

PRECISION ROBOTIC CONTROL OF AGRICULTURAL VEHICLES ON REALISTIC FARM TRAJECTORIES

A DISSERTATION
SUBMITTED TO THE DEPARTMENT OF AERONAUTICS AND
ASTRONAUTICS
AND THE COMMITTEE ON GRADUATE STUDIES
OF STANFORD UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

Thomas Bell

June 1999

© Copyright 1999 by Thomas Bell
All Rights Reserved

I certify that I have read this dissertation and that in my opinion it is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

Professor Bradford W. Parkinson
(Principal Advisor)

I certify that I have read this dissertation and that in my opinion it is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

Professor Stephen M. Rock

I certify that I have read this dissertation and that in my opinion it is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

Professor Claire J. Tomlin

Approved for the University Committee on Graduate Studies:

Dean of Graduate Studies

Abstract

High-precision “autofarming”, or precise agricultural vehicle guidance, is rapidly becoming a reality thanks to increasing computing power and carrier-phase differential GPS (“CPDGPS”) position and attitude sensors. Realistic farm trajectories will include not only rows but also arcs created by smoothly joining rows or by path-planning algorithms, spirals for farming center-pivot irrigated fields, and curved trajectories dictated by nonlinear field boundaries. In addition, fields are often sloped, and accurate control may be required either on linear trajectories or on curved contours.

A three-dimensional vehicle model which adapts to changing vehicle and ground conditions was created, and a low-order model for controller synthesis was extracted based on nominal conditions. The model was extended to include a towed implement. Experimentation showed that an extended Kalman filter could identify the vehicle’s state in real-time. An approximation was derived for the additional positional uncertainty introduced by the noisy “lever-arm correction” necessary to translate the GPS position measurement at the roof antenna to the vehicle’s control point on the ground; this approximation was then used to support the assertion that attitude measurement accuracy was as important to control point position measurement as the original position measurement accuracy at the GPS antenna.

The low-order vehicle control model was transformed to polar coordinates for control on arcs and spirals. Experimental data showed that the tractor’s control point tracked an arc to within a -0.3 cm mean and a 3.4 cm standard deviation and a spiral to within a -0.2 cm mean and a 5.3 cm standard deviation.

Cubic splines were used to describe curve trajectories, and a general expression for the time-rate-of-change of curve-related parameters was derived. Four vehicle

control algorithms were derived for curve tracking: linear local-error control based on linearizing the vehicle about the curve’s radius of curvature, linear finite-preview control using discrete linear quadratic tracking, nonlinear local error control based on feedback linearization, and nonlinear finite-preview control using nonlinear optimization techniques. The first three algorithms experimentally demonstrated mean tracking errors between zero and four centimeters and standard deviations of roughly four to ten centimeters. The fourth algorithm was computationally too expensive to implement with current technology. In experiment, the feedback linearization algorithm outperformed the other two control algorithms and also used the most control effort.

For control on sloped terrain, a variation on bias estimation (termed slope-adjusted bias estimation) was created, based on the terrain slope information calculated from vehicle attitude measurements. Slope-adjusted bias estimation demonstrated a 25% improvement in the standard deviation of the tractor’s row-tracking error over “normal” bias estimation on terrain sloped at grades up to 28%. The CPDGPS attitude information was also used to develop a contour-tracking controller that tracked a contour to within a mean *height* error of 0.5 cm and a standard deviation of 4.3 cm without any prior knowledge of the terrain.

These real-time vehicle control results, applicable to any front-wheel-steered vehicle, demonstrate that accurate real-time control is possible over a variety of trajectories needed in a commercial autofarming system. This research is a significant step towards completely automating tractor control because farmers can now build global trajectories composed of the different types of trajectory “building blocks” developed here. Experimental results demonstrate that farmers can expect precision tracking down to the limit of the GPS position and attitude sensors.

In memory of my brother
Christopher Bell
March 30, 1970–October 7, 1996

*How good life was
until he was just a memory,
a face in a photograph.*

Acknowledgments

To my advisor, Prof. Brad Parkinson. His dynamic personality, enthusiasm, optimism, and experience made the most challenging problems seem conquerable. To Dr. Mike O'Connor. For inviting me to his wedding (where I met my future wife), and for being the most patient person who has ever had to explain something to me. Thanks also for graduating before me—your old desk chair has been really comfortable. To Bob Mayfield and Wayne Smith at John Deere. For foreseeing the value of high-precision GPS in agriculture. For funding this project...and me. To fellow tractor students Andy Rekow and Dave Bevly. For great memories of the good times and the frustrating ones too. To Andy, for his endless enthusiasm and eagerness. To Dave, for his practical experience and sense of what's worth pursuing and what's not. And Dave, I've heard enough about Texas for a while, thanks. To Jock Christie and Andy Barrows, those voices on the other side of the cubicle wall. Thanks for answering one question after another. To Matt Rabinowitz. Thanks for letting me borrow your GPS receiver to conduct the final round of tests. To Jeff Wade, System Administrator for Gravity Probe B and one of the hardest working people at GP-B. For providing a lot of the hardware, software, and support needed to make these results happen. To Vivian Drew, Prof. Parkinson's Administrative Assistant. For coordinating numerous visits and demonstrations. For doing all the "little" things which weren't so little. To Jeri Stallard, Mike Killian, Larry Green, and the rest of the usual suspects at the H.E.P.L. administrative offices. To the folks at Stanford Land Management and the Stanford Equestrian Center. As precious as land is in Silicon Valley, they generously allowed us valuable testing time on their fields at no cost. To Peter Benner at the University of Bremen's Zentrum für Technomathematik. For his assistance with the

Fortran F77 SLICOT subroutine library, without which we would not have been able to solve for control gains in real-time. The Internet has made the world smaller, and Peter was a faceless e-mail address on the other side of the globe who freely offered his help. To the IntegriNautics Corporation. Their high-precision GPS systems used in this research stand head and shoulders above the competitions'—a tribute to their enormous talent.

To Natalie, for her love. And for proofreading this entire thesis, though the responsibility for all errors are, unfortunately, still mine. For tolerating me when I was stuck yet again on another software bug.

To my parents, Ronnie and Herta Bell. For everything else.

Contents

| | |
|-----------------------------------|--------------|
| Abstract | v |
| Acknowledgments | ix |
| Nomenclature | xxiii |
| Explanation of Notation | xxiii |
| Variables | xxiv |
| Roman Variables | xxiv |
| Greek Variables | xxvi |
| Frames | xxvii |
| Superscripts | xxvii |
| Subscripts | xxvii |
| Acronyms | xxviii |
| 1 The Autofarming Concept | 1 |
| 1.1 Autofarming | 1 |
| 1.2 Previous Research | 4 |
| 1.3 Objectives | 7 |
| 1.4 Organization | 7 |
| 1.5 Contributions | 9 |
| 2 Problem Statement | 13 |
| 2.1 Approach | 13 |
| 2.2 System Overview | 14 |

| | | |
|----------|--|-----------|
| 2.3 | Verification | 17 |
| 3 | Vehicle Modeling | 19 |
| 3.1 | Factors Influencing Tractor Dynamics | 20 |
| 3.2 | Previous Model | 23 |
| 3.3 | New, More Accurate Model | 26 |
| 3.3.1 | Effective Steer Angle, Steering Proportionality Constant, and Steering Bias Angle | 26 |
| 3.3.2 | Attitude | 27 |
| 3.3.3 | Position, Velocity, and Crab Angle | 28 |
| 3.3.4 | Equations of Motion | 30 |
| 3.3.5 | Linearized Model | 31 |
| 3.3.6 | Model Analysis | 32 |
| 3.4 | Disturbance Model | 36 |
| 4 | Towed Implement Modeling | 41 |
| 4.1 | Dynamic Model | 42 |
| 4.1.1 | Overview | 42 |
| 4.1.2 | Equations of Motion | 43 |
| 4.1.3 | Linearized Model | 48 |
| 4.1.4 | Model Analysis | 50 |
| 4.2 | Disturbance Model | 53 |
| 5 | Nonlinear Real-Time Estimation | 55 |
| 5.1 | Measurements | 57 |
| 5.1.1 | Position Measurement | 58 |
| 5.1.2 | GPS Attitude Measurement | 60 |
| 5.1.3 | Steer Angle Measurement | 60 |
| 5.2 | The Kalman Filter | 62 |
| 5.3 | The Extended Kalman Filter | 63 |
| 5.3.1 | First-Order Filter | 63 |
| 5.3.2 | Second-Order Filter | 64 |

| | | |
|----------|---|-----------|
| 5.4 | Simulation Results | 65 |
| 5.5 | Experimental Results | 66 |
| 6 | The Lever Arm Correction | 71 |
| 6.1 | Approximating the Additional Uncertainty | 72 |
| 6.2 | Verifying the Additional Uncertainty Approximation | 75 |
| 6.3 | Effects of Attitude Noise and Lever Arm Length | 78 |
| 7 | Vehicle Control on Arcs | 81 |
| 7.1 | Specifying Arc Trajectories | 81 |
| 7.2 | Modeling the Vehicle in Polar Coordinates | 82 |
| 7.2.1 | Vehicle Yaw Error | 82 |
| 7.2.2 | Vehicle Steer Angle | 84 |
| 7.2.3 | Vehicle Tracking Error | 85 |
| 7.2.4 | Equations of Motion | 85 |
| 7.3 | Tracking Control Algorithm | 86 |
| 7.4 | Experimental Results | 86 |
| 8 | Vehicle Control on Spirals | 89 |
| 8.1 | Specifying Spiral Trajectories | 89 |
| 8.1.1 | Radius of Curvature | 90 |
| 8.2 | Modeling the Vehicle in Polar Coordinates | 91 |
| 8.2.1 | Vehicle Yaw Error | 92 |
| 8.2.2 | Vehicle Steer Angle | 93 |
| 8.2.3 | Vehicle Tracking Error | 94 |
| 8.2.4 | Equations of Motion | 94 |
| 8.2.5 | Analysis | 94 |
| 8.3 | Experimental Results | 94 |
| 9 | Vehicle Control on Curves: Specifying Curve Trajectories | 99 |
| 9.1 | Generating Curve Trajectories from Position Data | 100 |
| 9.2 | Cubic Splines | 101 |

| | | |
|-----------|---|------------|
| 9.3 | Locating the Vehicle on the Curve | 102 |
| 9.4 | Differentiating Curve Parameters | 105 |
| 9.5 | Predicting Future Vehicle Locations on a Curve | 106 |
| 9.6 | Predicting Future Reference State Information | 107 |
| 10 | Vehicle Control on Curves: Control Algorithms | 109 |
| 10.1 | Linear Local-Error Control | 110 |
| 10.1.1 | Vehicle Yaw Error | 110 |
| 10.1.2 | Vehicle Steer Error | 111 |
| 10.1.3 | Vehicle Tracking Error | 112 |
| 10.1.4 | Equations of Motion | 112 |
| 10.2 | Linear Finite-Horizon Control | 113 |
| 10.3 | Feedback Linearization | 115 |
| 10.3.1 | Feedforward Control Signal | 116 |
| 10.4 | Control as a Nonlinear Optimization Problem | 118 |
| 10.4.1 | Calculating the Cost Function Jacobian | 119 |
| 10.4.2 | Calculating the Cost Function Hessian | 123 |
| 11 | Vehicle Control on Curves: Simulation and Experimental Results | 125 |
| 11.1 | Simulation Results | 125 |
| 11.2 | Experimental Results | 127 |
| 11.2.1 | Linear Local-Error Control | 128 |
| 11.2.2 | DLQ Tracker Control | 128 |
| 11.2.3 | Feedback Linearization Control | 129 |
| 11.2.4 | Summary of Results | 129 |
| 12 | Vehicle Control on Sloped Terrain | 137 |
| 12.1 | Standard Bias Estimation | 138 |
| 12.2 | Slope-Adjusted Bias Estimation | 139 |
| 12.3 | Experimental Results | 142 |

| | |
|---|------------|
| 13 Vehicle Control on Contours | 153 |
| 13.1 Control Algorithms | 153 |
| 13.2 Specifying Contour Trajectories | 154 |
| 13.2.1 Measuring Yaw Error | 154 |
| 13.2.2 Measuring Steering Error | 155 |
| 13.2.3 Measuring Tracking Error | 155 |
| 13.3 Measurement Variances | 156 |
| 13.3.1 Tracking Error Measurement Variance | 156 |
| 13.3.2 Yaw Error Measurement Variance | 157 |
| 13.4 Experimental Results | 158 |
| 14 Conclusions and Future Research | 161 |
| 14.1 Summary of Results | 162 |
| 14.1.1 Vehicle Modeling | 162 |
| 14.1.2 Analysis of the Lever Arm Correction | 163 |
| 14.1.3 Control on Arcs and Spirals | 163 |
| 14.1.4 Control on Curves | 164 |
| 14.1.5 Control on Sloped Terrain and Contours | 164 |
| 14.2 Future Research | 165 |
| 14.3 Implications | 168 |
| A Test Equipment | 169 |
| A.1 Deere and Co. Model 7800 | 169 |
| A.2 Deere and Co. Model 8400 | 170 |
| A.3 Sensor and Actuator Hardware | 171 |

List of Tables

| | | |
|------|--|-----|
| 3.1 | Estimated disturbance variances for the tractor model. | 39 |
| 4.1 | Estimated disturbance variances for the implement model. | 54 |
| 5.1 | Static horizontal precision of GPS position measurements. | 59 |
| 5.2 | Observed high-frequency variances in GPS attitude measurement. . . | 60 |
| 6.1 | Comparison of singular values for predicted and experimental covari- ance matrices. | 76 |
| 6.2 | Attitude measurement noise level yielding 2.5-cm positional uncer- tainty for various lever arm lengths | 80 |
| 11.1 | Summary of tracking error history for local-error, DLQ tracker, feed- back linearization, and optimized control algorithms. | 126 |
| 11.2 | Experimental curve tracking performance. | 132 |
| 11.3 | Experimental curve tracking control effort. | 132 |
| 12.1 | Experimental row tracking performance on sloped terrain. | 151 |
| 14.1 | Summary of experimental tracking performance for all trajectory types. | 165 |

List of Figures

| | | |
|-----|---|------|
| 1 | Diagram of a variable and its associated qualifiers. | xxiv |
| 2.1 | Photograph of the Deere and Co. Model 7800 farm tractor used in experiments. | 15 |
| 2.2 | Chisel plow used for testing on the Model 7800. | 15 |
| 2.3 | Block diagram of experimental system. | 16 |
| 3.1 | Diagram showing definitions of variables used in the vehicle model. . | 24 |
| 3.2 | Pole locations for 5 Hz discrete closed-loop vehicle model. | 33 |
| 3.3 | Norm of the dominant eigenvalue λ as a function of the tracking error penalty term. | 33 |
| 3.4 | Pole locations for 5 Hz discrete closed-loop vehicle model for a hitch control point. | 34 |
| 3.5 | Norm of the dominant eigenvalue λ as a function of the tracking error penalty term for a hitch control point. | 35 |
| 3.6 | Pole locations for 5 Hz discrete closed-loop vehicle model with reduced front-wheel traction. | 35 |
| 3.7 | Norm of the dominant eigenvalue λ as a function of the steering constant. . | 36 |
| 3.8 | Pole locations for 20 Hz discrete closed-loop vehicle model. | 37 |
| 4.1 | Diagram showing definitions of variables used in the towed implement model. | 43 |
| 4.2 | Continuous-time root locus of the linearized vehicle/implement model. . | 51 |
| 4.3 | Tracking error response for both the vehicle and vehicle/implement for the same initial conditions. | 52 |

| | | |
|-----|---|----|
| 4.4 | Steering response for both the vehicle and vehicle/implement for the same initial conditions. | 52 |
| 5.1 | Photo of rubber mounts between tractor cab and frame. | 59 |
| 5.2 | Yaw rate as a function of the steering potentiometer reading. | 61 |
| 5.3 | Steer angle measurement variance as a function of steering angle. . . | 62 |
| 5.4 | Comparison of first- and second-order filtering algorithms for estimating the steering constant. | 66 |
| 5.5 | Comparison of first- and second-order filtering algorithms for estimating the steering bias. | 67 |
| 5.6 | Filter estimates of K_δ for eleven experimental runs. | 68 |
| 5.7 | Extended Kalman filter's tracking error estimate for control along a straight row. | 69 |
| 6.1 | Projection of 500 noise-corrupted lever arm corrections generated in a Monte Carlo simulation onto the North-East plane. | 76 |
| 6.2 | Projection of 500 noise-corrupted lever arm corrections generated in a Monte Carlo simulation onto the East-Down plane. | 77 |
| 6.3 | Projection of 500 noise-corrupted lever arm corrections generated in a Monte Carlo simulation onto the North-Down plane. | 77 |
| 6.4 | The additional uncertainty introduced by the noise-corrupted lever arm correction: a comparison between approximation and Monte Carlo simulation. | 78 |
| 6.5 | Comparison between the maximum uncertainty introduced by the noise-corrupted lever arm correction and the GPS position measurement uncertainty. | 79 |
| 7.1 | Diagram of vehicle moving along arc. | 83 |
| 7.2 | Overhead view of experimental arc trajectory. Coordinates are relative to the reference station antenna. | 87 |
| 7.3 | Tracking error along experimental arc trajectory. | 88 |
| 8.1 | Photo of center-pivot irrigation systems in the Libyan Desert. | 90 |
| 8.2 | Overhead view of simulated spiral trajectory. | 95 |
| 8.3 | Reference steer angle as a function of spiral radius. | 95 |

| | | |
|-------|--|-----|
| 8.4 | Overhead view of experimental spiral trajectory. Coordinates are relative to the reference station antenna. | 96 |
| 8.5 | Tracking error along experimental spiral trajectory. | 96 |
| 9.1 | Overhead view of spline interpolation of discrete noisy position data. | 102 |
| 9.2 | Overhead view of a potential problem encountered when determining the tractor's location on the curve. | 104 |
| 9.3 | Diagram showing the current vehicle location on the curve and three predicted future locations. | 107 |
| 11.1 | Curved trajectory used for comparing control algorithms. | 126 |
| 11.2 | Curve tracking error history for local-error, DLQ tracker, feedback linearization, and optimized control algorithms. | 127 |
| 11.3 | Control signal history for optimized control algorithm. | 128 |
| 11.4 | Overhead view of actual position data for a not-so-typical vehicle trajectory. | 129 |
| 11.5 | Overhead view of actual position data for four curves. | 130 |
| 11.6 | Experimental curve tracking error using linear local-error controller. | 130 |
| 11.7 | Experimental curve tracking error using DLQ tracker controller. | 131 |
| 11.8 | Experimental curve tracking error using feedback linearization controller. | 131 |
| 11.9 | Experimental curve tracking control effort using linear local-error controller. | 133 |
| 11.10 | Experimental curve tracking control effort using DLQ tracker controller. | 133 |
| 11.11 | Experimental curve tracking control effort using feedback linearization controller. | 134 |
| 12.1 | Simplified force balance on tractor and tire. | 140 |
| 12.2 | Time history of typical terrain grade along path. | 142 |
| 12.3 | Overhead view of typical trajectory used for row control tests on sloped terrain. | 143 |
| 12.4 | Photograph of the tractor climbing over a sloped ridge. | 144 |
| 12.5 | Experimental results for row guidance on near-flat terrain with standard bias estimation. | 145 |

| | | |
|-------|--|-----|
| 12.6 | Terrain grade for row guidance on near-flat terrain with standard bias estimation. | 145 |
| 12.7 | Experimental results for row guidance under human control on sloped terrain. | 146 |
| 12.8 | Terrain grade for row guidance under human control on sloped terrain. | 146 |
| 12.9 | Experimental results for row guidance on sloped terrain without bias estimation. | 147 |
| 12.10 | Terrain grade for row guidance on sloped terrain without bias estimation. | 148 |
| 12.11 | Experimental results for row guidance on sloped terrain with standard bias estimation. | 148 |
| 12.12 | Terrain grade for row guidance on sloped terrain using standard bias estimation. | 149 |
| 12.13 | Experimental results for row guidance on sloped terrain with slope-adjusted bias estimation. | 149 |
| 12.14 | Terrain grade for row guidance on sloped terrain using slope-adjusted bias estimation. | 150 |
| 13.1 | Diagram showing the effect of terrain slope angle on tracking uncertainty. | 158 |
| 13.2 | Experimental results for automatic control along a contour. | 159 |
| 13.3 | Yaw error measurement variance for automatic control along a contour. | 159 |
| 13.4 | Tracking error measurement variance for automatic control along a contour. | 160 |
| A.1 | Photograph of the Deere and Co. Model 8400 farm tractor used in experiments. | 170 |
| A.2 | Photograph of GPS and computer equipment mounted inside the 7800 cab. | 171 |
| A.3 | Photograph of the electrohydraulic valve and steering angle potentiometer on the Model 7800 tractor. | 172 |

Nomenclature

Explanation of Notation

Fig. (1) shows a variable and its associated qualifiers. The qualifiers' roles are explained below. Subsequent sections explain the possible values variables and their qualifiers may have.

Variable The variable itself. May be a scalar or a vector.

Subscript Specifies what object or location the variable corresponds to. For example, the subscript c on the velocity variable V denotes the velocity of the control point.

Superscript Specifies whether the variable corresponds to the vehicle or to the towed implement.

Underset The reference frame that the variable is coordinatized in.

Overset Denotes differentiation with respect to time taken in the (overset) frame. This notation is used only when there are multiple frames (*i.e.*, in three dimensions). Repeated overset frames denotes multiple time derivatives. For example, the overset iii denotes the third time derivative taken in the inertial frame.

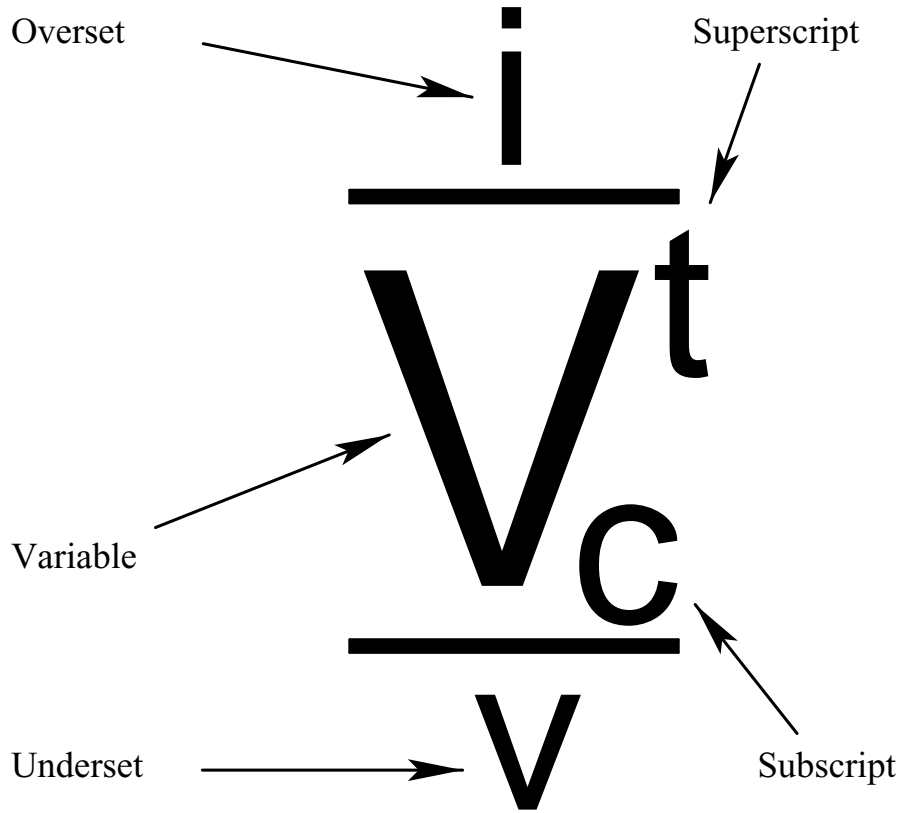


Figure 1: Diagram of a variable and its associated qualifiers.

Variables

Roman Variables

| | |
|-----------|---|
| b_i | Byte value i in the steering look-up table. |
| \hat{b} | Temporary vector used in the modified Gaussian second-order filter. |
| d | Tracking error of the control point from the reference trajectory. |
| e | East. |
| $f(x)$ | The vector set of equations describing the vehicle motion. |
| h | Height error. |

| | |
|-------------|---|
| $h(x)$ | The measurement or observation vector as a function of the vehicle's state x . |
| k_i | Temporary vector number i used in Runge-Kutta integration. |
| l | Scalar distance. Defined positive forward. |
| n | North. |
| p | Unit vector along the desired trajectory. |
| q | Unit vector out of the roof of the vehicle (<i>i.e.</i> , along the vertical axis in the vehicle's coordinate frame.). |
| r | The radial distance of vehicle's control point from the curve center or a general position vector. |
| s | Independent parameter used to specify cubic splines. Also used to denote unit vector pointing in direction of steepest terrain descent. |
| u | Control signal (units of angular rate). |
| w | Implement width. |
| x | State vector. |
| y | Measurement vector. |
| A_k | Coefficient in cubic spline for interval k . |
| B_k | Coefficient in cubic spline for interval k . |
| \hat{B}_m | Temporary matrix used in the modified Gaussian second-order filter. |
| C_k | Coefficient in cubic spline for interval k . |
| D_k | Coefficient in cubic spline for interval k . |
| E | Expected value operator. |
| F | Force. |
| G | The Jacobian of the disturbance vector in the equations of motion. |
| H | The Jacobian of the measurement vector (<i>i.e.</i> , the observation matrix). Also used to denote the Hamiltonian matrix. |
| I | Identity matrix. |
| J | Cost function. |
| K | Control gain vector or proportionality constant. |

| | |
|-------|--|
| L | Curve length. |
| L_c | Estimator gain matrix. |
| N | Normal force or the number of look-ahead intervals for finite-horizon control. |
| P | Estimator inertia matrix. |
| R | Radius of curvature. |
| R_v | Measurement noise covariance matrix. |
| Q | Process noise covariance matrix. |
| T_s | Sample time. |
| T | Transformation matrix. |
| V | Velocity. |

Greek Variables

| | |
|-----------|---|
| α | Angle swept out by the vehicle when traveling on arcs, spirals, or curves. |
| β | Ratio used in specifying spiral trajectories. |
| δ | Front-wheel angle relative to the vehicle's centerline. |
| ζ | Terrain slope angle along the trajectory. |
| η | Yaw angle incurred by the spiral's changing radius of curvature. |
| θ | Pitch angle. |
| λ | Eigenvalue. |
| μ | Crab angle of vehicle required to maintain tracking on sloped terrain. |
| ν_i | Disturbance with zero mean and Gaussian distribution affecting state i . |
| ξ | Vehicle yaw angle relative to a trajectory incurred by non-collocated pivot and control points. |
| ρ | Distance from the spiral's center to a point along the spiral. |
| τ | Time constant. |
| ϕ | Roll angle. |

| | |
|----------|---|
| ψ | Yaw angle. |
| ω | Attitude rate vector. |
| Γ | Discrete control input matrix in the linearized vehicle model. |
| Θ | Attitude vector. |
| Σ | Covariance matrix. |
| Φ | Discrete state transition matrix in the linearized vehicle model. |

Frames

| | |
|-------|---|
| i | Inertial frame. |
| r_2 | Coordinate frame with x -axis aligned with the r_2 vector and y -axis aligned with the implement's y -axis. |
| t | Towed implement frame. |
| v | Vehicle frame. |

Superscripts

| | |
|-----|------------------|
| t | Towed implement. |
| v | Vehicle. |

Subscripts

| | |
|-------|---|
| a | Point on the ground beneath the a xle center. |
| b | B ias. |
| c | C ontrol point, c losest point, or c ommanded value. |
| d | D iscrete. |
| e | E rror. |
| m | M easured value. |
| max | M aximum value. |

| | |
|----------|--|
| p | Tow p in. |
| r | R epreference value. |
| s | Variable is a function of the independent curve parameter s . |
| x | X direction (north). |
| y | Y direction (east). |
| z | Z direction (down). |
| δ | Steer angle. |
| ρ | Lever arm. |
| ϕ | Roll. |
| θ | Pitch. |
| ψ | Yaw. |
| ζ | Steepest descent. |
| Θ | Attitude. |
| ∞ | Steady-state value. |

Acronyms

| | |
|--------|--|
| CPDGPS | Carrier-Phase Differential GPS. A high-precision variant of differential GPS that uses phase measurements of the GPS carrier signal for position and attitude calculation. |
| DLQ | Discrete Linear Quadratic. |
| GPS | The Global Positioning System. A constellation of satellites operated by the U.S. Department of Defense used for determining a user's location. |
| LQR | Linear Quadratic Regulator. A type of [linear] regulator derived from the optimal solution of a quadratic cost function for a linear system. |
| NASA | The National Aeronautics and Space Administration. |
| NED | North-East-Down coordinate frame. |
| RTK | Real-Time Kinematic. A form of differential GPS positioning. |

CHAPTER 1

The Autofarming Concept

1.1 Autofarming

Automatic guidance of wheeled vehicles has a wide variety of applications including minefield clearing, hazardous chemical clean-up, transportation of nuclear material, agricultural tractor guidance, automated highways, and delivery vehicles inside factories or hospitals. Autonomous vehicles require high-precision control, continuous operation, increased efficiency, and the removal of a human operator from an unsafe environment. Although autonomous vehicles have long been the subject of research, only recently has sensor and computer technology made autonomous vehicles practical. The advent of Carrier-Phase Differential Global Positioning System (CPDGPS) [PS96] sensors for both position and attitude (roll, pitch, and yaw) sensing has offered engineers the high precision necessary for accurate vehicle control at a fraction of the cost of comparable inertial sensors. Relatively inexpensive computers are now available that are capable of running control and estimation algorithms at rates practical for real-time control, algorithms that would have been computationally too demanding only a few years ago.

With all the tools necessary for economical real-time land-vehicle control, specific commercial applications are stimulating research into effective vehicle control systems. Agriculture has emerged as one of the first potential applications of real-time vehicle control. CPDGPS has several constraints: a clear view of the sky¹, proximity

¹Unless one uses pseudo-satellite transmitters (“pseudolites”) to replace or augment the GPS satellite signals.

to a fixed reference station, and radio contact with the reference station's radio transmitter. Agricultural tractors almost always meet these constraints. Additionally, the high precision possible with an autonomous tractor guidance system has numerous benefits to farmers:

- Certain types of farming applications such as row-crop applications could benefit from high-precision control, control which is available in all visibility conditions. As an example, inaccurate tractor steering is estimated to cause roughly 13% of total losses during sugar beet harvesting [Bot82]. Another example of an application that could benefit from high-precision control is drip-tape irrigation. In drip-tape irrigation, perforated hose or "tape" is buried beneath the crop's roots. Water pumped through the tapes irrigates the crop roots directly, avoiding irrigation water loss through evaporation. In certain arid regions of the world, such as the U.S. southwest, where water availability is a deciding factor in whether or not to farm available land, farmers are very interested in applying drip-tape irrigation. Unfortunately, once buried, the tapes are very susceptible to being cut by farm implements. Once cut, a tape section must be replaced. Unfortunately, a farmer may not realize the tape has been cut until the section of crop "downstream" from the cut has died. Despite the costs resulting from crop losses and labor for tape repair,² tape irrigation techniques are still economically viable for some farmers. An autonomous tractor guidance system could significantly reduce damage to a tape irrigation system, not only because the tractor could be controlled precisely, but because the high repeatability of the CPDGPS sensors would allow the tractor to drive the same trajectories year after year.
- In operations such as spraying, drivers deliberately overlap previous rows, especially in low-visibility conditions such as during the night, to avoid missing areas. Eliminating overlap decreases work time and material requirements, and reduces chemical run-off.
- Hardware, such as marker arms used to assist the driver in offsetting rows evenly,

²One farmer running a large farm expressed a figure on the order of \$100,000 per year.

would not be necessary. These marker arms, which may extend laterally 10 m or more, are often damaged on fences or telephone poles during U-turns.

- Fine-scale topographic field maps could be generated from position data collected while the tractor operates autonomously. High accuracy topographic data is vital even on flat fields because of the cumulative effects of poor drainage. Currently, code-phase differential GPS systems, accurate only to one to two meters, are usually used to collect such data in tractors. Centimeter-level CPDGPS surveying systems are also used, but the amount of data collected is two to three orders of magnitude less than what an autonomous tractor could collect. The high-resolution maps resulting from real-time CPDGPS data could help farmers manage irrigation better.
- Tasks requiring high-level steering accuracy could be accomplished faster. Farmers have noted that vehicle speed and hence productivity is often limited by the user's ability to control the tractor, not by mechanical factors such as engine power. The limitation of the human driver's ability to process information is one of the major hindrances to increasing work rates [Bot82].

With these practical benefits in mind and the sensor and computer hardware available, research on applying CPDGPS to farming systems led to the development of a high-accuracy control algorithm that steered a large farm tractor along straight rows on level terrain [O'C97]. In April 1996, Dr. M. O'Connor demonstrated the first automatically steered farm tractor that used only CPDGPS for both position and attitude sensing. By not including lasers, cameras/pattern recognition, buried wires, or inertial components as additional sensors, Dr. O'Connor developed a system which is flexible, accurate, and inexpensive enough to be practical.

This research begins where Dr. O'Connor's research finished. Realistic farm trajectories that an automatic tractor would be expected to follow are frequently nonlinear, fields are often sloped, and changing ground conditions and forward speeds mean tractor models are time-varying and nonlinear. This research attempts to solve these and other issues by developing and experimentally demonstrating algorithms that may one day become essential components in a practical system.

The term *autofarming* was created to embody the new possibilities offered by CPDGPS-based autonomous tractor guidance. The term encompasses not only automatic tractor guidance, but the entire range of possible developments that could arise from such a system. For example, human factors such as the driver's ability to see the ground, driver comfort, and operator safety are major constraints that tractor engineers have to incorporate when designing tractors. Without a human operator on board, the engine could be redesigned so that the cylinders are horizontally opposed to improve cooling and lower the center of gravity. Air conditioning, suspended seats, or even an enclosed cab would no longer be necessary. Larger radiators could offer improved engine cooling while reducing the engine power siphoned off to spin the cooling fan. Finally, tractors could be run at speeds constrained not by the operator's comfort but by the steering controller's accuracy.

1.2 Previous Research

Previous research on autonomous farm vehicles centered on sonar or vision-based guidance [T⁺85, RS87]. One or more cameras were used along with pattern-recognition techniques to "recognize" the edge of the crop row and adjust the steering accordingly. With the advent of GPS, research focused on using GPS to augment other sensors. Some research efforts, including this one, used GPS exclusively. There are at least eight other efforts aimed at automating farm vehicles:

1. *Dr. O'Connor at Stanford University.* As a graduate student at Stanford University, Dr. M. O'Connor demonstrated that tractors could track rows to high precision using only GPS to measure the tractor's position and attitude. He developed a low-order model of the tractor's dynamics that performed well at low speeds and on flat terrain. He showed how ground disturbances could destabilize the compensator if the tractor's roll motion was not accounted for. He used a video camera mounted beneath the tractor and a rope placed along a row to verify the accuracy of the GPS tracking error measurements. Dr. O'Connor's work was constrained to tracking straight rows on flat terrain. His research

was a significant step towards a viable commercial autofarming system not only because of the row-tracking accuracies he demonstrated, but because the position and attitude sensors did not depend on field-based cues such as markers or buried wires [O'C97]. This lack of dependence on field-specific hardware meant that his system could be used with no advance preparation on any field.

2. *New-Holland/Carnegie-Mellon/NASA*. A cooperative effort between the National Robotics Engineering Consortium at Carnegie Mellon University, tractor manufacturer New Holland, and NASA has led to the development of an autonomous windrower. The Model 2550 Speed Rower uses inertial sensors and “dead-reckoning”³ for measuring attitude and differential GPS for sensing position at 5 Hz. Cameras and pattern recognition software “recognize” the edge of the crop row at 2 Hz. In the Fall of 1997, the windrower cut 40 hectares (99 acres) of alfalfa without a human operator. The transmission can vary the speed of each wheel independently; therefore, researchers were able to control the steering as well as the forward velocity. The system incorporates path planning algorithms for covering the work area as well as avoiding obstacles. The researchers hope that the primary benefit to farmers will come from increased productivity through higher operating speeds [PF97, Rid98, Vyn98].
3. *Trimble Navigation*. Trimble Navigation mounted a code-phase differential GPS system on a large spray rig. Tracking error was used to direct the human driver through a light bar mounted on the windshield of the sprayer. The driver attempted to keep the left/right error light on the light bar centered for accurate tracking. No measure of attitude was used [WB98].
4. *Holland*. A Dutch research effort recently demonstrated low-speed closed-loop control along straight lines on smooth ground. Unlike most other research efforts, the controller did not steer the tractor. Rather, the hitched implement was hydraulically actuated from side to side to track the desired trajectory. The Real-Time Kinematic (RTK) GPS position sensor was mounted on the implement, and a human driver was used to roughly track the row [vZ98].

³Estimating heading based on position measurements and a vehicle model.

5. *Japan's IAM-BRAIN*. At the **I**nstitute of **A**gricultural **M**achinery Department of Japan's **B**io-oriented **T**echnology **R**esearch **A**dvancement **I**Nstitution, researchers use a geomagnetic direction sensor in conjunction with a modified optoelectric surveying system in an autonomous tractor control system. The sophisticated system uses a combination of buried wires, lasers, and cameras to guide a tractor over a small field. The system controlled speed, the transmission, the hitch, and steering [N⁺97b, N⁺97a].
6. *Ohio State University*. Researchers at Ohio State University outfitted a Case-International Harvester Model 7220 tractor with RTK-GPS and conducted a series of system identification tests aimed at verifying a proposed dynamic model for higher ground speeds. They then used their model to demonstrate tractor control at speeds realistic in spraying applications [S⁺].
7. *Finland's Modulaire*. In Turenki, Finland, the Modulaire Oy company started out with a video-assisted remotely piloted tracked vehicle designed not only for farming, but also for other applications such as explosive ordinance disposal, surveying, and firefighting. The company plans on outfitting production versions of the 1,200 kg (2,650 lb) vehicle with a GPS-based positioning system, and the company has demonstrated automatic control using RTK-GPS coupled with fiber-optic gyros [N⁺96, M⁺95].
8. *Australia's Mailer family and AgSystems Pty Ltd*. In Wonga South, Boggabilla, New South Wales, Australia, the Mailer family is using CPDGPS for automatic tractor guidance along straight rows. Although initially installed as a driver-assist system, they recently installed a steering controller and demonstrated automatic closed-loop control in August 1997. The terrain farmed is flat, and lines are specified by way points at each end. U-turns are done manually. Tracking ability for straight rows is specified at 8-10 cm. Unlike research efforts previously mentioned, the AgSystems effort is a commercial endeavor aimed at increasing farming efficiency [Mai97, Joh97, Nas97].

1.3 Objectives

This research sought to develop practical vehicle estimation and control algorithms enabling a wheeled vehicle to accurately track nonlinear trajectories. Although this research had agricultural applications in mind, the algorithms developed are general enough to be applied to any front-wheel-steered robotic vehicle. Four different types of nonlinear trajectories were identified as important “building blocks” for realistic global paths: arcs, spirals, curves, and contours. Mathematical models of these paths were developed. The models had to be simple enough that locating the vehicle along the trajectory was fast and reliable in real-time. Once each type of path had been specified, tracking control algorithms were developed for a generic robotic vehicle. Because of the computational expense involved in generating the control signal, algorithms with a low computational expense were sought that did not sacrifice too much performance. Controllers were also developed for vehicles operating on sloped terrain.

1.4 Organization

This research is organized as follows:

1. *Problem statement.* Ch. 2 explains the fundamental problem to be solved and the approach taken.
2. *Modeling.*
 - (a) *Vehicle.* In Ch. 3, a nonlinear tractor model based on vehicle geometry was derived which accommodated time-varying conditions such as terrain slope, traction, and vehicle configuration changes.
 - (b) *Towed implement.* In Ch. 4, a three-dimensional nonlinear model of a towed implement was derived based on implement and tractor geometry. The model was linearized so that linear control algorithms could be applied.
3. *Estimation.*

- (a) *Vehicle state estimation.* In Ch. 5, a nonlinear real-time estimator was created that successfully estimated not only the vehicle's position and attitude, but also other parameters crucial to accurate control. The first-order filter was compared in simulation against a second-order filter.
- (b) *Accounting for the lever arm correction.* Since the control point on the vehicle was not collocated with the position measurement point, a vector lever arm correction, which was a nonlinear function of the noisy attitude measurements, had to be added to the original position measurement to calculate the control point position. In Ch. 6, an approximation was derived that showed how the uncertainty in both the measured attitude and the lever arm itself added uncertainty to the position estimate at the control point. The approximation was verified with a Monte Carlo simulation and used to illustrate the adverse effects of poor attitude measurements on control point position uncertainty.

4. *Control.*

- (a) *Tractor control on arcs.* In Ch. 7, a control algorithm was presented for control along arc trajectories. The vehicle model presented in Ch. 3 was rederived in polar coordinates, and experimental results showed accurate tracking control. Some of the material in this chapter as well as in Chs. 8–11 was originally published by the author at the International Association of the Institutes of Navigation's 1997 Conference [B⁺97] and republished by invitation in the Journal of Navigation [B⁺98b].
- (b) *Tractor control on spirals.* Spirals could be used as part of circular irrigation patterns found in arid farm regions such as the southwest United States. In Ch. 8, research from the previous chapter on arcs was extended to spiral trajectories. Reference state information was derived, and experimental results were presented that showed accurate tractor control along spirals.
- (c) *Tractor control on curves.* In Chs. 9–11, cubic splines were used to smooth through discrete position data collected during a human-controlled pass.

Several control algorithms were compared in simulation. Real-time experimental results demonstrated accurate vehicle control. These chapters could be called the heart of this thesis: curved trajectories were the most general trajectories possible, and accurately controlling the tractor along them was the most difficult control problem encountered.

- (d) *Tractor control on sloped terrain.* In Ch. 12, a modified form of bias estimation was presented and experimentally shown to improve row tracking on steep terrain. Some of the material in this chapter as well as in Ch. 13 was originally published by the author at the 1998 Precision Agriculture Conference [B⁺98a].
- (e) *Tractor control on contours.* In Ch. 13, a new algorithm was developed that used the attitude information from the GPS attitude sensor to track a specified contour height. Experimental results verified the algorithm's performance. This research is noteworthy in that no terrain map or other prior knowledge of the terrain was required for accurate control.

1.5 Contributions

This research is the first to present a systems-level solution to steering a wheeled vehicle along different types of non-linear trajectories identified as possible components of realistic global farming trajectories. By using GPS position and attitude sensors, this research offers farmers a system that can track global trajectories to centimeter-level precision in all visibility conditions and without advance field preparation. Because the objective of this research was to demonstrate tracking precision at the limit of the sensors' accuracies, sophisticated regulators were derived that were tailored to each trajectory type. The contributions required to develop these high-accuracy controllers were:

- Developed a wheeled tractor model that could adapt to different vehicle configurations and changing ground conditions. Developed controllers and estimators based on that model, then experimentally demonstrated that the revised model

developed here and a first-order extended Kalman filter yielded improved controller performance by estimating time-varying nonlinear parameters in real-time. An example of typical controller performance along a straight line was 0.9 cm mean and 2.8 cm standard deviation (one-sigma) in row tracking error.

