry measurement. The full circumferential coverage of the cross sections makes it competitive in a complete pipe assessment program. Contact probes, despite their speed limitation, take direct measurement regardless of the pipe's interior surface condition and shall be a good

complement to the laser–SONAR system. Such a triple-play system, though not commercially available yet, shall offer a comprehensive and robust solution for the purpose of 2-D cross section geometry measurement.

Many states require laser scanning of pipes for any new installations along or crossing a DOT (Department of Transportation) right-of-way. The state of Florida required new large diameter, rigid and flexible pipes installed after 2006 to be laser scanned on a regular basis. Scanners that utilize laser as the sole or part of the methodology are the trend for quantitative inspection.

### **2.3 Scanner Carrying Robotic Platforms**

Factors that have to be defined, examined and weighed when selecting a proper robotic platform include budget, time, environmental impact, supporting equipment capability, pipe and joint strength, and market availability. For underground water and sewer pipes, a light-weight robot that is less expensive, that is configurable to handle complex interior conditions (curvature, bends, bumps, water, etc.) robustly, that performs the inspection quickly, that does not require an entire service line shutdown, that easily telecommunicates with the control center, and that is capable of estimating its own pose shall be given prior considerations. All the aforementioned criteria shall be based on field experience, contract terms and sound engineering judgment.

Many robotic systems have been prototyped in research facilities and commercialized by vendors or suppliers. Several examples of these advanced robots with their application markets are shown in Figure 2.17 [33], Figure 2.18 [35], and Figure 2.19.



Figure 2.17 ROVVER 600 Inspection Robotic Platform for Water and Sewer Pipes

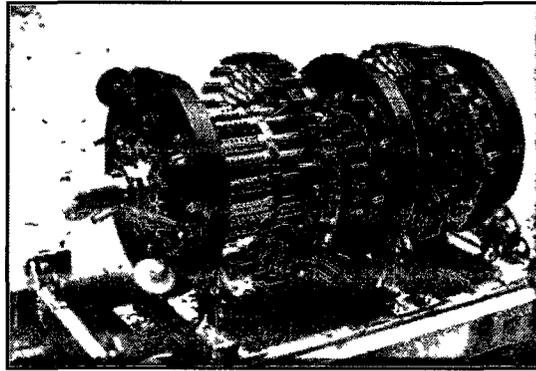


Figure 2.18 Rosen Geometry PIG for Oil and Gas Transmission Pipes

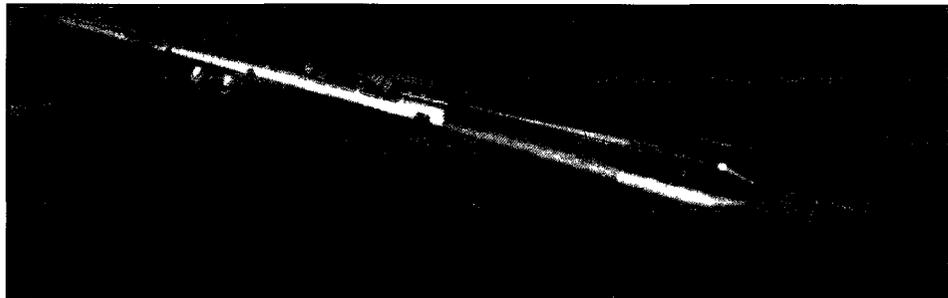


Figure 2.19 Cable Powered WellTec Hybrid Robotic for Oil and Gas Wells

The majority of commercialized robotic platforms used for water and sewer pipe inspection are the vehicle-like robot with wheels at the bottom. Figure 2.20 [31] demonstrates the concept of such a typical robot platform carrying a 2-D laser scanner.

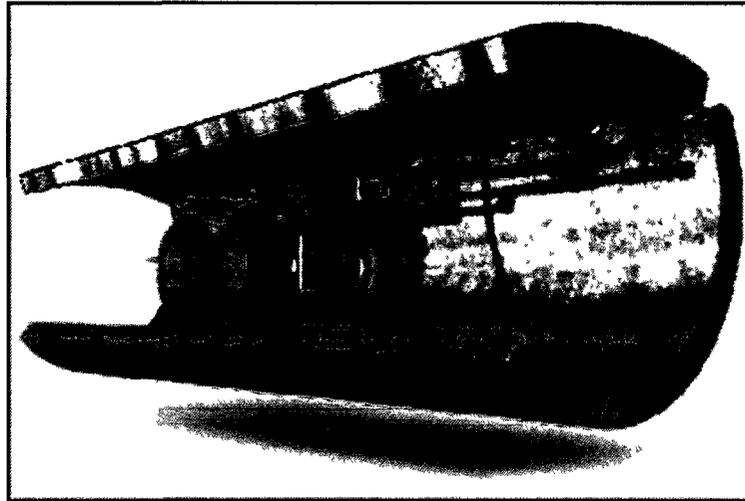


Figure 2.20 Common Robotic Platform with Laser Rings

Schempf [26] showed a detailed comparison of robotic platforms according to the criteria as follow: pipe market segment, technology maturity, design architecture, obstacle handling, locomotion mode or type, repair capability, inspection capability, sensor type, tethered, power and communication type, and range per launch. The findings for civil infrastructure is of interest and reproduced as in Table 2.2. Corresponding letter codes for the water and sewer pipes are as follow: 1. water (W), sewer (S); 2. mature, reliable & cash generator (Comm); 3. monolithic (M); 4. straight pipe only (SPO), minimal curvature (MC); 5. fixed-wheel crawler (FW), fixed-tracked (FT), push-pull tether (PPT); 6. video (V), ultrasonic (UT); 7. line voltage (LV), tether link (TL). Once again, traditional bottom-wheel driven robotic platforms serve well for water and sewer pipes and dominate the market.

Table 2.2 Inspection Robotic Platforms for Water and Sewer Pipes

Category	Civil Infrastructure
Pipe Market Segment	W/S
Technology Maturity	Comm
Design Architecture	M
Obstacle Handling	SPO, MC
Locomotion Mode/Type	FW, FT, PPT
Repair Capability	Y
Inspection Capability	Y
Sensor Type	V, UT
Tethered	Y
Power & Communication Type	LV, TL
Range per Launch	<200 feet

## 2.4 Robotic Deviation

Using the terminology from aeronautics, a whole set of pitch, yaw and roll is needed to fully explain the robotic deviation [32], as demonstrated in Figure 2.21. It shall be noted that pitch, yaw and roll shall be designated in a predefined coordinates system, usually the earth's coordinates system.

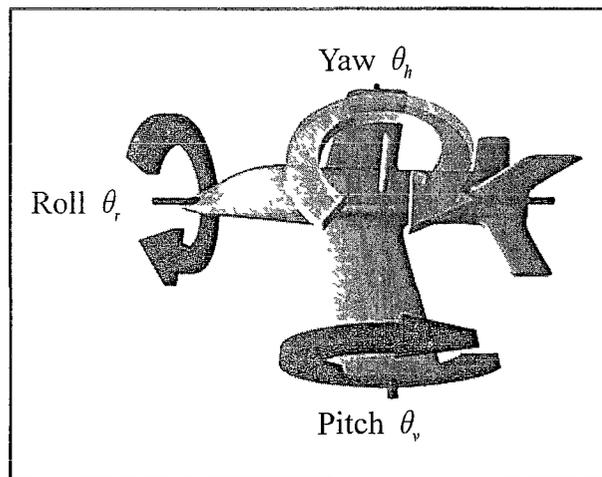


Figure 2.21 Pitch, Yaw and Roll

In order to calculate the ovality accurately, cross sectional measurement shall be taken to derive the maximum and minimum diameter. A true cross section is defined as a portion of the plane that is perpendicular to the pipe axis and confined by the intersection curve between the plane and the pipe circumference. For a 2-D scanner, the configuration of laser projector, SONAR emitter or contact probe shall be well calibrated to generate the exact cross section coverage when the robotic platform moves parallel to the pipe axis. A deviation off the pipe axis yields an oblique measurement that shall not be used for the calculation of factors such as diameter, ovality, curvature, and area.

If the pipe diameter is large compared to the width between two front wheels or two rear wheels, the robot seems to travel on a “flat” territory, which does not offer enough constraint against the wheels’ side movement. True cross sectional measurement is not achieved with robot turning left and right in Figure 2.22, according to Dettmer [8].

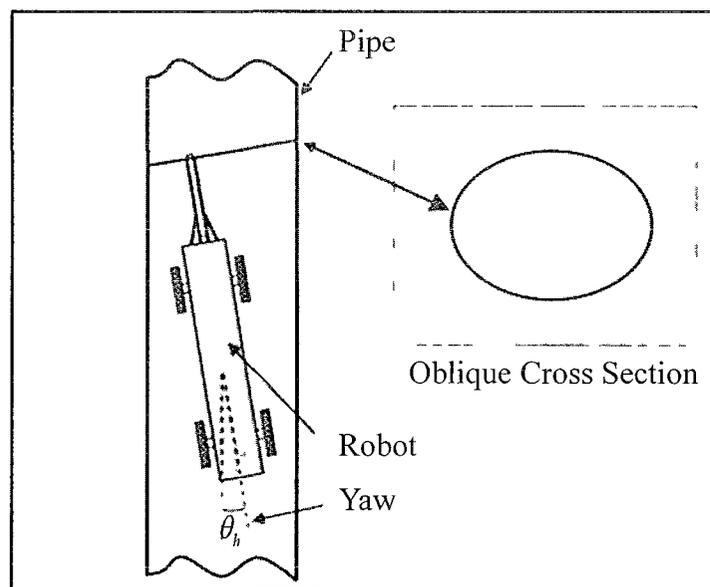


Figure 2.22 Robotic Yaw with Oblique Cross Sectional Scan

Any sediments or bottom deformation tilt the robot up and down, therefore also causing erroneous cross section measurements in Figure 2.23, according to Dettmer [8].

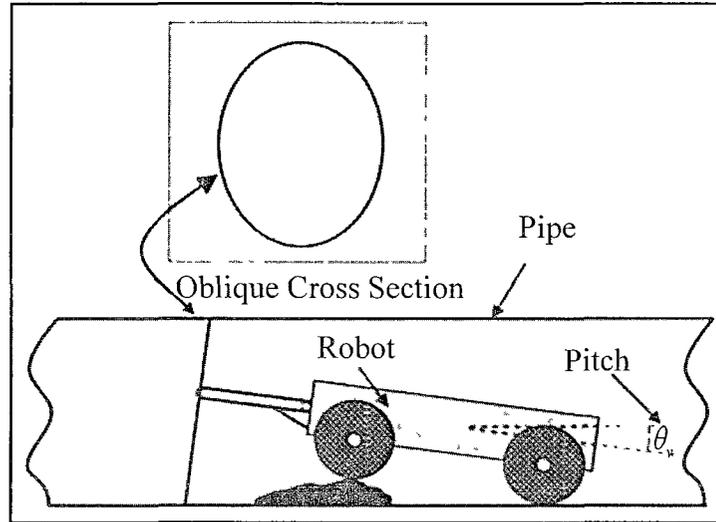


Figure 2.23 Robotic Pitch with Oblique Cross Sectional Scan

In reality, a robot not only experiences horizontal and vertical deviations, but it is also possible to rotate around the pipe's centerline. Such a rotation affects the cross sectional measurement by changing the indexes of the sampled points, causing a wireframe distortion as shown in Figure 2.24. Such a distortion affects the interpolation between cross sections, especially sampled sparsely, and shall be corrected in order to apply texture to the facets.

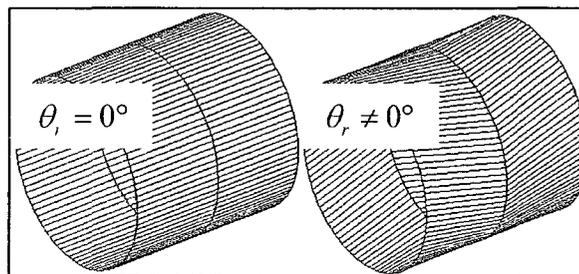


Figure 2.24 Wireframe Distortion Caused by Robotic Roll

For a 3-D scanner, the orientation configuration requirement is less stringent because theoretically alignment of successive scans with sufficient overlap can restore any deviation. Since the main purpose of a 3-D scan is to collect more data for each scan and therefore speed up the scanning process, allowing too much overlap requires more time for scanning the same region twice; besides, interpolation has to be employed to locate the cross section. It is still preferable to set up the 3-D scanner so that direct cross section measurement can be easily obtained when there is no robotic deviation.

### 2.5 Inertial Navigation Systems

Some commercialized laser scanners for oil and gas pipelines use on-board inertial navigation systems (INS) such as accelerometers and gyroscopes to track the movement of the robot in the earth's coordinates system. Accelerometers measure the linear acceleration and gyroscopes measure the angular velocity (pitch, yaw and roll) of the robot in the inertial coordinates system, as shown in Figure 2.25.

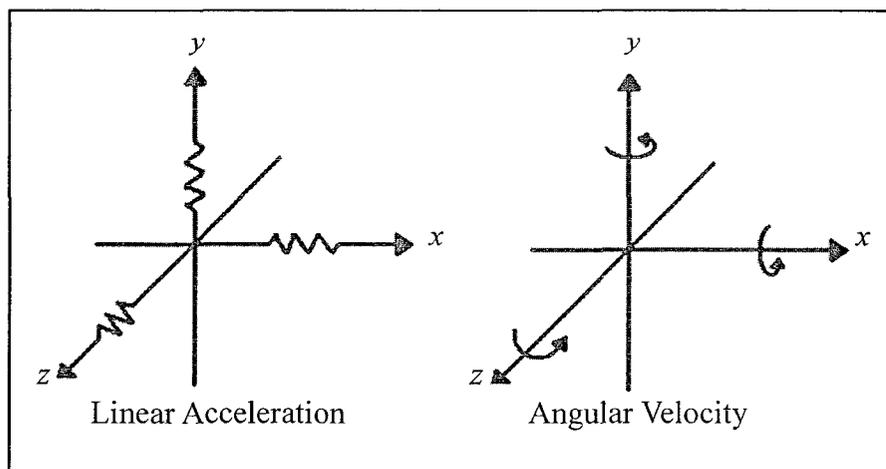


Figure 2.25 Linear Acceleration and Angular Velocity

Integration of the angular velocity with initial orientation yield the current orientation, while integration of the linear acceleration and angular velocity with initial velocity yield the current linear velocity, which is integrated again with the initial position to obtain the current position. So with known initial pose (position and orientation), the current pose can be calculated. However, since integration is applied twice, small measurement errors of linear acceleration from accelerometers and angular velocity from gyroscopes are integrated into larger errors in linear velocity and orientation estimation, and even larger errors in position estimation. Because new poses is deducted from previous poses, the integration errors increase cumulatively at a rate roughly proportional to the time. Therefore, the pose information shall be periodically corrected (zero updating) by input from some other type of navigation system such as global positioning systems (GPS).

INS is often installed on smart pipeline inspection gauges (PIG), as shown in Figure 2.18, to calculate the geographical location of oil and gas pipes. PIGs are pushed by the product inside the pipe and require the internal pipe diameter to remain relatively constant to guarantee a safe passage. A PIG equipped with accelerometer and gyroscope-based sensors can take geometric measurement, record the pose of the robot, and log the information together. PIG's hardware configuration is in a sense preferable than the regular bottom-wheeled robot, because the multiple contact points between the PIG and the pipe limit the undesirable robotic deviation that can be misrepresented as a change of direction of the pipe. The reliability of the 3-D model of the pipe depends on the accuracy of the pose recorder.

For a pipeline above the ground, GPS units can be placed at predefined locations and triggered by the robot as it passes through, supposedly to help reduce the integration

error accumulation in the algorithm used by accelerometers and gyroscopes. But in the underground pipe inspection, geographical layout of the pipe is not always available before measurement in a non-destructive way. There is no guarantee that GPS units can be placed directly above or even close to the pipe; therefore, the deduced pipe layout can be greatly distorted without accurate placement of the GPS units.

## **2.6 Geometry Data Study**

The pipe geometry is acquired using a certain type of scanning technology. Each technology yields a unique type of data and affects the format of how the data is stored. Factors that can create various data types include: dimensionality (2-D or 3-D), coordinates system (Cartesian or polar), on-board sensors for pose tracking (data self-correction), the circumferential and longitudinal sampling resolution (hardware limitation), and the probability of data loss (vision occlusion).

### **2.6.1 2-D Geometry Data**

A side-facing scanner was mounted and realized on a robot at Louisiana Tech University, as shown in Figure 2.26 (top left). The scanning head consisted of an active laser projector and a passive photogrammetric camera, facing the side along the radial axis of the pipe. Under the control of step motors and LabVIEW®, the scanner collected range images of the full circumference. An image of a laser line captured by a side-facing camera is shown in Figure 2.26 (middle left). Optical triangulation technique was applied to calculate and retrieve geometry information of the interest points lightened within the laser line at the stop position. Then the robot travelled to the next position to take another set of full circumferential measurements. Longer pipe geometry can be acquired by this stop-scan-go fashion. Final representation of data was in a form of consecutive rings of

points with their longitudinal separations based on the odometer, shown in Figure 2.26 (bottom left). Longitudinal and circumferential resolutions were determined by the resolutions of the step motors used to propel the wheels and to revolve of the scanning head. The side-facing scanner was slow due to its revolving mechanism and small coverage from each scanning action, but it was robust against vision occlusion so that a more complete model can be generated.

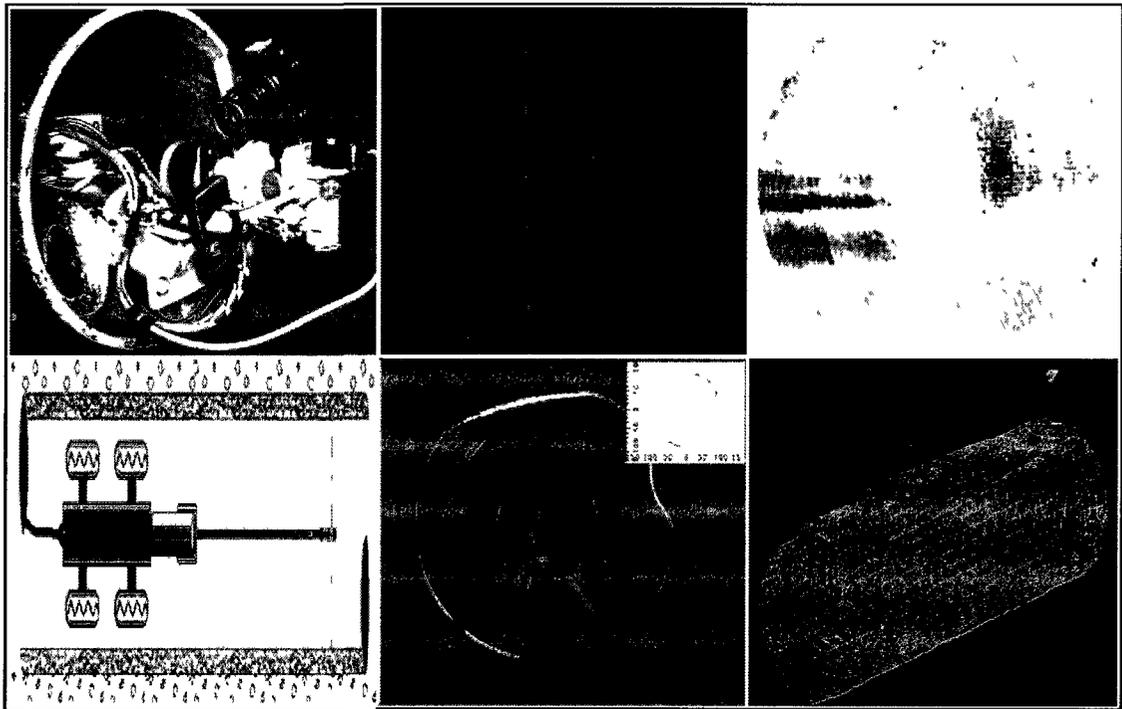


Figure 2.26 Two Dimensional Scanners with Measurements

The most popular type of scanner in the pipe inspection industry utilizes a forward-facing camera. Usually, a laser ring is projected onto the pipe surface in front of the camera. Forward-facing scanners can operate in a more continuous fashion because videos can be recorded as the robot moves forward. Geometry information can be acquired offline by analyzing video frames. This hardware configuration has an infinitely large

resolution in theory, but sampling, both longitudinal and circumferential, shall be taken to reduce the data size for wireframe modeling and geometry information storage. Also, vision occlusion can be a problem if a large area of deformation gets in the way of the camera's viewing angle and hides a portion of the laser behind its back. An illustration of this scanner type, the laser placement and the generated wireframe, is shown in Figure 2.26 (bottom row).

### 2.6.2 3-D Geometry Data

A 3-D laser scanner has the merit of acquiring more data for each scanning action and reserving depth information that is valuable for software-based deviation correction. If adjacent scans allows for data redundancy (a segment is measured at least twice), the scanned points shall be transformed to have very similar coordinates or minimize some error metric since the points come from the overlapping area. In this case, the data is supposed to find the transformation that can approximately correct the unwanted coordinate's difference caused by the robotic deviation, so INS is not a necessity in a 3-D scanner; however, it is still desirable because it provides a good initial guidance of where to search in the transformation space.

There are two common methodologies to construct the scanning head with step motors in the pipe inspection industry. In the first kind, a stable camera looks constantly forward, and a laser unit, controlled by a step motor, independently rotates to project or sweep laser dots onto the pipe's interior surface in the front; in the second kind, a camera and a laser unit, fixed together and controlled by a step motor, rotates to enlighten and capture the laser line. The range of the scanning is usually within  $180^\circ$ , regardless of which methodology is used to construct the scanning head. Because the step motor usually

provides an equal-angled increment, the placement of laser dots has different densities along the pipe so that points closer to the scanner are denser than those far away from it, as shown in Figure 2.27.

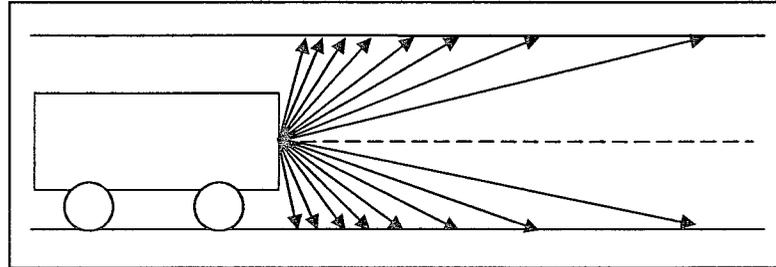


Figure 2.27 Three Dimensional Scanner with Non-Uniform Measurement Density

### 2.6.3 Sampling, Noise and Storage

The measurements have to be sampled for either further processing or storage for plotting purpose. A 3-D scanner, using a points-cloud style, samples the geometry by hardware settings; a side-facing 2-D scanner also takes discrete measurements. For a forward-facing 2-D laser ring, however, the measurements are continuous in the absence of vision occlusion. There are two ways to sample the circumference of such a laser ring: rectangular grid or radial grid.

Rectangular grid sampling involves two phases. First, a vertical grid is used to find the intersection of the grid lines and the laser ring, which gives a good coverage of the top and bottom regions but a sparse coverage of the left and right sides; then a refining horizontal grid is applied to compensate. Random measurement noises are in line with the grid, as shown on the left side in Figure 2.28. Rectangular grid sampling is invariant to the measurement origin offset, but the origin has to be estimated for the placement of the second set of grid lines.

A radial grid samples circumferential data at a constant angular increment. A more uniform coverage is achieved using the same amount of data points in the radial grid than in the rectangular grid when the measurement origin is close to the pipe's centerline. Random measurement noises are in line with the radial direction, as shown on the right side in Figure 2.28. However, radial grid sampling is subject to the measurement origin offset. The cause of the measurement origin offset is to be discussed more closely in Section 4.1.

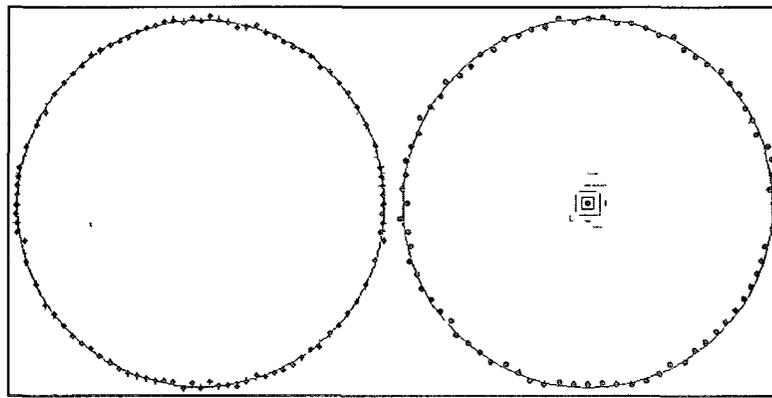


Figure 2.28 Rectangular and Radial Grid Sampling

Pipe geometry data can be stored in a variety of ways depending on the specifications and preferences. The data format from a 2-D scanner is examined next. For odometer-based readings, the data can be stored in a pseudo 3-D coordinates system  $(x', y, z)$ :  $y$  and  $z$  indicate the cross sectional measurement, perpendicular or oblique;  $x'$  indicates the longitudinal location of the measurement. Note that the  $x'$ -axis is not referenced to the earth, but simply a straight line in the camera's coordinates system where pairs of  $(y, z)$  are given for every point on the straight line, i.e.,  $x'$ . A cylindrical system  $(x', r, \theta_a)$ , shown in Figure 2.29, can be designated to display the coordinates for a better understanding of the circumferential condition, especially when the measurement origin

stays close to the pipe's centerline. The pair  $(r, \theta_a)$  converts easily to the Cartesian coordinates by  $z = r \cos \theta_a$  and  $y = r \sin \theta_a$ , assuming  $\theta_a$  is the angle between the  $z$ -axis and the line connecting the interest point to the origin.

For a 3-D scanner, since the data can partially correct the robotic deviation if an overlap exists, the triplet  $(x, y, z)$  is in the earth's coordinates system, where the  $z$ -axis represents the same path of the robot. A spherical system  $(\rho, \theta_z, \theta_a)$  directly read from the scanner's revolving configuration, requires no extra calculation and can be used to record the position of an interest point to the origin, and it can be converted to the Cartesian coordinates by  $x = \rho \cos \theta_z$ ,  $y = \rho \sin \theta_z \sin \theta_a$ , and  $z = \rho \sin \theta_z \cos \theta_a$ , also shown in Figure 2.29.

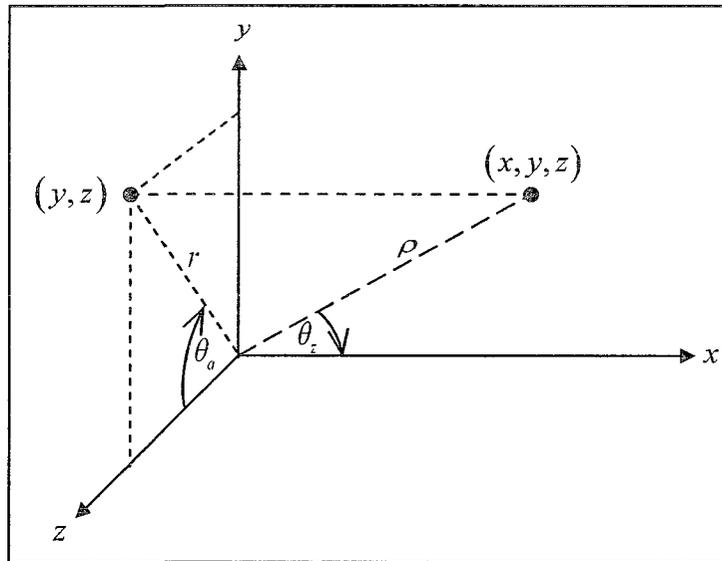


Figure 2.29 Cylindrical and Polar Coordinates System for Geometry Data Storage

In conclusion, the pipe's geometry data can be samples and stored in different ways with different noise characteristics. The natural way to sample and save the data is radial

sampling with polar coordinates, a form that is consistent with the hardware mechanism and easy to interpret. For visualization purpose, data can always be converted into Cartesian coordinates. In this research, polar and Cartesian coordinates is used interchangeably.

## 2.7 Camera Lens Models

Images are often collected using cameras in the process of geometry scanning. Image-based detection of features such as joints and laterals is possible and complementary to the geometry-based detection. For this purpose, the models of the camera lens need to be established.

Every point, relative to the 3-D camera's coordinates system, has an azimuth angle  $\theta_a$  and a zenith angle  $\theta_z$ , as shown in Figure 2.29.  $\theta_a$  can be thought as an indicator of which quadrant an interest point lies in the image plane, i.e., left, right, top, or bottom;  $\theta_a$  remains the same in the 3-D space as in the image plane.  $\theta_z$  is defined as the angle between the incoming light and lens' optical axis and determines the radius of that point to the image center in the image plane.

$$\theta_z = \arctan \sqrt{\frac{z^2 + y^2}{x^2}},$$

$$\theta_a = \arctan \frac{y}{z}.$$

It is the goal of the intrinsic camera calibration to find the mapping function between the zenith angle  $\theta$  and the radius  $r_i$  in the image. Such a mapping can be established as  $r_i = f(\theta)$ . Many specific models have been proposed to approximate the

mapping function. Some popular models are listed as Eq. (2.1), Eq. (2.2), Eq. (2.3), Eq. (2.4), and Eq. (2.5):

Equi-distance Projection

$$r_i(\theta_z) = k\theta_z, \quad (2.1)$$

Sine-Law Projection

$$r_i(\theta_z) = k \sin \theta_z, \quad (2.2)$$

Equi-Solid Angle Projection

$$r_i(\theta_z) = k \sin \frac{\theta_z}{2}, \quad (2.3)$$

Perspective Projection

$$r_i(\theta_z) = k \tan \theta_z, \quad (2.4)$$

Stereographic Projection

$$r_i(\theta_z) = k \tan \frac{\theta_z}{2}, \quad (2.5)$$

where  $r_i(\theta_z)$  is the distance from an interest point to the center of the image,  $\theta_z$  is the zenith angle between the incoming light and the lens' optical axis, and  $k$  is a scaling factor. These five projection models usually suffice to calibrate the lens used in a specific application and the best model that fits the calibration data shall be used to calculate the distance between the interest point and a chosen origin (usually fixed with the camera).

## 2.8 Trenchless Polymeric Liners

The traditional way for rehabilitation of a deteriorated pipeline is digging trenches, removing problematic pipe segments and replacing them with new ones. This methodology, though straightforward, poses many environmental and safety issues to community residents and municipal governments. The cost effectiveness can be adversely influenced by significant road damages, disruption of traffic and long period of operation shutdown.

In an attempt to reduce the cost, disruptions, inception to completion time and other dormant problems associated with excavation and replacement of underground pipes, trenchless technologies have been developed over the years with many field successes. Among the many trenchless rehabilitation techniques proposed to restore the structural and fluid capacity of underground pipelines, polymer pipe lining has drawn much attention in the industry. The inherent strong corrosion resistance make polymeric lining products very competitive compared to the traditional pipe materials such as clay, concrete and metal. Cured-In-Place Pipe (CIPP), Deform-and-Reform Pipe (DRP) and Fold-and-Form Pipe (FFP), widely used across the country, are some popular examples of polymer pipe lining technologies.

CIPP is a proven, long-lasting trenchless method for renewing aging and deteriorated water and sewer pipes. The product placement is carried out by cable pulling or water/air pressure propelling, followed by a curing phase using steam, hot water, or ultraviolet light, so the resin-impregnated thermosetting liner is inverted to form a durable and seamless surface inside the host pipe. Figure 2.30 shows the inside of a cracked and corroded pipe before and after CIPP lining.

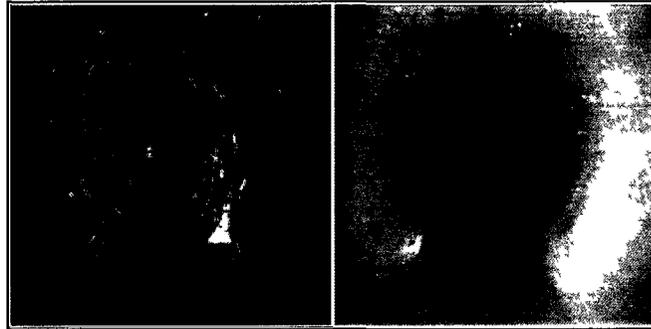


Figure 2.30 Interior Surface of Pipe before and after CIPP Lining

DRP is usually made of high density polyethylene (HDPE). The liner is extruded in a desired shape during the manufacturing process, deformed using a combination of heat and pressure, and wound onto spools ready for installation. Once in place, hot water or steam under pressure is fed through the deformed liner to soften and reform the pipe. The difference between FFP (Figure 2 31) and DRP is that FFP liner is usually made of polyvinyl chloride (PVC) and the liner is extruded in a folded shape during the manufacturing process. The placement and reforming of a FFP liner follows similar procedures as in DRP.



Figure 2.31 FFP Liners before and after Deployment

Before a liner installation, the host pipe shall be inspected by Closed Circuit Television (CCTV) for root penetrations, sediment deposits and lateral locations. Then a

proper cleaning tool shall be selected based on the obstacle severity to perform a thorough removal of obstacles. Another CCTV inspection may be necessary to ensure that the pipe is free of large obstacles. After the liner is installed, lateral connections are cut open either manually or by a remote controlled robot. Liners are seamless and not prone to cracks and corrosion like metallic pipes, but lining defects can occur with inappropriate cleaning tool selection, inaccurate lateral location/size information, and non-uniformly resin saturation in Figure 2.32 (left to right: wrinkle, lateral overcut, and blister).

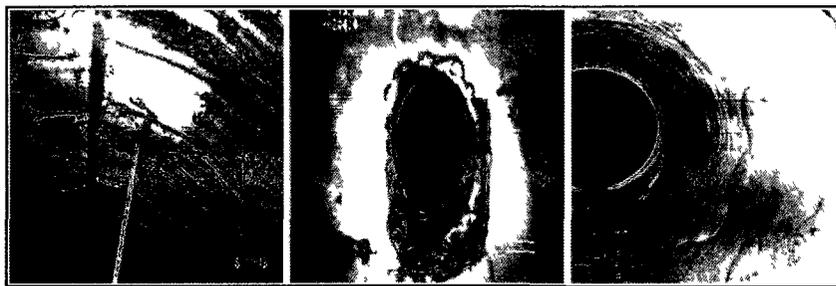


Figure 2.32 Lining Defects after Installation

Such lining defects are usually inevitable, especially in old pipes with complicated internal conditions. Fortunately, most water and sewer pipes are rated as low-consequence infrastructures in terms of hazard levels compared to oil and gas pipes; therefore, more tolerances are given to the rehabilitation outcome.

Polymeric liners are usually designed to meet at least a lifetime requirement of fifty years to be continuously functional without failure. Calculation of the actual loadings on a liner can be quite complicated because of the complex boundary and loadings distribution within a soil–pipe–liner system. Previous research and field tests often assumed that a partially deteriorated host pipe was strong enough to carry the static and dynamic loads passed from the soil and ground above. Therefore, for a partially deteriorated host pipe, the

liner is only subject to the hydrostatic pressures introduced by the internal fluid flow and the external groundwater. In this case, the host pipe provides a well-defined boundary for the liner.

Under this assumption, many researchers (Cai [6]; Chicurel [7]; Falter [9]; Glock [12]; Guice [14]; Gumble [15]; Knight [17]; Li [18]; Omara [19], [20], Straughan [27]; Zhao [30]) studied the structural performance of pipe liners. At Trenchless Technology Center of Louisiana Tech University, long-hour experiments were conducted to collect the data that facilitated the numerical modeling of the buckling behavior of CIPP liners only subject to external pressure.

Knight [17] used a flexible bladder between the host pipe and liner to simulate the external hydrostatic pressure and tested the buckling behavior of CIPP liners as shown in Figure 2.33. There was no initial imperfection of the liner in the left picture and the deformed liner clearly exhibits an elliptical cross section. In the right picture, a fiber optic cable was embedded between the bladder and liner to introduce an initial imperfection, and the deformed liner had a heart-like cross section with one lobe.

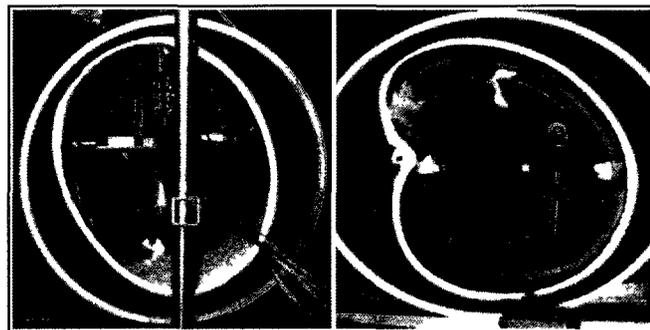


Figure 2.33 Elliptical and Lobed Cross Sections of Liners

Theoretically, the deformed cross sections can take any free form. In this research, elliptical and lobed cross sections are chosen to be the two basic shapes for liners as they are mostly observed in the field. As in pipes, non-uniform cross sectional deformations along the pipeline can occur. It is assumed that the shape mode does not change over a short distance and only the severity of deformation changes. In other words, elliptical and lobed cross sections do not occur adjacent to each other.

## 2.9 Mathematical Principles and Techniques

Since both geometry and image information can be acquired, mathematical techniques about processing of spatial coordinates and image pixels are of interest. Circle and ellipse fitting algorithms, Discrete Fourier Transform (DFT) of a contour and morphological operations on an image are introduced. This section provides introductory information regarding each of these topics. In the course of this research, these tools are to be used to develop new or improve existing solutions to engineering challenges associated with robotic deviation related distortions.

### 2.9.1 Circle and Ellipse Fitting

Circle fitting, as a special case of ellipse fitting, can be solved with unique mathematical forms. Given the coordinates of the  $N$  points along the intersection curve between the laser ring and pipe interior surface, namely  $(z_1, y_1), (z_2, y_2), \dots, (z_N, y_N)$ , a circle  $(z - z_c)^2 + (y - y_c)^2 = r^2$  is to be found that minimizes the sum of the squared distances from the points to the circle, i.e., the parameters (center  $z_c$  and  $y_c$ , and radius  $r$ ) shall minimize the geometric measure:

$$\text{SSE}(z_c, y_c, r) = \sum_{i=1}^N \left[ r - \sqrt{(z_i - z_c)^2 + (y_i - y_c)^2} \right]^2.$$

The first-order derivatives of the sum of squared errors  $\text{SSE}(z_c, y_c, r)$  with regard to  $z_c$ ,  $y_c$ , and  $r$  are equal to zero, i.e.,

$$\frac{\partial \text{SSE}}{\partial r} = 2Nr - 2 \sum_{i=1}^N \sqrt{(z_i - z_c)^2 + (y_i - y_c)^2} = 0,$$

$$\frac{\partial \text{SSE}}{\partial z_c} = 2r \sum_{i=1}^N \frac{z_i - z_c}{\sqrt{(z_i - z_c)^2 + (y_i - y_c)^2}} - 2N\bar{z} + 2Nz_c = 0,$$

$$\frac{\partial \text{SSE}}{\partial y_c} = 2r \sum_{i=1}^N \frac{y_i - y_c}{\sqrt{(z_i - z_c)^2 + (y_i - y_c)^2}} - 2N\bar{y} + 2Ny_c = 0,$$

where  $(\bar{z}, \bar{y})$  is the average of  $(z_1, y_1), (z_2, y_2), \dots, (z_N, y_N)$ .

There are no closed form solutions for this system of nonlinear equations, so iterative techniques like the Gauss–Newton or Levenberg–Marquardt methods shall be employed to solve the equations. Local minimum is sometimes achieved for the set of nonlinear equation, yielding inaccurate estimates of  $z_c$ ,  $y_c$ , and  $r$ .

The minimization measure can also be chosen in an algebraic least-squares sense.

An algebraic representation of the circle can be expressed as Eq. (2.6):

$$\mathbf{F}(z, y) = a_c(z, y)(z, y)^T + (b_c, c_c)(z, y)^T + d_c = 0, \quad (2.6)$$

where  $a_c \neq 0$ . Substituting the coordinates of the points  $(z_i, y_i)$ ,  $i = 1, 2, \dots, N$  into

Eq. (2.6), a linear system of equations  $\mathbf{D}(a_c, b_c, c_c, d_c)^T = 0$  can be obtained, where

$$\mathbf{D} = \begin{pmatrix} z_1^2 + y_1^2 & z_1 & y_1 & 1 \\ \vdots & \vdots & \vdots & \vdots \\ z_N^2 + y_N^2 & z_N & y_N & 1 \end{pmatrix}.$$

For  $n > 3$ ,  $\mathbf{D}(a_c, b_c, c_c, d_c)^T$  has to be minimized in a least-square sense. Another representation form of a circle is:

$$\left(z + \frac{b_c}{2a_c}\right)^2 + \left(y + \frac{c_c}{2a_c}\right)^2 = \frac{b_c^2 + c_c^2}{4a_c^2} - \frac{d_c}{a_c}.$$

Once the vector  $(a_c, b_c, c_c, d_c)^T$  is found, the center and radius of the circle can be calculated accordingly as Eq. (2.7), Eq. (2.8), and Eq. (2.9):

$$z_c = -\frac{b_c}{2a_c}, \quad (2.7)$$

$$y_c = -\frac{c_c}{2a_c}, \quad (2.8)$$

$$r = \frac{b_c^2 + c_c^2}{4a_c^2} - \frac{d_c}{a_c}. \quad (2.9)$$

One way to solve  $(a_c, b_c, c_c, d_c)^T$  is using the singular value decomposition (Golub [13]), i.e.,

$$\mathbf{U} \cdot \mathbf{S} \cdot \mathbf{V}^T = \mathbf{D}.$$

Singular value decomposition of an  $m$ -by- $n$  matrix  $\mathbf{D}$  involves an  $m$ -by- $m$  matrix  $\mathbf{U}$ , an  $m$ -by- $n$  matrix  $\mathbf{S}$ , and an  $n$ -by- $n$  matrix  $\mathbf{V}$ . In other words,  $\mathbf{U}$  and  $\mathbf{V}$  are both squares and  $\mathbf{S}$  is of the same size as  $\mathbf{D}$ . Since  $\mathbf{D}$  is an  $N$ -by-4 matrix, therefore  $\mathbf{V}$  is a 4-by-4 matrix:

$$\mathbf{V} = \begin{pmatrix} v_{11} & v_{12} & v_{13} & v_{14} \\ v_{21} & v_{22} & v_{23} & v_{24} \\ v_{31} & v_{32} & v_{33} & v_{34} \\ v_{41} & v_{42} & v_{43} & v_{44} \end{pmatrix}.$$

The least squares estimate of  $(a_c, b_c, c_c, d_c)^T$  is the right singular vector corresponding to the smallest singular value of  $\mathbf{D}$  as in Eq. (2.10),

$$(a_c, b_c, c_c, d_c)^T = (v_{14}, v_{24}, v_{34}, v_{44})^T. \quad (2.10)$$

Another way to solve  $(a_c, b_c, c_c, d_c)^T$  takes a more direct approach by transforming Eq. (2.10) into Eq. (2.11):

$$(z, y)(z, y)^T + \left( \frac{b_c}{a_c}, \frac{c_c}{a_c} \right) (z, y)^T + \frac{d_c}{a_c} = 0,$$

$$(z, y)(z, y)^T + \left( \frac{b_c}{a_c}, \frac{c_c}{a_c}, \frac{d_c}{a_c} \right) (z, y, 1)^T = 0,$$

$$\left( \frac{b_c}{a_c}, \frac{c_c}{a_c}, \frac{d_c}{a_c} \right) \begin{pmatrix} z \\ y \\ 1 \end{pmatrix} = -(z^2 + y^2). \quad (2.11)$$

Substituting the coordinates of the points  $(z_i, y_i), i=1, 2, \dots, N$  into Eq. (2.11) to obtain Eq. (2.12):

$$\left( \frac{b_c}{a_c}, \frac{c_c}{a_c}, \frac{d_c}{a_c} \right) \begin{pmatrix} z_1 & \cdots & z_N \\ y_1 & \cdots & y_N \\ 1 & \cdots & 1 \end{pmatrix} = -(z_1^2 + y_1^2 \quad \cdots \quad z_N^2 + y_N^2). \quad (2.12)$$

For  $n > 3$ , Eq. (2.12) becomes an over-determined system of equations and can be minimized in a least-square sense using the QR decomposition (Press [22]), which is

implemented as the left matrix division operator (backslash “\”) in MATLAB®. After  $(a_c, b_c, c_c, d_c)^T$  is acquired by either singular value decomposition (Eq. (2.10)) or the direct approach (Eq. (2.12)), it can be then substituted into Eq. (2.7), Eq. (2.8), and Eq. (2.9) to obtain  $(z_c, y_c, r)$ .

Different levels of noise with a normal distribution and different sampling resolutions can be added to the sampled points on the circumference of a circle to test the algebraic and geometric circle fitting methods. A test example using dimensionless parameters is shown in Figure 2.34, where it is assumed that the circle’s radius is 10, the circle’s center is at  $(-2, 0)$ , the angular sampling resolution is  $3^\circ$ , and the noise is of 5% of the radius.

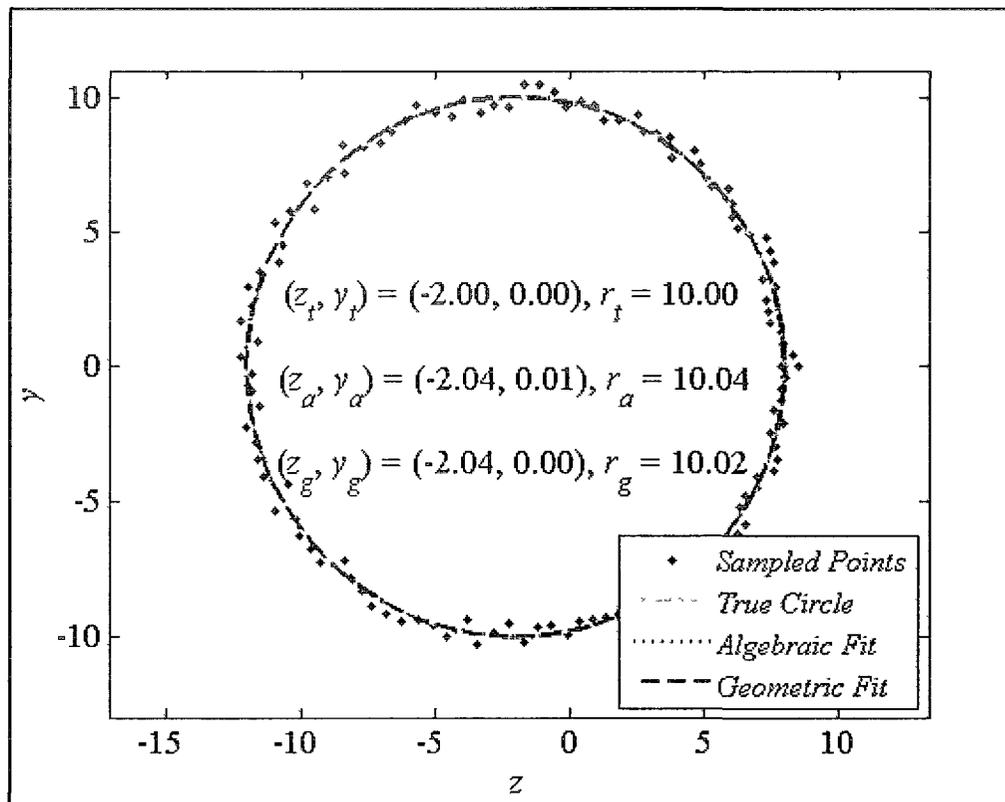


Figure 2.34 Circle Fitting Outputs

It is natural that the accuracy increases with increasing resolutions and decreasing noise levels. For a full  $360^\circ$  sampling coverage, the fitted circles using least-square algebraic and geometric circle fitting algorithms do not differ much. The  $3^\circ$  resolution and 5% noise are believed to be an upper threshold for the geometry data; the quality of data normally exceeds these specifications. In fact, it can be discovered that both circle fitting algorithms perform equally well because even with low resolution ( $10^\circ$ ) and a high level of noise (5%), a circle with a radius of about 10 and a center near  $(-2, 0)$  can be found. For a partial coverage, geometric fitting tends to give a more stable and close fit to the true circle. Therefore, in case vision occlusion exists, geometric fitting shall be used. However, because pipeline geometry scanning takes place inside the pipes, a full  $360^\circ$  sampling coverage is not hard to achieve. Once the true center is recovered, data of the regarding cross section shall be translated to get their new coordinates.

The general implicit equation of ellipse is:

$$F(z, y) = \mathbf{a} \cdot \mathbf{z} = a_e z^2 + b_e zy + c_e y^2 + d_e z + e_e y + f_e = 0,$$

where  $\mathbf{a} = (a_e, b_e, c_e, d_e, e_e, f_e)$  and  $\mathbf{z} = (z^2, zy, y^2, z, y, 1)$ . It is an over-determined problem if there are more than five data points, so

$$\sum_{i=1}^N (F(z_i, y_i))^2, i = 1, 2, \dots, N$$

has to be minimized in a least-square sense. To avoid the trivial solution  $\mathbf{a} = 0$ , different constraints on  $\mathbf{a}$  that fit different ellipses have been proposed by several authors (Bookstein [5], Fitzgibbon [10], and Gander [11]). It is desirable to choose a constraint that

is invariant under rigid body transformations, so that the same ellipse fit can be obtained regardless of where the point set is located in the plane.

Rewriting the implicit equation in another form:

$$F(\mathbf{p}) = \mathbf{p}^T \cdot \mathbf{A} \cdot \mathbf{p} + \mathbf{B} \cdot \mathbf{p} + f_e = 0,$$

where  $\mathbf{A} = \begin{pmatrix} a_e & b_e/2 \\ b_e/2 & c_e \end{pmatrix}$ ,  $\mathbf{B} = (d_e, e_e)$  and  $\mathbf{p} = (z, y)^T$ .

Any function of the eigenvalues of  $\mathbf{A}$  can reserve the invariance under translation and rotation. Two eigenvalues  $\lambda_1$  and  $\lambda_2$  can be calculate as:

$$\lambda_1 = \frac{1}{2} \left( \sqrt{a_e^2 + b_e^2 + c_e^2 - 2a_e c_e} + a_e + c_e \right),$$

$$\lambda_2 = \frac{1}{2} \left( -\sqrt{a_e^2 + b_e^2 + c_e^2 - 2a_e c_e} + a_e + c_e \right).$$

For the constraint, Bookstein [5] proposed  $\lambda_1^2 + \lambda_2^2 = 1$ , which is equivalent to  $a_e^2 + b_e^2/2 + c_e^2 = 1$ ; Fitzgibbon [10] used  $4\lambda_1\lambda_2 = 1$ , which is equivalent to  $4a_e c_e - b_e^2 = 1$ ; Gander [11] suggested  $\lambda_1 + \lambda_2 = 1$ , which is equivalent to  $a_e + c_e = 1$ .

When a geometric distance minimization is desired, the problem becomes nonlinear and iterative algorithms such as Gauss–Newton or Levenberg–Marquardt methods can be applied. Solutions obtained according to an algebraic measure can be used as a starting guess for the iterations.

## 2.9.2 DFT of Contours

The forward DFT decomposes a signal from the time or spatial domain into the frequency domain with lower frequency components outlining the coarser shape and higher frequency component representing the finer details. The inverse DFT reconstructs

the signal using the constituent frequencies. The input to the DFT is a finite set of real or complex numbers, making the DFT ideal for digital image processing. The DFT of  $N$  sample points can be computed efficiently using a fast Fourier transform (FFT) algorithm, which reduces the number of operations from  $O(N^2)$  to  $O(N \log N)$ .

A sequence of  $N$  spatial-domain complex numbers  $c_0, \dots, c_{N-1}$ , according to Zhang [29], can be transformed into a sequence of  $N$  frequency-domain complex numbers  $f_0, \dots, f_{N-1}$  by the DFT, or backwards by the inverse DFT, as in Eq. (2.13) and Eq. (2.14):

$$\mathbf{f}_k = \sum_{n=0}^{N-1} \mathbf{c}_n e^{-\frac{2\pi i}{N} kn}, k = 0, \dots, N-1, \quad (2.13)$$

$$\mathbf{c}_n = \sum_{k=0}^{N-1} \mathbf{f}_k e^{\frac{2\pi i}{N} kn}, n = 0, \dots, N-1, \quad (2.14)$$

where  $i$  is the imaginary unit and  $e^{-\frac{2\pi i}{N}}$  is a primitive  $N$ -th root of unity.

As shown in Figure 2.35, to calculate the DFT of a contour, the coordinate of each point shall be written as a complex number  $c_n = (z_n + iy_n)$ ,  $n = 0, \dots, N-1$ , and ordered in a clockwise or counterclockwise direction. The absolute value and ratio of the real and imaginary parts of  $f_k$  imply the power and phase angle of the  $k$ -th frequency element, respectively.

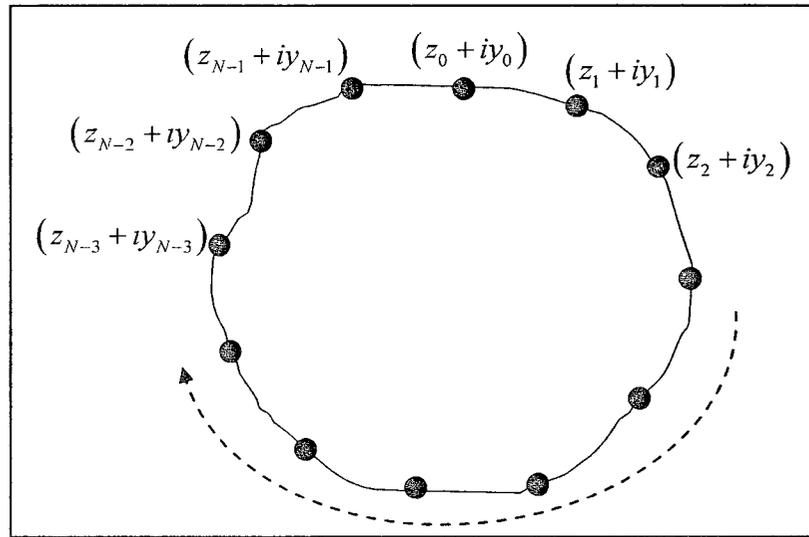


Figure 2.35 Complex Coordinates of Contour

Common variations of a contour  $c_n$  include affine transformations like translation, scaling and rotation, and a different order of the elements in  $c_n$ . These spatial variations of a contour are shown in Figure 2.36.