- Expanded the vehicle model to include a three-dimensional model of a generic *towed* implement.
- Performed a generalized error analysis of the additional uncertainty introduced by a noisy vector correction necessary to translate the vehicle's position measurement to the control point. Showed that for a typical tractor configuration, an attitude uncertainty greater than 0.4° (one-sigma) meant that the additional uncertainty introduced by the lever arm eclipsed the nominal CPDGPS measurement uncertainty of 2.5 cm (one-sigma). Supported the assertion that precise attitude measurements were as valuable in CPDGPS-based vehicle control as precise position measurements.
- Developed basic “building-block” controllers for arcs and spirals. Experimentally demonstrated arc tracking performance of -0.3 cm mean and 3.4 cm standard deviation. Spiral tracking performance was -0.2 cm mean and 5.3 cm standard deviation.
- Developed basic building-block controllers for arbitrary curve trajectories. Created a method for generating curved trajectories based on discrete position data using cubic splines. Showed how the spline parameterization could provide data necessary for control algorithms. Developed three real-time tracking control algorithms. Formulated the problem as a nonlinear optimization problem, and derived analytic expressions for the Jacobian and Hessian of the sum of the squares of the tracking error as a function of the curve and control signal sequence. Experimentally demonstrated three tracking control algorithms with an average curve tracking performance of zero to four centimeters mean and four to ten centimeters standard deviation.

- Developed a new bias estimation technique that improved row tracking performance on steeply sloped terrain by 25% over a compensator that used “standard” bias estimation. Despite slopes of up to 15° , tracking was maintained to within -0.8 cm mean and 6.4 cm standard deviation in tracking error under full engine load.
- Developed a general technique to automatically guide a vehicle along the terrain’s contours without prior knowledge of the terrain. Experimentally demonstrated contour tracking to within 0.5 cm mean and 4.3 cm standard deviation in vertical error under realistic operating conditions.

CHAPTER 2

Problem Statement

The ultimate goal of this research was to demonstrate practical limits on tractor control using GPS as a sensor. While a variety of control algorithms may be able to demonstrate stable control on linear trajectories, the non-linear trajectories explored here required relatively sophisticated algorithms to achieve high precision. These algorithms steered a wheeled vehicle so that a specified point, termed the “control point”, on the vehicle tracked one of several different trajectory types. Since the reference trajectories were specified on the ground, the control point was always located on the ground. Because the vehicle was left-right symmetric, the control point was assumed to lie along the centerline of the vehicle. Analysis is also presented for a control point located on a towed implement.

2.1 Approach

A three-dimensional wheeled vehicle model was first developed without regard for real-time computational constraints. The model accurately describes vehicle motion at the relatively low forward speeds typical in farming applications requiring high precision control. Similar analysis was then extended to develop a model for a towed implement attached to the vehicle at a tow pin. Simple examples were used to compare the relative difficulty of steering the vehicle or the towed implement along a linear trajectory.

The following measurements were assumed available: (a) the position of the vehicle’s master GPS antenna¹ relative to the reference station GPS antenna, (b) the vehicle’s attitude, (c) the steer angle of the front wheels, and (d) the relative yaw angle between the vehicle and the implement. Estimators were needed to estimate unmeasurable states, and two non-linear filters were explored as possible solutions.

Complete information about the reference trajectory was assumed known. Analysis showed how to merge geometric information from the different trajectories and the vehicle model to create tracking control algorithms. For all controllers developed, the vehicle was assumed to pivot about the center of its rear axle and lateral slip was assumed to be negligible. Based on observation, these assumptions were valid at low speed and on flat terrain.

2.2 System Overview

A photograph of the first experimental system used in this research, a Deere and Co. Model 7800 tractor, is shown in Fig. (2.1). The same tractor with a towed implement attached is shown in Fig. (2.2). The chisel plow implement was sized for tractors larger than the Model 7800. A physical description of the Model 7800 test tractor is in App. A. Fig. (2.3) shows a block diagram of the complete experimental system.

Although various control techniques such as robust control could have been applied to stabilize the tractor along a trajectory, using these techniques meant sacrificing tracking performance for robustness. Because this research sought to develop high performance controllers that could track trajectories to within the limits of the GPS sensors, robust controllers were neglected in favor of algorithms tailored to the different trajectory types. Many of the difficulties encountered in developing algorithms that delivered high performance over a range of operating conditions arose from the tractor model’s inability to quantify factors such as ballast and soil conditions. These two factors as well as others that had a strong influence on the tractor’s dynamic

¹The GPS attitude system required four GPS antennas. One of these four antennas was designated as the “master” antenna and was used to measure the vehicle’s position relative to the reference station antenna.



Figure 2.1: Deere and Co. Model 7800 farm tractor.



Figure 2.2: Chisel plow used for testing on the Model 7800. This particular towed implement was considered too large for the 7800.

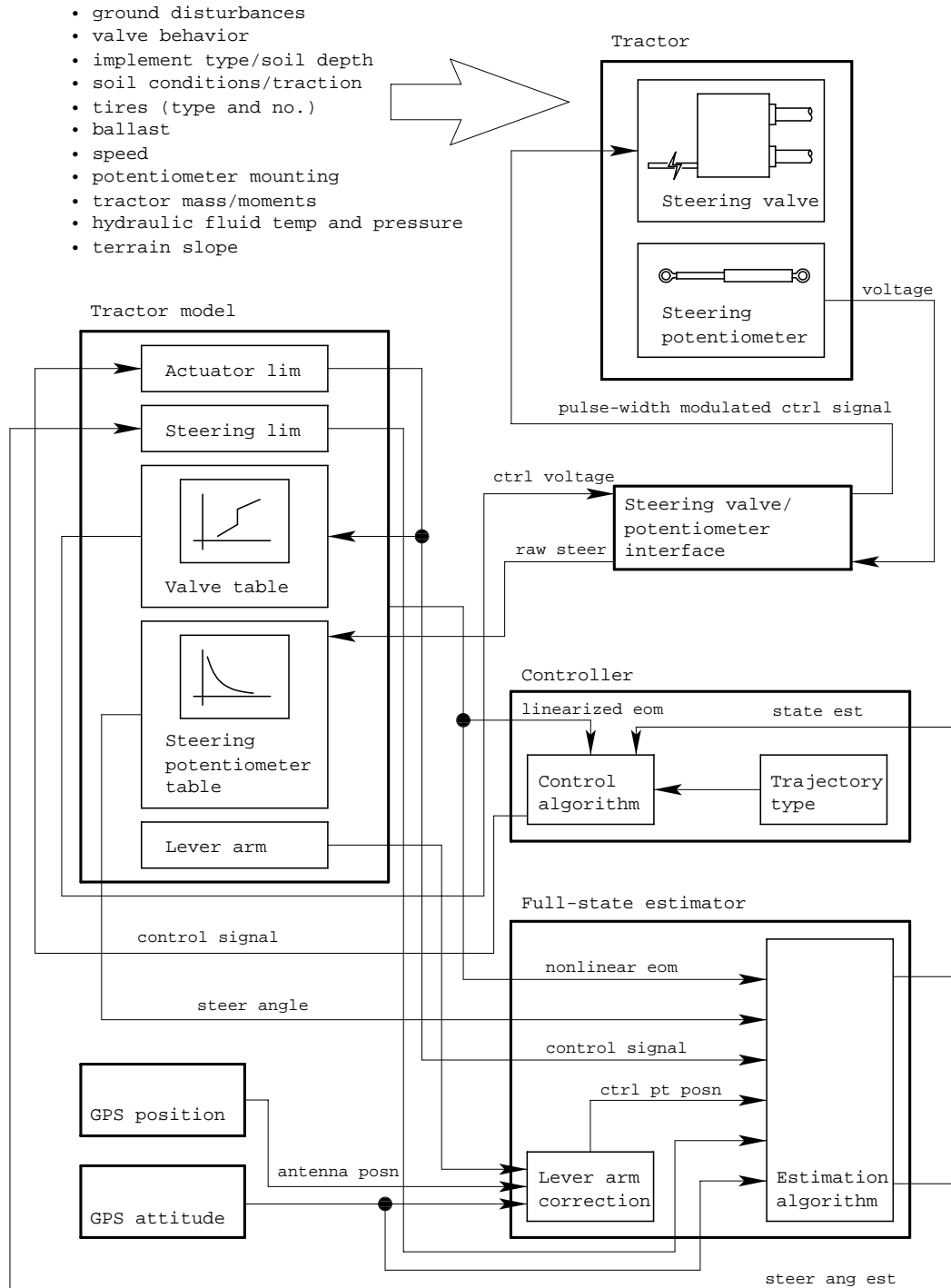


Figure 2.3: Block diagram of experimental system.

response are listed in the upper left corner of Fig. (2.3). Although various sensors could have been installed to measure some of these factors and the user could have specified others through an interface, this research sought to create a practical and relatively inexpensive autofarming system; adding sensors and requirements on user input increase system cost and complexity.

Another complication was limited computational power available to the estimation and control software, which was written by the author in the C programming language. Therefore, the non-linear vehicle model and filter could be used only on the simplest linear trajectories. For more complicated trajectories, the number of states in the model had to be significantly reduced and a linear filter was used to reduce the computational cost. If greater computer power becomes available, the more accurate models, developed here, could be used if warranted. Once vehicle states had been estimated, linear control techniques were usually applied to generate a control signal that steered the vehicle along the trajectory.

2.3 Verification

Regulator performance was judged by the mean and standard deviation of the control point's tracking error, measured by the GPS sensors at 5 Hz, along the trajectory. Although no alternative sensors were available to verify the GPS system's measurements, observation showed that the tractor's path along any trajectory was repeatable to sub-centimeter levels. In addition, previous flight test research has verified the integrity and accuracy of the GPS sensors used in this research [Per96]. Results are presented for real-time control of the vehicle along several different types of trajectories. In all tests, the control point location was the same: a point on the ground directly beneath the center of the rear axle.

CHAPTER 3

Vehicle Modeling

The vehicle model previously developed in [O’C97] assumed various factors such as surface friction and vehicle configuration were constant. Although these and other factors could vary, the model was calibrated for nominal conditions and performed well under those conditions. This chapter develops a model that is applicable over a much wider range of conditions by accounting for terrain and vehicle configuration changes. By developing a model that is accurate over a wider range, high-precision tracking could be maintained in spite of varying conditions.

A rainy morning in March 1998 showed how rapidly surface friction could change. At that time, the Model 7800 tractor detailed in Sec. A.1 of App. A was being tested on dry ground on a farm in the southwest along curved trajectories. At the end of each curve, the vehicle executed an automatic U-turn with a radius of five meters. A sudden rain storm changed the ground conditions to slippery mud within thirty minutes. The tractor’s U-turn radius increased to no less than approximately twenty meters in that time period.¹ Along the curves, the controller did not account for the change in traction, and tracking performance suffered significantly. Had surface friction information been available, tracking degradation could have been minimized. However, other vehicle models that included friction information were found to be relatively high-order in the number of friction coefficients [Won93] which were difficult to identify in real-time using limited sensor information. Most of these models were

¹This increase was brought on in part by the heavy hitched implement attached to the back of the tractor, which significantly *decreased* the down-force on the front wheels. Also note that a tractor operator would have used the tractor’s differential braking capability to turn the tractor under these conditions; the computer could not. A more sophisticated controller might employ such techniques.

developed to describe vehicle behavior on paved highways where friction properties were relatively constant. A low-order model that incorporated time-varying friction information was needed because the linearized model would subsequently be used to compute control gains in real-time.

Research has shown that no one model of an off-road vehicle is widely accepted [Owe82]. The tractor handling study published in [Owe82] showed that at low speeds, steady-state yaw response was governed by vehicle geometry and was roughly linear, with forward velocity up to approximately 3.5 m/s. Factors such as ground surface, tires, axle load ratio, and ground slope significantly affected the steady-state yaw rate. Researchers showed in [CS84] that a more complicated model than the one developed in [O’C97] was required to account for vehicle side-slip, and that the model depended on time-varying parameters such as the friction coefficient(s) between the tractor tires and the ground. Nonetheless, a number of other researchers have shown that a highly simplified vehicle model can remain reasonably accurate at relatively low speeds [Bot82, DW97].

3.1 Factors Influencing Tractor Dynamics

A number of factors were observed to have a strong effect on the tractor dynamics, particularly on the relationship between the front-wheel angle and the yaw rate. These factors are grouped below by the two possible control/estimation techniques that could minimize these factors’ adverse effects on tracking performance.

1. *Lateral slip or “crabbing”*. Various effects forced the tractor to slip sideways, and steady-state tracking errors resulted unless the controller accounted for lateral slip. Lateral slip appeared as a “crab” angle between the tractor heading and the desired heading.² In aviation and shipping, the direction of actual vehicle travel (*i.e.*, the direction of the vehicle’s velocity vector) is called the “course”. The difference between the vehicle’s course and its heading is known as the “drift” angle. In agriculture, the drift angle is more commonly known as the crab

²Note that this crabbing effect is analogous to an airplane landing in the presence of a cross-wind [HJ77].

angle. In the estimator/controller architecture, lateral slip could be handled by estimating yaw and steering angle “biases” or by estimating the vehicle’s lateral slip velocity. Lateral slip could be induced by factors such as:

- (a) *Sloped terrain.* On sloped terrain, the tractor tended to slip downhill. The amount of slip depended on the tires used, soil conditions, and the implement load. Each of these factors could vary widely, and soil conditions and implement load were time-varying. Implement load was observed to vary the most, changing significantly with the terrain slope. Therefore, without prior knowledge of the terrain, the effect of sloped terrain on the tractor dynamics could not be predicted ahead of time.
- (b) *Uneven implement load.* Occasionally, a hitched or towed implement would pull the tractor to one side. This usually happened when soil resistance differed between the two outer edges of the implement. Frequently, a small ridge of soil less than six inches in height remained from the previous pass, and one side of the implement encountered increased resistance from it.
- (c) *Previous field tracks.* The tractor might drive in a direction almost parallel to old paths. In such conditions, the tractor would have a tendency to slide in and out of old tracks or grooves in the ground. This effect was most pronounced when traveling over old furrows. Farmers prepare their fields for precise operations like bedding by smoothing the soil surface with an implement like a disk. If an unprepared field could be accurately bedded through precise tractor control, the farmer could save time and fuel. Therefore, there is an economic incentive to be able to identify and compensate for the tractor’s lateral slip rate.

2. *Front-wheel slip while turning.* The yaw response of the tractor to a steer angle input was affected by a wide variety of factors, most of which were time-varying. Later in this chapter, an improved vehicle model was developed which introduced a proportional relationship between steer angle and yaw rate, a relationship which was variable but could be identified in real-time.

- (a) *Ballasting*.³ Tractor ballast influenced the tractor’s yaw response to a steer angle input. This response depended on the front-axle load. If the tractor was ballasted too lightly at the front, there was less vertical load on the front wheels and hence less traction. Front-axle loading also decreased if a hitched implement was in the raised position. Although front-wheel traction could be improved by changing the ballast, a realistic control system had to be able to deal with improper ballasting as well as changes in the effective ballast due to the changing operating conditions mentioned above.
- (b) *Dual rear wheels*. Attaching a second set of rear wheels (see Fig. 2.1 in App. A) for additional traction and less surface pressure (*i.e.*, less “soil compaction”) affected yaw response. The inside and outside rear wheels on each axle were forced to rotate at the same rate during a turn even though their radial distance from the center of the turn was different. The resulting ground-slip between the dual rear wheels and the ground opposed large yaw rates, resulting in a reduction of the steering effectiveness.
- (c) *Differential lock*. If the rear differential was locked so that both rear axles turned as a unit, rear-wheel ground friction opposed yaw changes because of the different turning radii. Note that the differential was supposed to be unlocked when turning; however, an operator might lock the differential over a gentle curve if the ground was soft and bogging down was a threat.
- (d) *Lowered hitched implement*. If a hitched implement was being used during a turn, the velocity gradient along the implement created a large torque opposing yaw changes. Although hitched implements were not supposed to be in the lowered position during turns, they can be used along gentle curves.
- (e) *Soil conditions*. As previously mentioned, soil conditions affected yaw response. Rain could rapidly change dry soil to slippery mud and increase front-wheel slippage.

³“Ballasting” refers to adding weight to the tractor to increase traction. Weight is usually either cast iron added to axles on a rack on the front of the tractor, or water introduced inside the tires.

3.2 Previous Model

Previous research [O'C97] detailed a general vehicle model based on five states. As shown in Fig. (3.1), vehicle coordinates were described in the Forward-Right-Down frame (vehicle-referenced) or the North-East-Down frame (Earth-referenced). Therefore, yaw and steer angles were defined positive clockwise when viewed from above. The model's states and dynamics were:

- Yaw error between the vehicle's course and the desired trajectory ψ_e .
- Yaw error rate $\dot{\psi}_e$. The equation governing yaw response was

$$\ddot{\psi}_e = \frac{1}{\tau_\psi} (\dot{\psi}_c - \dot{\psi}_e) \quad (3.1)$$

The lag state in yaw captured the effect of the vehicle's rotational inertia. If the front-wheel angle changed suddenly, the yaw rate would not change instantly, but behave as a first-order lag system: the front wheels would skid laterally, then slowly take hold and begin to yaw the vehicle. The variable $\dot{\psi}_c$ was the yaw rate "commanded" by the steer angle. From vehicle geometry and a no-slip assumption [O'C97]:

$$\dot{\psi}_c = \frac{1}{l_1} V_x \tan \delta \quad (3.2)$$

The variable l_1 was the distance between the front and rear axles (see Fig. 3.1). In this chapter, the velocity of the point on the ground beneath the center of the vehicle's (superscript v) rear axle (subscript a) expressed in the vehicle's coordinate frame (underset v) was denoted by

$$\frac{V_a^v}{v} \equiv \begin{bmatrix} V_x \\ V_y \\ V_z \end{bmatrix} \quad (3.3)$$

The time constant τ_ψ in Eqn. (3.1) was identified from experimental data.

- The angle of the front wheels relative to the centerline of the vehicle δ . The model assumed the front wheels effectively acted as one [Won93]. The front wheels could not turn beyond roughly $\pm 35^\circ$.

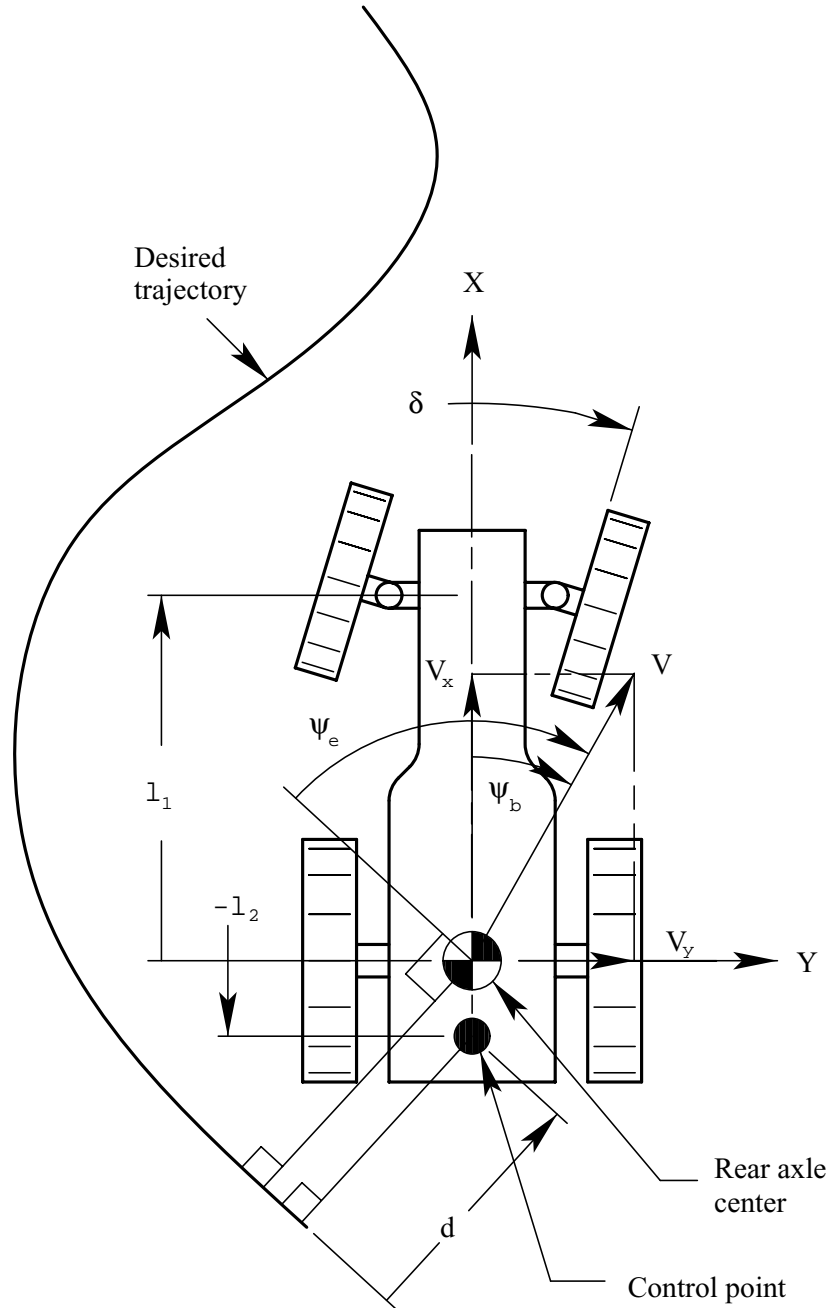


Figure 3.1: Variable definitions for vehicle model. Note that dimension arrows indicate direction of positive value except for l_2 , which is drawn with a negative value.

- The front-wheel angle rate $\dot{\delta}$. The dynamics of the front-wheels were described by

$$\ddot{\delta} = \frac{1}{\tau_u} (\dot{\delta}_c - \dot{\delta}) \quad (3.4)$$

The lag state modeled the dynamics of the hydraulic steering valve. The commanded wheel rate $\dot{\delta}_c$ was the control signal u , which drove a hydraulic pump that turned the front wheels.⁴ Conceptually, u would correspond to how fast the driver was turning the steering wheel. The time constant τ_u was identified from experimental data.

- The tracking error of the vehicle's control point from the desired trajectory d . The tracking error dynamics were described by

$$\dot{d} = V_x \sin \psi_e + (V_y + l_2 \dot{\psi}_e) \cos \psi_e \quad (3.5)$$

This model was created with linear filtering and control algorithms in mind. Therefore, several assumptions were made which could have lead to modeling errors:

1. Equation (3.1) expressed vehicle yaw rate in the inertial frame based on the assumption that the ground was close to level. Vehicle yaw rate should actually be expressed in the vehicle frame. If the vehicle was on level ground, the vehicle and inertial frames' yaw axes were aligned. If, however, the terrain was sloped, the axes were no longer aligned and Eqn. (3.1) was no longer an accurate description of yaw dynamics.
2. Factors outlined in Sec. 3.1 could change both the rate of yaw response and the steady-state yaw rate, which may never reach the commanded value in Eqn. (3.2) because of changes in the vehicle's yaw resistance.
3. Because Eqns. (3.2) and (3.5) were a function of forward velocity, the velocity was estimated separately in real-time. Forward velocity was estimated by finite-differencing position measurements based on the assumption that lateral and

⁴The tractor's front wheels were not mechanically actuated or even power-assisted as in most passenger cars, but rather hydraulically actuated.

vertical velocities were negligible (*i.e.*, $V_y = 0$ and $V_z = 0$), frequently a false assumption on steep or slippery terrain.

4. The parameters τ_ψ and τ_u were calibrated off-line and assumed constant. However, based on the factors mentioned in Sec. 3.1, yaw response (and hence τ_ψ) was time-varying.

While this simple model was adequate for earlier phases of Stanford's research, a model which accounted for the more general three-dimensional vehicle motion was needed because this simple model was developed for specific tractor and field conditions, conditions which often changed rapidly. Although the simpler model would have still been able to control the tractor under different conditions, improved controller performance could be achieved with a model that adapted to changing conditions. The remainder of this chapter details the derivation of this new, more accurate model.

3.3 New, More Accurate Model

3.3.1 Effective Steer Angle, Steering Proportionality Constant, and Steering Bias Angle

Steer angle, the proportionality constant K_δ , and the steering bias angle δ_b were included as states in the improved vehicle model. The steer angle was assumed biased to account for vehicle crabbing and for calibration errors so that the effective steer angle was the measured steer angle δ_m minus the bias:

$$\delta = \delta_m - \delta_b \quad (3.6)$$

The electro-hydraulic valve installed in the newer Model 8400 test tractor (see Sec. A.2 in App. A) had a significantly faster response time than the older valve installed on the Model 7800, and the lag state in steer rate in Eqn. (3.4) was unnecessary for this class of valve. Instead, steer angle rate is described by

$$\dot{\delta} = u \quad (3.7)$$

3.3.2 Attitude

Vehicle attitude Θ^v was required to transform vectors to and from the vehicle and inertial frames. Vehicle yaw rate was redefined to be in the vehicle frame. The lag state in yaw rate was omitted because changes in the vehicle's resistance to yaw maneuvers invalidated Eqn. (3.1). For example, a lowered hitched implement or a locked differential created torques opposing yaw changes. This meant that the vehicle's steady-state yaw rate never reached the value commanded by the front wheels. Mismodeling yaw response as a lag state led to divergent estimates of τ_ψ when experimental data was post-processed through a first-order filter. Instead of modeling the yaw response with a lag state, a proportional relationship was assumed between commanded yaw rate and actual yaw rate. In addition, the lateral slip at the rear wheels had to be accounted for:

$$\frac{\omega_{v/i,z}}{v} = \frac{1}{l_1} K_\delta (V_x \tan \delta - V_y) \quad (3.8)$$

where $\frac{\omega_{v/i,z}}{v}$ is the yaw component of the vehicle's angular rate $\frac{\omega_{v/i}}{v}$ relative to the inertial frame, and K_δ is a proportionality constant. The mean values of the angular rates in roll and pitch were assumed negligible so that

$$\frac{\omega_{v/i}}{v} = \begin{bmatrix} 0 \\ 0 \\ \frac{1}{l_1} K_\delta (V_x \tan \delta - V_y) \end{bmatrix} \quad (3.9)$$

The angular acceleration $\frac{\dot{\omega}_{v/i}}{v}$ can be derived by differentiating Eqn. (3.9) using Eqn. (3.7). Assuming that the nominal acceleration of the vehicle about the ground point beneath the center of the rear axle was zero,⁵

$$\frac{\dot{V}_a^v}{v} = \dot{V} = 0 \quad (3.10)$$

⁵This assumption was false when the operator changed gears or the throttle setting, or when the vehicle's rear wheels had significant side-slip. However, when the implement is in the ground, both the gear setting and throttle position are rarely changed.

Therefore, the angular acceleration vector was

$$\frac{\dot{\omega}_{v/i}}{v} = \begin{bmatrix} 0 \\ 0 \\ \frac{1}{l_1} K_\delta V_x u (1 + \tan^2 \delta) \end{bmatrix} \quad (3.11)$$

Note that the angular acceleration $\frac{\dot{\omega}_{v/i}}{v}$ is the same for both coordinate frames since

$$\frac{\frac{i}{\omega_{v/i}}}{v} = \frac{\frac{v}{\omega_{v/i}}}{v} + \frac{\omega_{v/i}}{v} \times \frac{\omega_{v/i}}{v} \quad (3.12)$$

$$= \frac{\frac{v}{\omega_{v/i}}}{v} \quad (3.13)$$

Therefore, the “dot” notation is used instead of oversetting a frame label.

3.3.3 Position, Velocity, and Crab Angle

To adequately describe the vehicle’s position, the three-dimensional position of the vehicle’s control point in the inertial coordinate frame $\frac{r_c^v}{i}$ and the control point velocity in the vehicle coordinate frame $\frac{V_c^v}{v}$ were necessary. If the control point were not located beneath the center of the rear axle,⁶ a vector correction $\frac{r_0}{v}$ had to be applied to account for the relative movement of the control point. From dynamics [Par95]:

$$\frac{\frac{i}{r_c^v}}{v} = \frac{\frac{i}{r_a^v}}{v} + \frac{\omega_{v/i}}{v} \times \frac{r_0}{v} \quad (3.14)$$

$$= \frac{V_a^v}{v} + \frac{\omega_{v/i}}{v} \times \frac{r_0}{v} = \frac{V_c^v}{v} \quad (3.15)$$

$$\frac{\frac{ii}{r_c^v}}{v} = \frac{\frac{ii}{r_a^v}}{v} + \frac{\dot{\omega}_{v/i}}{v} \times \frac{r_0}{v} + 2 \frac{\omega_{v/i}}{v} \times \frac{v}{v} + \frac{\omega_{v/i}}{v} \times \frac{\omega_{v/i}}{v} \times \frac{r_0}{v} = \frac{\frac{i}{V_c^v}}{v} \quad (3.16)$$

The control point could have been located some distance l_2 forward or behind the rear axle, but was always assumed to be on the ground and along the vehicle’s

⁶The distance l_2 is not usually equal to zero in Fig. (3.1).

centerline so that

$$\frac{r_0}{v} = \begin{bmatrix} l_2 \\ 0 \\ 0 \end{bmatrix} \quad (3.17)$$

The distance l_2 was assumed constant, and

$$\frac{\dot{r}_0}{v} = 0 \quad (3.18)$$

The acceleration of the ground point beneath the rear axle was also zero by Eqn. (3.10).

Finally, the first-order differential equations describing the control point motion were

$$\frac{\dot{r}_c^v}{i} = T_{\frac{i}{v}} \frac{V_c^v}{v} \quad (3.19)$$

$$\frac{\dot{V}_c^v}{v} = \frac{\dot{\omega}_{\frac{v}{i}}}{v} \times \frac{r_0}{v} + \frac{\omega_{\frac{v}{i}}}{v} \times \frac{\omega_{\frac{v}{i}}}{v} \times \frac{r_0}{v} \quad (3.20)$$

where the nonlinear transformation matrix for a 3-2-1 (ψ, θ, ϕ) Euler angle sequence⁷ from the vehicle frame to the inertial frame was [Gre88]

$$T_{\frac{i}{v}} = \begin{bmatrix} \cos \phi \cos \theta & \cos \psi \sin \theta \sin \phi - \sin \psi \cos \phi & \sin \psi \sin \phi + \cos \psi \sin \theta \cos \phi \\ \sin \psi \cos \theta & \cos \psi \cos \phi + \sin \psi \sin \theta \sin \phi & \sin \psi \sin \theta \cos \phi - \cos \psi \sin \phi \\ -\sin \theta & \cos \theta \sin \phi & \cos \theta \cos \phi \end{bmatrix} \quad (3.21)$$

where ψ , θ , and ϕ represent yaw, pitch, and roll, respectively.

The vehicle's yaw bias or crab angle ψ_b , shown in Fig. (3.1), could be estimated from the forward and lateral components of the velocity beneath the rear axle center:

$$\psi_b = \tan^{-1} \left(\frac{\frac{V_{a,y}^v}{v}}{\frac{V_{a,x}^v}{v}} \right) \quad (3.22)$$

⁷Although the Euler angle parameterization has a well-known singularity at pitch angles of $\pm 90^\circ$ [Jun97], if the tractor is pitched straight up or down, most likely the operator has more pressing concerns than whether or not his attitude parameterization has become singular.

where $\frac{V_a^v}{v}$ could be calculated from Eqn. (3.15) using estimates of $\frac{\omega_v}{v}$ and $\frac{V_c^v}{v}$. Since the yaw bias angle could be expressed in terms of $\frac{V_a^v}{v}$ and $\frac{V_c^v}{v}$ was related to $\frac{V_c^v}{v}$ through Eqn. (3.15), estimating both $\frac{V_c^v}{v}$ and ψ_b was redundant. In experiment, the yaw bias was estimated instead of the lateral velocity because estimating the yaw bias was linear while estimating the lateral velocity was not: evaluating the y -component of Eqn. (3.20) using Eqns. (3.9), (3.11), and (3.17),

$$\frac{\frac{i}{V_{c,y}^v}}{v} = \frac{l_1}{l_2} V_x K_\delta u (1 + \tan^2 \delta) \quad (3.23)$$

The dynamics of the control point's lateral velocity were therefore computationally more expensive to implement in a filter.

3.3.4 Equations of Motion

The revised state vector is:

$$x \equiv \begin{bmatrix} \frac{r_c^v}{i} \\ \frac{V_c^v}{v} \\ \Theta^v \\ \delta \\ K_\delta \\ \delta_b \end{bmatrix} \quad (3.24)$$

Neglecting disturbances, the improved equations of motion are

$$\frac{\frac{i}{r_c^v}}{i} = T_{\frac{i}{v}} \frac{V_c^v}{v} \quad (3.25)$$

$$\frac{\frac{i}{V_c^v}}{v} = \frac{\dot{\omega}_{\frac{v}{i}}}{v} \times \frac{r_0}{v} + \frac{\omega_{\frac{v}{i}}}{v} \times \frac{\omega_{\frac{v}{i}}}{v} \times \frac{r_0}{v} \quad (3.26)$$

$$\dot{\Theta}^v = T_{\Theta^v} \frac{\omega_{\frac{v}{i}}}{v} \quad (3.27)$$

$$\dot{\delta} = u \quad (3.28)$$

$$\dot{K}_\delta = 0 \quad (3.29)$$

$$\dot{\delta}_b = 0 \quad (3.30)$$

where the attitude rate transformation matrix used in Eqn. (3.27) is [Gre88]

$$T_{\Theta^v} = \begin{bmatrix} 1 & \tan \theta \sin \phi & \tan \theta \cos \phi \\ 0 & \cos \phi & -\sin \phi \\ 0 & \frac{\sin \phi}{\cos \theta} & \frac{\cos \phi}{\cos \theta} \end{bmatrix} \quad (3.31)$$

3.3.5 Linearized Model

A simplified version of Eqns.(3.25–3.30) can be used when generating control signals. States such as velocity, roll, and pitch were recognized as uncontrollable. This helped create a low-order model that sped up the computationally intensive control algorithms. Assuming zero roll and pitch:⁸

$$\dot{\psi} = \frac{1}{l_1} K_\delta (V_x \tan \delta - V_y) \quad (3.32)$$

$$\dot{\delta} = u \quad (3.33)$$

$$\dot{n} = V_x \cos \psi - (V_y + l_2 \dot{\psi}) \sin \psi \quad (3.34)$$

$$\dot{e} = V_x \sin \psi + (V_y + l_2 \dot{\psi}) \cos \psi \quad (3.35)$$

On flat terrain, V_y was usually negligible, and the equations of motion simplified to

$$\dot{\psi} = \frac{1}{l_1} K_\delta V_x \tan \delta \quad (3.36)$$

$$\dot{\delta} = u \quad (3.37)$$

$$\dot{n} = V_x \cos \psi - l_2 \dot{\psi} \sin \psi \quad (3.38)$$

$$\dot{e} = V_x \sin \psi + l_2 \dot{\psi} \cos \psi \quad (3.39)$$

About a linear trajectory, the model reduced to

$$\dot{\psi}_e = \frac{1}{l_1} K_\delta V_x \tan \delta \quad (3.40)$$

$$\dot{\delta} = u \quad (3.41)$$

$$\dot{d} = V_x \sin \psi_e + l_2 \dot{\psi}_e \cos \psi_e \quad (3.42)$$

⁸Note that the model could be reduced for non-zero roll and pitch as well.

Linearized about zero steer angle and zero yaw error, the model became

$$\dot{\psi}_e \cong \left[\frac{K_\delta V_x}{l_1} \right] \delta \quad (3.43)$$

$$\dot{\delta} = u \quad (3.44)$$

$$\dot{d} \cong V_x \psi_e + \left[V_x \frac{l_2}{l_1} K_\delta \right] \delta \quad (3.45)$$

Note that if V_y was not negligible, linear control and estimation techniques could still have been used: a linear filter could have incorporated the effects of lateral slip by estimating the crab angle as a yaw measurement bias.

3.3.6 Model Analysis

To illustrate how various parameters such as control point location and K_δ could affect controller performance, a LQR controller was created based on the model in Eqns. (3.43–3.45). All states were assumed to be available for feedback and were known perfectly (*i.e.*, no estimator was used). For nominal values of $V_x = 1.0$ m/s, $K_\delta = 1.0$, $l_1 = 2.8$ m, and $l_2 = 0.0$ m, the locus of the discrete system's poles at 5 Hz are shown in Fig. (3.2). Open-loop poles reside at 1.0, reflecting the triple-integrator nature of the control-signal-to-tracking-error plant. As full-state LQR feedback gain is introduced, the poles move inside the unit circle. With the gain values used in experiment, the discrete poles were located at roughly $0.88 \pm 0.17i$ and 0.80. Discrete LQR gains were computed with control effort weighted by $1/u_{max}^2$ and tracking error weighted by $1/d_{max}^2$ [F⁺90]. No other states were penalized. In Fig. (3.3), the norm of the dominant eigenvalue pair is plotted as a function of d_{max} (u_{max} was held constant at 0.38 rad). The dashed vertical line represents the experimental value of d_{max} that was used. Even as the penalty on tracking error is increased (*i.e.*, d_{max} is decreased), the performance of the system does not improve dramatically. Modeling error and limits on control authority and steer angle prevented increasing the penalty on tracking error too much in experiment.

If the control point was moved to the center of a hitched implement, then a reasonable value for l_2 would be -2.0 m. The continuous linearized system is now non-minimum phase with a zero at 0.5 and three poles at the origin. The system

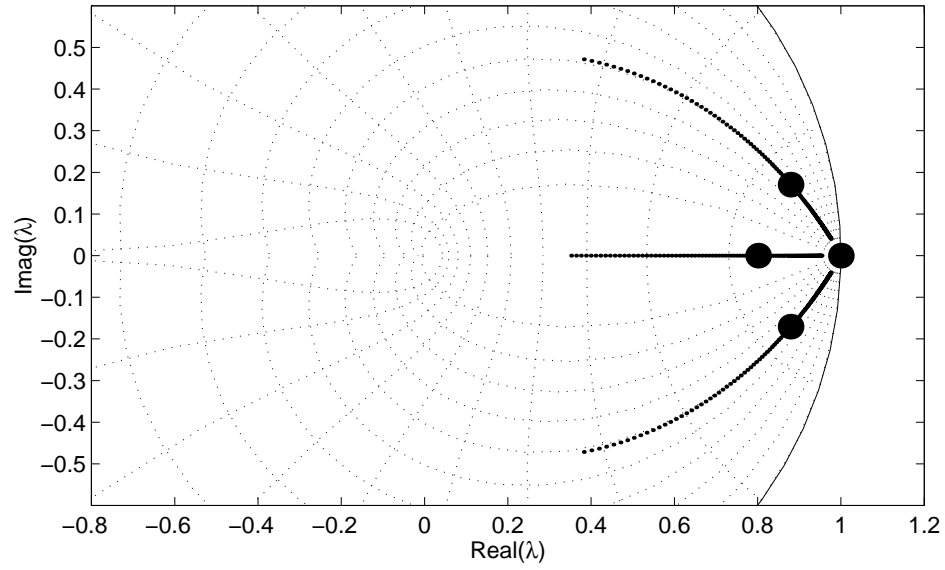


Figure 3.2: Pole locations for 5 Hz discrete closed-loop vehicle model with LQR control gains and full state feedback. Circle(s) at 1.0 are the three open-loop poles, and three other circles at $0.88 \pm 0.17i$ and 0.80 are the closed-loop poles using experimental gains.

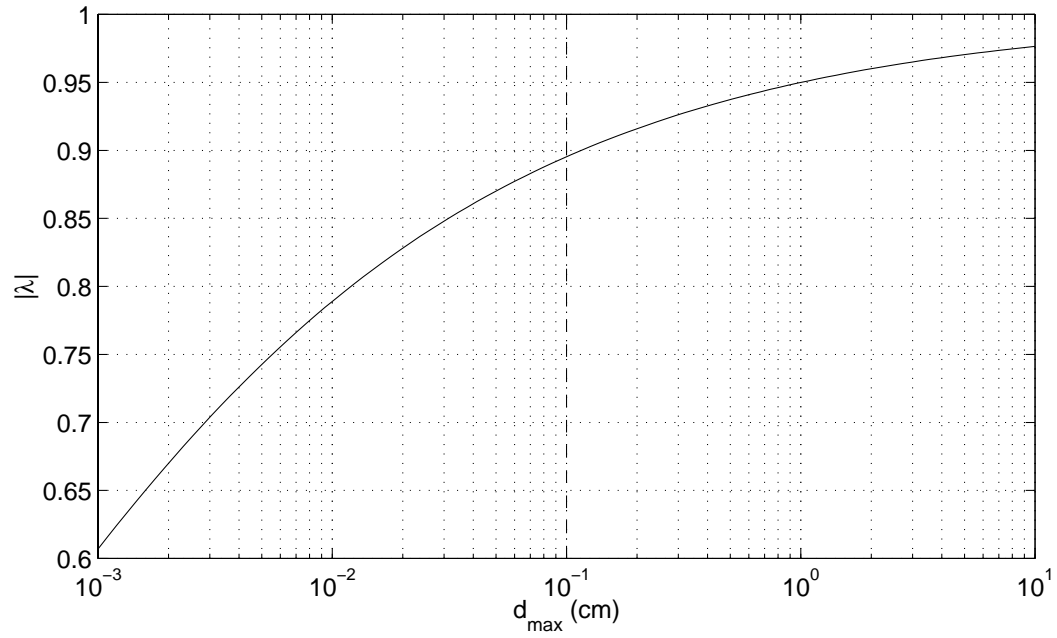


Figure 3.3: Norm of the dominant eigenvalue λ as a function of the tracking error penalty term d_{max} . Vertical dashed line represents the d_{max} value used in experiment.

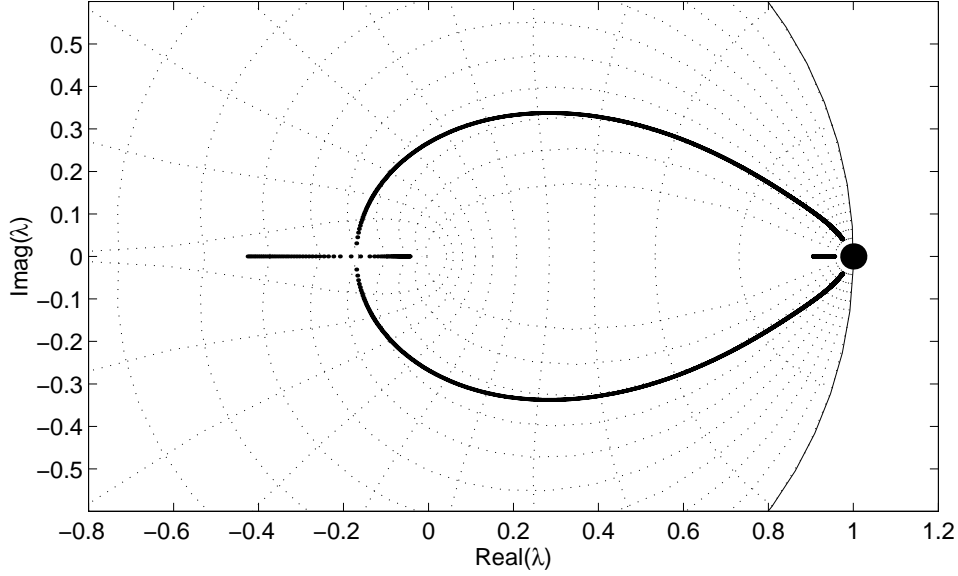


Figure 3.4: Pole locations for 5 Hz discrete closed-loop vehicle model for a hitch control point at $l_2 = -2.0$ m assuming full-state feedback and perfect state measurements. Circle(s) at 1.0 are the three open-loop poles.

also shows greater error: in Fig. (3.5), the distance of the dominant pole to the origin is shown for the discrete root locus in Fig. (3.4). Even as the penalty on error is increased, the dominant pole(s) remain 0.905 away from the origin and represent an upper bound on how fast tracking error could be attenuated.

As the ground conditions deteriorate and front-wheel slip increases, K_δ decreases. Figs. (3.6–3.7) show how K_δ affects the controller’s ability to track a trajectory. In Fig. (3.7), a decreasing K_δ increases the norm of the dominant eigenvalues. A realistic range for K_δ was roughly 0.35 to 1.2. Fig. (3.6) shows how the closed-loop poles change as a function of K_δ for a d_{max} of 10 cm. These results match intuition: controller performance will deteriorate as the front wheels lose their ability to yaw the tractor.