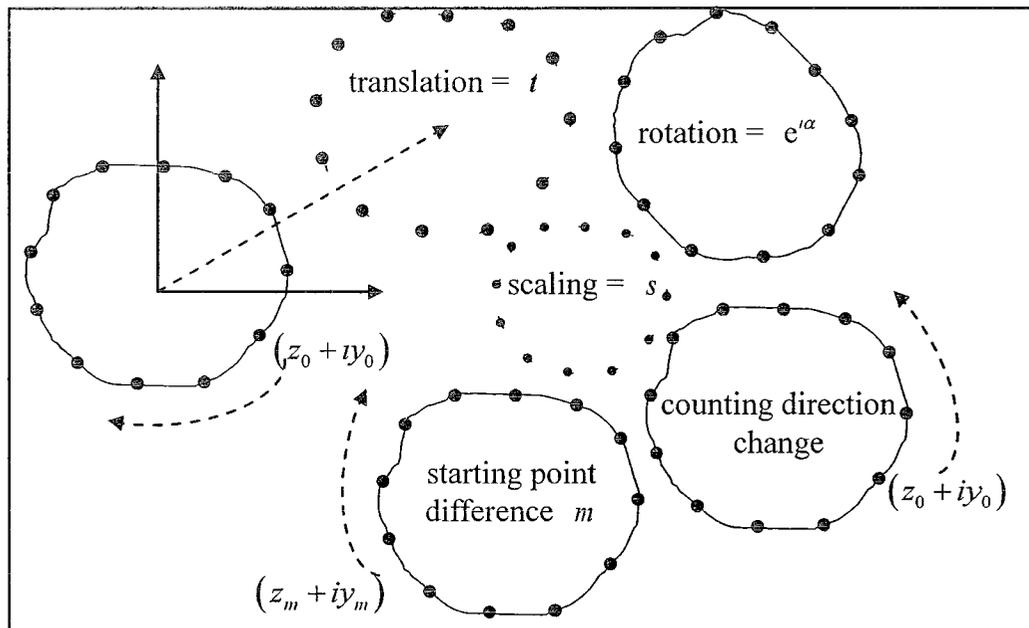


Figure 2.36 Common Spatial Variations of Contour

The corresponding transformations in the frequency domain are discussed next.

Recall that

$$\mathbf{f}_k = \sum_{n=0}^{N-1} \mathbf{c}_n e^{-\frac{2\pi i}{N} kn}, k = 0, \dots, N-1.$$

For a translation  $\mathbf{t} = z_l + iy_l$ ,  $\mathbf{c}'_n = \mathbf{c}_n + \mathbf{t}$ ,

$$\mathbf{g}_k = \sum_{n=0}^{N-1} \mathbf{c}'_n e^{-\frac{2\pi i}{N} kn} = \sum_{n=0}^{N-1} (\mathbf{c}_n + \mathbf{t}) e^{-\frac{2\pi i}{N} kn} = \sum_{n=0}^{N-1} \mathbf{c}_n e^{-\frac{2\pi i}{N} kn} + \sum_{n=0}^{N-1} \mathbf{t} e^{-\frac{2\pi i}{N} kn} = \mathbf{f}_k + \sum_{n=0}^{N-1} \mathbf{t} e^{-\frac{2\pi i}{N} kn},$$

for  $k = 0$ ,

$$\mathbf{g}_0 = \mathbf{f}_0 + \sum_{n=0}^{N-1} \mathbf{t} e^{-\frac{2\pi i}{N} 0n} = \mathbf{f}_0 + N\mathbf{t},$$

for  $k > 0$ ,

$$\mathbf{g}_k = \mathbf{f}_k + \sum_{n=0}^{N-1} \mathbf{t} e^{-\frac{2\pi i}{N} kn} = \mathbf{f}_k.$$

Therefore, a translation only add the first frequency element by  $N\mathbf{t}$ .

For a scaling  $s$ ,  $\mathbf{c}'_n = s\mathbf{c}_n$ ,

$$\mathbf{g}_k = \sum_{n=0}^{N-1} \mathbf{c}'_n e^{-\frac{2\pi i}{N} kn} = \sum_{n=0}^{N-1} s\mathbf{c}_n e^{-\frac{2\pi i}{N} kn} = s \sum_{n=0}^{N-1} \mathbf{c}_n e^{-\frac{2\pi i}{N} kn} = s\mathbf{f}_k.$$

For a rotation  $e^{i\alpha}$ ,  $\mathbf{c}'_n = e^{i\alpha} \mathbf{c}_n$ ,

$$\mathbf{g}_k = \sum_{n=0}^{N-1} \mathbf{c}'_n e^{-\frac{2\pi i}{N} kn} = \sum_{n=0}^{N-1} e^{i\alpha} \mathbf{c}_n e^{-\frac{2\pi i}{N} kn} = e^{i\alpha} \sum_{n=0}^{N-1} \mathbf{c}_n e^{-\frac{2\pi i}{N} kn} = e^{i\alpha} \mathbf{f}_k.$$

For a counting direction change,  $\mathbf{c}'_n = \mathbf{c}_{N-n}$ ,

$$\mathbf{g}_k = \sum_{n=0}^{N-1} \mathbf{c}_n' e^{-\frac{2\pi i}{N} kn} = \sum_{n=0}^{N-1} \mathbf{c}_{N-n} e^{-\frac{2\pi i}{N} kn} = \sum_{n=0}^{N-1} \mathbf{c}_n e^{-\frac{2\pi i}{N} k(N-n)} = \sum_{n=0}^{N-1} \mathbf{c}_n e^{\frac{2\pi i}{N} kn} = \overline{\sum_{n=0}^{N-1} \mathbf{c}_n e^{-\frac{2\pi i}{N} kn}} = \overline{\mathbf{f}_k(\mathbf{c}_n)}.$$

where  $\mathbf{f}_k(\overline{\mathbf{c}_n})$  is the DFT of  $\overline{\mathbf{c}_n}$ , a complex conjugate of  $\mathbf{c}_n$ .

For a starting point difference  $m$ ,

$$\mathbf{g}_k = \sum_{n=0}^{N-1} \mathbf{c}_n e^{-\frac{2\pi i}{N} k(n+m)} = \sum_{n=0}^{N-1} \mathbf{c}_n e^{-\frac{2\pi i}{N} kn} e^{-\frac{2\pi i}{N} km} = e^{-\frac{2\pi i}{N} km} \mathbf{f}_k.$$

Normalization is needed to unify contours with similar shape but different positions, sizes and orientations. For a translation, because only  $\mathbf{g}_0$ , the zero frequency element is affected, so  $\mathbf{g}_0 = 0$  can be used to filter out the translation. For a scaling, because  $\mathbf{g}_k = s\mathbf{f}_k$ , naturally  $\mathbf{g}_1 = s\mathbf{f}_1$ , so a division by the first non-zero frequency element can be used to filter out the scaling, i.e.,  $\mathbf{g}_k/\mathbf{g}_1 = \mathbf{f}_k/\mathbf{f}_1$ . Rotation, starting point difference and counting direction change can all be viewed as rotational difference; it can be observed that only the phase angle of  $\mathbf{f}_k$  is affected, so the absolute value of  $\mathbf{f}_k$  can be used to filter out the rotational difference.

### 2.9.3 Morphological Operations

Morphological operations, a subset of shape-based image processing techniques, use a small structuring element as a mask to an input image, yielding an output image of the same size. The structuring element is a matrix filled with binary values that can be of different shapes and sizes. The origin of the structuring element is placed onto the pixel being processed in the input image. The other pixels in the structuring element with values of 1's are called the neighborhood. The value of each pixel in the output image is based on a comparison of the corresponding pixel in the input image with its neighbors defined by

the structuring element. With various choices of the size and shape of the structuring element, the morphological operation can be customized to retain specific shapes in the input image. Figure 2.37 [32] shows the two basic morphological operations, dilation (left) and erosion (right), with a circular structuring element.

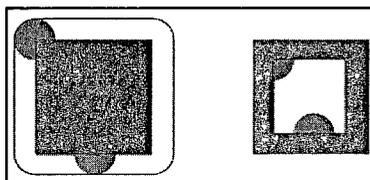


Figure 2.37 Morphological Dilation and Erosion

Dilation adds pixels to an object boundary in an input image. For the circular structuring element, the circle's center rolls along a square's boundary and dilates the smaller square into a larger one with round corner traced by circle's boundary. Similarly, erosion removes pixels to an object boundary. The circle's boundary rolls inside a square and erodes the larger square into a smaller one traced by the circle's center. Dilation and erosion are combined to achieve more image processing operations. For example, a morphological opening is an erosion followed by a dilation using the same structuring element. Morphological closing of an image is the reversed operation and consists of a dilation followed by an erosion using the same structuring element for both operations.

## 2.10 Chapter Summary

The following concludes the findings corresponding to the key points in the beginning of this chapter:

1. It is possible to detect cracks and joints dislocation in an underground pipeline.

Image processing techniques are usually task-specific and prior knowledge of

features of interest is beneficial. Pipes can deform smoothly into ellipses or discontinuously into hinges; liners can develop into wavy lobes. These three basic forms are mutually exclusive and can be used to model composite modes.

2. For a quantitative inspection of water and sewer pipelines, laser triangulation combined with SONAR shall be given prior considerations in terms of speed, accuracy and robustness.
3. Many commercial robots are available for different applications. Traditional bottom-wheeled robots are dominant in the water and sewer inspection practice.
4. Oblique measurements caused by robotic pitch and yaw shall not be used without knowledge of the robot–pipe relationship. Robotic roll alters the points order, therefore distorting the wireframe if the cross section is not circular.
5. INS sensors diverge after extended usage, causing the measured global layout of a pipeline untrue to the actual layout and difficulty in comparison at different times.
6. Geometry data can be acquired, processed and stored in various ways. 3-D data has incomparable advantages over 2-D data.
7. Intrinsic parameter of cameras used in laser triangulation shall be found. Five camera models are popular and able to identify most lenses.
8. Algebraic and geometric circle and ellipse fitting algorithms work equally well if the sampling density is large, or points cloud cover the full circumference, or the noise is low. Algebraic fitting is easy to implement and fast. While geometric fitting requires more computational time, the result is affine invariant and more robust. Morphological operations with structuring elements offer an alternative to popular edge detectors.

## CHAPTER 3

### MATHEMATICAL MODELING OF PIPE GEOMETRIES

One goal of this research is to mathematically model the three most frequent cross sections discussed in Section 2.1.4 and Section 2.8, namely ellipses, hinges, and lobes. Because of the material properties, loadings and constraints differences between pipes and liners, it makes sense to discuss the types of deformed cross sections in free-standing pipes and in encased liners separately. Rigid pipe geometry modeling is to be addressed in this chapter, while flexible liner geometry modeling is to be studied in Section 7.1 and Section 7.3 .

#### 3.1 Elliptical Cross Section (Mode 1)

Rigid pipes exhibit strong resistance against external groundwater pressure and dead and live loads from above. During the process of transportation and installation, a rigid pipe can show a certain level of conformability to the overall loadings, and its cross sections can deform into ellipses without exceeding the strength of the material. The equation of an ellipse is expressed as Eq. (3.1):

$$\frac{z^2}{r_1^2} + \frac{y^2}{r_2^2} = 1, \quad (3.1)$$

where  $r_1$  is the semi-major axis length and  $r_2$  is the semi-minor axis length and  $yz$ -plane denotes the cross sectional plane. In most cases, a pipe is compressed from top (crown) to bottom (invert), so the major axis ( $z$ -axis) is almost horizontal and the minor axis ( $y$ -axis) is almost vertical relative to the earth, as shown in Figure 0.1. Such an elliptical deformation level is dependent on the rigidity and buckling strength of the pipe.

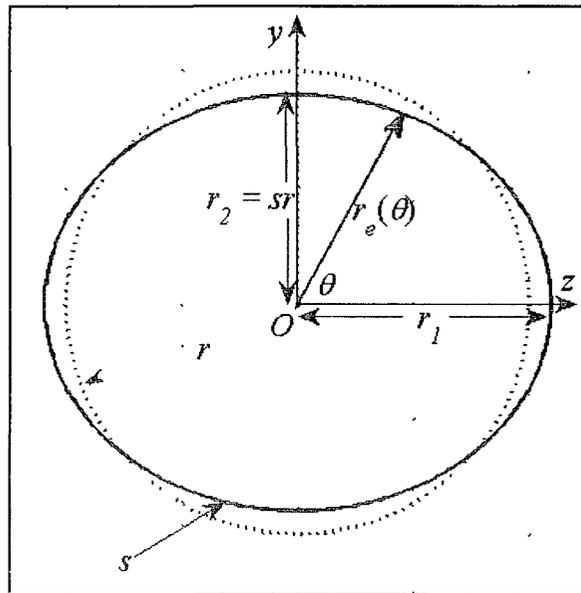


Figure 0.1 Cross Sectional View of Elliptical Deformation (Mode 1)

The circumferential elongation in rigid pipes is usually negligible so that the perimeter can be assumed to remain constant. For the elliptical cross section, the semi-major axis length and the semi-minor axis length are not independent of each other.

The perimeter of a circle is expressed as  $p = 2\pi r$ , where  $p$  and  $r$  are the perimeter and radius of the circle, respectively. Now the circular cross section deforms into an ellipse. According to Ramanujan [23], the perimeter of an ellipse can be approximated by:

$$p = \pi \left[ 3(r_1 + r_2) - \sqrt{(r_1 + 3r_2)(3r_1 + r_2)} \right],$$

where  $r_1$  and  $r_2$  are the semi-major and the semi-minor axis lengths, respectively.

Therefore,

$$3(r_1 + r_2) - \sqrt{(r_1 + 3r_2)(3r_1 + r_2)} = 2r.$$

Assuming  $r_2 = sr$ , where  $s \in (0, 1]$ , i.e., the semi-minor axis shall be no larger than the circle's radius. After substitution,

$$3(r_1 + sr) - \sqrt{(r_1 + 3sr)(3r_1 + sr)} = 2r.$$

Rearranging the square root to the left side of the equation and squaring both sides,

$$3r_1^2 + 10r_1sr + 3s^2r^2 = 9r_1^2 + 6(3s - 2)r_1r + (3s - 2)^2r^2.$$

Treating  $r_1$  as the variable and rearranging in the order of the power of  $r_1$ ,

$$6r_1^2 + (8sr - 12r)r_1 + (6s^2r^2 - 12sr^2 + 4r^2) = 0.$$

Solving the equation,

$$r_1 = \frac{-8sr + 12r \pm \sqrt{\Delta}}{12},$$

where  $\Delta = (8sr - 12r)^2 - 24(6s^2r^2 - 12sr^2 + 4r^2) = 16r^2(-5s^2 + 6s + 3)$ .

Because  $s \in (0, 1]$ , it is obvious that  $\Delta > 0$ , proving that the solutions are real numbers. There are two solutions for  $r_1$ . Assuming the minus sign provides a valid solution, because  $r_1 \geq r$ , it can be derived that

$$r_1 = \frac{-8sr + 12r - \sqrt{\Delta}}{12} \geq r,$$

$$-8sr - \sqrt{\Delta} \geq 0,$$

which is contradictory because the left side is always a negative value. Therefore, the minus sign is abandoned and the plus sign is retained:

$$r_1 = \frac{-8sr + 12r + \sqrt{16r^2(-5s^2 + 6s + 3)}}{12} = \frac{r}{3} \left( -2s + 3 + \sqrt{-5s^2 + 6s + 3} \right).$$

In conclusion, the constraint can be expressed as Eq. (3.2) and Eq. (3.3):

$$r_1 = \frac{\left( -2s + 3 + \sqrt{-5s^2 + 6s + 3} \right) r}{3}, \quad (3.2)$$

$$r_2 = sr, \quad (3.3)$$

where  $s \in (0, 1]$ .

### 3.2 Circular Hinged Cross Section (Mode 2)

As the service age builds up, the soil–pipe interaction usually results in stress concentration, mostly near the top (crown) and the bottom (invert). These two discontinuities along the vertical direction essentially divide the pipe into two halves, with each one representing a vertical arch subject to compressions at both ends. The very left and right spots (spring lines) suffer from the largest bending moment, where two new hinged lines occur. For very rigid pipes, it is most likely the circular shape directly develops these discontinuities before going through any significant elliptical change. As a result, four hinges are visible near the crown, invert and spring lines, as shown in Figure 0.2.

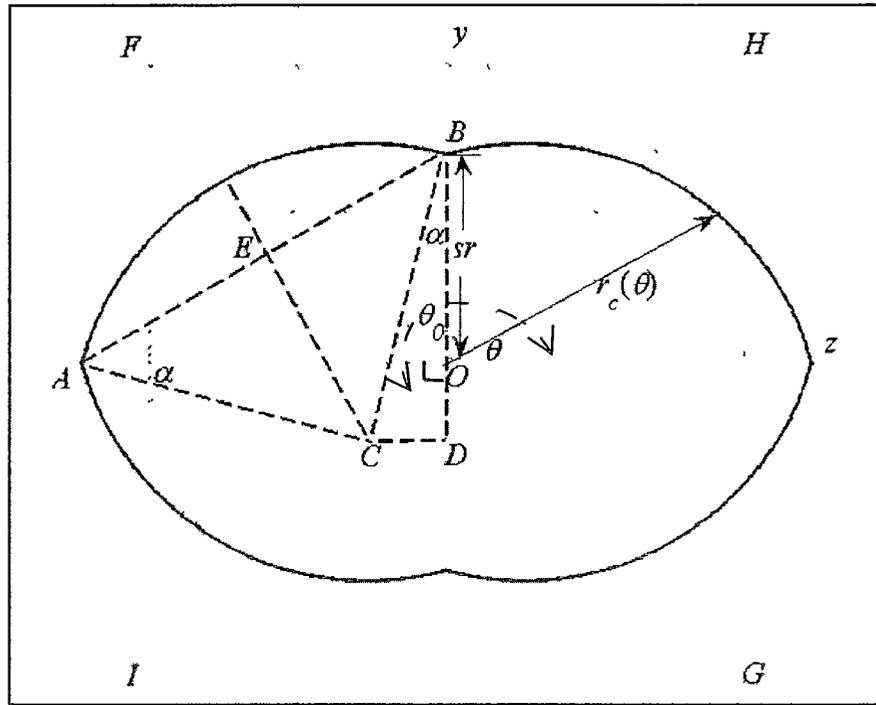


Figure 0.2 Cross Sectional View of Circular Hinged Deformation (Mode 2)

In order to plot the hinged arcs, center  $C$  and the half hinged angle  $\alpha$  are needed.

Assuming Eq. (3.4):

$$OB = sr, \quad (3.4)$$

where  $s \in (0,1]$ , i.e., the semi-minor axis shall be no larger than the circle's radius. The parameter  $s$  indicates how much the cross section has deformed; the smaller  $s$  is, the more ovality the cross section exhibits. Because  $AB$  is the chord of the arc and does not change after deformation, so the semi-major axis  $OA$  can be found as Eq. (3.5):

$$OA = \sqrt{AB^2 - OB^2} = \sqrt{2r^2 - OB^2} = \sqrt{2 - s^2}r. \quad (3.5)$$

The coordinate of the middle point  $E$  of  $AB$  is:

$$(E_z, E_y) = \left( -\frac{OA}{2}, \frac{OB}{2} \right) = \left( -\frac{\sqrt{2-s^2}r}{2}, \frac{sr}{2} \right).$$

The slope of line  $AB$  is:

$$\frac{OB}{OA} = \frac{s}{\sqrt{2-s^2}}.$$

So the slope of the perpendicular line  $EC$  is:

$$-\frac{OA}{OB} = -\frac{\sqrt{2-s^2}}{s}.$$

Normalizing the slope of line  $EC$  into a vector form:

$$(v_z, v_y) = \left( \frac{s}{\sqrt{2}}, -\sqrt{1-\frac{s^2}{2}} \right),$$

where  $v_z$  and  $v_y$  are the vector component along  $z$ -axis and  $y$ -axis, respectively.

Then the normal line through the middle point  $E$  can be expressed parametrically as:

$$z = E_z + v_z t = -\frac{\sqrt{2-s^2}r}{2} + \frac{s}{\sqrt{2}} t,$$

$$y = E_y + v_y t = \frac{sr}{2} - \sqrt{1-\frac{s^2}{2}} t,$$

where  $t$  is a dummy variable, indicating the length of  $EC$ . In the original circle,

$t = |EC| = r/\sqrt{2}$ . By substituting  $t$ , Eq. (3.6) and Eq. (3.7) can be obtained as:

$$CD = z_c = -\frac{\sqrt{2-s^2}r}{2} + \frac{s}{\sqrt{2}} \frac{\sqrt{2}}{2} r = -\frac{r}{2} \left( -s + \sqrt{2-s^2} \right), \quad (3.6)$$

$$OD = y_c = \frac{sr}{2} - \sqrt{1 - \frac{s^2}{2}} \frac{\sqrt{2}}{2} r = -\frac{r}{2} \left( -s + \sqrt{2 - s^2} \right). \quad (3.7)$$

Because  $s \in (0, 1]$ , then  $-s + \sqrt{2 - s^2} \geq 0$ . Therefore, the location of the new center  $C$  of the arc  $AB$  is in the third quadrant.

For the four hinges deformation, centers of the four hinged arcs move along two lines  $FG$  and  $HI$  with slope  $-1$  and  $1$ , respectively, as shown in Figure 0.2 Cross Sectional View of Circular Hinged Deformation. The angle of rotation of the arcs with respect to the original circle can be found as Eq. (3.8):

$$\tan \alpha = \frac{|CD|}{|BD|} = \frac{|CD|}{|OB| + |OD|} = \frac{\frac{r}{2} \left( -s + \sqrt{2 - s^2} \right)}{sr + \frac{r}{2} \left( -s + \sqrt{2 - s^2} \right)} = \frac{-s + \sqrt{2 - s^2}}{s + \sqrt{2 - s^2}},$$

$$\alpha = \tan^{-1} \left( \frac{-s + \sqrt{2 - s^2}}{s + \sqrt{2 - s^2}} \right). \quad (3.8)$$

### 3.3 Composite Cross Sections

Sometimes a single deformation mode is not adequate to account for the deformed cross section; therefore, it is necessary to use the combined modes to achieve a better description of the acquired cross sectional geometry. In the case of a relatively flexible pipe, partial circumference may develop a discontinuity due to cracks-induced stress concentration while the rest is still in the phase of elliptical deformation. Three modes of composite cross sections are to be discussed next.

### 3.3.1 Composite Mode 3

Three hinges occur at the crown and along the spring lines; the bottom area is subject to an elliptical deformation. Therefore, the top portion comprises of two circular arcs and the bottom portion is an elliptical arc. Combining the plotting techniques in Section 3.1 and Section 3.2, Figure 0.3 shows such a mode.

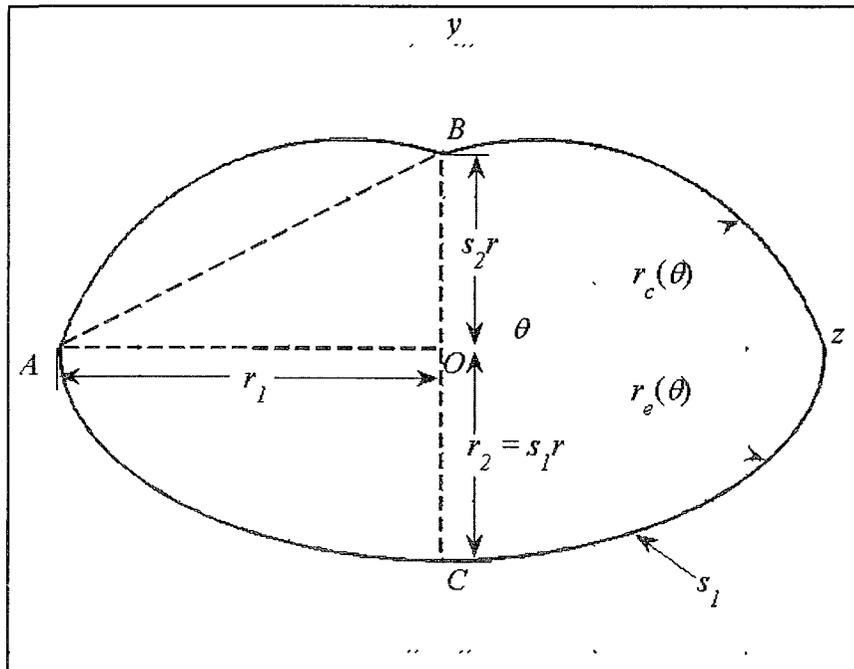


Figure 0.3 Composite Cross Section (Mode 3)

A compatibility condition shall be satisfied for the composite cross section to be geometrically continuous: the semi-major axis  $r_1$  induced by the bottom elliptical half shall be equal to  $OA$  induced by the top hinged half. Assigning different deformation indicator  $s_1$  and  $s_2$ , and reproducing Eq. (3.2), Eq. (3.3), Eq. (3.4) and Eq. (3.5) here:

$$r_1 = \left( -2s_{02} + 3 + \sqrt{-5s_{02}^2 + 6s_{02} + 3} \right) r / 3, \quad r_2 = s_1 r, \quad OA = \sqrt{2 - s_2^2} r, \quad OB = s_2 r, \text{ then Eq. (3.9)}$$

can be obtained:

$$s_2 = \sqrt{2 - \frac{(-2s_1 + 3 + \sqrt{-5s_1^2 + 6s_1 + 3})^2}{9}}. \quad (3.9)$$

The lower limit of  $s_1$  can be determined by noticing that right side of Eq. (3.9) positive. It can be obtained that  $-2s_1 + 3 + \sqrt{-5s_1^2 + 6s_1 + 3} \leq 3\sqrt{2}$ , which is plotted in Figure 0.4; two values that make the equal sign valid are found as  $s_1 \approx -0.349049$  and  $s_1 \approx 0.463431$ . The negative value does not have a physical meaning, so only the positive value is retained. A plot of Eq. (3.9) reveals the relationship between  $s_1$  and  $s_2$ , shown in Figure 0.5.

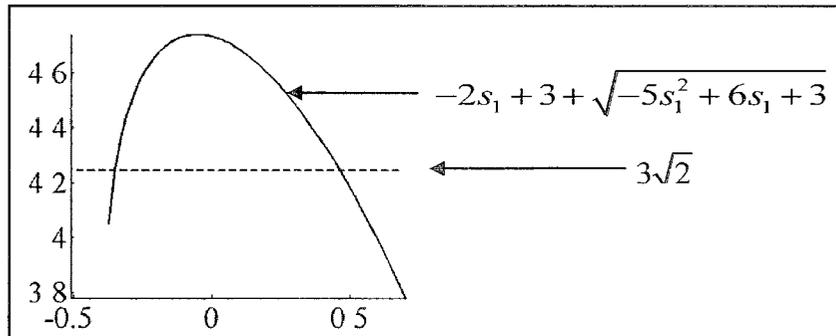


Figure 0.4 Determination of Lower Limit for Elliptical Deformation Indicator

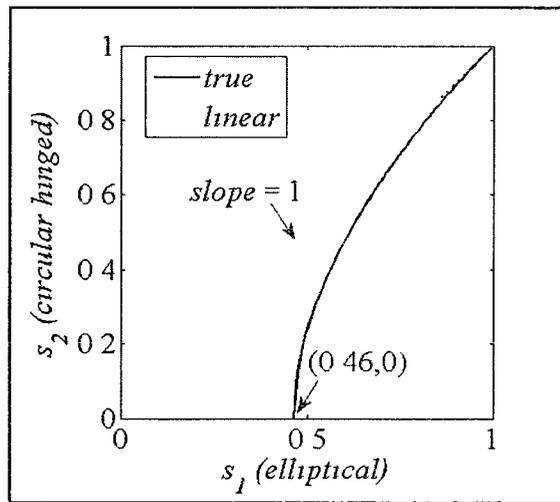


Figure 0.5 Relationship between Top and Bottom Deformation (Mode 3)



$\angle OBE = \angle CAE$ . Because  $BE = OB/\cos \alpha$ , so  $CE = BC - BE = BC - OB/\cos \alpha$ . Also, because  $\tan \alpha = CE/AC = (BC - OB/\cos \alpha)/AC$ , so  $AC \sin \alpha - BC \cos \alpha = -OB$  and

$$\sqrt{AC^2 + BC^2} \sin(\alpha + \alpha_o) = -OB,$$

where  $\alpha_o = -\tan^{-1}(BC/AC)$ . So the center  $C$  and the rotation angle  $\alpha$  and can be found as Eq. (3.10), Eq. (3.11), and Eq. (3.12):

$$z_c = CD = BC \sin \alpha, \quad (3.10)$$

$$y_c = -OD = OB - BD = OB - BC \cos \alpha, \quad (3.11)$$

$$\alpha = -\sin^{-1} \frac{OB}{\sqrt{AC^2 + BC^2}} + \tan^{-1} \frac{BC}{AC}. \quad (3.12)$$

Retaining the same symbol nominations as in Section 3.3.1 and assuming that the initial deformation severity of arc  $AB$  is  $s_{02}$ , the only difference is that the chord  $AB$  is no longer  $\sqrt{2}r$ , but the chord in the initial deformed ellipse, whose semi axes are  $AC = (-2s_{02} + 3 + \sqrt{-5s_{02}^2 + 6s_{02} + 3})r/3$  and  $BC = s_{02}r$ . Therefore, chord  $AB$  can be calculated as Eq. (3.13):

$$AB^2 = \frac{\left[(-2s_{02} + 3 + \sqrt{-5s_{02}^2 + 6s_{02} + 3})r\right]^2}{9} + (s_{02}r)^2. \quad (3.13)$$

The rest follows the same procedures as in the development of Eq. (3.9). By substituting  $OB = s_2r$ ,  $OA = (-2s_1 + 3 + \sqrt{-5s_1^2 + 6s_1 + 3})r/3$  and Eq. (3.13) into  $OB = \sqrt{AB^2 - OA^2}$ , and then taking out the common factor  $r$  on both sides, Eq. (3.14) can be derived as:

$$s_2^2 = s_{02}^2 + \frac{\left(-2s_{02} + 3 + \sqrt{-5s_{02}^2 + 6s_{02} + 3}\right)^2 - \left(-2s_1 + 3 + \sqrt{-5s_1^2 + 6s_1 + 3}\right)^2}{9}. \quad (3.14)$$

Different initial deformation  $s_{02}$  yields different relationships between  $s_1$  and  $s_2$ , as shown in Figure 0.7. Numerical methods are applied to determine the lower limit for  $s_1$  for each given  $s_{02} = (1, 0.9, 0.8, 0.7, 0.6, 0.5, 0.4, 0.3, 0.2, 0.1)$  in Eq. (3.14); the following values can be obtained: (0.463, 0.457, 0.440, 0.414, 0.378, 0.334, 0.283, 0.224, 0.158, 0.085). It can be seen that the initial  $s_{02}$  sets up the upper limits for  $s_1$  and  $s_2$ ; the relationship between  $s_1$  and  $s_2$ , however, are similar, regardless of  $s_{02}$ .

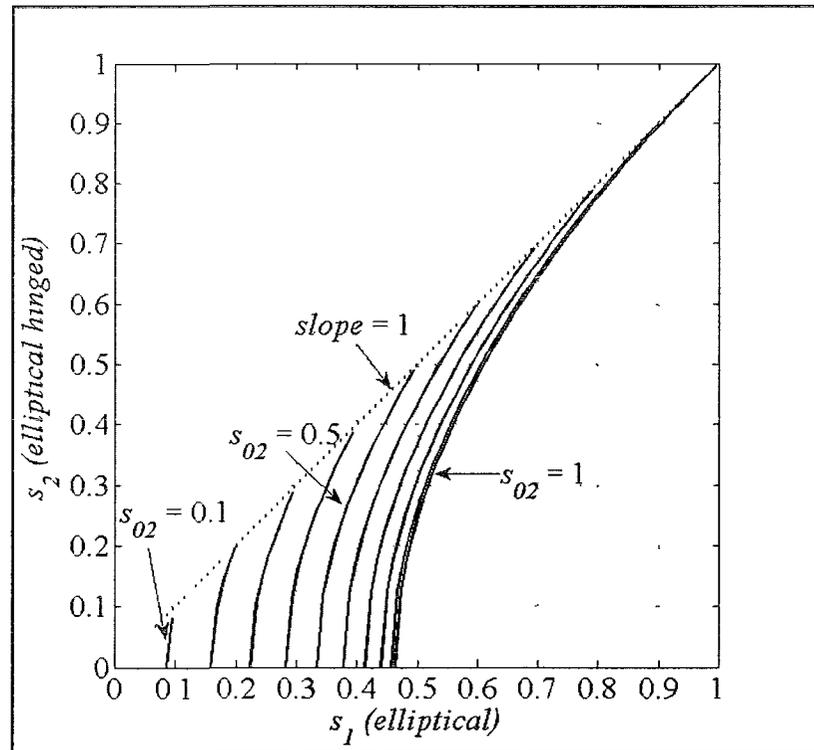


Figure 0.7 Generalized Relationship between Top and Bottom Deformation (Mode 4)

### 3.3.3 Composite Mode 5

An even more generalized mode follows Section 3.3.2. The top hinge keeps concaving in with the ovality indicator  $s_{02}$ ; the bottom portion develops a hinge after reaching a different level of ovality  $s_{01} \leq s_{02}$ , as shown in Figure 0.8. Therefore, the initial deformation severities for the top and bottom arcs are  $s_{02}$  and  $s_{01}$ , respectively.

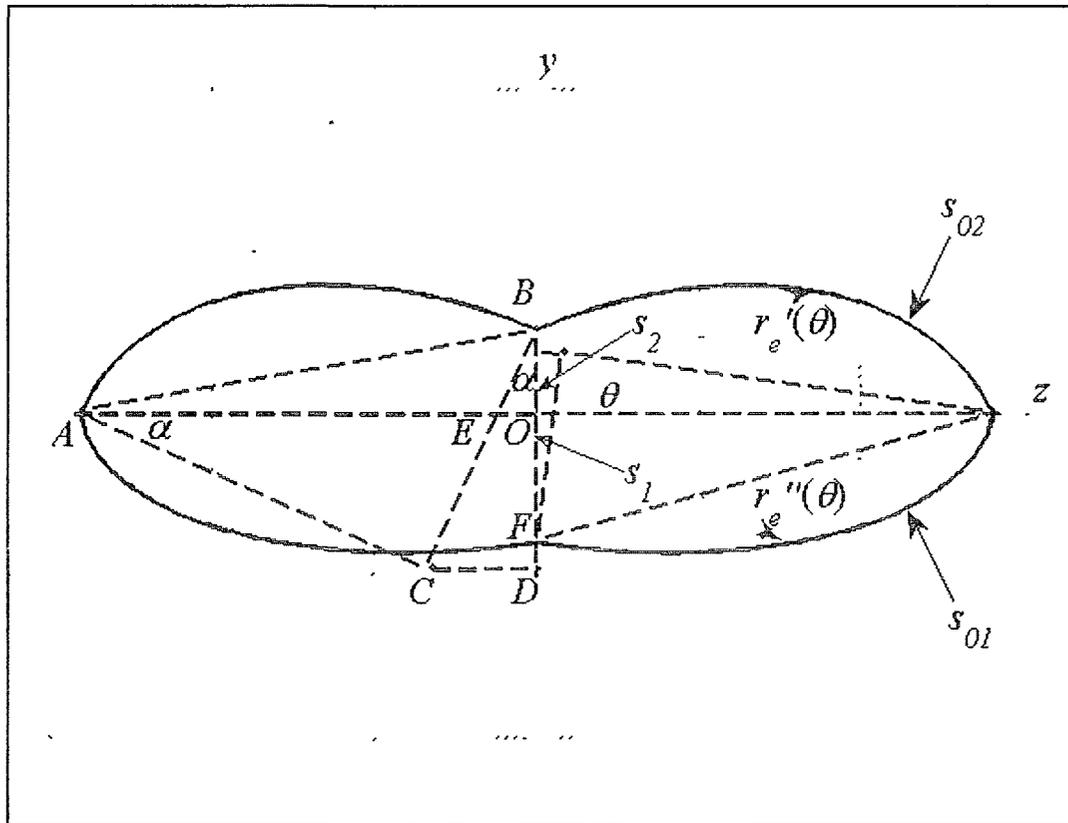


Figure 0.8 Composite Cross Section (Mode 5)

The plotting of such a mode is based on the research groundwork developed for composite mode 4; for the continuity requirement, Eq. (3.15) can be derived:

$$\left[ \frac{(-2s_{02} + 3 + \sqrt{-5s_{02}^2 + 6s_{02} + 3})r}{3} \right]^2 + s_{02}^2 - s_2^2 = \left[ \frac{(-2s_{01} + 3 + \sqrt{-5s_{01}^2 + 6s_{01} + 3})r}{3} \right]^2 + s_{01}^2 - s_1^2,$$

$$s_2^2 = s_1^2 + s_{02}^2 - s_{01}^2 + \frac{(-2s_{02} + 3 + \sqrt{-5s_{02}^2 + 6s_{02} + 3})^2 - (-2s_{01} + 3 + \sqrt{-5s_{01}^2 + 6s_{01} + 3})^2}{9}. \quad (3.15)$$

Because the bottom hinge develops after having a ovality  $s_{01}$ , the upper limit for  $s_1$  is  $s_{01}$ ;

the lower limit for  $s_1$  can be determined by noticing the right side of Eq. (3.15) positive:

$$s_1^2 \geq s_{01}^2 - s_{02}^2 + \frac{(-2s_{01} + 3 + \sqrt{-5s_{01}^2 + 6s_{01} + 3})^2 - (-2s_{02} + 3 + \sqrt{-5s_{02}^2 + 6s_{02} + 3})^2}{9}.$$

Two examples are shown in Figure 0.10 ( $s_{01} = 0.8$ ) and Figure 0.10 ( $s_{01} = 0.5$ ).

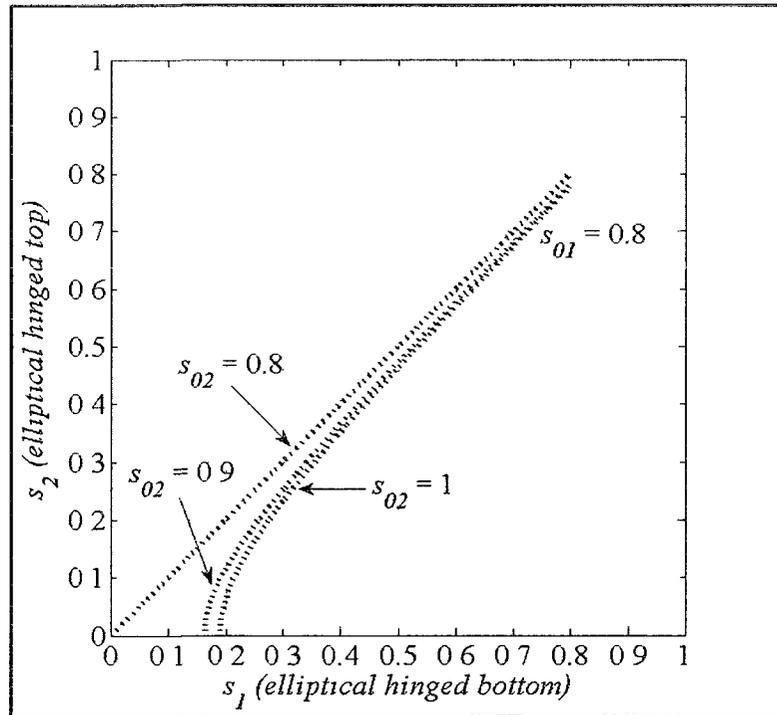


Figure 0.9 General Relationship of Composite Cross Section (Mode 5, Case 1)

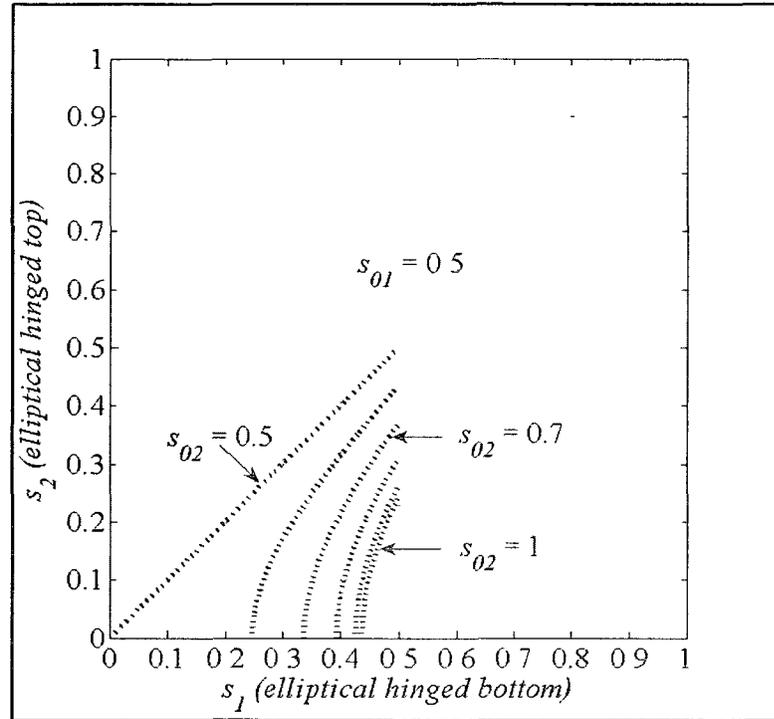


Figure 0.10 Generalized Relationship of Composite Cross Section (Mode 5, Case 2)

### 3.3.4 Summary of Composite Modes

To wrap up the results from this section, Figure 0.7, Figure 0.9, and Figure 0.10 are combined to generate Figure 0.11 ( $s_{01} = 0.8$ ) and Figure 0.12 ( $s_{01} = 0.5$ ). A natural deformation pathway can be seen in both;  $s_{02}$  determines where the green line branches from the red, indicating the emergence of the top hinge, and  $s_{01}$  determines where the blue line branches from the green, indicating the emergence of the bottom hinge.

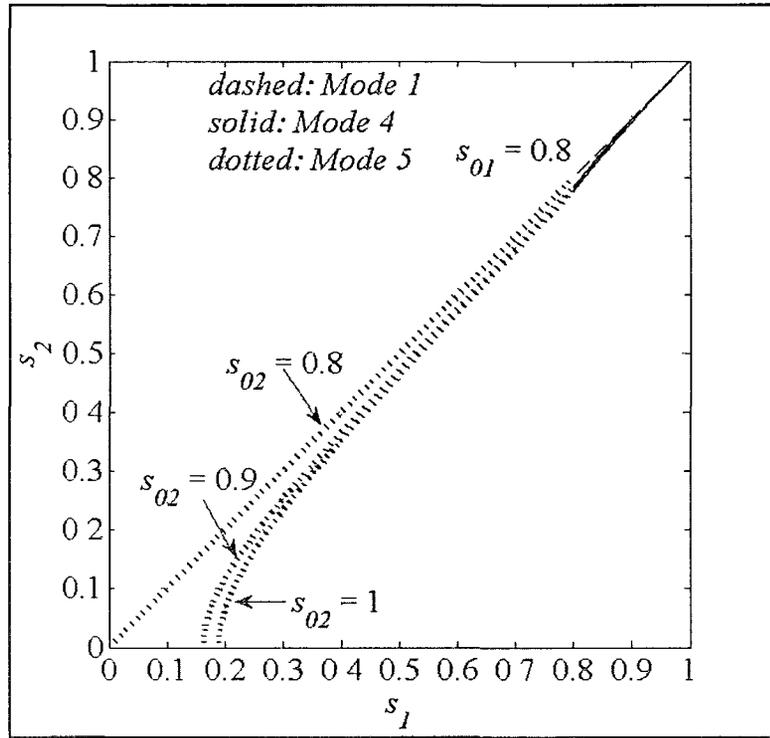


Figure 0.11 Deformation Pathway of Composite Cross Section (Case 1)

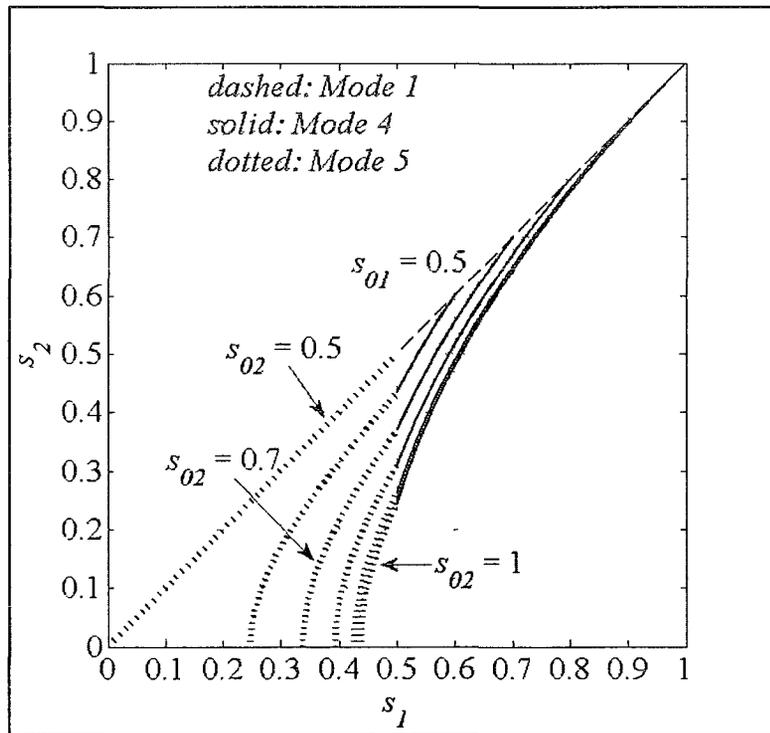


Figure 0.12 Deformation Pathway of Composite Cross Section (Case 2)

### 3.4 Geometry Modeling Extensions

This section discusses some applications of the parametric modeling of the five cross sections, including area loss, 3-D pipe segment generation, and accurate ovality calculation.

#### 3.4.1 Area Loss

Deformed cross section reduced the flow capacity of a pipe. The area for a circle and an ellipse is  $\pi r^2$  and  $\pi r_1 r_2$ , respectively, as shown in Figure 0.1. Therefore, the area loss for a whole elliptical deformation is  $\pi r^2 - \pi r_1 r_2$ ; the area loss contribution of a bottom elliptical deformation in the composite mode 3 and mode 4 is  $(\pi r^2 - \pi r_1 r_2)/2$ .

To calculate the area loss introduced by a hinge, a top left portion is shown in Figure 0.13, the segment surrounded by line  $AC$ , line  $BC$  and arc  $AB$  represent the original area and the segment surrounded by line  $AO$ , line  $BO$  and arc  $AB$  represent the reduced area. With common areas surrounded by line  $AE$ , line  $BE$  and arc  $AB$ , the area loss is the area difference between  $\Delta CAE = CA \cdot CE/2 = (CA^2 \tan \alpha)/2$  and  $\Delta OBE = OB \cdot OE/2 = (OB^2 \tan \alpha)/2$ , where  $CA$ ,  $OB$  and  $\alpha$  are discussed for each hinged mode in this chapter.

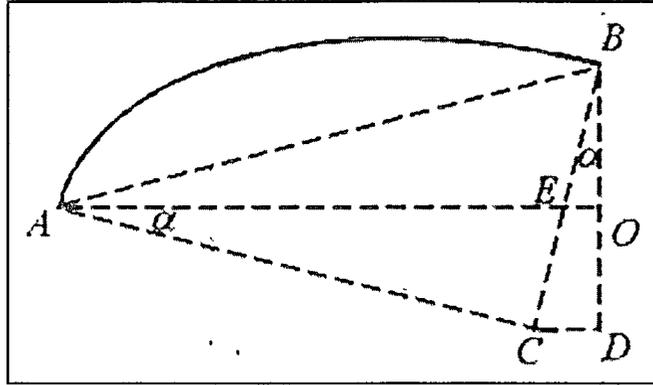


Figure 0.13 Reduced Area of Hinged Portion

Area losses associated with each mode can be calculated with the results developed in this chapter. An example ( $s_{02} = 0.8$ ,  $s_{01} = 0.6$ ) is shown in Figure 0.14. For small deformation severities ( $s_1 \geq 0.9$ ), the area loss is less than 3% for all modes.

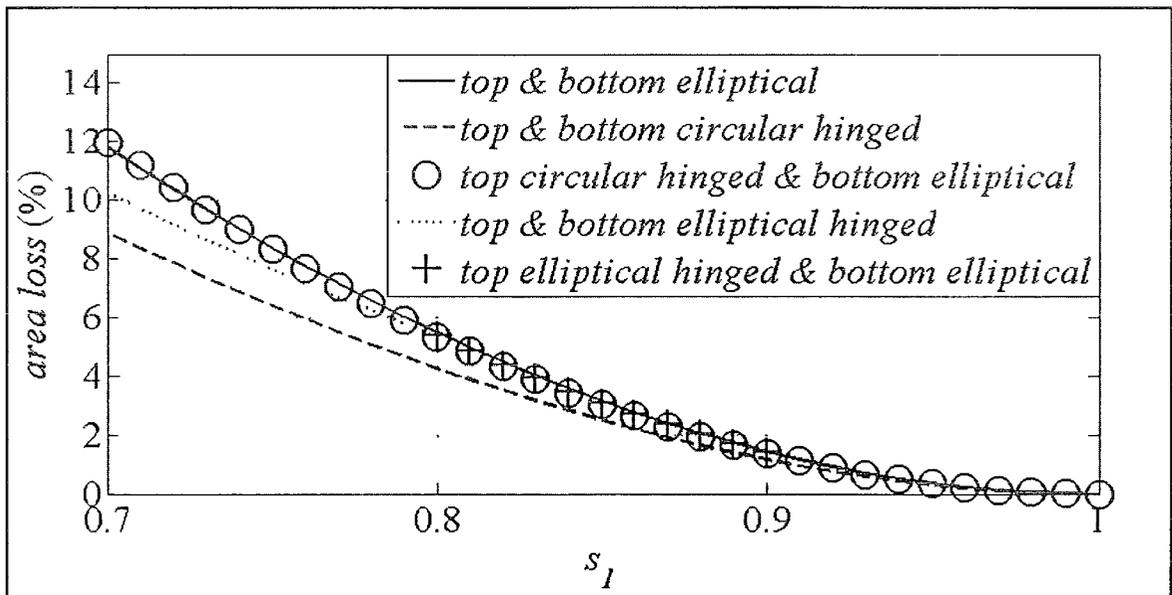


Figure 0.14 Area Losses for Different Cross Section Modes

### 3.4.2 3-D Pipe Segment

3-D RGB plots can be made easily by controlling  $s_1$  along the pipe length. Constant, linearly varying and sinusoidally varying  $s_1$ 's are used from left to right columns in Figure 0.15 to show such effects. From top to bottom rows are: “whole elliptical”, “whole circular hinged”, “top: circular hinged and bottom: elliptical”, “top: elliptical hinged and bottom: elliptical”, and “top: elliptical hinged and bottom: another level of elliptical hinged” modes, in the same order as discussed in Section 3.1, Section 3.2 and Section 3.3.

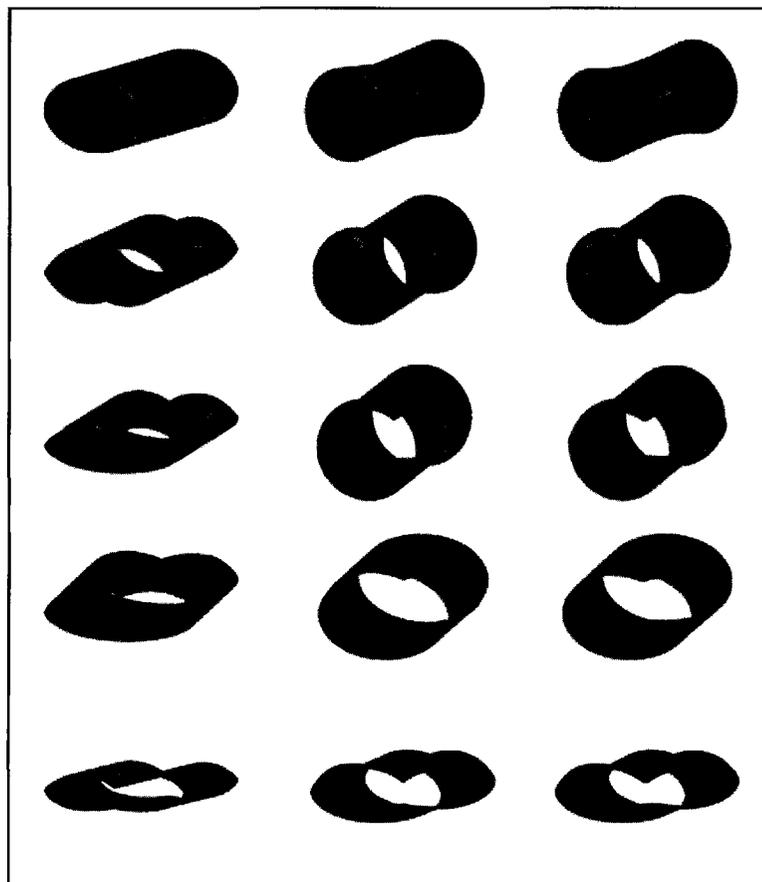


Figure 0.15 Three Dimensional RGB Plots of Cross Sectional Modes

### 3.4.3 Accurate Ovality Calculator

Recall Eq. (1.3) used in ASTM F1216 for the calculation of percent ovality, Currently, the pipeline industry often uses Eq. (3.16) to estimate the MeanInsideDiameter:

$$\text{MeanInsideDiameter} = \frac{r_1 + r_2}{2}. \quad (3.16)$$

Such an estimator is biased when the out-of-roundness level is high. In Figure 0.1, the polar expression for an ellipse is shown in Eq. (3.17):

$$r_e(\theta) = \frac{r_1 r_2}{\sqrt{(r_1 \sin \theta)^2 + (r_2 \cos \theta)^2}}. \quad (3.17)$$

The indefinite integration of the Eq. (3.17) is shown in Eq. (3.18):

$$\int r_e(\theta) d\theta = \frac{r_1}{2\pi} F\left(\theta \middle| 1 - \frac{r_2^2}{r_1^2}\right) + \text{constant}, \quad (3.18)$$

where  $F\left(\theta \middle| 1 - \frac{r_2^2}{r_1^2}\right)$  is the elliptical integral of the first kind and only numerical solutions are available. To calculate MeanInsideDiameter, take the definite integral in Eq. (3.18) from 0 to  $2\pi$  and then divide the definite integral by  $2\pi$  as Eq. (3.19):

$$\text{MeanInsideDiameter} = \frac{\int_0^{2\pi} r_e(\theta) d\theta}{2\pi}. \quad (3.19)$$

Figure 0.16 depicts the differences of percent ovality  $q$  (Eq. (1.3)) and ovality reduction factor  $C$  (Eq. (1.2)) between using Eq. (3.16) and using Eq. (3.19).

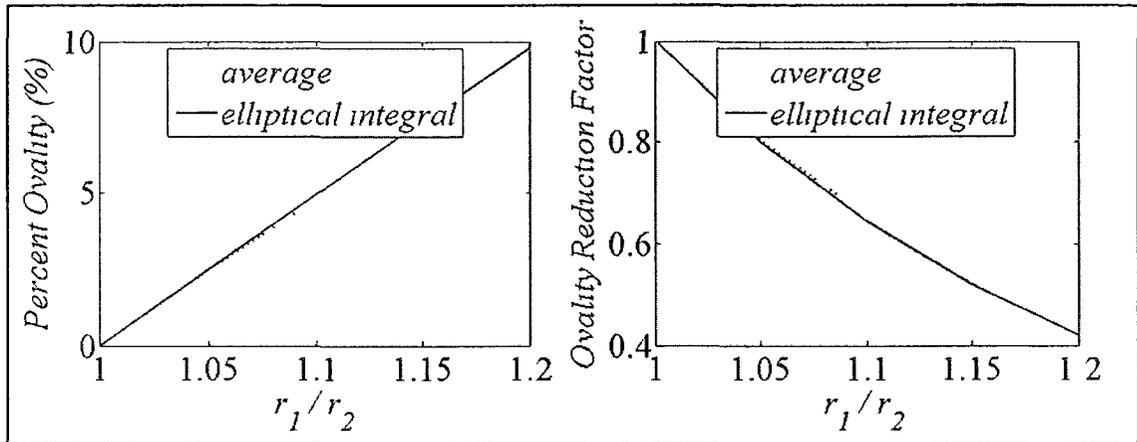


Figure 0.16 Mean Diameter, Percent Ovality and Ovality Reduction Factor

To discover the influence of the improved ovality calculator on the critical pressure estimation, the following values are used by Cai [6] as a test case:  $K = 7$ ,  $E_L = 72500$  psi,  $\nu = 0.35$ ,  $D_m = 8$  in,  $t = 0.2462$  in and  $N = 2$ , so that

$$SDR - 1 = \frac{D_o}{t} - 1 = \frac{D_o - t}{t} = \frac{D_m}{t} = 32.49,$$

where  $D_m$  is the mean diameters of the liner. Figure 0.17 is generated by substituting these values into Eq. (1.1).

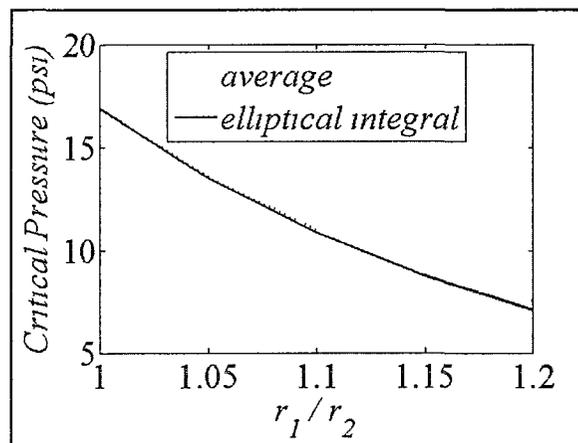


Figure 0.17 Accurate Calculation of Critical Pressure using Elliptical Integral

As another test case, it is of interest to discover the influence of the improved ovality calculator on the liner thickness design. Rewriting Eq. (1.1) into:

$$P_o = \frac{2KE_L}{1-\nu^2} \frac{1}{\left(\frac{D_m}{t}\right)^3} \frac{C}{N},$$

$$t = \left[ \frac{P_o (1-\nu^2) N}{2KE_L C} \right]^{\frac{1}{3}} D_m.$$

Then the same set of values and  $P_o = 20$  psi are used to generate Figure 0.18.

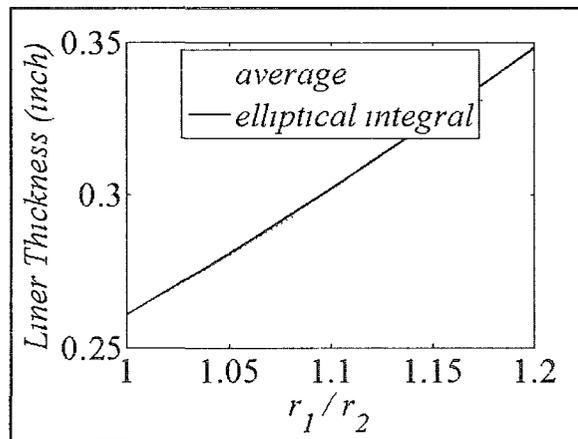


Figure 0.18 Accurate Calculation of Liner Thickness using Elliptical Integral

Table 0.1 quantifies the two test cases in this section.  $(r_1/r_2)$  ranges from 1.00 to 1.20 to control the ovality  $q$  under 10%. The last two columns indicate the improvement of using elliptical integral becomes more significant as the deformation level  $(r_1/r_2)$  gets more severe.

Table 0.1 Influences of Integral Methods on Mode 1

$r_1 / r_2$	using average				using integral				difference	
	q	C	P (psi)	t (inch)	q	C	P (psi)	t (inch)	P (psi)	t (inch)
1.00	0.00%	1.00	16.86	0.2606	0.00%	1.00	16.86	0.2606	0.00%	0.00%
1.05	2.44%	0.80	13.55	0.2803	2.49%	0.80	13.49	0.2807	0.41%	0.14%
1.10	4.76%	0.65	11.02	0.3004	4.94%	0.64	10.84	0.3019	1.58%	0.53%
1.15	6.98%	0.54	9.05	0.3206	7.37%	0.52	8.75	0.3244	3.41%	1.16%
1.20	9.09%	0.45	7.51	0.3412	9.77%	0.42	7.08	0.3481	5.82%	2.02%

The accurate ovality can also be calculated for the other four modes in this chapter. Note that once hinges emerge, the center  $C$  of the circular and elliptical arcs are not aligned with the origin  $O$ . In order to calculate MeanInsideDiameter, the polar expressions for circles and ellipses are needed. The polar coordinates for an arbitrary circle is expressed in Eq. (3.20):

$$r_c(\theta) = r_0 \cos(\theta - \theta_0) + \sqrt{r^2 - r_0^2 \sin^2(\theta - \theta_0)}, \quad (3.20)$$

where  $r_c$  is the polar distance;  $\theta$  is the polar angle;  $r$  is the circle's radius and is taken to be dimensionless as  $r = 1$ ; and  $(r_0, \theta_0)$  is the polar coordinates for the circle's center.  $r_0$  corresponds to  $OC$  in Figure 0.2, and can be obtained by Eq. (3.6) and Eq. (3.7) as  $r_0 = \sqrt{2}(-s + \sqrt{2 - s^2})/2$ .  $\theta_0$  is shown in Figure 0.2, indicating the counterclockwise angle from the positive  $z$ -axis to the line connecting the origin  $O$  to the circle's center  $C$  (line  $OC$ ) and  $\theta_0 = 5\pi/4$ . Now Eq. (3.20) can be expressed in terms of  $s$ . Integrating  $r_c(\theta)$  from  $\pi/2$  to  $\pi$  and rearranging terms, the MeanInsideDiameter of the circular arc  $AB$  can be calculated as Eq. (3.21):

$$\text{MeanInsideDiameter}(s) = \frac{2 \int_{\frac{\pi}{2}}^{\pi} r_c(\theta) d\theta}{\frac{\pi}{2}} \quad (3.21)$$

$$= \frac{4}{\pi} \int_{\frac{\pi}{2}}^{\pi} \left[ \frac{\sqrt{2}}{2} (-s + \sqrt{2-s^2}) \cos\left(\theta - \frac{5}{4}\pi\right) + \sqrt{1 - (1-s\sqrt{2-s^2}) \sin^2\left(\theta - \frac{5}{4}\pi\right)} \right] d\theta.$$

The polar coordinates for an arbitrary ellipse is expressed in Eq. (3.22):

$$r_e(\theta) = \frac{P(\theta) + Q(\theta)}{R(\theta)}, \quad (3.22)$$

where  $P(\theta) = r_0 \left[ (r_2^2 - r_1^2) \cos(\theta + \theta_0 - 2\alpha) + (r_1^2 + r_2^2) \cos(\theta - \theta_0) \right],$

$$Q(\theta) = \sqrt{2} r_1 r_2 \sqrt{R(\theta) - 2r_0^2 \sin^2(\theta - \theta_0)},$$

$$R(\theta) = (r_2^2 - r_1^2) \cos(2\theta - 2\alpha) + r_1^2 + r_2^2.$$

$r_e$  is the polar distance;  $\theta$  is the polar angle;  $r_1$  and  $r_2$  are the ellipse's semi axes;  $(r_0, \theta_0)$  is the polar coordinates for the ellipse's center; and  $\alpha$  is the counterclockwise angle from the positive  $z$ -axis to the ellipse's major axis (line  $AC$  in Figure 0.6). The top right elliptical arc  $AB$  in Figure 0.6 has a hinged deformation  $s_2$  and an inherent deformation  $s_{02}$ .  $r_1$  and  $r_2$  corresponds to  $AC$  and  $BC$ , respectively ( $r=1$  to make it dimensionless) as expressed in Eq. (3.23) and Eq. (3.24):

$$r_1 = \left( -2s_{02} + 3 + \sqrt{-5s_{02}^2 + 6s_{02} + 3} \right) / 3, \quad (3.23)$$

$$r_2 = s_{02}. \quad (3.24)$$

$\alpha$  can be obtained by Eq. (3.12) as Eq. (3.25):

$$\alpha = \tan^{-1} \frac{3s_{02}}{-2s_{02} + 3 + \sqrt{-5s_{02}^2 + 6s_{02} + 3}} - \sin^{-1} \frac{3s_{02}}{\sqrt{9s_{02}^2 + \left(-2s_{02} + 3 + \sqrt{-5s_{02}^2 + 6s_{02} + 3}\right)^2}}. \quad (3.25)$$

$r_0$  can be obtained by Eq. (3.10) and Eq. (3.11) as Eq. (3.26):

$$\begin{aligned} r_0^2 &= (s_{02} \sin \alpha)^2 + (s_{02} \cos \alpha - s_2)^2 \\ &= \left\{ s_{02} \sin \left[ \begin{array}{c} \tan^{-1} \frac{3s_{02}}{-2s_{02} + 3 + \sqrt{-5s_{02}^2 + 6s_{02} + 3}} \\ -\sin^{-1} \frac{3s_{02}}{\sqrt{9s_{02}^2 + \left(-2s_{02} + 3 + \sqrt{-5s_{02}^2 + 6s_{02} + 3}\right)^2}} \end{array} \right] \right\}^2 \\ &\quad + \left\{ s_{02} \cos \left[ \begin{array}{c} \tan^{-1} \frac{3s_{02}}{-2s_{02} + 3 + \sqrt{-5s_{02}^2 + 6s_{02} + 3}} \\ -\sin^{-1} \frac{3s_{02}}{\sqrt{9s_{02}^2 + \left(-2s_{02} + 3 + \sqrt{-5s_{02}^2 + 6s_{02} + 3}\right)^2}} \end{array} \right] - s_2 \right\}^2. \end{aligned} \quad (3.26)$$

$\theta_0$  can be obtained as Eq. (3.27):

$$\begin{aligned} \theta_0 &= \frac{\pi}{2} + \angle COA = \frac{\pi}{2} + \angle OCD = \frac{\pi}{2} + \tan^{-1} \frac{OD}{CD} = \frac{\pi}{2} + \tan^{-1} \frac{s_{02} \cos \alpha - s_2}{s_{02} \sin \alpha} \\ &= \frac{\pi}{2} + \tan^{-1} \frac{s_{02} \cos \left[ \begin{array}{c} \tan^{-1} \frac{3s_{02}}{-2s_{02} + 3 + \sqrt{-5s_{02}^2 + 6s_{02} + 3}} \\ -\sin^{-1} \frac{3s_{02}}{\sqrt{9s_{02}^2 + \left(-2s_{02} + 3 + \sqrt{-5s_{02}^2 + 6s_{02} + 3}\right)^2}} \end{array} \right] - s_2}{s_{02} \sin \left[ \begin{array}{c} \tan^{-1} \frac{3s_{02}}{-2s_{02} + 3 + \sqrt{-5s_{02}^2 + 6s_{02} + 3}} \\ -\sin^{-1} \frac{3s_{02}}{\sqrt{9s_{02}^2 + \left(-2s_{02} + 3 + \sqrt{-5s_{02}^2 + 6s_{02} + 3}\right)^2}} \end{array} \right]}. \end{aligned} \quad (3.27)$$

Now Eq. (3.22) can be expressed in terms of  $s_2$  and  $s_{02}$ . Integrating  $r_e(\theta)$  from  $\pi/2$  to  $\pi$  and with substitutions of Eq. (3.23), Eq. (3.24), Eq. (3.25), Eq. (3.26), and Eq. (3.27) into Eq. (3.22), the MeanInsideDiameter of the elliptical arc  $AB$  can be calculated as Eq. (3.28):

$$\text{MeanInsideDiameter}(s) = \frac{4 \int_{\frac{\pi}{2}}^{\pi} r_e(\theta) d\theta}{\pi}. \quad (3.28)$$

The differences of using the average method (Eq. (3.16), with  $r_1$  and  $r_2$  representing the maximum and minimum diameters for each mode) and the integral methods (Eq. (3.21) and Eq. (3.28)) are quantified in Table 0.2 for Mode 2, in Table 0.3 for Mode 3, in Table 0.4 for Mode 4, and in Table 0.5 for Mode 5. Values of  $s_{01}$  and  $s_{02}$  are shown in these tables where appropriate;  $s$  ranges from 1.00 to 0.90 to control the ovality  $q$  under 10%. It can be observed that the industrial average method underestimates the ovality  $q$  and overestimates the ovality reduction factor  $C$ ; also, hinges do not produce as large errors as the initial ovalities  $s_{01}$  and  $s_{02}$  do.

Table 0.2 Influences of Integral Methods on Mode 2

s	using average		using integral		difference	
	q	C	q	C	q	C
1.00	0.00%	1.00	0.00%	1.00	0.00%	0.00%
0.99	1.00%	0.91	1.00%	0.91	-0.32%	0.03%
0.98	1.98%	0.84	1.99%	0.84	-0.64%	0.11%
0.97	2.96%	0.77	2.99%	0.77	-0.97%	0.26%
0.96	3.93%	0.70	3.98%	0.70	-1.30%	0.45%
0.95	4.89%	0.65	4.97%	0.64	-1.63%	0.71%
0.94	5.84%	0.59	5.95%	0.59	-1.97%	1.01%
0.93	6.79%	0.55	6.94%	0.54	-2.31%	1.38%
0.92	7.73%	0.50	7.93%	0.49	-2.66%	1.79%
0.91	8.66%	0.46	8.92%	0.45	-3.01%	2.27%
0.90	9.59%	0.43	9.91%	0.41	-3.36%	2.79%

Table 0.3 Influences of Integral Methods on Mode 3

s	using average		using integral		difference	
	q	C	q	C	q	C
1.00	0.00%	1.00	0.00%	1.00	0.00%	0.00%
0.99	1.00%	0.91	1.00%	0.91	-0.54%	0.05%
0.98	2.00%	0.84	2.02%	0.83	-1.09%	0.19%
0.97	2.99%	0.77	3.04%	0.76	-1.64%	0.44%
0.96	3.98%	0.70	4.07%	0.69	-2.21%	0.78%
0.95	4.98%	0.64	5.11%	0.63	-2.78%	1.22%
0.94	5.97%	0.59	6.17%	0.58	-3.36%	1.76%
0.93	6.96%	0.54	7.23%	0.53	-3.96%	2.40%
0.92	7.95%	0.49	8.31%	0.48	-4.56%	3.15%
0.91	8.94%	0.45	9.40%	0.43	-5.18%	3.99%
0.90	9.93%	0.41	10.51%	0.39	-5.81%	4.94%

Table 0.4 Influences of Integral Methods on Mode 4

( $s_0 = 0.95$ )	using average		using integral		difference	
$s_1$	q	C	q	C	q	C
0.95	4.94%	0.64	5.13%	0.63	-3.90%	1.69%
0.94	5.93%	0.59	6.26%	0.57	-5.47%	2.83%
0.93	6.92%	0.54	7.39%	0.52	-6.79%	4.06%
0.92	7.91%	0.49	8.54%	0.47	-7.96%	5.40%
0.91	8.90%	0.45	9.71%	0.42	-9.03%	6.83%
0.90	9.89%	0.42	10.89%	0.38	-10.04%	8.35%

Table 0.5 Influences of Integral Methods on Mode 5

$(s_{02} = 0.95,$ $s_{01} = 0.92)$	using average		using integral		difference		
	$s_1$	q	C	q	C	q	C
	0.92	7.91%	0.49	8.54%	0.47	-7.96%	5.40%
	0.91	8.85%	0.46	9.71%	0.42	-9.73%	7.29%
	0.90	9.77%	0.42	10.87%	0.38	-11.27%	9.22%

### 3.5 Chapter Summary

This chapter parametrically models common cross sectional geometries of rigid pipes. Circular cross sections are the most common ones in underground pipes; elliptical cross sections are also common due to imperfect manufacturing and deformation. Therefore, circular and elliptical arcs are used to describe the composite pipe geometries with possible initial ovalities; and with hinges at the crown, at the invert, and along the spring line. Area loss, 3-D pipe segments, and accurate ovality calculation are explored as extensions of the modeling. The improvement of using the integral method over the industrial average method in calculating the ovality and the ovality reduction factor are quantified. APPENDIX A gives some of the MATLAB® code for this chapter.

## CHAPTER 4

### DISTORTED GEOMETRY SIMULATION

The desirable measurement plane should be perpendicular to the pipe's centerline with the measurement origin on the centerline. Such an ideal case happens only if the pipe offers a smooth yet constrained contact area for non-articulated robotic wheels. In this way, the wireframe can be formed with 2-D data plotted along a straight line. However, mineral deposits, lack of side wall support, and articulated wheels make the robot deviate from the centerline, and this tendency continues until gravity pulls the robot back on track. Such a process can be viewed as transformations between unstable states and stable states of equilibrium.

Robotic deviation yields an oblique measurement plane. A correct wireframe can be recovered with coordinate transformations defined by the robot-pipe relationship. But, if the robot-pipe relationship cannot be determined or is determined wrongly, the only choice is plotting the 2-D data along a straight line, resulting in wireframe distortion. Industry practice usually neglects or fails to document corrections with regard to this issue. This chapter depicts the necessity of a correction algorithm by simulating the distorted wireframe of the a straight pipe and a bend with circular cross sections, caused by pitch, yaw and roll with offset, individually.

## 4.1 Measurement Origin Offset Relative to Pipe

One problem caused by measurement origin offsets is the non-uniform coverage of the interior surface, as mentioned in Section 2.6.3. For a side-facing scanner, if the center of the revolving scanning head, which is usually chosen to be the origin, is not properly aligned with the pipe's centerline, then the regions close to the origin are scanned more densely than the regions far away from the origin. For a forward-facing scanner, if radial sampling is used, as discussed in Section 2.6.1, a similar point density distribution results. Figure 4.1 shows the influence of an offset on the point density distribution. A coarse angular resolution together with a large offset yields poor coverage over some regions.

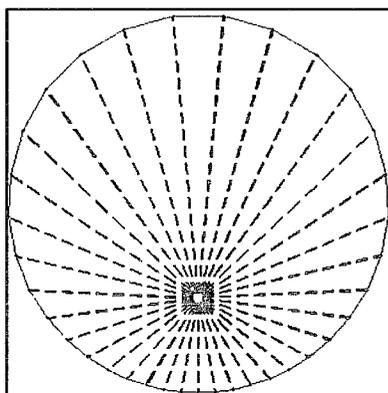


Figure 4.1 Non Uniform Point Coverage Caused by Measurement Origin Offset

Measurement origin offsets are almost inevitable when the pipe geometry changes or the robot deviates from the centerline. This section examines the offsets caused by both, namely the deformation-induced offset and the pitch-induced offset.

### 4.1.1 Deformation Induced Offset

Pipe deformation usually shortens the vertical height of the cross section, elevating the origin from the centerline; on the other hand, deformation also flattens the terrain that the wheels travel on, sinking the origin from the centerline. The dominant effect depends

on the ratio of the wheels span to the pipe radius  $w = W/r$  as shown in Figure 4.2, and the pipe deformation severity  $s = r_2/r$  as defined in Section 3.1.

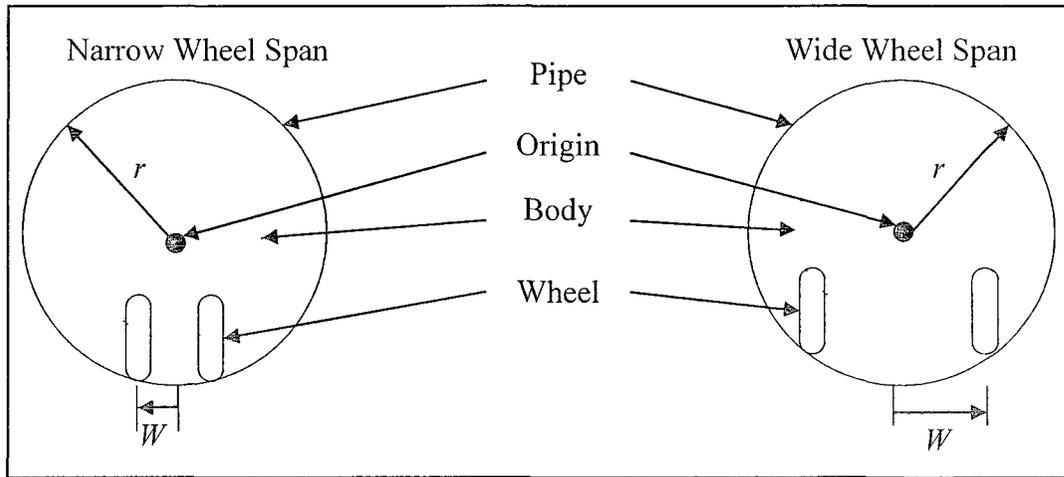


Figure 4.2 Different Configurations of Wheel Span

Figure 4.3 shows a cross sectional rear view of the geometry of a robot inside two pipes with  $O_1$  and  $O_2$  denoting the center of the original circle and the deformed ellipse. The  $x$ -axis goes perpendicularly into the paper, so the  $xy$ -plane divides the left and right of the robot symmetrically. Consider only the right half:  $W$  is the half span between the left and right wheels;  $C$  and  $D$  indicate where the right wheel lands for each pipe, respectively, so  $AC$  and  $BD$  denote the bottom plane of the robot. Suppose the robot has a height of  $H$  from the bottom plane to the measurement origin and also suppose the robot is carefully chosen to match the origin with the pipe's centerline for the circular pipe, i.e.,  $O_1A = H$  and  $H^2 + W^2 = r^2$  with  $r$  being the circle's radius. If the same robot is used to scan the deformed pipe and the new origin lands at point  $O_3$ , it is of interest to find the

value of  $O_f = O_2O_3$ , which is the offset between the new pipe's centerline and the new measurement origin.

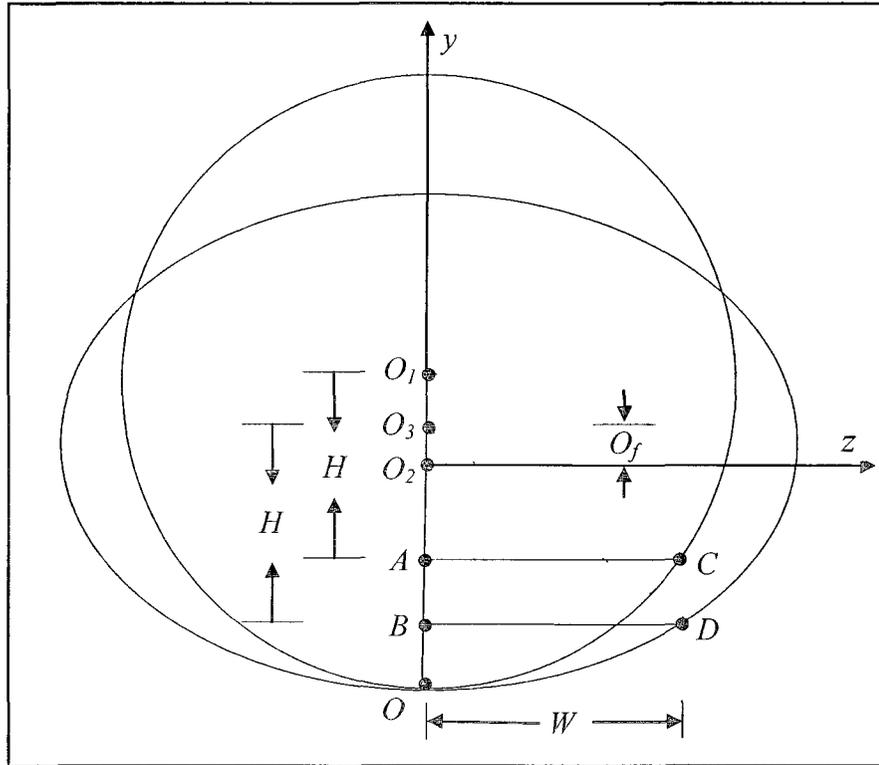


Figure 4.3 Cross Sectional Geometry of Relating Wheel Span and Pipe

The deformed cross section is described in Eq. (3.1) and is reproduced as Eq. (4.1):

$$\frac{z^2}{r_1^2} + \frac{y^2}{r_2^2} = 1, \quad (4.1)$$

where  $r_1$  and  $r_2$  are the semi-major and semi-minor axis length. By substituting

$BD=z=W$  into Eq. (4.1), Eq. (4.2) can be derived as:

$$BO_2 = y = \frac{r_2}{r_1} \sqrt{r_1^2 - W^2},$$

$$\begin{aligned}
OB &= OO_2 - BO_2 = r_2 - \frac{r_2}{r_1} \sqrt{r_1^2 - W^2}, \\
OO_3 &= OB + BO_3 = r_2 - \frac{r_2}{r_1} \sqrt{r_1^2 - W^2} + H, \\
O_f &= OO_3 - OO_2 = r_2 - \frac{r_2}{r_1} \sqrt{r_1^2 - W^2} + H - r_2 = \sqrt{r^2 - W^2} - \frac{r_2}{r_1} \sqrt{r_1^2 - W^2}. \quad (4.2)
\end{aligned}$$

$r_1$  and  $r_2$  are defined in Section 3.1 and reproduced as Eq. (4.3) and Eq. (4.4):

$$r_1 = \frac{\left(-2s + 3 + \sqrt{-5s^2 + 6s + 3}\right)r}{3}, \quad (4.3)$$

$$r_2 = sr. \quad (4.4)$$

By substituting Eq. (4.3) and Eq. (4.4) into Eq. (4.2) to normalize  $O_f$  in terms of the circular pipe's radius  $r$ , it can be obtained that

$$o_f = \frac{\sqrt{r^2 - W^2} - \frac{r_2}{r_1} \sqrt{r_1^2 - W^2}}{r} = \sqrt{1 - w^2} - \sqrt{s^2 - \left(\frac{3s}{-2s + 3 + \sqrt{-5s^2 + 6s + 3}} w\right)^2}.$$

The boundary condition is that the origin cannot exceed the limit of the pipe's circumference, i.e.,  $|O_f| \leq r_2$ . Therefore any out of boundary  $o_f$  shall be discarded. Figure 4.4 shows a RGB plot of  $o_f$  versus the ratio  $w$  and the deformation severity  $s$ . As predicted, the new origin can be either above or below the new pipe axis; therefore,  $o_f$  can be either positive or negative. Green indicates  $o_f = 0$ ; warm colors (positive values) are dominant, meaning that the new measurement origin is usually above the new pipe axis

for ordinary wheels spans; cool colors (negative values) only exist for extremely large wheels spans, which is rarely true for a real robot

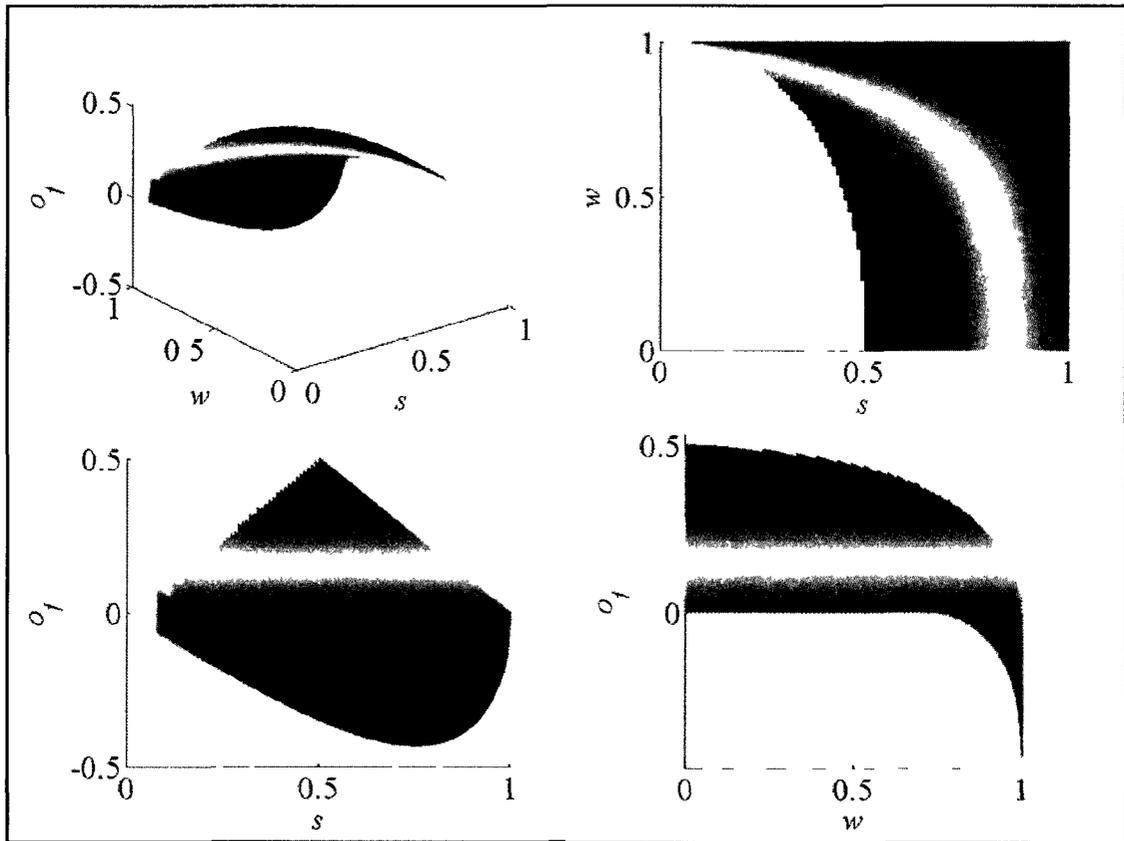


Figure 4.4 Offset Induced by Pipe Deformation and Wheel Span

It is clear from this plot that for different pipe deformations, the selection range of the robot varies. From a boundary-condition point of view, all wheels span is applicable for pipe roundness larger than 0.5. It is true because the minor axis length  $2r_2$  shall be larger than the robotic height  $H = r$  for a zero wheels span, i.e.,  $r_2/r \geq 0.5$ . As the roundness level  $s$  gets smaller, the selection range shrinks dramatically. For example, the lowest  $w_{min} \approx 0.7$  for  $s = 0.4$  and  $w_{min} \approx 0.85$  for  $s = 0.3$ . Therefore, if severely deformed

pipe segments  $r_2/r \leq 0.5$  are present, a large wheel span offers flexibility through such difficult areas. On the contrary, for slightly deformed pipes ( $0.8 \leq r_2/r < 1$ ), geometry-induced offsets are small except for extremely large wheel spans.

#### 4.1.2 Pitch Induced Offset

Robotic pitch takes the measurement origin off the pipe axis and the terrain determines the magnitude and duration of such a temporary offset. The offset is usually coarsely calculated as Eq. (4.5):

$$O_f = L \sin \alpha = \left( \frac{L}{r} \right) r \sin \theta_v \quad (4.5)$$

where  $L$  is the length from the measurement origin to the rear wheel's center and  $\theta_v$  is the pitch angle. For a robot stepping down a bump, i.e., only the rear wheel is elevated,  $L$  is the length from the measurement origin to the front wheel's center along the robotic body.

However, since the rotational axis does not pass through the wheel's center, the transformation involves a rotation together with a translation. Therefore,  $L$  shall be corrected for different  $\theta_v$ 's, i.e.,  $L_2$  is the correct value to substitute  $L$  in Eq. (4.5) as  $O_f = L_2 \sin \theta_v$ . In Figure 4.5, known variables are the pipe's radius  $r$ , the wheel's radius  $R_w$ , the robotic length  $L$ , and the pitch angle  $\theta_v$ ;  $O_f$  is the quantity of interest.

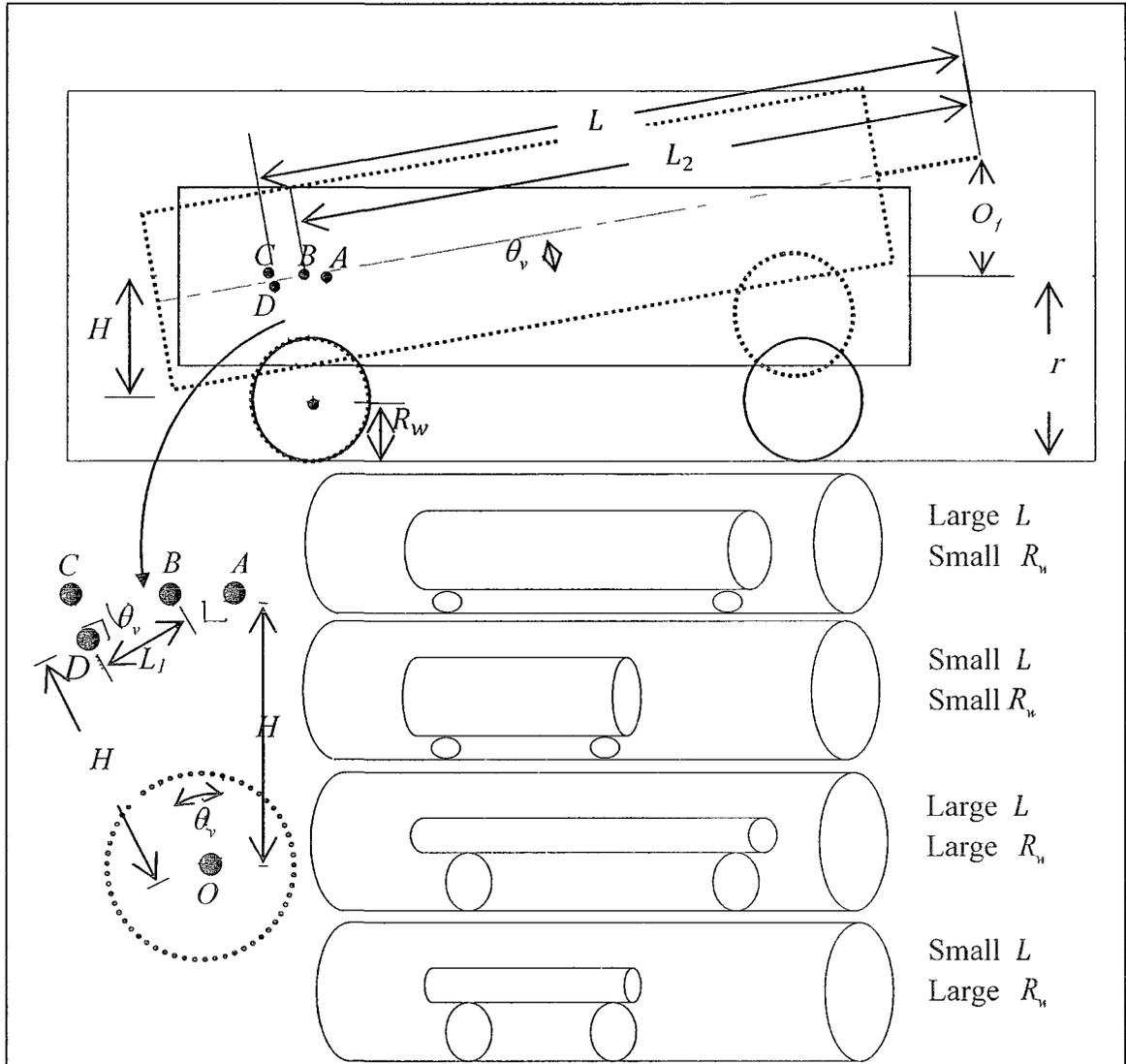


Figure 4.5 Geometry of Robot with Different Length and Wheel Radius

It can be seen in the zoom-in view that  $L_2 = L - L_1$ , so  $L_1$  is needed. It is known that  $\angle OAC = \angle CDB = \pi/2$  and  $\angle CBD = \theta_v$ . Because  $OAC$  and  $BCD$  are similar triangles, so  $\angle AOC = \theta_v$ . Because  $OA$  and  $OD$  are the perpendicular distances from the wheel's center to the robotic body's centerline; therefore,  $OAB$  and  $OBD$  are equal triangles,  $\angle DOB = \theta_v/2$ , and  $L_1 = H \tan(\theta_v/2)$ . Note that  $H = R - R_w$ . After substitutions, the dimensionless form for the pitch-induced offset is:

$$O_f = \left( L - H \tan \frac{\theta_v}{2} \right) \sin \alpha = \left[ L - (r - R_w) \tan \frac{\theta_v}{2} \right] \sin \theta_v = \left[ \frac{L}{r} - \left( 1 - \frac{R_w}{r} \right) \tan \frac{\theta_v}{2} \right] r \sin \theta_v,$$

$$o_f = \left[ \frac{L}{r} - \left( 1 - \frac{R_w}{r} \right) \tan \frac{\theta_v}{2} \right] \sin \theta_v.$$

Compared to the Eq. (4.5), it is obvious that  $O_f$  is dependent on the ratio of the robotic length to the pipe's radius  $l = L/r$ , the ratio of the wheel's radius to the pipe's radius  $r_w = R_w/r$ , and the pitch angle  $\theta_v$ . The constraint are  $L \geq 4R_w$ ,  $R_w \leq r/2$  and  $O_f \leq r$ . RGB plots are generated to show the influences of  $l$ ,  $r_w$  and  $\theta_v$  on  $o_f$ .

In the top four plots of Figure 4.6,  $l$  is set to be (1.0, 1.5, 2.0, 4.0), the maximum  $r_w$  is allowed to be 0.25, and the maximum  $\theta_v$  is allowed to be  $30^\circ$ . As the robotic length  $l$  increases, it is becoming dominant in determining  $o_f$ , making  $o_f$  increase almost linearly with  $\theta_v$ , regardless of the wheel radius  $r_w$ . The bottom four plots show the effects of different  $r_w$  (0.1, 0.2, 0.3, 0.4) on  $o_f$ ; the maximum  $l$  is allowed to be 2, and the maximum  $\theta_v$  is allowed to be  $30^\circ$ . For a small  $\theta_v$ , though the lower limit for  $l$  is determined by  $l \geq 4r_w$ ,  $o_f$  does not vary much with different  $r_w$ 's.

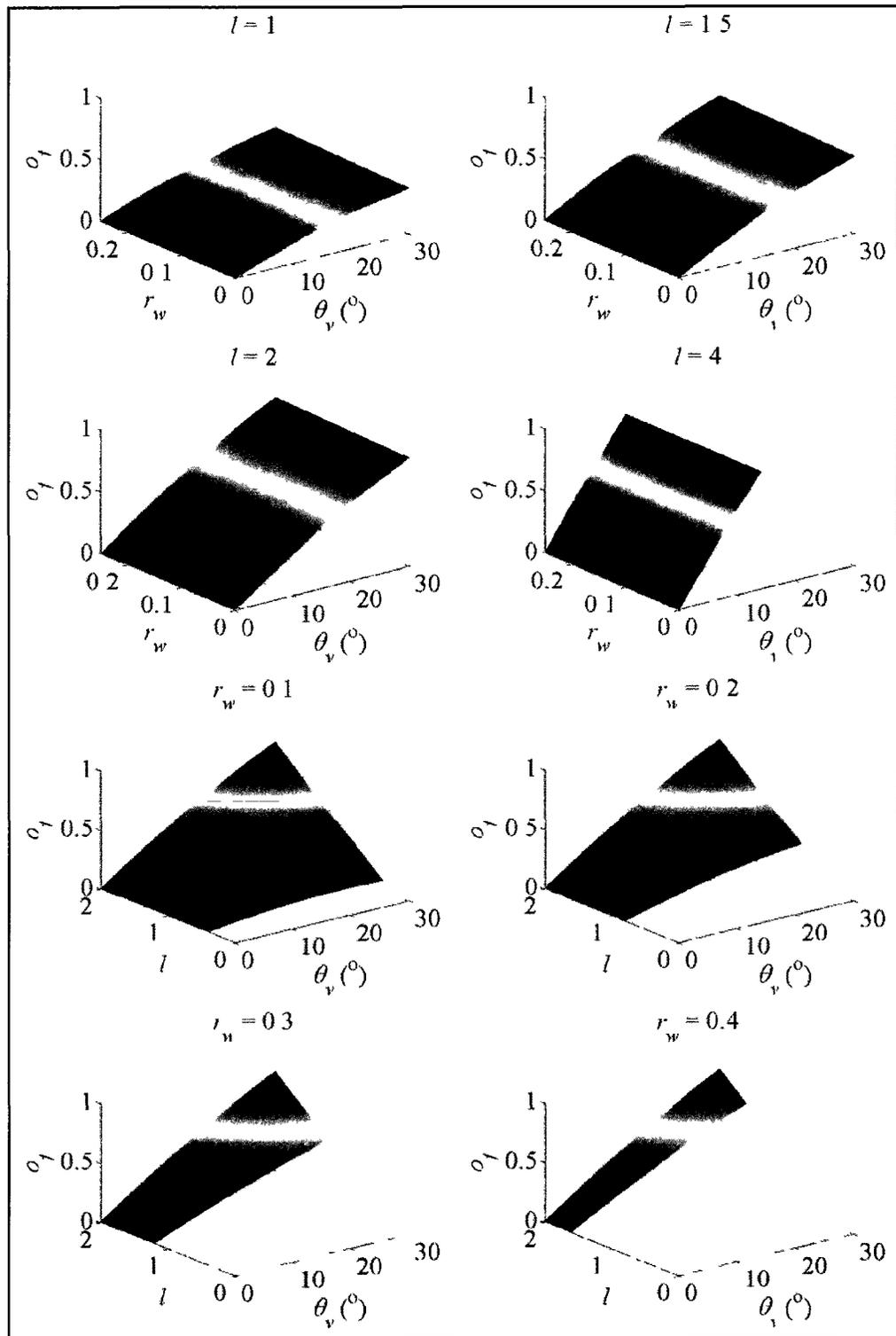


Figure 4.6 Offset Influenced by Robotic Length and Wheel Radius

## 4.2 Oblique Cross Sections of Bend

Bends are often used to guide a single pipe around difficult topography with layout optimization. Tough terrains for digging and crowded existing pipeline network are two niches where bends can be useful. Bends serve as a valuable feature for geometry co-registration. This section explores the geometric shapes acquired by a 2-D scanner.

### 4.2.1 Geometry of Bend

Bends are manufactured with different configurations, as shown in Figure 4.7 [36]. Letter code designations are as follow:  $H$  : height of bend;  $L$  : length of bend;  $R$  : centerline radius;  $W$  : pipe wall thickness;  $a$  : first straight;  $b$  : second straight;  $r$  : pipe radius ;  $w$  : width of bend.

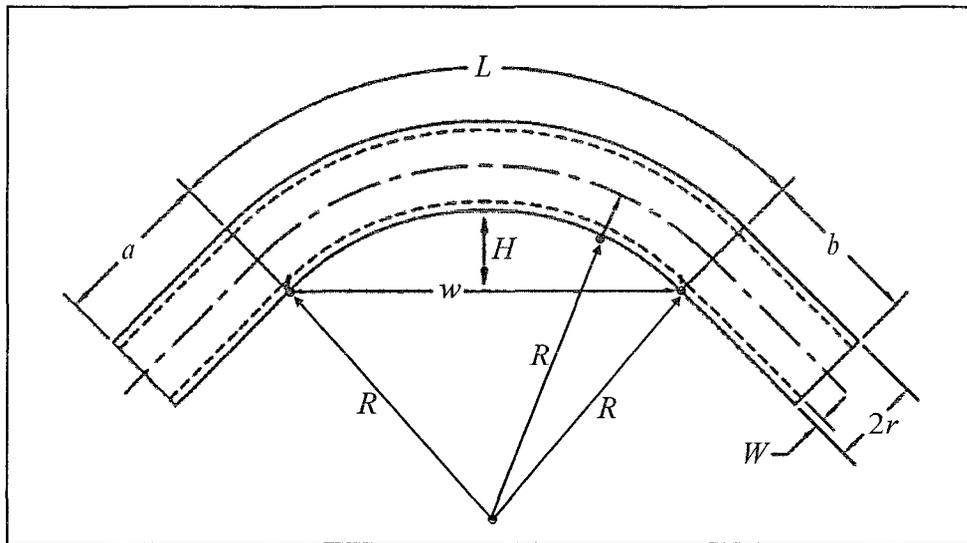


Figure 4.7 Pipe Bend with Manufacturing Parameters

A bend is usually manufactured to have a constant centerline radius, namely  $R$  in Figure 4.7, so that it can be best modeled by a torus. If the connecting straight pipe has a

circular cross section, so shall the bend. The equation for a circular torus is expressed in Eq. (4.6):

$$\left(R - \sqrt{x^2 + y^2}\right)^2 + z^2 = r^2, \quad (4.6)$$

where  $R$  is the major radius of the centerline and  $r$  is the minor radius of the inner surface; the coordinates system  $xyz$  remain the same, with  $z$ -axis indicating the revolving axis. Figure 4.8 shows two types of bends (left:  $R=20$ ,  $r=10$ , level of bend =  $90^\circ$ ; right:  $R=100$ ,  $r=10$ , level of bend =  $60^\circ$ ).

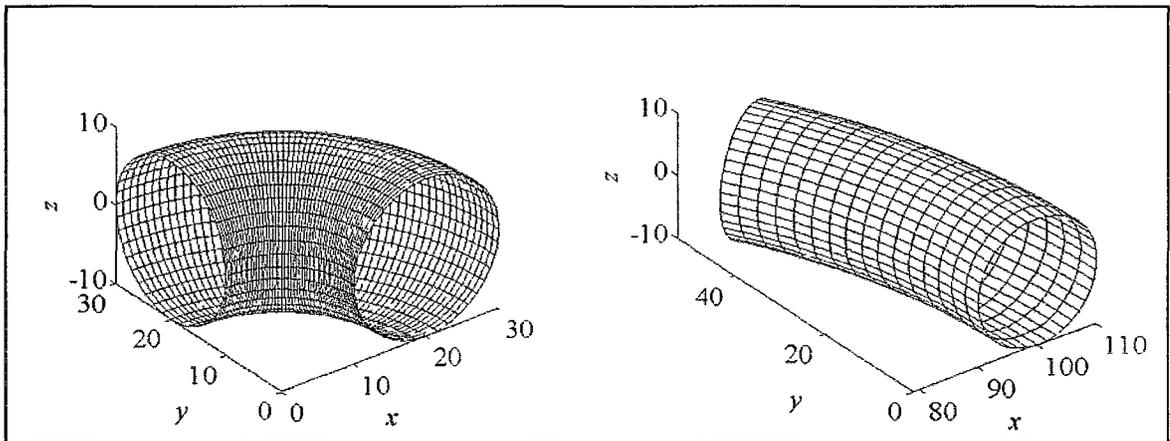


Figure 4.8 Three Dimensional Torus

If the connecting straight pipe has an elliptical cross section, a more general equation for an elliptical torus is needed as expressed in Eq. (4.7):

$$\left(r_2^2 \left(\frac{z^2}{r_1^2} - 1\right) - R^2 + x^2 + y^2\right)^2 = 4r_1^2 R^2 \left(1 - \frac{z^2}{r_1^2}\right), \quad (4.7)$$

where  $r_1$  and  $r_2$  are the semi-major and semi-minor axes, respectively.

#### 4.2.2 Derivation

If the robotic orientation stays tangent to the pipe's centerline, the cross sectional shape collected by a laser ring is a circle or an ellipse, which shall be the same as that in a connecting straight pipe. However, the laser ring is usually projected well in front of the wheels. Therefore, the laser ring has already entered the bend zone while the wheels are still in the straight portion of the pipe, or has already passed the straight portion while the wheels are still in the bend. Figure 4.9 [33] shows this almost inevitable phenomenon as a prototype robot is traveling through a sharp bend[33].

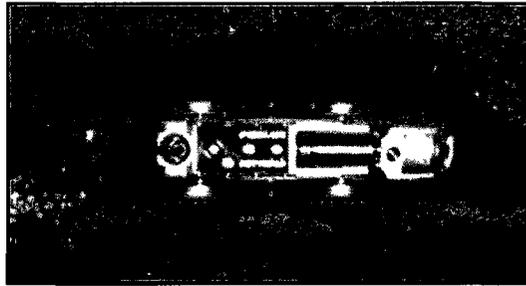


Figure 4.9 Robot Traveling in Bend

It is of interest to derive the oblique cross sectional shapes of a bend, generated by  $\theta_v$  and  $\theta_h$ . For easiness, only the circular cross section is considered. An equivalent form of Eq. (4.6) is expressed in Eq. (4.8):

$$\left(x^2 + y^2 + z^2 + R^2 - r^2\right)^2 = 4R^2 \left(x^2 + y^2\right). \quad (4.8)$$

Without loss of generality, the  $\theta_v$  and  $\theta_h$  discussed in this chapter are considered with regard to a vertical bend, which essentially changes the elevation of a pipeline. It is worth noting that  $\theta_v$  in a vertical bend is equivalent to  $\theta_h$  in a horizontal bend, and vice

versa; also, pure  $\theta_v$  or  $\theta_h$  in a composite bend (vertical plus horizontal) is equivalent to some combined angle ( $\theta_v$  plus  $\theta_h$ ) in a pure vertical or horizontal bend. Therefore, the equations derivations herein are comprehensive.

$\theta_v$  in a vertical bend can be equivalently thought of as moving the laser ring plane for a distance  $c = R \sin \theta_v$  along the  $x$ -axis. In this case, Eq. (4.8) becomes

$\left[ (x-c)^2 + y^2 + z^2 + R^2 - r^2 \right]^2 = 4R^2 \left[ (x-c)^2 + y^2 \right]$ . Letting  $x=0$ , expanding and rearranging terms, Eq. (4.9) can be obtained as:

$$\left( y^2 + z^2 \right)^2 = 2(r^2 + R^2 - c^2)y^2 + 2(r^2 - R^2 - c^2)z^2 + 4R^2c^2 - (c^2 + R^2 - a^2)^2, \quad (4.9)$$

where  $c = R \sin \theta_v$ .

There is no shortcut to derive the oblique shape caused by  $\theta_h$  in a vertical bend. Coordinates transformation has to be utilized. Because  $\theta_h$  is essentially a rotation about  $y$ -axis, therefore substituting  $x = x_1 \cos \theta_h - z_1 \sin \theta_h$ ,  $y = y_1$  and  $z = x_1 \sin \theta_h + z_1 \cos \theta_h$  into Eq. (4.8),

$$\begin{aligned} & \left[ (x_1 \cos \theta_h - z_1 \sin \theta_h)^2 + y_1^2 + (x_1 \sin \theta_h + z_1 \cos \theta_h)^2 + R^2 - r^2 \right]^2 \\ & = 4R^2 \left[ (x_1 \cos \theta_h - z_1 \sin \theta_h)^2 + y_1^2 \right], \\ & \left[ x_1^2 + y_1^2 + z_1^2 + R^2 - r^2 \right]^2 = 4R^2 \left[ (x_1 \cos \theta_h - z_1 \sin \theta_h)^2 + y_1^2 \right]. \end{aligned}$$

Letting  $x_1 = 0$ , removing the subscripts, and expanding and rearranging terms, Eq. (4.10) can be obtained as:

$$(y^2 + z^2)^2 = 2(r^2 + R^2)y^2 + 2(r^2 - R^2 + 2c^2)z^2 - (r^2 - R^2)^2, \quad (4.10)$$

where  $c = R \sin \theta_h$ .