At 5 Hz, the LQR controller’s bandwidth was wide enough that increasing the sampling rate offered little performance improvement. Fig. (3.8) shows the locus of the 20 Hz discrete system’s poles for $V_x = 1.0$ m/s, $K_\delta = 1.0$, and $l_2 = 0.0$ m. The locus is indistinguishable from the 5 Hz system’s locus in Fig. (3.2). Therefore, even

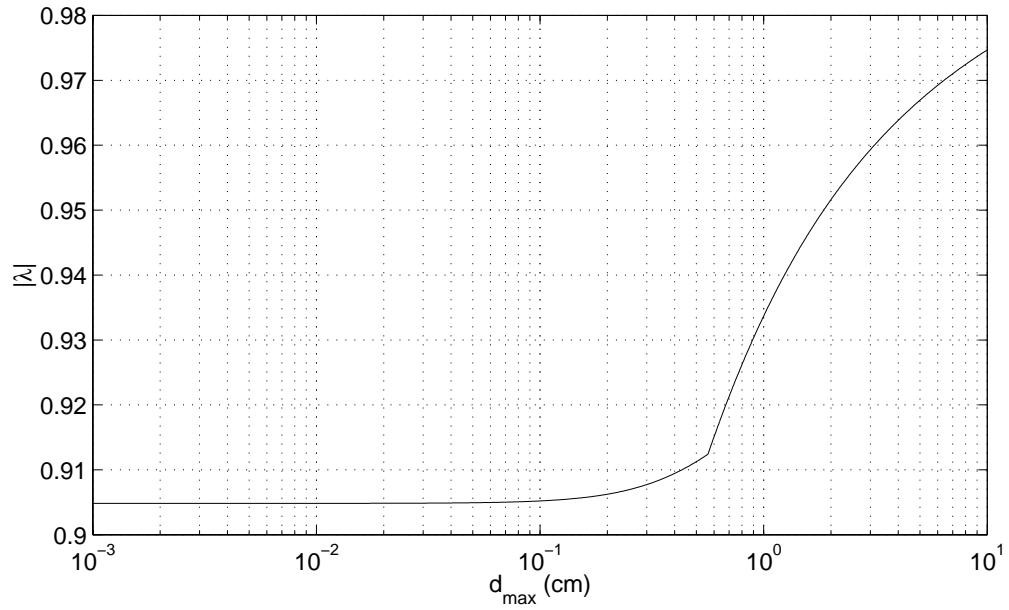


Figure 3.5: Norm of the dominant eigenvalue λ as a function of the tracking error penalty term d_{max} for a hitch control point at $l_2 = -2.0$ m assuming full-state feedback and perfect state measurements.

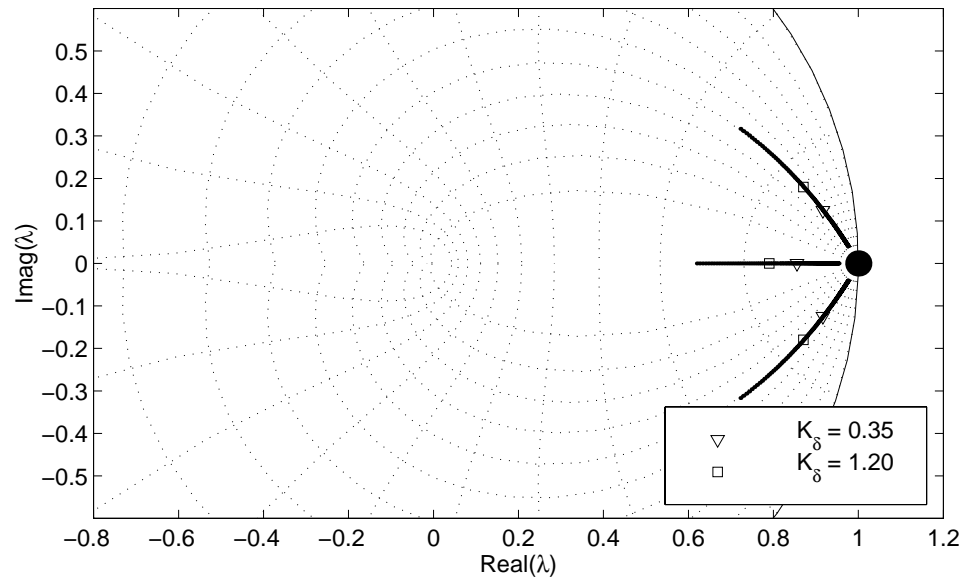


Figure 3.6: Pole locations for 5 Hz discrete closed-loop vehicle model with reduced front-wheel traction as a function of K_δ assuming full-state feedback and perfect measurements. Circle(s) at 1.0 are the three open-loop poles. Triangles and squares denote the minimum and maximum expected values for K_δ .

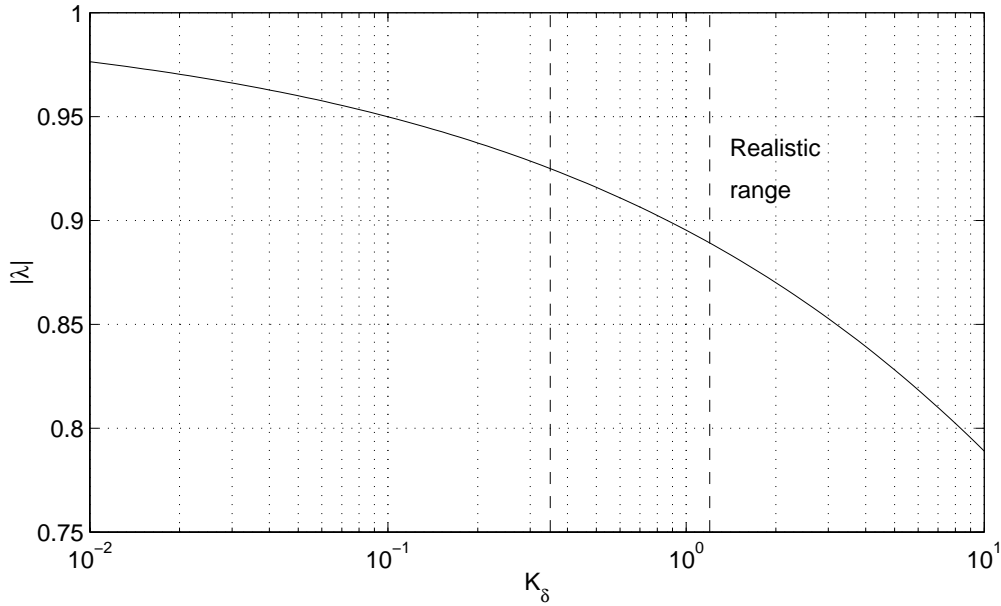


Figure 3.7: Norm of the dominant eigenvalue λ as a function of K_δ assuming full-state feedback and perfect state measurements. The range of K_δ considered realistic is shown.

if the computational power was available to increase the sampling rate from 5 Hz to 20 Hz, the difference in the controller’s tracking ability would be negligible.⁹

3.4 Disturbance Model

The constantly changing ground conditions and the absence of a suspension system on the tractor made an accurate disturbance model difficult to create for all situations. Ground disturbances varied widely depending on how the field had been prepared. A field that had been “disked” smooth was virtually free of significant disturbances, while a field that had been “bedded” had saw-tooth ridges roughly 20 cm in height that could create violent disturbances when the tractor’s forward velocity was above 8 km/h.

Disturbances could affect the tractor through forces in all three axes. In addition

⁹Note that this analysis does not imply that increasing the sampling rate would not improve the *compensator’s* tracking performance. If measurements were available at rates above 5 Hz, the estimator’s state estimate could be improved by faster sampling.

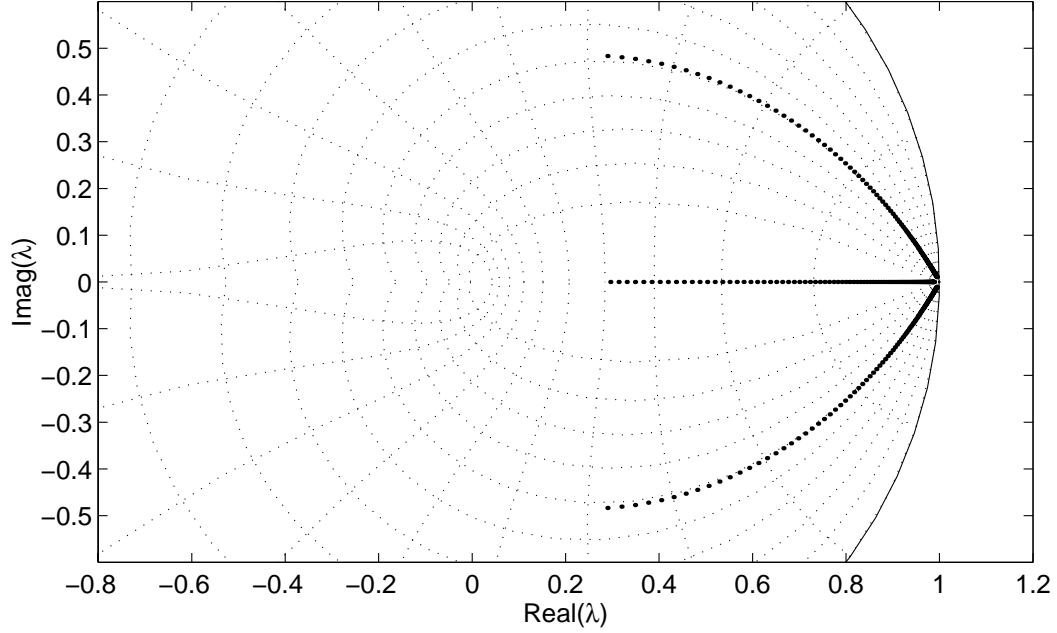


Figure 3.8: Pole locations for 20 Hz discrete closed-loop vehicle model with LQR control gains and full state feedback.

to ground disturbances, the front-wheel angle could also be disturbed by changes in hydraulic pressure.¹⁰ Based on observation, the disturbance forces increased approximately linearly with forward velocity. The assumption of white disturbances was considered relatively accurate under normal conditions.¹¹ An accurate description of ground disturbances' effect on position was therefore

$$\frac{\dot{i}}{V_c^v} = \frac{\dot{\omega}_v}{v} \times \frac{r_0}{v} + \frac{\omega_v}{v} \times \frac{\dot{\omega}_v}{v} \times \frac{r_0}{v} + \frac{V_{c,x}^v}{v} \nu_{V^v} \quad (3.46)$$

Disturbance torques also were observed to increased roughly linearly with velocity. Since the model included attitude but not attitude rate, angular rate had to be added into the state vector to model rotational disturbances accurately. However,

¹⁰Not only the implement but the brakes as well were hydraulically actuated.

¹¹Old tracks or beds on an unprepared field created periodic disturbances that could no longer be deemed white. However, high-precision tractor control would not normally be expected in such conditions.

the assumption was made that

$$\dot{\Theta}^v = T_{\frac{i}{v}} \frac{\omega_v}{v} + \frac{V_{c,x}^v}{v} \nu_{\Theta^v} \quad (3.47)$$

could approximate actual conditions with the caveat that ν_{Θ^v} was no longer white but colored. Steer angle and wheel slippage, accounted for in K_δ , also varied with velocity:

$$\dot{\delta} = u + \frac{V_{c,x}^v}{v} \nu_\delta \quad (3.48)$$

$$\dot{K}_\delta = \frac{V_{c,x}^v}{v} \nu_{K_\delta} \quad (3.49)$$

Only the steering bias was observed to be relatively independent of the tractor's forward speed:

$$\dot{\delta}_b = \nu_{\delta_b} \quad (3.50)$$

Disturbance covariances were estimated empirically. “Fictitious” process noise [F⁺90] was added in the time update to prevent the filter from “going to sleep”. Since disturbance strength often changed rapidly, off-line identification of disturbance covariances was of limited value. Instead, nominal values shown in Tbl. 3.1 based on empirical observation were used in lieu of a real-time disturbance covariance identification algorithm. The large variance in forward velocity was created because shifting the transmission caused large changes in forward speed.

Note that the vehicle model presented in this chapter changed only slightly when the human operator was in control. In this situation, control effort was zero, and the disturbance variance in steer angle was significantly higher. The fact that the model developed in this chapter worked under both control scenarios meant a nonlinear filter could estimate the vehicle's state regardless of how the vehicle was being controlled. The vehicle model created in this chapter also could describe the motion of the tractor in reverse; only the sign of the forward velocity changed.

The twelve-state nonlinear model in Eqns. (3.25–3.30) could be used to estimate the vehicle state and also account for changing ground conditions. The linearized model in Eqns. (3.43–3.45) could then use the information from the nonlinear filter

| Variable | Standard Dev (one- σ) | Name |
|------------------|----------------------------------|---------------------|
| ν_{V_x} | 0.05 s^{-1} | <i>Velocity</i> |
| ν_{V_y} | 0.005 s^{-1} | <i>disturbances</i> |
| ν_{V_z} | 0.005 s^{-1} | |
| ν_ϕ | 0.2 deg-m^{-1} | <i>Angular</i> |
| ν_θ | 0.2 deg-m^{-1} | <i>disturbances</i> |
| ν_ψ | 0.5 deg-m^{-1} | |
| ν_δ | 0.5 deg-m^{-1} | <i>Steering-</i> |
| ν_{K_δ} | 0.001 m^{-1} | <i>related</i> |
| ν_{δ_b} | 0.03 deg-s^{-1} | <i>disturbances</i> |

Table 3.1: Estimated disturbance variances for the tractor model.

to form a low-order linearized model to compute linear control gains rapidly in real-time.

CHAPTER 4

Towed Implement Modeling

Agricultural implements commonly come in two forms: hitched and towed. A hitched implement is mounted to a hitch on the tractor usually located on the back of the tractor. The hydraulically powered hitch raises and lowers the implement into the soil and can sway approximately 5° from side to side. Hitched implements are usually used in “row-crop” applications such as listing (“bedding”) or cultivating where high precision control is needed. The disadvantage of a hitched implement is that significant yaw rates or side-slip can impart strong lateral forces on the lowered implement as the implement resists lateral movement through the soil. These lateral forces can damage the implement or significantly increase component wear. Hitched implement size is partly limited by the hitch power available to lift the implement and by the tractor’s weight. A raised hitched implement decreases the available down-force on the tractor’s front wheels. In cases where a hitched implement is too heavy for a tractor, the front wheels have little traction and the yaw response to steering input is so poor that changing the vehicle’s direction requires either advance planning or differential braking.

A towed implement attaches to a tow pin mounted on a drawbar at the rear of the tractor. Towed implements can be heavier than hitched implements because they do not have to be lifted completely off the ground. Towed implements are usually used in less-precise agricultural applications. If a towed implement could be controlled as precisely as a hitched implement, farmers might be able to use larger towed implements in some applications previously reserved for lighter hitched implements, decreasing

the work time. Additionally, towed implements can easily be used on curved trajectories while hitched implements can not. With this in mind, this research derived the equations of motion for a specified control point on a towed agricultural implement with a goal of real-time accurate control.

Although control of a towed trailer is a well-known problem, previous research on modeling trailers assumed no significant roll and pitch of the tractor. These research efforts were aimed at either controlling tractor-trailer robots in factories or offices [H⁺97] or on highways [SF89, Bra91, B⁺96]; in both scenarios, the terrain can safely be assumed flat. In [H⁺97], a fuzzy controller looked ahead a finite number of steps along the desired trajectory and generated a control signal that forced the simulated tractor to track the trajectory and avoid obstacles inside a factory. In [S⁺95], the equations of motion for the tractor-trailer were derived based on the assumptions of flat ground for path planning purposes. These researchers assumed the steering wheel angle was constant and were able to obtain a closed-form solution for the system state as a function of time. Unfortunately, they also found that the equations of motion could not be integrated if the steering wheel angle was time-varying, as is the case for the test platforms used in this research.

4.1 Dynamic Model

4.1.1 Overview

To derive the motion of a control point on a towed implement, the motion of the tow pin in the vehicle coordinate frame was transformed into the implement frame. The non-holonomic constraint that the implement move about some pivot point was used to calculate the pivot point's angular and linear rates and accelerations. Finally, the motion of the control point about the pivot point was calculated.

In experiment, it was observed that the working portion of the implement that was buried in the ground resisted lateral motion. Although the resistance to lateral motion varied between implements, there typically was enough resistance to lateral slip that, to first order, a towed implement could be assumed to pivot about some "center of

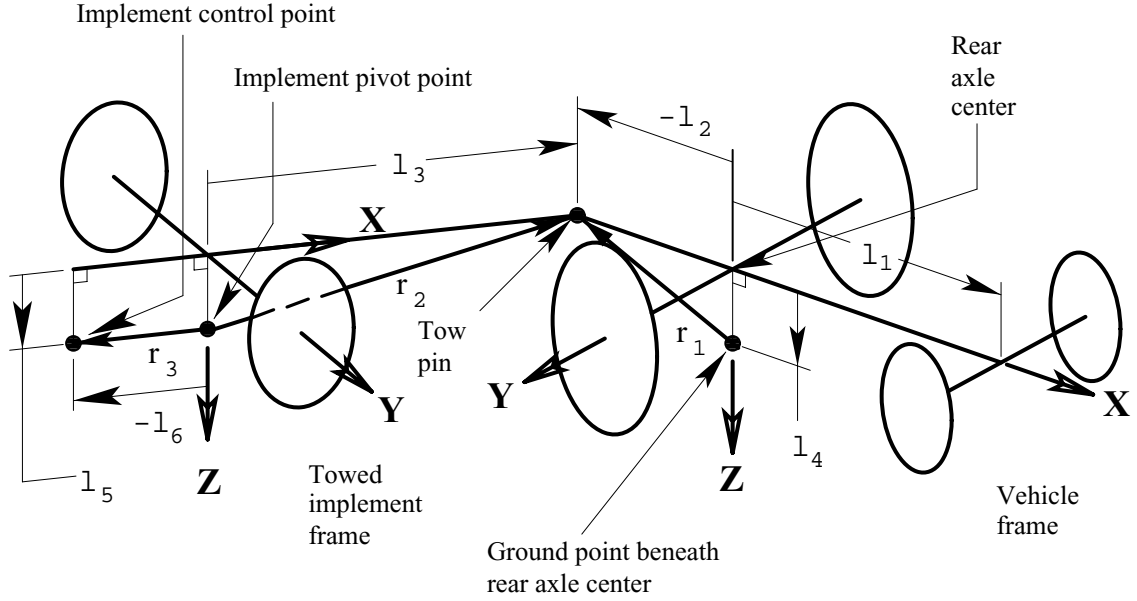


Figure 4.1: Variable definitions for towed implement model.

force” in the middle of the implement’s tools. For simplicity, this center of force was assumed constant and located on the ground directly beneath the implement’s axle at a distance l_3 behind the attachment point (see Fig. 4.1).¹

4.1.2 Equations of Motion

The velocity \underline{V}_p and acceleration $\underline{\ddot{V}}_p$ of the vehicle’s *tow pin* were derived in an identical manner as the expression for the velocity and acceleration of the vehicle’s *control point* except that the vector \underline{r}_0 in Eqn. (3.17) was replaced by a vector \underline{r}_1 extending from the ground point beneath the center of the rear axle to the tow pin as shown in Fig. (4.1):

¹Another incentive to assume the axle is located at the center of force is that the equations of motion developed in this chapter then extend to any towed trailer as long as there is no lateral slip at the trailer’s wheels.

$$\frac{r_1}{v} = \begin{bmatrix} l_2 \\ 0 \\ l_4 \end{bmatrix} \quad (4.1)$$

Therefore,

$$\frac{V_p}{t} = T_{\frac{t}{v}} \frac{V_p}{v} \quad (4.2)$$

$$= T_{\frac{t}{v}} \left[\frac{V_a^v}{v} + \frac{\omega_{\frac{v}{i}} \times r_1}{v} \right] \quad (4.3)$$

Using Eqn. (3.20),

$$\frac{\frac{i}{V_p}}{t} = T_{\frac{t}{v}} \frac{\frac{i}{V_p}}{v} \quad (4.4)$$

$$= T_{\frac{t}{v}} \left[\frac{\dot{\omega}_{\frac{v}{i}} \times r_1}{v} + \frac{\omega_{\frac{v}{i}} \times \frac{\omega_{\frac{v}{i}} \times r_1}{v}}{v} \right] \quad (4.5)$$

The angular rates and accelerations of the implement about its pivot point can be calculated from the non-holonomic constraint that the implement pivots about its pivot point. The situation is slightly more complicated when the height of the tow pin above the ground (l_5) is non-zero. In this case, calculating the angular rates imparted on the implement by the tow pin's velocity requires transforming the tow pin's velocity vector yet again so that the vector is aligned with a line between the implement's pivot point and the tow pin. In Fig. (4.1), this line is shown as the vector r_2 . Once the tow pin velocity vector has been coordinatized in the implement frame from Eqn. (4.3), performing the second transformation to align the vector with r_2 is relatively simple. The vector r_2 is pitched downward at an angle $\tan^{-1}(-l_5/l_3)$. Therefore, the transformed velocity is

$$\frac{V_p}{r_2} = \frac{1}{\sqrt{l_3^2 + l_5^2}} \begin{bmatrix} l_3 & 0 & -l_5 \\ 0 & 1 & 0 \\ l_5 & 0 & l_3 \end{bmatrix} \frac{V_p}{t} \quad (4.6)$$

where underset r_2 denotes the r_2 coordinate frame. Now that the tow pin velocity vector has been transformed into the r_2 coordinate frame, the angular rate of the

implement can be calculated (also in the r_2 frame):

$$\frac{\omega_{\frac{t}{i}}}{r_2} = \frac{1}{\|r_2\|} \begin{bmatrix} 0 \\ -\frac{V_{p,z}}{r_2} \\ \frac{V_{p,y}}{r_2} \end{bmatrix} \quad (4.7)$$

Transforming the angular velocity vector back into the implement frame and simplifying terms,

$$\frac{\omega_{\frac{t}{i}}}{t} = \frac{1}{l_3^2 + l_5^2} \begin{bmatrix} 0 & l_5 & 0 \\ -l_5 & 0 & -l_3 \\ 0 & l_3 & 0 \end{bmatrix} \frac{V_p}{t} \quad (4.8)$$

$$\frac{\dot{\omega}_{\frac{t}{i}}}{t} = \frac{1}{l_3^2 + l_5^2} \begin{bmatrix} 0 & l_5 & 0 \\ -l_5 & 0 & -l_3 \\ 0 & l_3 & 0 \end{bmatrix} \frac{\dot{V}_p}{t} \quad (4.9)$$

Only the x -component of the tow pin's velocity contributes to the implement's pivot point velocity $\frac{V_a^t}{r_2}$, since only the forward velocity along the r_2 axis affects the implement velocity:

$$\frac{V_a^t}{r_2} = \frac{V_{p,x}}{r_2} \quad (4.10)$$

$$= \frac{1}{\sqrt{l_3^2 + l_5^2}} \begin{bmatrix} l_3 & 0 & -l_5 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \frac{V_p}{t} \quad (4.11)$$

Transforming Eqn. (4.11) into the implement frame,

$$\frac{V_a^t}{t} = \frac{1}{l_3^2 + l_5^2} \begin{bmatrix} l_3^2 & 0 & -l_3 l_5 \\ 0 & 0 & 0 \\ l_3 l_5 & 0 & -l_5^2 \end{bmatrix} \frac{V_p}{t} \quad (4.12)$$

Similarly, Eqn. (4.5) can be transformed so that the acceleration of the implement's pivot point is

$$\frac{\frac{i}{V_a^t}}{t} = \frac{1}{l_3^2 + l_5^2} \begin{bmatrix} l_3^2 & 0 & -l_3 l_5 \\ 0 & 0 & 0 \\ l_3 l_5 & 0 & -l_5^2 \end{bmatrix} \frac{\frac{i}{V_p}}{t} \quad (4.13)$$

If $\frac{r_3}{t}$ is the (constant) vector from the implement's pivot point to its control point, the velocity and acceleration vectors of the implement control point in the implement frame were

$$\frac{V_c^t}{t} = \frac{V_a^t}{t} + \frac{\omega_{\frac{i}{t}} \times r_3}{t} \quad (4.14)$$

$$\frac{\frac{i}{V_c^t}}{t} = \frac{\frac{i}{V_a^t}}{t} + \frac{\dot{\omega}_{\frac{i}{t}} \times r_3}{t} + \frac{\omega_{\frac{i}{t}} \times \omega_{\frac{i}{t}} \times r_3}{t} \quad (4.15)$$

Implement Control Point Position and Velocity

The implement model now augments the vehicle's state vector by nine additional states for a total of 21 states: the position of the implement in the inertial frame, the velocity of the implement in the implement frame, and the implement attitude. The equations of motion for the implement's control point were

$$\frac{\frac{i}{r_c^t}}{t} = T_{\frac{i}{t}} \frac{V_c^t}{t} \quad (4.16)$$

$$\frac{\frac{i}{V_c^t}}{t} = \frac{\frac{i}{V_a^t}}{t} + \frac{\dot{\omega}_{\frac{i}{t}} \times r_3}{t} + \frac{\omega_{\frac{i}{t}} \times \omega_{\frac{i}{t}} \times r_3}{t} \quad (4.17)$$

Using Eqns. (4.13) and (4.5), Eqn. (4.17) becomes

$$\frac{\dot{V}_c^t}{t} = \frac{1}{l_3^2 + l_5^2} \begin{bmatrix} l_3^2 & 0 & -l_3 l_5 \\ 0 & 0 & 0 \\ l_3 l_5 & 0 & -l_5^2 \end{bmatrix} \left[\frac{\dot{V}_p^t}{t} + \frac{\dot{\omega}_t^t \times r_3}{t} + \frac{\omega_t^t \times \omega_t^t \times r_3}{t} \right] \quad (4.18)$$

$$\begin{aligned} &= \frac{1}{l_3^2 + l_5^2} \begin{bmatrix} l_3^2 & 0 & -l_3 l_5 \\ 0 & 0 & 0 \\ l_3 l_5 & 0 & -l_5^2 \end{bmatrix} T_{\frac{t}{v}} \left[\frac{\dot{\omega}_v^v \times r_1}{v} + \frac{\omega_v^v \times \omega_v^v \times r_1}{v} \right] \\ &\quad + \frac{\dot{\omega}_t^t \times r_3}{t} + \frac{\omega_t^t \times \omega_t^t \times r_3}{t} \end{aligned} \quad (4.19)$$

Vehicle “Axle Point” Position and Velocity

Instead of estimating the position and velocity of the vehicle’s control point, the position and velocity of the ground point beneath the vehicle’s rear axle center are estimated instead:

$$\frac{\dot{r}_a^v}{i} = T_{\frac{i}{v}} \frac{V_a^v}{v} \quad (4.20)$$

From Eqn. (3.10),

$$\frac{\dot{V}_a^v}{v} = 0 \quad (4.21)$$

Vehicle Attitude

Vehicle attitude dynamics were given in Eqn. (3.27) as

$$\dot{\Theta}^v = T_{\Theta^v} \frac{\omega_v^v}{v} \quad (4.22)$$

Implement Attitude

Implement attitude dynamics can be expressed using Eqns. (4.8) and (4.3) as

$$\dot{\Theta}^t = T_{\Theta^t} \omega_{\frac{t}{t}} \quad (4.23)$$

$$= \frac{T_{\Theta^t}}{l_3^2 + l_5^2} \begin{bmatrix} 0 & l_5 & 0 \\ -l_5 & 0 & -l_3 \\ 0 & l_3 & 0 \end{bmatrix} \frac{V_p}{t} \quad (4.24)$$

$$= \frac{T_{\Theta^t}}{l_3^2 + l_5^2} \begin{bmatrix} 0 & l_5 & 0 \\ -l_5 & 0 & -l_3 \\ 0 & l_3 & 0 \end{bmatrix} T_{\frac{t}{v}} \left[\frac{V_a^v}{v} + \frac{\omega_{\frac{v}{t}}}{v} \times \frac{r_1}{v} \right] \quad (4.25)$$

Effective Steer Angle, Steering Proportionality Constant, and Steering Bias Angle

Finally, Eqns. (3.28–3.30) for the dynamics of the steering angle, proportionality constant, and steering bias are unchanged:

$$\dot{\delta} = u \quad (4.26)$$

$$\dot{K}_\delta = 0 \quad (4.27)$$

$$\dot{\delta}_b = 0 \quad (4.28)$$

4.1.3 Linearized Model

As in Ch. 3, several states were recognized as uncontrollable and could not be included in a model used for controlling the implement. In addition, a simpler lower-order model was needed to calculate the control signal in real-time. Therefore, the towed implement model was reduced to four states: vehicle yaw error ψ_e , effective steer angle δ , implement yaw angle ψ_e^t , and the tracking error of the implement control point d^t . For zero roll and pitch in both vehicle and implement, expressions for $\dot{\psi}$

and $\dot{\delta}$ were given in Eqns. (3.36) and (3.37). Expressions for $\dot{\psi}^t$, \dot{n}^t , and \dot{e}^t are

$$\dot{\psi}^t = \frac{l_3}{l_3^2 + l_5^2} \begin{bmatrix} \cos(\psi^t - \psi) \left(V_y + \frac{l_2}{l_1} K_\delta (V_x \tan \delta - V_y) \right) \\ - V_x \sin(\psi^t - \psi) \end{bmatrix} \quad (4.29)$$

$$\begin{aligned} \dot{n}^t &= \frac{l_3}{l_3^2 + l_5^2} \cos \psi^t \begin{bmatrix} l_3 \sin(\psi^t - \psi) \left(V_y + \frac{l_2}{l_1} K_\delta (V_x \tan \delta - V_y) \right) \\ + l_3 V_x \cos(\psi^t - \psi) - l_5 V_z \end{bmatrix} \\ &\quad - \frac{l_3 l_6}{l_3^2 + l_5^2} \sin \psi^t \begin{bmatrix} \cos(\psi^t - \psi) \left(V_y + \frac{l_2}{l_1} K_\delta (V_x \tan \delta - V_y) \right) \\ - V_x \sin(\psi^t - \psi) \end{bmatrix} \end{aligned} \quad (4.30)$$

$$\begin{aligned} \dot{e}^t &= \frac{l_3}{l_3^2 + l_5^2} \sin \psi^t \begin{bmatrix} l_3 \sin(\psi^t - \psi) \left(V_y + \frac{l_2}{l_1} K_\delta (V_x \tan \delta - V_y) \right) \\ + l_3 V_x \cos(\psi^t - \psi) - l_5 V_z \end{bmatrix} \\ &\quad + \frac{l_3 l_6}{l_3^2 + l_5^2} \cos \psi^t \begin{bmatrix} \cos(\psi^t - \psi) \left(V_y + \frac{l_2}{l_1} K_\delta (V_x \tan \delta - V_y) \right) \\ - V_x \sin(\psi^t - \psi) \end{bmatrix} \end{aligned} \quad (4.31)$$

On flat terrain, V_y and V_z are usually negligible, simplifying Eqns. (4.29–4.31) to

$$\dot{\psi}^t = \frac{l_3}{l_3^2 + l_5^2} \begin{bmatrix} \cos(\psi^t - \psi) \frac{l_2}{l_1} K_\delta V_x \tan \delta - V_x \sin(\psi^t - \psi) \end{bmatrix} \quad (4.32)$$

$$\begin{aligned} \dot{n}^t &= \frac{l_3}{l_3^2 + l_5^2} \cos \psi^t \begin{bmatrix} l_3 \sin(\psi^t - \psi) \frac{l_2}{l_1} K_\delta V_x \tan \delta + l_3 V_x \cos(\psi^t - \psi) \end{bmatrix} \\ &\quad - \frac{l_3 l_6}{l_3^2 + l_5^2} \sin \psi^t \begin{bmatrix} \cos(\psi^t - \psi) \frac{l_2}{l_1} K_\delta V_x \tan \delta - V_x \sin(\psi^t - \psi) \end{bmatrix} \end{aligned} \quad (4.33)$$

$$\begin{aligned} \dot{e}^t &= \frac{l_3}{l_3^2 + l_5^2} \sin \psi^t \begin{bmatrix} l_3 \sin(\psi^t - \psi) \frac{l_2}{l_1} K_\delta V_x \tan \delta + l_3 V_x \cos(\psi^t - \psi) \end{bmatrix} \\ &\quad + \frac{l_3 l_6}{l_3^2 + l_5^2} \cos \psi^t \begin{bmatrix} \cos(\psi^t - \psi) \frac{l_2}{l_1} K_\delta V_x \tan \delta - V_x \sin(\psi^t - \psi) \end{bmatrix} \end{aligned} \quad (4.34)$$

Along a linear trajectory, the model reduced to

$$\dot{\psi}_e^t = \frac{l_3}{l_3^2 + l_5^2} \begin{bmatrix} \cos(\psi_e^t - \psi_e) \frac{l_2}{l_1} K_\delta V_x \tan \delta - V_x \sin(\psi_e^t - \psi_e) \end{bmatrix} \quad (4.35)$$

$$\begin{aligned} \dot{d}^t &= \frac{l_3}{l_3^2 + l_5^2} \sin \psi_e^t \begin{bmatrix} l_3 \sin(\psi_e^t - \psi_e) \frac{l_2}{l_1} K_\delta V_x \tan \delta + l_3 V_x \cos(\psi_e^t - \psi_e) \end{bmatrix} \\ &\quad + \frac{l_3 l_6}{l_3^2 + l_5^2} \cos \psi_e^t \begin{bmatrix} \cos(\psi_e^t - \psi_e) \frac{l_2}{l_1} K_\delta V_x \tan \delta - V_x \sin(\psi_e^t - \psi_e) \end{bmatrix} \end{aligned} \quad (4.36)$$

Linearizing the model about zero vehicle yaw error, zero steer angle, and zero implement yaw error, the linearized equations of motion are

$$\dot{\psi}_e = \left[\frac{1}{l_1} V_x K_\delta \right] \delta \quad (4.37)$$

$$\dot{\delta} = u \quad (4.38)$$

$$\dot{\psi}_e^t = \left[\frac{l_3}{l_3^2 + l_5^2} V_x \right] \psi_e - \left[\frac{l_3}{l_3^2 + l_5^2} V_x \right] \psi_e^t + \left[\frac{l_2 l_3}{l_1 (l_3^2 + l_5^2)} K_\delta V_x \right] \delta \quad (4.39)$$

$$\dot{d}^t = \left[\frac{l_3 l_6}{l_3^2 + l_5^2} V_x \right] \psi_e + \left[\frac{l_3 (l_3 - l_6)}{l_3^2 + l_5^2} V_x \right] \psi_e^t + \left[\frac{l_2 l_3 l_6}{l_1 (l_3^2 + l_5^2)} K_\delta V_x \right] \delta \quad (4.40)$$

4.1.4 Model Analysis

To illustrate controller performance, a LQR controller was created based on the linearized model. All states were assumed to be available for feedback and were known perfectly (*i.e.*, no estimator was used). For nominal values of $V_x = 1.0$ m/s, $V_x = V_y = 0.0$ m/s, $K_\delta = 1.0$, $l_1 = 2.8$ m, $l_2 = -1.5$ m, $l_3 = 3.0$ m, $l_4 = 0.5$ m, $l_5 = 0.4$ m, and $l_6 = -0.5$ m, Fig. (4.2) shows the root locus for the continuous-time linearized model. The open-loop continuous linearized model of Eqns. (4.37–4.40) has three poles at the origin and the fourth at -0.33, showing how the implement will tend to straighten itself when pulled forward. The continuous-time transfer function for the tracking error shows two zeros in the right-half plane, one at 1.97 and the other at 0.67. These two zeros stem from the fact that both the tow pin and the implement control point are located behind the axles of the vehicle and implement respectively in this example model.

In Figs. (4.3–4.4), the tracking error and steering history are plotted as solid lines for the linearized vehicle/implement system under the relatively mild initial conditions $\psi_0 = 5.3^\circ$, $\delta_0 = 0.1^\circ$, $\psi_{e,0}^t = 8.9^\circ$, and $d_0 = 2.5$ cm.² Note that d_0 is the initial tracking error of the *implement*. In the same figures, the dashed lines are the response of the vehicle alone with no implement attached given the same initial conditions. Here, d_0

²The eigenvector (from the solution of the discrete control Riccati equation) corresponding to the largest closed-loop eigenvalue represents the ratio of the worst initial conditions for the closed-loop system. These initial conditions were chosen using that eigenvector to highlight the differences in system performance.

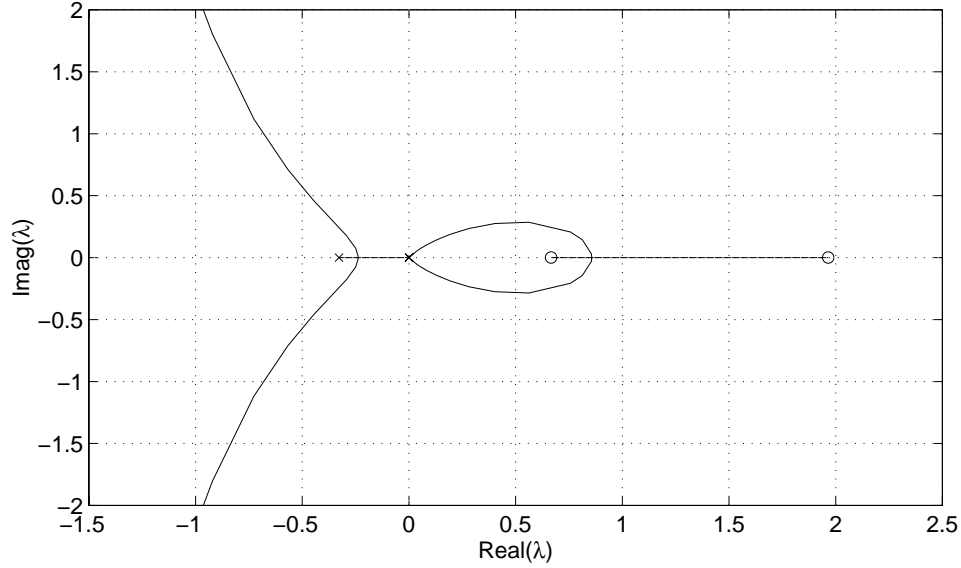


Figure 4.2: Continuous-time root locus of the linearized vehicle/implement model. Note the two right-half plane zeros.

was the initial tracking error of the *vehicle's* control point with l_2 now set to zero. Both controllers used in the simulation used the same penalty functions u_{max} and d_{max} when generating the LQR control gain vector.

Analysis of the two plots shows that although both controllers manage to zero out the tracking error relatively quickly, the implement controller had to use a significantly greater amount of steering to do so and violated the steering angle constraints shown as two horizontal dashed lines (see Fig. 4.4). In the actual nonlinear system, encountering the steering limits would have degraded performance.

Another problem associated with towed implement control is that without an attitude sensor on the implement, the implement attitude must be estimated. Implement attitude is required in the transformation matrix $T_{\frac{i}{t}}$ to measure the control point position:

$$\frac{r_c^t}{i} = \frac{r_m}{i} + T_{\frac{i}{v}} \frac{r_p}{v} + T_{\frac{i}{t}} \left(\frac{r_2}{t} + \frac{r_3}{t} \right) \quad (4.41)$$

where $\frac{r_p}{v}$ is a vector from the vehicle's master GPS antenna to the tow pin. One possible estimate of the implement's roll and pitch could come from the knowledge

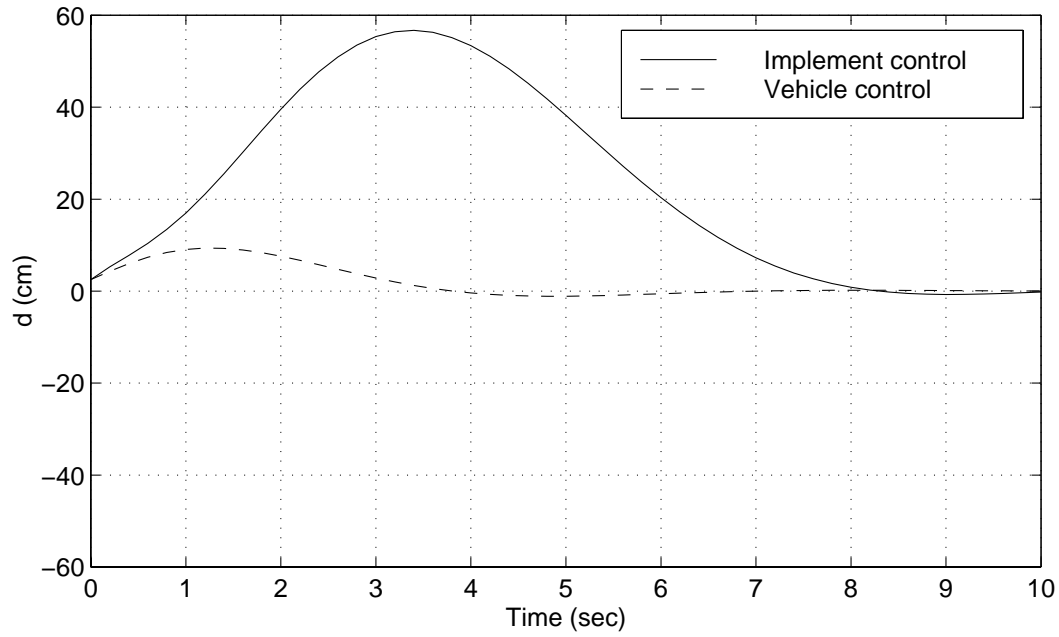


Figure 4.3: Tracking error response for both the vehicle (dashed line) and vehicle/implement (solid line) for the same initial conditions.

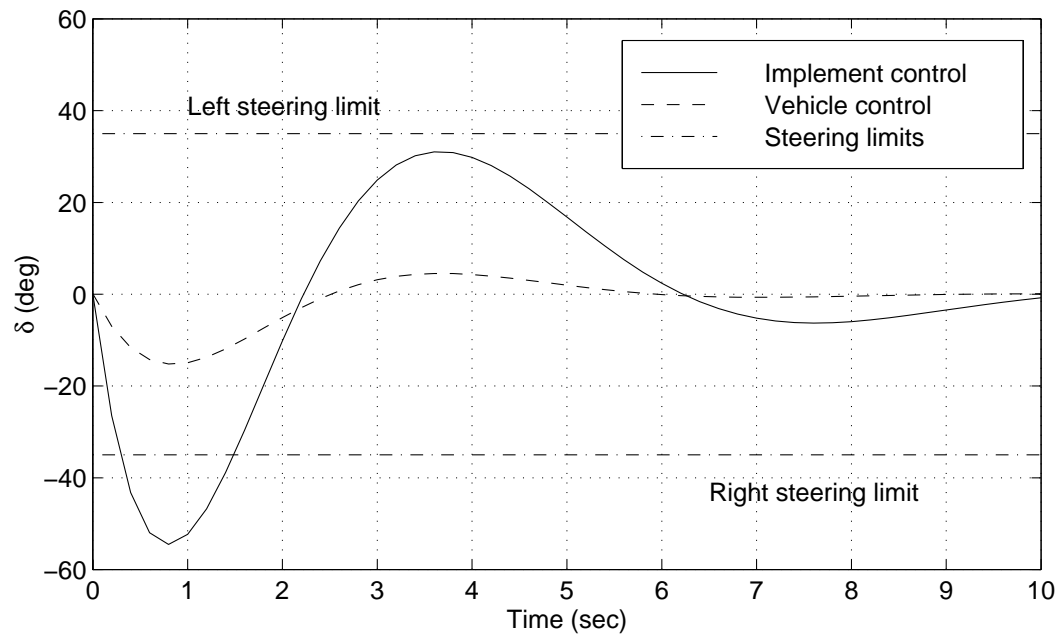


Figure 4.4: Steering response for both the vehicle (dashed line) and vehicle/implement (solid line) for the same initial conditions.

that the implement will travel the path of the tractor a prescribed distance later. Therefore, if the roll and pitch angles of the tractor were temporarily stored, they could be later reused as “pseudo-measurements” for the implement but with higher measurement noise to account for the higher uncertainty. Since the implement pivot point is a distance $l_3 - l_2$ behind the vehicle’s rear axle, the past $(l_3 - l_2)/V_x$ seconds worth of roll and pitch information would have to be stored. Vehicle yaw information could similarly be stored and later retrieved, but the yaw dynamics of the implement would have to be accounted for. Of course, this second problem of estimating the implement’s attitude disappears with an implement attitude sensor.

These problems associated with towed implement control mean that developing a towed implement controller that can deliver the same level of tracking performance as a vehicle controller would be very difficult without altering the vehicle/implement configuration. Nonetheless, the accuracy such an implement control system could deliver still may be good enough to open up new farming techniques previously believed impractical because the human driver could not control the implement accurately enough.

4.2 Disturbance Model

As was the case in the vehicle model, towed implement disturbances were found to increase with velocity. However, the smoothness of the soil was observed to have the largest impact on implement disturbance variances. Without a real-time estimate of the disturbance variances, only nominal values can be estimated. A realistic model for implement disturbances therefore was

$$\frac{\dot{V}_c^t}{t} = \frac{\dot{V}_a^t}{t} + \frac{\dot{\omega}_i}{t} \times \frac{r_3}{t} + \frac{\omega_i}{t} \times \frac{\omega_i}{t} \times \frac{r_3}{t} + \frac{V_{a,x}^t}{t} \nu_{V^t} \quad (4.42)$$

$$\dot{\Theta}^t = T_{\Theta^t} \frac{\omega_i}{t} + \frac{V_{a,x}^t}{t} \nu_{\Theta^t} \quad (4.43)$$

Estimates of the disturbance variances are given in Tbl. 4.1. Compared to the vehicle disturbance variances, the implement disturbance variances are the same in forward speed because the tractor and the implement are linked together and are usually

| Variable | Standard Dev (one- σ) | Name |
|------------------|----------------------------------|---------------------|
| $\nu_{V_x^t}$ | 0.05 s ⁻¹ | <i>Velocity</i> |
| $\nu_{V_y^t}$ | 0.03 s ⁻¹ | <i>disturbances</i> |
| $\nu_{V_z^t}$ | 0.01 s ⁻¹ | |
| ν_{ϕ^t} | 0.5 deg-m ⁻¹ | <i>Angular</i> |
| ν_{θ^t} | 0.5 deg-m ⁻¹ | <i>disturbances</i> |
| ν_{ψ^t} | 1.0 deg-m ⁻¹ | |

Table 4.1: Estimated disturbance variances for the implement model.

nearly lined up. The lateral velocity variance is significantly higher than for the vehicle because the towed implement is lighter and is subject to greater disturbances. For the same reason, the variances in vertical velocity and attitude are also slightly higher than the corresponding variances for the vehicle.

CHAPTER 5

Nonlinear Real-Time Estimation

As was shown in Chs. 3 and 4, the vehicle model was nonlinear in states such as velocity and the steering constant because they involved products and transcendental functions. These states had to be known when developing a locally linear model for a linear controller. These states were not measured and thus had to be estimated in real-time. One possible solution was to break the estimation problem up into pieces: a locally linear model could be used to create a linear filter, and nonlinear states such as velocity could be filtered by “measuring” the velocity through finite-differenced position data. This approach, though used in previous research on flat terrain, had several shortfalls:

- Lateral velocity from side-slip could not be estimated, and a new yaw-bias state had to be created.
- Because the filter was linearized about the local trajectory, the filter had to be restarted at the beginning of each trajectory. This meant that valuable bias information was lost at the end of the previous trajectory. Some states, such as K_δ , required a significant amount of time to form a reasonably accurate estimate. Also, certain important states such as K_δ and lateral velocity did not depend on whether the driver or the computer was controlling the vehicle, yet the linear filter did not use the information available when the driver had control to estimate those states.

- Measuring the control point’s position required either a measurement or estimate of the vehicle’s attitude: the lever arm vector needed to “move” the position measurement from the GPS antenna to the control point had to be transformed from the vehicle frame to the inertial frame. If the attitude sensor experienced intermittent outages or sampled at a lower rate than the position sensor, position measurements could not be completed with a linear filter’s state estimate since the attitude dynamics were nonlinear.

A second possible solution was a nonlinear filter which would run constantly, regardless of the nature of vehicle control. At each point along the trajectory, the filter would build a linearized controllable model based on the state estimate and then hand that model off to the control algorithm. The net result would be more accurate trajectory tracking through improved modeling. One disadvantage of a nonlinear filter was that the higher number of states meant an increased computational cost. However, computational cost was only a concern if the computer was unable to finish the required calculations by the end of the sample period. Any computational task that finished in less time than the sample interval length was considered acceptable. The second issue with a nonlinear filter was divergence. Poor modeling could lead to state estimate divergence. This threat was minimized with what was believed to be a reasonably accurate vehicle model. Therefore, because of a nonlinear filter’s advantages, this type of algorithm was sought to estimate the vehicle state in real-time. The assumption that model disturbances were zero-mean with Gaussian distributions was believed to be reasonable, so extended Kalman filters of first and second-order were examined as possible real-time filtering algorithms. This chapter explains the measurements available during the filtering process and details some of the measurement issues specific to the experimental platforms used in this research. This chapter then identifies two possible filtering algorithms and compares their performance in simulation. Finally, experimental results are presented for real-time filtering.

Extended Kalman filters have been used for real-time estimation of system parameters in a wide variety of systems. In [GS93], researchers estimated the time-varying inertial parameters of a nonlinear robot model in simulation. In [A⁺97], the authors estimated a wheeled mobile robot’s state, sensor biases, and slip characteristics. In

[SC87], a variant of the extended Kalman filter, the modified-gain extended Kalman filter, was used in simulation to estimate the stability derivatives of an aircraft similar to the F-111. Researchers in [BDW95] demonstrated that the extended Kalman filter could be used in inertially-guided mobile robot applications to accurately estimate the position, orientation, and sensor biases of a mobile robot.

5.1 Measurements

In experiment, position measurements were available at 5 Hz, attitude measurements were available at 10 Hz and steering measurements at 20 Hz. To take advantage of the higher measurement rates, the filter could be run at 20 Hz. At every measurement epoch (50 msec), the latest measurements could be used to update the filter. By incorporating the extra attitude measurement and the three extra steering measurements available between position measurements, state estimate accuracy could be improved.

The vehicle measurement vector was

$$y = \begin{bmatrix} \frac{r_c^v}{i} \\ \Theta^v \\ \delta_m \end{bmatrix} = \begin{bmatrix} \frac{r_m}{i} + T_{\frac{i}{v}} \frac{r_\rho}{v} \\ \Theta^v \\ \delta + \delta_b \end{bmatrix} \quad (5.1)$$

where $\frac{r_m}{i}$ was the measured position of the vehicle's master GPS antenna and $\frac{r_\rho}{v}$ was a "lever arm" vector from the master GPS antenna to the vehicle control point. Because the transformation matrix from the vehicle frame to the inertial frame $T_{\frac{i}{v}}$ was a nonlinear function of vehicle attitude Θ^v , the measurement update was nonlinear. The attitude measurement was very accurate, and the raw attitude measurement could have been used in the transformation matrix to make the measurement update linear. Although this approach worked well in experiment, it relied on having an attitude measurement available every time a position measurement was processed. This approach also assumed little error was introduced by attitude measurement noise. Because of the constraints imposed by these assumptions, the measurement

update remained nonlinear to create a filtering system robust enough to withstand intermittent attitude outages¹ and degradations in attitude measurement accuracy.

5.1.1 Position Measurement

Position measurements were available every 200 msec. Position was measured at the master GPS antenna (one of the four antennas mounted to the vehicle roof). The GPS positioning software output a position measurement and a measurement covariance matrix that expressed the uncertainty in the position measurement. Two important issues were encountered when measuring the vehicle's position: (1) additional uncertainty introduced by the noisy lever arm correction, and (2) antenna movement relative to the vehicle frame through cab sway or antenna vibration.

As will be shown in Ch. 6, an approximation for the additional uncertainty introduced by the lever arm correction can be calculated at every measurement epoch and added to the position measurement covariance matrix at the antenna to create a position uncertainty matrix at the control point. The additional uncertainty approximation typically added 4 mm (one-sigma) to the position uncertainty at the control point as long as the attitude measurements were of nominal accuracy.

The cab of both test platforms was only semi-rigidly mounted to the tractors' frames, and cab sway could have been a significant source of measurement error. At each bottom corner of the cab, large rubber mounts held the cab to the frame and insulated the cab from engine and transmission noise. See Fig. (5.1). Although the cab was observed even in very dynamic environments to be virtually rigid relative to the tractor frame, a rough test was devised to measure cab movement: modeling clay was wedged between the cab and the frame, the tractor driven over a rough field at high speed, and the clay compression measured. Although the measurement was rough (approximate tolerance 1 mm), there was no noticeable deformation. Therefore, cab sway was considered negligible.²

Static position data collected over a ten-minute period showed that high-frequency

¹Or possibly a slower attitude sample rate than position sample rate.

²Note also that the rack mounting the GPS antennas to the roof of the cab was very stiff and was therefore assumed to be rigid.



Figure 5.1: Photo of rubber mounts between tractor cab and frame. The camera is at the rear of the tractor looking forward. The inside of the left rear wheel can be seen on the left.

position measurement variations were roughly 30% of the measurement uncertainty predicted by the GPS positioning software (see Tbl. 5.1). The discrepancy stemmed mainly from (slowly varying) uncertainty in the CPDGPS cycle ambiguities. Because the additional uncertainty varied much more slowly compared to the control bandwidth, the additional uncertainty could be considered negligible, and the GPS position measurement noise was considered white.

| Axis | Std. Dev. (one- σ) | |
|--------------|------------------------------------|----------------------------------|
| | Measured High-frequency Variations | Measurement Uncertainty Estimate |
| <i>North</i> | 0.8 cm | 3.0 cm |
| <i>East</i> | 0.8 cm | 2.9 cm |

Table 5.1: Static horizontal precision of GPS position measurements. The left column shows the high-frequency variations for static data collected over a ten-minute period. The right column is the GPS positioning software's *estimate* of position measurement uncertainty.

| Variable | one- σ |
|----------|---------------|
| ϕ | 0.05° |
| θ | 0.08° |
| ψ | 0.06° |

Table 5.2: High-frequency variations in GPS attitude measurements. Static data was collected over a ten-minute period.

5.1.2 GPS Attitude Measurement

Attitude measurements, available at 10 Hz, contained little noise. Analysis of static data showed the measurement standard deviations were all less than 0.1° as summarized in Tbl. 5.2. Attitude measurement accuracy also could suffer from cab motion and movement of one antenna relative to another. As previously discussed, both factors were considered negligible.

5.1.3 Steer Angle Measurement

The front wheel angle was measured at 20 Hz by a linear potentiometer. Measurements ranged between -0.7 and 0.7 rad. Although the eight-bit steer angle data packet received at the computer had values that ranged between 0 and 255, the potentiometer mounting limited the range to 6–180. In addition, measurement resolution varied from left to right. A look-up table was used to translate the steer angle byte to a steering angle in the range of ± 0.7 rad. The table was created by allowing the computer to hold the front wheels at a specific potentiometer reading while recording the yaw angle and forward speed of the tractor under dry relatively skid-free conditions. The relationship $\Delta\psi/T_s = (V_x/l_1) \tan \delta$ was then used to obtain a “best-fit” value of δ for that particular potentiometer reading. In Fig. (5.2), yaw rate is plotted against the potentiometer reading for a forward velocity of 1.0 m/s and a wheelbase of 2.80 m. Note that this calibration procedure implicitly assumed a nominal K_δ value of 1.0. The look-up table also gave the “width” or “bin-size” of the steering angle for each byte value. This “width” was used to establish a measurement variance for a

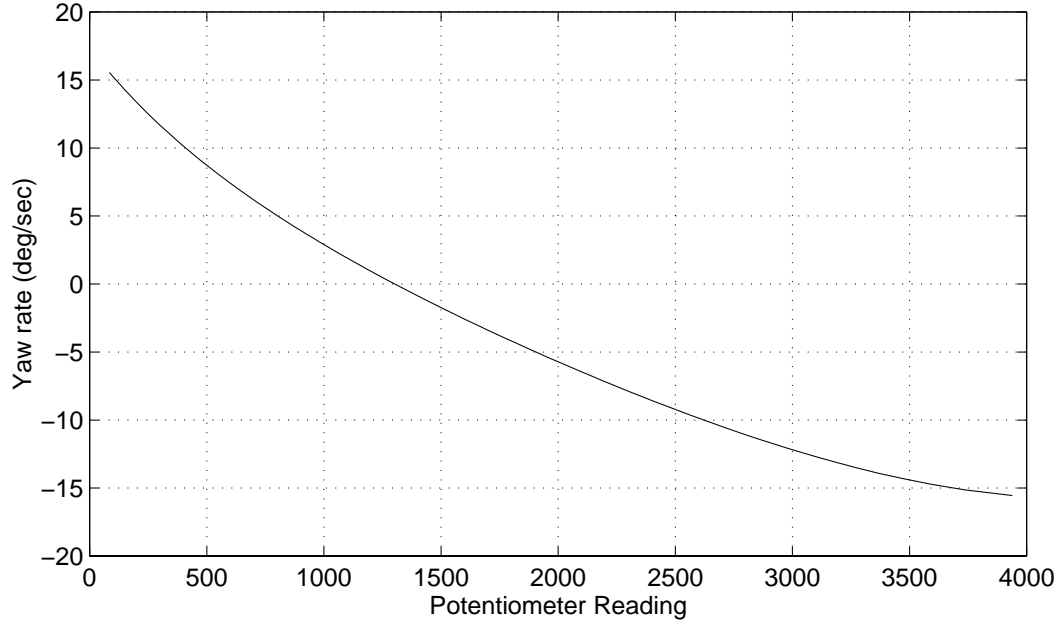


Figure 5.2: Yaw rate as a function of the steering potentiometer reading.

particular byte value b_i [F⁺90]:

$$R_v(b_i) = \frac{(b_{i+1} - b_i)^2}{12} \quad (5.2)$$

Since the measurement variance changed as a function of the measured angle as shown in Fig. (5.3), another look-up table could have been generated to obtain the steer angle measurement variance for a given steer angle. However, as seen in Fig. (5.3), the measurement variance changed smoothly, and the third-order polynomial

$$\begin{aligned} R_v(\delta) = & -9.5014 \times 10^{-6} \delta^3 + 1.3853 \times 10^{-5} \delta^2 \\ & - 1.0436 \times 10^{-5} \delta + 5.5109 \times 10^{-6} \end{aligned} \quad (5.3)$$

provided a close approximation. At every steer angle measurement, the measurement variance was calculated using Eqn. (5.3) and was then incorporated into the estimator.

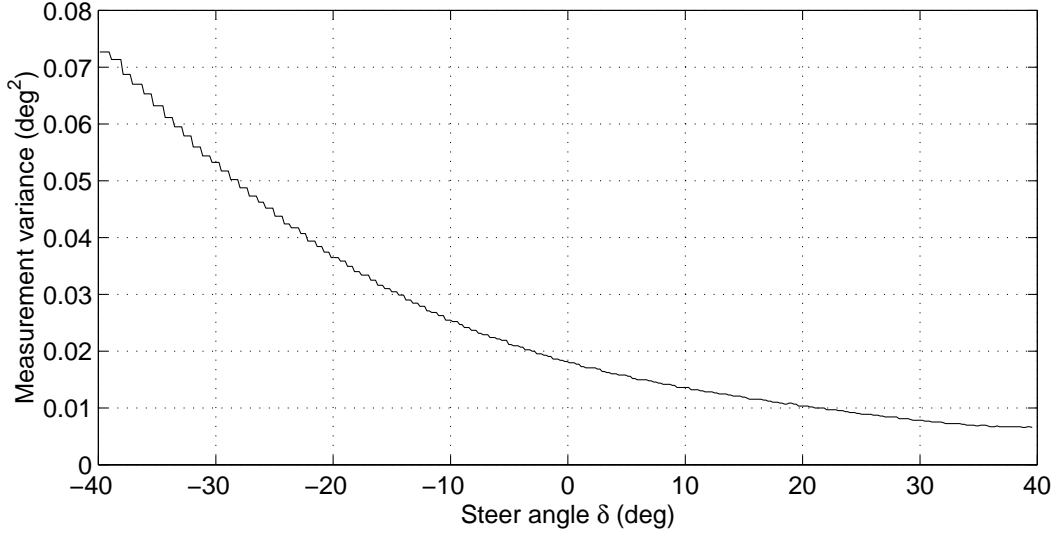


Figure 5.3: Steer angle measurement variance as a function of steering angle.

5.2 The Kalman Filter

Previous research [O’C97] used the well-known Kalman filter linearized along the reference trajectory to filter the measurements and estimate the vehicle state. The time update was [Ste86]

$$\hat{x}_{k+1} = \Phi_k \hat{x}_k + \Gamma_k u_k \quad (5.4)$$

$$P_{k+1} = \Phi_k P_k \Phi_k^T + Q_d \quad (5.5)$$

where \hat{x}_k was the estimated state, Φ_k was the discretized state transition matrix, Γ_k was the discretized control input matrix for the control signal u_k , P_k was the estimator’s covariance matrix, and Q_d was the discretized process noise covariance matrix [vL78]. If the measurement vector at epoch k was given by

$$y_k = h(x_k) \quad (5.6)$$

then the measurement update is

$$L_c = P_k H_k^T (H_k P_k H_k^T + R_v)^{-1} \quad (5.7)$$

$$\hat{x}_k = \hat{x}_k + L_c (y_k - \hat{y}_k) \quad (5.8)$$

$$P_k = (I - L_c H_k) P_k \quad (5.9)$$

where H_k is the observation matrix (*i.e.*, the Jacobian of the observation vector) and R_v is the measurement noise covariance matrix.

Although a discrete square root filter [K⁺71] would have offered improved numerical stability over the standard filter implementation detailed above, square root filtering techniques use a linear time update, and the equations of motion derived above were nonlinear. A first order approximation to the equations of motion could have been used to linearize the system's time update. However, concerns over loss of accuracy in approximating the non-linear dynamics as linear and the good performance of the non-linear filter in experiment precluded a square root filter implementation.

5.3 The Extended Kalman Filter

Since the equations of motion and measurements are nonlinear, a nonlinear filter was used to estimate the vehicle state.

5.3.1 First-Order Filter

The time update for the first-order filter involved integrating the equations of motion outlined in Ch. 3. The fourth-order Runge-Kutta integration scheme was used for the time update [Kre93]. The same integration scheme could also have been used to time-update the covariance matrix using the differential equation for the covariance matrix [Kas95]

$$\dot{P} = \left(\frac{\partial f(\hat{x})}{\partial x} \right) P + P \left(\frac{\partial f(\hat{x})}{\partial x} \right)^T + G Q G^T \quad (5.10)$$

$$\dot{x} = f(x) \quad (5.11)$$

However, simulation showed that forming

$$\Phi_k = e^{T_s \left(\frac{\partial f(x)}{\partial x} \right)} \quad (5.12)$$

and updating the covariance matrix using Eqn. (5.5) incurred little performance penalty because the dynamics of the vehicle changed little within one discretization interval.

The measurement update was linearized using the Jacobian of the measurement vector [BH75]:

$$L_c = P_k^- \left(\frac{\partial h(\hat{x}_k^-)}{\partial x} \right)^T \left[\left(\frac{\partial h(\hat{x}_k^-)}{\partial x} \right) P_k^- \left(\frac{\partial h(\hat{x}_k^-)}{\partial x} \right)^T + R_v \right]^{-1} \quad (5.13)$$

$$\hat{x}_k^+ = \hat{x}_k^- + L_c [y_k - h(\hat{x}_k^-)] \quad (5.14)$$

$$P_k^+ = \left[I - L_c \left(\frac{\partial h(\hat{x}_k^-)}{\partial x} \right) \right] P_k^- \quad (5.15)$$

5.3.2 Second-Order Filter

Higher-order nonlinear filters can be generated by retaining higher-order terms in the time and measurement updates. Although a variety of higher-order filters exist [May82], the modified Gaussian second-order filter was chosen as a possible alternative to the first-order extended Kalman filter because the modified Gaussian filter's assumptions were considered more realistic: the filter assumed that the conditional density was nearly Gaussian, leading to the assumptions that third and higher-order odd moments were zero, and higher-order even moments could be expressed in terms of the covariance [May82]. The differential equations for the modified Gaussian second-order filter's time update are [May82]

$$\dot{\hat{x}} = f(\hat{x}) + \hat{b}_p \quad (5.16)$$

$$\hat{b}_p(i) = \frac{1}{2} \text{tr} \left[\left(\frac{\partial^2 f_i(\hat{x}_k^-)}{\partial x^2} \right) P \right] \quad (5.17)$$

$$\dot{P} = \left(\frac{\partial f(\hat{x})}{\partial x} \right) P + P \left(\frac{\partial f(\hat{x})}{\partial x} \right)^T + G \widehat{QG^T} \quad (5.18)$$

$$\left[\widehat{GQG^T} \right] (i, j) = \sum_{l=1}^s \left[\begin{aligned} & G_{il} G'_{lj}{}^T + \text{tr} \left\{ \left(\frac{\partial G'_{il}}{\partial x} \frac{\partial G'_{lj}{}^T}{\partial x} \right) P \right\} \\ & + \frac{1}{2} G'_{il} \text{tr} \left\{ \frac{\partial^2 G'_{lj}{}^T}{\partial x^2} P \right\} \\ & + \frac{1}{2} \text{tr} \left\{ P \frac{\partial^2 G'_{il}}{\partial x^2} \right\} G'_{lj}{}^T \\ & + \frac{1}{4} \text{tr} \left\{ \frac{\partial^2 G'_{il}}{\partial x^2} P \right\} \text{tr} \left\{ \frac{\partial^2 G'_{lj}{}^T}{\partial x^2} P \right\} \\ & + \frac{1}{2} \text{tr} \left\{ \frac{\partial^2 G'_{il}}{\partial x^2} P \frac{\partial^2 G'_{lj}{}^T}{\partial x^2} P \right\} \end{aligned} \right] \quad (5.19)$$

where $G \in \Re^{n \times s}$ and

$$G' \equiv GQ^{1/2} \quad (5.20)$$

The measurement update equations are

$$L_c = P_k^- \left(\frac{\partial h(\hat{x}_k^-)}{\partial x} \right)^T \left[\left(\frac{\partial h(\hat{x}_k^-)}{\partial x} \right) P_k^- \left(\frac{\partial h(\hat{x}_k^-)}{\partial x} \right)^T + \hat{B}_m + R_v \right]^{-1} \quad (5.21)$$

$$\hat{x}_k^+ = \hat{x}_k^- + L_c \left[y_k - h(\hat{x}_k^-) - \hat{b}_m \right] \quad (5.22)$$

$$P_k^+ = \left[I - L_c \left(\frac{\partial h(\hat{x}_k^-)}{\partial x} \right) \right] P_k^- \quad (5.23)$$

$$\hat{B}_m(i, j) = \frac{1}{2} \text{tr} \left[\left(\frac{\partial^2 h_i(\hat{x}_k^-)}{\partial x^2} \right) P_k^- \left(\frac{\partial^2 h_j(\hat{x}_k^-)}{\partial x^2} \right) P_k^- \right] \quad (5.24)$$

$$\hat{b}_m(i) = \frac{1}{2} \text{tr} \left[\left(\frac{\partial^2 h_i(\hat{x}_k^-)}{\partial x^2} \right) P_k^- \right] \quad (5.25)$$

5.4 Simulation Results

The first- and second-order filters were compared in a noise- and disturbance-free simulation to highlight performance differences. A vehicle model was simulated with a randomly varying control input for ten minutes. The two filters were run at 20 Hz. Differences in estimated position, velocity, attitude, and steer angle were insignificant. The most noticeable differences in performance was in the estimates of the steering proportionality constant and steering bias. Fig. (5.4) shows the two filters' estimates of K_δ (dashed line and dash-dot line) converging from their initial guesses of 0.8 to

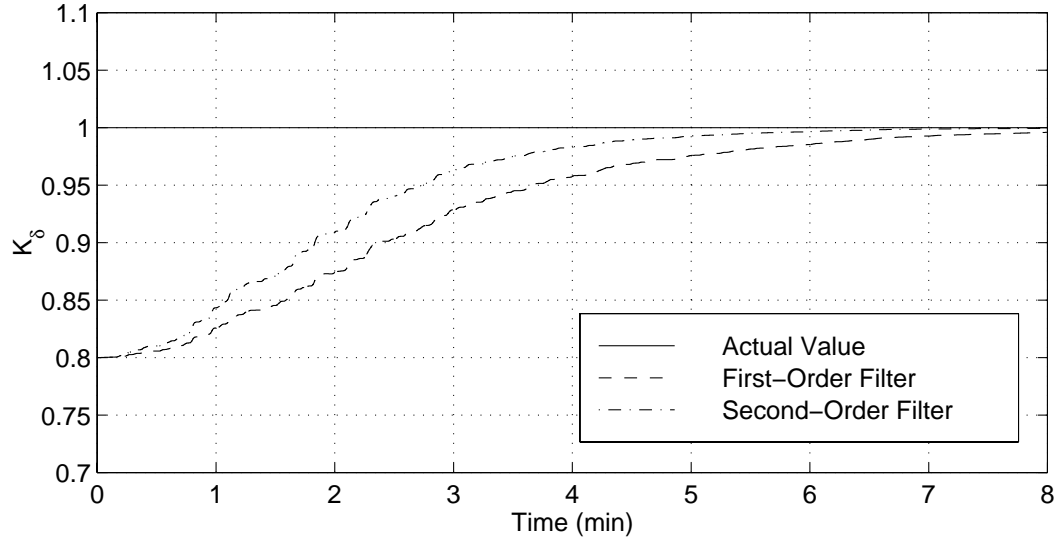


Figure 5.4: Comparison of first- and second-order filtering algorithms for estimating K_δ .

the actual value of 1.0 (solid line). Fig. (5.5) shows the filters' estimates of δ_b (dashed line and dash-dot line, respectively) converging from their initial guesses of 0.0° to the actual value of -2.0° (solid line). These results demonstrate that the performance difference between the first- and second-order filters was relatively small. Both filters effectively estimated the two most important unmeasured states, K_δ and δ_b , in a reasonable amount of time. Although the second-order filter converged more rapidly, the computational cost prevented real-time implementation.

5.5 Experimental Results

In experiment, the first-order filter could not be run faster than 5 Hz because of the computational load.³ The Model 8400 tractor was used as a test platform. The tractor was controlled along a row using a linear (LQR) control law based on the simple linearized model in Eqns. (3.43–3.45). No implement was used. The outside

³The real-time filter also shared the computer's processing power with the GPS positioning and attitude software.

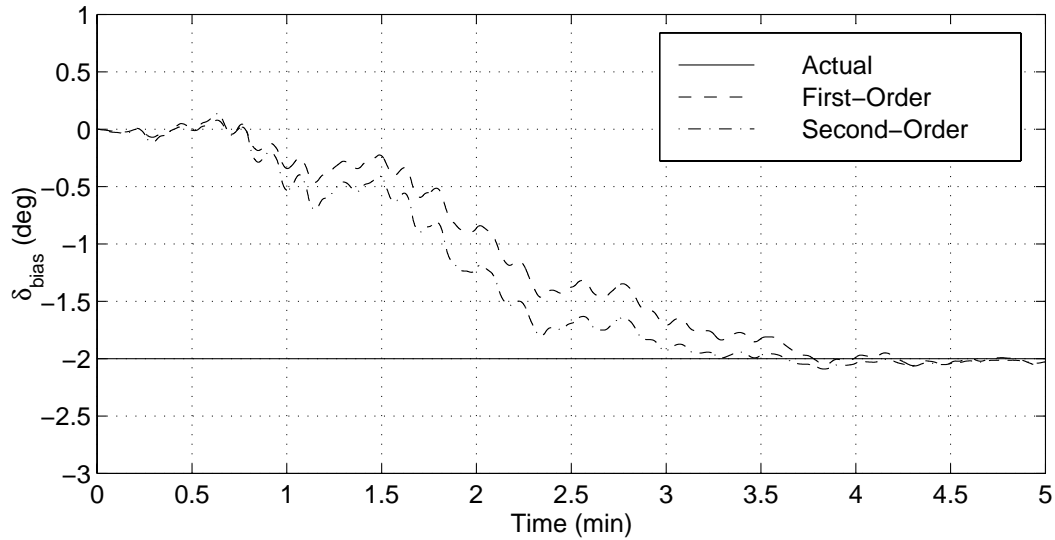


Figure 5.5: Comparison of first- and second-order filtering algorithms for estimating δ_b .

set of dual rear wheels had been removed for shipment, and thus ground disturbances had a pronounced effect on the tractor's roll angle.

The filter's estimate(s) of K_δ are used to illustrate the filter's performance for two reasons. First, as previously mentioned, one of the most important benefits of using a nonlinear filter is the filter's ability to estimate the key parameter K_δ . Second, the potentiometer used to measure the front-wheel angle was calibrated using the procedure outline in Sec. 5.1.3 with the second set of dual rear wheels removed. Therefore, the actual value of K_δ was known to be very close to 1.0.⁴ Fig. (5.6) shows how the filter's estimate of K_δ routinely converged in eleven experiments from initial guesses of either 0.5, 1.0, or 1.5 to the correct value of 1.0. Note that an initial guess beneath the actual value could have destabilized the estimator/controller since the controller would have believed a larger steer angle than necessary was required to turn the tractor. Note also that the process noise variance for K_δ was turned up artificially high to speed up convergence on this short test field. Despite these two

⁴The tractor configuration was identical to its configuration during the steering calibration tests. Soil conditions were also approximately equal.

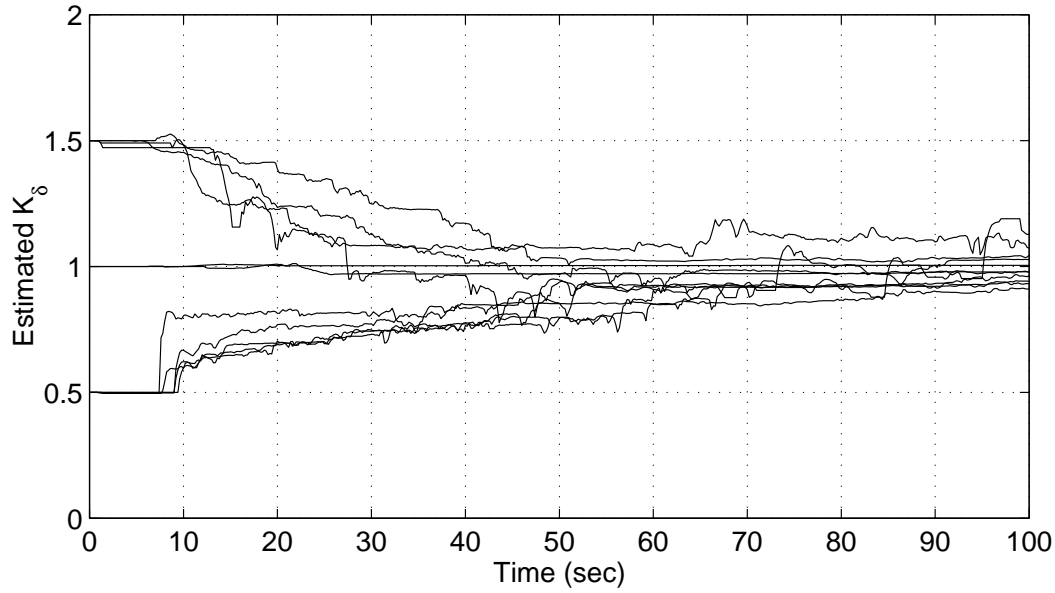


Figure 5.6: Filter estimates of K_δ for eleven experimental runs.

destabilizing factors, convergence to the actual K_δ value was reliable.

In Fig. (5.7), the estimated tracking error once the tractor had acquired the row is shown for a typical experiment. The mean tracking error was 0.86 cm and the standard deviation 2.84 cm at a nominal forward speed of 1.1 m/s. The significant effects of the ground disturbances can be seen at several points along the row.

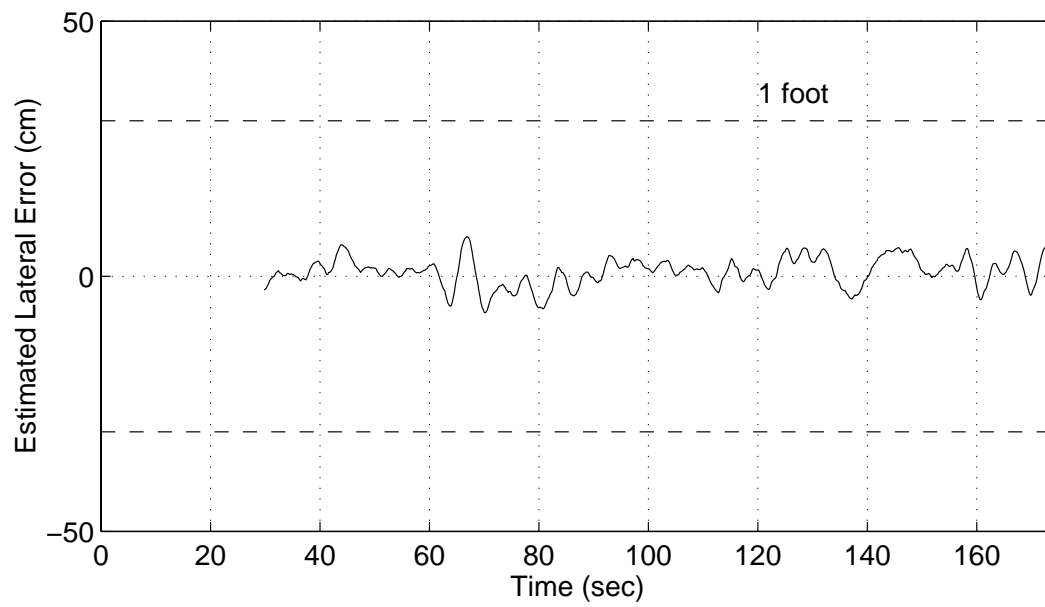


Figure 5.7: Extended Kalman filter's tracking error estimate for control along a straight row. Mean tracking error was 0.86 cm, and the standard deviation was 2.84 cm (one-sigma).

CHAPTER 6

The Lever Arm Correction

To accurately filter the control point position, an estimate of the position measurement uncertainty was required. Although the position software provided an estimate of the position measurement covariance, that covariance was valid at the measurement point, not the control point. The point on the vehicle to be controlled was not collocated with the GPS position antenna, and the position measurement at the antenna had to be moved or corrected through the addition of a lever arm vector from the antenna to the control point (see Eqn. 5.1). This lever arm vector was a nonlinear function of uncertain attitude and lever arm measurements. Therefore, the noise-corrupted lever arm correction introduced additional uncertainty into the control point position measurement. In this chapter, error analysis was used to show that control point position measurement uncertainty can be expressed as the sum of the uncertainty of the original measurement at the antenna and an additional uncertainty stemming from the lever arm correction. Error analysis was then used to derive an approximation for that additional uncertainty. The approximation was verified through a Monte Carlo simulation and was then used to show that, above a certain sensor noise level, attitude uncertainty could become the dominant source of position measurement uncertainty at the vehicle control point.

In [Jun97], Junkins used the first two statistical moments of an orbiting body to show how its positional uncertainty can evolve over time. He showed that although a first-order expression could approximate the evolving uncertainty, significant nonlinearities in the equations of motion meant that the approximation's fidelity depended on the coordinate system used. Junkins' work focused on propagating the uncertainty

forward in time through the nonlinear equations of motion; this research focused instead on propagating the uncertainty through nonlinear measurements.

6.1 Approximating the Additional Uncertainty

The GPS antennas were mounted on the roof of the vehicle because they required an unrestricted view of the GPS satellites overhead. The lever arm vector from the master GPS antenna to the vehicle control point was added to the position measurement vector:

$$\frac{r_c^v}{i} = \frac{r_m}{i} + T_{\frac{i}{v}} \frac{r_\rho}{v} \quad (6.1)$$

where $T_{\frac{i}{v}}$ is the nonlinear transformation matrix from the vehicle coordinate frame to the inertial coordinate frame (see Eqn. 3.21).

The uncertainty in the control point position measurement can be expressed as

$$\underline{\underline{\Sigma}}_c \equiv E \left[\left(\frac{r_c^v}{i} - \bar{r}_c^v \right) \left(\frac{r_c^v}{i} - \bar{r}_c^v \right)^T \middle| \Theta^v, \frac{r_m}{i}, \frac{r_\rho}{v} \right] \quad (6.2)$$

where Θ^v is the vehicle's attitude vector, $\underline{\underline{\Sigma}}_c$ is the variance/covariance matrix (coordinated in the inertial frame) of the position measurement at the control point, \bar{r}_c^v is the mean, or expected, value of the vehicle control point position, and E denotes

taking the expected value. Substituting Eqn. (6.1) into (6.2),

$$\underline{\underline{\Sigma}}_i = E \left[\left(\begin{array}{c} \frac{r_m}{i} - \frac{\bar{r}_m}{i} \\ + T_{\frac{i}{v}} r_{\rho} - \bar{T}_{\frac{i}{v}} \bar{r}_{\rho} \end{array} \right) \left(\begin{array}{c} \frac{r_m}{i} - \frac{\bar{r}_m}{i} \\ + T_{\frac{i}{v}} r_{\rho} - \bar{T}_{\frac{i}{v}} \bar{r}_{\rho} \end{array} \right)^T \right] \Theta^v, \frac{r_m}{i}, \frac{r_{\rho}^v}{v} \quad (6.3)$$

$$= E \left[\begin{array}{c} \left(\frac{r_m}{i} - \frac{\bar{r}_m}{i} \right) \left(\frac{r_m}{i} - \frac{\bar{r}_m}{i} \right)^T \\ + \left(\frac{r_m}{i} - \frac{\bar{r}_m}{i} \right) \left(T_{\frac{i}{v}} r_{\rho} - \bar{T}_{\frac{i}{v}} \bar{r}_{\rho} \right)^T \\ + \left(T_{\frac{i}{v}} r_{\rho} - \bar{T}_{\frac{i}{v}} \bar{r}_{\rho} \right) \left(\frac{r_m}{i} - \frac{\bar{r}_m}{i} \right)^T \\ + \left(T_{\frac{i}{v}} r_{\rho} - \bar{T}_{\frac{i}{v}} \bar{r}_{\rho} \right) \left(T_{\frac{i}{v}} r_{\rho} - \bar{T}_{\frac{i}{v}} \bar{r}_{\rho} \right)^T \end{array} \right] \Theta^v, \frac{r_m}{i}, \frac{r_{\rho}^v}{v} \quad (6.4)$$

Expressions involving

$$T_{\frac{i}{v}} r_{\rho} - \bar{T}_{\frac{i}{v}} \bar{r}_{\rho}$$

are nonlinear in attitude and lever arm measurements and can not be evaluated directly in terms of the known statistical properties of the attitude and lever arm measurements. A multivariate Taylor series expansion [Hil76] can approximate the transformed lever arm by expanding $T_{\frac{i}{v}} r_{\rho}$ about $\bar{T}_{\frac{i}{v}} \bar{r}_{\rho}$ [Tay82]:

$$T_{\frac{i}{v}} r_{\rho} \cong \bar{T}_{\frac{i}{v}} \bar{r}_{\rho} + \nabla \left(T_{\frac{i}{v}} r_{\rho} \right) \Big|_{\bar{x}} (x - \bar{x}) + \dots \quad (6.5)$$

$$x \equiv \begin{bmatrix} \Theta^v \\ \frac{r_{\rho}}{v} \end{bmatrix} \quad (6.6)$$

where the gradient operator ∇ is taken with respect to attitude and the lever arm. This is because unmodeled high-frequency vibration of the cab relative to the tractor frame and vibration of the antenna mounts could be approximated as lever arm measurement noise. If Eqn. (6.5) is rewritten as

$$T_{\frac{i}{v}} r_{\rho} - \bar{T}_{\frac{i}{v}} \bar{r}_{\rho} \cong \nabla \left(T_{\frac{i}{v}} r_{\rho} \right) \Big|_{\bar{x}} (x - \bar{x}) \quad (6.7)$$

then Eqn. (6.4) becomes

$$E \left[\begin{array}{c} \left(\frac{r_m}{i} - \bar{r}_m \right) \left(\frac{r_m}{i} - \bar{r}_m \right)^T + \left(\frac{r_m}{i} - \bar{r}_m \right) (x - \bar{x})^T \nabla \left(T_{\frac{i}{v}} \frac{r_\rho}{v} \right)^T \bigg|_{\bar{x}} \\ + \nabla \left(T_{\frac{i}{v}} \frac{r_\rho}{v} \right) \bigg|_{\bar{x}} (x - \bar{x}) \left(\frac{r_m}{i} - \bar{r}_m \right)^T \\ + \nabla \left(T_{\frac{i}{v}} \frac{r_\rho}{v} \right) \bigg|_{\bar{x}} (x - \bar{x}) (x - \bar{x})^T \nabla \left(T_{\frac{i}{v}} \frac{r_\rho}{v} \right)^T \bigg|_{\bar{x}} \end{array} \middle| \bar{x}, \frac{r_m}{i} \right] \quad (6.8)$$

If the effects of satellite geometry on both attitude and position measurement noises are ignored,¹ the measurement noises in attitude, lever arm, and position are uncorrelated. The mean measurement error in both attitude and position is assumed to be zero, and the expected values of

$$\begin{aligned} & \left(\frac{r_m}{i} - \bar{r}_m \right) (x - \bar{x})^T \nabla \left(T_{\frac{i}{v}} \frac{r_\rho}{v} \right)^T \\ & \nabla \left(T_{\frac{i}{v}} \frac{r_\rho}{v} \right) (x - \bar{x}) \left(\frac{r_m}{i} - \bar{r}_m \right)^T \end{aligned}$$

are therefore zero. The expected value of

$$\left(\frac{r_m}{i} - \bar{r}_m \right) \left(\frac{r_m}{i} - \bar{r}_m \right)^T$$

is known from the position sensor and is denoted by $\underline{\underline{\sum}}_m$. Eqn. (6.8) reduces to

$$\underline{\underline{\sum}}_c = \underline{\underline{\sum}}_m + \left[\nabla \left(T_{\frac{i}{v}} \frac{r_\rho}{v} \right) \right] \underline{\underline{\sum}}_{\Theta^v, \rho} \left[\nabla \left(T_{\frac{i}{v}} \frac{r_\rho}{v} \right) \right]^T \quad (6.9)$$

Eqn. (6.9) shows how, to first order, uncertainty in the lever arm correction adds uncertainty to the original GPS position measurement. If the noise in attitude and

¹If these effects can not be ignored, then the expected values of cross-terms involving both position and attitude will be non-zero. Expressions for these cross-terms could be derived based on an error analysis of the algorithms used to generate position and attitude measurements using the GPS carrier-phase measurements.

lever arm are uncorrelated, $\underline{\underline{\Sigma}}_{\Theta^v, \rho}$ is block-diagonal:

$$\underline{\underline{\Sigma}}_{\Theta^v, \rho} = \begin{bmatrix} \underline{\underline{\Sigma}}_{\Theta^v} & 0 \\ 0 & \underline{\underline{\Sigma}}_{\rho} \end{bmatrix} \quad (6.10)$$

If the vehicle's attitude was measured and used to transform the lever arm, then $\underline{\underline{\Sigma}}_{\Theta^v}$ is the attitude measurement noise. If an attitude measurement is not available when the position measurement is taken, an estimate of the vehicle's attitude from a nonlinear filter could be used. In this case, $\underline{\underline{\Sigma}}_{\Theta^v}$ could be extracted from the filter's inertia matrix. $\underline{\underline{\Sigma}}_{\rho}$ could be estimated from a model of the vehicle/cab dynamics.