As shown in Figure 4.10, the order of applying  $\theta_v$  and  $\theta_h$  makes different robotic orientations that have the same set of  $\theta_v$  and  $\theta_h$ . ( $xyz$ : original coordinates system;  $x_1y_1z_1$ : coordinates system after  $\theta_v$ ;  $x_2y_2z_2$ : coordinates system after  $\theta_h$ ).

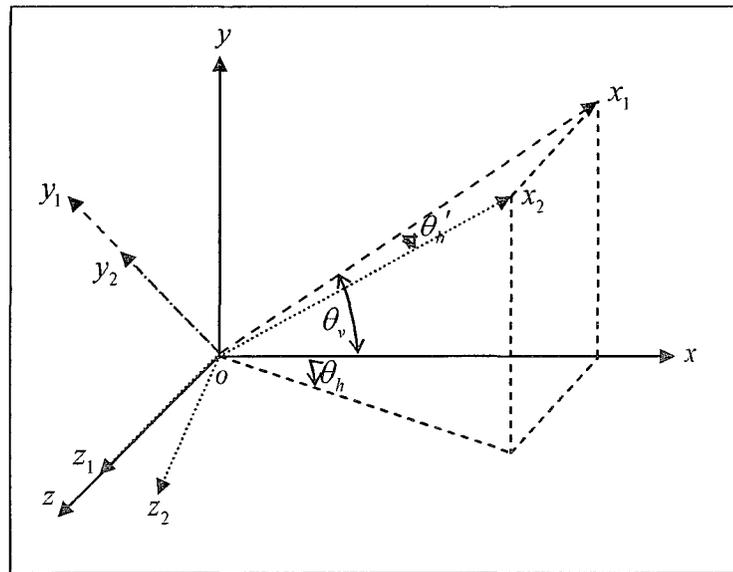


Figure 4.10 Two Sequential Coordinates Transformations

$\theta_v$  is applied first to rotate the original coordinates system  $xyz$  about the  $z$ -axis; and a new coordinates system  $x_1y_1z_1$  is generated:

$$\left[ (x_1 - c)^2 + y_1^2 + z_1^2 + R^2 - r^2 \right]^2 = 4R^2 \left[ (x_1 - c)^2 + y_1^2 \right],$$

where  $c = R \sin \theta_v$ . It is not hard to find the new yaw angle  $\theta_h'$  from  $x_1y_1z_1$  to  $x_2y_2z_2$  from the geometry,  $\theta_h' = \tan^{-1}(\tan \theta_h \cos \theta_v)$ . Then another rotation  $\theta_h'$  about the new

$y_1$ -axis brings the  $x_2$ -axis in place, i.e.,  $x_1 = x_2 \cos \theta'_h - z_2 \sin \theta'_h$ ,  $y_1 = y_2$  and

$z_1 = x_2 \sin \theta'_h + z_2 \cos \theta'_h$ . After substitution,

$$\begin{aligned} & \left[ \left( x_2 \cos \theta'_h - z_2 \sin \theta'_h - c \right)^2 + y_2^2 + \left( x_2 \sin \theta'_h + z_2 \cos \theta'_h \right)^2 + R^2 - r^2 \right]^2 \\ & = 4R^2 \left[ \left( x_2 \cos \theta'_h - z_2 \sin \theta'_h - c \right)^2 + y_2^2 \right]. \end{aligned}$$

Letting  $x_2 = 0$ ,

$$\begin{aligned} & \left[ \left( z_2 \sin \theta'_h + c \right)^2 + y_2^2 + \left( z_2 \cos \theta'_h \right)^2 + R^2 - r^2 \right]^2 = 4R^2 \left[ \left( z_2 \sin \theta'_h + c \right)^2 + y_2^2 \right], \\ & \left[ y_2^2 + z_2^2 + 2cz_2 \sin \theta'_h + c^2 + R^2 - r^2 \right]^2 = 4R^2 \left[ y_2^2 + \left( z_2 \sin \theta'_h \right)^2 + 2cz_2 \sin \theta'_h + c^2 \right]. \end{aligned}$$

Removing the subscripts, Eq. (4.11) can be obtained as:

$$\left[ y^2 + z^2 + 2cz \sin \theta'_h + c^2 + R^2 - r^2 \right]^2 = 4R^2 \left[ y^2 + \left( z \sin \theta'_h \right)^2 + 2cz \sin \theta'_h + c^2 \right], \quad (4.11)$$

where  $c = R \sin \theta_v$  and  $\theta'_h = \tan^{-1}(\tan \theta_h \cos \theta_v)$ .

#### 4.2.3 Plotting

Because Eq. (4.9), Eq. (4.10) and Eq. (4.11) are all quadric polynomials, the lower and upper limits for  $y$  or  $z$  have to be found in order to avoid imaginary values for the plotting purpose. Note that Eq. (4.9) and Eq. (4.10) have a similar form:

$$\left( y^2 + z^2 \right)^2 = dy^2 + ez^2 + f.$$

Expanding and rearranging terms, and treating  $z^2$  as an unknown value,

$$z^4 + (2y^2 - e)z^2 + y^4 - dy^2 - f = 0,$$

$$z^2 = \frac{1}{2}(-2y^2 + e + \sqrt{\Delta}),$$

where  $\Delta = (2y^2 - e)^2 - 4(y^4 - dy^2 - f) = 4(d - e)y^2 + e^2 + 4f$ . Because  $z^2 \geq 0$ , so the boundary limits for  $y$  can be found by taking  $z^2 = 0$ , i.e.,

$$(-2y^2 + e)^2 = 4(d - e)y^2 + e^2 + 4f,$$

$$y^4 - dy^2 - f = 0,$$

$$y^2 = \frac{1}{2}(d \pm \sqrt{d^2 + 4f}).$$

Suppose the vertical bend is concaving upwards, then

$$-\sqrt{\frac{1}{2}(d - \sqrt{d^2 + 4f})} \leq y \leq -\sqrt{\frac{1}{2}(d + \sqrt{d^2 + 4f})}.$$

All there is left to do is substituting the proper terms for  $d$ ,  $e$  and  $f$  from Eq. (4.9) and Eq. (4.10). Figure 4.11 depicts the oblique shapes in a sharp bend, caused by  $\theta_v$  or  $\theta_h$ , respectively.  $R = 20$  and  $r = 10$  are used, the same as in the left picture of Figure 4.8. Figure 4.12 depicts the oblique shapes in a sharp bend, caused by  $\theta_v$  or  $\theta_h$ , respectively.  $R = 100$  and  $r = 10$  are used, the same as in the right picture of Figure 4.8.

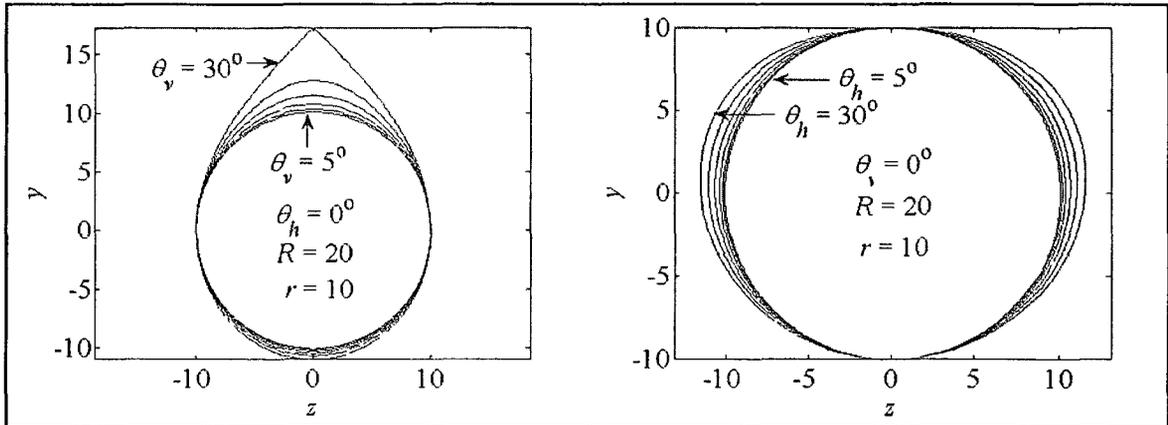


Figure 4.11 Oblique Shapes in Sharp Bend Caused by Pitch or Yaw Only

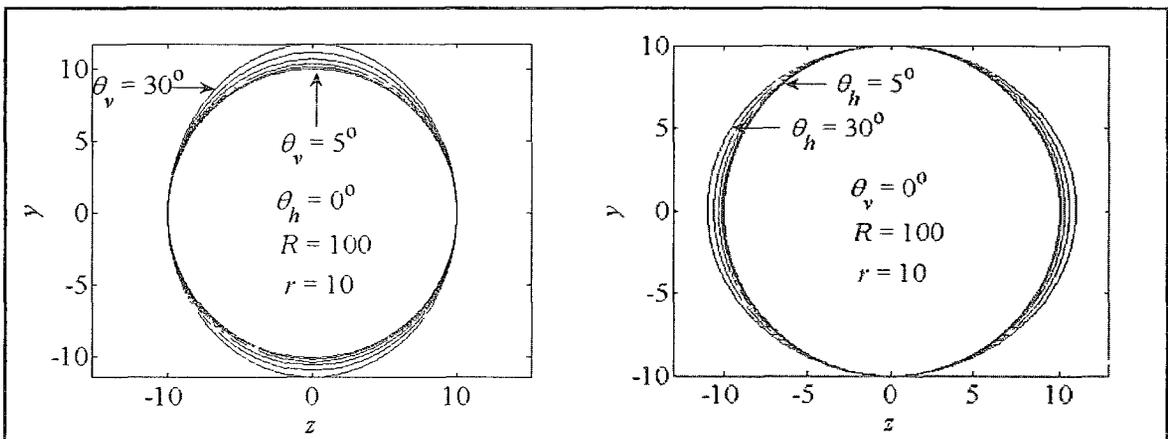


Figure 4.12 Oblique Shapes in Smooth Bend Caused by Pitch or Yaw Only

For Eq. (4.11), rewrite the constant that  $d^2 = c^2 + R^2 - r^2$ , then

$$\left[ y^2 + z^2 + 2cz \sin \theta_h' + d^2 \right]^2 = 4R^2 \left[ y^2 + \left( z \sin \theta_h' \right)^2 + 2cz \sin \theta_h' + c^2 \right].$$

Solving for  $y^2$ ,

$$y^2 = 2R^2 \pm \sqrt{2} \sqrt{2R^4 + 2R^2 c^2 - 2R^2 d^2 - R^2 z^2 \cos(2\theta_h') - R^2 z^2 - 2cz \sin \theta_h' - d^2 - z^2}.$$

The values that make the value in the squared root equal to zero are the lower and upper limits for  $z$ , so

$$-\sqrt{\frac{2R^4 + 2R^2c^2 - 2R^2d^2}{R^2 \cos(2\theta_h') + R^2}} \leq z \leq \sqrt{\frac{2R^4 + 2R^2c^2 - 2R^2d^2}{R^2 \cos(2\theta_h') + R^2}}$$

Figure 4.13 depict the oblique shapes in a sharp bend, caused by  $\theta_v$  and  $\theta_h$  together.  $R = 20$  and  $r = 10$  are used, the same as in the left picture of Figure 4.8. Figure 4.14 depict the oblique shapes in a smooth bend, caused by  $\theta_v$  and  $\theta_h$  together.  $R = 100$  and  $r = 10$  are used, the same as in the right picture of Figure 4.8.

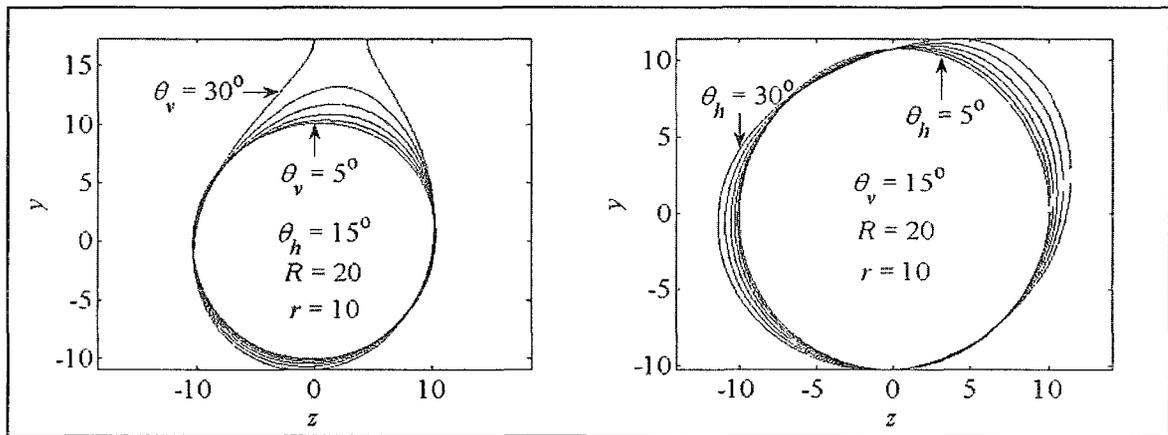


Figure 4.13 Oblique Shapes in Sharp Bend Caused by Pitch and Yaw Combined

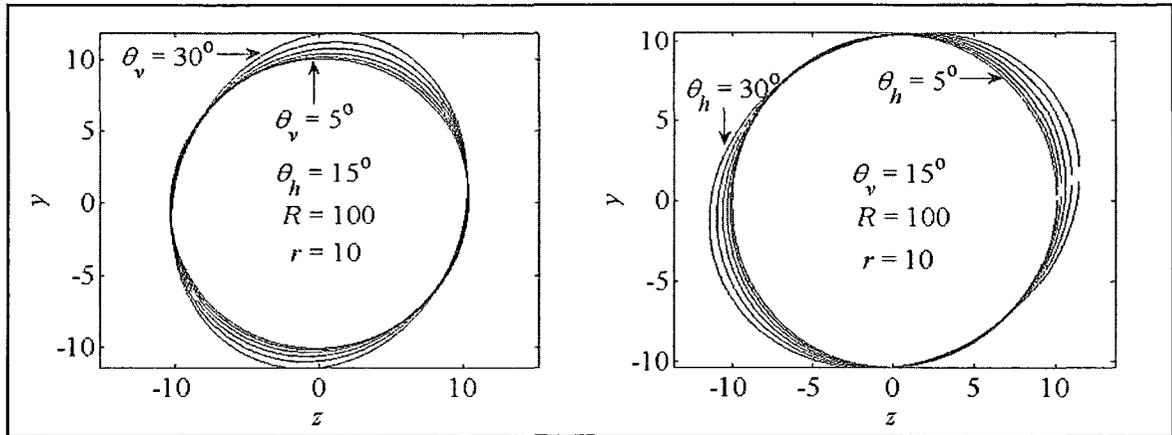


Figure 4.14 Oblique Shapes in Smooth Bend Caused by Pitch and Yaw Combined

#### 4.2.4 Verification

For a 2-D scanner, the coordinates of the radial measurements can be determined using the same methods as in the elliptical case. Suppose the measurement origin locates at  $(x_0, y_0, z_0)$  and any laser ray originated from the origin has a direction vector  $(u, v, w)$ .

By substituting  $x = x_0 + ut$ ,  $y = y_0 + vt$  and  $z = z_0 + wt$  into the Eq. (4.7),

$$\left[ r_2^2 \left( \frac{(z_0 + wt)^2}{r_1^2} - 1 \right) - R^2 + (x_0 + ut)^2 + (y_0 + vt)^2 \right]^2 = 4r_1^2 R^2 \left[ 1 - \frac{(z_0 + wt)^2}{r_1^2} \right].$$

Here,  $t$  represents the distance from the origin to the circumference. Treating  $t$  as the unknown variable; expanding and rearranging the terms, Eq. (4.12) can be obtained:

$$c_4 t^4 + c_3 t^3 + c_2 t^2 + c_1 t + c_0 = 0, \quad (4.12)$$

where coefficients  $(c_4, c_3, c_2, c_1, c_0)$  are defined as:

$$c_4 = (1 + kw^2)^2,$$

$$c_3 = 4(1 + kw^2)(x_0 u + y_0 v + z_0 w + z_0 w k),$$

$$c_2 = 4(x_0u + y_0v + z_0w + z_0wk)^2 + 2(1 + kw^2)(x_0^2 + y_0^2 + z_0^2 + z_0^2k - r_2^2 - R^2) + 4\left(\frac{r_2R}{r_1}\right)^2 w^2,$$

$$c_1 = 4(x_0u + y_0v + z_0w + z_0wk)(x_0^2 + y_0^2 + z_0^2 + z_0^2k - r_2^2 - R^2) + 8\left(\frac{r_2R}{r_1}\right)^2 z_0w,$$

$$c_0 = (x_0^2 + y_0^2 + z_0^2 + z_0^2k - r_2^2 - R^2)^2 - 4r_2^2R^2 + 4\left(\frac{r_2R}{r_1}\right)^2 z_0^2,$$

$$k = \frac{r_2^2 - r_1^2}{r_1^2}.$$

Because Eq. (4.12) is a quadric polynomial function of  $t$ , numerical methods are used for solutions, resulting four roots. Only two roots are of interest; the other two indicating the intersection points on the other half torus if the roots are real-valued. After  $t$ 's are solved, the laser dots projected sampled from the laser ring can be plotted for the verification purpose of Eq. (4.9), Eq. (4.10) and Eq. (4.11). Note that in the derivation of Eq. (4.11), the coordinates transformation is taken with  $\theta_v$  being first applied. Such a sequential style coordinates transformation results in an implicit roll angle, which is described in Eq. (5.17) and is to be discussed in detail in Section 5.2.2. Figure 4.15 shows two test cases; all laser dots closely match the theoretical derivations.

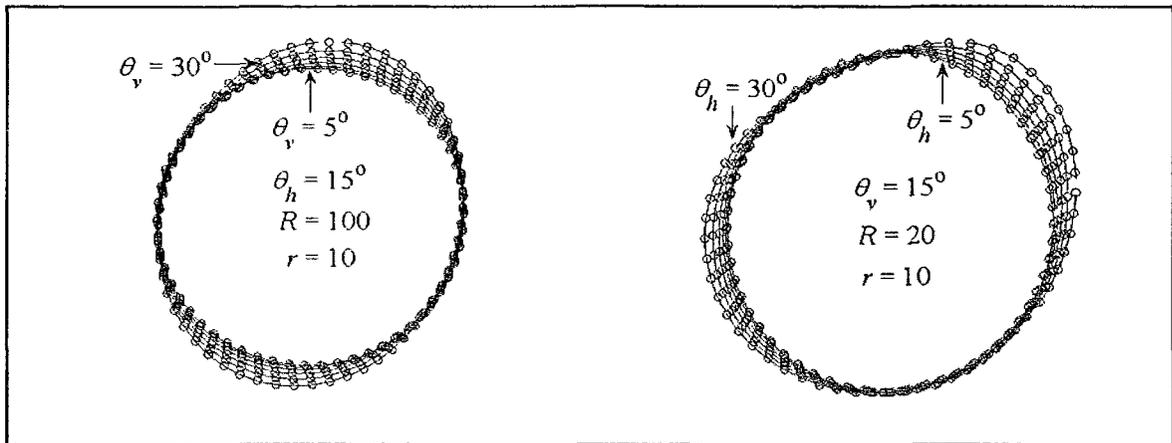


Figure 4.15 Verification of Oblique Shapes in Bend by Laser Dots Simulation

### 4.3 Chapter Summary

This chapter models the measurement origin offsets in a circular cylindrical pipe, caused by pipe deformation or by robotic pitch. The information on the position of the scanner's origin is critical for later-on sensor fusion. The offset is related to external dimensional characteristic such as the deformation indicator and the pitch angle; the offset is also related to inherent robotic dimensional characteristics such as wheels' span, wheels' radius, and robotic length. This chapter also simulates the distorted geometries in a bend for a 2-D scanner. One example application of such modeling can be the quantification of the erroneous ovality introduced by robotic deviation, especially in a bend with a modest curvature caused by loadings, using the average method and the integral method discussed in Chapter 3. APPENDIX B gives some of the MATLAB® code for this chapter.

## CHAPTER 5

### ROBOTIC ORIENTATION AND PIPE GEOMETRY CALCULATION

It is certainly beneficial to laser scan a pipeline immediately after installation or at an early stage of service, not only because there are little or no mineral deposits that can complicate the inspection task, but because such a database is very important as an anchor for the future database comparisons. It is not rare for a pipe segment to develop ovality in the process of transportation and installation. It is also quite possible that a pipe segment, especially a short one, deforms uniformly along the length when the external loading condition does not have a gradient over such a span. In the case of a uniform pipe segment with elliptical cross sections, an elliptical cylinder is adequate for fitting purposes. Robotic deviation causes the laser ring to measure the interior surface on a non-perpendicular plane relative to the pipe axis, therefore yielding oblique cross sectional geometries, which are still ellipses, but with different sizes or orientations.

This chapter looks into the three shape parameters, namely the semi-major axis length  $r_{major}$ , the semi-minor axis length  $r_{minor}$  and the angle  $\beta$  between the major axis and the  $z$ -axis, of the oblique cross sectional measurement caused by the pitch angle  $\theta_v$ , the yaw angle  $\theta_h$  and the roll angle  $\theta_r$ . Then, a new strategy is proposed for the robot-pipe relationship calculation based on data rings from a single-ended 2-D scanner.

The main goal is to recover the true semi-major axis  $r_1$  and semi-minor axis  $r_2$  of the perpendicular cross section. Throughout this chapter,  $r_1 = 11$  and  $r_2 = 10$  are used.

### 5.1 Oblique Shapes by Robotic Pitch or Yaw or Roll

In general,  $\theta_v$  is always associated with a vertical measurement origin offset unless the terrain happens to hold the origin to stay coincident with the pipe's centerline. Pitch-induced measurement origin offsets are explored in Section 4.1.2 and only  $\theta_v$  is of interest here. Pure  $\theta_v$  happens as both of the front wheels or both of the rear wheels climb over a bump.  $\theta_v$  changes the axis length in the vertical direction in the camera's coordinates system and has no effect on the horizontal axis. If  $\theta_v$  is small, the following relationship is valid:  $r_{major} = r_1$ ,  $r_{minor} = r_2 / \cos \theta_v$ ,  $\beta = 0$ . As  $\theta_v$  increases, the ovality of the scanned shape becomes smaller. When  $r_2 / \cos \theta_v = r_1$ , i.e., the major and minor axes are equal, a circle forms. With  $\theta_v$  continuing to grow,  $r_{minor}$  exceeds  $r_{major}$  and the two switch roles;  $\beta$  also jump from  $0^\circ$  to  $90^\circ$ , i.e.,  $r_{major} = r_2 / \cos \theta_v$ ,  $r_{minor} = r_1$ ,  $\beta = \pi/2$ .

Like  $\theta_v$ ,  $\theta_h$  is always associated with a horizontal measurement origin offset. Pure  $\theta_h$  is unlikely to occur unless the terrain is flat or happens to hold the origin to stay coincident with the pipe's centerline. Rough terrain gives random disturbance to the wheel and changes the robotic orientation.  $\theta_h$  changes the axis length in the horizontal direction in the camera's coordinates system and has no effect on the vertical axis. Regardless of the magnitude of  $\theta_h$ , the following relationship is valid:  $r_{major} = r_1 / \cos \theta_h$ ,  $r_{minor} = r_2$ ,  $\beta = 0$ .

Like  $\theta_v$  and  $\theta_h$ ,  $\theta_r$  is always associated with a measurement origin offset, both vertically and horizontally, unless the robot manages to retain its parallelism with the pipe's centerline. Pure  $\theta_r$  happens as both of the left wheels, for example, travel on the same height with the right pair on a different height. Note that one of the left wheels may hang in the air if all four wheels cannot be in the same plane.  $\theta_r$  does not change the axis length but does alter the starting position of each scan, i.e. the angle  $\beta$ . Regardless of the magnitude of  $\theta_r$ , the following relationship is valid:  $r_{major} = r_1$ ,  $r_{minor} = r_2$ ,  $\beta = -\theta_r$ .

## 5.2 Oblique Shapes by Robotic Pitch and Yaw

Shapes of an oblique laser ring, induced by robotic  $\theta_v$ ,  $\theta_h$  or  $\theta_r$  alone, are discussed in Section 5.1.  $\theta_r$  does not affect the size of the oblique shape and is not considered at this moment. The shape of an oblique laser ring with  $\theta_v$  and  $\theta_h$  together, striking the pipe segment with elliptical cross section is still an ellipse from intuition; however, solutions of semi-major axis  $r_{major}$ , semi-minor axis  $r_{minor}$  and angle  $\beta$  between semi-major and the new  $z_3$ -axis are non-trivial. A theoretical development of  $r_{major}$ ,  $r_{minor}$  and  $\beta$  is developed next with a given pipe geometry and robotic deviation. Then ellipse fitting algorithms, introduced in Section 2.9.1, are applied to the simulated data to verify the correctness of the theoretical derivations.

### 5.2.1 Derivation of Semi Axes

As shown in Figure 0.1, the robot points along the unit vector  $(a, b, c)$  defined by  $\theta_v$  and  $\theta_h$ . Because  $\tan \theta_v = b/a$  and  $\tan \theta_h = c/a$ , it is easy to derive Eq. (5.1), Eq. (5.2) and Eq. (5.3):

$$a = \sqrt{1 + (\tan \theta_v)^2 + (\tan \theta_h)^2}, \quad (5.1)$$

$$b = (\tan \theta_v) \sqrt{1 + (\tan \theta_v)^2 + (\tan \theta_h)^2}, \quad (5.2)$$

$$c = (\tan \theta_h) \sqrt{1 + (\tan \theta_v)^2 + (\tan \theta_h)^2}. \quad (5.3)$$

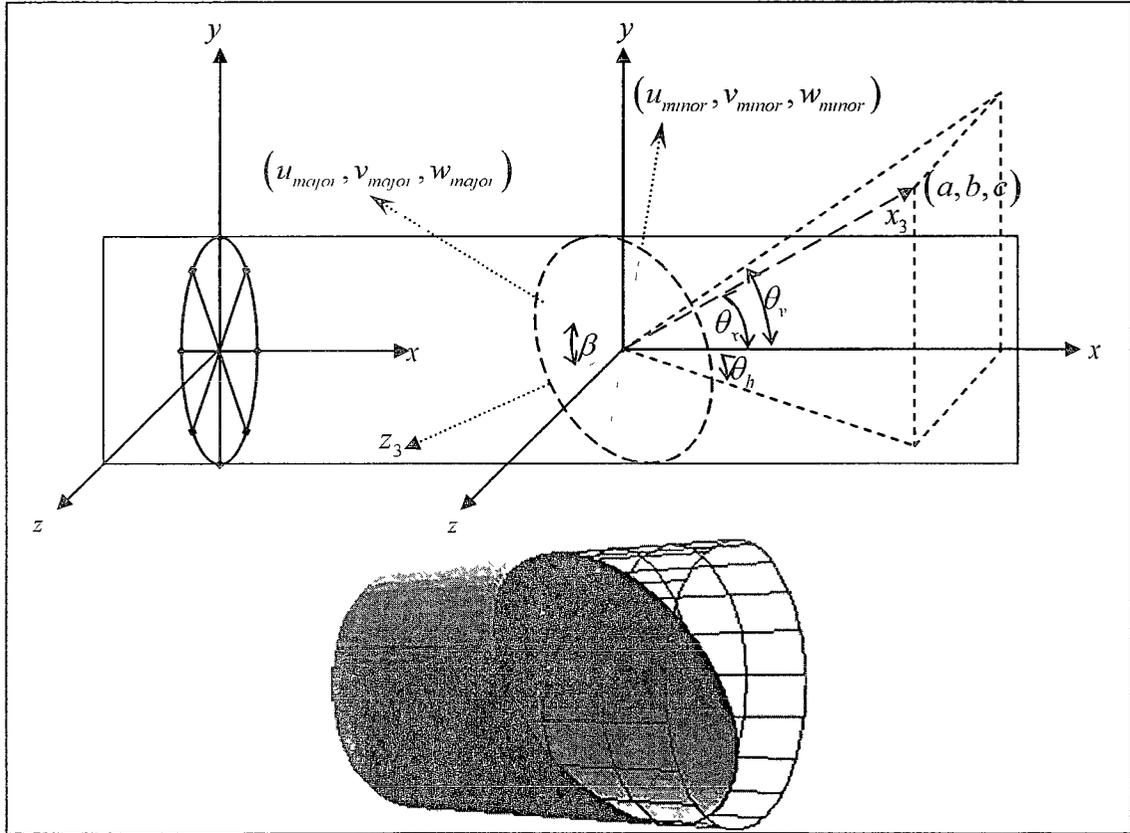


Figure 0.1 Coordinates System Setup for Robotic Pitch and Yaw

A laser beam emits from the measurement origin and strikes on a spot with a unit vector  $(u, v, w)$ . The major axis of the ellipse is in the direction of the laser beam with the largest length determined by  $(u, v, w)$ . In order to find this particular set of  $(u, v, w)$ , the parametric form of a 3-D line is used, as shown in Eq. (5.4), Eq. (5.5) and Eq. (5.6), and it is assumed that there is no measurement origin offset:

$$x = x_0 + ut = ut, \quad (5.4)$$

$$y = y_0 + vt = vt, \quad (5.5)$$

$$z = z_0 + wt = wt. \quad (5.6)$$

The ellipse function, Eq. (3.1), shall also be satisfied. By substituting of Eq. (5.5) and Eq. (5.6) into Eq. (3.1),

$$(r_1^2 v^2 + r_2^2 w^2) t^2 = r_1^2 r_2^2,$$

$$t_{1,2} = \pm \frac{r_1 r_2}{\sqrt{r_1^2 v^2 + r_2^2 w^2}}.$$

Therefore, the length of the laser beam is expressed as Eq. (5.7):

$$\frac{|t_1 - t_2|}{2} = \frac{r_1 r_2}{\sqrt{r_1^2 v^2 + r_2^2 w^2}}. \quad (5.7)$$

So to find the major and minor axis, the minimum and maximum of  $r_1^2 v^2 + r_2^2 w^2$  shall be solved. Because  $(u, v, w)$  is a unit vector in a plane perpendicular to  $(a, b, c)$ , Eq. (5.8) and Eq. (5.9) can be obtained as:

$$ua + vb + wc = 0, \quad (5.8)$$

$$u^2 + v^2 + w^2 = 1. \quad (5.9)$$

Solving  $u$  from Eq. (5.8) as Eq. (5.10),

$$u = -\frac{vb + wc}{a}. \quad (5.10)$$

By substituting Eq. (5.10) into Eq. (5.9),

$$\frac{(vb + wc)^2}{a^2} + v^2 + w^2 = 1.$$

Solving  $v$  in terms of  $w$ ,

$$v^2 b^2 + w^2 c^2 + 2vwbc + v^2 a^2 + w^2 a^2 = a^2,$$

$$(a^2 + b^2)v^2 + (2wbc)v + (a^2 + c^2)w^2 - a^2 = 0,$$

$$v = \frac{-wbc \pm \sqrt{-(a^4 + a^2 b^2 + a^2 c^2)w^2 + (a^2 + b^2)a^2}}{a^2 + b^2}.$$

Because  $a^2 + b^2 + c^2 = 1$ , it can be obtained that

$$v = \frac{-wbc \pm a\sqrt{-w^2 + a^2 + b^2}}{a^2 + b^2},$$

and  $-w^2 + a^2 + b^2 \geq 0$ .

Only the positive sign for  $v$  is needed because laser beams in opposite directions relative to the origin produce the same effect, so  $v$  can be obtained as Eq. (5.11):

$$v = \frac{-wbc + a\sqrt{-w^2 + a^2 + b^2}}{a^2 + b^2}. \quad (5.11)$$

By substituting Eq. (5.11) into Eq. (5.10),  $u$  can be obtained as Eq. (5.12):

$$u = \frac{-wac - b\sqrt{-w^2 + a^2 + b^2}}{a^2 + b^2}. \quad (5.12)$$

Defining  $f$  as:

$$\begin{aligned}
f &= r_1^2 v^2 + r_2^2 w^2 \\
&= r_1^2 \left( \frac{-wbc + a\sqrt{-w^2 + a^2 + b^2}}{a^2 + b^2} \right)^2 + r_2^2 w^2 \\
&= \frac{r_1^2}{(a^2 + b^2)^2} \left[ w^2 b^2 c^2 + a^2 (-w^2 + a^2 + b^2) - 2abcw\sqrt{-w^2 + a^2 + b^2} \right] + r_2^2 w^2 \\
&= \left[ \frac{r_1^2 (b^2 c^2 - a^2)}{(a^2 + b^2)^2} + r_2^2 \right] w^2 - \frac{2abc r_1^2}{(a^2 + b^2)^2} w\sqrt{-w^2 + a^2 + b^2} + \frac{r_1^2 a^2}{a^2 + b^2}.
\end{aligned}$$

For compactness, define the following four constants:

$$A = \left[ \frac{r_1^2 (b^2 c^2 - a^2)}{(a^2 + b^2)^2} + r_2^2 \right],$$

$$B = -\frac{2abc r_1^2}{(a^2 + b^2)^2},$$

$$C = a^2 + b^2,$$

$$D = \frac{r_1^2 a^2}{a^2 + b^2}.$$

So  $f$  can be rewritten as  $f = Aw^2 - Bw\sqrt{-w^2 + C} + D$ . In order to find the maximum and minimum values of  $f$ , the derivative of  $f$  is set to be zero, i.e.,

$$\frac{df}{dw} = 2Aw + \frac{B(c - 2w^2)}{\sqrt{-w^2 + C}} = 0.$$

Because  $-w^2 + C = -w^2 + a^2 + b^2 \geq 0$ , it can be obtained that

$$2Aw\sqrt{-w^2 + C} = B(2w^2 - C).$$

Taking the square of each side,

$$4A^2w^2(-w^2 + C) = B^2(2w^2 - C)^2,$$

$$4(A^2 + B^2)w^4 - 4C(A^2 + B^2)w^2 + B^2C^2 = 0.$$

Defining  $x = w^2$ , then

$$4(A^2 + B^2)x^2 - 4C(A^2 + B^2)x + B^2C^2 = 0.$$

For compactness, define the following three constants:

$$E = 4(A^2 + B^2),$$

$$F = -2C(A^2 + B^2),$$

$$G = B^2C^2.$$

So the equation can be rewritten as  $4Ex^2 + 2Fx + G = 0$  and be solved as

$$x_1 = \frac{-F + \sqrt{F^2 - EG}}{E},$$

$$x_2 = \frac{-F - \sqrt{F^2 - EG}}{E}.$$

Since  $C \geq 0$ , then  $F \leq 0$ . Since  $E \geq 0$ ,  $G \geq 0$  and  $0 \leq \sqrt{F^2 - EG} \leq -F$ , it is true that

$$0 \leq x_2 \leq x_1.$$

Because  $x = w^2$ , there are four solutions for  $w$ , only two of which are true solutions. To find the true solutions, a plot of  $f$  versus  $w$  is shown in Figure 0.2. For an illustrative purpose,  $\theta_v$  varies from  $0^\circ$  to  $45^\circ$  with a  $15^\circ$  increment and  $\theta_h$  is  $15^\circ$ . There are two distinct global extreme values that are annotated by up and down arrows. Except

for a  $0^\circ \theta_v$  ( $\theta_h$  only), global extreme values do not occur at boundary limits, meaning that local and global extreme values happen to be the same.

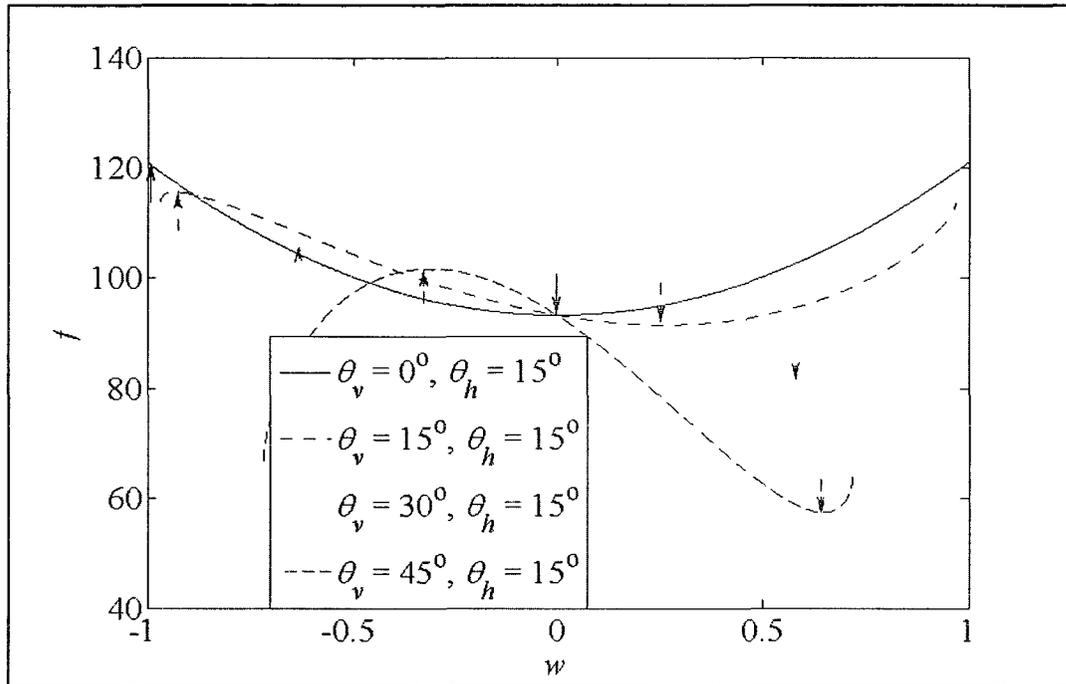


Figure 0.2 Global Extremes Location of W Vector with Varying Pitch and Yaw

Now it is clear that the location ( $w$ ) of local minimum and maximum of  $f$ , which corresponds to the major and minor axis respectively, varies with  $\theta_v$  and  $\theta_h$ . Because  $0 \leq x_2 \leq x_1$ , for the dotted ( $\theta_v = 30^\circ, \theta_h = 15^\circ$ ) and dash-dotted curve ( $\theta_v = 15^\circ, \theta_h = 15^\circ$ ), the following solutions are correct:

$$w_{major} = \sqrt{x_2} = \sqrt{\frac{-F - \sqrt{F^2 - EG}}{E}},$$

$$w_{minor} = -\sqrt{x_1} = \sqrt{\frac{-F + \sqrt{F^2 - EG}}{E}}.$$

while for the dashed curve ( $\theta_v = 45^\circ$ ,  $\theta_h = 15^\circ$ ), the following solutions are correct:

$$w_{major} = \sqrt{x_1} = \sqrt{\frac{-F + \sqrt{F^2 - EG}}{E}},$$

$$w_{minor} = -\sqrt{x_2} = -\sqrt{\frac{-F - \sqrt{F^2 - EG}}{E}}.$$

After  $w_{major}$  and  $w_{minor}$  are found, the direction triplets  $(u_{major}, v_{major}, w_{major})$  and  $(u_{minor}, v_{minor}, w_{minor})$  can be solved using Eq. (5.11) and Eq. (5.12). The semi-major and semi-minor axis length can be found with substitutions of  $(u_{major}, v_{major}, w_{major})$  and  $(u_{minor}, v_{minor}, w_{minor})$  into Eq. (5.7), as expressed in Eq. (5.13) and Eq. (5.14):

$$r_{major} = \frac{r_1 r_2}{\sqrt{r_1^2 v_{major}^2 + r_2^2 w_{major}^2}}, \quad (5.13)$$

$$r_{minor} = \frac{r_1 r_2}{\sqrt{r_1^2 v_{minor}^2 + r_2^2 w_{minor}^2}}. \quad (5.14)$$

### 5.2.2 Derivation of Angle

Because of  $\theta_v$  and  $\theta_h$ , the horizontal axis in the oblique plane is no longer the  $z$ -axis. To find the angle  $\beta$ , the vector for the new  $z_3$ -axis is needed. The new  $x_3$ -axis, denoted by  $(a, b, c)$ , is still perpendicular to the  $z_3$ -axis. It can be pictured as a rigid body rotation that transforms  $x$  into  $x_3$  and  $z$  into  $z_3$  simultaneously.

In order to carry out such a transformation, the axis of rotation  $\mathbf{l}$  is to be found as  $\mathbf{l} = \mathbf{x} \times \mathbf{x}_3$ . Since the angle between  $x$ -axis and  $x_3$ -axis is  $\theta_x = \cos^{-1} a$ , the  $x \rightarrow x_3$

coordinates transformation is a counter-clockwise rotation of  $\theta_x$  about the axis

$(1, 0, 0)^T \times (a, b, c)^T = (0, -c, b)^T$  with vector normalization, as expressed in Eq. (5.15):

$$\mathbf{l} = (l_x, l_y, l_z)^T = \left( 0, -\frac{c}{\sqrt{b^2 + c^2}}, \frac{b}{\sqrt{b^2 + c^2}} \right)^T. \quad (5.15)$$

It is known that the matrix  $\mathbf{R}$  for a rotation about an arbitrary axis  $\mathbf{l}$  can be stated as:

$$\mathbf{R} = \begin{bmatrix} \cos \theta_x + (1 - \cos \theta_x) l_x^2 & (1 - \cos \theta_x) l_y l_x - (\sin \theta_x) l_z & (1 - \cos \theta_x) l_z l_x + (\sin \theta_x) l_y \\ (1 - \cos \theta_x) l_x l_y + (\sin \theta_x) l_z & \cos \theta_x + (1 - \cos \theta_x) l_y^2 & (1 - \cos \theta_x) l_z l_y - (\sin \theta_x) l_x \\ (1 - \cos \theta_x) l_x l_z - (\sin \theta_x) l_y & (1 - \cos \theta_x) l_y l_z + (\sin \theta_x) l_x & \cos \theta_x + (1 - \cos \theta_x) l_z^2 \end{bmatrix}.$$

Note that  $\mathbf{z} = (0, 0, 1)^T$  and by substituting Eq. (5.15) into  $\mathbf{R}$ ,

$$\mathbf{z}_3 = \mathbf{R} \cdot \mathbf{z} = \mathbf{R} \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} = \begin{bmatrix} (1 - \cos \theta_x) l_z l_x + (\sin \theta_x) l_y \\ (1 - \cos \theta_x) l_z l_y - (\sin \theta_x) l_x \\ \cos \theta_x + (1 - \cos \theta_x) l_z^2 \end{bmatrix} = \begin{pmatrix} \frac{(\sqrt{1-a^2})c}{\sqrt{b^2+c^2}} \\ -\frac{(1-a)bc}{b^2+c^2} \\ a + \frac{(1-a)b^2}{b^2+c^2} \end{pmatrix}.$$

After  $\mathbf{z}_3$  is solved, angle  $\beta$  can be calculated as Eq. (5.16):

$$\beta = \cos^{-1} \left[ (u, v, w) \left( -\frac{(\sqrt{1-a^2})c}{\sqrt{b^2+c^2}}, -\frac{(1-a)bc}{b^2+c^2}, a + \frac{(1-a)b^2}{b^2+c^2} \right) \right]. \quad (5.16)$$

There is also an implicit roll angle  $\theta_r$  depending on the pathway that the robot takes to reach the final orientation, shown in Figure 0.3. ( $xyz$ : original coordinates system;

$x_1y_1z_1$  : coordinates system after  $\theta_1$  ;  $x_2y_2z_2$  : coordinates system after  $\theta_h$  ;  $x_3y_3z_3$  : coordinates system after direct rotation about the axis of rotation  $\mathbf{l}$ ).

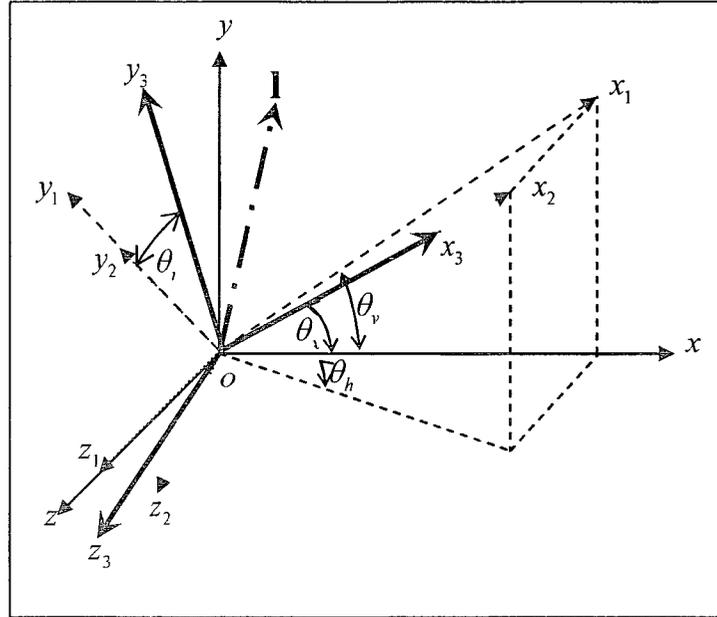


Figure 0.3 Implicit Roll by Direct and Stepwise Rotation

If a direct rotation about  $\mathbf{l}$  is taken, then the resulting  $y_3$ -axis can be calculated

as:

$$\mathbf{y}_3 = \mathbf{R} \cdot \mathbf{y} = \mathbf{R} \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} = \begin{bmatrix} (1 - \cos \theta_x) l_y l_x - (\sin \theta_x) l_z \\ \cos \theta_x + (1 - \cos \theta_x) l_y^2 \\ (1 - \cos \theta_x) l_y l_z + (\sin \theta_x) l_x \end{bmatrix} = \begin{pmatrix} \frac{(\sqrt{1-a^2})b}{\sqrt{b^2+c^2}} \\ a + \frac{(1-a)c^2}{b^2+c^2} \\ -\frac{(1-a)bc}{b^2+c^2} \end{pmatrix}.$$

While using a stepwise fashion, for example,  $\theta_v$  transforms the coordinates system to  $x_1y_1z_1$  and then  $\theta_h$  follows to transform the coordinates system to  $x_2y_2z_2$ . The resulting

$y_2$ -axis can be calculated as  $\mathbf{y}_2 = (-\sin \theta_v, \cos \theta_v, 0)$ . Therefore, the implicit roll angle  $\theta_r$  is the angle between  $y_2$  and  $y_3$ , expressed as Eq. (5.17),

$$\theta_r = \cos^{-1} \left[ (-\sin \theta_v, \cos \theta_v, 0) \left( -\frac{(\sqrt{1-a^2})b}{\sqrt{b^2+c^2}}, a + \frac{(1-a)c^2}{b^2+c^2}, -\frac{(1-a)bc}{b^2+c^2} \right) \right]. \quad (5.17)$$

### 5.2.3 Verification

It is necessary to make sure that Eq. (5.13), Eq. (5.14) and Eq. (5.16) derived in Section 5.2.2 are correct. First, oblique cross sectional data points need to be generated with different oblique angles. Recall that in Figure 0.1, in order to get the new  $z_3$ -axis, a rotation matrix  $\mathbf{R}$  is applied to the old  $z$ -axis. To generate the circumferential data for an oblique laser ring, all vectors in the old vertical plane shall be transformed back by  $\mathbf{R}$ .

Algebraic ellipse fitting techniques, as discussed in Section 2.9.1, are fitted to the data points. The resulting parameters of an algebraic fitting algorithm are in the form of  $a_e z^2 + b_e zy + c_e y^2 + d_e z + e_e y + f_e = 0$ . A conversion from the fitted parameters  $a_e, b_e, c_e, d_e, e_e, f_e$  to  $r_{major\_est}, r_{minor\_est}$  and  $\beta_{est}$  can be derived as:

$$r_{major\_est} = \frac{\sqrt{2(a_e f_e^2 + c_e d_e^2 + g_e b_e^2 - 2b_e d_e f_e - a_e c_e g_e)}}{\sqrt{(b_e^2 - a_e c_e) \left[ \sqrt{(a_e - c_e)^2 + 4b_e^2} - a_e - c_e \right]}},$$

$$r_{minor\_est} = \frac{\sqrt{2(a_e f_e^2 + c_e d_e^2 + g_e b_e^2 - 2b_e d_e f_e - a_e c_e g_e)}}{\sqrt{(b_e^2 - a_e c_e) \left[ -\sqrt{(a_e - c_e)^2 + 4b_e^2} - a_e - c_e \right]}}$$

$$\beta_{est} = \begin{cases} 0 & b_e = 0, a_e < c_e \\ \frac{\pi}{2} & b_e = 0, a_e > c_e \\ \frac{\cot^{-1}\left(\frac{a_e - c_e}{2b_e}\right)}{2} & b_e \neq 0, a_e < c_e \\ \frac{\pi}{2} + \frac{\cot^{-1}\left(\frac{a_e - c_e}{2b_e}\right)}{2} & b_e \neq 0, a_e > c_e \end{cases}.$$

The estimated  $r_{major\_est}$ ,  $r_{minor\_est}$  and  $\beta_{est}$  are plotted together with the theoretical solutions for verification. It is assumed that  $r_1 = 12$  and  $r_2 = 10$ . In the first two rows of Figure 0.4,  $\theta_h$  is set to be constant ( $0^\circ$ ,  $15^\circ$ ,  $30^\circ$ ) with  $\theta_v$  varying from  $0^\circ$  to  $30^\circ$ ; in the next two rows,  $\theta_v$  is set to be constant ( $0^\circ$ ,  $15^\circ$ ,  $30^\circ$ ) with  $\theta_h$  varying from  $0^\circ$  to  $30^\circ$ . The parameter estimations fit well with the theoretical solutions for all cases, so the correctness of the theoretical development is validated. 3-D plots with contours of  $r_{major}$ ,  $r_{minor}$  and  $\beta$  versus  $\theta_v$  and  $\theta_h$  are shown in the last row. It can be seen that  $r_{major}$  varies greater with  $\theta_h$  than  $\theta_v$ ;  $r_{minor}$  varies greater with  $\theta_v$  than  $\theta_h$ ; and  $\beta$ , initially smooth with  $\theta_v$  and  $\theta_h$ , starts to increase dramatically at:

$$\theta_v = \cos^{-1}\left(\frac{r_2}{r_1}\right)\frac{180^\circ}{\pi} = \cos^{-1}\left(\frac{10}{11}\right)\frac{180^\circ}{\pi} \approx 25^\circ.$$

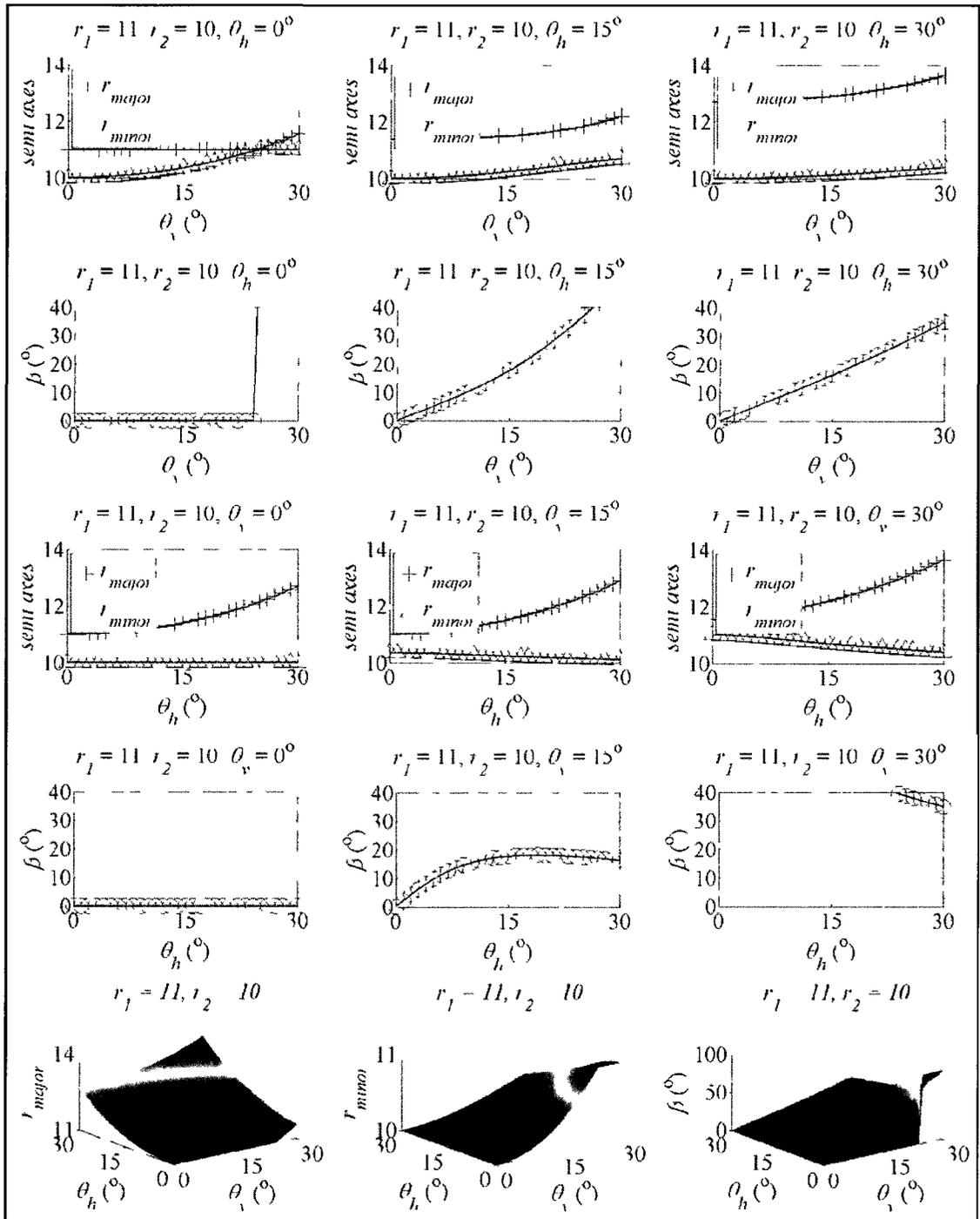


Figure 0.4 Verification of Theoretical Solutions for Oblique Ellipse

### 5.3 Pipe Geometry Calculation with Known Robotic Orientation

INS sensors, as discussed in Section 2.5, do not exhibit severe error accumulation problem unless being used for a long time without zero updating. Since the pipes of concern are straight, the pipe and the earth can be thought of sharing the same coordinates system. Therefore, any  $\theta_v$ ,  $\theta_h$  and  $\theta_r$  relative to the earth detected by the sensors also indicate the robotic deviation relative to the pipe. The feedback from sensors' shall be trustworthy for a certain length of pipe segments. If such a confident reading is achievable, the semi axes of pipe can be deduced. Using the same letter assignments as in Section 5.2, now the known values are  $r_{major}$ ,  $r_{minor}$ ,  $\beta$ ,  $\theta_v$ ,  $\theta_h$  and the values to seek are  $r_1$  and  $r_2$ .