6.2 Verifying the Additional Uncertainty Approximation

A Monte Carlo simulation was conducted to show that Eqn. (6.9) represented a reasonable approximation to the additional uncertainty introduced by the lever arm correction. For a medium-sized farm tractor, a reasonable lever arm in forward-right-down coordinates from a control point on the ground directly beneath the center of the rear axle to a GPS antenna mounted on the upper right side of the cab is (in meters)

$$\frac{r_{\rho}}{v} \equiv \begin{bmatrix} 0.5 \\ 1.0 \\ -3.3 \end{bmatrix} \quad (6.11)$$

The lever arm was assumed to have zero measurement noise.² A randomly chosen measured attitude of

$$\begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix} = \begin{bmatrix} -1.1^\circ \\ 0.8^\circ \\ 283.0^\circ \end{bmatrix} \quad (6.12)$$

²Note that lever arm noise, which could come from high-frequency antenna vibration, would add uncertainty.

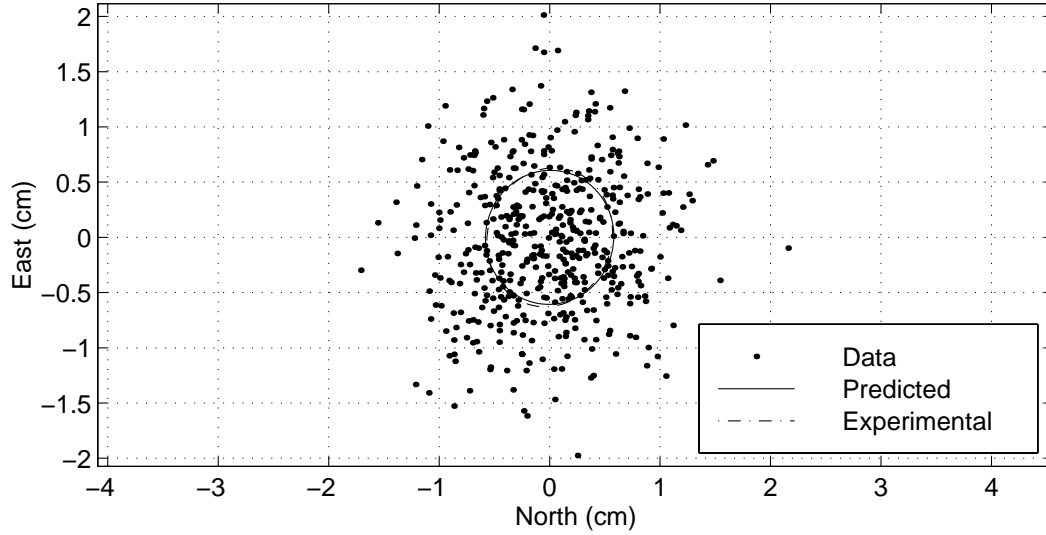


Figure 6.1: Projection of 500 noisy lever arm corrections onto the North-East plane for the assumed attitude of -1.1° in roll, 0.8° pitch, and 283.0° yaw. One-sigma ellipsoids (predicted as solid line, experimental as dashed) shown. Note that the approximation is good enough that the difference between the two ellipsoids is hard to distinguish.

| | σ_1 | σ_2 | σ_3 |
|---------------------|------------|------------|------------------------|
| <i>Predicted</i> | 0.371 | 0.368 | 1.10×10^{-17} |
| <i>Experimental</i> | 0.390 | 0.373 | 1.15×10^{-6} |

Table 6.1: Comparison of singular values for predicted and experimental covariance matrices.

was assumed with measurement noise variances of 0.1° in roll, pitch, and yaw. Five hundred noise-corrupted lever arm corrections were calculated using Eqn. (6.1). One-sigma ellipsoids of the resulting 500 corrected positions are plotted against the uncertainty ellipsoid predicted by Eqn. (6.9) in Figures (6.1–6.3). Note that the additional uncertainty primarily shows up in the North-East plane, the plane most crucial for precise control measurements. The small difference between the predicted and Monte Carlo uncertainty ellipsoids indicates that not only do the magnitudes of the singular values closely approximate each other, but the predicted directions of uncertainty are closely aligned as well. In Tbl. 6.1, the predicted versus experimental singular values are shown. The maximum predicted uncertainty in any direction (the square root of the covariance matrix's largest singular value) was 0.61 cm, while the maximum

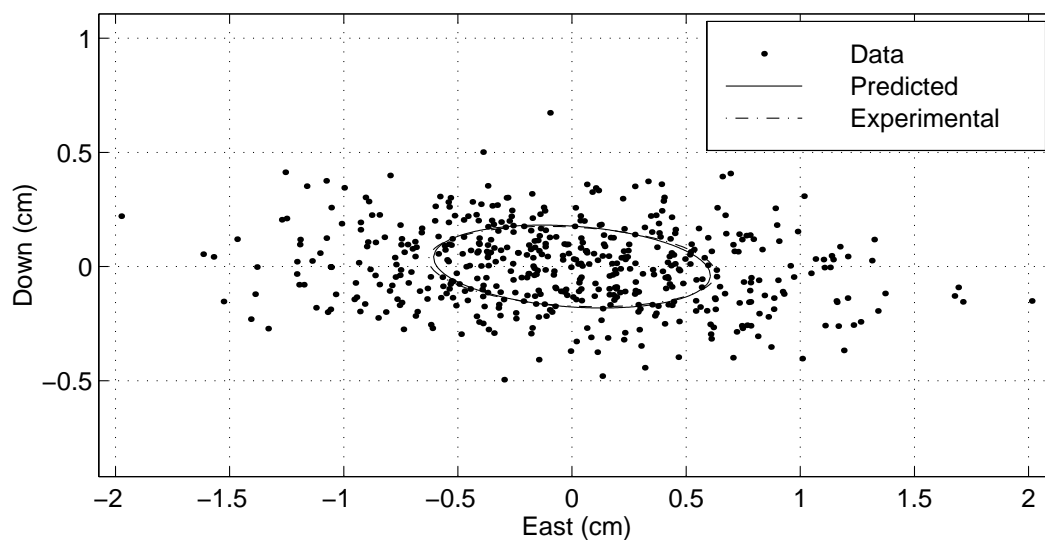


Figure 6.2: Projection of 500 noisy lever arm corrections onto the East-Down plane for the assumed attitude of -1.1° in roll, 0.8° pitch, and 283.0° yaw. One-sigma ellipsoids (predicted as solid line, experimental as dashed) shown.

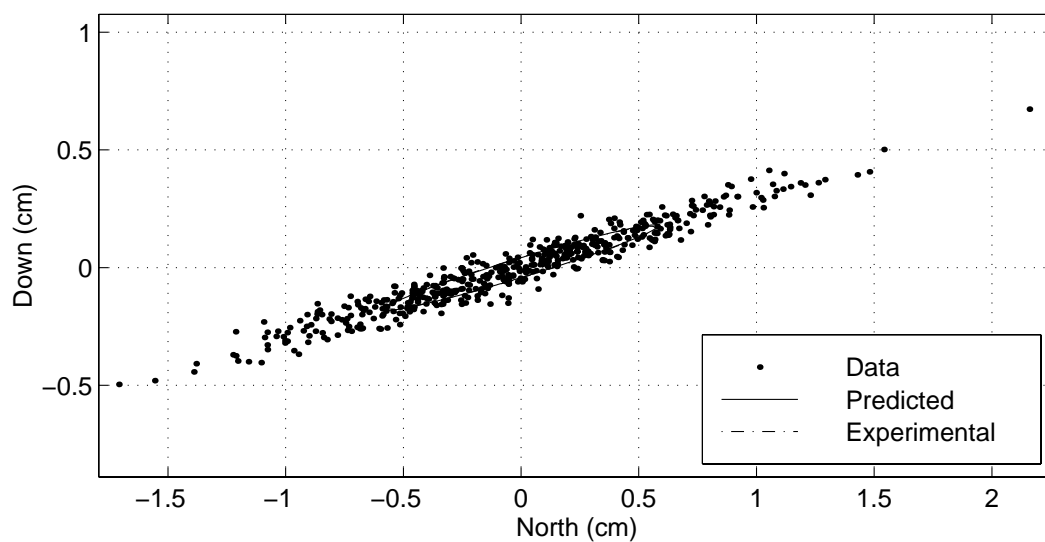


Figure 6.3: Projection of 500 noisy lever arm corrections onto the North-Down plane for the assumed attitude of -1.1° in roll, 0.8° pitch, and 283.0° yaw. One-sigma ellipsoids (predicted as solid line, experimental as dashed) shown, but barely visible from this viewpoint.

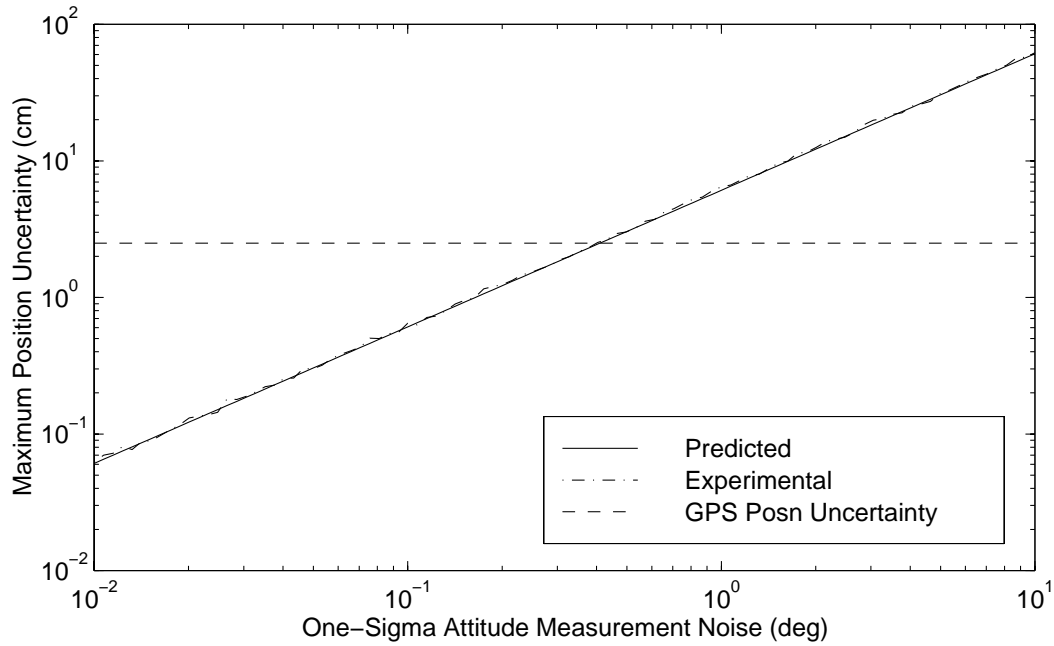


Figure 6.4: Maximum position uncertainty vs. attitude measurement noise (predicted as solid line, experimental as dashed). GPS positioning system nominal measurement uncertainty of 2.5 cm shown as dashed horizontal line.

experimental uncertainty in any direction was 0.62 cm. The square root of the norm of the difference between the predicted and experimental uncertainty ellipsoids was 0.15 cm. Therefore, Eqn. (6.7) represents a reasonably accurate approximation to the additional uncertainty introduced by the lever arm correction.

6.3 Effects of Attitude Noise and Lever Arm Length

To quantify the effect of various attitude noise levels, attitude measurement noise was allowed to vary between 0.01° and 10° (one-sigma). Fig. (6.4) shows the maximum additional control point position uncertainty introduced by the lever arm as a function of increasing attitude noise using the same lever arm and attitude measurements specified in Eqns. (6.11-6.12). The horizontal dashed line at 2.5 cm represents the nominal measurement noise that can be expected from the positioning system. From Fig. (6.4), it is clear that as attitude measurement noise increases above approximately

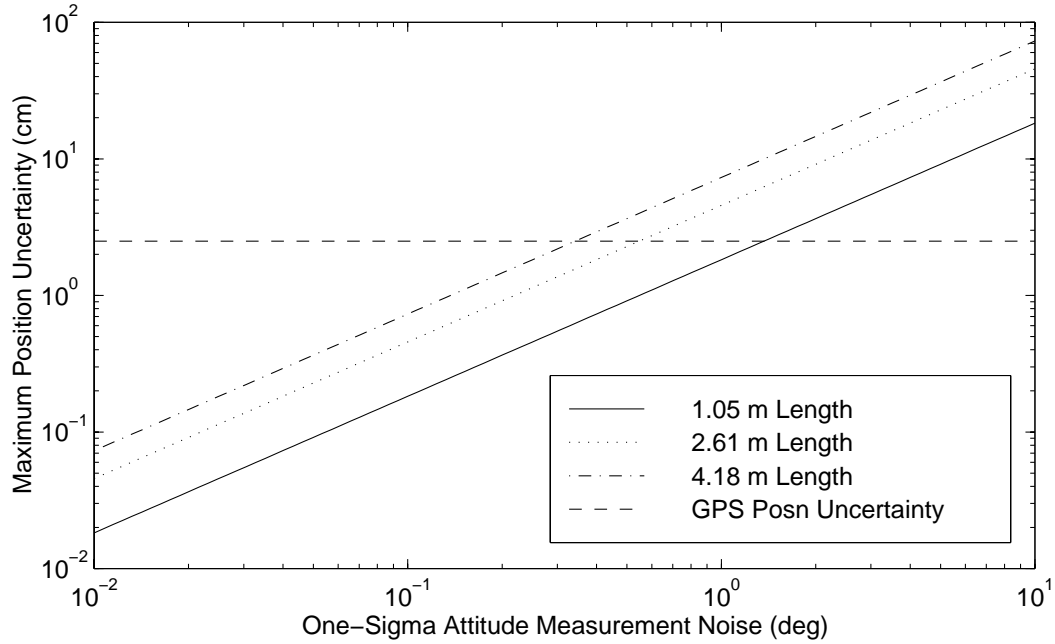


Figure 6.5: Maximum position uncertainty vs. attitude measurement noise (predicted as solid line, experimental as dashed) for 3.5 m lever arm length. GPS positioning system nominal measurement uncertainty of 2.5 cm shown as dashed horizontal line.

0.4° (one-sigma), *the dominant source of positional uncertainty at the control point is not the GPS position sensor, but the GPS attitude sensor.*

Maximum position uncertainty introduced by the lever arm versus attitude measurement noise for lever arms of varying lengths are shown in Fig. (6.5). The lever arm used in the previous simulations was scaled by 30%, 75%, and 120%. Clearly, as the lever arm becomes longer, the adverse effect of attitude measurement noise becomes more pronounced. Tbl. 6.2 lists lever arm lengths against the level of attitude measurement noise at which the lever arm uncertainty eclipses the nominal GPS positional uncertainty of 2.5 cm (one-sigma).

Some researchers have proposed using a rough heading measurement from a compass and terrain knowledge to estimate vehicle attitude. This rougher estimate of the vehicle's attitude implies a degradation in the position uncertainty. Under the following assumptions,

| Lever arm length | Attitude measurement noise for 2.5 cm uncertainty |
|---------------------|--|
| 1.1 m | 1.4° |
| 2.6 m | 0.6° |
| 3.5 m | 0.4° |
| 4.2 m | 0.3° |

Table 6.2: Attitude measurement noise level yielding 2.5-cm positional uncertainty for various lever arm lengths

1. No attitude measurement beyond a rough yaw measurement with a measurement uncertainty of 1.0° was available.
2. The tractor was being used on a flat field at low speed.
3. Roll and pitch measurements ranges up to $\pm 3^\circ$.
4. Roll and pitch disturbances had one-sigma standard deviations of 2.0° .

it can be shown that the largest uncertainty in any direction was 12.2 cm (one-sigma). With a nominal position uncertainty of 2.5 cm, the poor attitude measurement became the dominant source of positional uncertainty by incurring additional uncertainty of almost five times the original uncertainty. Therefore, some measure of attitude, even on flat terrain, is vital if attitude uncertainty is not to dominate the measurement noise.

CHAPTER 7

Vehicle Control on Arcs

Realistic farm trajectories will most likely contain circular arcs for two reasons. First, farmers prefer to minimize the amount of time the implement spends out of the ground. Therefore, if discontinuous linear path segments could be smoothly joined by gentle arcs while keeping the implement in the ground,¹ this wasted time could be minimized. Second, some path-planning algorithms use arcs to satisfy non-holonomic constraints [Lat91]. Some sort of path-planning algorithm will almost certainly be required in a practical automatic vehicle controller, and the resulting paths generated by the planner might include arcs. Therefore, a control algorithm was sought that could control a wheeled vehicle accurately on circular arcs. Controller performance was measured by the mean and standard deviation of the radial tracking error.

7.1 Specifying Arc Trajectories

An arc trajectory was specified by a center, start point, and an interior turning angle (see Fig. 7.1). The arc radius was assumed to be larger than the minimum turning radius of the vehicle. The current location of the vehicle along the arc was calculated by finding the intersection of a line between the vehicle's control point and the arc center. The vehicle's yaw angle relative to the arc could then be calculated. Arcs could be specified as either clockwise or counter-clockwise.

¹As opposed to lifting the implement out of the ground and reorienting the vehicle.

7.2 Modeling the Vehicle in Polar Coordinates

To track an arc accurately, the linearized vehicle model in Eqns. (3.43–3.45) had to be transformed into polar coordinates. Attempts to linearize the model along straight rows would lead to significant (at our control levels) steady-state radial tracking errors.

7.2.1 Vehicle Yaw Error

From Fig. (7.1), the angle of the curve tangent is $\alpha + \alpha_0 \pm \pi/2$, where α is the angle swept out by the vehicle, and α_0 is the “start angle” of the arc. A control point not collocated with the vehicle’s pivot point meant that, in steady-state, a small heading angle correction ξ had to be accounted for. From geometry,

$$\xi = \mp \sin^{-1} \left(\frac{l_2}{R_0} \right) \quad (7.1)$$

where R_0 is the arc radius. Since ξ is positive clockwise in keeping with the NED coordinate frame, ξ will be less than zero if the arc is turning clockwise and l_2 is positive. Note that in Fig. (7.1), the vehicle’s control point is located to the rear of the pivot point and l_2 would be negative in this case. Therefore, the vehicle yaw angle in the inertial frame could be expressed as

$$\psi = \alpha + \alpha_0 \pm \frac{\pi}{2} + \xi + \psi_e \quad (7.2)$$

where ψ_e is the vehicle’s yaw error from the steady state yaw angle required to track the arc. The $\pi/2$ correction stems from the fact that the vehicle travels nearly tangent to the arc ($+\pi/2$ for clockwise arcs). Solving for ψ_e and taking the time derivative,

$$\dot{\psi}_e = \dot{\psi} - \dot{\alpha} - \dot{\xi} \quad (7.3)$$

Since the radius of the arc remains constant, $\dot{\xi}$ is zero. An expression for $\dot{\alpha}$ can be derived by transforming the north and east components of the vehicle’s velocity in the body frame

$$\dot{n} = V_x \quad (7.4)$$

$$\dot{e} = l_2 \dot{\psi} \quad (7.5)$$

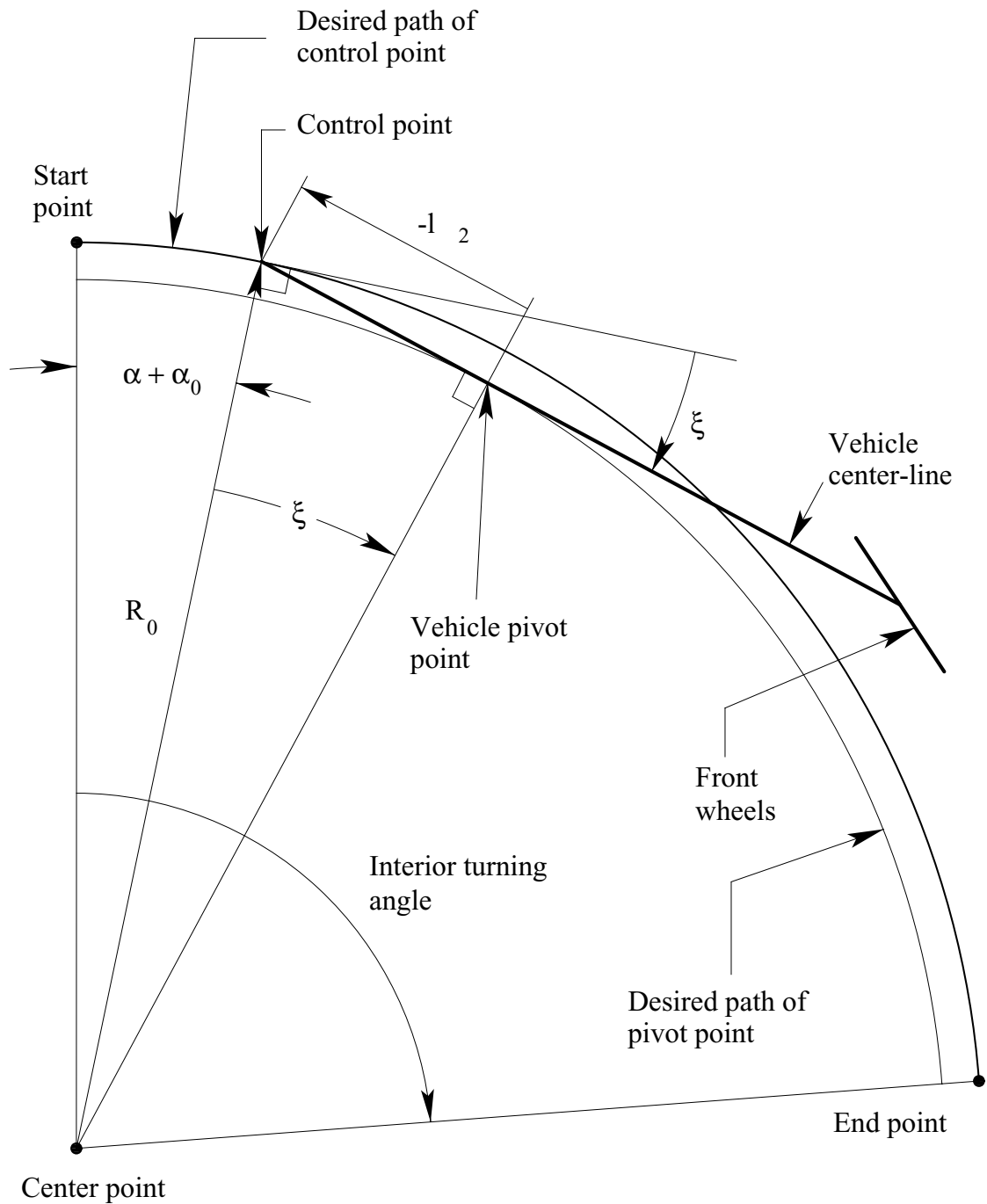


Figure 7.1: Diagram of vehicle moving along arc.

into a polar coordinate frame aligned with the arc's center. The forward velocity of the vehicle will then be tangential to the arc and will be equal to $r\dot{\alpha}$ where r is defined as the distance of the vehicle's control point from the arc's center. The angle needed to transform the control point velocity vector will be the angle between the vehicle's centerline and the arc tangent: $\psi_e + \xi$. In polar coordinates, the tangential and radial velocities are

$$r\dot{\alpha} = V_x \cos(\psi_e + \xi) - l_2 \dot{\psi} \sin(\psi_e + \xi) \quad (7.6)$$

$$\dot{r} = V_x \sin(\psi_e + \xi) + l_2 \dot{\psi} \cos(\psi_e + \xi) \quad (7.7)$$

The expressions for $\dot{\alpha}$ in Eqn. (7.6) and $\dot{\psi}$ in Eqn. (3.36) can be substituted into Eqn. (7.3) to solve for $\dot{\psi}_e$:

$$\dot{\psi}_e = \dot{\psi} - \frac{1}{r} \left[V_x \cos(\psi_e + \xi) - l_2 \dot{\psi} \sin(\psi_e + \xi) \right] \quad (7.8)$$

$$= \frac{1}{l_1} K_\delta V_x \tan \delta \left[1 + \frac{l_2}{r} \sin(\psi_e + \xi) \right] + \frac{1}{r} V_x \cos(\psi_e + \xi) \quad (7.9)$$

Note that as r approaches infinity, $\dot{\psi}_e$ approaches $\dot{\psi}$.

7.2.2 Vehicle Steer Angle

The vehicle's steady-state steer angle can be computed from the steady-state yaw rate. Since $\dot{\xi}$ is zero,

$$\dot{\psi}_\infty = \dot{\alpha}_\infty \quad (7.10)$$

An expression for $\dot{\alpha}_\infty$ can be derived from Eqn. (7.6) for zero error in yaw, steering, and tracking:

$$\dot{\alpha}_\infty = \frac{1}{R_0} \left[V_x \cos \xi - l_2 \dot{\psi}_\infty \sin \xi \right] = \dot{\psi}_\infty \quad (7.11)$$

Solving for $\dot{\psi}_\infty$ and using Eqn. (3.36),

$$\dot{\psi}_\infty = \frac{V_x \cos \xi}{R_0 + l_2 \sin \xi} = \frac{1}{l_1} V_x K_\delta \tan \delta_\infty \quad (7.12)$$

$$\delta_\infty = \pm \tan^{-1} \left(\frac{l_1}{K_\delta \sqrt{R_0^2 - l_2^2}} \right) \quad (7.13)$$

where the sign of δ_∞ is defined as positive for clockwise arcs. Note that Eqn. (7.13) could also have been derived from geometry. Clearly, Eqn. (7.13) becomes singular as l_2 approaches R_0 . However, tracking an arc with radius less than reasonable values of l_2 would be unrealistic.

Steer angle could be linearized about some (nominally zero) steer angle δ_e :

$$\delta = \delta_e + \delta_\infty \quad (7.14)$$

so that

$$\dot{\delta}_e = \dot{\delta} - \dot{\delta}_\infty \quad (7.15)$$

Since the arc radius is constant, $\dot{\delta}_\infty$ is zero. Therefore, as in the linear case,

$$\dot{\delta}_e = u \quad (7.16)$$

7.2.3 Vehicle Tracking Error

The distance of the vehicle control point from the center of the arc can be expressed as

$$r = r_e + R(\alpha) \quad (7.17)$$

$$= r_e + R_0 \quad (7.18)$$

Since the radius of the arc is constant, the expression for \dot{r}_e is the same as for \dot{r} in Eqn. (7.7).

7.2.4 Equations of Motion

Eqns. (7.9), (7.16), and (7.7) can be combined to form the equations describing the motion of a vehicle tracking an arc. The three states in the model, ψ_e , δ_e , and r_e , are

zero when the arc is being tracked perfectly:

$$\begin{aligned} \dot{\psi}_e &= \frac{1}{l_1} K_\delta V_x \tan(\delta_e + \delta_\infty) \left[1 + \frac{l_2}{r_e + R_0} \sin(\psi_e + \xi) \right] \\ &\quad + \frac{1}{r_e + R_0} V_x \cos(\psi_e + \xi) \end{aligned} \quad (7.19)$$

$$\dot{\delta}_e = u \quad (7.20)$$

$$\dot{r}_e = V_x \sin(\psi_e + \xi) + \frac{l_2}{l_1} K_\delta V_x \tan(\delta_e + \delta_\infty) \cos(\psi_e + \xi) \quad (7.21)$$

7.3 Tracking Control Algorithm

Simulations showed that a linear control algorithm created by linearizing Eqns. (7.19–7.21) yielded good tracking performance. Therefore, Eqns. (7.19–7.21) were linearized about the nominally zero operating conditions:

$$\begin{aligned} \dot{\psi}_e &\cong \left[\frac{l_2}{l_1 R_0} K_\delta V_x \tan \delta_\infty \cos \xi - \frac{1}{R_0} V_x \sin \xi \right] \psi_e \\ &\quad + \left[\frac{1}{l_1} K_\delta V_x (1 + \tan^2 \delta_\infty) \left(1 + \frac{l_2}{R_0} \sin \xi \right) \right] \delta_e \end{aligned} \quad (7.22)$$

$$\dot{\delta}_e = u \quad (7.23)$$

$$\begin{aligned} \dot{r}_e &\cong \left[V_x \cos \xi - \frac{l_2}{l_1} K_\delta V_x \tan \delta_\infty \sin \xi \right] \psi_e \\ &\quad + \left[\frac{l_2}{l_1} K_\delta V_x (1 + \tan^2 \delta_\infty) \cos \xi \right] \delta_e \end{aligned} \quad (7.24)$$

As the arc approaches infinite radius, Eqns. (7.22) and (7.24) approach the linearized equations for a straight line given in Eqns. (3.43) and (3.45).

The discrete root locus for the linearized system with a nominal arc radius of ten meters was indistinguishable from the discrete root locus shown in Fig. (3.2) for the model linearized about a straight line.

7.4 Experimental Results

The vehicle model varied with velocity, the steering proportionality constant K_δ , and the control point's longitudinal position (through l_2). Therefore, scheduling the

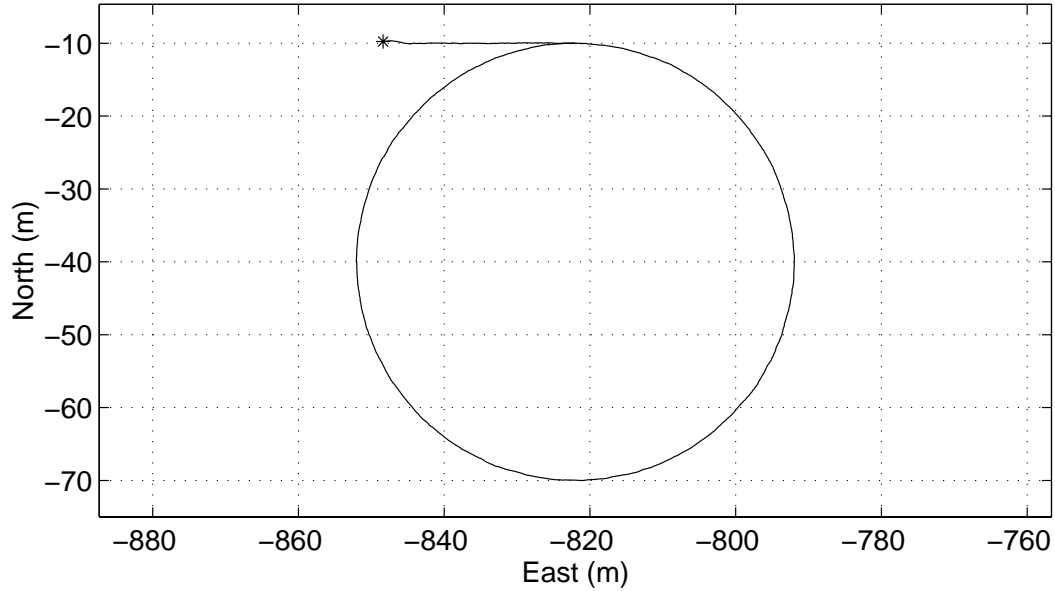


Figure 7.2: Overhead view of experimental arc trajectory. Coordinates are relative to the reference station antenna.

control gains against these factors was impractical. Instead, LQR control gains were solved for in real-time. Although MatLab's `dlqr.m` function can be translated into an executable file for many computer operating systems, no such translator existed for this project's operating system (LynxOS). Therefore, software was sought that could solve for the control gains by finding the solution of the control discrete algebraic Riccati equation. The freeware Fortran 77 library, SLICOT, provided a subroutine SB02MD,² which successfully solved the Riccati equation in less than 5% of a 200 ms sample interval.

For the experimental trajectory shown in Fig. (7.2), the radius was 30.0 m. The experiment was performed using the Model 8400 test platform. The measured tracking error is shown in Fig. (7.3). A single set of rear wheels was used on a rough field; therefore, ground disturbances were significantly larger than what could normally be expected. No implement was allowed on this particular field. The mean of the

²SB02MD used the Schur vector method proposed by Laub. Interested readers are referred to [Lau79, Sim96]. Software is available at <http://www.win.tue.nl/wgs/> and <ftp://wgs.esat.kuleuven.ac.be/pub/WGS/SLICOT/>

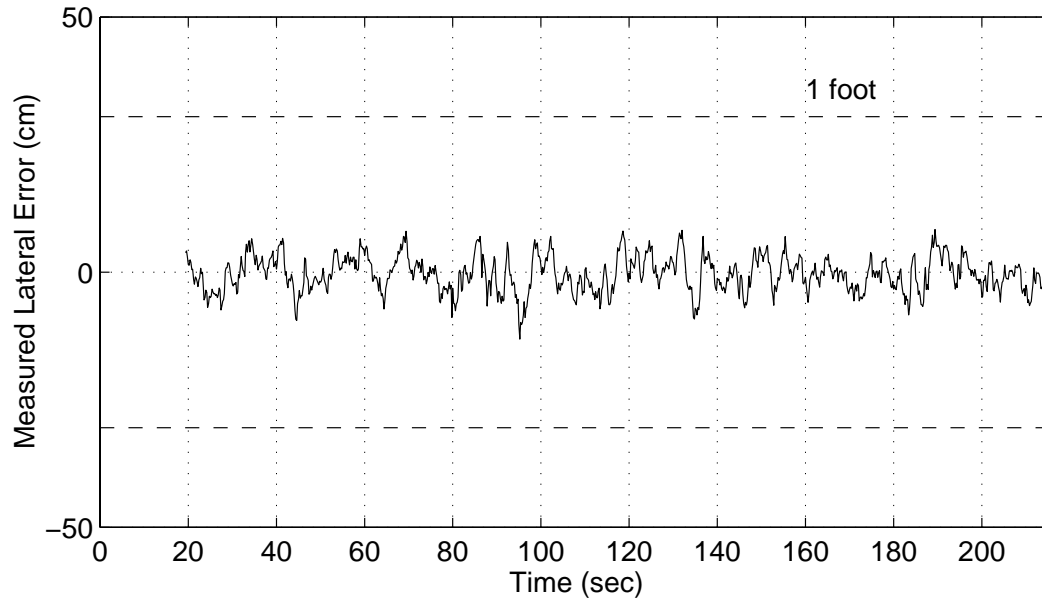


Figure 7.3: Tracking error along experimental arc trajectory. Mean tracking error was -0.28 cm, and the standard deviation was 3.43 cm one-sigma.

control point's tracking error was -0.28 cm and the standard deviation was 3.43 cm (one-sigma). Bias states in yaw and steering were used to compensate for modeling errors. Vehicle velocity was 1.1 m/s. Tracking performance did degrade with higher velocities, but the amount of deterioration for a given velocity increase depended strongly on the ground conditions. Because most tractors (including the two used in this research), had no suspension, even relatively small ground disturbances were significant. In a field that had not been smoothed,³ forward velocities of over approximately $2\text{--}3$ m/s meant disturbances were intolerably large. In a smoothed field, velocities all the way up to the maximum forward velocity of 10.0 m/s were possible.

³Usually with an implement such as a disk.

CHAPTER 8

Vehicle Control on Spirals

In arid and semi-arid regions of the world, “center-pivot” irrigation systems are often used. In such systems, a central water source feeds water to a long pipe with sprinklers mounted on six to seven mobile towers. Center-pivot irrigation systems, patented in 1952, offer farmers automatic operation, reduced labor, control of application rate, and precise application of herbicides and fertilizers. A satellite study conducted in the summer of 1976 revealed that 1.3 million acres of Nebraska farm land were being irrigated by center-pivot irrigation [Spl76, M⁺92]. The resulting crop area is circular, and though the area may be farmed in rows, the area could be more efficiently farmed if the tractor could operate continuously without removing the implement from the ground during U-turns. A spiral trajectory would allow the operator to drive without stopping, decreasing the time required to farm the area. Although the spiral pattern could be broken up into sequential arcs, the junctions would be discontinuous, and a spiral pattern would be more efficient. Therefore, an algorithm was sought which could control the tractor accurately on a spiral trajectory. The algorithm had to be flexible enough so that the tractor could spiral either inward or outward, traveling either clockwise or counter-clockwise.

8.1 Specifying Spiral Trajectories

A spiral trajectory was specified by a center, start point, interior angle, and “width”, denoted w . The spiral decreased or increased, depending on the sign of w , one width



Figure 8.1: Center-pivot irrigation systems measuring almost a mile in diameter in the Kufra oasis in the Libyan Desert [Spl76].

per revolution. Therefore, the distance from the center to a point on the spiral, called the spiral radius and denoted by ρ , as a function of the swept angle α was

$$\rho = \rho_0 + \beta\alpha \quad (8.1)$$

where

$$\beta = \pm \frac{w}{2\pi} \quad (8.2)$$

The initial spiral radius ρ_0 was the distance between the center and start points, and α could range from zero to $2\pi n$ where n was the number of revolutions completed.

8.1.1 Radius of Curvature

The radius of curvature R of the spiral did *not* equal the spiral radius ρ because ρ changed as a function of the swept angle. Any coordinate on the spiral can be written

as

$$x(\alpha) = \begin{bmatrix} (\rho_0 + \beta\alpha) \cos(\alpha + \alpha_0) \\ (\rho_0 + \beta\alpha) \sin(\alpha + \alpha_0) \end{bmatrix} \quad (8.3)$$

The radius of curvature was [Kre93]

$$R(\alpha) = \frac{\left(\frac{dx}{d\alpha} \cdot \frac{dx}{d\alpha}\right)^{3/2}}{\sqrt{\left(\frac{dx}{d\alpha} \cdot \frac{dx}{d\alpha}\right) \left(\frac{d^2x}{d\alpha^2} \cdot \frac{d^2x}{d\alpha^2}\right) - \left(\frac{dx}{d\alpha} \cdot \frac{d^2x}{d\alpha^2}\right)^2}} \quad (8.4)$$

With

$$\frac{dx}{d\alpha} = \begin{bmatrix} \beta \cos(\alpha + \alpha_0) - (\rho_0 + \beta\alpha) \sin(\alpha + \alpha_0) \\ \beta \sin(\alpha + \alpha_0) + (\rho_0 + \beta\alpha) \cos(\alpha + \alpha_0) \end{bmatrix} \quad (8.5)$$

$$\frac{d^2x}{d\alpha^2} = \begin{bmatrix} -2\beta \sin(\alpha + \alpha_0) - (\rho_0 + \beta\alpha) \cos(\alpha + \alpha_0) \\ 2\beta \cos(\alpha + \alpha_0) - (\rho_0 + \beta\alpha) \sin(\alpha + \alpha_0) \end{bmatrix} \quad (8.6)$$

the radius of curvature reduced to

$$R = \frac{[\rho^2 + \beta^2]^{3/2}}{\rho^2 + 2\beta^2} \quad (8.7)$$

The maximum angle that the vehicle's front wheels were capable of turning specified the minimum radius of curvature possible. Equating yaw rate in Eqn. (3.36) with V_x/R ,

$$\frac{1}{R_{min}} V_x = \frac{1}{l_1} V_x K_\delta \tan \delta_{max} \quad (8.8)$$

$$R_{min} = \frac{l_1}{K_\delta \tan \delta_{max}} \quad (8.9)$$

Eqn. (8.7) was equated with Eqn. (8.9) to determine how close the spiral could have wound into its center by calculating ρ given β .

8.2 Modeling the Vehicle in Polar Coordinates

The polar analysis developed for arcs in Ch. 7 was adapted to spiral trajectories. The time-varying radius of curvature meant that several time derivatives in the equations of motion were no longer zero, as was the case for arcs.

8.2.1 Vehicle Yaw Error

The yaw angle of any point along the spiral was no longer given by Eqn. (7.2). Instead, because the spiral curved either inward or outward, a small angle η resulting from the changing radius had to be accounted for:

$$\psi = \alpha + \alpha_0 \pm \frac{\pi}{2} + \xi + \eta + \psi_e \quad (8.10)$$

The yaw error dynamics were described by

$$\dot{\psi}_e = \dot{\psi} - \dot{\alpha} - \dot{\xi} - \dot{\eta} \quad (8.11)$$

where ξ was specified in Eqn. (7.1) and $\dot{\psi}$ in Eqn. (3.36).

If α was changed by a small amount $d\alpha$, then the arc length along the arc of radius ρ will be $\rho d\alpha$ and the radius would have changed by $\rho(\alpha) - \rho(\alpha + d\alpha)$. Therefore,

$$\tan \eta = \frac{\rho(\alpha) - \rho(\alpha + d\alpha)}{\rho(\alpha) d\alpha} \quad (8.12)$$

$$= -\frac{\beta d\alpha}{\rho(\alpha) d\alpha} \quad (8.13)$$

$$= -\frac{\beta}{\rho} \quad (8.14)$$

The rate of change of the swept angle $\dot{\alpha}$ was calculated from Eqn. (7.6) with the slight change that the yaw angle of the vehicle relative to the spiral tangent now included the angle η :

$$r\dot{\alpha} = V_x \cos(\psi_e + \xi + \eta) - l_2 \dot{\psi} \sin(\psi_e + \xi + \eta) \quad (8.15)$$

$$\dot{\alpha} = \frac{1}{r_e + \rho} \left[V_x \cos(\psi_e + \xi + \eta) - l_2 \dot{\psi} \sin(\psi_e + \xi + \eta) \right] \quad (8.16)$$

The rate of change of η can be derived by differentiating Eqn. (8.14):

$$\dot{\eta} = -\frac{\beta^2 \dot{\alpha}}{\beta^2 + \rho^2} \quad (8.17)$$

Similarly, the rate of change of ξ was found by differentiating Eqn. (7.1):

$$\dot{\xi} = \pm \frac{l_2 \dot{R}}{R \sqrt{R^2 - l_2^2}} \quad (8.18)$$

$$= \pm \frac{l_2 \frac{dR}{d\alpha} \dot{\alpha}}{R \sqrt{R^2 - l_2^2}} \quad (8.19)$$

From Eqn. (8.7),

$$\frac{dR}{d\alpha} = \frac{3\rho\sqrt{\rho^2 + \beta^2}}{\rho^2 + 2\beta^2} - \frac{2\rho(\rho^2 + \beta^2)^{3/2}}{(\rho^2 + 2\beta^2)^2} \quad (8.20)$$

Finally,

$$\dot{\psi}_e = \frac{1}{l_1} V_x K_\delta \tan \delta - \dot{\alpha} \left[1 - \frac{\beta^2}{\beta^2 + \rho^2} \pm \frac{l_2 \frac{dR}{d\alpha}}{R\sqrt{R^2 - l_2^2}} \right] \quad (8.21)$$

8.2.2 Vehicle Steer Angle

As for control on arcs, steer angle was broken apart into a steady-state value δ_∞ and an error value δ_e so that

$$\delta_e = \delta - \delta_\infty \quad (8.22)$$

$$\dot{\delta}_e = \dot{\delta} - \dot{\delta}_\infty \quad (8.23)$$

The steady-state value was determined by solving Eqn. (8.21) for $\delta = \delta_\infty$ given zero error in yaw, steering, and tracking:

$$\dot{\psi}_\infty = \dot{\alpha}_\infty \left[1 - \frac{\beta^2}{\beta^2 + \rho^2} \pm \frac{l_2 \frac{dR}{d\alpha}}{R\sqrt{R^2 - l_2^2}} \right] \quad (8.24)$$

$$\dot{\psi}_\infty = \frac{V_x \cos(\xi + \eta)}{\rho + l_2 \sin(\xi + \eta)} \left[\begin{array}{c} 1 - \frac{\beta^2}{\beta^2 + \rho^2} \\ \pm \frac{l_2 \frac{dR}{d\alpha}}{R\sqrt{R^2 - l_2^2}} \end{array} \right] \quad (8.25)$$

$$\tan \delta_\infty = \frac{l_1 \cos(\xi + \eta)}{K_\delta [\rho + l_2 \sin(\xi + \eta)]} \left[\begin{array}{c} 1 - \frac{\beta^2}{\beta^2 + \rho^2} \\ \pm \frac{l_2 \frac{dR}{d\alpha}}{R\sqrt{R^2 - l_2^2}} \end{array} \right] \quad (8.26)$$

An expression for $\dot{\delta}_\infty$ was derived by manipulating Eqn. (8.26) to solve for δ_∞ and then differentiating with respect to time.