Reproducing Eq. (5.11), Eq. (5.12), and Eq. (5.16) as Eq. (5.18), Eq. (5.19), and Eq. (5.20):

$$v = \frac{-wbc + a\sqrt{-w^2 + a^2 + b^2}}{a^2 + b^2}, \quad (5.18)$$

$$u = \frac{-wac - b\sqrt{-w^2 + a^2 + b^2}}{a^2 + b^2}, \quad (5.19)$$

$$-\frac{(\sqrt{1-a^2})c}{\sqrt{b^2+c^2}}u - \frac{(1-a)bc}{b^2+c^2}v + \left[ a + \frac{(1-a)b^2}{b^2+c^2} \right]w = \cos \beta. \quad (5.20)$$

By substituting Eq. (5.18) and Eq. (5.19) into Eq. (5.20),

$$\begin{aligned} & \frac{(\sqrt{1-a^2})c}{\sqrt{b^2+c^2}} \frac{-wac - b\sqrt{-w^2 + a^2 + b^2}}{a^2 + b^2} - \frac{(1-a)bc}{b^2+c^2} \frac{-wbc + a\sqrt{-w^2 + a^2 + b^2}}{a^2 + b^2} \\ & + \left[ a + \frac{(1-a)b^2}{b^2+c^2} \right]w = \cos \beta. \end{aligned}$$

Rearranging terms according to  $w$  and eliminating the denominators,

$$\begin{aligned} & \left[ \sqrt{b^2 + c^2} \sqrt{1 - a^2} bc - (1 - a) abc \right] \sqrt{-w^2 + a^2 + b^2} \\ & + \left[ \sqrt{b^2 + c^2} \sqrt{1 - a^2} ac^2 + (1 - a) b^2 c^2 + (a^2 + b^2)(ac^2 + b^2) \right] w = (a^2 + b^2)(b^2 + c^2) \cos \beta. \end{aligned}$$

Because  $a^2 + b^2 + c^2 = 1$ , it can be obtained that

$$\begin{aligned} & \left[ \sqrt{1 - a^2} \sqrt{1 - a^2} bc - (1 - a) abc \right] \sqrt{-w^2 + a^2 + b^2} \\ & + \left[ \sqrt{1 - a^2} \sqrt{1 - a^2} ac^2 + (1 - a) b^2 c^2 + (a^2 + b^2)(ac^2 + b^2) \right] w = (a^2 + b^2)(b^2 + c^2) \cos \beta. \end{aligned}$$

Rearranging terms according to  $w$ ,

$$(bc - abc) \sqrt{-w^2 + a^2 + b^2} + (ac^2 + b^2) w = (a^2 + b^2)(b^2 + c^2) \cos \beta.$$

For compactness, define four constants as in Eq. (5.21), Eq. (5.22), Eq. (5.23), Eq. (5.24):

$$A = bc - abc, \quad (5.21)$$

$$B = ac^2 + b^2, \quad (5.22)$$

$$C = a^2 + b^2, \quad (5.23)$$

$$D = (a^2 + b^2)(b^2 + c^2) \cos \beta. \quad (5.24)$$

So the equation can be rewritten as  $A\sqrt{-w^2 + C} = -Bw + D$ . Taking squares on each side and rearranging terms,  $(A^2 + B^2)w^2 - 2BDw + D^2 - A^2C = 0$ . For compactness, define three constants as expressed in Eq. (5.25), Eq. (5.26), and Eq. (5.27),

$$E = A^2 + B^2, \quad (5.25)$$

$$F = -BD, \quad (5.26)$$

$$G = D^2 - A^2C, \quad (5.27)$$

where  $D = (a^2 + b^2)(b^2 + c^2)\cos\beta$  and  $D = -(a^2 + b^2)(b^2 + c^2)\sin\beta$  for the major and minor axis, respectively. Then the equation can be rewritten as:

$$Ew^2 + 2Fw + G = 0.$$

Solving the equation,

$$w = \frac{-F \pm \sqrt{F^2 - EG}}{E}.$$

To determine whether a plus or minus sign shall be used, it is helpful to plot the function  $f(w) = A\sqrt{-w^2 + C} + Bw - D$  versus some values of  $A, B, C, D$ .  $\theta_h = 45^\circ$  and  $\theta_v = 22.5^\circ$  are used to calculate  $a, b, c$  as in Eq. (5.1), Eq. (5.2) and Eq. (5.3) Then  $A, B, C, D$  is calculated as in Eq. (5.21), Eq. (5.22), Eq. (5.23), and Eq. (5.24). Also,  $E, F, G$  is calculated as in Eq. (5.25), Eq. (5.26), and Eq. (5.27).  $f(w)$  is plotted as the solid line and solutions for  $w$  using both plus and minus signs are also indicated in Figure 0.5. The minus sign gives the correct solutions for  $w_{major}$  and  $w_{minor}$ ; therefore,  $w$  can be obtained as:

$$w = \frac{-F - \sqrt{F^2 - EG}}{E}.$$

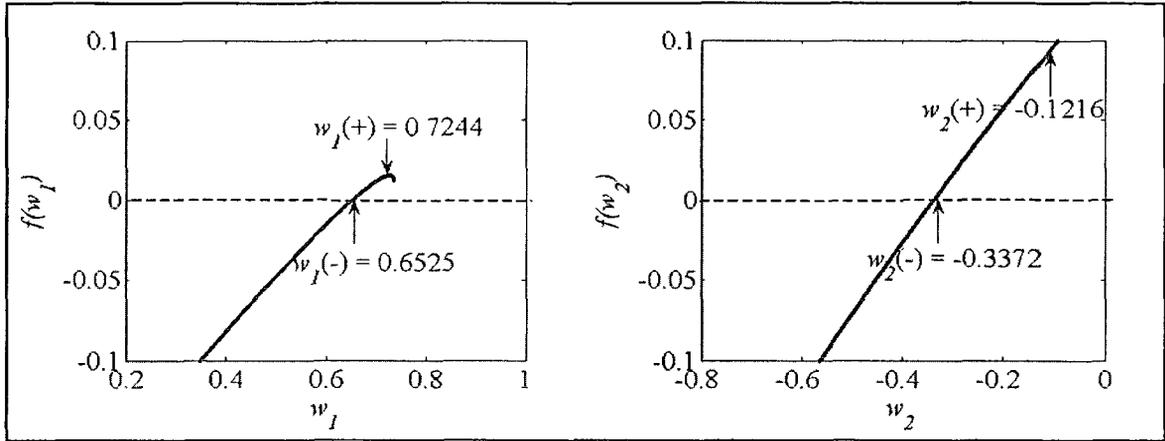


Figure 0.5 Determination of Solution Sign for Extreme  $w$  Vectors

Rewriting Eq. (5.13) and Eq. (5.14) as:

$$\frac{v_{major}^2}{r_2^2} + \frac{w_{major}^2}{r_1^2} = \frac{1}{r_{major}^2},$$

$$\frac{v_{minor}^2}{r_2^2} + \frac{w_{minor}^2}{r_1^2} = \frac{1}{r_{minor}^2}.$$

Rearranging into a matrix form as expressed in Eq. (5.28):

$$\begin{pmatrix} v_{major}^2 & w_{major}^2 \\ v_{minor}^2 & w_{minor}^2 \end{pmatrix} \begin{pmatrix} \frac{1}{r_2^2} \\ \frac{1}{r_1^2} \end{pmatrix} = \begin{pmatrix} \frac{1}{r_{major}^2} \\ \frac{1}{r_{minor}^2} \end{pmatrix}, \quad (5.28)$$

where  $(v_{major}, w_{major})$  and  $(v_{minor}, w_{minor})$  correspond to the major and minor axis of the oblique ellipse, respectively. Therefore, there is a system of linear equations for  $1/r_2^2$  and  $1/r_1^2$ , where  $r_1$  and  $r_2$  can be derived. Different  $\theta_v$ 's and  $\theta_h$ 's are tested; the resulting estimations  $r_{1\_est}$  and  $r_{2\_est}$  are all equal to their true value:  $r_1 = 11$  and  $r_2 = 10$ .

### 5.4 Pipe Geometry Calculation with Unknown Robotic Orientation

After INS sensors have been used for a long time, the recorded orientation is not always trustworthy. Some means for resetting the starting orientation for the double integration, as introduced in the INS sensors in Section 2.5, is needed to reduce divergence. Using the same letter assignments as in Section 5.2, now the known values are  $r_{major}$ ,  $r_{minor}$ ,  $\beta$  and the values to seek are  $\theta_v$ ,  $\theta_h$ ,  $r_1$  and  $r_2$ .

Eq. (5.28) is no longer a system of linear equations and shall be rewritten as Eq. (5.29):

$$\begin{bmatrix} v_{major}(a, b, c, \beta)^2 & w_{major}(a, b, c, \beta)^2 \\ v_{minor}(a, b, c, \beta)^2 & w_{minor}(a, b, c, \beta)^2 \end{bmatrix} \begin{pmatrix} \frac{1}{r_2^2} \\ \frac{1}{r_1^2} \end{pmatrix} = \begin{pmatrix} \frac{1}{r_{major}^2} \\ \frac{1}{r_{minor}^2} \end{pmatrix}. \quad (5.29)$$

Because  $a, b, c, \beta, r_{major}, r_{minor}$  are all functions of  $\theta_v$  and  $\theta_h$ , so substituting  $a = a(\theta_v, \theta_h)$ ,  $b = b(\theta_v, \theta_h)$ ,  $c = c(\theta_v, \theta_h)$ ,  $\beta = \beta(\theta_v, \theta_h)$ ,  $r_{major} = r_{major}(\theta_v, \theta_h)$  and  $r_{minor} = r_{minor}(\theta_v, \theta_h)$  into Eq. (5.29), Eq. (5.30) can be obtained:

$$\begin{bmatrix} v_{major}(\theta_v, \theta_h)^2 & w_{major}(\theta_v, \theta_h)^2 \\ v_{minor}(\theta_v, \theta_h)^2 & w_{minor}(\theta_v, \theta_h)^2 \end{bmatrix} \begin{pmatrix} \frac{1}{r_2^2} \\ \frac{1}{r_1^2} \end{pmatrix} = \begin{pmatrix} \frac{1}{r_{major}(\theta_v, \theta_h)^2} \\ \frac{1}{r_{minor}(\theta_v, \theta_h)^2} \end{pmatrix}. \quad (5.30)$$

Recall in Section 5.3, only one ring of measurements is enough to deduce two unknown variables. Now there are four unknown variables; therefore, at least two rings of measurements are needed. Although the current orientation  $\theta_v$  and  $\theta_h$  may not be

trustworthy, the incremental changes  $\Delta\theta_v$  and  $\Delta\theta_h$  in a short distance shall be accurate.

Therefore, Eq. (5.30) can be expressed as Eq. (5.31):

$$\left\{ \begin{array}{l} \frac{v_{major}(\theta_v, \theta_h)^2}{r_2^2} + \frac{w_{major}(\theta_v, \theta_h)^2}{r_1^2} - \frac{1}{r_{major}(\theta_v, \theta_h)^2} = 0 \\ \frac{v_{minor}(\theta_v, \theta_h)^2}{r_2^2} + \frac{w_{minor}(\theta_v, \theta_h)^2}{r_1^2} - \frac{1}{r_{minor}(\theta_v, \theta_h)^2} = 0 \\ \frac{v_{major}(\theta_v + \Delta\theta_v, \theta_h + \Delta\theta_h)^2}{r_2^2} + \frac{w_{major}(\theta_v + \Delta\theta_v, \theta_h + \Delta\theta_h)^2}{r_1^2} - \frac{1}{r_{major}(\theta_v + \Delta\theta_v, \theta_h + \Delta\theta_h)^2} = 0 \\ \frac{v_{minor}(\theta_v + \Delta\theta_v, \theta_h + \Delta\theta_h)^2}{r_2^2} + \frac{w_{minor}(\theta_v + \Delta\theta_v, \theta_h + \Delta\theta_h)^2}{r_1^2} - \frac{1}{r_{minor}(\theta_v + \Delta\theta_v, \theta_h + \Delta\theta_h)^2} = 0 \end{array} \right. \quad (5.31)$$

Highly nonlinearity exists in Eq. (5.31), considering that Eq. (5.1), Eq. (5.2) and Eq. (5.3) involve tangential components, that two of the four unknowns ( $r_1$  and  $r_2$ ) are in the denominators, and that all unknowns are in quadratic orders.

The set of  $(\theta_v, \theta_h, r_1, r_2)$  used to simulate the geometry of the first ring is (15.0, 7.5, 11.0, 10.0). Then the second ring is set as  $\Delta\theta_v = 25.0^\circ$  and  $\Delta\theta_h = 12.5^\circ$ . An initial guess of the solution of  $(\theta_v, \theta_h, r_1, r_2)$  is given as (25.0, 20.0, 12.0, 9.0). Note that even with such a very wild guess, the solver still converges after 25 iterations at the desired solution, as shown in Figure 0.6. A close initial guess (20.0, 10.0, 12.0, 9.0) results in less iterations; the algorithm converges to the desired solution after 6 iterations, as shown in Figure 0.7. “Norm of Step” indicates the leap at each iteration in the parameter space

searching the optimum and “First-Order Optimality” measures how close the current solution is to the optimum.

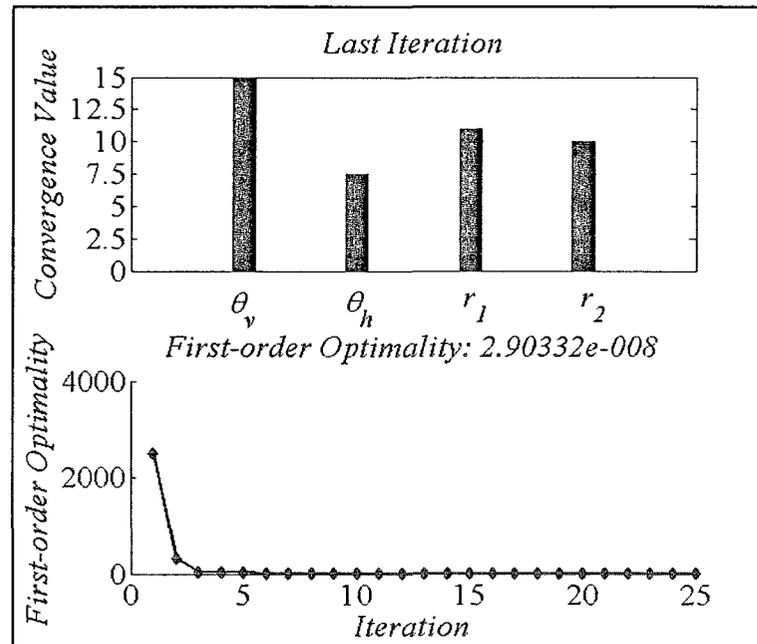


Figure 0.6 Algorithm Performance (Four Unknowns, Wild Initial Guess)

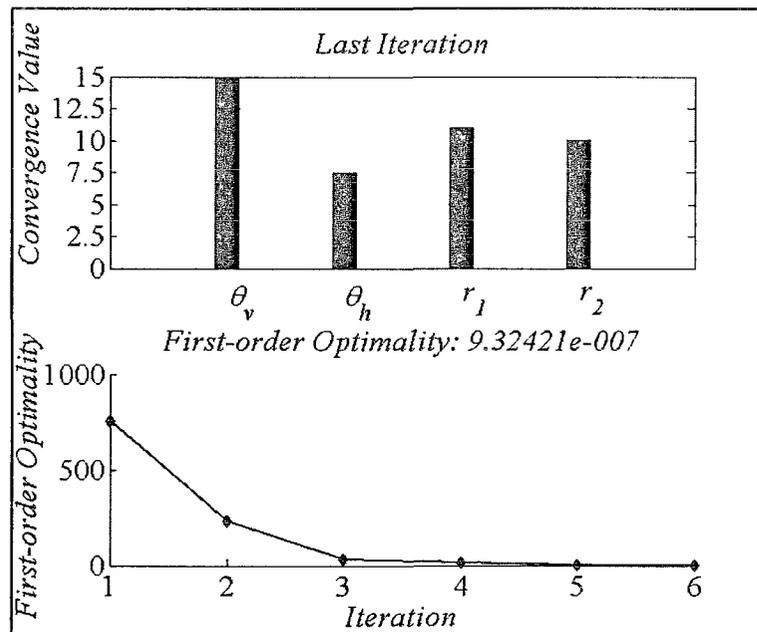


Figure 0.7 Algorithm Performance (Four Unknowns, Close Initial Guess)

Roll  $\theta_r$  can be incorporated in the same way. Now there are five unknown variables; therefore, at least three rings of measurements are needed. Although the current orientation  $\theta_v$ ,  $\theta_h$ , and  $\theta_r$  may not be trustworthy, the incremental changes  $\Delta\theta_v$ ,  $\Delta\theta_h$ ,  $\Delta\theta_r$  and  $\Delta\theta'_v$ ,  $\Delta\theta'_h$ ,  $\Delta\theta'_r$  in a short distance shall be accurate. Any five equations out of the six constraints from the three rings of measurement can be used; one of the combinations is shown as follows:

$$\left\{ \begin{array}{l} \frac{v_{major}(\theta_v, \theta_h, \theta_r)^2}{r_2^2} + \frac{w_{major}(\theta_v, \theta_h, \theta_r)^2}{r_1^2} - \frac{1}{r_{major}(\theta_v, \theta_h)^2} = 0 \\ \frac{v_{minor}(\theta_v, \theta_h, \theta_r)^2}{r_2^2} + \frac{w_{minor}(\theta_v, \theta_h, \theta_r)^2}{r_1^2} - \frac{1}{r_{minor}(\theta_v, \theta_h)^2} = 0 \\ \frac{v_{major}(\theta_v + \Delta\theta_v, \theta_h + \Delta\theta_h, \theta_r + \Delta\theta_r)^2}{r_2^2} + \frac{w_{major}(\theta_v + \Delta\theta_v, \theta_h + \Delta\theta_h, \theta_r + \Delta\theta_r)^2}{r_1^2} - \frac{1}{r_{major}(\theta_v + \Delta\theta_v, \theta_h + \Delta\theta_h)^2} = 0 \\ \frac{v_{minor}(\theta_v + \Delta\theta_v, \theta_h + \Delta\theta_h, \theta_r + \Delta\theta_r)^2}{r_2^2} + \frac{w_{minor}(\theta_v + \Delta\theta_v, \theta_h + \Delta\theta_h, \theta_r + \Delta\theta_r)^2}{r_1^2} - \frac{1}{r_{minor}(\theta_v + \Delta\theta_v, \theta_h + \Delta\theta_h)^2} = 0 \\ \frac{v_{major}(\theta_v + \Delta\theta'_v, \theta_h + \Delta\theta'_h, \theta_r + \Delta\theta'_r)^2}{r_2^2} + \frac{w_{major}(\theta_v + \Delta\theta'_v, \theta_h + \Delta\theta'_h, \theta_r + \Delta\theta'_r)^2}{r_1^2} - \frac{1}{r_{major}(\theta_v + \Delta\theta'_v, \theta_h + \Delta\theta'_h)^2} = 0 \end{array} \right.$$

The set of  $(\theta_v, \theta_h, r_1, r_2)$  used to simulate the geometry of the first ring is (15.0, 7.5, 7.5, 11.0, 10.0). Then the second ring is set as  $\Delta\theta_v = 10.0^\circ$ ,  $\Delta\theta_h = 5.0^\circ$ , and  $\Delta\theta_r = 5.0^\circ$ ; the third ring is set as  $\Delta\theta'_v = 15.0^\circ$ ,  $\Delta\theta'_h = 7.5^\circ$ , and  $\Delta\theta'_r = 7.5^\circ$ . The initial

guess shall be closer in the five-unknown system of equation than in the four-unknown counterpart. An initial guess of the solution of  $(\theta_v, \theta_h, r_1, r_2)$  is given as (20.0, 10.0, 10.0, 12.0, 9.0). Because of a closer initial guess, the solver converges in less iterations: after 13 iterations at the desired solution as shown in Figure 0.8.

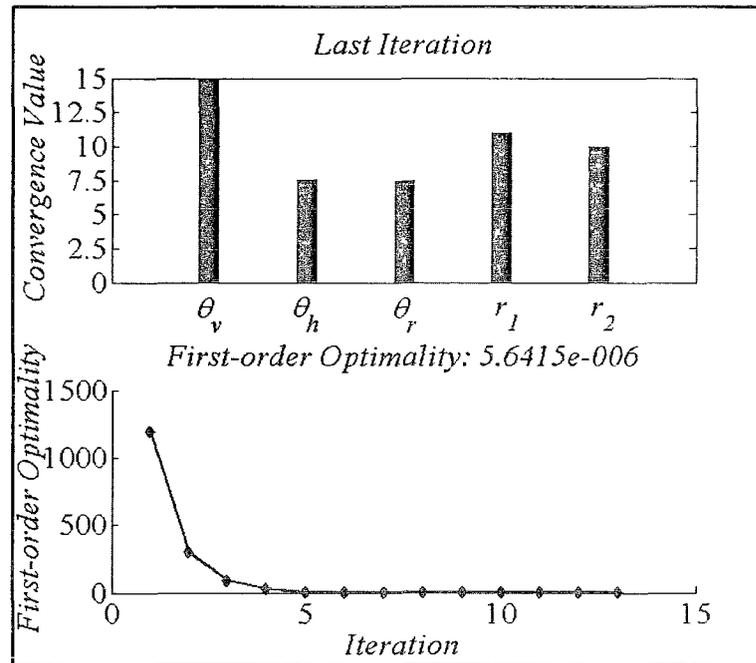


Figure 0.8 Algorithm Performance (Close Initial Guess)

### 5.5 Enhanced Algorithm for Double Ended Scanner

The orientation calculation algorithm for a single-ended scanner is developed in Section 5.3 and Section 5.4. In the course of orientation correction research, the algorithm for double-ended scanners developed by Dettmer [8] is also reviewed. A double-ended scanner utilizes one camera and laser ring on both front and rear ends of the robot, as shown in Figure 0.9. The robotic length is denoted by  $PQ = L$ .  $Q$  and  $P$  represent the measurement origins;  $C_f$  and  $C_r$  represent the centers;  $O_{fh}$  and  $O_{rh}$  represent the

horizontal offset; and  $O_{fv}$  and  $O_{rv}$  represent the vertical offset, in the front and rear plane, respectively. Therefore, the relative offset between  $P$  and  $Q$  are:  $O_h = O_{fh} - O_{rh}$  and  $O_v = O_{fv} - O_{rv}$ . The coordinates system for deducing  $\theta_v$  and  $\theta_h$  follows the consistency in this research, as shown in Figure 0.10.

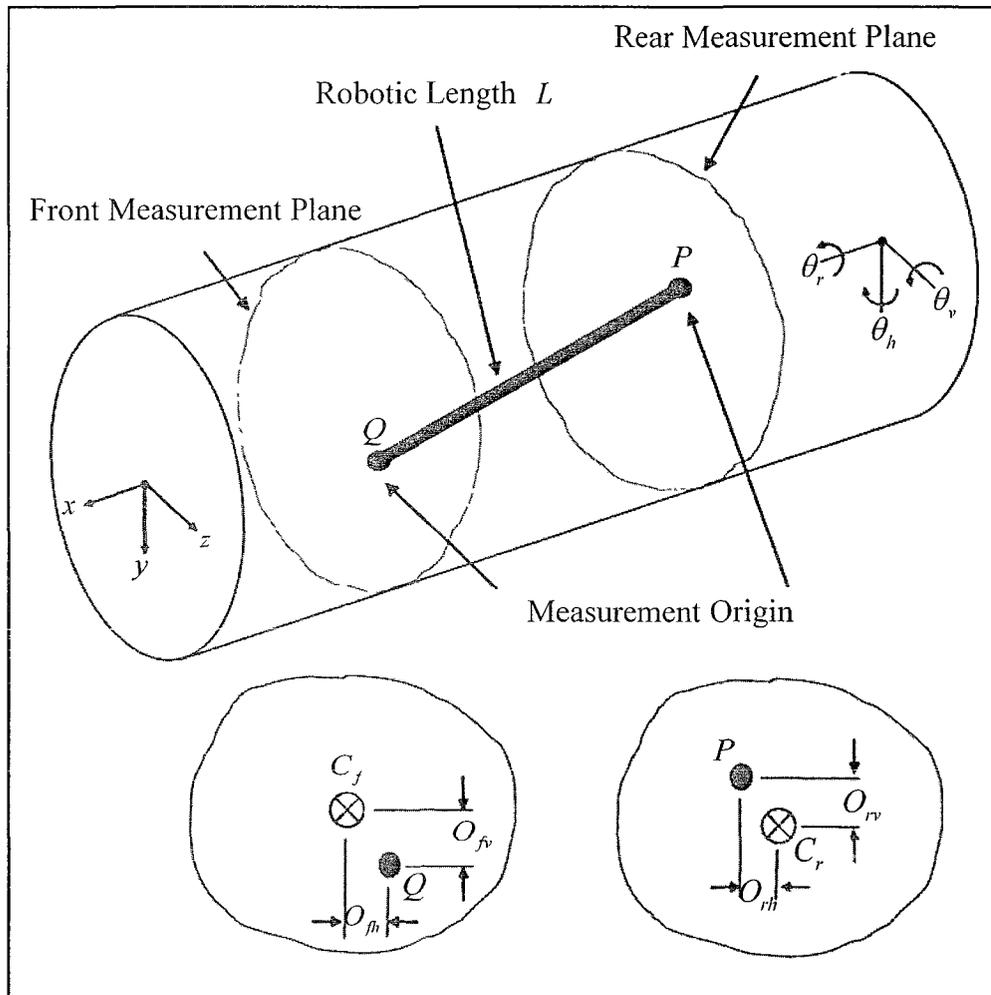


Figure 0.9 Double Ended Scanner

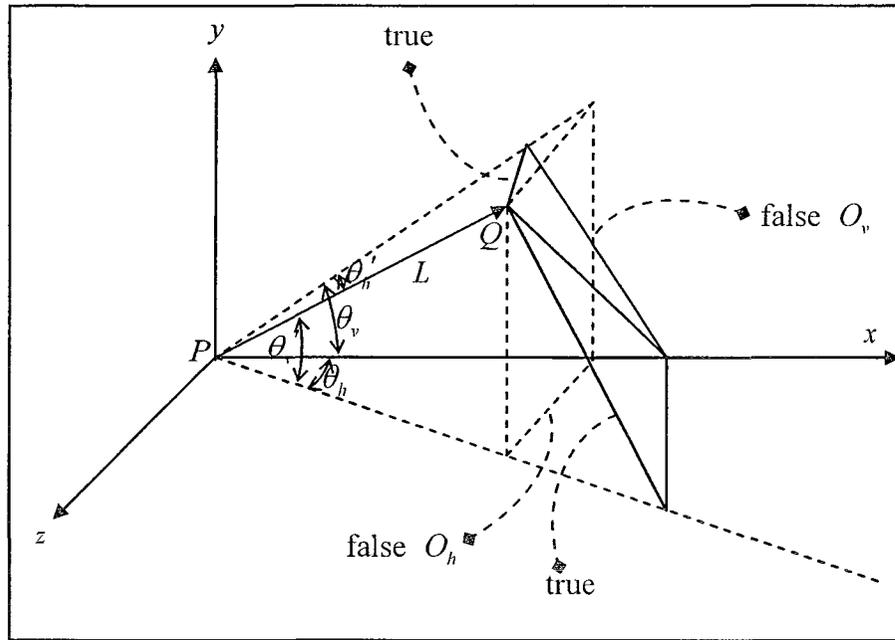


Figure 0.10 Coordinates System for Double Ended Scanner

$\theta_h$  and  $\theta_v$  were assumed to be small by Dettmer [8] and were approximated as Eq. (5.32) and Eq. (5.33):

$$\theta_h = \tan^{-1}\left(\frac{O_h}{L}\right), \quad (5.32)$$

$$\theta_v = \tan^{-1}\left(\frac{O_v}{L}\right). \quad (5.33)$$

A careful examination reveals that an accurate representation can be obtained as Eq. (5.34) and Eq. (5.35):

$$\theta_h' = \sin^{-1}\left(\frac{O_h}{L}\right), \quad (5.34)$$

$$\theta_v' = \sin^{-1}\left(\frac{O_v}{L}\right). \quad (5.35)$$

Combining Eq. (5.32), Eq. (5.33), Eq. (5.34), and Eq. (5.35) and taking out  $O_h/L$  and  $O_v/L$ , Eq. (5.36) and Eq. (5.37) can be obtained:

$$\tan \theta_h' = \tan \theta_h \cos \theta_v, \quad (5.36)$$

$$\tan \theta_v' = \tan \theta_v \cos \theta_h. \quad (5.37)$$

It is clear from Eq. (5.36) and Eq. (5.37) that  $\theta_h$  and  $\theta_v$  can be obtained by either solving a nonlinear system of equations using numerical iterations, or by solving  $\theta_h$  in Eq. (5.38) first and then substituting into Eq. (5.39).

$$\tan \left[ \cos^{-1} \frac{\tan(\sin^{-1}(O_h))}{\tan \theta_h} \right] \cos \theta_h = \tan[\sin^{-1}(O_v)], \quad (5.38)$$

$$\theta_v = \cos^{-1} \left[ \frac{\tan(\sin^{-1}(O_h))}{\tan \theta_h} \right]. \quad (5.39)$$

The difference between using approximations and accurate equations are depicted in Figure 0.11. As expected, Eq. (5.32) and Eq. (5.33) yield good orientation approximations if the level of robotic deviation is low. However, if the level of robotic deviation is not negligible, the accurate representation should be used, as the orientation enhancement can achieve larger than  $4^\circ$ . Such a phenomenon is more obvious for a smaller robotic length  $L$  as larger  $O_h/L$  and  $O_v/L$  arises for the small level of offset  $O_h$  and  $O_v$ .

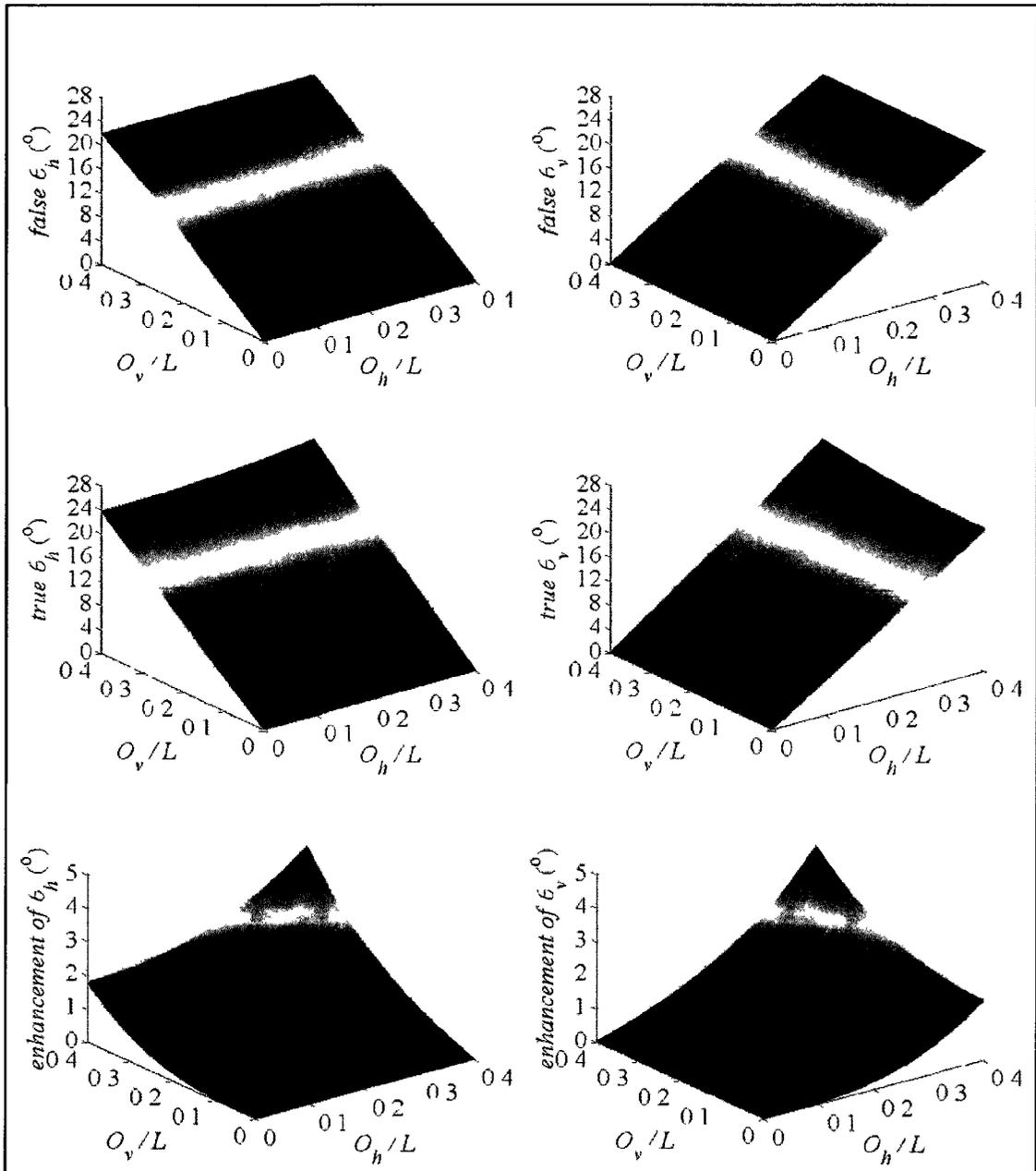


Figure 0.11 Orientation Calculation and Enhancement

## 5.6 Chapter Summary

This chapter develops the three shape parameters of an oblique cross section in an elliptical cylindrical pipe. The relationship between the semi-major axis  $r_{major}$ , the

semi-minor axis  $r_{minor}$ , and the rotational angle  $\beta$  are established in terms of the pipe geometry  $r_1$ ,  $r_2$  and the robotic pitch  $\theta_v$ , yaw  $\theta_h$ , and roll  $\theta_r$ ; this relationship is verified with the help of laser dots placement simulation and ellipse fitting. A new single-ended algorithm reverses the theoretical derivation and relates the pipe geometry and robotic deviation to the acquired laser rings. By taking multiple laser ring scans, a system of nonlinear equations can be established and solved to retrieve the pipe geometry and robotic deviation. Also, a correct version of an existing double-ended algorithm is explored to enhance the robotic deviation estimation accuracy. APPENDIX C gives some of the MATLAB® code for this chapter.

## **CHAPTER 6**

### **IMAGE AND GEOMETRY FUSION FOR FEATURE DETECTION**

Joint dislocation can be caused by improper pipe connections, heavy ground loadings and root penetration. It is also common to see joint dislocation problems after extreme weather changes such as long droughts, heavy precipitation, or a succession of the two. Even if a joint is not dislocated, the connection between pipe segments often requires a small gap to allow expansion. Such discontinuities in the surface normal vectors affect the reflectance of light. So in a clean pipe, all joints, dislocated or not, are easily recognizable by human eyes. Before a lining rehabilitation, joints are useful stable features for additional geometry data processing or fusion with other sensor data.

If all joints can be detected, an articulated model can be formed for a coarse temporal geometry co-registration where joints are forced to match correspondingly. Such a coarse co-registration essentially disassembles the whole length of pipe into smaller segments to improve the confidence and accuracy of the following refined co-registration. Note that a joint detected during one inspection may not be present during another. The detection algorithm can fail because of the noise introduced by the interior surface condition and poor illumination. It is also possible that a dislocated joint has been repaired. Therefore, care should be taken when matching joints one by one.

After the host pipe is rehabilitated with a liner product, joint features are obscured or eliminated in the inspection image because the seamless liner has covered the gaps between individual host pipe segments. In this case, another time-invariant feature available is laterals. A lateral cut, if performed well, shall match the shape of the lateral pipe. Since almost all lateral pipes in a water and sewer system have circular cross sections, an ideal lateral cut shall exhibit either a circular or an elliptical contour, depending on whether the lateral is perpendicular to the host pipe or not.

### **6.1 Image Based Joint and Lateral Detection**

Digital image processing are a cluster of techniques in which images are digitized and various mathematical operations are applied, to perform some interpretation and recognition tasks usually performed manually, or to create an enhanced image that is more meaningful to humans. Image processing has extensive applications in many areas, including medical imaging, astronomy, robotics and satellites remote sensing. According to the typical operations involved, image processing techniques can be categorized into color corrections, image registration, object recognition, image segmentation, image recovery, Euclidean geometry transformations, etc.

Images can suffer from noise, blurring, non-uniform illumination, low resolution, and low contrast. After image acquisition, a pre-processing step is usually applied to correct or extract some specific information, because it is mandatory to process images in a fashion that the assumptions required by processing algorithms are satisfied. Pre-processing operations can include, for instances, a noise reduction phase to assure that sensor noise does not introduce false information; a re-sampling phase to assure that the image coordinates system is correct; and a contrast enhancement phase to assure that

relevant feature can be detected. Additional optical system (e.g. camera) characteristics can be used as input parameters that can help to guide the pre-processing and therefore improve the quality of the image restoration.

For an illustrative purpose, two images containing both a joint and a lateral from Ahrary [2] are used. Figure 0.1 shows edges detected using various techniques including Canny, Log, Sobel, Prewitt, and Roberts. As depicted, Canny is able to detect small edges; Log yields small grain noise; Sobel, Prewitt, and Roberts are robust to white noise and yield equally good edges as desired.

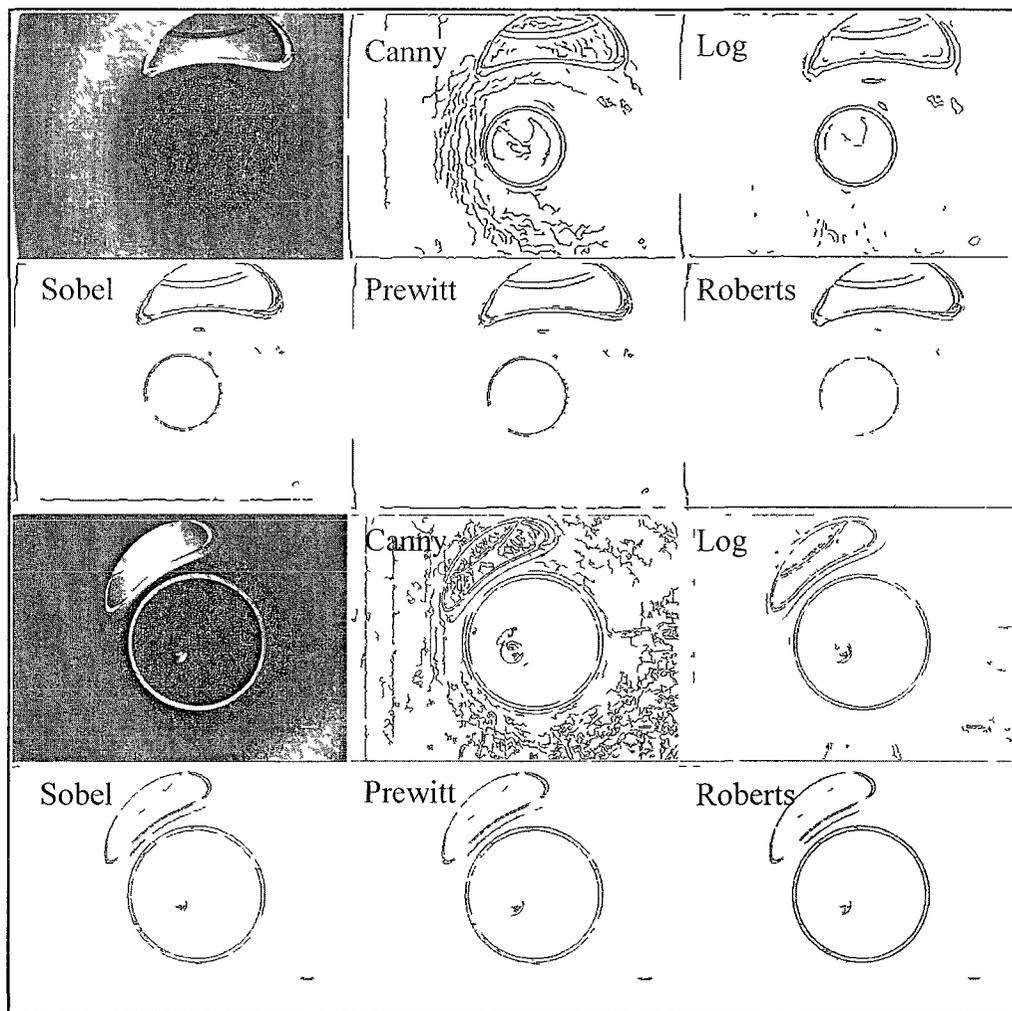


Figure 0.1 Joint and Lateral Detection with Different Edge Detectors

If none of the popular techniques is used, morphological operations, as introduced in Section 2.9.3, provide an alternative to edge detection. Outputs from thresholding with non-uniform illumination removal are shown in Figure 0 2. The first row depicts the background approximation surface; the second is the image background; the third is the subtraction of the background image from the original image; the fourth is the new binary image by apply a threshold to the adjusted image.

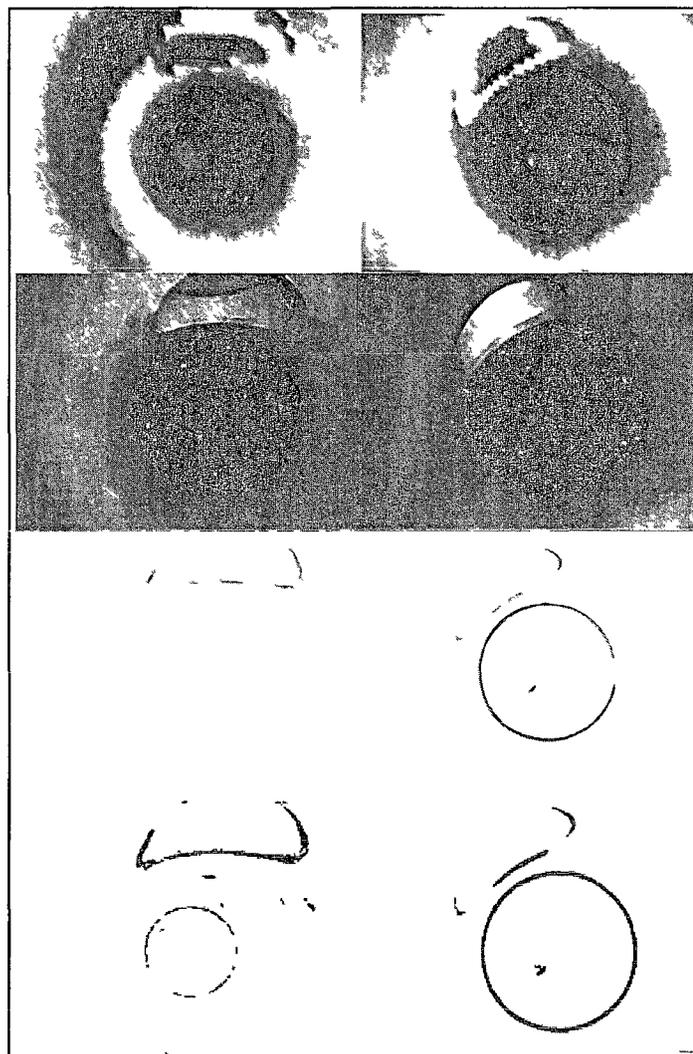


Figure 0.2 Joint and Lateral Detection with Morphological Operations

## 6.2 Lateral and Joint Geometry Modeling

A lateral cut can be thought as the intersection of two cylinders. 3-D and cross sectional views of a pipe segment with two joints and a lateral are shown in Figure 0.3; the Cartesian coordinates system is defined consistently throughout this research.

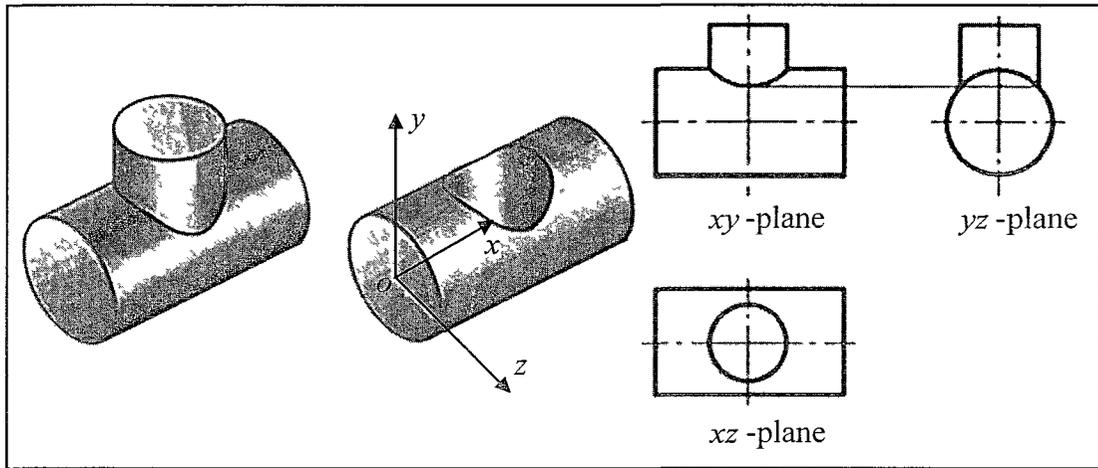


Figure 0.3 Coordinates System for Joint and Lateral

Without loss of generality, the lateral is assumed to have  $y$ -axis being the central axis; it shall satisfy  $x^2 + z^2 = r_l^2$  and  $z^2 + y^2 = r^2$ , where  $r_l$  is the lateral radius and  $r$  is the pipe radius. The parametric form is as follows:  $x = r_l \cos \theta$ ,  $y = \sqrt{r^2 - (r_l \sin \theta)^2}$ ,  $z = r_l \sin \theta$ , where  $\theta \in [0, 2\pi]$ .  $y$  is set to be positive because laterals are usually located at the top half of the pipe.

Since the joint represents a point set that are the circumference of on the pipe, so in the same coordinates system and without loss of generality, it shall satisfy  $z^2 + y^2 = r^2$ , where  $r$  is the radius of the pipe. The parametric form is as follows:  $z = r \cos \theta$ ,  $y = r \sin \theta$ , where  $\theta \in [0, 2\pi]$ .

### 6.3 Camera's Pose Modeling

The robotic platform cannot guarantee a smooth path along the pipe axis, i.e., the camera can have a orientation defined by  $\theta_v$ ,  $\theta_h$  and  $\theta_r$  relative to the earth's coordinates system. The geometric information shall be transformed to the camera's coordinates system in order to derive the azimuth and zenith angles. This section is devoted to lateral and joint geometry modeling for any camera's pose.

In the earth's coordinates system in Figure 0.4, **eye** indicates the camera position (usually the lens' focus); **look** is a point on the camera's optical axis, as its name suggests, indicating where the camera is looking. The vector **n**, opposite to the viewing direction, is taken to be one of the three principle axes. An approximate **up** vector is then assigned to describe the direction from the camera's bottom to the top if the camera is held up straight; **up** is chosen to be perpendicular to **n**, so the other two principle axes can be determined by cross product operations.

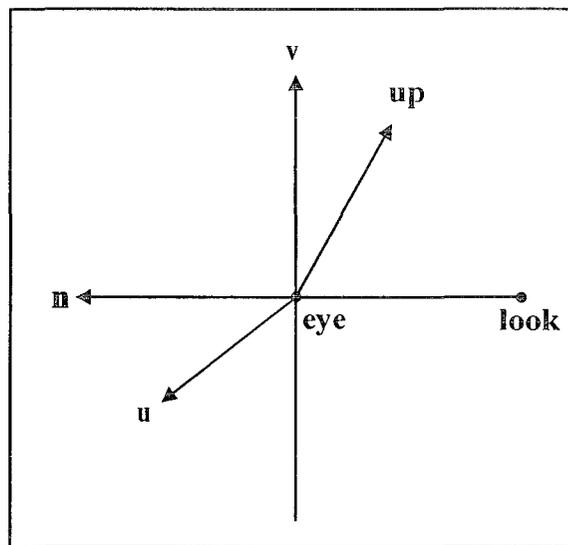


Figure 0.4 Camera's Pose in Earth's Coordinates System

The new unitary principle axes are  $\mathbf{n} = \mathbf{eye} - \mathbf{look}$ ,  $\mathbf{u} = \mathbf{up} \times \mathbf{n}$ , and  $\mathbf{v} = \mathbf{n} \times \mathbf{u}$ . An intuitive way to remember these three vectors is:  $\mathbf{n}$  points from the camera's front to the back;  $\mathbf{u}$  points from the camera's left to the right; and  $\mathbf{v}$  points from the camera's bottom to the top. After the axes of the camera's coordinates system are defined, a transformation matrix  $\mathbf{V}$  that maps  $\mathbf{n}$  to the  $x$ -axis,  $\mathbf{v}$  to the  $y$ -axis, and  $\mathbf{u}$  to the  $z$ -axis of the earth's coordinates system in a homogeneous system can be found as:

$$\mathbf{V} = \begin{pmatrix} n_x & n_y & n_z & d_x \\ v_x & v_y & v_z & d_y \\ u_x & u_y & u_z & d_z \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

where  $(-\mathbf{eye} \cdot \mathbf{n}, -\mathbf{eye} \cdot \mathbf{v}, -\mathbf{eye} \cdot \mathbf{u}) = (d_x, d_y, d_z)$ ,  $\mathbf{u} = (u_x, u_y, u_z)$ ,  $\mathbf{v} = (v_x, v_y, v_z)$ , and  $\mathbf{n} = (n_x, n_y, n_z)$ .

Figure 0.5 shows the contour of a lateral before and after applying  $\mathbf{V}$  to transform the contour from the earth's coordinates system to the camera's coordinates system. The following values are used:  $r = 10$ ,  $r_l = 5$ ,  $\mathbf{eye} = (-15, -1, -4)$ ,  $\mathbf{look} = (0.71, 0.71, 0)$ , and  $\mathbf{up} = (0, 1, 0)$ . Using the same values, Figure 0.6 shows the contour of a joint before and after applying  $\mathbf{V}$ .

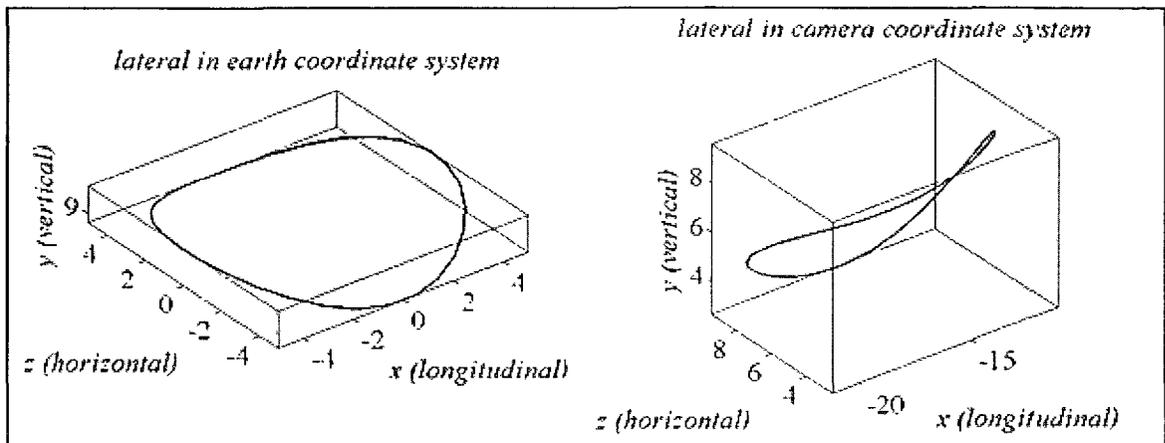


Figure 0.5 Lateral Contours in Earth's and Camera's Coordinates System

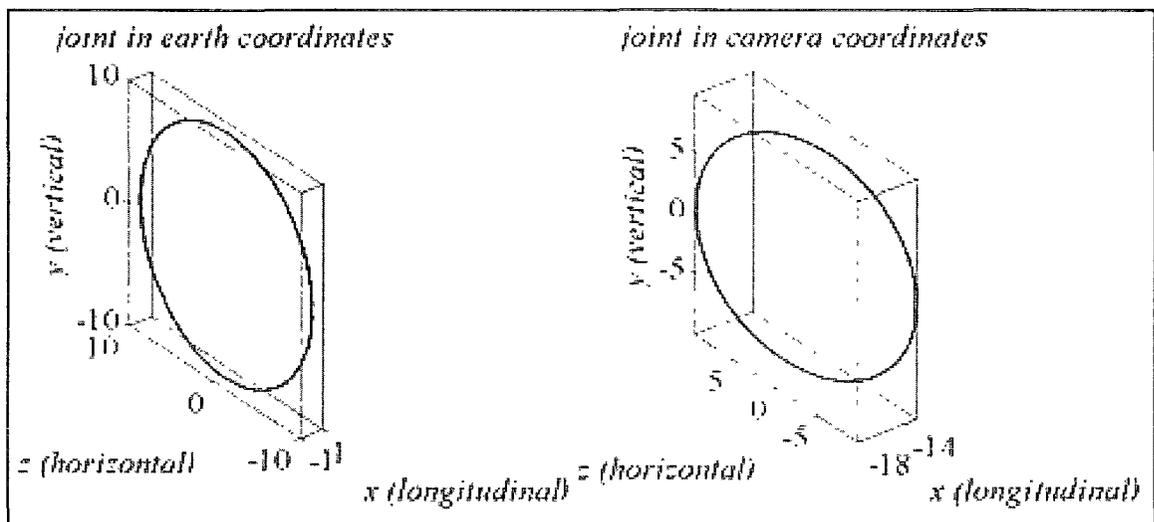


Figure 0.6 Joint Contours in Earth's and Camera's Coordinates System

#### 6.4 Feature in Image by Different Camera Lenses

Calculating the azimuth angle  $\theta_a$  and the zenith angle  $\theta_z$  in the camera's coordinates system and applying each of the five camera lens models (Eq. (2.1), Eq. (2.2), Eq. (2.3), Eq. (2.4) and Eq. (2.5) ) presented in Section 2.7, distortions in the image can be shown for different lens models and varying camera's poses. Such an evaluation is valuable for lens selection before an inspection, image selection for processing after an

inspection with known lens characteristics, and template-based feature detection or segmentation with a known camera's pose. The scaling factor  $k$  in the lens models is not considered for a comparative study.