8.2.3 Vehicle Tracking Error

As for control on arcs, tracking error was broken apart into a steady-state value and an error value so that

$$r_e = r - \rho \quad (8.27)$$

$$\dot{r}_e = \dot{r} - \dot{\rho} \quad (8.28)$$

$$= V_x \sin(\psi_e + \xi + \eta) + l_2 \dot{\psi} \cos(\psi_e + \xi + \eta) - \beta \dot{\alpha} \quad (8.29)$$

8.2.4 Equations of Motion

The equations of motion are given by Eqn. (8.21), differentiating Eqn. (8.26), and Eqn. (8.29). These equations can then be linearized to form a simple three-state model for computing linear control gains.

8.2.5 Analysis

To show the effect of spiral radius on reference states, a spiral was created with representative parameter values $w = 5.0$ m, $\rho_0 = 4.0$ m, $V_x = 1.0$ m/s, $l_2 = 0.0$ m, $l_1 = 2.80$ m, and $K_\delta = 1.0$. An overhead view of the spiral is shown in Fig. (8.2). Fig. (8.3) shows the reference steer angle as a function of radius. As the distance between the front axle and rear pivot point decreased, the required steer angle also decreased. The fundamental limit on how small the spiral could become was the vehicle's turning radius, although spirals of radius less than 5-6 meters could safely be considered unrealistic.

8.3 Experimental Results

Fig. (8.4) shows an overhead view of an actual spiral trajectory completed in early December 1997 using the Model 7800 test platform detailed in App. A. The trajectory was completed in under 35 minutes with a towed implement attached and the tractor operating under full throttle and full engine load. Fig. (8.5) shows the tracking error as a function of time. A small portion of the test area had been bedded, and the

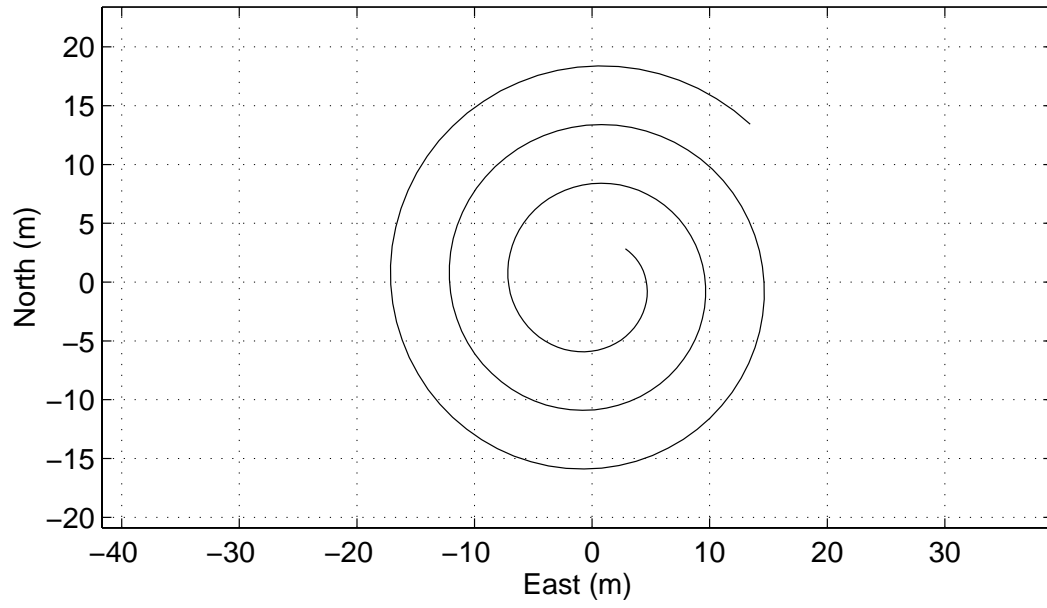


Figure 8.2: Overhead view of simulated spiral trajectory.

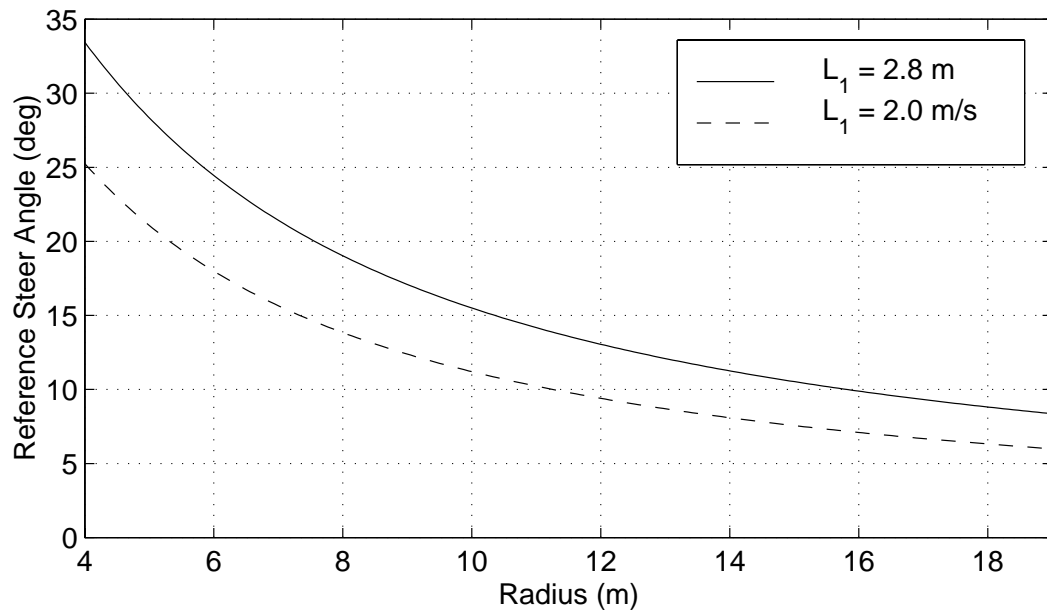


Figure 8.3: Reference steer angle as a function of spiral radius.

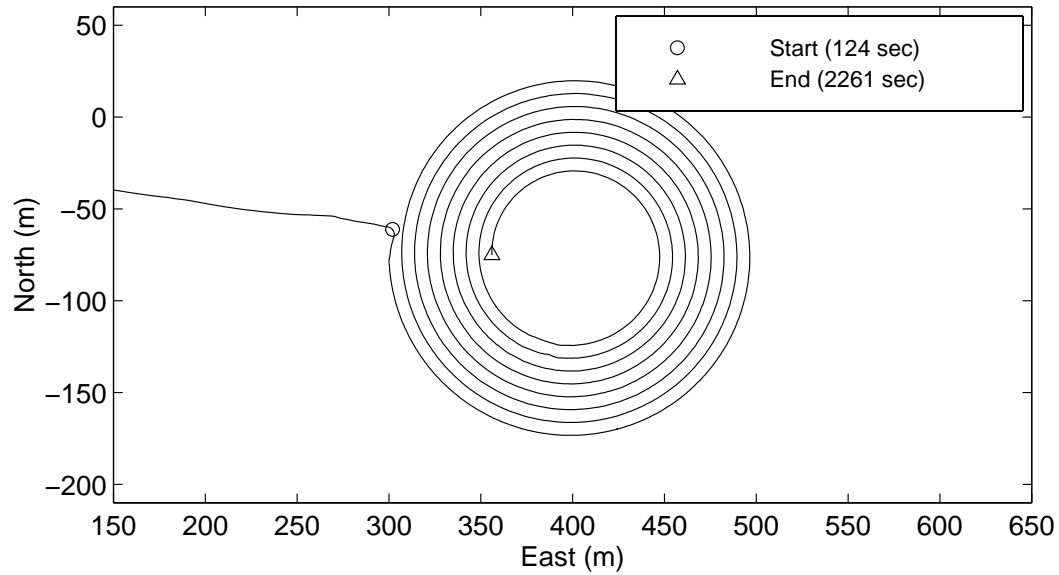


Figure 8.4: Overhead view of experimental spiral trajectory. Coordinates are relative to the reference station antenna.

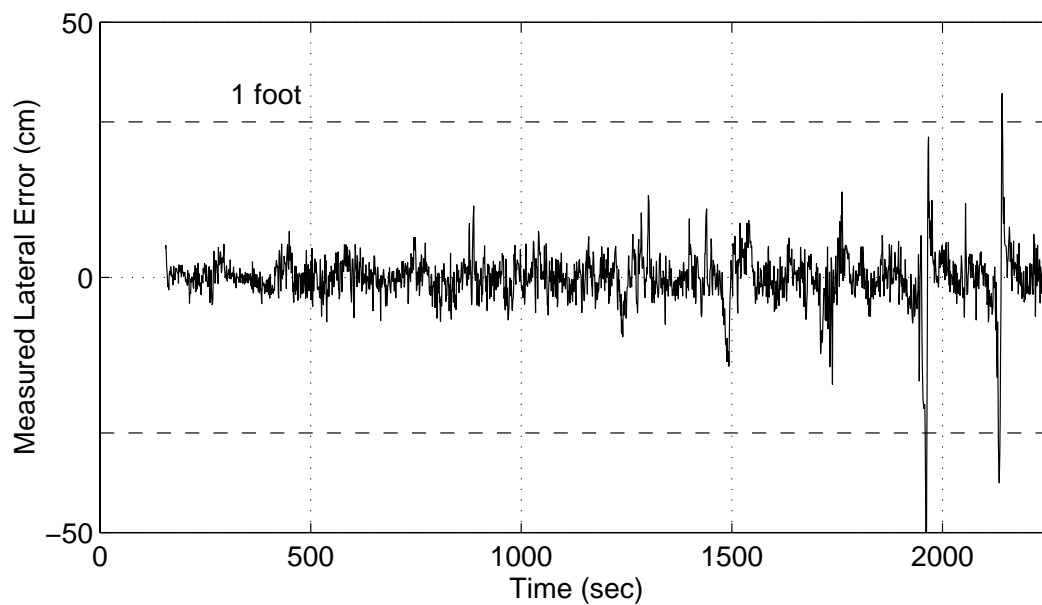


Figure 8.5: Tracking error along experimental spiral trajectory. Mean error was -0.22 cm and standard deviation (one-sigma) was 5.27 cm.

sawtooth-shaped beds caused the tractor to slide sideways down into the furrows. The disturbances were compounded by the large towing force required to pull the implement. The disturbances from the beds are visible as spikes near the end of the tracking error history. In general, as the spiral radius decreased, disturbances' effects on tracking error became more pronounced. Nonetheless, the mean tracking error was -0.22 cm and the standard deviation 5.27 cm (one-sigma) at a forward velocity of 2.1 m/s. A small amount of integral control was used to compensate for modeling errors, and the LQR control gains were computed in real-time using the real-time discrete algebraic Riccati equation solver mentioned in Ch. 7. These quantitative results were typical of other spiral trajectories completed.

Control along spirals was typically very stable for reasonable radii, even at higher forward speeds. Hitched implements tend to resist a yaw change, while towed implements tend to pull the back end of the tractor inward. Integral control helped compensate for both effects. Factors affecting yaw response to front-wheel angle had a greater detrimental impact on spiral controller performance than when the tractor tracked straight lines.

CHAPTER 9

Vehicle Control on Curves: Specifying Curve Trajectories

This chapter will create or adapt algorithms that could generate curve trajectories from discrete position data. These trajectories must closely match input data, yet filter out measurement and disturbance noise that corrupted the data. Control on curves has many applications in a variety of areas including missile guidance [K⁺98], aircraft guidance, satellite control, computer-aided machining [Tom93], and ship control [God98]. In agriculture, curved trajectories may arise when a river or a road defines a field boundary. Avoiding in-field obstacles such as telephone poles may also require nonlinear trajectories. Curved trajectories could be specified by topographic map data or by position data collected during a human-controlled trajectory. Since a trajectory could be derived from either topographic data or real-time position data, a flexible trajectory generation algorithm was needed that could generate a smooth trajectory from both types of inputs without affecting controller performance. Control on curve trajectories revealed several issues: vehicle location on the curve had to be calculated, time-derivatives of curve parameters had to be estimated, and previewing future curve information in the control algorithm required predicting the vehicle's future location(s) on the curve.

9.1 Generating Curve Trajectories from Position Data

There were at least four possible approaches to generating a trajectory given position data. The first approach was to treat the trajectory as a series of “waypoints” and control the vehicle from one waypoint to the next. Although there was no computational cost in generating the trajectory, the resulting vehicle trajectory would not match the original trajectory if the distance between waypoints was large. In the second approach, line segments interconnect trajectory points, and the vehicle tracks each line as it moves on the trajectory. This second approach suffers from the same poor fidelity as the first if points are located far apart. The third approach and the one taken in this research was to use cubic splines to interpolate smooth continuous curves between the data points. The fourth, and “optimal approach”, created a trajectory which passed as close as possible to the data points while satisfying the vehicle’s non-holonomic constraints. This approach had a high computational cost and could diverge with a poor initial guess.

Various curves could specify parts of or entire trajectories: clothoids, polar splines, B-splines, and splines [Nel89, SO95]. A variety of spline interpolants exist including polynomials of degree two and higher, rational spline interpolants, and exponential spline interpolants [Spa95]. Third-order polynomial spline interpolants were chosen because the resulting curves appeared the most “natural”, and generating cubic spline interpolants also had a relatively low computational cost.

Splines have been used to generate smooth trajectories for ships [God98]. In [God98], a series of arcs and lines were used to prescribe the shortest trajectory for a ship given a minimum turning radius. Discrete points on the trajectory were chosen, and seventh-order polynomial splines¹ were used to fit the discrete data.

¹The splines were seventh-order because they had to satisfy dynamic constraints based on the ship’s equations of motion.

9.2 Cubic Splines

Given a series of $N \geq 3$ data points (x_k, y_k) and x_k strictly increasing, cubic polynomials interpolate between the data points:

$$y_k(x) = A_k + B_k(x - x_k) + C_k(x - x_k)^2 + D_k(x - x_k)^3 \quad (9.1)$$

where $x_k \leq x \leq x_{k+1}$. The endpoints as well as the first and second derivatives of adjoining splines are required to match, yielding the boundary conditions for each spline. The equations in the unknown spline coefficients A_k , B_k , C_k , and D_k can be easily solved using linear algebra. Cubic spline interpolants could be forced to be periodic, which might be required when working on a field boundary, for example [Spa95].

Though the requirement that x_k be strictly increasing seemed to present a problem for curves that “doubled back”, the north and east coordinates could actually be parameterized by an independent parameter s . If (n_k, e_k) represented a north-east coordinate pair, one choice of an independent parameter was the sum of the Euclidean distances between points up to number k was:

$$s_k = s_{k-1} + \sqrt{(n_k - n_{k-1})^2 + (e_k - e_{k-1})^2} \quad (9.2)$$

$$s_1 = 0 \quad (9.3)$$

Trajectory coordinates were

$$n(s) = A_{n,k} + B_{n,k}(s - s_k) + C_{n,k}(s - s_k)^2 + D_{n,k}(s - s_k)^3 \quad (9.4)$$

$$e(s) = A_{e,k} + B_{e,k}(s - s_k) + C_{e,k}(s - s_k)^2 + D_{e,k}(s - s_k)^3 \quad (9.5)$$

Therefore, any point on the trajectory was uniquely determined by a value of s .

The cubic spline trajectory passed through all data points. To smooth through noisy input data, the condition that the interpolated trajectory pass through each data point could be relaxed by specifying a positive “fit” or “tightness” parameter p_k for each point. The interested reader is referred to [Spa95]. To demonstrate how splines could be used to smooth through noisy input data, a simulated vehicle trajectory is shown in Fig. (9.1) as a dashed line. Position data was created at 1 Hz,

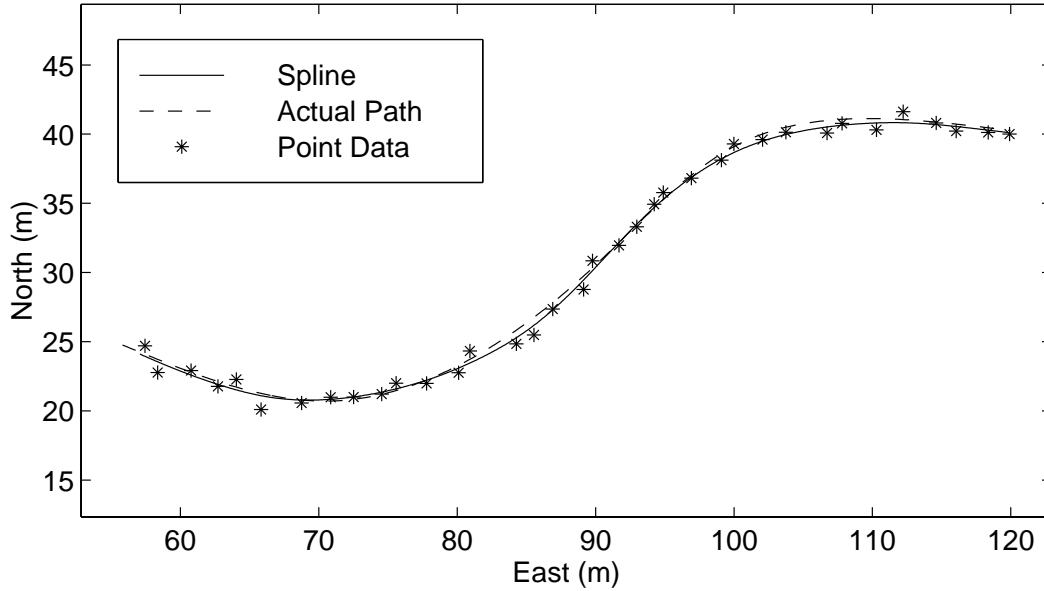


Figure 9.1: Overhead view of spline interpolation of discrete noisy position data.

but was corrupted by measurement noise with a large standard deviation of 50 cm. The interpolated trajectory (solid line) closely approximated the original trajectory in spite of the high measurement noise. Note that the fidelity of the interpolated trajectory increased as the number of points on the trajectory increased.

9.3 Locating the Vehicle on the Curve

The location of the vehicle on the curve had to be determined to calculate tracking error. If the vehicle were tracking the curve perfectly, the distance between the vehicle's control point position r_c and the closest point on the curve r_s was zero. Such a condition rarely existed, and the closest point on the curve minimized $\|r_c - r_s\|^2$.² Specifically, the value of the independent parameter s_s was sought which minimized $\|r_c - r_s\|$. Some empirical knowledge had to be incorporated when solving for s_s . For example, if the vehicle was progressing forward on the curve, the closest point would

²The $\|\cdot\|$ will denote the 2-norm unless otherwise specified.

probably lie forward of the previous point. Further constraints were derived based on the maximum distance the vehicle could travel in the time since the last point was calculated.

The first and second-order necessary conditions on the solution were [Ber95]

$$\frac{d}{ds} [\|r_c - r_s\|^2] = 0 \quad (9.6)$$

$$\frac{d^2}{ds^2} [\|r_c - r_s\|^2] \geq 0 \quad (9.7)$$

Eqn. (9.6) expanded and simplified was:

$$0 = (n_s - n_i) n'_s + (e_s - e_i) e'_s \quad (9.8)$$

where (n_s, e_s) were the north and east components of r_s , (n_i, e_i) was the vehicle's control point position, and the $'$ operator denoted $d(\cdot)/ds$. Several constraints narrowed down solution candidates:

- Roots had to be real.
- Roots had to be positive since s was positive throughout the curve.
- Roots had to lie within the interval s_k to s_{k+1} .

Although there could have been two or more equidistant points within $[s_k, s_{k+1}]$, this situation was never encountered in practice. Logic was incorporated that accepted the most likely solution based on the distance traveled on the curve from the last known point. A situation could have arisen where the vehicle was progressing on the curve and suddenly left the curve entirely. In Fig. (9.2), the solid curved line is the desired trajectory, the series of solid straight lines are normals to the curve, the asterisks are the data points defining the curve, and the U-shaped line is an actual simulated trajectory. If the vehicle is just to the right of region "A", the closest point on the curve is somewhere near "B" since the normal line passing through the vehicle's location will also pass through the closest point on the curve. If the vehicle moves into region "A" because of poor control decisions, the vehicle will suddenly leave the curve entirely! The closest point now lies on a line leading into the start

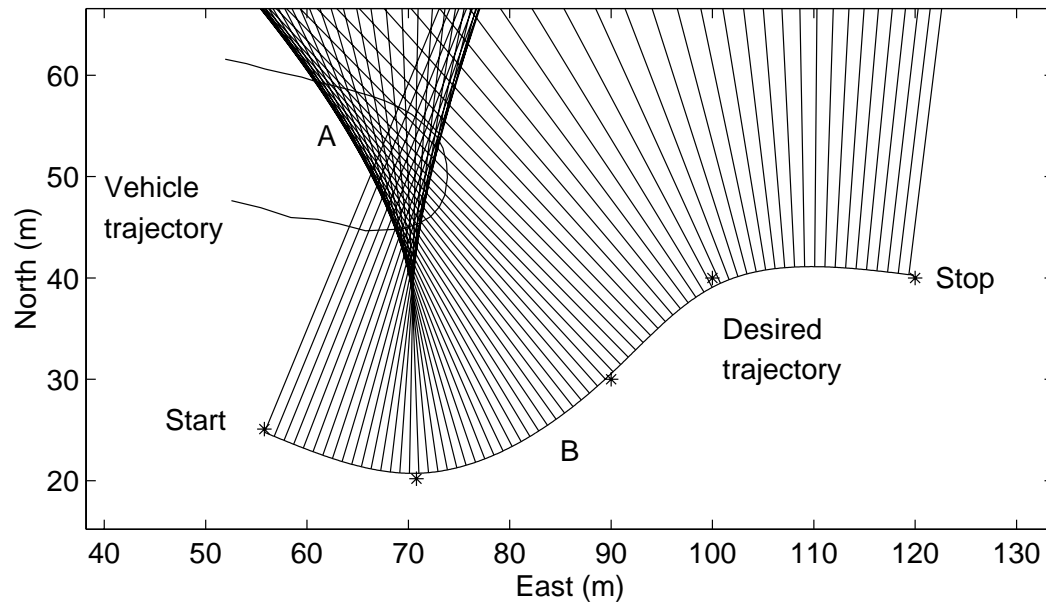


Figure 9.2: Curve with normal lines shown to illustrate a potential locating problem: if the vehicle’s closest point on the curve lies near “B”, but the vehicle moves into region “A”, the vehicle will have left the curve entirely.

of the curve, and the vehicle’s location on the curve is discontinuous. Although this situation was encountered in simulation, it was never encountered in experiment.

The discontinuity of the closest point along the curve can be predicted at every epoch by noting that this discontinuity will occur if the vehicle is traveling on the inside of the curve and has a tracking error larger than the curve’s radius of curvature. There are two possible approaches to this discontinuity problem. In the first approach, the algorithm used to search for the vehicle’s closest point can check to see if the tracking error is greater than the radius of curvature. If so, the algorithm expands its search for the closest point. The second approach involves realizing that since the vehicle drove the original trajectory, no section of the curve should have a radius of curvature below the vehicle’s turning radius. The tractors used in this research had turning radii on the order of 3.5 to 4 m. With an accurate control algorithm and realistic vehicle speeds, the tracking error should remain well below this radius. In row-crop applications, the minimum turn-radius is further limited by the fact that the front wheels should not drive over any crops. Since the front wheels can not leave

the narrow gap between the crops, their maximum angle is relatively shallow and the corresponding turning radius is large.

9.4 Differentiating Curve Parameters

As will be seen in subsequent chapters, time derivatives of various curve parameters were required for some control algorithms. Since any curve parameter $x(s)$ was a function of s only, time-differentiation led to

$$\dot{x}(s) = \frac{\partial x}{\partial s} \frac{ds}{dt} \quad (9.9)$$

$$= x' \dot{s} \quad (9.10)$$

The time rate-of-change of s was broken apart into

$$\dot{s} = \frac{\partial s}{\partial L} \frac{dL}{dt} \quad (9.11)$$

where L represents the curve length. In general [Kre93],

$$\frac{dL}{ds} = \|r'_s(s)\| = \frac{1}{\frac{\partial s}{\partial L}} \quad (9.12)$$

$$= \sqrt{n_s'^2 + e_s'^2} \quad (9.13)$$

The rate of the change of the curve length was the projection of the vehicle's velocity onto the curve tangent. For the control model developed earlier,

$$\dot{L} = V_x \cos(\psi - \psi_s) - l_2 \dot{\psi} \sin(\psi - \psi_s) \quad (9.14)$$

where ψ_s was the angle of the curve's tangent:

$$\tan \psi_s = \frac{e'_s}{n'_s} \quad (9.15)$$

Finally,

$$\dot{x}(s) = \frac{x'(s)}{\sqrt{n_s'^2(s) + e_s'^2(s)}} \left[V_x \cos(\psi - \psi_s(s)) - l_2 \dot{\psi} \sin(\psi - \psi_s(s)) \right] \quad (9.16)$$

This differentiation process was repeated for higher-order derivatives and soon led to lengthy expressions for derivatives third-order and above.

9.5 Predicting Future Vehicle Locations on a Curve

Control algorithms that previewed the curve ahead required predicting N future vehicle locations on the curve. The argument was circular: future state information was needed to calculate the vehicle's location on the curve and subsequently generate the control signal sequence, but the control signal sequence was needed to predict future states. To circumvent this problem, the [necessary] assumption could be made that the yaw deviation from the curve was small, and \dot{L} in Eqn. (9.14) could be approximated by

$$\dot{L} \cong V_x \quad (9.17)$$

Under this assumption, the vehicle traveled $V_x \Delta t$ of the curve every epoch where each epoch lasted Δt seconds. The problem of predicting a future vehicle location therefore involved finding the value of s which satisfied

$$L(s - s_0) = \int_{t_0}^{t_0 + \Delta t} \dot{L} dt = \int_{s_0}^s \|r'_s(s)\| ds \quad (9.18)$$

$$V_x T_s \cong \int_{s_0}^s \|r'_s(s)\| ds \quad (9.19)$$

Eqn. (9.19) had no exact solution, and a numerical solution was required. Note that if the curve could have been parameterized by its own arc length, solving for the N future values of s would have been trivial. Unfortunately, an analytical solution for such a parameterization was either difficult or impossible to find [ST82, HJ80] since the relationship

$$\|r'_s(s)\| = 1 \quad (9.20)$$

had to hold true for all s in each splined interval.

Solving for the independent parameter value giving a particular arc length is a problem encountered in computer graphics. Graphic engineers found that certain problems in animation such as generating smooth motion and warping objects become easier when curves are parameterized by their arc length. The research in this thesis used a fourth-order Runge-Kutta integration technique to integrate Eqn. (9.19); other

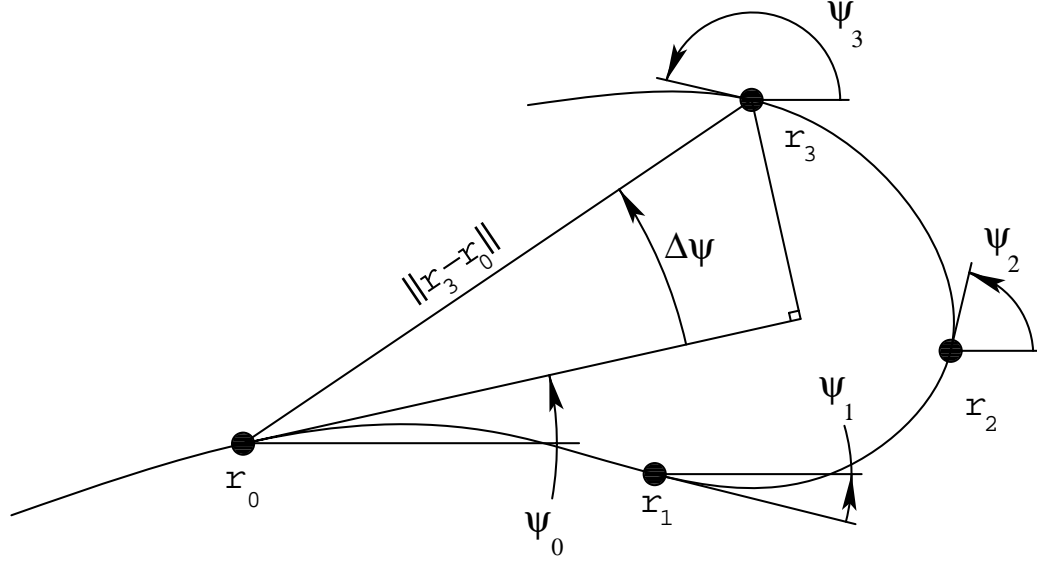


Figure 9.3: Diagram showing current vehicle location (r_0) on a curve and three predicted future locations (r_1 – r_3).

researchers have used the Romberg technique [ST82] as well as Gaussian quadrature [GP90]. In [GP90], researchers tabulated the resulting curve length data and then interpolated to solve for a particular s given an arc length to speed up the computational process.

9.6 Predicting Future Reference State Information

Information about future vehicle states could be calculated once future locations had been predicted. In particular, if a local coordinate frame was constructed tangent to the curve at the vehicle's current position, then the relative position of predicted vehicle locations could be calculated. One of the control algorithms explored required the relative lateral components of predicted vehicle locations. To calculate the relative lateral components, a local coordinate frame was constructed tangent to the curve as shown in Fig. (9.3). The reference lateral deviation three epochs into the future was

$$d_3 = \|r_3 - r_0\| \sin(\Delta\psi) \quad (9.21)$$

where r_0 was the position at the current epoch and r_3 was the vehicle's predicted location on the curve three epochs into the future. Predicted relative yaw angles could also be calculated and used to linearize the equations of motion:

$$\Delta\psi = \tan^{-1} \left[\frac{(r_3 - r_0)_{east}}{(r_3 - r_0)_{north}} \right] - \psi_0 \quad (9.22)$$

CHAPTER 10

Vehicle Control on Curves: Control Algorithms

This chapter presents four possible solutions for curve tracking control: linear control based only on the current error, linear control which incorporated future curve information, nonlinear control based on the current error, and nonlinear control which incorporated future curve information. Each of the four algorithms was developed for the low-order vehicle model in Ch. 3 and could be applied to any wheeled vehicle. Simulation and experimental results are presented in Ch. 11.

As will be shown, a wealth of information about changing curve conditions could be inferred from the cubic spline interpolation. Each of the control algorithms used this information in different ways:

1. The linear local-error control algorithm was based on the idea that curve control may be viewed as control on an arc whose radius was time-varying.
2. The linear look-ahead controller used information about future curve conditions.
3. The nonlinear local-error controller used feedback linearization to linearize the system exactly about the current curve conditions.
4. The nonlinear look-ahead controller posed the control problem as an optimization problem. This last control algorithm is presented, not as a viable option for real-time control (because of its prohibitive computational cost), but to demonstrate how curve information could be input into an optimization routine

to generate a control signal sequence satisfying various nonlinear system constraints such as limits on actuator authority and front-wheel angle. Given the current rate of advancement in computer technology, this last algorithm may be feasible in the near future.

10.1 Linear Local-Error Control

Control based only on local errors would be faster to compute than control incorporating future information since calculating future state information was computationally expensive. Since curve control may be viewed as control on a time-varying arc, the polar analysis developed for arcs was extended to the more general case of a time-varying radius to create a low-cost curve control law based only on local errors in yaw, steering, and tracking.

10.1.1 Vehicle Yaw Error

Vehicle yaw angle was expressed as

$$\psi = \psi_s(s) + \xi(s) + \psi_e(s) \quad (10.1)$$

From Eqn. (7.1), the required vehicle yaw angle was

$$\xi(s) = \mp \sin^{-1} \left[\frac{l_2}{R(s)} \right] \quad (10.2)$$

where $R(s)$ represented the curve's radius of curvature, denoted by R for the remainder of this chapter. Differentiating Eqn. (10.1),

$$\dot{\psi}_e = \dot{\psi} - \dot{\psi}_s - \dot{\xi} \quad (10.3)$$

An expression for $\dot{\psi}$ was given by Eqn. (3.38), and ψ_s and ξ were differentiated according to the chain rule established in Sec. 9.4 for curve-dependent variables:

$$\dot{\psi}_s = \psi'_s \frac{\dot{L}}{\|r'_s\|} \quad (10.4)$$

An expression for ψ'_s could be derived by differentiating $\psi_s(s)$ in Eqn. (9.15) with respect to s :

$$\psi'_s = \frac{n'_s e''_s - n''_s e'_s}{n'^2_s + e'^2_s} \quad (10.5)$$

Similarly, ξ could be differentiated (done here for clockwise rotation):

$$\dot{\xi} = \frac{l_2 \dot{R}}{R \sqrt{R^2 - l_2^2}} \quad (10.6)$$

where the radius of curvature for any point along the curve was given by Eqn. (8.4):

$$R = \frac{[n'^2_s + e'^2_s]^{3/2}}{\sqrt{(n'^2_s + e'^2_s)(n''^2_s + e''^2_s) - (n'_s n''_s + e'_s e''_s)^2}} \quad (10.7)$$

and

$$\dot{R} = R' \frac{\dot{L}}{\|r'_s\|} \quad (10.8)$$

$$R' = \frac{3(n'_s n''_s + e'_s e''_s) \sqrt{n'^2_s + e'^2_s}}{\sqrt{(n'^2_s + e'^2_s)(n''^2_s + e''^2_s) - (n'_s n''_s + e'_s e''_s)^2}} - \frac{[n'^2_s + e'^2_s]^{3/2} \left[\begin{aligned} &(n'_s n''_s + e'_s e''_s)(n''^2_s + e''^2_s) \\ &+ (n'^2_s + e'^2_s)(n''_s n'''_s + e''_s e'''_s) \\ &- (n'_s n''_s + e'_s e''_s) \begin{pmatrix} n''^2_s + e''^2_s \\ + n'_s n'''_s + e'_s e'''_s \end{pmatrix} \end{aligned} \right]}{[(n'^2_s + e'^2_s)(n''^2_s + e''^2_s) - (n'_s n''_s + e'_s e''_s)^2]^{3/2}} \quad (10.9)$$

10.1.2 Vehicle Steer Error

The dynamics of the front-wheel angle error were expressed from Eqn. (7.15) as

$$\dot{\delta}_e = \dot{\delta} - \dot{\delta}_\infty \quad (10.10)$$

$$= u - \dot{\delta}_\infty \quad (10.11)$$

where, from Eqn. (7.13),

$$\delta_\infty = \pm \tan^{-1} \left(\frac{l_1}{K_\delta \sqrt{R^2 - l_2^2}} \right) \quad (10.12)$$

For clockwise vehicle rotation,

$$\dot{\delta}_\infty = \left(\frac{-\frac{l_1}{K_\delta} R \dot{R}}{1 + \frac{l_1^2}{K_\delta^2 (R^2 - l_2^2)}} \right) (R^2 - l_2^2)^{-3/2} \quad (10.13)$$

10.1.3 Vehicle Tracking Error

The dynamics of the vehicle tracking error were

$$r_e = r - R \quad (10.14)$$

$$\dot{r}_e = \dot{r} - \dot{R} \quad (10.15)$$

Eqn. (7.7) specified \dot{r} , and Eqn. (10.8) specified \dot{R} .

10.1.4 Equations of Motion

Consolidating the expressions for the time-rate-of-change of the three state variables ψ_e , δ_e , and r_e , the equations of motion were

$$\begin{aligned} \dot{\psi}_e = & \frac{1}{l_1} V_x K_\delta \tan(\delta_e + \delta_\infty) \\ & - \frac{1}{\|r'_s\|} \psi'_s \left[V_x \cos(\psi_e + \xi) - \frac{l_2}{l_1} V_x K_\delta \tan(\delta_e + \delta_\infty) \sin(\psi_e + \xi) \right] \\ & - \frac{l_2 R' \left[V_x \cos(\psi_e + \xi) - \frac{l_2}{l_1} V_x K_\delta \tan(\delta_e + \delta_\infty) \sin(\psi_e + \xi) \right]}{R^2 \|r'_s\| \sqrt{1 - \left(\frac{l_2}{R}\right)^2}} \end{aligned} \quad (10.16)$$

$$\dot{\delta}_e = u + l_1 R R' \frac{\left[V_x \cos(\psi_e + \xi) - \frac{l_2}{l_1} V_x K_\delta \tan(\delta_e + \delta_\infty) \sin(\psi_e + \xi) \right]}{K_\delta \|r'_s\| [R^2 - l_2^2]^{3/2} \left[1 + \frac{l_1^2}{K_\delta^2 (R^2 - l_2^2)} \right]} \quad (10.17)$$

$$\begin{aligned} \dot{r}_e = & V_x \sin(\psi_e + \xi) + \frac{l_2}{l_1} V_x K_\delta \tan(\delta_e + \delta_\infty) \cos(\psi_e + \xi) \\ & - \frac{1}{\|r'_s\|} R' \left[V_x \cos(\psi_e + \xi) - \frac{l_2}{l_1} V_x K_\delta \tan(\delta_e + \delta_\infty) \sin(\psi_e + \xi) \right] \end{aligned} \quad (10.18)$$

Linearized about zero yaw, steering, and tracking error, Eqns. (10.16–10.18) became

$$\begin{aligned} \dot{\psi}_e \cong & \left[\begin{pmatrix} \frac{1}{\|r'_s\|} \psi'_s \\ + \frac{l_2 R'}{R^2 \|r'_s\| \sqrt{1 - (\frac{l_2}{R})^2}} \end{pmatrix} \begin{pmatrix} V_x \sin \xi \\ + \frac{l_2}{l_1} V_x K_\delta \tan \delta_\infty \cos \xi \end{pmatrix} \right] \psi_e \\ & + \left[\begin{array}{c} \frac{1}{l_1} V_x K_\delta (1 + \tan^2 \delta_\infty) \\ + \frac{1}{\|r'_s\|} \frac{l_2}{l_1} \psi'_s V_x K_\delta \sin \xi (1 + \tan^2 \delta_\infty) \\ + \frac{l_2^2 R' V_x K_\delta (1 + \tan^2 \delta_\infty) \sin \xi}{l_1 \|r'_s\| R^2 \sqrt{1 - (\frac{l_2}{R})^2}} \end{array} \right] \delta_e \end{aligned} \quad (10.19)$$

$$\begin{aligned} \dot{\delta}_e \cong & \left[\frac{-l_1 R R' \left(V_x \sin \xi + \frac{l_2}{l_1} V_x K_\delta \tan \delta_\infty \cos \xi \right)}{K_\delta \|r'_s\| [R^2 - l_2^2]^{3/2} \left(1 + \frac{l_1^2}{K_\delta^2 (R^2 - l_2^2)} \right)} \right] \psi_e \\ & + \left[-\frac{R R' l_2 V_x (1 + \tan^2 \delta_\infty) \sin \xi}{\|r'_s\| [R^2 - l_2^2]^{3/2} \left(1 + \frac{l_1^2}{K_\delta^2 (R^2 - l_2^2)} \right)} \right] \delta_e + u \end{aligned} \quad (10.20)$$

$$\begin{aligned} \dot{r}_e \cong & \left[\begin{array}{c} V_x \cos \xi - \frac{l_2}{l_1} V_x K_\delta \tan \delta_\infty \sin \xi \\ + \frac{1}{\|r'_s\|} R' \left(V_x \sin \xi + \frac{l_2}{l_1} V_x K_\delta \tan \delta_\infty \cos \xi \right) \end{array} \right] \psi_e \\ & + \left[\begin{array}{c} \frac{l_2}{l_1} V_x K_\delta (1 + \tan^2 \delta_\infty) \cos \xi \\ + \frac{1}{\|r'_s\|} \frac{l_2}{l_1} R' V_x K_\delta (1 + \tan^2 \delta_\infty) \sin \xi \end{array} \right] \delta_e \end{aligned} \quad (10.21)$$

Eqns. (10.19–10.21) approximated the dynamics of the vehicle's yaw, steering, and tracking errors on curve trajectories. At every control epoch, local curve information was calculated. The linearized equations of motion were formed, discretized, and then fed into a linear quadratic control algorithm.

10.2 Linear Finite-Horizon Control

A well-known linear control algorithm which offered good performance at a reasonable computational cost was the discrete linear quadratic tracker (“DLQ tracker”) [LS95]. This algorithm, outlined below, used information about future curve conditions when calculating the control signal. Therefore, this algorithm was explored because the

untraveled curve contained information which, if used when computing the control signal, might have allowed the controller to improve tracking.

In [K⁺98], the authors showed how a receding horizon predictive control law similar to the DLQ tracker could be used to guide a bank-to-turn missile to its target. Tomizuka showed in [Tom93] how future reference information may be of less and less value depending on the linearized equations of motion. Once the dependency of the control signal on future information was sufficiently small, future information could be neglected. In [DW97], a quarter-scale car-trailer system was used as a test-bed for path planning, trajectory generation, and real-time control tests. Researchers used a control algorithm based on a feedforward term and a feedback term to track difficult paths despite tight steering angle constraints. As will be shown below, their control algorithm could not be applied here because feedforward control for the vehicle used in this research was time-varying in forward velocity and could not be precomputed. On the test platforms used in this research, forward velocity could range between 0.3 m/s and 10 m/s.

If the system could be linearized and discretized to the form

$$x_{k+1} = \Phi_k x_k + \Gamma_k u_k \quad (10.22)$$

then there existed a unique control signal sequence u_k that minimized the performance index

$$J = \frac{1}{2} \sum_{k=i}^N [(x_k - x_{r,k})^T Q_k (x_k - x_{r,k}) + u_k^T R_k u_k] \quad (10.23)$$

provided the system was controllable, Q_k was symmetric and positive semi-definite, and R_k was symmetric and positive definite. The reference state vector at epoch k was denoted by $x_{r,k}$. Briefly, the Hamiltonian function could be defined as

$$\begin{aligned} H_k = & \frac{1}{2} [(x_k - x_{r,k})^T Q_k (x_k - x_{r,k}) + u_k^T R_k u_k] \\ & + \lambda_{k+1}^T (\Phi_k x_k + \Gamma_k u_k) \end{aligned} \quad (10.24)$$

The state and costate equations were

$$x_{k+1} = \frac{\partial H_k}{\partial \lambda_{k+1}} = \Phi_k x_k + \Gamma_k u_k \quad (10.25)$$

$$\lambda_k = \frac{\partial H_k}{\partial x_k} = Q_k(x_k - x_{r,k}) + \Phi_k^T \lambda_{k+1} \quad (10.26)$$

and the stationarity condition

$$0 = \frac{\partial H_k}{\partial u_k} = R_k u_k + \Gamma_k^T \lambda_{k+1} \quad (10.27)$$

After some lengthy manipulation, the optimal control signal that minimized J was

$$K_k = -(\Gamma_k^T S_{k+1} \Gamma_k + R_{k+1})^{-1} \Gamma_k^T S_{k+1} \Phi_k \quad (10.28)$$

$$b_k = (\Phi_k^T + K_k^T \Gamma_k^T) b_{k+1} - Q_k x_{r,k} \quad (10.29)$$

$$S_k = \Phi_k^T \left[S_{k+1} - S_{k+1} \Gamma_k (\Gamma_k^T S_{k+1} \Gamma_k + R_{k+1})^{-1} \Gamma_k^T S_{k+1} \right] \Phi_k + Q_k \quad (10.30)$$

$$u_k = -(\Gamma_k^T S_{k+1} \Gamma_k + R_{k+1})^{-1} \Gamma_k^T [S_{k+1} \Phi_k x_k + b_{k+1}] \quad (10.31)$$

with boundary conditions

$$S_{N+1} = Q_{N+1} \quad (10.32)$$

$$b_{N+1} = Q_{N+1} x_{r,N+1} \quad (10.33)$$

The interested reader is referred to [LS95, AM91].

This linear algorithm looked ahead on the curve and incorporated information about the changing curve into the control signal. At every epoch, the vehicle's $N+1$ future locations on the curve were predicted, and reference states were calculated.¹ The algorithm began at epoch N and traveled backwards in time to the current epoch. The control signal was then computed using Eqn. (10.31). The N future control signals resulting from the “back-propagation” were ignored, and the process reinitialized at the next control epoch.