#### 6.4.1 Laterals in Image

Figure 0.7 simulates what a small lateral ( $r_l/r = 5\%$ ) looks like in the image with different lenses and gradually closing distances (20, 10, 5, 1) from the camera to the lateral. The lateral radius  $r_l$  sets the lower range limit as 0.5. Vector **look** and **up** are kept constant as  $(0, 0, 0)$  and  $(0.71, 0.71, 0)$ , respectively. The viewing angle is  $120^\circ$ ; the image has 768 pixels, both horizontally and vertically.

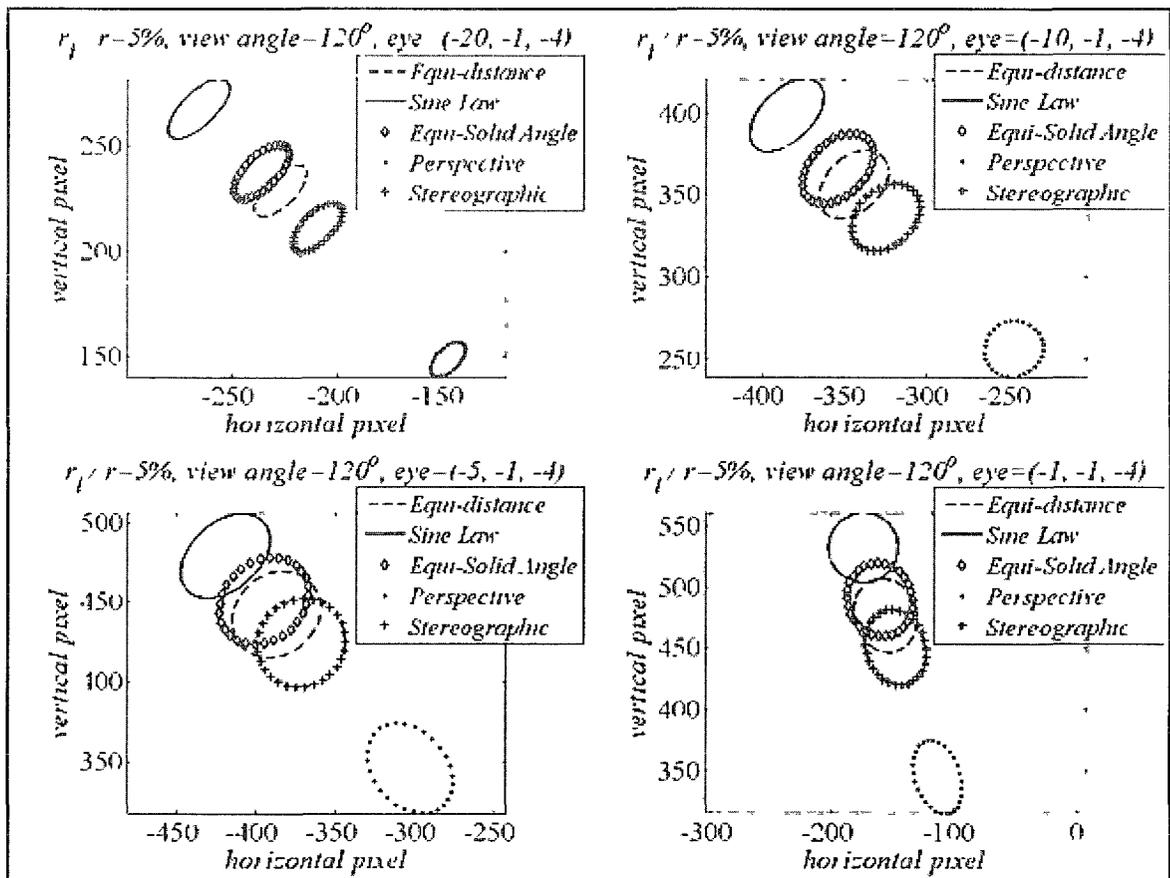


Figure 0.7 Small Laterals in Image Plane Acquired by Different Lenses

Figure 0.8 simulates what a large lateral ( $r_l/r = 50\%$ ) looks like in the image with different lenses and gradually closing distances (30, 20, 10, 8) from the camera to the lateral. The lateral radius  $r_l$  sets the lower range limit as 5. Vector **look** and **up**, the viewing angle, and the image resolution are kept the same as in Figure 0.7.

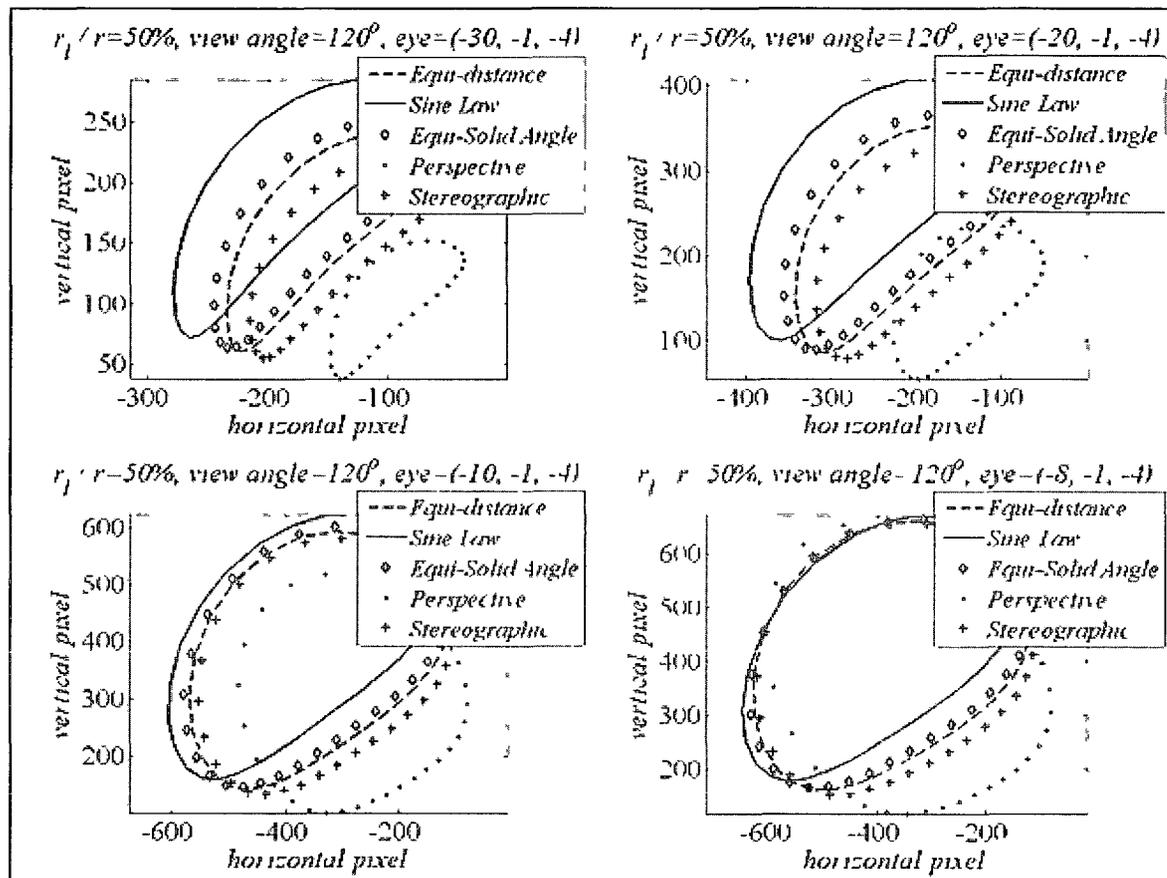


Figure 0.8 Large Laterals in Image Plane Acquired by Different Lens

For a long viewing range, all lens models yield similar contours, only at different pixel locations in the image. As the camera approaches the lateral, a small lateral is able to retain similar contours; while a large lateral undergoes more prominent contour changes. Perspective projection seems to produce a more dramatic contour change with a close viewing range, but the image plane has to be placed extremely close to the lens, which is

physically impractical. Sine-law projection is more preferable than the rest because its less varying contours offer a valuable characteristic for the template-based invariant lateral detection.

It is of interest to study the how the lateral contour size changes with the five lens models and various viewing angle upon entering the camera. It is assumed that the robot does not have much deviation and the camera looks straight ahead; then the distance  $2r_i$  between  $A_i$  and  $B_i$ , i.e., the projections of the nearest point  $A$  and farthest point  $B$  on the lateral onto the image, can be used as an indicator for the contour size, as shown in Figure 0.9. With the sizes of pipe  $r$  and of lateral  $r_l$  known, the relationship between the two zenith angles  $\theta_A$  and  $\theta_B$  can be obtained.

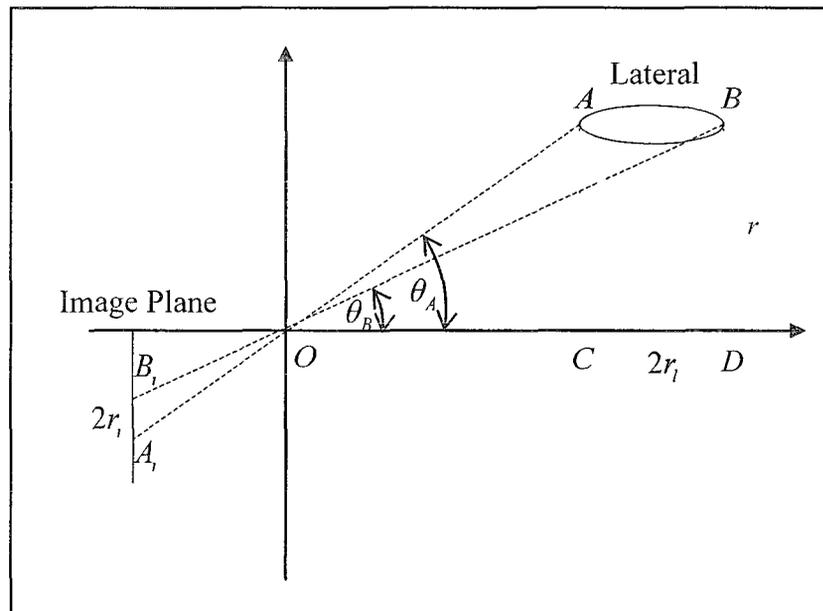


Figure 0.9 Projection of Lateral onto Image Plane

From Figure 0.9,  $CD = 2r_l$  and  $AC = BD = r$ , so it can be obtained that  $CD = OD - OC = BD/\tan\theta_B - AC/\tan\theta_A$  and  $2r_l = r/\tan\theta_B - r/\tan\theta_A$ ; therefore, the relationship between  $\theta_B$  and  $\theta_A$  can be expressed as Eq. (6.1):

$$\tan\theta_B = \frac{r \tan\theta_A}{r + 2r_l \tan\theta_A}. \quad (6.1)$$

Proper substitutions of Eq. (6.1) into the five lens models give the lateral contour size indicator  $r_l$ . Now the zenith angle  $\theta_A$  varies from the maximum viewing angle to  $0^\circ$ , indicating the process of a lateral entering the camera and then fading out. It is assumed that  $r = 10$ ,  $r_l = 0.5$  or  $r_l = 5$ , the maximum viewing angle is  $150^\circ$  and the image has a 768-pixel horizontal and vertical resolution. The lateral size in the image (in pixels) with an increasing viewing angle is shown in Figure 0.10. It is interesting to point out that for the case of a small lateral ( $r/R = 5\%$ ), the sine-law projection may not be monotonic. In this illustration, no requirement is imposed on lens' focus, i.e., the image plane can be adjusted in order to accommodate the lateral to fall into the image for any viewing angle.

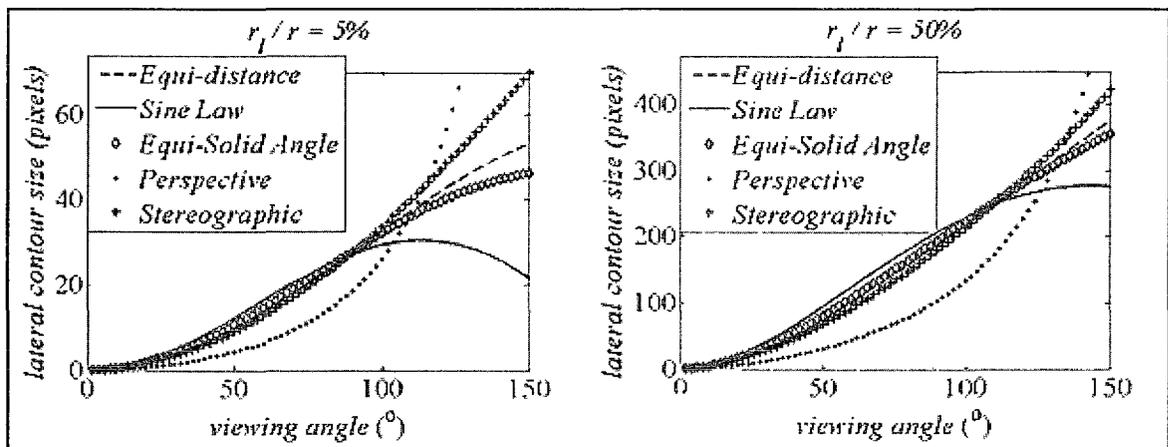


Figure 0.10 Lateral Size in Image with Viewing Angle

### 6.4.2 Joints in Image

A laser ring in the object plane perpendicular to the optical axis always projects to another ring in the image plane, regardless of the projection model. Joints shall appear as a circle if faced perpendicularly towards the camera, so the distance between the camera and the joint does not affect the circularity of the contour. It is the camera's orientation relative to the joint that can distort the contour in the image and this issue is to be examined next. Vector **look** and **up** are chosen to be  $(0, 0, 0)$  and  $(0.71, 0.71, 0)$ . The viewing angle is  $120^\circ$ ; the image has 768 pixels both horizontally and vertically. All five lens models yield circles in the image for joints; gradually closing in the range does not have a signification effect on the shape but the size of the contour, as shown in Figure 0.11.

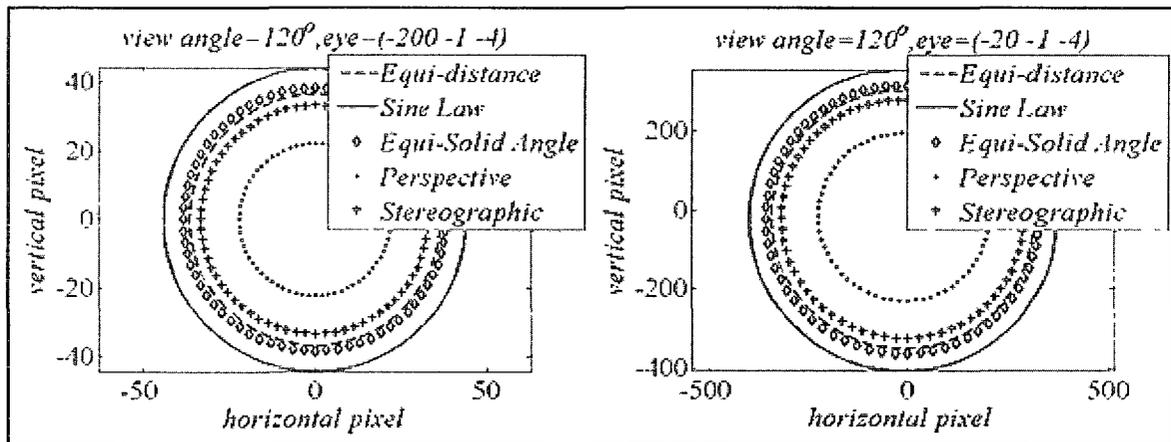


Figure 0.11 Joints in Image Plane Acquired by Different Lens

Note that a robot rarely exceeds the  $\theta_v$  and  $\theta_h$  selected for the simulations in Figure 0.7, Figure 0.8, and Figure 0.11. Sine-law projection, which is preferred over the others in lateral detection due to its invariant contour, and which is equivalent to the others in joint detection, is overall a good choice for common feature detection.

### 6.4.3 Comparison with Real Image

Figure 0.1, which shows two images of a pipe with a round hole on the top taken by a camera, is reproduced in the top row of Figure 0.12. The pipe's radius ( $r$ ), the lateral radius ( $r_l$ ) and the camera's pose ( $\mathbf{eye}, \mathbf{look}, \mathbf{up}$ ) are all unknown. It is assumed that the viewing angle is  $120^\circ$  and the image has 1024 horizontal pixels and 768 vertical pixels.  $r = 10$ ,  $r_l = 8$ , and  $(\mathbf{eye}, \mathbf{look}, \mathbf{up}) = ((-15, -1, -4), (0, 0, 0), (0, 1, 0))$  are intuitively set for the left image;  $r_l = 6$  and  $(\mathbf{eye}, \mathbf{look}, \mathbf{up}) = ((-25, 0, -7), (0, 0, 0), (0, 1, 1.5))$  are set for the right image. The lateral contours after coordinates transformation and sine-law projection are shown in the bottom row of Figure 0.12. Note how similar the extracted contours are to the ones in the original images, even with rough estimates of the camera pose.

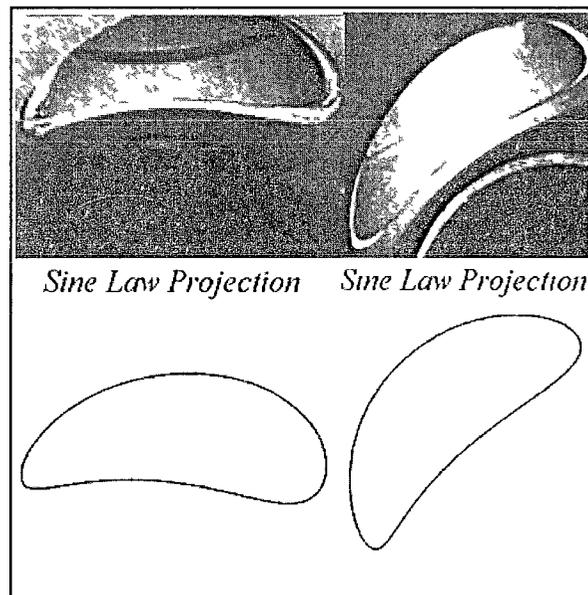


Figure 0.12 Lateral Simulations by Intuition

With the camera pose being tracked using sensors, modeled contours will be closer to real contours. It is worth noting that a lateral may not be perfectly cut open. This implies that if the contour of a lateral is known by image/geometry analysis at an earlier time, such information can be used for image detection of this particular lateral at a later time.

### **6.5 Relationship between Image and Geometry**

The camera's viewing angle and zoom level are two of the many factors in determining lateral locations. In this research, it is hypothesized that the viewing angle remains constant; and that there is no automatic zoom action, unless manually controlled, throughout the inspection, which is the usual practice in the pipe inspection industry. It is impractical to process all images or video frames one by one to identify laterals because of the following reasons: first, the workload is so overwhelming that it can take days to finish processing all images; second, laterals only exist at sparse locations, so it is a waste of time to analyze the lateral-free images; third, deposit buildups and pipe deformation change the light reflectance property of the interior surface, and can yield false lateral alarms because of poor illumination. Therefore, it is desirable to use the geometry measurements as a guide to distill the large amount of images into a much smaller subset. Note that this subset of images may or may not contain laterals due to vision occlusion.

Geometries at lateral connections are either recorded as outwards coordinates larger than the pipe radius or discarded as missing readings, while geometries at deposit buildups and pipe deformations are recorded as inwards coordinates smaller than the pipe radius. Image processing algorithms are then applied to the aforementioned image subset to detect laterals. Geometry measurement and image processing complements each other

by reducing the ambiguity of situations where using one method alone is insufficient, as shown in Figure 0.13.

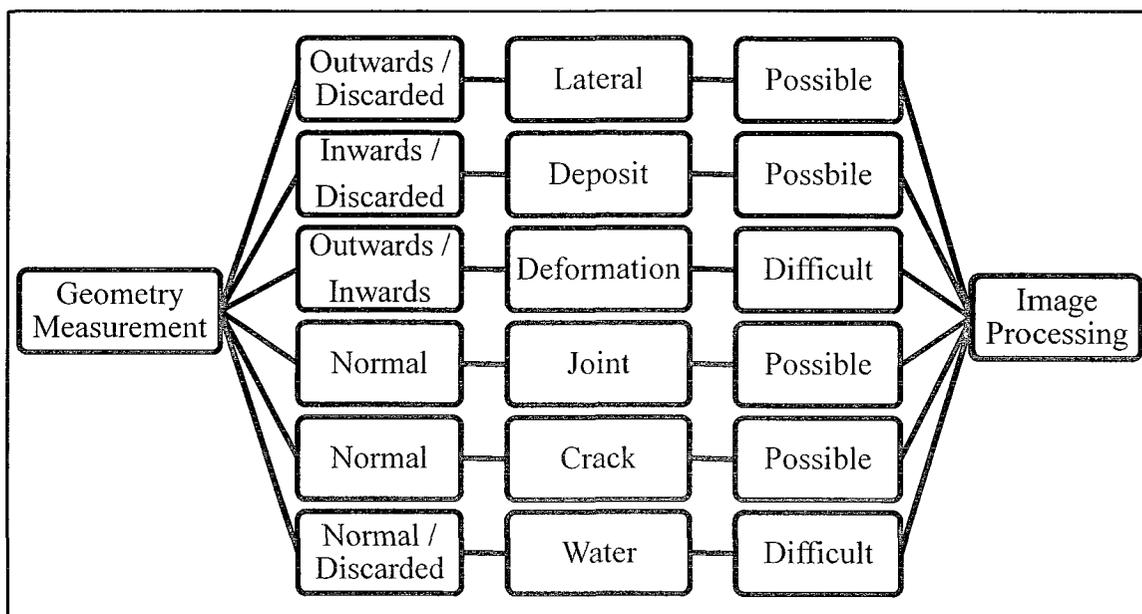


Figure 0.13 Complementary Relationship between Geometry and Image

Based on odometer readings, the correspondence between geometry measurements and images can be found. Note the “correspondence” does not necessarily mean that a lateral detected by image processing locates exactly where the simultaneous geometry measurement takes place. If only one camera is utilized to collect both laser dots and interior snapshot, then the correspondence is exact. However, two robotic runs are needed if laser is used, because laser measurements require a dark ambience while the image snapshot needs good illumination. Therefore, it is possible to use the laser and the image camera on opposite ends of the robot to speed up the inspection, with a light source mainly lighting the image camera end. In this robotic configuration, there can be a distance bias between what is acquired by the image camera and by laser, as shown Figure 0.14.

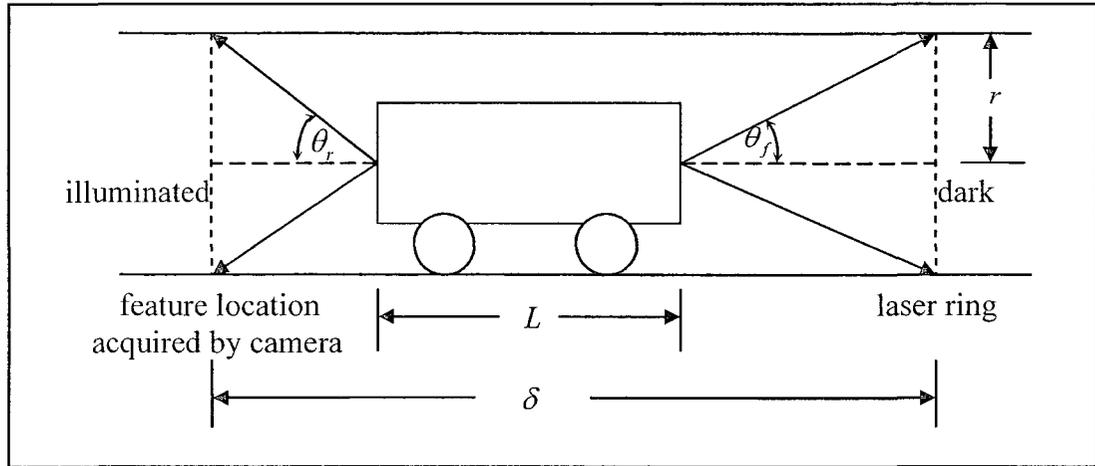


Figure 0.14 Distance Bias Correction for Double Camera System

This bias  $\delta$  can be corrected by calculating the distance between the geometry measurement plane determined by where the laser strikes the pipe and the image snapshot plane determined by the camera's viewing angle and zoom level, as expressed in Eq. (6.2):

$$\delta = L + r(\cos \theta_f + \cos \theta_r), \quad (6.2)$$

where  $L$  is the distance between the two cameras along the robotic body,  $r$  is the pipe's radius,  $\theta_f$  and  $\theta_r$  is half the viewing angle of the front and the rear camera, respectively.

Assuming the geometry measurement detects a suspicious lateral location at location  $x$  from the odometer and the robot travels strictly along the pipe axis, then  $x + \delta$  is where the lateral emerges in the image if the geometry measurement device is in front of the image camera and  $x - \delta$  vice versa. A coefficient  $\mu$  from 1.5 to 3 is suggested in this research, i.e.,  $x - \mu\delta$  or  $x + \mu\delta$ . Such a value usually ensures that the lateral can be captured neither too far away from the image camera, where less visible; nor near the edge of acquired image, where the contour is greatly distorted when a fisheye lens is used.

When the parameter configuration of hardware, i.e., the camera's viewing angle and zoom level, and the robotic length are not known, Eq. (6.2) cannot be established accurately. An alternative way to this case is to process consecutive neighbor images near the suspicious lateral location. A stepwise training is helpful in determining the optimal bias and reducing computational time. A large range of images shall be assigned to the first suspicious lateral location for a good coverage. The nearest image that contains a clear view of a lateral is calculated and the interval is adjusted accordingly. For the second suspicious lateral location, use the updated interval as a starting point, search for lateral images and calculate a new interval again. If the corresponding image can be found within the current interval, keep the upper limit unchanged; if this new value is larger than the current interval, update the upper limit accordingly. Usually, a few such experiments are sufficient to find the optimal interval that guarantees both inclusiveness and a clear view of the laterals.

### **6.6 Template Based Lateral Image Detection**

The light reflectance of the interior pipe surface is non-uniform at layers of mud or fading coatings. Pixel intensity varies across the contour of such places, which are detected by image processing techniques as edges, as discussed in Section 6.1. Also, as illustrated in Figure 0.13, other geometry features can create a false indication of laterals. Therefore, ambiguity arises if decisions are solely based on image or geometry information. In order to reduce this ambiguity, both image and geometry information shall be used if available.

Images shall be aligned with geometry if needed, as discussed in Section 6.5; the robot-pipe relationship shall be estimated by the single-ended or double-ended scanner algorithm if feasible. However, even after these two steps, the camera pose relative to the

lateral can only be coarsely estimated from the suspicious geometry location, and the exact lateral radius is still unknown. The following accepts and isolates the presence of these three uncertainties (lateral size, camera position, and camera orientation); and demonstrates the usage of contour templates and Discrete Fourier Transform (DFT), as introduced in Section 2.9.2, for lateral detection against joint detection. The left picture of Figure 0.15 shows three templates of laterals varying from  $r_l = 0.5$  to  $r_l = 6$  with equal increments. More of these templates are used as inputs to DFT to approximate the true lateral radius. Five pixels noise in the  $y$ -direction are added to the image with a resolution of  $768 \times 768$  pixels. The right picture of Figure 0.15 shows normalized versions of these templates.

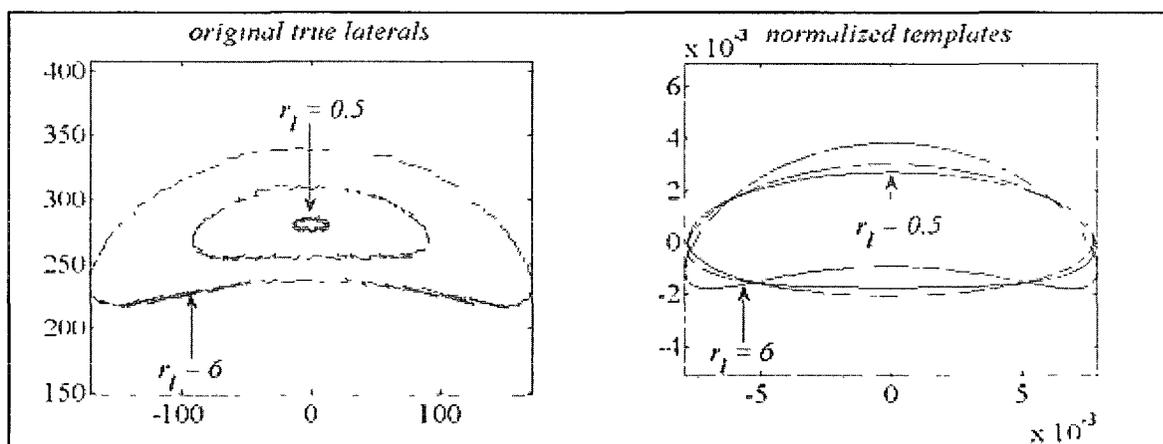


Figure 0.15 Lateral Templates with Different Levels of Noise

Since joints are the most common features in a pipeline, a circle and an ellipse (ovality less than 10%) are used to simulate circular and elliptical joints as benchmarks to check the performance of DFT for lateral detection. The same amount of noise is added to these two benchmarks in the  $y$ -direction. In the following, the performance criterion is set as differences of normalized frequency norms between templates and three shapes (true

lateral contour, circular joint, and elliptical joint). It is desirable to have smaller differences between templates and true contours, and larger differences between templates and joints.

First, it is assumed that the position and orientation of the camera can be accurately estimated but the lateral size is not known beforehand. Figure 0.16 depicts the frequency norm differences by using  $r_l = 6$  and  $r_l = 0.5$  as templates, in the left and right column respectively, to match true lateral contours within this range. Long ( $3r$ ) and short ( $0.8r$ ) viewing distances to the lateral are simulated in the top and bottom row, respectively. It can be seen that a large lateral template ( $r_l = 6$ ) is more distinguishable from a joint regardless of the viewing distance. It is not surprising because a large lateral exhibits a very unique eclipse-shaped contour. Large and small templates both give tolerance to inaccurate lateral size estimation. With a shorter viewing distance ( $0.8r$ ), laterals in the image become more like circles, thus yielding less frequency norm difference.

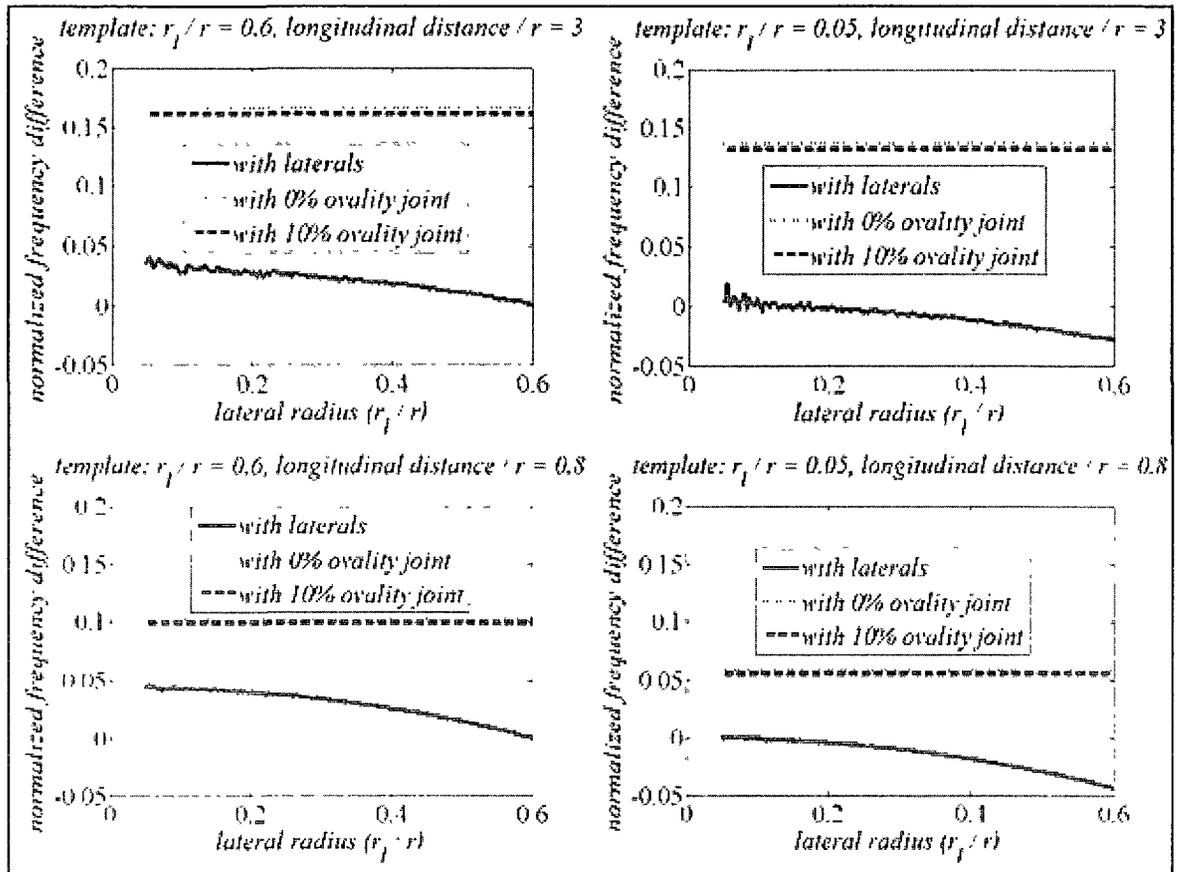


Figure 0.16 DFT with Inaccurate Lateral Size Estimation

Second, the impact of camera position errors is examined, i.e., the camera orientation and the lateral size are known but the camera position relative to the lateral cannot be accurately estimated. The frequency norm difference for a large lateral ( $r_l = 6$ ) is shown in Figure 0.17. The ranges of positional errors in all three directions are chosen to be  $-0.5r$  to  $0.5r$  except in the bottom right picture, where  $-0.1r$  to  $0.1r$  to ensure the camera does not pass the lateral. Negative longitudinal errors indicate a shorter viewing distance. When the viewing distance is long ( $3r$ , left column), horizontal and longitudinal errors still yield consistent frequency norms. Upward and downward vertical movement

make the lateral more compressed and less compressed in the image, respectively. As in Figure 0.16, when the viewing distance is short ( $0.8r$ ), less difference is observed.

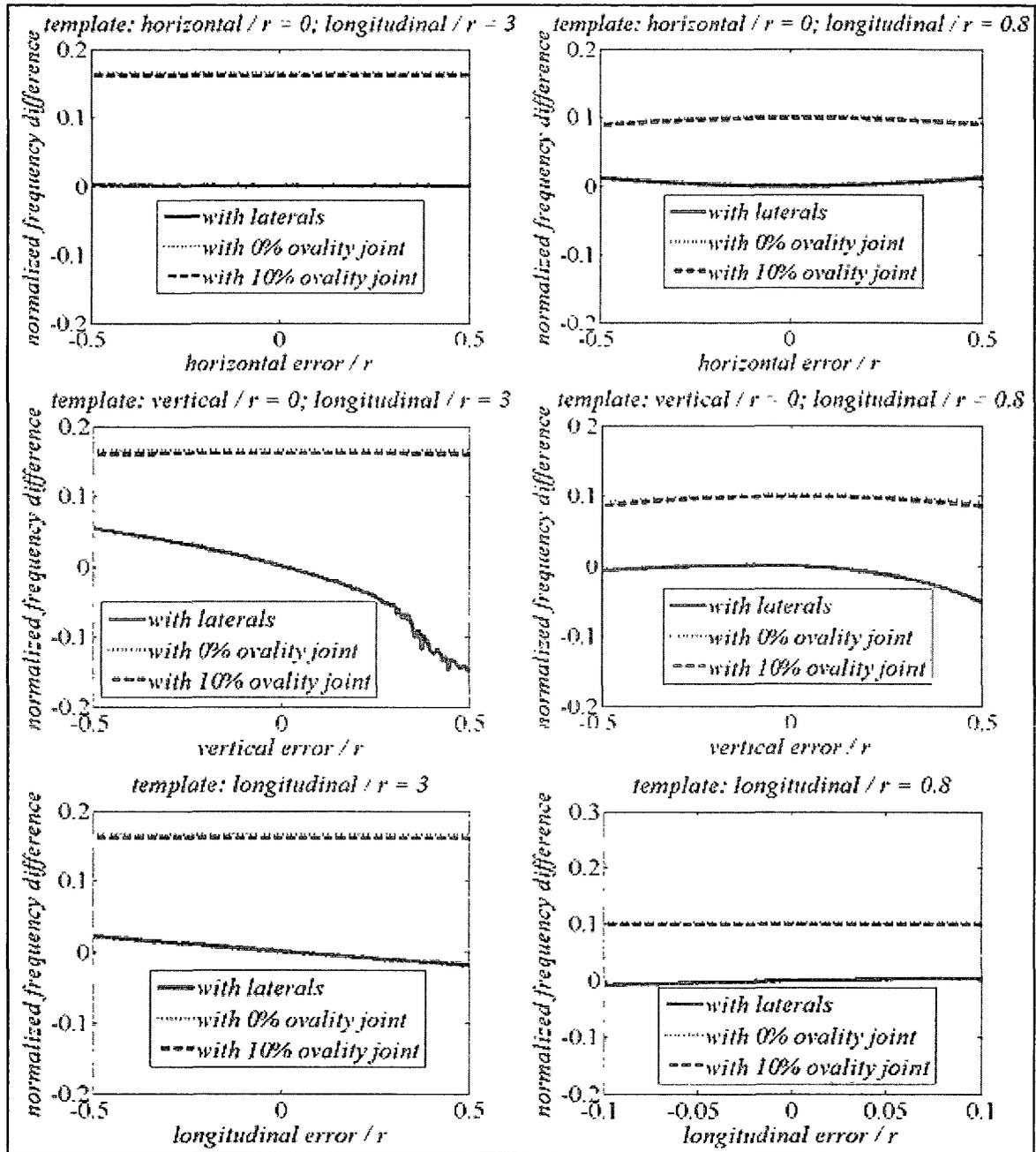


Figure 0.17 DFT with Inaccurate Camera Position Estimation (Large Lateral)

The frequency norm difference for a small lateral ( $r_l = 0.5$ ) is shown in Figure 0 18. The long (left column) and short (right column) viewing distance is  $2r$  and  $0.2r$ , respectively. Similar explanations and conclusions can be made as in Figure 0 17

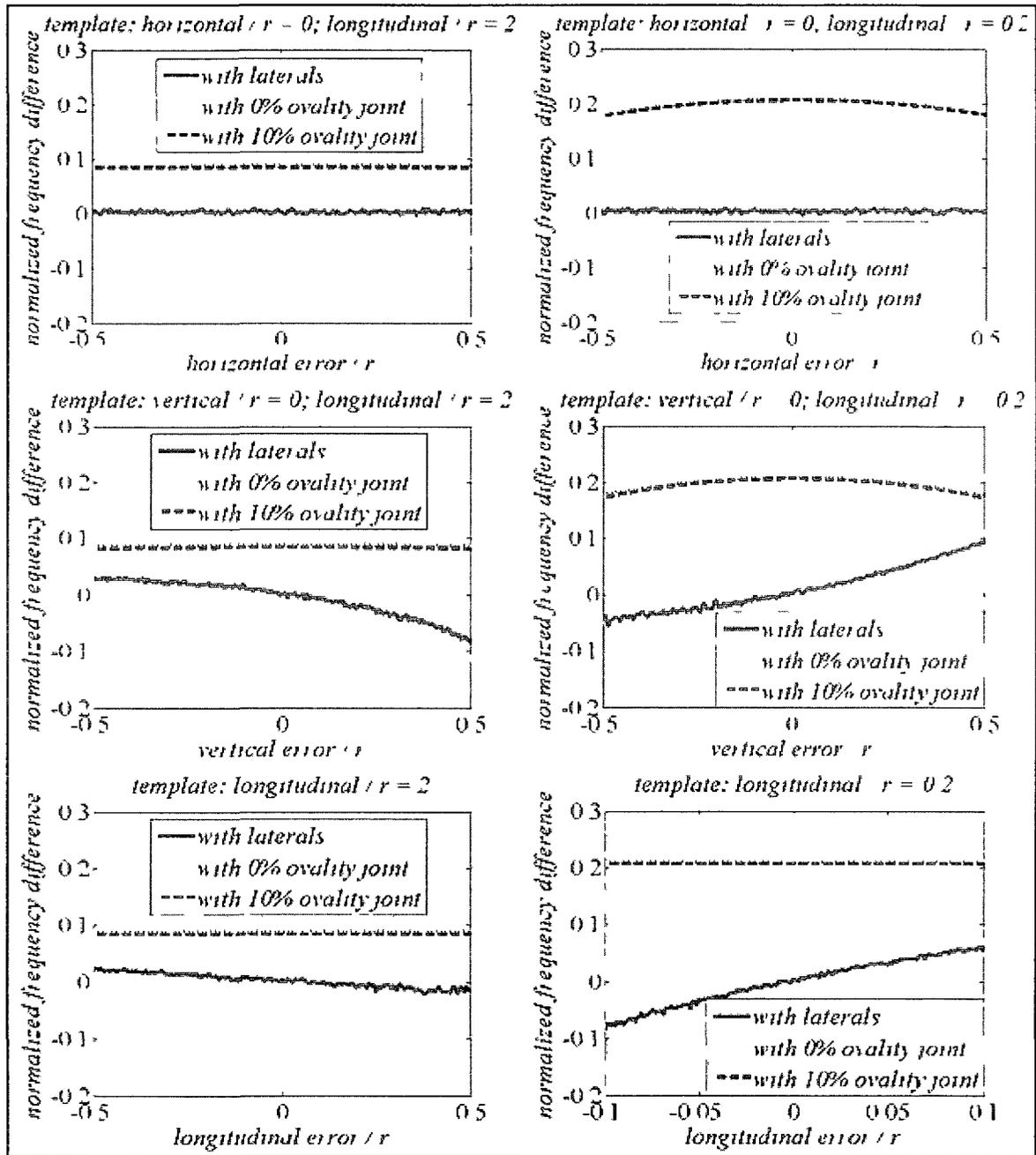


Figure 0 18 DFT with Inaccurate Camera Position Estimation (Small Lateral)

Third, the impact of camera orientation errors is examined, i.e., the camera position and the lateral size are known but the camera orientation relative to the lateral cannot be accurately estimated. The frequency norm difference for a large lateral ( $r_l = 6$ ) is shown in Figure 0.19. As in the discussion of position errors, the left and right column depicts a long ( $3r$ ) and short ( $0.8r$ ) viewing distance, respectively. Pitch, yaw and roll range from  $-30^\circ$  to  $30^\circ$  in all cases. Yaw and roll give relatively consistent frequency norm difference, regardless of the viewing distance. Pitch does not alter the difference much for a long viewing distance, but changes the frequency characteristics dramatically for a short one; especially when the camera is looking upwards, a lateral appear more like a joint in the image. Similar explanations and conclusions can be made for a small lateral ( $r_l = 0.5$ ) in Figure 0.20, where the long and short viewing distance are set to be  $2r$  and  $0.2r$ , respectively.

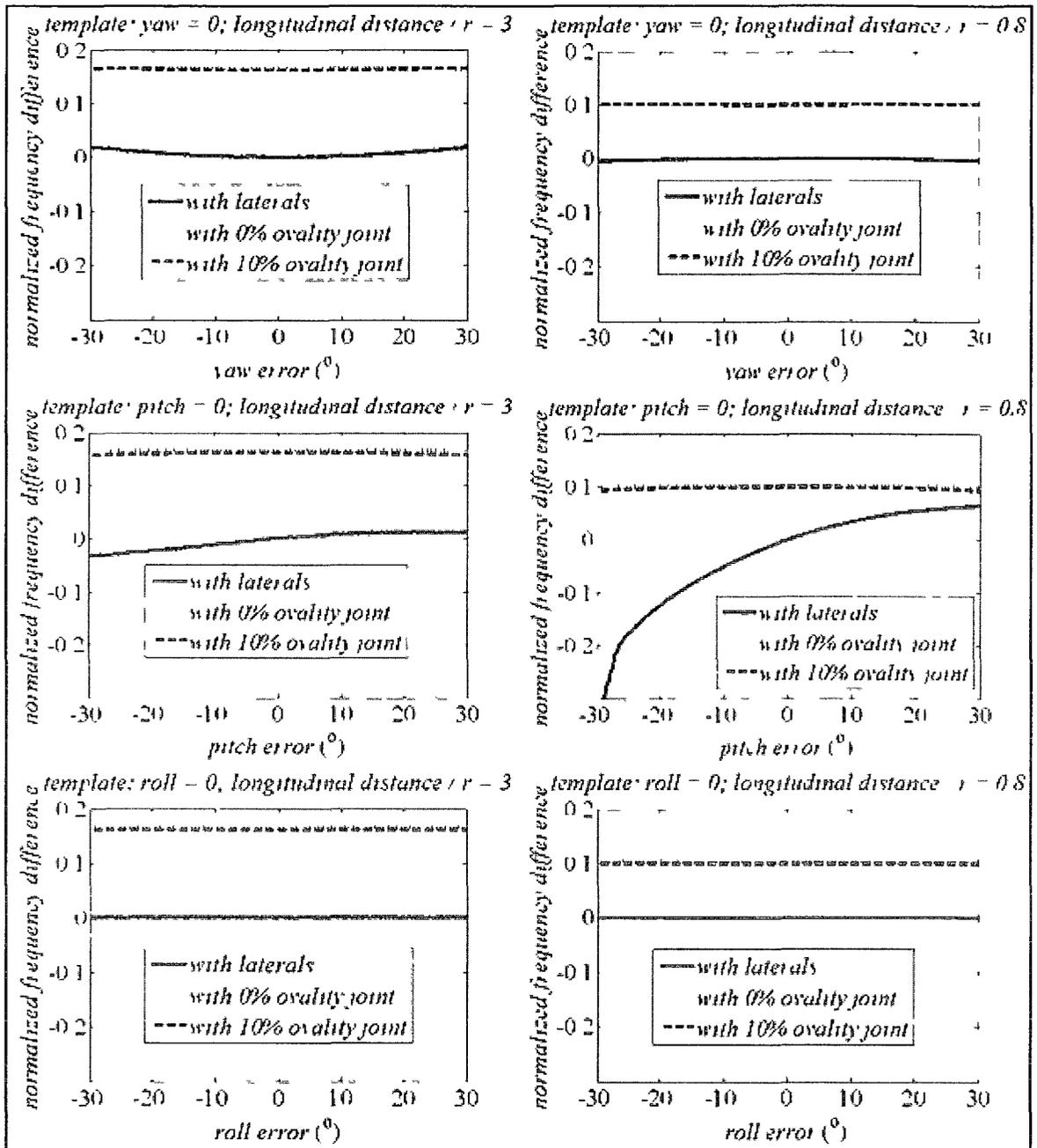


Figure 0.19 DFT with Inaccurate Camera Orientation Estimation (Large Lateral)

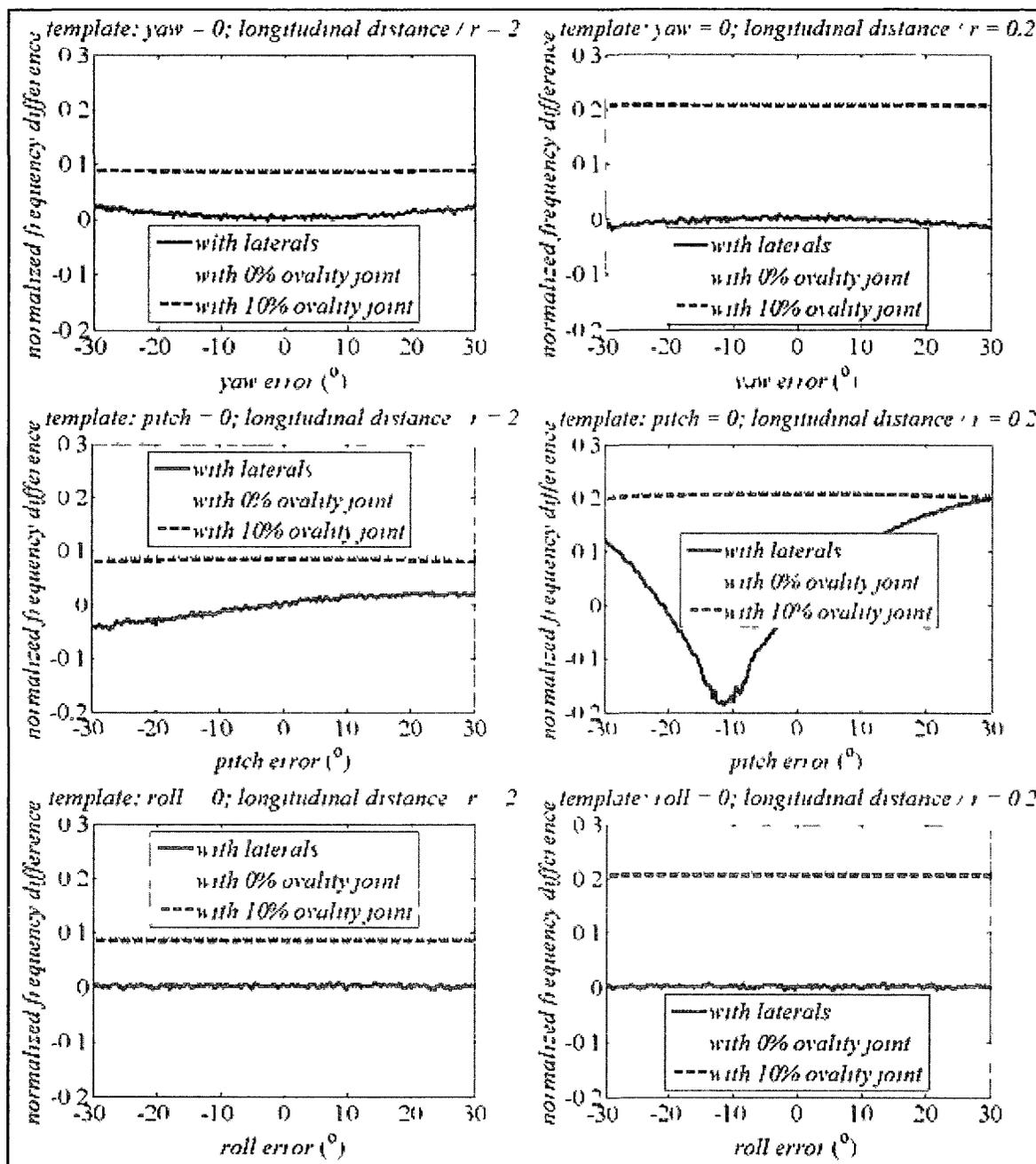


Figure 0.20 DFT with Inaccurate Camera Orientation Estimation (Small Lateral)

## 6.7 Chapter Summary

This chapter models common features such as laterals and joints in the earth's coordinates system. Homogeneous coordinates transformation are then applied to obtain the features in the camera's coordinates system. Camera lens models are used to illustrate

the contour of such features in the image. Certain laterals and camera positions yields a unique contour, resembling an eclipse. A template-based paradigm is carried out to examine the feasibility of templates in separating laterals from joints in the image. Three possible errors (lateral size, camera position, and camera orientation) are introduced to verify the robustness of using the DFT in such a matter. In the process of isolating all the three uncertainties, the “frequency norm difference” is proven to be a feasible metric in terms of lateral detection with joints present. Other error metrics are to be tested, for example, the normalized frequency with phase, the non-normalized version of frequency, and other non-frequency domain metrics. APPENDIX D gives some of the MATLAB® code for this chapter.