10.3 Feedback Linearization

A nonlinear control algorithm based only on local-error and fast enough to run in real-time was feedback linearization. In feedback linearization, the nonlinear system model

¹Note that only the reference states penalized in the state cost matrix Q had to be calculated.

could be linearized exactly and linear control techniques applied [SL91]. Specifically, if the tracking error d as a function of the system's state x was expressed as

$$d = h(x) \quad (10.34)$$

where x was some known vehicle state vector, then $h(x)$ could be differentiated until the control signal appeared. One of the difficulties in feedback linearization is that the “leftover dynamics” associated with a non-minimum phase system are unstable since the right-half-plane zero in the original system becomes an unstable pole in the inverted dynamics. Though approximation techniques exist which can stabilize these unstable dynamics, this research was confined to minimum-phase systems. A wheeled vehicle was non-minimum phase if the control point was located behind the vehicle's pivot point. Therefore, the control point was assumed collocated with the pivot point when using feedback linearization.

Since the low-order model developed for control was order three and the system was minimum phase if the control and pivot points were collocated, $h(x)$ had to be differentiated three times before u would appear in the differentiation. The result was an expression of the form

$$\ddot{d} = f(\psi, \psi_s, r_s, x, y, \delta) + b(\psi, \psi_s, r_s, x, y, \delta)u \quad (10.35)$$

To linearize the system exactly by cancelling the nonlinearities, u was of the form

$$u = \frac{1}{b}(v - f) \quad (10.36)$$

where v was a feedback control signal for the linear system with equations of motion

$$\frac{d}{dt} \begin{bmatrix} d \\ \dot{d} \\ \ddot{d} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} d \\ \dot{d} \\ \ddot{d} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} v \quad (10.37)$$

The feedforward control signal u_{ff} could be calculated by solving $\ddot{d} = 0$.

10.3.1 Feedforward Control Signal

For the model in Eqns.(3.36–3.39), the vehicle's tracking error was

$$d = -\sin \psi_s (n - n_s) + \cos \psi_s (e - e_s) \quad (10.38)$$

Differentiating,

$$\begin{aligned}\dot{d} = & -\dot{\psi}_s \cos \psi_s (n - n_s) - \sin \psi_s (\dot{n} - \dot{n}_s) \\ & - \dot{\psi}_s \sin \psi_s (e - e_s) + \cos \psi_s (\dot{e} - \dot{e}_s)\end{aligned}\quad (10.39)$$

$$\begin{aligned}\ddot{d} = & -\ddot{\psi}_s \cos \psi_s (n - n_s) + \dot{\psi}_s^2 \sin \psi_s (n - n_s) \\ & - 2\dot{\psi}_s \cos \psi_s (\dot{n} - \dot{n}_s) - \sin \psi_s (\ddot{n} - \ddot{n}_s) \\ & - \ddot{\psi}_s \sin \psi_s (e - e_s) - \dot{\psi}_s^2 \cos \psi_s (e - e_s) \\ & - 2\dot{\psi}_s \sin \psi_s (\dot{e} - \dot{e}_s) + \cos \psi_s (\ddot{e} - \ddot{e}_s)\end{aligned}\quad (10.40)$$

$$\begin{aligned}\dddot{d} = & -\dddot{\psi}_s \cos \psi_s (n - n_s) + 3\dot{\psi}_s \ddot{\psi}_s \sin \psi_s (n - n_s) \\ & - 3\ddot{\psi}_s \cos \psi_s (\dot{n} - \dot{n}_s) + \dot{\psi}_s^3 \cos \psi_s (n - n_s) \\ & + 3\dot{\psi}_s^2 \sin \psi_s (\dot{n} - \dot{n}_s) - 3\dot{\psi}_s \cos \psi_s (\ddot{n} - \ddot{n}_s) \\ & - \sin \psi_s (\ddot{n} - \ddot{n}_s) - \ddot{\psi}_s \sin \psi_s (e - e_s) \\ & - 3\dot{\psi}_s \ddot{\psi}_s \cos \psi_s (e - e_s) - 3\ddot{\psi}_s \sin \psi_s (\dot{e} - \dot{e}_s) \\ & + \dot{\psi}_s^3 \sin \psi_s (e - e_s) - 3\dot{\psi}_s^2 \cos \psi_s (\dot{e} - \dot{e}_s) \\ & - 3\dot{\psi}_s \sin \psi_s (\ddot{e} - \ddot{e}_s) + \cos \psi_s (\ddot{e} - \ddot{e}_s)\end{aligned}\quad (10.41)$$

The control signal appeared in the third derivatives of the vehicle's position and the closest curve point, though if the assumption was made that $\dot{L} = V_x \cos(\psi - \psi_s) \cong V_x$, then u appeared only in \ddot{n} and \ddot{e} :

$$\ddot{n} = -\frac{V_x^2 K_\delta}{l_1} \left[\frac{V_x K_\delta}{l_1} \cos \psi \tan^2 \delta + \sin \psi (1 + \tan^2 \delta) u \right] \quad (10.42)$$

$$\ddot{e} = -\frac{V_x^2 K_\delta}{l_1} \left[\frac{V_x K_\delta}{l_1} \sin \psi \tan^2 \delta - \cos \psi (1 + \tan^2 \delta) u \right] \quad (10.43)$$

Other time-derivatives of curve variables were differentiated using the chain rule described in Sec. 9.4. The reference value of d was zero, and Eqn. (10.41) was used to solve for the feedforward control signal.

Eqns. (10.37) and (10.41) represent the feedback linearization algorithm applied to a general minimum-phase wheeled vehicle. Since the linearized system is velocity independent, feedback control gains may be solved for off-line, precluding the need for a possibly expensive real-time solution. The feedforward signal, based only on

local curve information, was continuous. The discrete feedforward signal² would not generally be the same as the continuous signal. However, if the control algorithm were run fast enough, the difference between the continuous and discrete signals would be beneath the noise level of vehicle disturbances and safely could be assumed negligible. This development of the feedback linearization algorithm for wheeled vehicles on curve trajectories is general: the above results can be applied to simpler trajectories such as straight rows by computing the appropriate values and derivatives of the trajectory variables.

10.4 Control as a Nonlinear Optimization Problem

Since the vehicle model and trajectory were nonlinear with bounds on both actuator authority and front-wheel angle, nonlinear optimization techniques could be applied to create a control signal sequence satisfying these constraints. The disadvantages of a nonlinear optimizer were possible divergence and high computational cost. Although simulation revealed that a good initial guess solved the divergence problem, the computational cost was prohibitive with today's affordable computers (Pentium 200 MHz). Nonetheless, the development below illustrates how the splined trajectories could be used in the optimization problem with the hope that increasing computer power will eventually make real-time nonlinear optimization a viable control algorithm. Briefly stated, the optimization problem for N epochs was

$$\min_u J = \frac{1}{2} \sum_{i=1}^N d_i^2 \quad (10.44)$$

$$\text{subject to } \dot{x}(t) = f(x, u) \quad (10.45)$$

$$u_{min} \leq u \leq u_{max} \quad (10.46)$$

$$\delta_{min} \leq \delta \leq \delta_{max} \quad (10.47)$$

²Discrete feedback linearization is a subject of active research in nonlinear control.

Although the equations of motion were nonlinear, the control signal was held constant over every epoch, and the constraint in Eqn. (10.47) could be rewritten in linear form:

$$\delta_{min} \leq \delta_0 + T_s u_0 \leq \delta_{max} \quad (10.48)$$

$$\delta_{min} \leq \delta_0 + T_s(u_0 + u_1) \leq \delta_{max} \quad (10.49)$$

$$\vdots$$

$$\delta_{min} \leq \delta_0 + T_s \sum_{i=0}^{N-1} u_i \leq \delta_{max} \quad (10.50)$$

Note that the optimization problem has been posed using the single-shooting method [G⁺81]. Although the multiple-shooting and collocation methods [G⁺81] are considered more desirable because of their increased stability, the single-shooting method was derived here because the two other methods are simpler variations of the same derivation.

10.4.1 Calculating the Cost Function Jacobian

Although control signal sequence optimization could be accomplished without knowing the Jacobian of the cost function in Eqn. (10.44) with respect to the control signal sequence, the Jacobian dJ/du could speed up the optimization process significantly. The square of the tracking error at any epoch i was

$$d_i^2 = (n_s - n_i)^2 + (e_s - e_i)^2 \quad (10.51)$$

where (n_s, e_s) was the closest point on the curve to the vehicle control point location (n_i, e_i) . Differentiating with respect to u_k with $k < i$ (because future control signals had no effect on previous tracking errors),

$$\frac{d}{du_j} \left[\frac{1}{2} d_i^2 \right] = (n_s - n_i) \left(\frac{dn_s}{du_j} - \frac{dn_i}{du_j} \right) + (e_s - e_i) \left(\frac{de_s}{du_j} - \frac{de_i}{du_j} \right) \quad (10.52)$$

Since the location of the closest point on the curve to the vehicle depended only on the vehicle's position, the change in the closest point on the curve with respect to a change in a control signal was

$$\frac{dn_s}{du_j} = \frac{\partial n_s}{\partial n_i} \frac{dn_i}{du_j} + \frac{\partial n_s}{\partial e_i} \frac{de_i}{du_j} \quad (10.53)$$

$$\frac{de_s}{du_j} = \frac{\partial e_s}{\partial e_i} \frac{de_i}{du_j} + \frac{\partial e_s}{\partial n_i} \frac{dn_i}{du_j} \quad (10.54)$$

The derivatives in Eqns. (10.53–10.54) could be broken down into

$$\frac{\partial n_s}{\partial n_i} = \frac{\partial n_s}{\partial s_s} \left(\frac{\partial n_i}{\partial s_s} \right)^{-1} \quad (10.55)$$

$$\frac{\partial n_s}{\partial e_i} = \frac{\partial n_s}{\partial s_s} \left(\frac{\partial e_i}{\partial s_s} \right)^{-1} \quad (10.56)$$

$$\frac{\partial e_s}{\partial n_i} = \frac{\partial e_s}{\partial s_s} \left(\frac{\partial n_i}{\partial s_s} \right)^{-1} \quad (10.57)$$

$$\frac{\partial e_s}{\partial e_i} = \frac{\partial e_s}{\partial s_s} \left(\frac{\partial e_i}{\partial s_s} \right)^{-1} \quad (10.58)$$

where s_s was the value of s at the closest point on the curve. Expressions for $\partial n_s/\partial s_s$ and $\partial e_s/\partial s_s$ came from differentiating Eqns. (9.4) and (9.5). Expressions for $\partial s_s/\partial n_i$ and $\partial s_s/\partial e_i$ could be derived from the first-order necessary conditions for the location of the closest point by solving for n_i or e_i in Eqn. (9.8) and differentiating with respect to s_s so that

$$\frac{\partial n_i}{\partial s_s} = (e_s - e_i) \left[\frac{\frac{\partial^2 e_s}{\partial s_s^2}}{\frac{\partial n_s}{\partial s_s}} - \frac{\frac{\partial e_s}{\partial s_s} \frac{\partial^2 n_s}{\partial s_s^2}}{\left(\frac{\partial n_s}{\partial s_s} \right)^2} \right] + \frac{\left(\frac{\partial e_s}{\partial s_s} \right)^2}{\frac{\partial n_s}{\partial s_s}} + \frac{\partial n_s}{\partial s_s} \quad (10.59)$$

$$\frac{\partial e_i}{\partial s_s} = (n_s - n_i) \left[\frac{\frac{\partial^2 n_s}{\partial s_s^2}}{\frac{\partial e_s}{\partial s_s}} - \frac{\frac{\partial n_s}{\partial s_s} \frac{\partial^2 e_s}{\partial s_s^2}}{\left(\frac{\partial e_s}{\partial s_s} \right)^2} \right] + \frac{\left(\frac{\partial n_s}{\partial s_s} \right)^2}{\frac{\partial e_s}{\partial s_s}} + \frac{\partial e_s}{\partial s_s} \quad (10.60)$$

Eqns. (10.55–10.58) become

$$\frac{\partial n_s}{\partial n_i} = \frac{\left(\frac{\partial n_s}{\partial s_s} \right)^3}{(e_s - e_i) \left(\frac{\partial n_s}{\partial s_s} \frac{\partial^2 e_s}{\partial s_s^2} - \frac{\partial e_s}{\partial s_s} \frac{\partial^2 n_s}{\partial s_s^2} \right) + \left(\frac{\partial e_s}{\partial s_s} \right)^2 \frac{\partial n_s}{\partial s_s} + \left(\frac{\partial n_s}{\partial s_s} \right)^3} \quad (10.61)$$

$$\frac{\partial e_s}{\partial n_i} = \frac{\left(\frac{\partial n_s}{\partial s_s} \right)^2 \frac{\partial e_s}{\partial s_s}}{(e_s - e_i) \left(\frac{\partial n_s}{\partial s_s} \frac{\partial^2 e_s}{\partial s_s^2} - \frac{\partial e_s}{\partial s_s} \frac{\partial^2 n_s}{\partial s_s^2} \right) + \left(\frac{\partial e_s}{\partial s_s} \right)^2 \frac{\partial n_s}{\partial s_s} + \left(\frac{\partial n_s}{\partial s_s} \right)^3} \quad (10.62)$$

$$\frac{\partial e_s}{\partial n_i} = \frac{\frac{\partial n_s}{\partial s_s} \left(\frac{\partial e_s}{\partial s_s} \right)^2}{(n_s - n_i) \left(\frac{\partial e_s}{\partial s_s} \frac{\partial^2 n_s}{\partial s_s^2} - \frac{\partial n_s}{\partial s_s} \frac{\partial^2 e_s}{\partial s_s^2} \right) + \left(\frac{\partial n_s}{\partial s_s} \right)^2 \frac{\partial e_s}{\partial s_s} + \left(\frac{\partial e_s}{\partial s_s} \right)^3} \quad (10.63)$$

$$\frac{\partial e_s}{\partial e_i} = \frac{\left(\frac{\partial e_s}{\partial s_s} \right)^3}{(n_s - n_i) \left(\frac{\partial e_s}{\partial s_s} \frac{\partial^2 n_s}{\partial s_s^2} - \frac{\partial n_s}{\partial s_s} \frac{\partial^2 e_s}{\partial s_s^2} \right) + \left(\frac{\partial n_s}{\partial s_s} \right)^2 \frac{\partial e_s}{\partial s_s} + \left(\frac{\partial e_s}{\partial s_s} \right)^3} \quad (10.64)$$

If $\partial n_s / \partial s_s$ and $\partial e_s / \partial s_s$ were zero (and hence $\partial^2 e_s / \partial s_s^2$ and $\partial^2 e_s / \partial s_s^2$ were zero as well), Eqns. (10.61–10.64) would be singular. However, though $\partial n_s / \partial s_s$ or $\partial e_s / \partial s_s$ could be zero, both could not be zero at the same point on the curve. By Eqn. (9.13),

$$\frac{dL}{ds} = \sqrt{\left(\frac{\partial n_s}{\partial s_s} \right)^2 + \left(\frac{\partial e_s}{\partial s_s} \right)^2} \quad (10.65)$$

where L is the curve length. If both $\partial n_s / \partial s_s$ and $\partial e_s / \partial s_s$ were zero, then the curve length does not change with s . This implies at least two values of s specify the same point on the curve, a violation of the spline specification.

Values for dn_i / du_j and de_i / du_j could be extracted from the derivative of the state vector with respect to the control input dx_i / du_j . Though this derivative could not be calculated exactly, an approximation for dx_i / du_j could be derived from the standard fourth-order Runge-Kutta integration scheme [Kre93]

$$k_1 = T_s f(t_{i-1}, x_{i-1}) \quad (10.66)$$

$$\tilde{x}_2 = x_{i-1} + \frac{1}{2} k_1 \quad (10.67)$$

$$k_2 = T_s f\left(t_{i-1} + \frac{1}{2} T_s, \tilde{x}_2\right) \quad (10.68)$$

$$\tilde{x}_3 = x_{i-1} + \frac{1}{2} k_2 \quad (10.69)$$

$$k_3 = T_s f\left(t_{i-1} + \frac{1}{2} T_s, \tilde{x}_3\right) \quad (10.70)$$

$$\tilde{x}_4 = x_{i-1} + k_3 \quad (10.71)$$

$$k_4 = T_s f(t_{i-1} + T_s, \tilde{x}_4) \quad (10.72)$$

$$x_i = x_{i-1} + \frac{1}{6} (k_1 + 2k_2 + 2k_3 + k_4) \quad (10.73)$$

which lead to

$$\frac{dx_i}{du_j} = \frac{\partial x_i}{\partial u_j} + \frac{\partial x_i}{\partial x_{i-1}} \frac{dx_{i-1}}{du_j} \quad (10.74)$$

If $j = i - 1$, then u_j was in effect over the integration interval $i - 1$ to i and had no effect on states prior to epoch $i - 1$ including x_{i-1} . In this case, Eqn. (10.74) reduced to

$$\frac{dx_i}{du_j} = \frac{\partial x_i}{\partial u_j} \quad (10.75)$$

$$\begin{aligned} &\cong \frac{1}{6} \frac{\partial k_1}{\partial u_j} + \frac{1}{3} \frac{\partial k_2}{\partial \tilde{x}_2} \left(\frac{\partial \tilde{x}_2}{\partial u_j} + \frac{1}{2} \frac{\partial k_1}{\partial u_j} \right) \\ &\quad + \frac{1}{3} \frac{\partial k_3}{\partial \tilde{x}_3} \left(\frac{\partial \tilde{x}_3}{\partial u_j} + \frac{1}{2} \frac{\partial k_2}{\partial u_j} \right) + \frac{1}{6} \frac{\partial k_4}{\partial \tilde{x}_4} \left(\frac{\partial \tilde{x}_4}{\partial u_j} + \frac{\partial k_3}{\partial u_j} \right) \end{aligned} \quad (10.76)$$

$$\cong \frac{1}{6} \left(\frac{\partial k_1}{\partial u_j} + \frac{\partial k_2}{\partial \tilde{x}_2} \frac{\partial k_1}{\partial u_j} + \frac{\partial k_3}{\partial \tilde{x}_3} \frac{\partial k_2}{\partial u_j} + \frac{\partial k_4}{\partial \tilde{x}_4} \frac{\partial k_3}{\partial u_j} \right) \quad (10.77)$$

Otherwise, the effect of u_j on the previous state vector was propagated forward with

$$\begin{aligned} \frac{\partial x_i}{\partial x_{i-1}} &= \frac{\partial x_{i-1}}{\partial x_{i-1}} + \frac{1}{6} \frac{\partial k_1}{\partial x_{i-1}} + \frac{1}{3} \frac{\partial k_2}{\partial \tilde{x}_2} \left(\frac{\partial \tilde{x}_2}{\partial x_{i-1}} + \frac{\partial \tilde{x}_2}{\partial k_1} \frac{\partial k_1}{\partial x_{i-1}} \right) \\ &\quad + \frac{1}{3} \frac{\partial k_3}{\partial \tilde{x}_3} \left(\frac{\partial \tilde{x}_3}{\partial x_{i-1}} + \frac{\partial \tilde{x}_3}{\partial k_2} \frac{\partial k_2}{\partial x_{i-1}} \right) \\ &\quad + \frac{1}{6} \frac{\partial k_4}{\partial \tilde{x}_4} \left(\frac{\partial \tilde{x}_4}{\partial x_{i-1}} + \frac{\partial \tilde{x}_4}{\partial k_3} \frac{\partial k_3}{\partial x_{i-1}} \right) \end{aligned} \quad (10.78)$$

$$\begin{aligned} &= I + \frac{T_s}{6} \nabla_x f(x_{i-1}) + \frac{T_s}{3} \nabla_x f(\tilde{x}_2) \left[I + \frac{T_s}{2} \nabla_x f(x_{i-1}) \right] \\ &\quad + \frac{T_s}{3} \nabla_x f(\tilde{x}_3) \left[I + \frac{T_s}{2} \nabla_x f(x_{i-1}) \right] \\ &\quad + \frac{T_s}{6} \nabla_x f(\tilde{x}_4) [I + T_s \nabla_x f(x_{i-1})] \end{aligned} \quad (10.79)$$

Eqn. (10.74) was propagated forward through each state vector at every epoch for a particular u_j . Note that time-dependency was omitted in Eqn. (10.79) since the equations of motion for the vehicle model used in this research were not a function of time.

Eqn. (10.52) then became

$$\frac{d}{du_j} \left[\frac{1}{2} d_i^2 \right] = (n_s - n_i) \left(\frac{\partial n_s}{\partial n_i} \frac{dn_i}{du_j} + \frac{\partial n_s}{\partial e_i} \frac{de_i}{du_j} - \frac{dn_i}{du_j} \right) \quad (10.80)$$

$$+ (e_s - e_i) \left(\frac{\partial e_s}{\partial e_i} \frac{de_i}{du_j} + \frac{\partial e_s}{\partial n_i} \frac{dn_i}{du_j} - \frac{de_i}{du_j} \right) \quad (10.81)$$

Eqn. (10.81) could be used in Eqn. (10.44) to solve for dJ/du .

10.4.2 Calculating the Cost Function Hessian

In a similar manner, the cost function Hessian could be calculated by differentiating Eqns.(10.81), (10.77), and (10.79) again. Although evaluating the Hessian may be too expensive during the optimization process, the Hessian could offer insight into the nature of the optimization problem.

$$\frac{dJ_i}{du_j du_k} = \frac{d}{du_j du_k} \left[\frac{1}{2} d_i^2 \right] \quad (10.82)$$

$$\begin{aligned} &= (n_s - n_i) \left(\frac{d^2 n_s}{du_j du_k} - \frac{d^2 n_i}{du_j du_k} \right) \\ &\quad + \left(\frac{dn_s}{du_j} - \frac{dn_i}{du_j} \right) \left(\frac{dn_s}{du_k} - \frac{dn_i}{du_k} \right) \\ &\quad (e_s - e_i) \left(\frac{d^2 e_s}{du_j du_k} - \frac{d^2 e_i}{du_j du_k} \right) \\ &\quad + \left(\frac{de_s}{du_j} - \frac{de_i}{du_j} \right) \left(\frac{de_s}{du_k} - \frac{de_i}{du_k} \right) \end{aligned} \quad (10.83)$$

where

$$\begin{aligned} \frac{d^2 n_s}{du_j du_k} &= \frac{\partial^2 n_s}{\partial n_i^2} \frac{dn_i}{du_j} \frac{dn_i}{du_k} + \frac{\partial^2 n_s}{\partial n_i \partial e_i} \frac{dn_i}{du_j} \frac{de_i}{du_k} + \frac{\partial n_s}{\partial n_i} \frac{d^2 n_i}{du_j du_k} \\ &\quad + \frac{\partial^2 n_s}{\partial n_i \partial e_i} \frac{dn_i}{du_k} \frac{de_i}{du_j} + \frac{\partial^2 n_s}{\partial e_i^2} \frac{de_i}{du_j} \frac{de_i}{du_k} + \frac{\partial n_s}{\partial e_i} \frac{d^2 e_i}{du_j du_k} \end{aligned} \quad (10.84)$$

$$\begin{aligned} \frac{d^2 e_s}{du_j du_k} &= \frac{\partial^2 e_s}{\partial n_i^2} \frac{dn_i}{du_j} \frac{dn_i}{du_k} + \frac{\partial^2 e_s}{\partial n_i \partial e_i} \frac{dn_i}{du_j} \frac{de_i}{du_k} + \frac{\partial e_s}{\partial n_i} \frac{d^2 n_i}{du_j du_k} \\ &\quad + \frac{\partial^2 e_s}{\partial n_i \partial e_i} \frac{dn_i}{du_k} \frac{de_i}{du_j} + \frac{\partial^2 e_s}{\partial e_i^2} \frac{de_i}{du_j} \frac{de_i}{du_k} + \frac{\partial e_s}{\partial e_i} \frac{d^2 e_i}{du_j du_k} \end{aligned} \quad (10.85)$$

The second-order derivatives of the closest north point on the curve were

$$\frac{\partial^2 n_s}{\partial n_i^2} = \frac{\partial}{\partial n_i} \left[\frac{\partial n_s}{\partial n_i} \right] + \frac{\partial^2 n_s}{\partial s_s^2} \left(\frac{\partial n_i}{\partial s_s} \right)^{-2} - \frac{\partial n_s}{\partial s_s} \frac{\partial^2 n_i}{\partial s_s^2} \left(\frac{\partial n_i}{\partial s_s} \right)^{-3} \quad (10.86)$$

$$\begin{aligned} \frac{\partial^2 n_s}{\partial n_i \partial e_i} &= \frac{\partial}{\partial e_i} \left[\frac{\partial n_s}{\partial n_i} \right] + \frac{\partial^2 n_s}{\partial s_s^2} \left(\frac{\partial n_i}{\partial s_s} \right)^{-1} \left(\frac{\partial e_i}{\partial s_s} \right)^{-1} \\ &\quad - \frac{\partial n_s}{\partial s_s} \frac{\partial^2 n_i}{\partial s_s^2} \left(\frac{\partial n_i}{\partial s_s} \right)^{-2} \left(\frac{\partial e_i}{\partial s_s} \right)^{-1} \end{aligned} \quad (10.87)$$

$$\frac{\partial^2 n_s}{\partial e_i^2} = \frac{\partial}{\partial e_i} \left[\frac{\partial n_s}{\partial e_i} \right] + \frac{\partial^2 n_s}{\partial s_s^2} \left(\frac{\partial e_i}{\partial s_s} \right)^{-2} - \frac{\partial n_s}{\partial s_s} \frac{\partial^2 e_i}{\partial s_s^2} \left(\frac{\partial e_i}{\partial s_s} \right)^{-3} \quad (10.88)$$

The second-order derivatives of the closest east point on the curve were similar. Expressions for $\partial^2 n_s / \partial s_s^2$ and $\partial^2 e_s / \partial s_s^2$ could be found by differentiating Eqns. (9.4) and (9.5) twice, and expressions for $\partial^2 n_i / \partial s_s^2$ and $\partial^2 e_i / \partial s_s^2$ could be found by differentiating Eqns.(10.59) and (10.60) again. Eqn. (10.74) expanded to second-order was

$$\frac{d^2 x_i}{du_j du_k} = \left(\frac{\partial^2 x_i}{\partial x_{i-1}^2} \frac{dx_{i-1}}{du_j} \right)^T \frac{dx_{i-1}}{du_k} + \frac{\partial x_i}{\partial x_{i-1}} \frac{d^2 x_{i-1}}{du_j du_k} \quad (10.89)$$

where the second-order expansion of the fourth-order Runge-Kutta method could be found by differentiating Eqn. (10.79) again.

Once the cost function's Jacobian and Hessian were calculated for a predetermined curve window, optimization software could attempt to calculate a control signal sequence that minimized the sum of the squares of the tracking error while satisfying the vehicle's linear and nonlinear constraints. Any one of the previous control algorithms could generate a control signal sequence as an initial solution guess. This fourth and final control algorithm was not implemented in experiment because of its computational cost, but served as a benchmark in simulation for comparing tracking performance with other algorithms.

CHAPTER 11

Vehicle Control on Curves: Simulation and Experimental Results

To compare control algorithms, a gently varying curve typical of what might be expected in realistic agricultural applications was selected. Each controller controlled the vehicle in a noise- and disturbance-free environment so that the tracking results might be compared. In all simulations and experiments, the control point was a point on the ground directly beneath the center of the vehicle's rear axle.

11.1 Simulation Results

The curve control algorithms in Ch. 10 were compared in simulation. The vehicle state was assumed to be known exactly, and no vehicle disturbances were created. Forward velocity was set to 1.0 m/s and the sample rate to 5 Hz. Fig. (11.1) shows an overhead view of the trajectory. The vehicle was allowed to start on the curve at the left, aligned directly on the curve with the wheels pointed straight ahead. The DLQ tracker controller was allowed to look ahead fifteen epochs. In the optimized solution, a linesearch algorithm using safe-guarded cubic approximations was used to calculate the control signal sequence [G⁺81]. The optimization software NPSOL [G⁺86], which was allowed to look ahead ten epochs with an initial guess of zero control, required

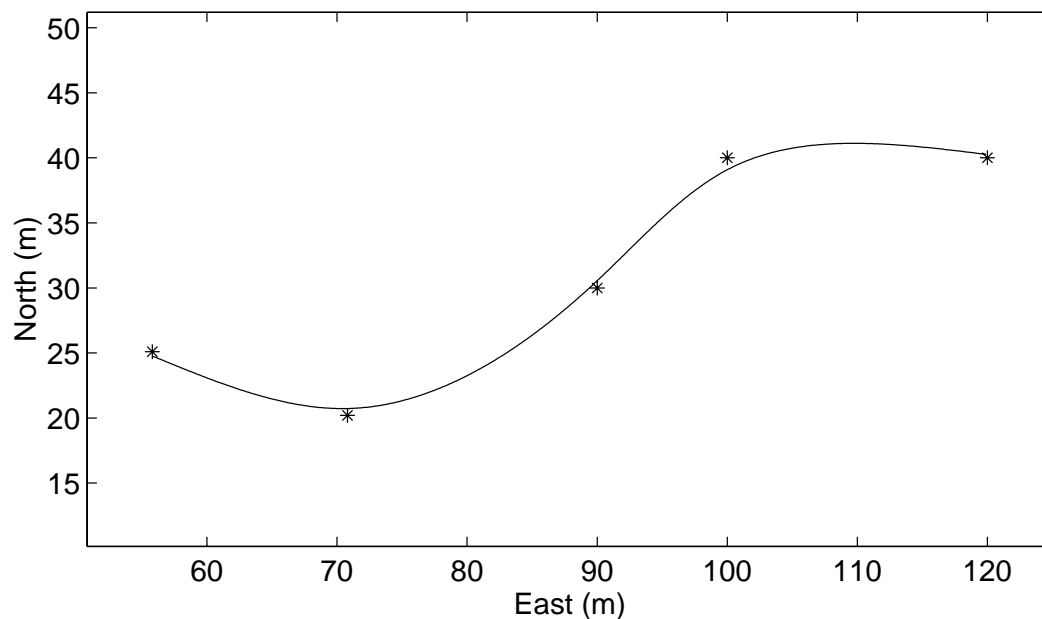


Figure 11.1: Curved trajectory used for comparing control algorithms. Asterisks (*) mark the data points used to create the curve.

| Control Law | \bar{d} (cm) | one- σ (cm) |
|---------------------|------------------------|-----------------------|
| <i>Local-error</i> | -1.06×10^{-2} | 0.655 |
| <i>DLQ tracker</i> | -0.457 | 4.11 |
| <i>Feedback lin</i> | 6.83×10^{-4} | 1.67×10^{-2} |
| <i>Optimized</i> | 1.03×10^{-6} | 1.14×10^{-5} |

Table 11.1: Summary of tracking error history for local-error, DLQ tracker, feedback linearization, and optimized control algorithms.

two hours on a Sun Ultra60 workstation.

Fig. (11.2) shows a history of the tracking errors for the local-error, DLQ tracker, feedback linearization, and optimized control algorithms. Tbl. 11.1 summarizes the results. The DLQ tracker performed the worst, though if the tracking error penalty in the quadratic cost function was increased above actual values used in experiment, the tracking error decreased significantly.

One interesting side effect of interpolating trajectories with cubic splines was that the optimal control signal was discontinuous because of the discontinuity in the splines' third derivatives at spline junctions. Enforcing continuity to third order

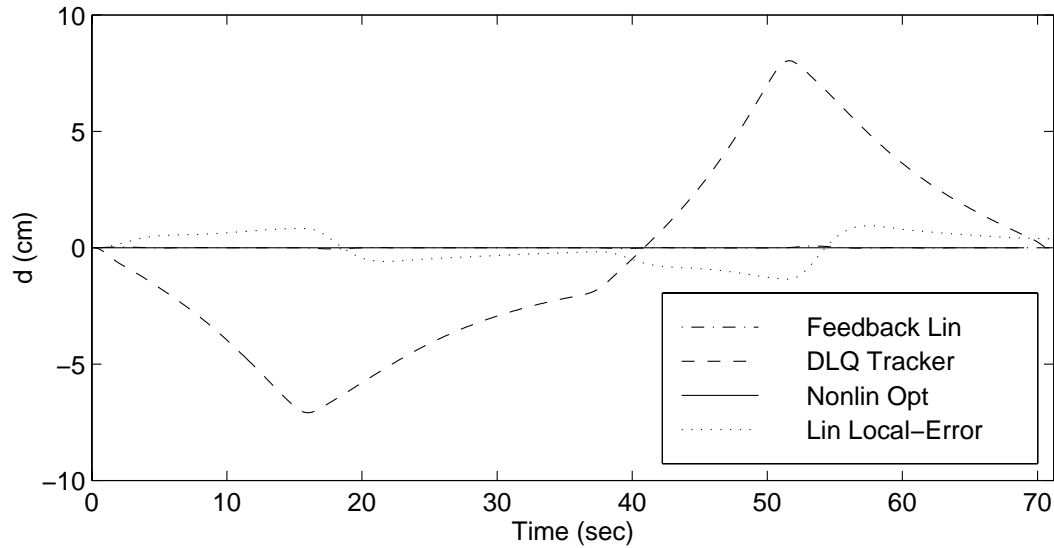


Figure 11.2: Curve tracking error history for local-error, DLQ tracker, feedback linearization, and optimized control algorithms. Note that feedback linearization and nonlinear optimization results are close enough to zero to be virtually indistinguishable.

could not be done without creating more degrees of freedom through higher-order splines. Fig. (11.3) shows the optimized control signal as a function of time. Note the four discontinuous junctions that resulted from the four splined intervals.

11.2 Experimental Results

As shown in Fig. (11.4), any variety of curve is possible given discrete position data. The leaping deer logo was copied from a Deere and Co. maintenance manual, enlarged, then scanned in. A computer-aided design program was used to pick 169 points along the perimeter, which were then scaled and rotated to fit the test field. The resulting actual vehicle position data is plotted on the North-East plane from the December 1997 test. The slight overlap in the back of the front hoof is the starting point of the curve, which took approximately 20 minutes to complete without an implement.

Fig. (11.5) shows a more realistic application of tractor control along curves. The northernmost curve was the original curve and was identical to the curve shown in

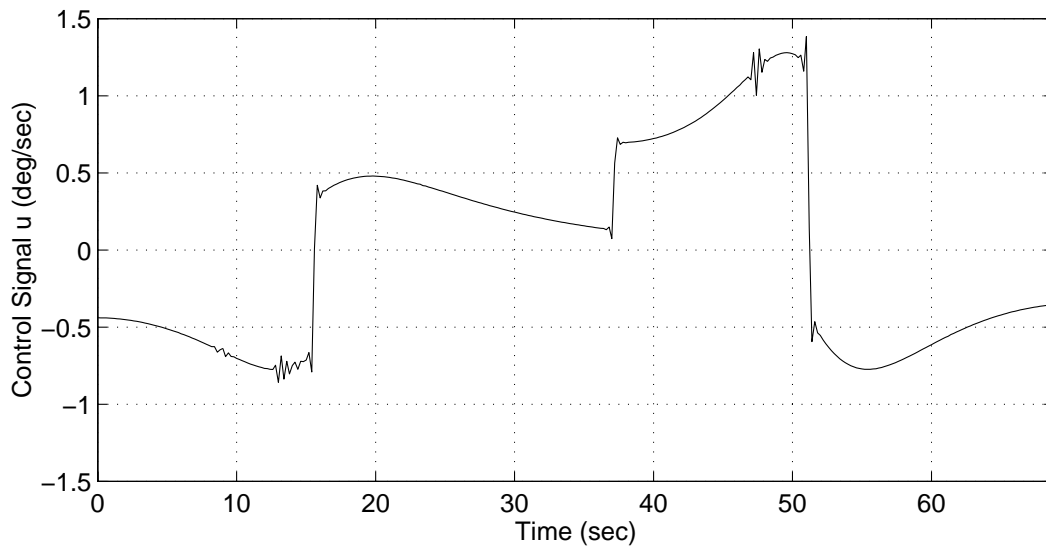


Figure 11.3: Control signal history for optimized control algorithm.

Fig. (11.1). Three curves, offset at 3.0 m, were automatically generated prior to starting real-time control. The vehicle began at the position marked with an asterisk and took seven minutes to complete the entire trajectory. A small cultivator was hitched to the tractor, and all U-turns and curve acquisition was done automatically. All three control algorithms were compared along the same trajectory. Straight lines were placed leading into the start of each curve to minimize initial transient errors.

11.2.1 Linear Local-Error Control

Fig. (11.6) shows the time history of the estimated tracking errors for all curves on one plot. The linear-local error control algorithm had a low computational cost compared to the DLQ tracking algorithm. Bias states in yaw and steering were estimated to compensate for modeling and calibration errors.

11.2.2 DLQ Tracker Control

Fig. (11.7) shows the estimated tracking errors for the DLQ tracker control algorithm.

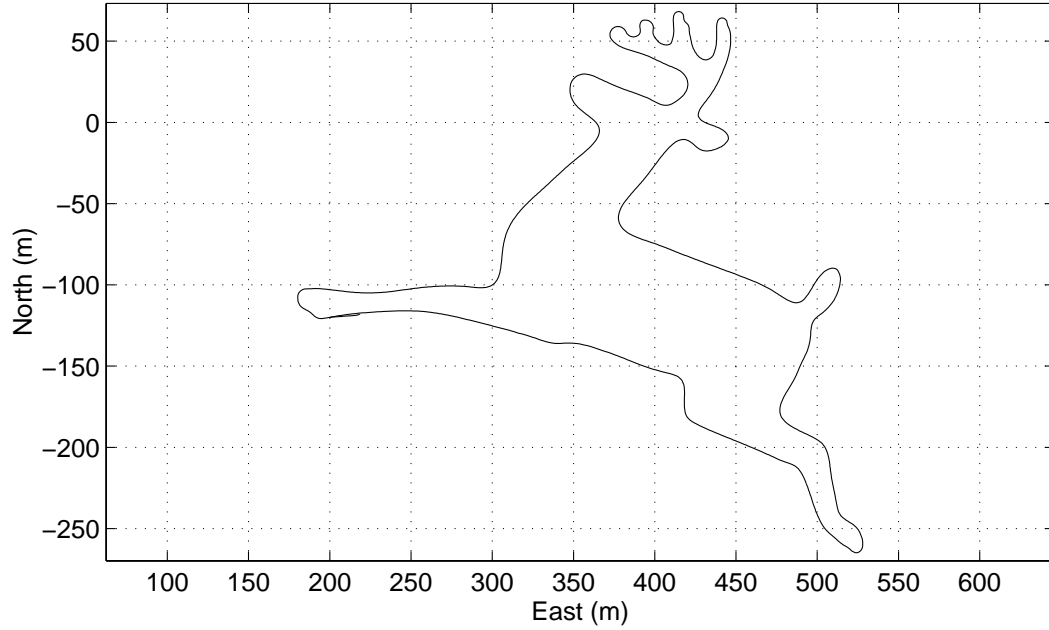


Figure 11.4: Overhead view of actual position data for a not-so-typical vehicle trajectory completed in December 1997. Coordinates are relative to the reference station antenna.

The controller's computational cost was relatively high, and the number of look-ahead intervals was constrained to be less than 40 when controlling at 5 Hz. Bias estimation was used to reduce mean tracking errors.

11.2.3 Feedback Linearization Control

Fig. (11.8) shows the estimated tracking errors for the feedback linearization control algorithm. Like the linear-local error controller, the feedback linearization controller had a relatively low computational cost. Control on the integral of the tracking error was added to the control algorithm.

11.2.4 Summary of Results

Tbl. 11.2 summarizes the average estimated tracking errors. The local-error controller performed the worst, most likely because the radius of curvature along the curves varied rapidly. Both the local-error and DLQ tracker controllers tended to swing

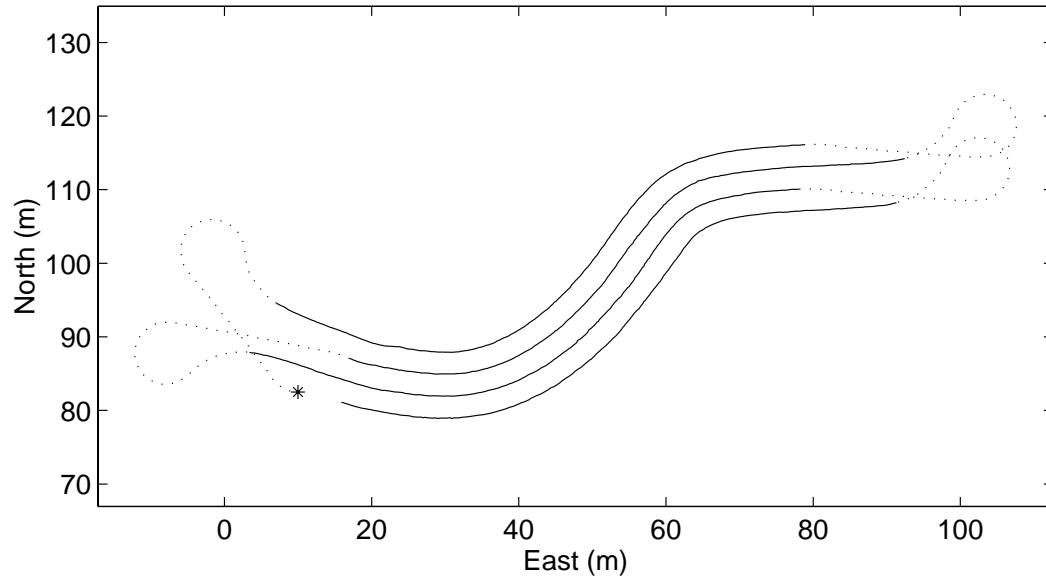


Figure 11.5: Overhead view of actual position data for four curves completed in April 1999 in California's San Joaquin Valley. Coordinates are relative to the reference station antenna.

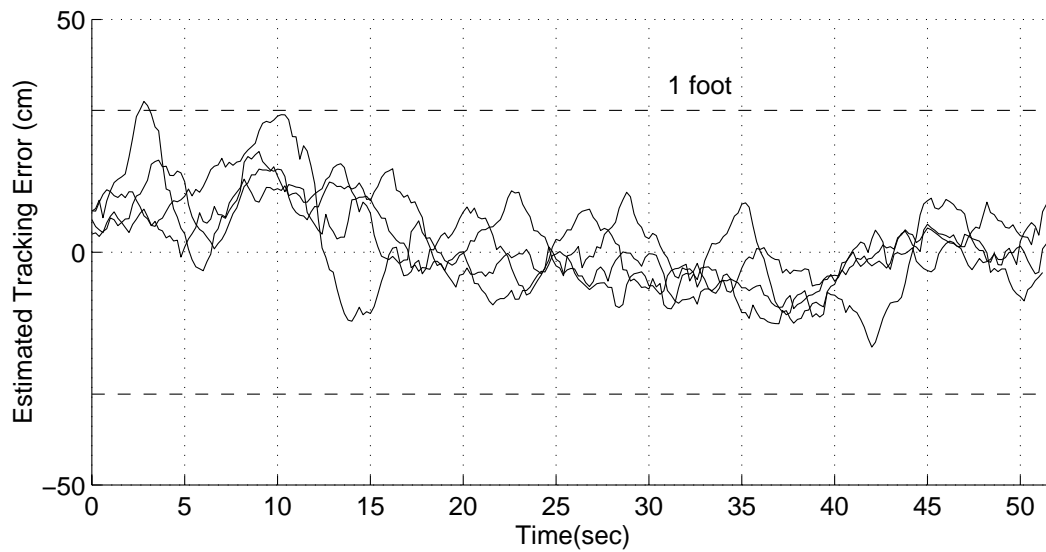


Figure 11.6: Experimental curve tracking error using linear local-error controller for curves shown in Fig. (11.5). Mean tracking error was 1.99 cm with a standard deviation of 9.07 cm.