## **CHAPTER 7**

### **MATHEMATICAL MODELING OF LINER GEOMETRIES**

Flexible polymeric liners, compared to rigid host pipes, have stronger resistance against corrosion but lower structural strength against external loadings. As a result, corrosion-induced hinged pipe liners are uncommon. Sudden collapse of liners is mitigated by their flexibility; the liners deform until they receive sidewall support from the host pipe, thereby increasing the buckling resistance of the liner to groundwater pressure.

Quite often a host pipe is lined when still sound enough to withstand the live and dead loadings from the soil and the traffic above the ground; therefore, the liner can be considered to be only subjected to the external groundwater pressure. Figure 0.1 illustrates two conceptual deformation pathways for a polymeric liner encased within a circular host pipe, according to Gumble [15].

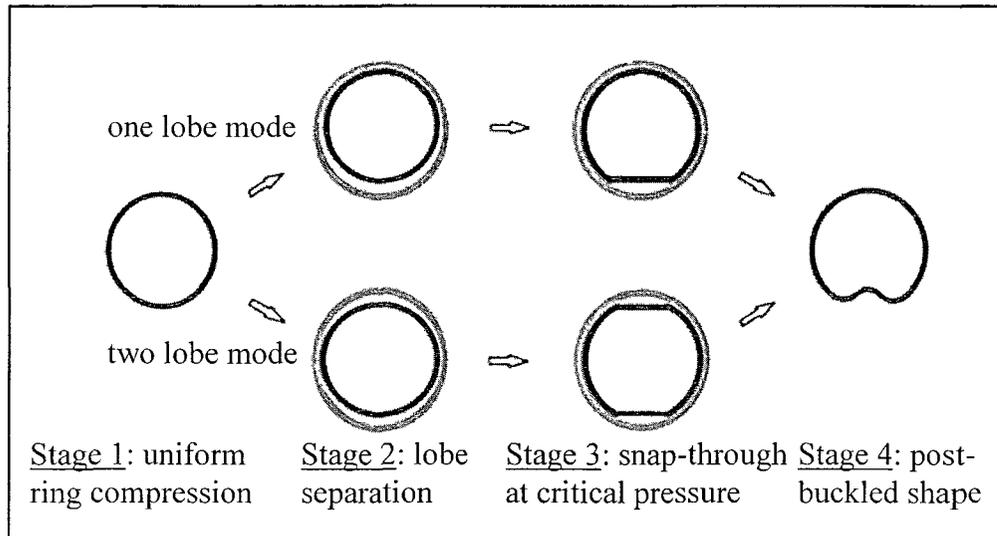


Figure 0.1 Possible Deformation Pathways for Restrained Circular Liners

### 7.1 Quasi Elliptical Cross Section

Though the cross section of a liner, subject to uniform external pressure, seems to be elliptical at Stage 2 in Figure 0.1, this section proves that it is actually not a perfect ellipse. According to Timoshenko [28], for a small deflection  $d_0$  on the horizontal axis, the vertical axis deflection can also be assumed as  $d_0$ , shown in Figure 0.2.

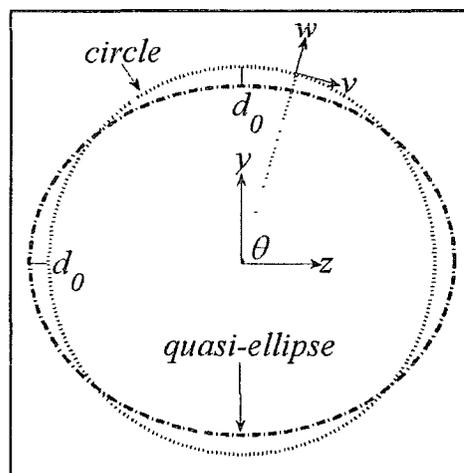


Figure 0.2 Small Deflection Theory for Liner

For any point  $(z, y)$  on the circle,  $z = r \cos \theta$  and  $y = r \sin \theta$ , where  $r$  is the radius of the circle. The radial deflection  $d_r$  and the tangential  $d_t$  is expressed as:

$$d_r = d_0 \cos(2\theta),$$

$$d_t = \frac{1}{2} d_0 \sin(2\theta).$$

Therefore, the coordinates for the new point  $(z_l, y_l)$  is:

$$\begin{aligned} z_l &= z + d_r \cos \theta + d_t \sin \theta \\ &= r \cos \theta + d_0 \cos(2\theta) \cos \theta + \frac{1}{2} d_0 \sin(2\theta) \sin \theta \\ &= r \cos \theta + d_0 \left( (\cos \theta)^2 - (\sin \theta)^2 \right) \cos \theta + d_0 (\sin \theta)^2 \cos \theta \\ &= r \cos \theta + d_0 (\cos \theta)^3, \end{aligned}$$

$$\begin{aligned} y_l &= y + d_r \sin \theta - d_t \cos \theta \\ &= r \sin \theta + d_0 \cos(2\theta) \sin \theta - \frac{1}{2} d_0 \sin(2\theta) \cos \theta \\ &= r \sin \theta + d_0 \left( (\cos \theta)^2 - (\sin \theta)^2 \right) \sin \theta - d_0 \sin \theta (\cos \theta)^2 \\ &= r \sin \theta - d_0 (\sin \theta)^3. \end{aligned}$$

Assuming  $d_0 = r/10 = 1$  and plotting  $(z_l, y_l)$ 's in Figure 0.3, it can be seen that the quasi-ellipse separates slightly from a perfect ellipse. Note that for the small deflection theory,  $d_0$  may have to be taken even smaller than 10% of the radius, therefore bringing the quasi-ellipse even closer to the perfect ellipse. Moreover, the initial gap between the host pipe and liner at Stage 1 in Figure 0.1 is usually much smaller than the radius, so the expansion of the liner stops when the liner contacts and takes the shape of the host pipe,

which is an ellipse, shown at Stage 3 in Figure 0.1. Therefore, it is safe to assume that the elliptical cross section is also a typical type for liners.

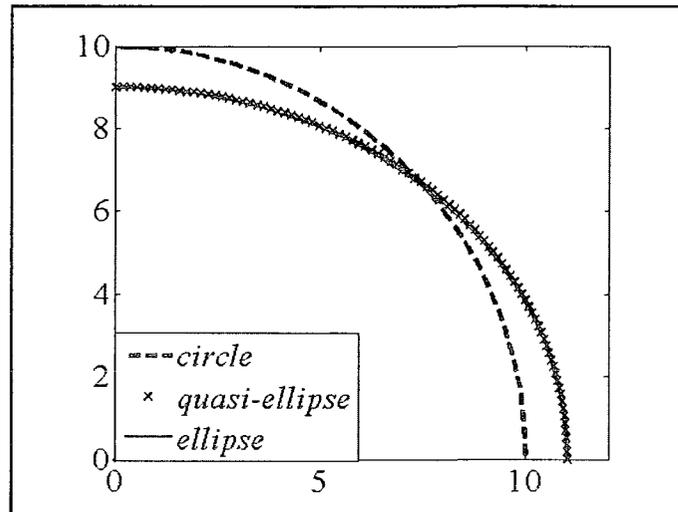


Figure 0.3 Quasi Elliptical and Perfect Elliptical Shape for Liner

## 7.2 Lobed Cross Section

After the critical external uniform pressure is reached, the liner buckles into a wavy form, usually at the crown or invert, as shown at Stage 4 in Figure 0.1. Chicurel [7] and Glock [12] suggested two equations to approximate a small wavy portion. In the following, these two equations are to be evaluated for a relatively large wavy portion in a circular host pipe.

### 7.2.1 Chicurel's Equation

A close-up view of the buckled portion is shown in Figure 0.4, according to Li [18]. Points  $A$  and  $D$  are the two endpoints of the curved arc that peels off the original shape. The coordinates system is chosen such that the  $z$ -axis points to  $D$  with the origin located at  $A$ . As observed, arc  $BC$  is below the  $z$ -axis, while arcs  $AB$  and  $CD$  are above the  $z$ -axis.

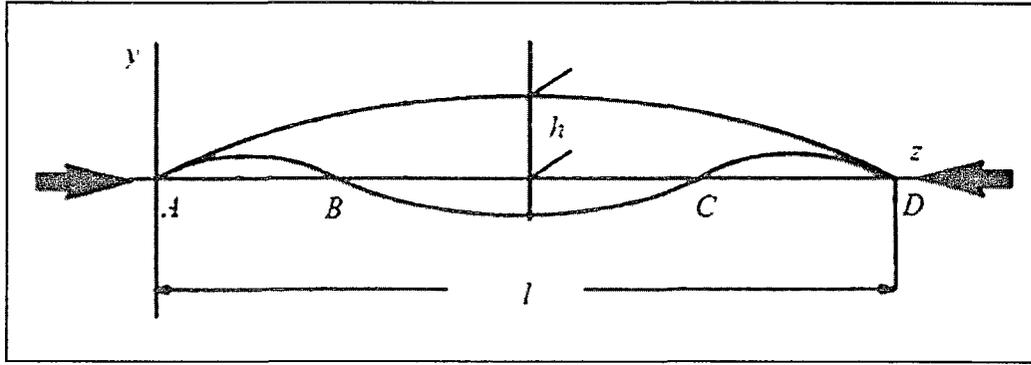


Figure 0.4 Chicurel's Coordinates System for Wavy Buckling Portion

Based on the coordinates system in Figure 0.4, it was suggested by Chicurel that the buckled portion be modeled by Eq. (7.1):

$$y = \frac{2h}{u^2} \left[ \frac{\cos \left[ \left( 1 - \frac{2z}{l} \right) u \right]}{\cos u} - 1 \right], \quad (7.1)$$

where

$h$  = largest height between the original curve to the  $z$ -axis ( $m$ ),

$l$  = length of arc  $ABCD$  along the  $z$ -axis ( $m$ ),

$$u^2 = \frac{Pl^2}{4EI},$$

$P$  = external uniform load ( $N$ ),

$E$  = modulus of elasticity ( $N/m^2$ ),

$I$  = moment of inertia of cross section ( $m^4$ ).

So  $u$  is a dimensionless value that can be determined by boundary conditions. It was assumed that the buckled portion is small and the arc  $AMD$  is approximated by:

$$y = \frac{4h}{l^2} z(l-z).$$

If the buckled portion in Figure 0.4 is large, then the arc  $AMD$  shall be better described by the original circle equation, as expressed in Eq. (7.2),

$$y = \sqrt{r^2 - z^2}, \quad (7.2)$$

where  $r$  is the host pipe radius. Now it is of interest to examine the behavior of Eq. (7.2) at boundaries for a large buckled portion. The derivative of Eq. (7.1) is expressed in Eq. (7.3):

$$\frac{dy}{dz} = \frac{4h \sin \left[ \left( 1 - \frac{2z}{l} \right) u \right]}{ul \cos u}. \quad (7.3)$$

The derivative of Eq. (7.2) is expressed in Eq. (7.4):

$$\frac{dy}{dz} = -\frac{z}{\sqrt{r^2 - z^2}}. \quad (7.4)$$

Note that although the circle is centered at  $(0,0)$ , it does not affect the calculation of the derivatives at point  $A$  as long as the correct corresponding spot is found, which is  $-l/2$ . Therefore, to make smooth the transition from a circular curve to a wavy curve at point  $A$ , the first-order derivatives in Eq. (7.3) and Eq. (7.4) shall be equal to each other, as expressed in Eq. (7.5):

$$\frac{4h \tan u}{ul} = \frac{\frac{l}{2}}{\sqrt{r^2 - \left(\frac{l}{2}\right)^2}}. \quad (7.5)$$

It is obvious that

$$r^2 - \left(\frac{l}{2}\right)^2 = (r-h)^2.$$

Assuming Eq. (7.6):

$$l = 2kr, \tag{7.6}$$

where  $k$  is the ratio of  $l/2$  to  $r$ , then it is easy to derive Eq. (7.7):

$$h = \left(1 - \sqrt{1 - k^2}\right)r. \tag{7.7}$$

By substituting Eq. (7.6) and Eq. (7.7) into Eq. (7.5) and factoring out common terms, Eq. (7.8) can be obtained:

$$\frac{\tan u}{u} = \frac{k^2}{2\sqrt{1-k^2} \left(1 - \sqrt{1-k^2}\right)}. \tag{7.8}$$

By varying  $k$  from 0 to 1, different values of  $l$ ,  $h$  and  $u$  can be calculated using Eq. (7.6), Eq. (7.7), and Eq. (7.8) to determine all the parameters in Eq. (7.1). A plot of the wavy lobes with various buckled angles is shown in Figure 0.5.

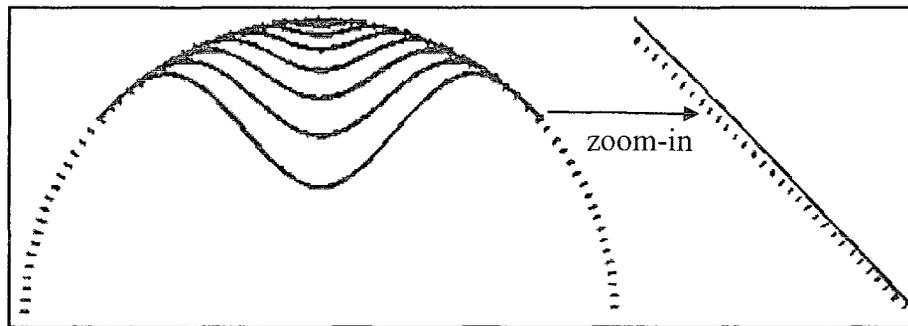


Figure 0.5 Wavy Lobes Approximated by Chicurel's Equation

However, a careful examination at a zoom-in view of the boundary in Figure 0.5 shows that the liner (solid curve) exceeds the confine of the host pipe (dotted curve), which is physically impossible. Next, the reason for this phenomenon is explained by comparing the second-order derivative of Eq. (7.1), as expressed in Eq. (7.9):

$$\frac{d^2y}{dz^2} = -\frac{8h \cos \left[ \left( 1 - \frac{2z}{l} \right) u \right]}{l^2 \cos u}, \quad (7.9)$$

and the second-order derivative of Eq. (7.2), as expressed in Eq. (7.10):

$$\frac{d^2y}{dz^2} = -\frac{r^2}{\sqrt[3]{r^2 - z^2}}. \quad (7.10)$$

As in the calculation of the first-order derivatives,  $z = 0$  and  $z = -l/2$  are substituted into Eq. (7.9) and Eq. (7.10), respectively; and in order for the Chicurel's curve not to go beyond the confines of the circle, Eq. (7.11) shall be satisfied:

$$-\frac{8h}{l^2} < -\frac{r^2}{\sqrt[3]{r^2 - \left(\frac{l}{2}\right)^2}}. \quad (7.11)$$

Replacing  $l$  and  $h$  by Eq. (7.6) and Eq. (7.7), and factoring out common terms, it can be obtained that

$$-\frac{2(1 - \sqrt{1 - k^2})}{k^2} < -\frac{1}{\sqrt[3]{1 - k^2}},$$

which has no solutions of  $k$  because the left side is never less than the right side, as shown in Figure 0.6.

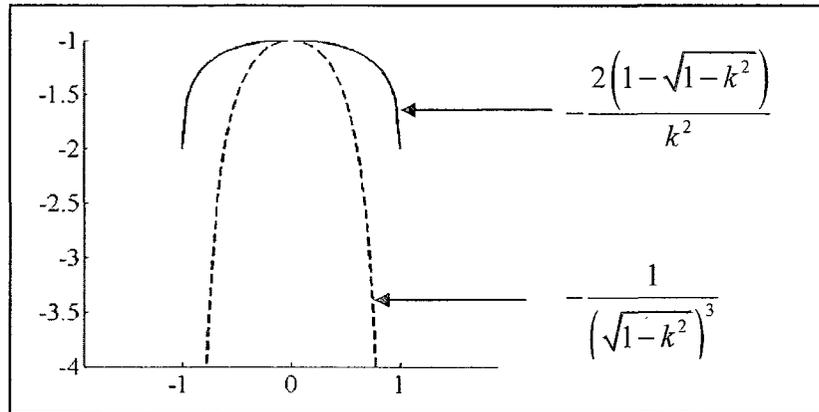


Figure 0.6 Comparison of Second Order Derivatives

Therefore, the decreasing rate of the slope of the Chicurel's equation is always slower than that of the circle equation at the boundary point, making part of the liner violate the physics law. As a matter of fact, for every lobe in Figure 0.5, this phenomenon can be observed if the zoom level is large enough and even more obvious when the host pipe is elliptical. Nevertheless, other than not complying with the common sense, Eq. (7.1) offers a good approximation of the lobe for a small buckled portion.

### 7.2.2 Glock's Equation

Another model of the wavy portion encased in a circular host pipe was described by Glock in Figure 0.7 as Eq. (7.12):

$$d_w(\theta) = d_0 \left( \sin \frac{\pi\theta}{2\varphi} \right)^2, \quad (7.12)$$

where

$2\varphi$  = open angle of buckled portion, and  $\theta \in (0, 2\varphi)$ ,

$d_0$  = largest radial deflection at  $\theta = \varphi$ ,

$d_w(\theta)$  = radial deflection at any  $\theta$ .

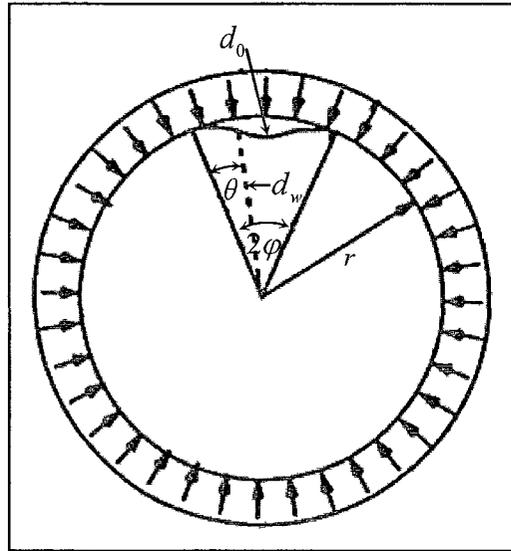


Figure 0.7 Glock's Coordinates System for Wavy Buckling Portion

Clearly, the polar coordinates system ensures that the wavy portion does not exceed the confine because the new radial distance is expressed as  $r_w(\theta) = r - d_w(\theta)$ . The polar expression for the circle is  $r_c(\theta) = r$ , so the first-order derivative of the circle at every  $\theta$  is:

$$\frac{dr_c(\theta)}{d\theta} = 0.$$

The first-order derivative of the wavy portion at  $\theta = 0$  is:

$$\frac{d(r_w(0))}{d\theta} = -\frac{d(d_w(0))}{d\theta} = -\frac{\pi \sin\left(\frac{\pi\theta}{\varphi}\right)}{2\varphi} = 0.$$

Therefore, the smoothness at the boundary is also guaranteed for a host pipe with circular cross sections.

### 7.3 Composite Cross Section

Like rigid pipes, a mode combination is also possible for liners. If the host pipe is still structurally sound, the liner deforms freely until touching the inside of the host pipe. Then that portion of the liner takes the shape of the corresponding portion of the host pipe. Since the liner is sometimes deployed to a host pipe that has already shown some ovality, a large portion of the liner has the same ovality while buckled portion becomes wavy.

However, since the first-order derivative of Eq. (7.12) is always zero at the boundary, obviously Glock's equation is not a good choice when the host pipe is elliptical, where the first-order derivative of the radial distances is non-zero for almost everywhere along the circumference. Moreover, as the deflection continues in the post-buckling phase, the liner is influenced by many factors and can exhibit unpredictable wavy forms, as shown in Figure 0.8. Therefore, a new equation is needed for the elliptical boundary and some flexibility in modeling different configurations of the wavy portion.

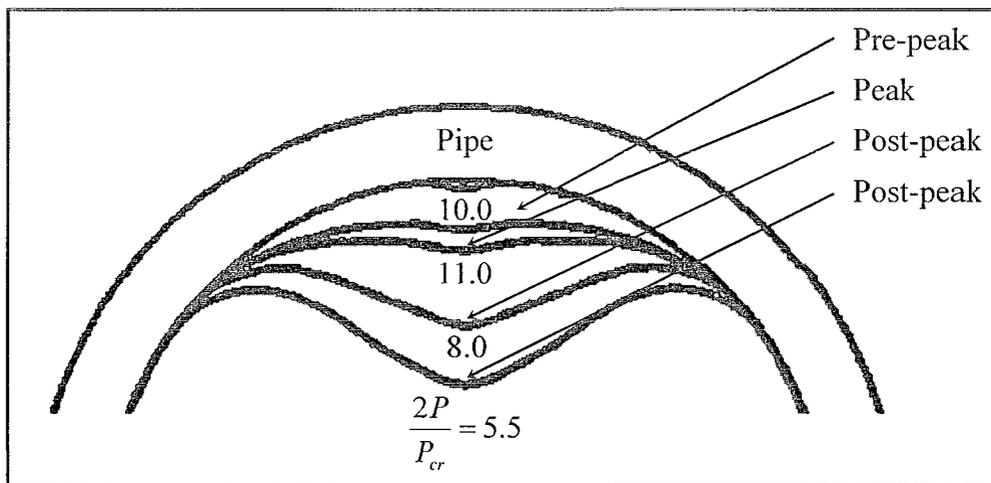


Figure 0.8 Post Buckling of Encased Liner

A careful examination of Chicurel's and Glock's equations reveals that both equations assume sinusoidal curves with constant amplitude. Polar coordinates system are adopted here, as shown in Figure 0.9. Here  $\theta = 0$  at  $OC$ ,  $\theta = \varphi$  at  $OA$ ,  $\theta = -\varphi$  at  $OB$  and  $OE = r_w(\theta)$ .

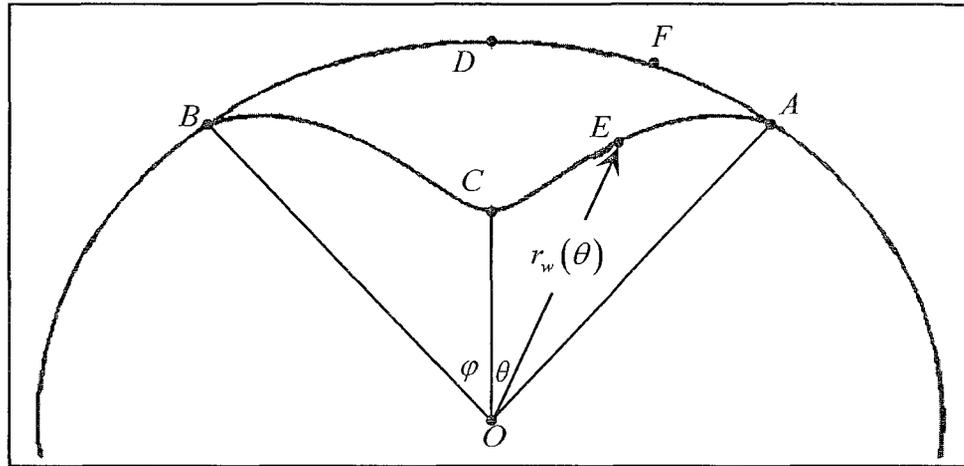


Figure 0.9 Polar Coordinates for Elliptical Host Pipe

In order to adapt to the elliptical boundary, varying amplitudes shall be imposed on the sinusoidal curve, which takes the form:

$$r_w(\theta) = (\text{Amplitude}(\theta))(\text{Sinusoidal}(\theta)).$$

This form can be met by many equations, one of which is proposed as Eq. (7.13):

$$r_w(\theta) = k_1 \left( -\frac{\theta^2}{k_2 \varphi^2} + 1 \right) \cos \left( \frac{\pi \theta}{2\varphi} \right) + k_3. \quad (7.13)$$

The first-order derivative of Eq. (7.13) is Eq. (7.14):

$$\frac{dr_w(\theta)}{d\theta} = \frac{k_1 \left[ \pi(\theta^2 - k_2\varphi^2) \sin\left(\frac{\pi\theta}{2\varphi}\right) - 4\theta\varphi \cos\left(\frac{\pi\theta}{2\varphi}\right) \right]}{2k_2\varphi^3}. \quad (7.14)$$

To determine  $k_1$ ,  $k_2$  and  $k_3$ , the boundary connectivity and smoothness condition, i.e., Eq. (7.15), Eq. (7.16), and Eq. (7.17) shall be satisfied:

$$r_w(\varphi) = OA, \quad (7.15)$$

$$r_w(0) = OC, \quad (7.16)$$

$$\frac{dr_w(\varphi)}{d\theta} = \frac{dr_e(\varphi)}{d\theta}. \quad (7.17)$$

The polar expression for an ellipse (Eq. (3.17)) is rewritten here as Eq. (7.18):

$$r_e(\theta) = \frac{r_1 r_2}{\sqrt{(r_1 \cos \theta)^2 + (r_2 \sin \theta)^2}}, \quad (7.18)$$

where  $r_1$ ,  $r_2$  are the semi-major and semi-minor axis, respectively; the first-order derivative of Eq. (7.18) is Eq. (7.19):

$$\frac{dr_e(\theta)}{d\theta} = \frac{r_1 r_2 (r_1^2 - r_2^2) \sin \theta \cos \theta}{\sqrt[3]{(r_1 \cos \theta)^2 + (r_2 \sin \theta)^2}}. \quad (7.19)$$

By substituting Eq. (7.15) and Eq. (7.16) into Eq. (7.13),  $k_3$  and  $k_1$  can be obtained as Eq. (7.20) and Eq. (7.21):

$$k_3 = OA = \frac{r_1 r_2}{\sqrt{(r_1 \cos \varphi)^2 + (r_2 \sin \varphi)^2}}, \quad (7.20)$$

$$k_1 = OC - OA = OC - \frac{r_1 r_2}{\sqrt{(r_1 \cos \varphi)^2 + (r_2 \sin \varphi)^2}}. \quad (7.21)$$

By substituting Eq. (7.14) and Eq. (7.19) into Eq. (7.17),  $k_2$  can be obtained as Eq. (7.22):

$$k_2 = \frac{1}{1 + \frac{2\varphi}{k_1\pi} \frac{r_1 r_2 (r_1^2 - r_2^2) \sin \varphi \cos \varphi}{\sqrt[3]{(r_1 \cos \varphi)^2 + (r_2 \sin \varphi)^2}}}. \quad (7.22)$$

In order to check if the proposed equation violates the common sense near the boundary as the Chicurel's equation does, very small values  $\varepsilon$  are added to  $\theta$  and it is found that  $r_e(\theta + \varepsilon)$  is always larger than  $r_w(\theta + \varepsilon)$ , so the wavy portion is well confined within the host pipe. By varying  $\varphi$  and  $OC$ , different shapes of the wavy portion can be formed. For illustrative purposes, it is assumed that  $r_1 = 12$  and  $r_2 = 10$ . In Figure 0.10, the maximum deflection  $OC = 8$ , then  $\varphi = 15^\circ$ ,  $\varphi = 30^\circ$  and  $\varphi = 45^\circ$  are applied to depict different buckled angles. In Figure 0.11, the buckled angle  $\varphi = 30^\circ$ , then  $OC = 6$ ,  $OC = 7$  and  $OC = 8$  are applied to depict different maximum deflections.

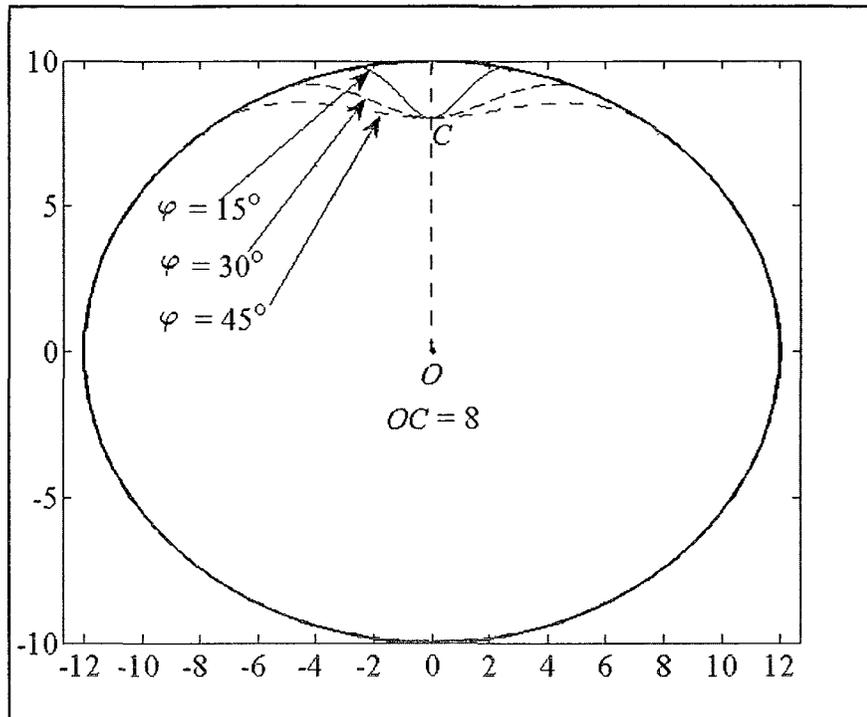


Figure 0.10 Different Wavy Shapes with Varying Buckled Angles

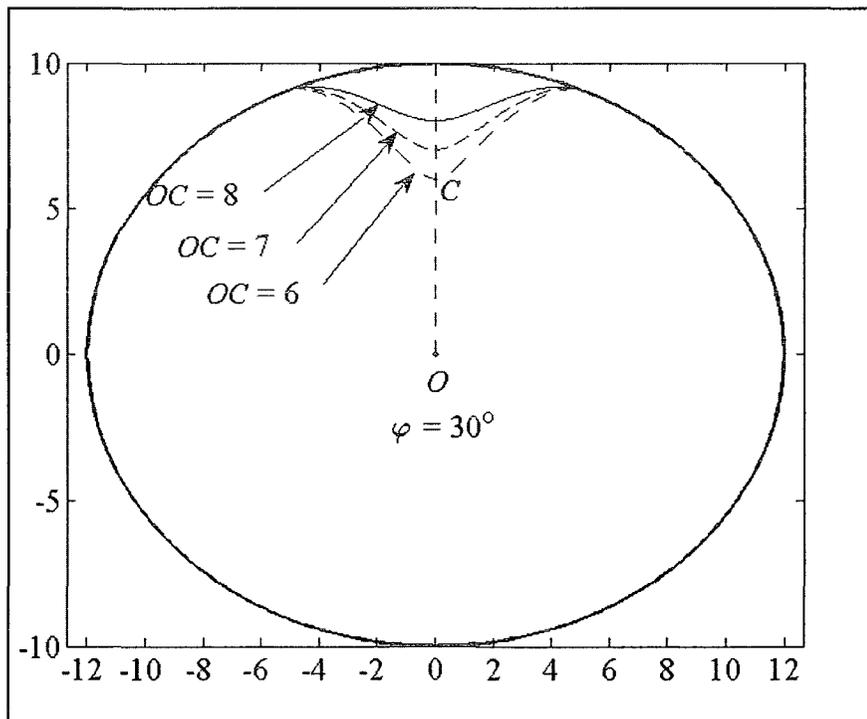


Figure 0.11 Different Wavy Shapes with Varying Maximum Deflection

A more generalized version of Eq. (7.13) replaces the power term  $\theta^2$  and  $\varphi^2$  by  $\theta^p$  and  $\varphi^p$ , as expressed in Eq. (7.23):

$$r_w(\theta) = k_1 \left( -\frac{\theta^p}{k_2 \varphi^p} + 1 \right) \cos\left(\frac{\pi\theta}{2\varphi}\right) + k_3. \quad (7.23)$$

In Figure 0.12, the buckled angle  $\varphi = 30^\circ$  and the maximum deflection  $OC = 8$ , then  $p = 2$ ,  $p = 4$ , and  $p = 6$  are applied to depict the effects of the different powers.

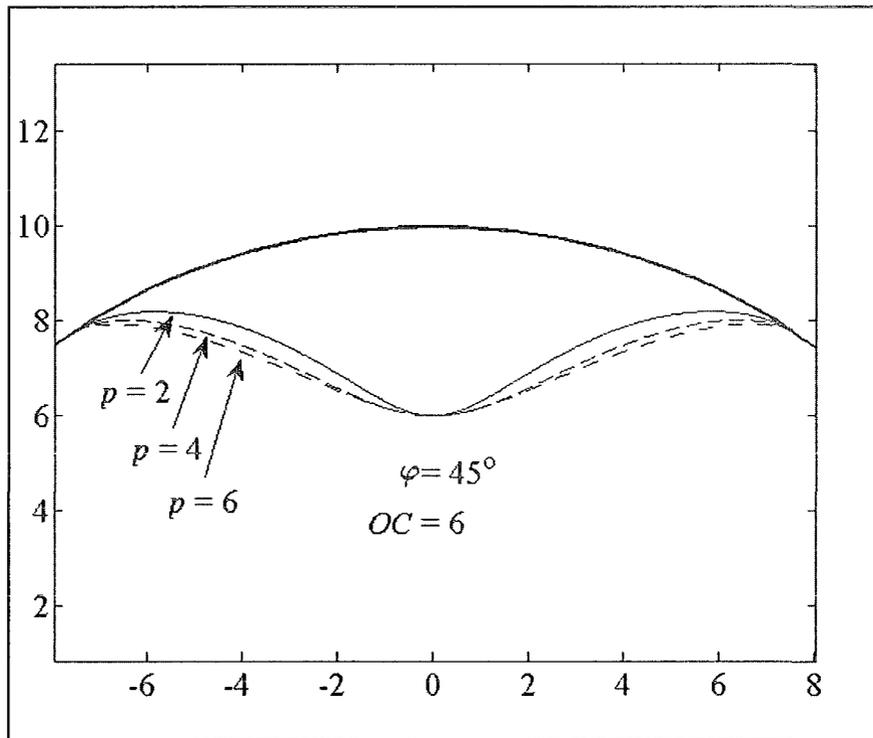


Figure 0.12 Different Wavy Shapes with Varying Powers

By observation, the proposed Eq. (7.13) gives a very good control on the modeling of the wavy portion, even for an elliptical boundary; if a finer control is necessary, Eq. (7.23) can be used with the expense of an additional power parameter.

3-D RGB plots can be made easily by controlling the buckled angle  $\varphi$ , the maximum deflection  $OC$  and the power  $p$  along the pipe length.  $r_1 = 11$ ,  $r_2 = 10$  and  $L = 20$  are used for the host pipe geometry. Figure 0.13 shows the top views of these plots. Constant, linearly varying and sinusoidally varying parameters are used from left to right columns. From top to bottom rows are: “ $\varphi = 30^\circ$  to  $60^\circ$ ,  $OC = 8$ ,  $p = 2$ ”, “ $\varphi = 45^\circ$ ,  $OC = 8$  to  $10$ ,  $p = 2$ ”, and “ $\varphi = 45^\circ$ ,  $OC = 8$ ,  $p = 2$  to  $8$ ”, in the same order as discussed in this section.

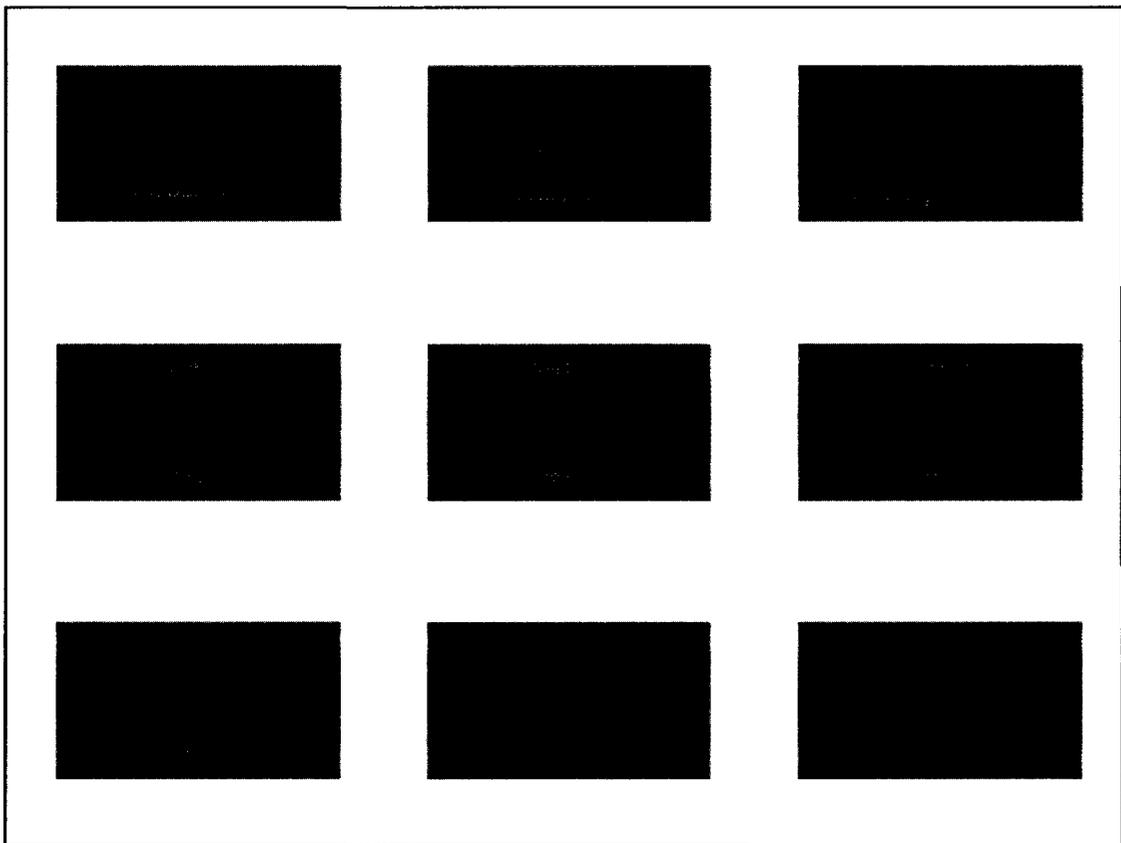


Figure 0.13 Top View of RGB Plots of Cross Sectional Modes

Based on previous research observations at Louisiana Tech University, two lobes can also occur in the early stage, simultaneously at the crown and the invert. As the buckling progresses, one of the lobes becomes dominant and keeps concaving, while the other lobe slowly reforms back to its original shape. A vivid reasoning for such a phenomenon is that for frictionless contact between the liner and the pipe, the dominant lobe “drags” more materials towards it, therefore changing the stress type at the less-dominant lobe from compressive stress, which causes the buckling, to tensile stress, which “pulls” the buckling portion to recover. But once the two lobes have both reached a significant deformation level or the friction is large enough to hold the liner from sliding, the recovery of the less-dominant lobe is less likely to happen. In this case, both lobes continue to develop.

Theoretically, multi-lobes can occur in a free ring buckling problem. However, only the single-lobed cross section is considered as representing the most common shape in the field. For a host pipe with a circular cross section, the lobe can develop anywhere around the circumference; for a host pipe with an elliptical cross section, the lobe usually develops near the crown or invert, where the buckling strength of the liner is the lowest.

#### **7.4 Chapter Summary**

This chapter models the cross sectional geometries of encased liners. Shapes before the liner touches the host pipe are modeled as a quasi-ellipse. Deficiencies of the wavy portion modeling equations after the liner develop into lobes are examined as non-flexible or discontinuous. A new modeling equation is proposed to overcome such limitations. Extension to double-lobe modeling with another lobe at the invert is straightforward. APPENDIX E gives some of the MATLAB® code for this chapter.

## **CHAPTER 8**

### **CONCLUSIONS AND RECOMMENDATIONS**

#### **8.1 Research Conclusions**

Underground pipelines, which are a critical component of both industrial and residential infrastructure systems, should be inspected regularly as a preventative measure to monitor deterioration and to minimize the chance of more costly and dangerous failures. Various technologies are available on the market or are being developed to enable users to better understand the pipe conditions and to make effective rehabilitation decisions. Among all the information that can be acquired, accurate pipe geometries are uniquely important for structural analysis of host pipes and for pipe liner design. However, the confidence in geometry measurements is low due to the inevitable deviation in a robot's path as it traverses a pipe and due to pipe irregularities.

This research addresses common pipe geometry measurement problems associated with robotic laser and camera inspection. These problems stem from the pipe geometry itself or are caused by robotic deviation inside the pipe. Each problem is addressed in an individual chapter as an attempt to provide relevant solutions, to enhance understanding, and to establish the groundwork for future research. The following conclusions can be drawn from this research:

1. Cross sectional geometries of originally round rigid pipes are studied based on a constant perimeter constraint. Common primitive cross sections include a full-circumference elliptical mode and a four-hinge circular mode. Composite modes are generalized from primitive modes to describe the deformation pathway of a perfectly round pipe going through varying elliptical and hinged modes, partially or wholly. 3-D geometries can be easily generated by adjusting the parameters along the pipe length. Other values such as ovality and area loss, which are widely used in the industry, are calculated for each mode. A more accurate mean diameter calculation using an elliptical integral is suggested to replace the traditional equation for pipe ovality that averages the maximum and minimum diameter.
2. Two isolated cases of measurement origin offsets are studied: the offset caused by pipe deformation without robotic deviation and the offset caused by robotic deviation without pipe deformation. The influences of dimensionless robotic length and wheels configurations on the offset are quantified. Equations for a 2-D laser ring shape in a bend are derived and validated.
3. A nonlinear robotic orientation correction algorithm is proposed for a single-ended laser scanner in an elliptical cylindrical pipe. The algorithm does not assume the major and minor axis length of the pipe is known. The algorithm simultaneously estimates robot orientation and the shape of the pipe. Also, an enhanced version of the algorithms for a double-ended scanner is given for accurate orientation estimation.

4. The matrix that transforms actual joints and laterals into contours in the image plane is presented. Five lens models are applied to simulate the contours. Relationships between geometry measurement and image processing are discussed. Lateral radius, distance from the camera to the lateral, and orientation of the camera are treated as three unknowns and are assumed to be coarsely estimated by geometry measurement. The impact of these three uncertainties are isolated and examined to validate the feasibility of geometry and image sensor fusion.
5. The two dimensional shape of encased liner cross sections is modeled to overcome the limitations of existing equations and to improve flexibility and smoothness at the boundary.
6. All mathematical modeling and algorithms are implemented using MATLAB®.

## **8.2 Recommendations for Future Work**

This research provided much needed groundwork for those who wish to expand the technologies used in robotic inspection of underground pipes. The author acknowledges that many unsolved problems remain; several of these problems are identified below for future research:

1. More mathematical models are needed for establishing a comprehensive geometry library for common pipe shapes, including inverted horseshoe, egg-shaped, and loaf. Ovality and area loss resulting from the unsymmetrical versions of the models presented in this research are more complex, but the underlying principles remain the same.

2. Hardware needs to be assembled to realize the robotic orientation and pipe dimension estimation algorithm. A mechanical device is needed to incrementally change the orientation angle of the laser ring.
3. The reflectance of a pipe's surface can be rough and diffuse, making it difficult for accurate edge detection. Mineral deposits and inward deformation both introduce difficulty in geometry measurement. Image processing techniques need to be tailored for underground pipe conditions for improved lateral contour extraction and deposit/deformation disambiguation.
4. It is desirable to develop powerful and user-friendly software that has built-in algorithms to preprocess and correct the possibly distorted geometry data to achieve an accurate 3-D wireframe. Incorporating texture mapping onto this wireframe geometry and enabling sensor fusion are needed to further the utility of pipe measurement systems.
5. Underground pipelines not only have interior deteriorations but also global layout shifts. Moreover, longitudinal robot position along the axis of a pipe based on either cable takeout or odometers is not satisfactory. Intelligent algorithms using maximum likelihood are needed to co-register two measured datasets of the same pipeline over time.

## **APPENDIX A**

### **MATLAB® CODE FOR CHAPTER 3**

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
CROSS SECTIONAL MODE 1
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

draw ellipse and circle
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
r = 10;
s = 0.9;
r2 = s*r;
r1 = r/3*(-2*s + 3 + sqrt(-5*s.^2 + 6*s + 3));
theta = 0:360;
z1 = r*cosd(theta);
y1 = r*sind(theta);
z2 = r1*cosd(theta);
y2 = r2*sind(theta);

plot(z1, y1, '-k', z2, y2, '-k', 0, 0, 'k')

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
area loss in the phase of mode 1
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
r = 10;
s = 0.5:0.01:1;
r2 = s*r;
r1 = r/3*(-2*s + 3 + sqrt(-5*s.^2 + 6*s + 3));
area_new = pi*r1.*r2;
ratio = (1 - area_new/(pi*r^2))*100;

plot(s, ratio, '-k', 'LineWidth', 1)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
average versus integral ovality
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
K = 7;
E = 72500;
v = 0.35;
N = 2;
Dm = 8;
t = 0.2462;
SDR = Dm/t;
P = 20;

```

```

min = 10/10;
max = [10 10.5 11 11.5 12]/10;

traditional_mean = (min + max)/2;
better_mean = [10 10.2454 10.4821 10.7107 10.9317]/10;

traditional_ovality = (max - traditional_mean)./traditional_mean*100;
better_ovality = (max - better_mean)./better_mean*100;

traditional_C = ((1 - traditional_ovality/100)./(1 + traditional_ovality/100).^2).^3;
better_C = ((1 - better_ovality/100)./(1 + better_ovality/100).^2).^3;

traditional_P = 2*K*E/(1 - v^2)/(SDR)^3*traditional_C/N;
better_P = 2*K*E/(1 - v^2)/(SDR)^3*better_C/N;

traditional_t = Dm*(P*(1 - v^2)*N./(2*K*E*traditional_C)).^(1/3);
better_t = Dm*(P*(1 - v^2)*N./(2*K*E*better_C)).^(1/3);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
CROSS SECTIONAL MODE 2
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

draw top left hinge as an example. the other three hinges just needs minor adjustments
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
r = 10;
s = 0.7;
OB = s*r;
yc = -r/2*(-s + sqrt(2 - s.^2));
zc = -r/2*(-s + sqrt(2 - s.^2));
alpha = -atan(abs(zc)/(OB + abs(yc)));

```

```

theta = pi/2:0.01:pi;
z = r*cos(theta);
y = r*sin(theta);
Z = cos(alpha)*z - sin(alpha)*y;
Y = sin(alpha)*z + cos(alpha)*y;
Z = Z + zc;
Y = Y + yc;

plot(Z, Y, '-k')

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
area loss in the phase of whole circular
                    hinged
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
r = 10;
s = 0.5:0.01:1;
OB = s*r;
CA = r;
alpha = atand((sqrt(2 - s.^2) - s)/(sqrt(2 -
s.^2) + s));
OE = OB.*tand(alpha);
area_OBE = 1/2*OB.*OE;
CE = CA.*tand(alpha);
area_CAE = 1/2*CA.*CE;
area_loss = 4*(area_CAE - area_OBE);
ratio = area_loss/(pi*r^2)*100;

plot(s, ratio, ':k', 'LineWidth', 1)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
true mean diameter of whole circular
                    hinged.
                    call function r =
whole_circular_hinged_r_true(s, theta)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
s = (1: -0.01:0.90)';
min = 2*s;
max = 2*sqrt(2 - s.^2);
traditional_mean = (min + max)/2;

Qv =
quadv(@(theta)whole_circular_hinged_r
_true(s, theta), pi/2, pi);
better_mean = 2*Qv/(pi/2);

```

```

traditional_ovality = (max -
traditional_mean)./traditional_mean;
better_ovality = (max -
better_mean)./better_mean;

```

```

traditional_C = ((1 -
traditional_ovality)/(1 +
traditional_ovality).^2).^3;
better_C = ((1 - better_ovality)/(1 +
better_ovality).^2).^3;

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

### **CROSS SECTIONAL MODE 3**

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
draw bottom ellipse, top circular hinges
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
m = 100; % m: angular resolution

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
draw bottom ellipse

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
r = 10;

```

```
s1 = 0.8;
```

```
r2 = s1*r;
```

```
r1 = r/3*(-2*s1 + 3 + sqrt(-5*s1.^2 +
6*s1 + 3));
```

```
theta = linspace(pi, 2*pi, 2*m);
```

```
z2 = r1*cos(theta);
```

```
y2 = r2*sin(theta);
```

```
plot(z2, y2, '-k')
```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
draw top left circular hinge as an
example, the other hinge just needs minor

```

adjustments

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

```
s2 = sqrt(2 - 1/9*(-2*s1 + 3 +
sqrt(-5*s1.^2 + 6*s1 + 3)).^2);
```

```
OB = s2*r;
```

```
yc = -r/2*(-s2 + sqrt(2 - s2.^2));
```

```
zc = -r/2*(-s2 + sqrt(2 - s2.^2));
```

```
alpha = -atan(abs(zc)/(OB + abs(yc)));
theta = linspace(pi/2, pi, m);
z = r*cos(theta);
y = r*sin(theta);
Z = cos(alpha)*z - sin(alpha)*y;
Y = sin(alpha)*z + cos(alpha)*y;
Z = Z + zc;
Y = Y + yc;
```

```
plot(Z, Y, '-k')
```

```
%%%%%%%%%%
circular hinged deformation severity
versus elliptical deformation severity
%%%%%%%%%%
s1 = 0.463432:0.01:1; ref = s1;
s2 = sqrt(2 - 1/9*(-2*s1 + 3 +
sqrt(-5*s1.^2 + 6*s1 + 3)).^2);
```

```
plot(s1, s2, '-k', s1, ref, ':k')
```

```
%%%%%%%%%%
area loss in the phase of bottom elliptical,
top circular hinged
%%%%%%%%%%
s1 = 0.5:0.01:1;
r2 = s1*r; r1 = r/3*(-2*s1 + 3 +
sqrt(-5*s1.^2 + 6*s1 + 3));
s2 = sqrt(2 - 1/9*(-2*s1 + 3 +
sqrt(-5*s1.^2 + 6*s1 + 3)).^2);
OB = s2*r;
CA = r; alpha = atand((sqrt(2 - s2.^2) -
s2)/(sqrt(2 - s2.^2) + s2));
OE = OB.*tand(alpha);
area_OBE = 1/2*OB.*OE;
CE = CA.*tand(alpha);
area_CAE = 1/2*CA.*CE;
area_loss_top = 2*(area_CAE -
area_OBE);
area_loss_bottom = 1/2*(pi*r^2 -
pi*r1.*r2);
area_loss = area_loss_top +
area_loss_bottom;
ratio = area_loss/(pi*r^2)*100;
```

```
plot(s1, ratio, '--k', 'LineWidth', 1)
```

```
%%%%%%%%%%
true mean diameter of bottom elliptical
and top circular hinged.
call function r =
bottom_elliptical_top_circular_hinged_r
_bottom(s, theta)
call function r =
bottom_elliptical_top_circular_hinged_r
_top(s, theta)
%%%%%%%%%%
s1 = (1: -0.01:0.90);
s2 = sqrt(2 - 1/9*(-2*s1 + 3 +
sqrt(-5*s1.^2 + 6*s1 + 3)).^2);
min = s1 + s2;
max = 2*1/3*(-2*s1 + 3 + sqrt(-5*s1.^2 +
6*s1 + 3));
traditional_mean = (min + max)/2;
```

```
Qv1 =
quadv(@(theta)bottom_elliptical_top_cir
cular_hinged_r_bottom(s1, theta), pi,
3*pi/2);
Qv2 =
quadv(@(theta)bottom_elliptical_top_cir
cular_hinged_r_top(s2, theta), pi/2, pi);
better_mean = (Qv1 + Qv2)/(pi/2);
```

```
traditional_ovality = (max -
traditional_mean)./traditional_mean;
better_ovality = (max -
better_mean)./better_mean;
```

```
traditional_C = ((1 -
traditional_ovality)/(1 +
traditional_ovality).^2).^3;
better_C = ((1 - better_ovality)/(1 +
better_ovality).^2).^3;
```

```
%%%%%%%%%%
%%%%%%%%%%
CROSS SECTIONAL MODE 4
%%%%%%%%%%
%%%%%%%%%%
```

```
%%%%%%%%%%
draw bottom ellipse, top elliptical hinges
```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
draw bottom ellipse
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
r = 10;
s1 = 0.5;
b = s1*r; a = r/3*(-2*s1 + 3 +
sqrt(-5*s1.^2 + 6*s1 + 3));
theta = 180:360;
z2 = a*cosd(theta);
y2 = b*sind(theta);

```

```
plot(z2, y2, '-k', 'LineWidth', 2)
```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
draw top left elliptical hinge as an
example, the other hinge just needs minor
adjustments
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
r = 10;
s02 = 0.7;
s2 = sqrt(1/9*(-2*s02 + 3 +
sqrt(-5*s02.^2 + 6*s02 + 3)).^2 + s02^2 -
1/9*(-2*s1 + 3 + sqrt(-5*s1.^2 + 6*s1 +
3)).^2);
BC = s02*r; % known variables
AC = 1/3*(-2*s02 + 3 + sqrt(-5*s02.^2 +
6*s02 + 3))*r; % known variables
OB = s2*r; % known variables
alpha_a = -atand(BC/AC);
alpha = abs(-asind(OB/sqrt(AC^2 +
BC^2)) - alpha_a); % absolute alpha
alpha = -alpha; % clockwise
z0 = -abs(BC*sind(alpha)); % z_center
y0 = -abs(BC*cosd(alpha) - OB);
a = 1/3*(-2*s02 + 3 + sqrt(-5*s02.^2 +
6*s02 + 3))*r;
b = s02*r;
theta = 90:1:180;
z = a*cosd(theta);
y = b*sind(theta);
Z = cosd(alpha)*z - sind(alpha)*y;
Y = sind(alpha)*z + cosd(alpha)*y;
Z = Z + z0;
Y = Y + y0;

```

```
plot(Z, Y, '-k', z0, y0, '.k', 'LineWidth', 2)
```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
elliptical hinged deformation severity
versus elliptical deformation severity
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
r = 10;
s02 = 1: -0.1:0.1;
lowerlimit = [0.463431, 0.457427,
0.440412, 0.413596, 0.377916, 0.334097,
0.282694, 0.224078, 0.158379,
0.0851487];

```

```
for i = 1:10
```

```

s1 = lowerlimit(i):0.01:s02(i);
s2 = sqrt(1/9*(-2*s02(i) + 3 +
sqrt(-5*s02(i).^2 + 6*s02(i) + 3)).^2
+ s02(i)^2 - 1/9*(-2*s1 + 3 +
sqrt(-5*s1.^2 + 6*s1 + 3)).^2);

```

```
plot(s1, s2, '-k', 'LineWidth', 2)
```

```
end
```

```
plot([0 1], [0 1], '.k', 'LineWidth', 2)
```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
true mean diameter of bottom elliptical
and top elliptical hinged.
call function r =
bottom_elliptical_top_circular_hinged_r
_bottom(s, theta)
call function F = polar_ellipse_r(s2, s02,
theta)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
s02 = 0.95;
s1 = (s02 - 1e-6: -0.01:0.89)';
s2 = sqrt(1/9*(-2*s02 + 3 +
sqrt(-5*s02.^2 + 6*s02 + 3)).^2 + s02^2 -
1/9*(-2*s1 + 3 + sqrt(-5*s1.^2 + 6*s1 +
3)).^2);
min = s1 + s2;
max = 2*1/3*(-2*s1 + 3 + sqrt(-5*s1.^2 +
6*s1 + 3));
traditional_mean = (min + max)/2;

```

```

Qv1 =
quadv(@(theta)bottom_elliptical_top_cir
cular_hinged_r_bottom(s1, theta), pi,
3*pi/2);
Qv2 = quadv(@(theta)polar_ellipse_r(s2,
s02, theta), pi/2, pi);
better_mean = (Qv1 + Qv2)/(pi/2);

```

```

traditional_ovality = (max -
traditional_mean)/traditional_mean;
better_ovality = (max -
better_mean)/better_mean;

```

```

traditional_C = ((1 -
traditional_ovality)/(1 +
traditional_ovality).^2).^3;
better_C = ((1 - better_ovality)/(1 +
better_ovality).^2).^3;

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
CROSS SECTIONAL MODE 5
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
draw bottom elliptical hinges, top
elliptical hinges
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
draw bottom left hinge as an example, the
other bottom hinge just needs minor
adjustments
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

```

r = 10;
s01 = 0.6;
s1 = 0.4;
BC = s01*r; % known variables
AC = 1/3*(-2*s01 + 3 + sqrt(-5*s01.^2 +
6*s01 + 3))*r; % known variables
OB = s1*r; % known variables
alpha_a = -atand(BC/AC);
alpha = abs(-asind(OB/sqrt(AC^2 +
BC^2)) - alpha_a); % absolute alpha
BE = OB/cosd(alpha);

```

```

zc = -abs(BC*sind(alpha)); % z_center
yc = abs(BC*cosd(alpha) - OB);
r = 10; a = 1/3*(-2*s01 + 3 +
sqrt(-5*s01.^2 + 6*s01 + 3))*r; b =
s01*r;
theta = 180:1:270;
z = a*cosd(theta);
y = b*sind(theta);
Z = cosd(alpha)*z - sind(alpha)*y;
Y = sind(alpha)*z + cosd(alpha)*y;
Z = Z + zc;
Y = Y + yc;

```

```

plot(Z, Y, '-k', 'LineWidth', 2)

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
draw top left hinge, the other top hinge
just needs minor adjustments
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
s02 = 0.8;

```

```

s2 = sqrt(s1.^2 + 1/9*(-2*s02 + 3 +
sqrt(-5*s02.^2 + 6*s02 + 3)).^2 + s02^2 -
1/9*(-2*s01 + 3 + sqrt(-5*s01.^2 + 6*s01
+ 3)).^2 - s01^2);

```

```

BC = s02*r; % known variables
AC = 1/3*(-2*s02 + 3 + sqrt(-5*s02.^2 +
6*s02 + 3))*r; % known variables
OB = s2*r; % known variables

```

```

alpha_a = -atand(BC/AC);
alpha = abs(-asind(OB/sqrt(AC^2 +
BC^2)) - alpha_a); % absolute alpha
alpha = -alpha; % clockwise
zc = -abs(BC*sind(alpha)); % z_center
yc = -abs(BC*cosd(alpha) - OB);

```

```

r = 10; a = 1/3*(-2*s02 + 3 +
sqrt(-5*s02.^2 + 6*s02 + 3))*r;
b = s02*r;

```

```

theta = 90:1:180;
z = a*cosd(theta);
y = b*sind(theta);
Z = cosd(alpha)*z - sind(alpha)*y;
Y = sind(alpha)*z + cosd(alpha)*y;
Z = Z + zc;
Y = Y + yc;

```

```

plot(Z, Y, '-k', zc, yc, 'k', 'LineWidth', 2)

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    elliptical hinged deformation severity
    versus another level of elliptical hinged
    deformation severity
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
s01 = 0.8; s02 = s01:0.1:1;

for i = 1:size(s02, 2)

    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    top and bottom both elliptical hinged
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    lowerlimit = sqrt(-1/9*(-2*s02(i) + 3
    + sqrt(-5*s02(i).^2 + 6*s02(i) +
    3)).^2 - s02(i)^2 + 1/9*(-2*s01 + 3 +
    sqrt(-5*s01.^2 + 6*s01 + 3)).^2 +
    s01^2);
    s1 = lowerlimit:0.001:s01;
    s2 = sqrt(s1.^2 + 1/9*(-2*s02(i) + 3 +
    sqrt(-5*s02(i).^2 + 6*s02(i) + 3)).^2
    + s02(i)^2 - 1/9*(-2*s01 + 3 +
    sqrt(-5*s01.^2 + 6*s01 + 3)).^2 -
    s01^2);

    plot(s1, s2, 'b', 'LineWidth', 4)

    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    top elliptical hinged, bottom elliptical
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    s1 = s01:0.001:s02(i);
    s2 = sqrt(1/9*(-2*s02(i) + 3 +
    sqrt(-5*s02(i).^2 + 6*s02(i) + 3)).^2
    + s02(i)^2 - 1/9*(-2*s1 + 3 +
    sqrt(-5*s1.^2 + 6*s1 + 3)).^2);
    plot(s1, s2, 'g', 'LineWidth', 2)

end

plot([s01 1], [s01 1], '--r', 'LineWidth', 2)

    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    area loss in the phase of top and bottom
    both elliptical hinged
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
r = 10;
s02 = 0.9; s01 = 0.7;

s1 = 0.5:0.005:s01;
b = s01*r;
a = r/3*(-2*s01 + 3 + sqrt(-5*s01.^2 +
6*s01 + 3));
b2 = s02*r;
a2 = r/3*(-2*s02 + 3 + sqrt(-5*s02.^2 +
6*s02 + 3));

s2 = sqrt(s1.^2 + 1/9*(-2*s02 + 3 +
sqrt(-5*s02.^2 + 6*s02 + 3)).^2 + s02^2 -
1/9*(-2*s01 + 3 + sqrt(-5*s01.^2 + 6*s01
+ 3)).^2 - s01^2);

BC = s02*r;
AC = 1/3*(-2*s02 + 3 + sqrt(-5*s02.^2 +
6*s02 + 3))*r;
OB = s2*r; % known variables
alpha_a = -atand(BC/AC);
alpha = abs(-asind(OB/sqrt(AC^2 +
BC^2)) - alpha_a); % absolute alpha
OE = OB.*tand(alpha);
area_OBE = 1/2*OB.*OE;
CE = AC.*tand(alpha);
area_CAE = 1/2*AC.*CE;
area_loss_top = 2*(area_CAE -
area_OBE) + 1/2*(pi*r^2 - pi*a2.*b2);

BC = s01*r;
AC = 1/3*(-2*s01 + 3 + sqrt(-5*s01.^2 +
6*s01 + 3))*r;
OB = s1*r; % known variables
alpha_a = -atand(BC/AC);
alpha = abs(-asind(OB/sqrt(AC^2 +
BC^2)) - alpha_a); % absolute alpha
OE = OB.*tand(alpha);
area_OBE = 1/2*OB.*OE;
CE = AC.*tand(alpha);
area_CAE = 1/2*AC.*CE;
area_loss_bottom = 2*(area_CAE -
area_OBE) + 1/2*(pi*r^2 - pi*a.*b);

area_loss = area_loss_top +
area_loss_bottom;
ratio = area_loss/(pi*r^2)*100;

plot(s1, ratio, 'k', 'LineWidth', 1)

```

```

%%%%%%%%%%
area loss in the phase of top elliptical
hinged, bottom elliptical
%%%%%%%%%%
r = 10;
s1 = s01:0.005:s02;
b = s1*r;
a = r/3*(-2*s1 + 3 + sqrt(-5*s1.^2 + 6*s1
+ 3));
b2 = s02*r;
a2 = r/3*(-2*s02 + 3 + sqrt(-5*s02.^2 +
6*s02 + 3));
s2 = sqrt(1/9*(-2*s02 + 3 +
sqrt(-5*s02.^2 + 6*s02 + 3)).^2 + s02^2 -
1/9*(-2*s1 + 3 + sqrt(-5*s1.^2 + 6*s1 +
3)).^2);
BC = s02*r;
AC = 1/3*(-2*s02 + 3 + sqrt(-5*s02.^2 +
6*s02 + 3))*r;
OB = s2*r; % known variables
alpha_a = -atand(BC/AC);
alpha = abs(-asind(OB/sqrt(AC^2 +
BC^2)) - alpha_a); % absolute alpha
OE = OB.*tand(alpha);
area_OBE = 1/2*OB.*OE;
CE = AC.*tand(alpha);
area_CAE = 1/2*AC.*CE;

area_loss_top = 2*(area_CAE -
area_OBE) + 1/2*(pi*r^2 - pi*a2.*b2);
area_loss_bottom = 1/2*(pi*r^2 -
pi*a.*b);
area_loss = area_loss_top +
area_loss_bottom;
ratio = area_loss/(pi*r^2)*100;

plot(s1, ratio, 'k', 'LineWidth', 1)

%%%%%%%%%%
true mean diameter of whole elliptical
hinged.
call function F = polar_ellipse_r(s2, s02,
theta)
%%%%%%%%%%
s02 = 0.95;
s01 = 0.92;

```

```

s1 = (s01 - 1e-6: -0.01:0.89)';
s2 = sqrt(s1.^2 + 1/9*(-2*s02 + 3 +
sqrt(-5*s02.^2 + 6*s02 + 3)).^2 + s02^2 -
1/9*(-2*s01 + 3 + sqrt(-5*s01.^2 + 6*s01
+ 3)).^2 - s01^2);
min = s1 + s2;
max = 2*(1/3*(-2*s02 + 3 +
sqrt(-5*s02.^2 + 6*s02 +
3))./cos(alpha(s2, s02))) +
s2.*tan(alpha(s2, s02)));
traditional_mean = (min + max)/2;

Qv1 = quadv(@(theta)polar_fit_r(s1, s01,
theta), pi/2, pi);
Qv2 = quadv(@(theta)polar_fit_r(s2, s02,
theta), pi/2, pi);
better_mean = (Qv1 + Qv2)/(pi/2);

traditional_ovality = (max -
traditional_mean)./traditional_mean;
better_ovality = (max -
better_mean)./better_mean;

traditional_C = ((1 -
traditional_ovality)/(1 +
traditional_ovality).^2).^3;
better_C = ((1 - better_ovality)/(1 +
better_ovality).^2).^3;

```

```

%%%%%%%%%%
%%%%%%%%%%
FUNCTION R =
WHOLE_CIRCULAR_HINGED_R_T
RUE(S, THETA)
%%%%%%%%%%
%%%%%%%%%%
function r =
whole_circular_hinged_r_true(s, theta)
r = r0(s).*cos(theta - theta0(s)) + sqrt(1^2
- (r0(s).*sin(theta - theta0(s))).^2);
function r0 = r0(s)
r0 = sqrt(2)/2*(-s + sqrt(2 - s.^2));
function theta0 = theta0(s)
theta0 = 5*pi/4;

```

```

%%%%%%%%%%
%%%%%%%%%%
FUNCTION R =
BOTTOM_ELLIPTICAL_TOP_CIRC
ULAR_HINGED_R_BOTTOM(S,
THETA)
%%%%%%%%%%
%%%%%%%%%%
function r =
bottom_elliptical_top_circular_hinged_r
_bottom(s, theta)
r = r1(s).*r2(s)./sqrt((r1(s)*sin(theta)).^2
+ (r2(s)*cos(theta)).^2);
function r1 = r1(s)
r1 = 1/3*(-2*s + 3 + sqrt(-5*s.^2 + 6*s +
3));
function r2 = r2(s)
r2 = s;

```

```

%%%%%%%%%%
%%%%%%%%%%
FUNCTION R =
BOTTOM_ELLIPTICAL_TOP_CIRC
ULAR_HINGED_R_UP(S, THETA)
%%%%%%%%%%
%%%%%%%%%%
function r =
bottom_elliptical_top_circular_hinged_r
_top(s, theta)
r = r0(s).*cos(theta - theta0(s)) + sqrt(1^2
- (r0(s).*sin(theta - theta0(s))).^2);
function r0 = r0(s)
r0 = sqrt(2)/2*(-s + sqrt(2 - s.^2));
function theta0 = theta0(s)
theta0 = 5*pi/4;

```

```

%%%%%%%%%%
%%%%%%%%%%
FUNCTION F =
POLAR_ELLIPSE_R(S2, S02,
THETA)
%%%%%%%%%%
%%%%%%%%%%
s2: vector, s02: scalar
%%%%%%%%%%

```

```

function F = polar_ellipse_r(s2, s02,
theta)
F = (polar_ellipse_P(s2, s02, theta) +
polar_ellipse_Q(s2, s02,
theta))./polar_ellipse_R(s2, s02, theta);
function F = polar_ellipse_P(s2, s02,
theta)
F = r0(s2, s02).*((b(s02)^2 -
a(s02)^2)*cos(theta + theta0(s2, s02) -
2*alpha(s2, s02)) + (a(s02)^2 +
b(s02)^2)*cos(theta - theta0(s2, s02)));
function F = polar_ellipse_Q(s2, s02,
theta)
F =
sqrt(2)*a(s02)*b(s02)*sqrt(polar_ellipse
_R(s2, s02, theta) - 2*r0(s2,
s02).^2.*(sin(theta - theta0(s2,
s02))).^2);
function F = polar_ellipse_R(s2, s02,
theta)
F = (b(s02)^2 - a(s02)^2)*cos(2*theta -
2*alpha(s2, s02)) + a(s02)^2 + b(s02)^2;

```

```

%%%%%%%%%%
the five parameters [a; b; r0; theta0;
alpha]

```

```

%%%%%%%%%%
function a = a(s02)
a = 1/3*(-2*s02 + 3 + sqrt(-5*s02.^2 +
6*s02 + 3));
function b = b(s02)
b = s02;
function r0 = r0(s2, s02)
r0 = sqrt((BC(s02)*sin(alpha(s2,
s02))).^2 + (BC(s02)*cos(alpha(s2, s02))
- s2).^2);
function theta0 = theta0(s2, s02)
theta0 = pi + atan((BC(s02)*cos(alpha(s2,
s02)) - s2)./(BC(s02)*sin(alpha(s2,
s02))));
% clockwise rotation
function alpha = alpha(s2, s02)
alpha = atan(BC(s02)/AC(s02)) -
asin(s2/sqrt(AC(s02)^2 + BC(s02)^2));

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
  supporting functions: scalar function
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function AC = AC(s02)
AC = 1/3*(-2*s02 + 3 + sqrt(-5*s02.^2 +
6*s02 + 3));
function BC = BC(s02)
BC = s02;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
DRAW 3- D RGB
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
use the z and y coordinates for each mode
to calculate the radial distance from
origin to inside surface
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

radial = sqrt(z.^2 + y.^2);

```

```

maximum = max(max(radial));
minimum = min(min(radial));

red = (radial - r)/(maximum - r);
red1 = red;
red1(radial <= r) = 0;
green1 = 1 - red1;
green1(radial <= r) = 0;
blue1 = zeros(size(t));

blue = (r - radial)/(r - minimum);
blue2 = blue;
blue2(radial > r) = 0;
green2 = 1 - blue2;
green2(radial > r) = 0;
red2 = zeros(size(t));

C(:, :, 1) = red1 + red2;
C(:, :, 2) = green1 + green2;
C(:, :, 3) = blue1 + blue2;
figure, h = surf(x, y, z, C);

```

## **APPENDIX B**

### **MATLAB® CODE FOR CHAPTER 4**

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
DEFORMATION INDUCED OFFSET
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
O_f: origin offset
r: pipe's radius
s: deformation indicator
W: half wheel span
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
r = 10;
[s,ratio] = meshgrid(0:0.01:1,0:0.01:1);
r2 = s*r; r1 = r/3*(-2*s + 3 + sqrt(-5*s.^2
+ 6*s + 3));
W = ratio*r;
O_f = sqrt(r.^2 - W.^2) - r2./r1.*sqrt(r1.^2
- W.^2);
ind = find(abs(O_f)>r2);
O_f(ind) = NaN;

```

```
figure, h = surf(s,ratio,O_f/r);
```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
PITCH INDUCED OFFSET
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
O_f: origin offset
Rw_ratio: wheel's radius
r: pipe's radius
theta_v: pitch
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
r = 1;
theta_v = 0:1:30;
Rw_ratio = 0:0.01:0.25;
L_ratio = 1;
xmin = min(theta_v);
xmax = max(theta_v);
ymin = min(Rw_ratio);
ymax = max(Rw_ratio);
[theta_v,Rw_ratio] =
meshgrid(theta_v,Rw_ratio);

```

```

O_f = (L_ratio - (1 -
Rw_ratio).*tand(theta_v/2))*r.*
sind(theta_v);
ind = (abs(O_f)>r | O_f<0);
O_f(ind) = NaN;

```

```
figure, h = surf(theta_v,Rw_ratio,O_f);
```

```

r = 1;
theta_v = 0:1:30;
Rw_ratio = 0.4;
L_ratio = 4*Rw_ratio:0.01:2;
xmin = min(theta_v);
xmax = max(theta_v);
ymin = min(L_ratio);
ymax = max(L_ratio);
[theta_v,L_ratio] =
meshgrid(theta_v,L_ratio);
O_f = (L_ratio - (1 -
Rw_ratio).*tand(theta_v/2))*r.*
sind(theta_v);
ind = (abs(O_f)>r | O_f<0);
O_f(ind) = NaN;

```

```
figure, h = surf(theta_v,L_ratio,O_f);
```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
OBLIQUE SHAPE CAUSED BY
DEVIATION
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
oblique shape caused by pitch
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
r = 10;
R = 20;

for theta = 5:5:25

c = R*sind(theta);
d = 2*(r.^2 + R.^2 - c.^2);
e = 2*(r.^2 - R.^2 - c.^2);
f = -(r + R + c)*(r + R - c)*(r - R +
c)*(r - R - c);

```



```

for pitch = 5:5:30

    sect = 30;
    t = zeros(4,sect - 1);
    r1 = 10;
    r2 = 10;
    R_b = 100;
    k = (r2^2 - r1^2)/r1^2;
    x0 = 0;
    y0 = -R_b;
    z0 = 0;

    n = tand(yaw).*(1 + tand(pitch).^2 +
    tand(yaw).^2).^(-1/2);
    m = tand(pitch).*(1 + tand(pitch).^2
    + tand(yaw).^2).^(-1/2);
    l = (1 + tand(pitch).^2 +
    tand(yaw).^2).^(-1/2);
    normal = [l m n];
    x_axis = [1 0 0];

    l1 = -sqrt(1 - l^2)*m/sqrt(m^2 + n^2);
    m1 = 1 + (1 - l)*n^2/(m^2 + n^2);
    n1 = -(1 - l)*m*n/(m^2 + n^2);
    l2 = -sind(pitch);
    m2 = cosd(pitch);
    n2 = 0;

    roll = -acosd(dot([l1 m1 n1], [l2 m2
    n2]));

    if l == 1

        alpha = linspace(0,pi,sect);
        alpha = alpha(1:end - 1);
        vector = [zeros(1,size(alpha,2));
        sin(alpha); cos(alpha)];

    else

        u = cross(normal, x_axis);
        u = u/norm(u);
        ux = u(1);
        uy = u(2);
        uz = u(3);

        theta = -acos(l);
        C = cos(theta);
        S = sin(theta);

        R = [C + (1 - C)*ux^2,
        (1 - C)*uy*ux - S*uz,
        (1 - C)*uz*ux + S*uy;
        (1 - C)*ux*uy + S*uz
        C + (1 - C)*uy^2
        (1 - C)*uz*uy - S*ux;
        (1 - C)*ux*uz - S*uy
        (1 - C)*uy*uz + S*ux
        C + (1 - C)*uz^2];

        alpha = linspace(0,pi,sect);
        alpha = alpha(1:end - 1);
        vector = [zeros(1,size(alpha,2));
        sin(alpha); cos(alpha)];
        vector = R*vector;

    end

    u = vector(1,:);
    v = vector(2,:);
    w = vector(3,:);

    c4 = (1 + k*w.^2).^2;
    c3 = 4*(1 + k*w.^2).*((u*x0 + v*y0
    + w*z0) + z0*w*k);
    c2 = 4*((u*x0 + v*y0 + w*z0) +
    z0*w*k).^2 + 2*(1 + k*w.^2).*(x0^2
    + y0^2 + z0^2 + z0^2*k - r2^2 -
    R_b^2) + 4*(r2*R_b/r1)^2*w.^2;
    c1 = 4*((u*x0 + v*y0 + w*z0) +
    z0*w*k)*(x0^2 + y0^2 + z0^2 +
    z0^2*k - r2^2 - R_b^2) +
    8*(r2*R_b/r1)^2*w*z0;
    c0 = (x0^2 + y0^2 + z0^2 + z0^2*k -
    r2^2 - R_b^2)^2 - 4*r2^2*R_b^2 +
    4*(r2*R_b/r1)^2*z0.^2;

    c = [c4; c3; c2; c1;
    c0*ones(size(c4))];

    for j = 1:sect - 1

```

```

    t(:,j) = roots(c(:,j));
end

t_hit = t(abs(imag(t))<1e - 6 &
abs(real(t))<20);
u_prepare = [u; u];
u_hit = u_prepare(:);
v_prepare = [v; v];
v_hit = v_prepare(:);
w_prepare = [w; w];
w_hit = w_prepare(:);
x = x0 + u_hit.*t_hit;
y = y0 + v_hit.*t_hit;
z = z0 + w_hit.*t_hit;

if l == 1

    x = x - x0;
    y = y - y0;
    z = z - z0;

else

    theta = -theta;
    C = cos(theta);
    S = sin(theta);

    R = [C + (1 - C)*ux^2,
(1 - C)*uy*ux - S*uz,
(1 - C)*uz*ux + S*uy;
(1 - C)*ux*uy + S*uz
C + (1 - C)*uy^2
(1 - C)*uz*uy - S*ux;
(1 - C)*ux*uz - S*uy
(1 - C)*uy*uz + S*ux
C + (1 - C)*uz^2];

    x = x - x0;
    y = y - y0;
    z = z - z0;
    B = R*[x';y';z'];
    y = B(2,:);
    z = B(3,:);

end

y = cosd(roll)*B(2,:) -
sind(roll)*B(3,:);
z = sind(roll)*B(2,:) +
cosd(roll)*B(3,:);

plot(z,y,'ok')

end

```

## **APPENDIX C**

### **MATLAB® CODE FOR CHAPTER 5**

```
%%%%%%%%%%
%%%%%%%%%%
```

**THEORETICAL ELLIPSE  
PARAMETERS:**

**SEMI AXES AND BETA**

```
%%%%%%%%%%
%%%%%%%%%%
```

```
r1 = 11;
r2 = 10;
n = 1000;
i = 0;
position = 0;
rmajor_plot = zeros(1,n);
rminor_plot = zeros(1,n);
beta_plot = zeros(1,n);
position_plot = zeros(1,n);
dotproduct = zeros(1,n);
pitch_plot = zeros(1,n);
yaw_plot = zeros(1,n);

for yaw = 0*pi/180:1*pi/180:30*pi/180

    pitch = 0.5*yaw;
    roll = 0.5*yaw;

    a = (1 + tan(pitch).^2 +
tan(yaw).^2).^(-1/2);
    b = tan(pitch).*(1 + tan(pitch).^2 +
tan(yaw).^2).^(-1/2);
    c = tan(yaw).*(1 + tan(pitch).^2 +
tan(yaw).^2).^(-1/2);
    normal = [a b c];
    x_axis = [1 0 0];

    A = r1^2*(b^2*c^2 - a^2)/(a^2 +
b^2)^2 + r2^2;
    B = -2*a*b*c*r1^2/(a^2 + b^2)^2;
    C = a^2 + b^2;
    D = r1^2*a^2/(a^2 + b^2);

    E = 4*(A^2 + B^2);
    F = -2*C*(A^2 + B^2);
    G = B^2*C^2;

    x1 = 1/E*(-F + sqrt(F^2 - E*G));
    x2 = 1/E*(-F - sqrt(F^2 - E*G));
```

```
w1 = sqrt(x2);
w2 = -sqrt(x1);
if (2*x2 - C)*B/(2*A)> = 0
```

```
w1 = sqrt(x2);
w2 = -sqrt(x1);
```

```
else
```

```
w1 = sqrt(x1);
w2 = -sqrt(x2);
```

```
end
```

```
v1 = 1/(a^2 + b^2)*(-w1*b*c +
a*sqrt(-w1.^2 + a^2 + b^2));
v2 = 1/(a^2 + b^2)*(-w2*b*c +
a*sqrt(-w2.^2 + a^2 + b^2));
u1 = -1/a*(v1*b + w1*c);
u2 = -1/a*(v2*b + w2*c);
```

```
if a == 1
```

```
    vector0 = [0; 0; 1];
```

```
else
```

```
    u = cross(normal, x_axis);
    u = u/norm(u);
    ux = u(1);
    uy = u(2);
    uz = u(3);
    theta = -acos(a);
    C = cos(theta);
    S = sin(theta);
```

```
    R = [C + (1 - C)*ux^2,
(1 - C)*uy*ux - S*uz,
(1 - C)*uz*ux + S*uy;
(1 - C)*ux*uy + S*uz
C + (1 - C)*uy^2
(1 - C)*uz*uy - S*ux;
(1 - C)*ux*uz - S*uy
(1 - C)*uy*uz + S*ux
C + (1 - C)*uz^2];
```

```

    vector0 = [0; 0; 1];
    vector0 = R*vector0;

end

w0 = vector0(3);
v0 = vector0(2);
u0 = vector0(1);

i = i + 1;

A2 = (r1*v2).^2 + (r2*w2).^2;
B2 = 0;
C2 = -(r1*r2)^2;

tminor = 2*r1*r2/sqrt(A2);

A1 = (r1*v1).^2 + (r2*w1).^2;
B1 = 0;
C1 = -(r1*r2)^2;

tmajor = 2*r1*r2/sqrt(A1);

rminor_plot(i) = tminor/2;
rmajor_plot(i) = tmajor/2;

dotproduct(i) = dot([u1 v1 w1], [u0
v0 w0]);

if tminor >= tmajor

    rminor_plot(i) = tmajor/2;
    rmajor_plot(i) = tminor/2;
    dotproduct(i) = dot([u2 v2 w2],
[u0 v0 w0]);

end

beta_plot(i) =
abs(acosd(abs(dotproduct(i))));

if abs(beta_plot(i)) < 1e - 5

    beta_plot(i) = 0;

end

```

```

    position_plot(i) = position;
    position = position + 5;
    pitch_plot(i) = pitch;
    yaw_plot(i) = yaw;

end

position_plot = position_plot(1:i);
rmajor_plot = rmajor_plot(1:i);
rminor_plot = rminor_plot(1:i);
beta_plot = beta_plot(1:i);
pitch_plot = pitch_plot(1:i)*180/pi;
yaw_plot = yaw_plot(1:i)*180/pi;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
SINGLE SCANNER ALGORITHM
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
iteration caller.
make a starting guess x0 = [20; 10; 10;
12; 9] of the true solution
[yaw; pitch; roll; r1; r2] = [15; 7.5; 7.5;
11; 10].
use previous code in this section for
rmajor_plot, rminor_plot, and beta_plot.
% first ring at yaw = 15
% second ring at yaw = 25
% third ring at yaw = 30
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

global rmajor1 rminor1 beta1
global rmajor2 rminor2 beta2
global rmajor3 rminor3 beta3

rmajor1 = rmajor_plot(16);
rminor1 = rminor_plot(16);
beta1 = beta_plot(16);
rmajor2 = rmajor_plot(16 + 10);
rminor2 = rminor_plot(16 + 10);
beta2 = beta_plot(16 + 10);
rmajor3 = rmajor_plot(16 + 15);
rminor3 = rminor_plot(16 + 15);
beta3 = beta_plot(16 + 15);

```

```

x0 = [20; 10; 10; 12; 9]; % Make a
starting guess
options = optimset('Display','iter -
detailed'); % Option to display output
[x,fval] =
fsolve(@yaw_pitch_r1_r2_roll_retrieval,
x0,options) % Call solver

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function F =
    yaw_pitch_r1_r2_roll_retrieval(x)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function F =
yaw_pitch_r1_r2_roll_retrieval(x)

global rmajor1 rminor1 beta1
global rmajor2 rminor2 beta2
global rmajor3 rminor3 beta3

F = [1/(calculate_v1(x, beta1)^2/x(5)^2 +
calculate_w1(x, beta1)^2/x(4)^2) -
rmajor1^2;
1/(calculate_v2(x, beta1)^2/x(5)^2 +
calculate_w2(x, beta1)^2/x(4)^2) -
rminor1^2;
1/(calculate_v1(x + [10; 0.5*10; 0.5*10;
0; 0], beta2)^2/x(5)^2 + calculate_w1(x +
[10; 0.5*10; 0.5*10; 0; 0],
beta2)^2/x(4)^2) - rmajor2^2;
1/(calculate_v2(x + [10; 0.5*10; 0.5*10;
0; 0], beta2)^2/x(5)^2 + calculate_w2(x +
[10; 0.5*10; 0.5*10; 0; 0],
beta2)^2/x(4)^2) - rminor2^2;
1/(calculate_v1(x + [15; 0.5*15; 0.5*15;
0; 0], beta3)^2/x(5)^2 + calculate_w1(x +
[15; 0.5*15; 0.5*15; 0; 0],
beta3)^2/x(4)^2) - rmajor3^2];

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function v1 = calculate_v1(x, beta)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function v1 = calculate_v1(x, beta)
v1 = 1/(a(x).^2 + b(x).^2).*(-w11(x,
beta).*b(x).*c(x) + a(x).*sqrt(-w11(x,
beta).^2 + a(x).^2 + b(x).^2));

```

```

function w11 = w11(x, beta)
w11 = (-F1(x, beta) - sqrt(abs(F1(x,
beta).^2 - E(x).*G1(x, beta))))./E(x);
function E = E(x)
E = A(x).^2 + B(x).^2;
function F1 = F1(x, beta)
F1 = -B(x)*D1(x, beta);
function G1 = G1(x, beta)
G1 = D1(x, beta).^2 - A(x).^2.*C(x);
function A = A(x)
A = b(x).*c(x) - a(x).*b(x).*c(x);
function B = B(x)
B = a(x).*c(x).^2 + b(x).^2;
function C = C(x)
C = a(x).^2 + b(x).^2;
function D1 = D1(x, beta)
D1 = (a(x).^2 + b(x).^2).*(b(x).^2 +
c(x).^2).*cosd(beta - x(3));
function a = a(x)
a = (1 + tand(x(2))).^2 +
tand(x(1)).^2.^(-1/2);
function b = b(x)
b = tand(x(2)).*(1 + tand(x(2))).^2 +
tand(x(1)).^2.^(-1/2);
function c = c(x)
c = tand(x(1)).*(1 + tand(x(2))).^2 +
tand(x(1)).^2.^(-1/2);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function v2 = calculate_v2(x, beta)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function v2 = calculate_v2(x, beta)
v2 = 1/(a(x).^2 + b(x).^2).*(-w22(x,
beta).*b(x).*c(x) + a(x).*sqrt(-w22(x,
beta).^2 + a(x).^2 + b(x).^2));

function w22 = w22(x, beta)
w22 = (-F2(x, beta) - sqrt(abs(F2(x,
beta).^2 - E(x).*G2(x, beta))))./E(x);
function E = E(x)
E = A(x).^2 + B(x).^2;
function F2 = F2(x, beta)
F2 = -B(x)*D2(x, beta);
function G2 = G2(x, beta)
G2 = D2(x, beta).^2 - A(x).^2.*C(x);
function A = A(x)

```

```

A = b(x).*c(x) - a(x).*b(x).*c(x);
function B = B(x)
B = a(x).*c(x).^2 + b(x).^2;
function C = C(x)
C = a(x).^2 + b(x).^2;
function D2 = D2(x, beta)
D2 = -(a(x).^2 + b(x).^2).*(b(x).^2 +
c(x).^2).*sind(beta - x(3));
function a = a(x)
a = (1 + tand(x(2))).^2 +
tand(x(1)).^2).^(-1/2);
function b = b(x)
b = tand(x(2)).*(1 + tand(x(2))).^2 +
tand(x(1)).^2).^(-1/2);
function c = c(x)
c = tand(x(1)).*(1 + tand(x(2))).^2 +
tand(x(1)).^2).^(-1/2);

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function w1 = calculate_w1(x, beta)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function w1 = calculate_w1(x, beta)
w1 = (-F1(x, beta) - sqrt(abs(F1(x,
beta).^2 - E(x).*G1(x, beta))))./E(x);

```

```

function E = E(x)
E = A(x).^2 + B(x).^2;
function F1 = F1(x, beta)
F1 = -B(x)*D1(x, beta);
function G1 = G1(x, beta)
G1 = D1(x, beta).^2 - A(x).^2.*C(x);
function A = A(x)
A = b(x).*c(x) - a(x).*b(x).*c(x);
function B = B(x)
B = a(x).*c(x).^2 + b(x).^2;
function C = C(x)
C = a(x).^2 + b(x).^2;
function D1 = D1(x, beta)
D1 = (a(x).^2 + b(x).^2).*(b(x).^2 +
c(x).^2).*cosd(beta - x(3));
function a = a(x)

```

```

a = (1 + tand(x(2))).^2 +
tand(x(1)).^2).^(-1/2);
function b = b(x)
b = tand(x(2)).*(1 + tand(x(2))).^2 +
tand(x(1)).^2).^(-1/2);
function c = c(x)
c = tand(x(1)).*(1 + tand(x(2))).^2 +
tand(x(1)).^2).^(-1/2);

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function w2 = calculate_w2(x, beta)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function w2 = calculate_w2(x, beta)
w2 = (-F2(x, beta) - sqrt(abs(F2(x,
beta).^2 - E(x).*G2(x, beta))))./E(x);

```

```

function E = E(x)
E = A(x).^2 + B(x).^2;
function F2 = F2(x, beta)
F2 = -B(x)*D2(x, beta);
function G2 = G2(x, beta)
G2 = D2(x, beta).^2 - A(x).^2.*C(x);
function A = A(x)
A = b(x).*c(x) - a(x).*b(x).*c(x);
function B = B(x)
B = a(x).*c(x).^2 + b(x).^2;
function C = C(x)
C = a(x).^2 + b(x).^2;
function D2 = D2(x, beta)
D2 = -(a(x).^2 + b(x).^2).*(b(x).^2 +
c(x).^2).*sind(beta - x(3));
function a = a(x)
a = (1 + tand(x(2))).^2 +
tand(x(1)).^2).^(-1/2);
function b = b(x)
b = tand(x(2)).*(1 + tand(x(2))).^2 +
tand(x(1)).^2).^(-1/2);
function c = c(x)
c = tand(x(1)).*(1 + tand(x(2))).^2 +
tand(x(1)).^2).^(-1/2);

```

## **APPENDIX D**

### **MATLAB® CODE FOR CHAPTER 6**

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
MORPHOLOGICAL OPERATIONS
FOR EDGE DETECTION
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
                    read image
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
I = imread('original image.jpg');

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
use morphological operation to estimate
                    the background
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
background = imopen(I,strel('disk',6));

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
subtract the background image from the
                    original image
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
I2 = I - background;
I2 = 255-I2;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
threshold the image, remove background
                    noise
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
level = graythresh(I2);
bw = im2bw(I2,level);
bw = bwareaopen(bw, 50);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
LATERAL DETECTION USING DFT
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
                    only code for lateral size estimation
error is shown; code for camera position
                    and orientation can be adjusted
                    accordingly.
                    R: pipe radius
                    r_l: lateral radius

```

```

                    x: horizontal coordinates
                    y: vertical coordinates
                    z: longitudinal coordinates
number: angular sampling resolution
                    D: viewing longitudinal distance
                    space: template radius increment
sizeofFD: number of frequency to retain
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
number = 200;
t = linspace(0,360,number);
R = 10;
r_template = 0.5;
D = 8;
space = 200;
noise = 5;
k = 1;
sizeofFD = 32;
F_Stack = zeros(space,sizeofFD);
G_Stack = zeros(space,sizeofFD);
H_Stack = zeros(space,sizeofFD);
I_Stack = zeros(space,sizeofFD);
r_Stack = zeros(1,space);

for r = linspace(0.5,6,space)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
                    true lateral geometry in earth's
                    coordinates
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
                    x_lateral = r*cosd(t);
                    z_lateral = r*sind(t);
                    y_lateral = sqrt(R^2 -
                    r^2*cosd(t).^2);
                    P_lateral = [x_lateral; y_lateral;
                    z_lateral; ones(1, size(t,2))];

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
                    0% ovality joint geometry in earth's
                    coordinates
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
                    x_joint1 = R*cosd(t);
                    y_joint1 = R*sind(t);
                    z_joint1 = 0*t;
                    P_joint1 = [x_joint1; y_joint1;
                    z_joint1; ones(1, size(t,2))];

```

```

%%%%%%%%%%
10% ovality joint geometry in earth's
coordinates

```

```

%%%%%%%%%%
s = 0.90;
r2 = s*R;
r1 = R/3*( -2*s + 3 + sqrt( -5*s.^2 +
6*s + 3));
x_joint2 = r1*cosd(t);
y_joint2 = r2*sind(t);
z_joint2 = 0*t;
P_joint2 = [x_joint2; y_joint2;
z_joint2; ones(1, size(t,2))];

```

```

%%%%%%%%%%
template geometry in earth's coordinates

```

```

%%%%%%%%%%
x_template = r_template*cosd(t);
z_template = r_template*sind(t);
y_template = sqrt(R^2 -
r^2*cosd(t).^2);
P_template = [x_template;
y_template; z_template; ones(1,
size(t,2))];

```

```

%%%%%%%%%%
set up transformation from earth's
coordinates to camera coordinates

```

```

%%%%%%%%%%
eyex = 0; eyey = 0; eyez = D;
eye = [0, eyey, eyez]; look = [0, 0, 0];
up = [0, 1, 0];
n = eye - look;
n = n./norm(n); up = up./norm(up);
u = cross(up,n); v = cross(n,u);
d(1) = -dot(eye, u); d(2) =
-dot(eye, v); d(3) = -dot(eye, n);
V = [u, d(1); v, d(2); n, d(3);
0, 0, 0, 1];

```

```

%%%%%%%%%%
use true lateral contour as benchmark 1
%%%%%%%%%%
%%%%%%%%%%
lateral geometry in camera coordinates
%%%%%%%%%%

```

```

P1 = V*P_lateral;
x2 = P1(1,:);
y2 = P1(2,:);
z2 = P1(3,:);

```

```

%%%%%%%%%%
assign zenith angle = theta1 & azimuth
angle = theta2

```

```

%%%%%%%%%%
theta1 = atan2(sqrt(x2.^2 +
y2.^2),abs(z2));
theta2 = atan2(y2,x2);

```

```

%%%%%%%%%%
scaling normalization (radius in image
are equal to each other at maximum
zenith)

```

```

%%%%%%%%%%
max = 60/180*pi;
k4 = 1/sin(max);

```

```

%%%%%%%%%%
quarter image pixel resolution 768*768

```

```

%%%%%%%%%%
m_rsln = 768;
v_rsln = 768;

```

```

%%%%%%%%%%
sine law projection

```

```

%%%%%%%%%%
x4 =
m_rsln*k4*sin(theta1).*cos(theta2);
y4 =
v_rsln*k4*sin(theta1).*sin(theta2) +
noise*(rand(1,number) - 0.5);

```

```

True_lateral = x4 + y4*1i;
G1 = extractDFT(True_lateral);
G2 = normalizeDFT(G1);
G = smoothDFT(G2, sizeofFD);

```

```

%%%%%%%%%%
use 0% ovality joint contour as
benchmark 2

```

```

%%%%%%%%%%
P1 = V*P_joint1;

```

```

x2 = P1(1,:);
y2 = P1(2,:);
z2 = P1(3,:);
theta1 = atan2(sqrt(x2.^2 +
y2.^2),abs(z2));
theta2 = atan2(y2,x2);
x4 =
m_rsln*k4*sin(theta1).*cos(theta2);
y4 =
v_rsln*k4*sin(theta1).*sin(theta2) +
noise*(rand(1,number) - 0.5);

Joint1 = x4 + y4*1i;
H1 = extractDFT(Joint1);
H2 = normalizeDFT(H1);
H = smoothDFT(H2, sizeofFD);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
use 10% ovality joint contour as
benchmark 3
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
P1 = V*P_joint2;
x2 = P1(1,:);
y2 = P1(2,:);
z2 = P1(3,:);
theta1 = atan2(sqrt(x2.^2 +
y2.^2),abs(z2));
theta2 = atan2(y2,x2);
x4 =
m_rsln*k4*sin(theta1).*cos(theta2);
y4 =
v_rsln*k4*sin(theta1).*sin(theta2) +
noise*(rand(1,number) - 0.5);

Joint2 = x4 + y4*1i;
I1 = extractDFT(Joint2);
I2 = normalizeDFT(I1);
I = smoothDFT(I2, sizeofFD);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
use fixed lateral size contour as template
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
P1 = V*P_template;
x2 = P1(1,:);
y2 = P1(2,:);
z2 = P1(3,:);

theta1 = atan2(sqrt(x2.^2 +
y2.^2),abs(z2));
theta2 = atan2(y2,x2);
x4 =
m_rsln*k4*sin(theta1).*cos(theta2);
y4 =
v_rsln*k4*sin(theta1).*sin(theta2);

Template = x4 + y4*1i;
F1 = extractDFT(Template);
F2 = normalizeDFT(F1);
F = smoothDFT(F2, sizeofFD);

F_Stack(k,:) = F;
G_Stack(k,:) = G;
H_Stack(k,:) = H;
I_Stack(k,:) = I;
r_Stack(k) = r;
k = k + 1;

end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
frequency chart
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Diff_with_Lateral = zeros(1,space);
Diff_with_Joint1 = zeros(1,space);
Diff_with_Joint2 = zeros(1,space);
if r_template == 0.5;

    template_radius_index = 1;
elseif r_template == 6;
    template_radius_index = space;

end

for k = 1:space

    Diff_with_Lateral(k) =
norm(F_Stack
(template_radius_index,:)) -
norm(G_Stack(k,:));
Diff_with_Joint1(k) = norm(F_Stack
(template_radius_index,:)) -
norm(H_Stack(k,:));

```



```
function G = smoothDFT(F, n)
```

```
G = F;
```

```
len = length(F);
```

```
%%%%%%%%%
```

```
move high frequency to center, take n
```

```
frequencies to retain detailed
```

```
information
```

```
%%%%%%%%%
```

```
if (len > n)
```

```
    T1 = fftshift(F);
```

```
    p = round((len - n) ./ 2);
```

```
    T2 = T1(p + 1 : p + n);
```

```
    G = ifftshift(T2);
```

```
end
```

## **APPENDIX E**

### **MATLAB® CODE FOR CHAPTER 7**

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

### **DRAW QUASI - ELLIPSE**

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

```

r = 10;
w0 = 1;
theta = 0:1:360;
x = r*cosd(theta);
y = r*sind(theta);
x1 = (r*cosd(theta)) +
(w0*cosd(theta).^3);
y1 = (r*sind(theta)) -
(w0*sind(theta).^3);
x2 = (r + w0)*cosd(theta);
y2 = (r - w0)*sind(theta);

figure, plot(x,y,'k',x1,y1,'-k',x2,y2,'
-.k',0,0,'k')
legend('circle','quasi-ellipse','ellipse')

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

### **DRAW CHICUREL'S LOBE EQUATION**

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

```

r = 10;
i = 1;
for l = 1:2:15

    k = 1/(2*r);
    coef = k.^2./(2*sqrt(1 - k.^2).*(1 -
sqrt(1 - k.^2)));
    fhandle = @(u) tan(u) - coef*u;
    u = fzero(fhandle,4);
    h = r - sqrt(r^2 - (l/2).^2);
    x = 0:0.01:l;
    y = 2*h./u^2.*(cos((1 -
2*x/l).*u)./cos(u) - 1);
    plot(x - l/2,y + r -
h,'-k','LineWidth',2)

```

```
end
```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

### **DRAW PROPOSED LOBE EQUATION IN 3-D RGB.**

**PREPARE Z AND Y COORDINATES  
VARYING PHI, OC, AND POWER.**

**SEE APPENDIX A FOR RGB  
COLORING.**

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

*m: angular resolution; n: longitudinal  
resolution; L: pipe length; sink: level of  
largest sinking; range\_angle: range of  
open angle; min\_angle: minimum open  
angle; range\_power: range of power;  
min\_power: minimum power;  
longitudinal positions.*

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

```

r = 10;
r1 = 11;
r2 = 10;
m = 50;
n = 40;
L = 20;
sink = 2;
range_angle = 30;
min_angle = 30;
range_power = 6;
min_power = 2;
x = linspace(-L,L,n);

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

*varying phi: non-uniform sinusoidally,  
non-uniform linearly, and uniform  
segments*

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
phi = abs((cosd(x/L*90))*range_angle) +
min_angle;
phi = ((L - abs(x))/L)*range_angle +
min_angle;
phi = min_angle*ones(1,size(x,2));
OC = 8;
phi_bench = phi;
power = 2;

```

```

theta = zeros(m,n);
for i = 1:n
    theta(:,i) =
        linspace(-phi_bench(i),phi_bench(i),
            m);
end
phi = repmat(phi,m,1);
OA =
r1*r2./sqrt(r1^2*cosd(phi).^2 +
r2^2*sind(phi).^2);
k3 = OA;
k1 = OC - OA;
derivative =
r1*r2*(r1^2 - r2^2) *sind(phi).*
cosd(phi)./(r1^2*cosd(phi).^2 +
r2^2*sind(phi).^2).^3/2);
k2 =
1./(1 +2*phi.*derivative./(k1*180));
rwy =
k1.*(-abs(theta).^power./
(k2.*phi.^power) +1).*
cos(pi*theta./(2*phi)) + k3;
Y1 = rwy.*cosd(theta);
Z1 = rwy.*sind(theta);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
varying OC: non-uniform sinusoidally,
non-uniform linearly, and uniform
segments; determination of Y1 and Z1
follows the code in varying phi with
minor changes.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
OC = r - (cosd(x/L*90))*sink;
OC = r - ((L - abs(x))/L)*sink;
OC = r - sink*ones(1,size(x,2));
phi = 45;
power = 2;
theta = linspace(-phi,phi,m);
[OC1,theta] = meshgrid(OC,theta);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
varying power: non-uniform sinusoidally,
non-uniform linearly, and uniform
segments; determination of Y1 and Z1
follows the code in varying phi with
minor changes.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
power = abs((cosd(x/L*90))*
range_power) + min_power;
power = ((L - abs(x))/L)* range_power +
min_power;
power = min_power*ones(1,size(x,2));
OC = 8;
phi = 45;
theta = linspace(-phi,phi,m);
[power1,theta] = meshgrid(power,theta);

```

## BIBLIOGRAPHY

- [1] American Society for Testing and Materials, “Standard practice for rehabilitation of existing pipelines and conduits by the inversion and curing of a resin-impregnated tube”, ASTM Designation: F1216.
- [2] Ahrary, A., Tian, L., Kamata, S. I., and Ishikawa, M., “Navigation of an autonomous sewer inspection robot based on stereo camera images and laser scanner data”, *International Journal on Artificial Intelligence Tools*, vol. 16, iss. 4, pp. 611–625, 2007.
- [3] Bennett, N. H. and Logan, G. I., “Laser profiling – a necessary tool for pipeline assessment”, *North American Society for Trenchless Technology, No-Dig Conference & Exhibition*, 2005.
- [4] Beuker, T., Palmer, J., and Quack, M., “In-line inspection using combined technologies – magnetic flux leakage and ultrasonic testing and their advantages”, *4<sup>th</sup> Pipeline Technology Conference*, 2009.
- [5] Bookstein, F., “Fitting conic sections to scattered data”, *Computer Graphics and Image Processing*, vol. 9, pp. 56–71, 1979.
- [6] Cai, Q., “Structural behavior of PVC pipe liners subjected to seasonal groundwater loading”, *Ph.D. Dissertation, Louisiana Tech University*, 2007.
- [7] Chicurel, R., “Shrink buckling of thin circular rings”, *Journal of Applied Mechanics, ASME*, pp. 608–610, 1968.
- [8] Dettmer, A. J., “Position and orientation correction for pipe profiling robots”, *Ph.D. Dissertation, Louisiana Tech University*, 2007.
- [9] Falter, B., “New developments in liner design due to ATV-M 127-2”, *North American Society for Trenchless Technology, No-Dig Conference & Exhibition*, 2008.
- [10] Fitzgibbon, A., Pilu, M., and Fisher, R.B., “Direct least square fitting of ellipses”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, iss. 5, 1999.
- [11] Gander, W., Golub, G. H., and Strebel, R., “Least-squares fitting of circles and ellipses”, *BIT Numerical Mathematics*, vol. 34, no. 4, pp. 558–578, 1994.

- [12] Glock, D., "Post-critical behavior of a rigidly encased circular pipe subject to external water pressure and temperature rise", *Der Stahlbau*, vol.46, no.7, pp. 212–217, 1977.
- [13] Golub, G. H. and Van Loan, C. F., "Matrix Computations", 3<sup>rd</sup> edition, Johns Hopkins University Press, pp. 70-71, 1998.
- [14] Guice, L. K. and Li, J. Y., "Buckling models and influencing factors for pipe rehabilitation design", North American Society for Trenchless Technology, No-Dig Conference & Exhibition, 1994.
- [15] Gumble, J., "New approach to design of circular liner pipe to resist external hydrostatic pressure", *Advances in Pipeline Engineering & Construction*, American Society of Civil Engineers, 2001.
- [16] Iyer, S. and Sinha, S. K., "A robust approach for automatic detection and segmentation of cracks in underground pipeline images", *Image and Vision Computing*, vol. 23, iss. 10, pp. 921–933, 2005.
- [17] Knight, M. A., "Comparison of three CIPP liners buckling strength using a flexible bladder – circular samples VS liner sample with a wavy imperfections due to a fiber optic cable", Technical Paper 179, North American Society for Trenchless Technology, 2002.
- [18] Li, J. Y. and Guice, L. K., "Buckling of encased elliptic thin ring", *Journal of Engineering Mechanics*. vol. 121, no. 12, pp. 1325–1329, 1995.
- [19] Omara, A. M., Guice, L. K., and Straughan W. T., "Buckling models of thin circular pipes encased in rigid cavity", *Journal of Engineering Mechanics*. vol. 123, no. 12, pp. 1294–1301, 1997.
- [20] Omara, A. M., Guice, L. K., and Straughan, W. T., "Instability of thin pipes encased in oval rigid cavity", *Journal of Engineering Mechanics*, vol. 126, no. 4, pp. 381–388, 2000.
- [21] Pan, X., Clarke, T. A., and Ellis, T. J., "The detection and tracking of pipe joints in noisy images", *Proceedings of SPIE, Videometrics III*, vol. 2350, pp. 136–147, 1994.
- [22] Press, W. H., Flannery, B. P., Teukolsky, S. A., and Vetterling W. T., "Numerical recipes in FORTRAN: the art of scientific computing", 2<sup>nd</sup> edition, Cambridge University Press, pp. 91-95, 1992.
- [23] Ramanujan, S. "Modular equations and approximations to pi", *Quarterly Journal, Pure and Applied Mathematics*, vol. 45, pp. 350–372, 1914.
- [24] Rogers, C. D. F., "Some observations on flexible pipe response to load", *Transportation Research Record 1191, TRB*, 1998.

- [25] Salagundi, J. R., “Development of a system for automated collection of radial pipe wall images”, Master Thesis, Louisiana Tech University, 2005.
- [26] Schempf, H., “Phase I state-of-the-art review: in-pipe-assessment robot platforms”, Department of Energy, National Energy Technology Laboratory, Contract DE-FC26-04NT42264, 2004.
- [27] Straughan, W. T., Guice, L. K., and Mal-Duraipandian, C., “Long-term structural behavior of pipeline rehabilitation systems”, *Journal of Infrastructure Systems*. vol. 1, no. 4, pp. 214–220, 1995.
- [28] Timoshenko, S.P., and Gere, J.M., “Theory of elastic stability”, 2<sup>nd</sup> Edition, McGraw-Hill, New York, pp. 290–292, 1961.
- [29] Zhang, D. and Lu, G., “A comparative study on shape retrieval using Fourier descriptors with different shape signatures”, *Journal of Visual Communication and Image Representation*, no. 14, pp. 41–60, 2003.
- [30] Zhao, J. Q. and Doherty, I.J., “Behavior and performance of liner pipe in trenchless and trenched portions of sliplining rehabilitation”, *International NoDig Conference*, pp. 1–14, 2003.
- [31] [http://aettopgun.com/laser\\_pipe.html](http://aettopgun.com/laser_pipe.html)
- [32] <http://en.wikipedia.org>
- [33] <http://mrobot.com>
- [34] <http://physicsarchives.com/index.php/courses/900-technic>
- [35] <http://roseninspection.net>
- [36] <http://webendtubing.com/BendingTerms.html>