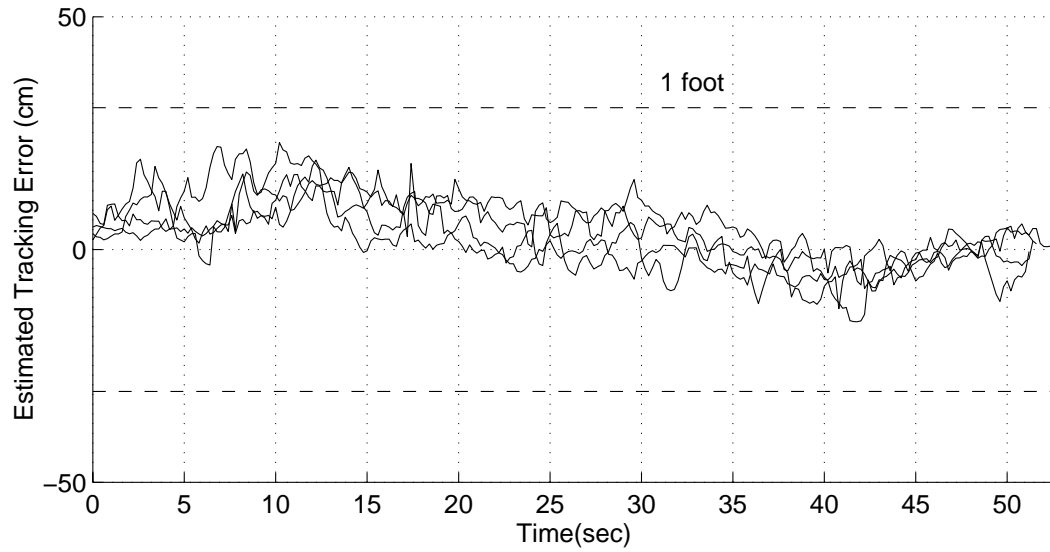


Figure 11.7: Experimental curve tracking error using DLQ tracker controller for curves shown in Fig. (11.5). Mean tracking error was 3.44 cm with a standard deviation of 6.69 cm.

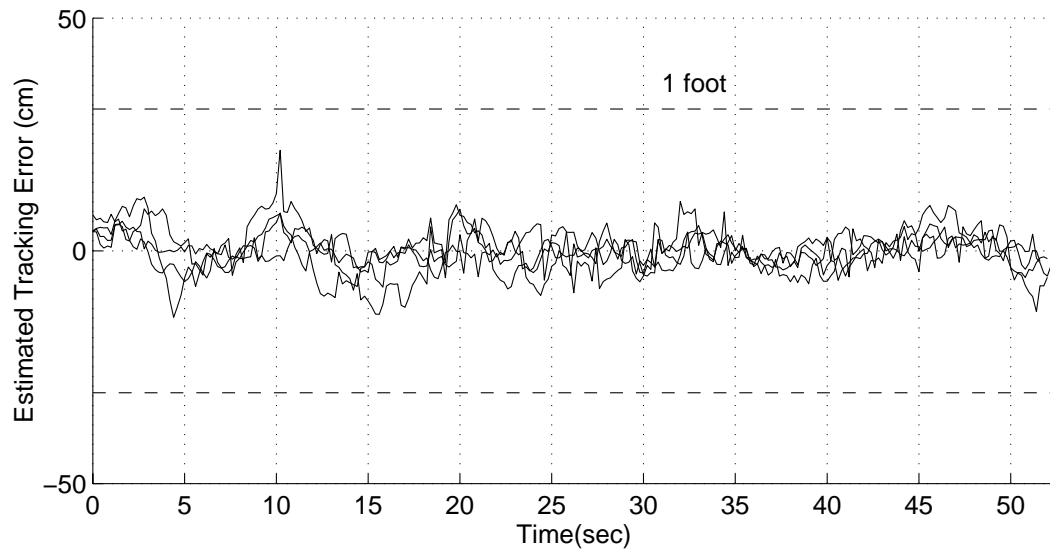


Figure 11.8: Experimental curve tracking error using feedback linearization controller for curves shown in Fig. (11.5). Mean tracking error was -0.01 cm with a standard deviation of 4.12 cm.

| Control Law | \bar{d} (cm) | one- σ (cm) | V_x (m/s) |
|---------------------|----------------|--------------------|-------------|
| <i>Local-error</i> | 1.99 | 9.07 | 1.42 |
| <i>DLQ tracker</i> | 3.44 | 6.69 | 1.42 |
| <i>Feedback lin</i> | -0.01 | 4.12 | 1.40 |

Table 11.2: Experimental tracking performance for curves shown in Fig. (11.5). Tracking error values listed are the average values over all four curves.

| Control Law | u_n |
|---------------------|-------|
| <i>Local-error</i> | 1.94 |
| <i>DLQ tracker</i> | 2.97 |
| <i>Feedback lin</i> | 3.41 |

Table 11.3: Experimental tracking control effort for curves shown in Fig. (11.5). Units are deg/sec/epoch.

towards the outside of the curves, possibly because of modeling errors. The feedback linearization controller demonstrated the best performance.

To compare the amount of control effort each controller used, the sum of the norms for each of the four curves was calculated. Each norm was normalized by the number of control epochs. Mathematically, this sum of the control norms u_n can be expressed as

$$u_n = \frac{1}{N_1} \sqrt{\sum_{i=1}^{N_1} u_i^2} + \cdots + \frac{1}{N_4} \sqrt{\sum_{i=1}^{N_4} u_i^2} \quad (11.1)$$

where N_1 – N_4 represent the number of control epochs for curves one through four. Tbl. 11.3 shows the normalized control effort for each control algorithm. Figs. (11.9–11.11) show the time histories of the control effort along all four trajectories for each of the control algorithms. The feedback linearization controller used the most control effort, yet offered the best performance. Though the linear local-error controller used the least control effort, the tracking error cost could not be increased without destabilizing the system.

The linear-local error controller was computationally the cheapest control algorithm as well as the simplest algorithm to implement. However, this controller performed relatively poorly in experiment. Although it used less control effort than the other two controllers, control gains could not be increased without destabilizing the

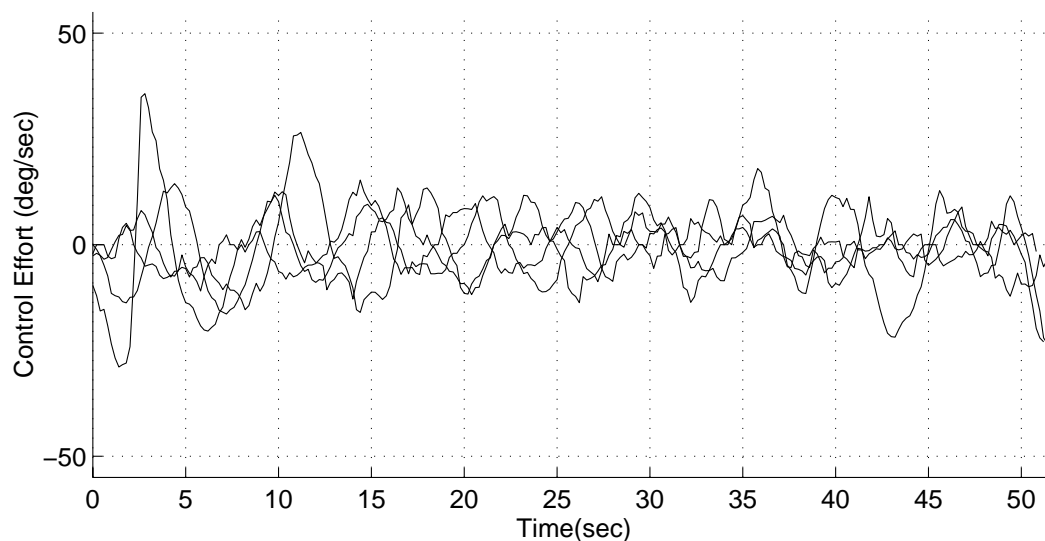


Figure 11.9: Experimental curve tracking control effort using linear local-error controller for curves shown in Fig. (11.5). The sum of the control effort norms for all four curves was 1.94 deg/sec/epoch.

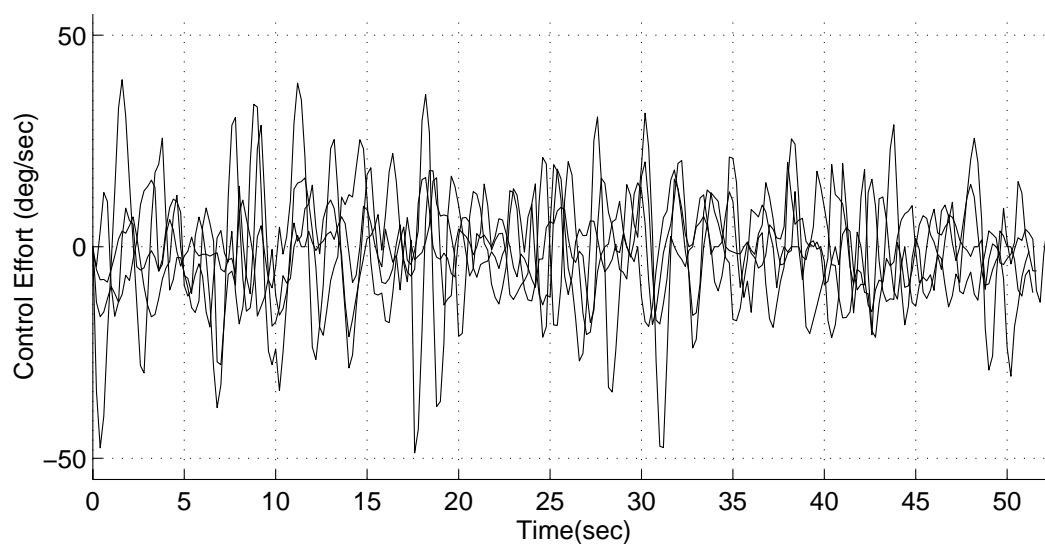


Figure 11.10: Experimental curve tracking control effort using DLQ tracker controller for curves shown in Fig. (11.5). The sum of the control effort norms for all four curves was 2.97 deg/sec/epoch.

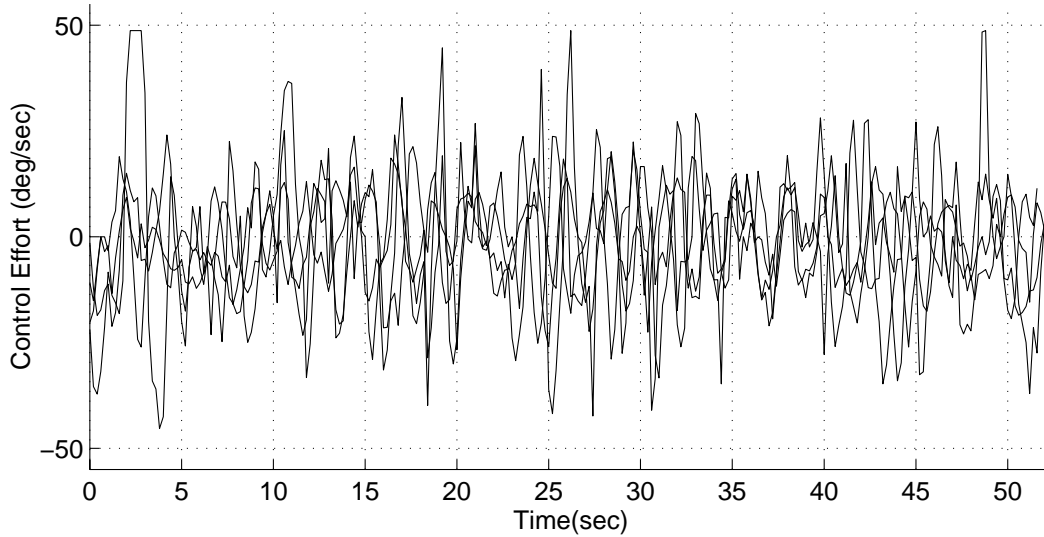


Figure 11.11: Experimental curve tracking control effort using feedback linearization controller for curves shown in Fig. (11.5). The sum of the control effort norms for all four curves was 3.41 deg/sec/epoch.

system. The problem with the linear local-error controller was that it assumed that the local curve conditions would exist indefinitely, and it created a control law based on these false assumptions. Another drawback to this controller was that it required solving the discrete algebraic Riccati equation in real-time at every epoch. This solution process was relatively fast but required extensive linear algebra software.

The DLQ tracker controller exhibited the smoothest performance of all three controllers. It used a medium amount of control effort and a great deal of computational effort. The computer's processing power limited the number of look-ahead intervals. Although the algorithm did not require solving the discrete algebraic Riccati equation in real-time, the linear algebra involved in solving the two-point boundary value problem made the control software reasonably complicated. This algorithm also required predicting the vehicle's future locations along the curve in the look-ahead interval, an additional and expensive computational burden.

The feedback linearization controller used the most control effort and exhibited the best tracking performance. The algorithm was fast since the control gains for

the feedback portion of the control signal could be precomputed ahead of time.¹ Although the equations for the time derivatives used by the algorithm were lengthy, the fundamental algorithm was straightforward to implement and the software was relatively simple. The controller's main drawback was the restriction that the system could not be non-minimum phase. If the control point was moved behind the center of the rear axle, modifications to the algorithm would have been required to maintain stability. Feedback linearization control of non-minimum phase systems is a topic of ongoing research in the controls community.

In all three controllers, either bias estimation or integral control was used to compensate for modeling and calibration errors. The gains on either the bias or integral states were tuned to reduce tracking error means without destabilizing the controller. Only the feedback linearization controller proved robust enough to allow the integral control gain to be increased enough to zero out the mean tracking errors. Increasing the gain on the bias states to eliminate the mean errors destabilized both the local-error and DLQ tracker controllers.

Controller accuracy did degrade as forward speed increased, but the degradation was smooth and roughly linear with forward speed. Disturbances significantly affected controller performance because of the absence of a suspension; the field conditions for these tests were not unusually rough. However, these tests were conducted without dual rear wheels. The rear tires were mounted at a relatively narrow spacing, and, as a result, ground disturbances created significant roll and tracking disturbances. Finally, the "smoothness" of the curve also had a significant effect on controller performance. Realistically, the tightness of curves shown in Fig. (11.5) show the worst that could reasonably be expected when tracking a riverbank, for example.

¹The state transition matrix in Eqn. (10.37) for the feedback-linearized equations of motion was constant.

CHAPTER 12

Vehicle Control on Sloped Terrain

The benefits of automatic vehicle control on flat terrain extend to sloped terrain. A practical autofarming system should be able to accommodate sloped terrain since virtually all fields have at least a slight slope angle. This chapter's objective was to improve row tracking performance by incorporating knowledge about how terrain slope affected the yaw and steering biases that arose as the vehicle “crabbed” to maintain tracking. On steeply sloped terrain, only crops that do not require high-precision tractor control are typically planted. If high-precision row tracking was possible on steeply sloped terrain, additional types of crops could be planted, increasing the ways in which a farmer could farm his land.

As an added benefit, the vehicle can produce contour maps of the field at a detail level well beyond that available from a commercial survey. This data can be valuable to farmers:

“By automatically processing this [three-dimensional position] data we can produce a highly detailed contour map of the paddock [field] revealing all the high and low spots and the direction of water flow across the paddock. While farmers will know these things from observation over many years, we feel the precise map we can produce just from cultivating a paddock is of great value. It can help to plan locations of contour banks and the best direction for laying down permanent wheel-tracks after considering the effects of water flow. This information can also be used to plan ground-leveling operations. We take far more site readings than you could realistically expect a surveyor to ever consider. With readings roughly once every two meters we pick up details between points that may have otherwise been missed.”

—Robert Mailer, *AgSystems Pty Ltd., Australia* [Mai97]

Tracking a trajectory when faced with a constant or slowly varying disturbance force is frequently encountered in other applications. Aircraft landing in the presence of cross-winds [HJ77] and ship navigation against a cross-current are two examples.

12.1 Standard Bias Estimation

Two common techniques in control/estimation theory will eliminate the steady-state tracking error caused by tractor side-slip on hills: integral control and bias estimation [F⁺90]. Bias estimation was chosen because it estimated two actual biases in the system—the tractor’s yaw and steering biases—and it divided the burden of zeroing out the steady-state tracking error into two variables (as opposed to only one variable in integral control), allowing adjustment of the disturbance variances for steering and yaw biases separately. The yaw bias can be thought of as the steady-state crab angle induced by the incline, and the steering bias can be thought of as the steering angle required to maintain the crab angle.

In this research, “standard” bias estimation is used to denote augmenting the state with two bias states in yaw and steering. In the observation matrix, the coefficients of the two bias states are simply set to one. This well-known technique will be referred to as “standard bias estimation”. The state vector was defined by

$$x = \begin{bmatrix} \psi & \delta & d \end{bmatrix}^T \quad (12.1)$$

and the observation vector y without biases was defined by

$$y = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \psi \\ \delta \\ d \end{bmatrix} \quad (12.2)$$

Standard bias estimation transforms the state vector into

$$x = \begin{bmatrix} \psi & \delta & d & \psi_b & \delta_b \end{bmatrix}^T \quad (12.3)$$

and the observation vector with biases becomes

$$y = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} \psi \\ \delta \\ d \\ \psi_b \\ \delta_b \end{bmatrix} \quad (12.4)$$

12.2 Slope-Adjusted Bias Estimation

Incorporating information about the tractor's dynamics into the bias estimation scheme could increase the estimator's ability to estimate the biases more accurately. It was observed that two rapidly varying parameters affect the yaw and steering biases: implement load and terrain slope. Other factors such as soil conditions, ballast, and tire configuration also affect these two biases but usually varied much more slowly. The experimental tractor used in this research did not measure the force of the towed implement; however, GPS attitude information could be used to incorporate some knowledge of how the biases are expected to vary with slope.

A simplified force balance diagram is shown in Fig. (12.1). N represents the downward force of the tractor, the force normal to the ground is F_1 , F_2 is the side-slip force tending to pull the tractor down-hill, and ζ is the slope angle of the hill as seen from behind the tractor. From trigonometry,

$$F_2 = N \sin \zeta \quad (12.5)$$

This side-slip force is the same force that the tractor will have to cancel out to track the trajectory. An overhead view of the forces exerted by a tractor tire on the ground is also shown in Fig. (12.1). If the tractor is "crabbed" at the correct angle μ , then the tire will exert a lateral force F_2 to balance the side-slip tendency. F_t represents the required traction force to pull both tractor and implement over the ground. Again

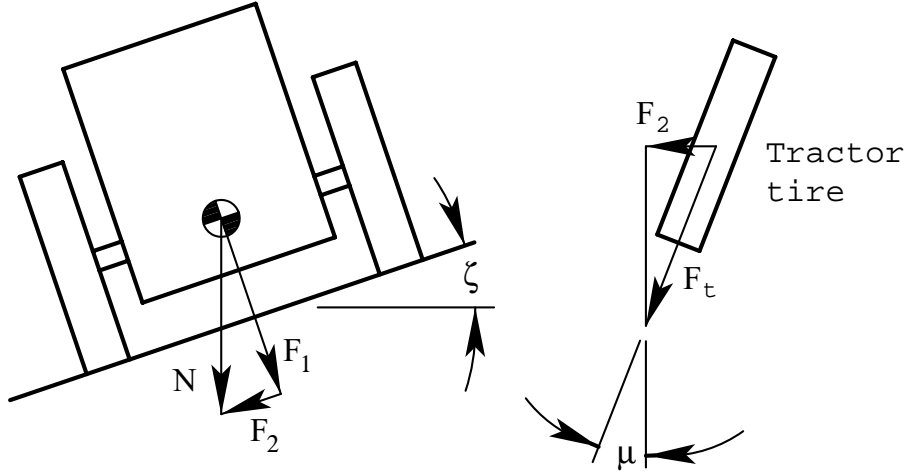


Figure 12.1: Simplified force balance on tractor and tire. The left diagram shows a rear view of the tractor looking forward, while the right diagram shows an overhead view of a tractor tire.

from trigonometry,

$$\frac{F_2}{F_t} = \sin \mu \quad (12.6)$$

$$\cong \mu \quad (12.7)$$

$$\frac{N \sin \zeta}{F_t} \cong \mu \quad (12.8)$$

$$\mu \cong K \sin \zeta \quad (12.9)$$

where K is a variable that incorporates the required traction force and weight of the tractor. These expressions for the required crab angle are intuitive: as the available traction force F_t increases, the required crab angle decreases. In other words, a tractor with a lower towing capacity will be crabbed at a larger angle than a tractor with a higher towing capacity for the same implement. Therefore, to first-order, yaw and steering biases should be proportional to the sine of the slope angle.

To incorporate this interdependency between slope and crab angle into the measurement equation, the observation equation becomes

$$y = \begin{bmatrix} 1 & 0 & 0 & \sin \zeta & 0 \\ 0 & 1 & 0 & 0 & \sin \zeta \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} \psi \\ \delta \\ d \\ K_1 \\ K_2 \end{bmatrix} \quad (12.10)$$

The proportionality constants K_1 and K_2 now become the estimated quantities. If the tractor reaches the end of one row and turns around to start the next, the biases should simply change in sign but not in magnitude; this estimation technique will do that (since $\sin \zeta$ changes sign with the slope) whereas standard bias estimation will not. Estimating the bias coefficients will be referred to as “slope-adjusted bias estimation”.

In calculating $\sin \zeta$, note that the slope of interest is not the slope of the terrain but rather the slope of the terrain perpendicular to the trajectory.¹ The cross-product of a unit vector, located in the horizontal plane of the tractor and oriented along the trajectory, with a unit vector out of the roof of the tractor was used to calculate this slope. This cross-product will be orthogonal to the trajectory *and* in the plane of the tractor. The unit vector q out of the roof of the tractor can be expressed in terms of the Euler angles for roll, pitch, and yaw. The desired direction of the trajectory yields the north and east components of the unit vector along the trajectory p . The down component of p that places p in the horizontal plane of the tractor comes from the fact that p and q should be orthogonal (*i.e.*, their dot product should be zero). Using this information,

$$q_1 p_1 + q_2 p_2 + q_3 p_3 = 0 \quad (12.11)$$

$$p_3 = -\frac{q_1 p_1 + q_2 p_2}{q_3} \quad (12.12)$$

with the subscripts 1, 2, and 3 used to denote north, east, and down respectively.

¹These slopes are identical if the tractor is moving perpendicular to the slope (*i.e.*, along a contour line).

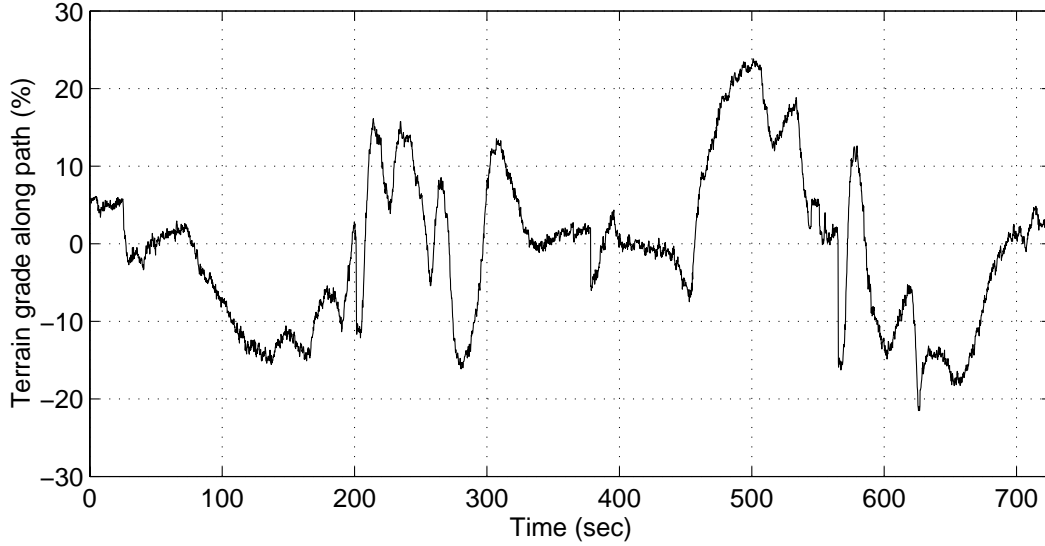


Figure 12.2: Time history of typical terrain grade along path.

Although this solution becomes singular for $q_3 = 0$, if q really has no vertical component, then most likely other issues are more pressing since the tractor is laying on its side.

The cross-product of p and q forms the slope vector s , which now forms a vector perpendicular to the trajectory and tangential to the ground. Thus,

$$\tan \zeta = \frac{s_3}{\sqrt{s_1^2 + s_2^2}} \quad (12.13)$$

$$\sin \zeta = s_3 \quad (12.14)$$

12.3 Experimental Results

The test platform used in this chapter was the Model 7800 tractor detailed in App. A. All data used in this chapter's experimental results was gathered at a farm in California's San Joaquin Valley. Terrain varied from 2-3° in slope (3% grade) to 20° and higher (36% grade). Fig. (12.2) shows a typical time-history of the terrain grade along the trajectory. Note how the grade varied widely along the path. For safety reasons,

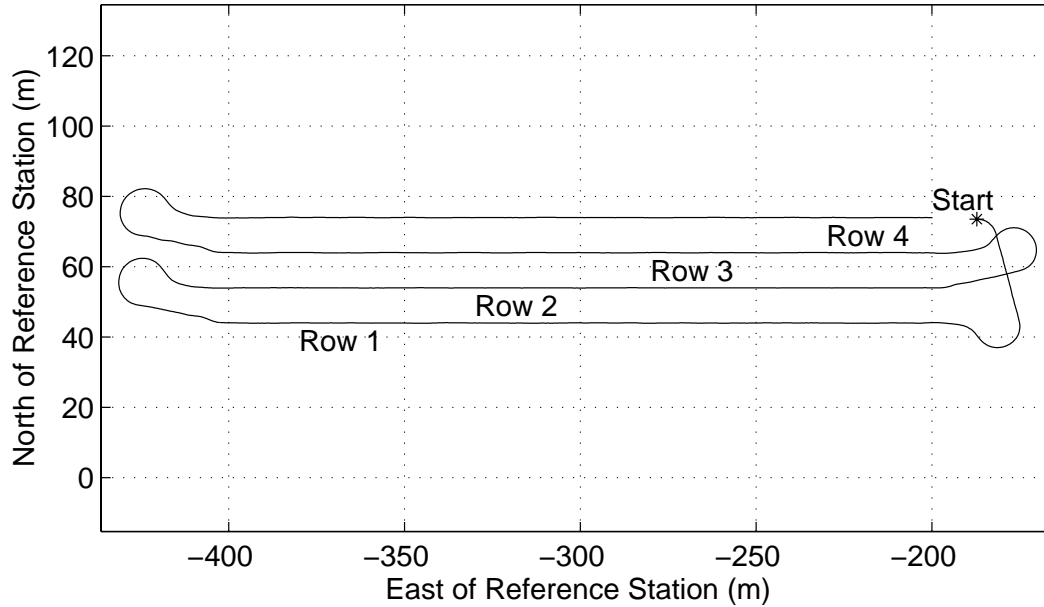


Figure 12.3: Overhead view of typical trajectory used for row control tests.

testing was limited to slopes of less than 15° (27% grade). All tests were conducted with a chisel plow in the lowered position and the engine under full load at 2,100 rpm and 8th gear. Ground speed was the same for all tests at roughly 1.6 m/s (3.3 mph). All driving, including U-turns and row acquisition, was done automatically. The driver, who remained in the cab during all tests, raised and lowered the chisel plow². Fig. (12.3) shows an overhead view of a typical four-row trajectory for row control tests. The start of the tractor's automatically controlled trajectory (*) and the row numbers are shown on the plot. Fig. (12.4) is a photo showing typical terrain slope and vehicle crab angle. Note that all tracking data shown is the estimated position of the vehicle's control point, which, in this set of experiments, was designated as a point on the ground directly beneath the center of the rear axle.

As a benchmark, a trajectory on a near-flat portion of the test field was used to test controller performance for comparison with sloped terrain row control. Standard

²This task could easily be automated.



Figure 12.4: The tractor climbing over a sloped ridge. The tractor is actually approaching the photographer with a significant crab angle and steering bias.

bias estimation was used to zero out the small biases introduced by the terrain, which was sloped at an average of -2.1° . Fig. (12.5) shows the tracking error as a function of time for each of the four rows. Note that the initial large error for each of the four rows stemmed from line acquisition after the previous U-turn. The periodic tracking errors on the third row in Fig. (12.5) are a result of the tractor side-slipping into small grooves from previous tests. In Fig. (12.6), the terrain slope for each of the four rows is shown. Standard bias estimation easily absorbed the effects of the shallow slope.

For comparison between automatic control and human control on sloped terrain, the author, not an experienced tractor driver, drove the tractor along the sloped row trajectory while using the tracking error information from the GPS system to help him judge his performance. Fig. (12.7) shows a time history of the tracking error along the four rows, and Fig. (12.8) shows the terrain grade along the four rows.

To support the assertion that some form of bias estimation was required for accurate control, a test was performed where the biases were not estimated. Fig. (12.9) shows a strong mean tracking error that ranged between -18.6 and 36.5 cm and a

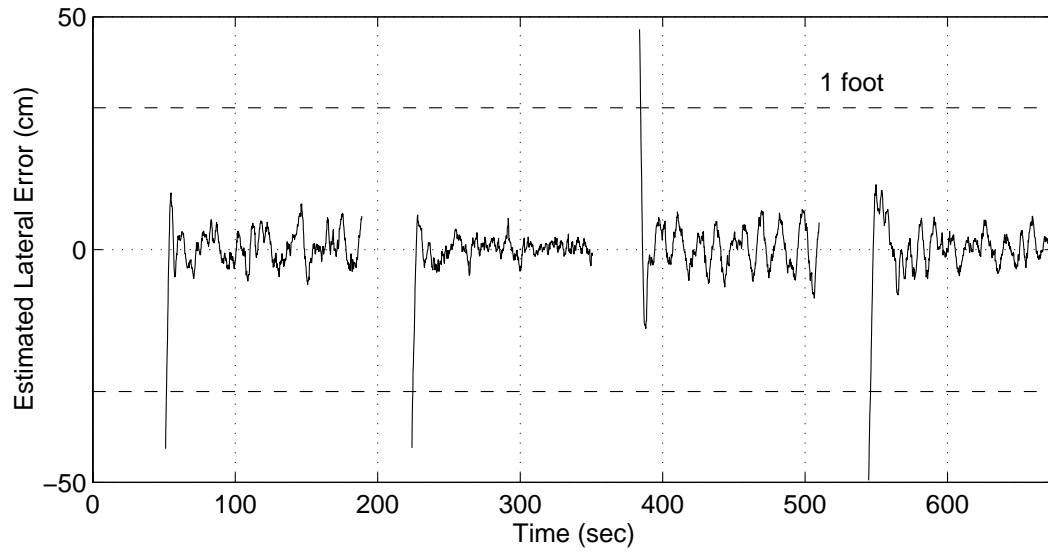


Figure 12.5: Row guidance on near-flat terrain with standard bias estimation: 0.4 cm mean, 4.0 cm standard deviation (one-sigma). Large initial spikes are from row acquisition. Terrain slope for these tests is shown in Fig. (12.6).

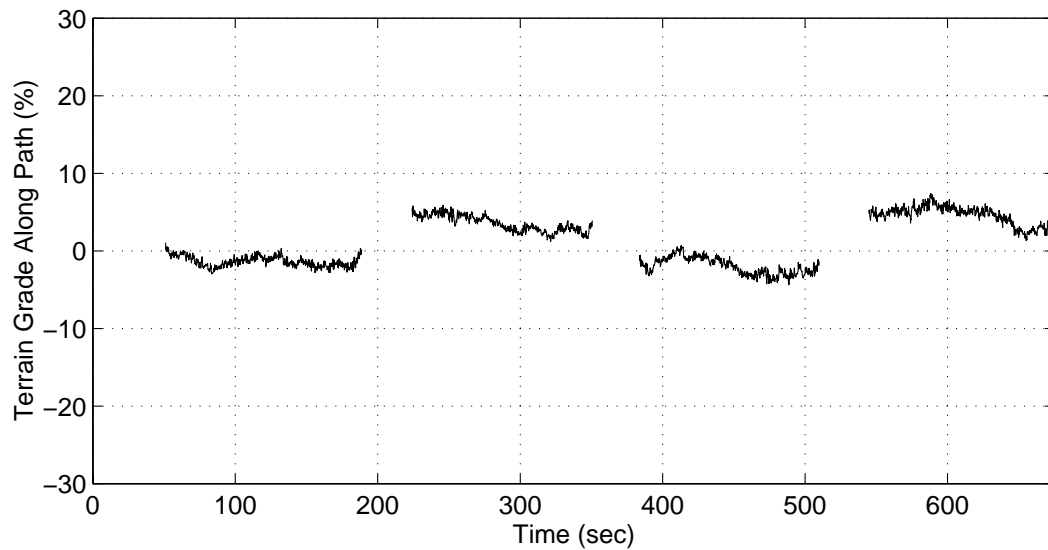


Figure 12.6: Terrain grade for row guidance results shown in Fig. (12.5).

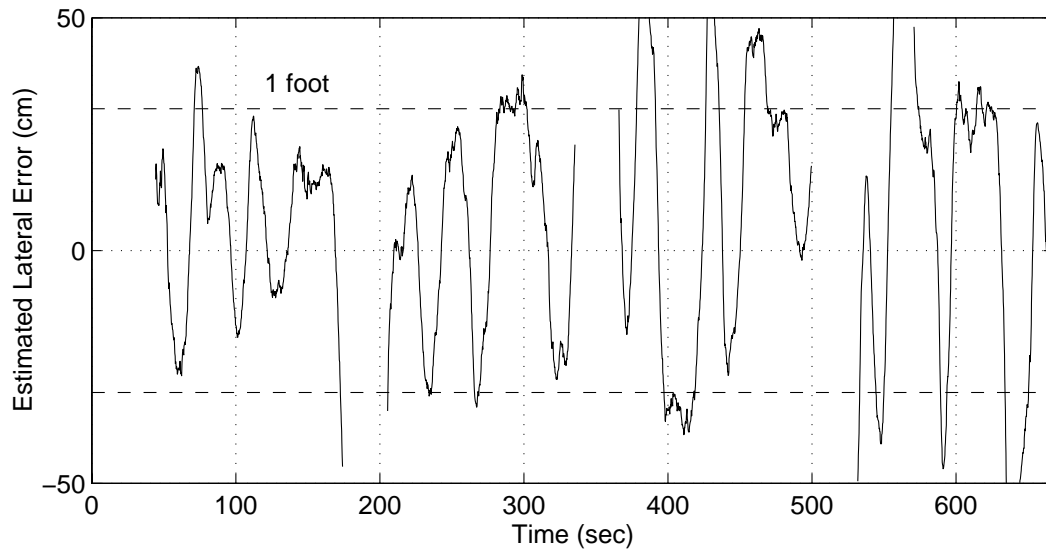


Figure 12.7: Human control on sloped terrain: 6.3 cm mean, 24.1 cm standard deviation (one-sigma). Terrain slope for these tests is shown in Fig. (12.8).

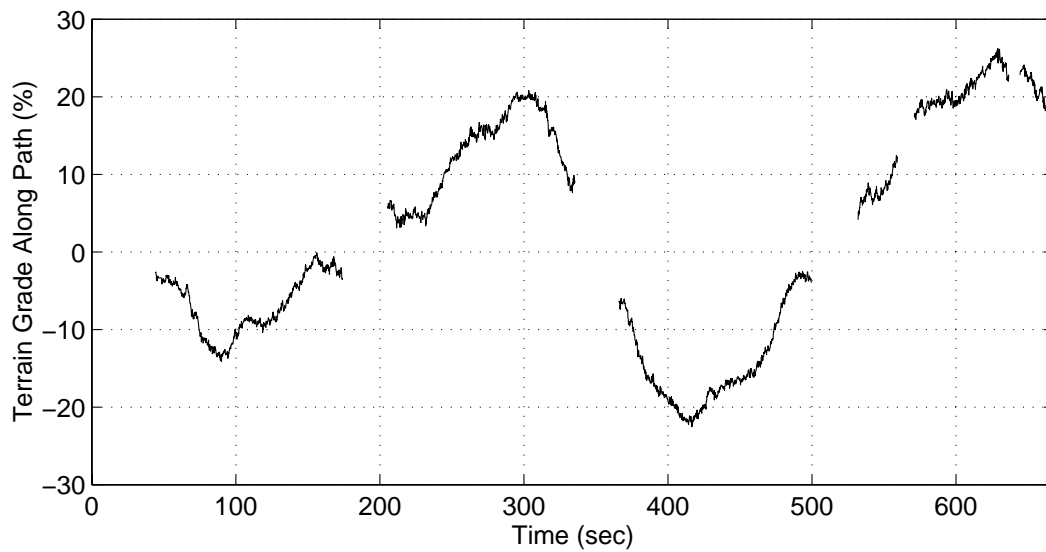


Figure 12.8: Terrain grade for row guidance results shown in Fig. (12.7).

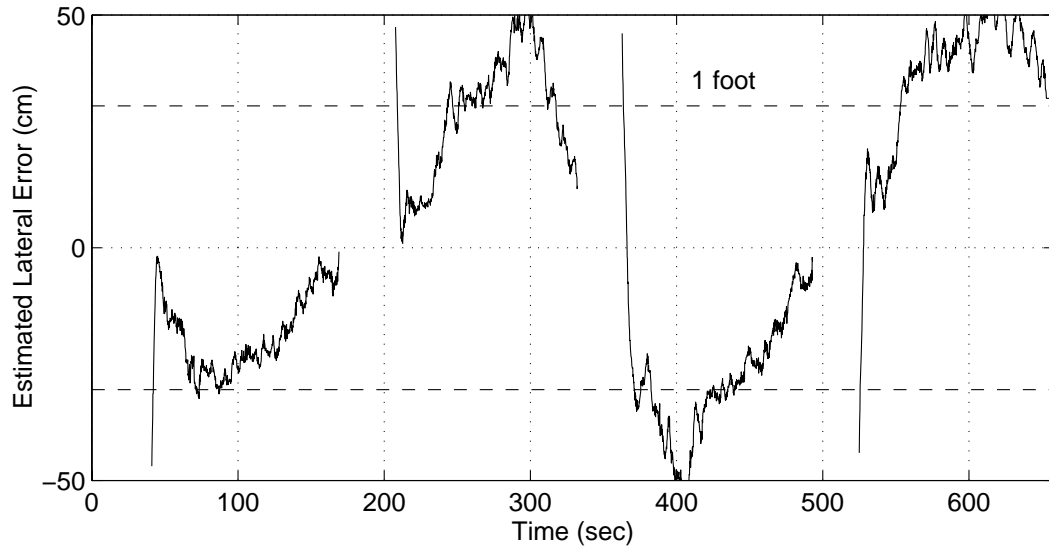


Figure 12.9: Row guidance on sloped terrain without bias estimation: 27.7 cm average *absolute* mean, 13.0 cm standard deviation (one-sigma). The effects of left- and right-hand slopes are obvious and can be correlated with the terrain grade along this path, shown in Fig. (12.10).

standard deviation between 8.4 and 15.5 cm for all four rows. A time history of the terrain grade along this path is shown in Fig. (12.10). Clearly, a control system with such large tracking errors would be unacceptable to farmers. When standard bias estimation was added to the control/estimation algorithm, controller performance improved dramatically. Fig. (12.11) shows controller performance for row guidance on sloped terrain while using standard bias estimation. Fig. (12.12) shows a time history of the terrain slope for the four rows in Fig. (12.11).

To compare slope-adjusted bias estimation against standard bias estimation, a four-row trajectory was completed with slope-adjusted bias estimation. The controller performance shown in Fig. (12.13) demonstrated roughly 25% improvement over controller performance with the standard bias estimation scheme. The terrain grade for the four rows in Fig. (12.13) is shown in Fig. (12.14). Note that the slope-adjusted bias estimation tests were conducted under more challenging conditions than the standard bias estimation tests since the terrain grade in Fig. (12.14) varied more rapidly than the grade shown in Fig. (12.12).

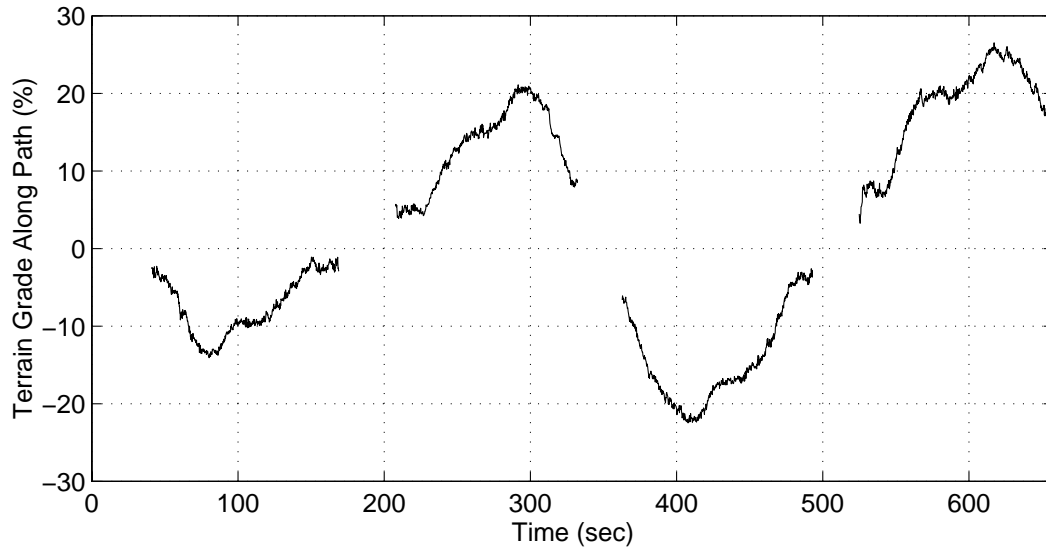


Figure 12.10: Terrain grade for row guidance results shown in Fig. (12.9). The correspondence can be seen between the tracking errors in Fig. (12.9) and the terrain grade shown here.

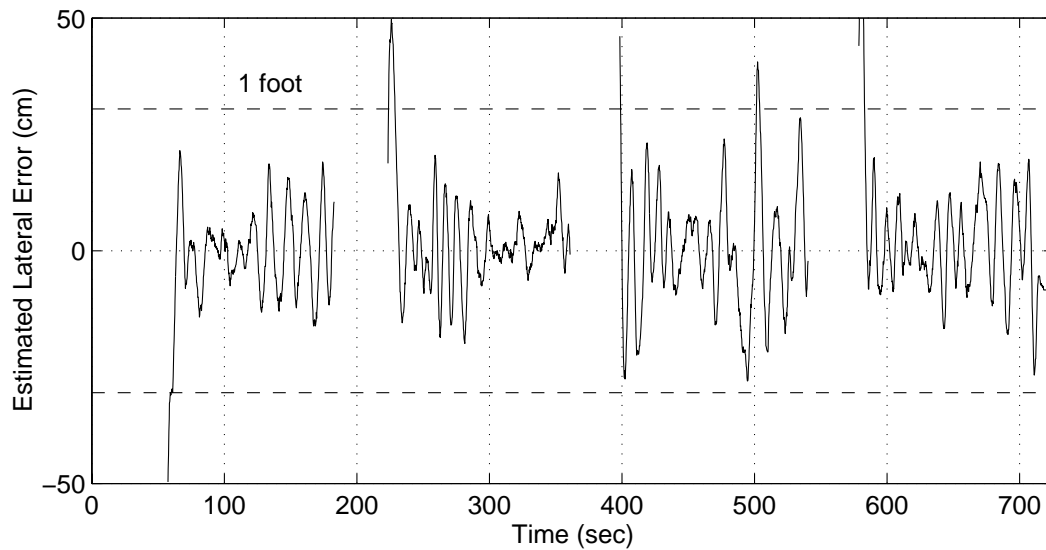


Figure 12.11: Row guidance on sloped terrain with standard bias estimation: 0.9 cm mean, 8.5 cm standard deviation (one-sigma). Note no effect of left- and right-hand bias. Terrain slope for these tests is shown in Fig. (12.12).

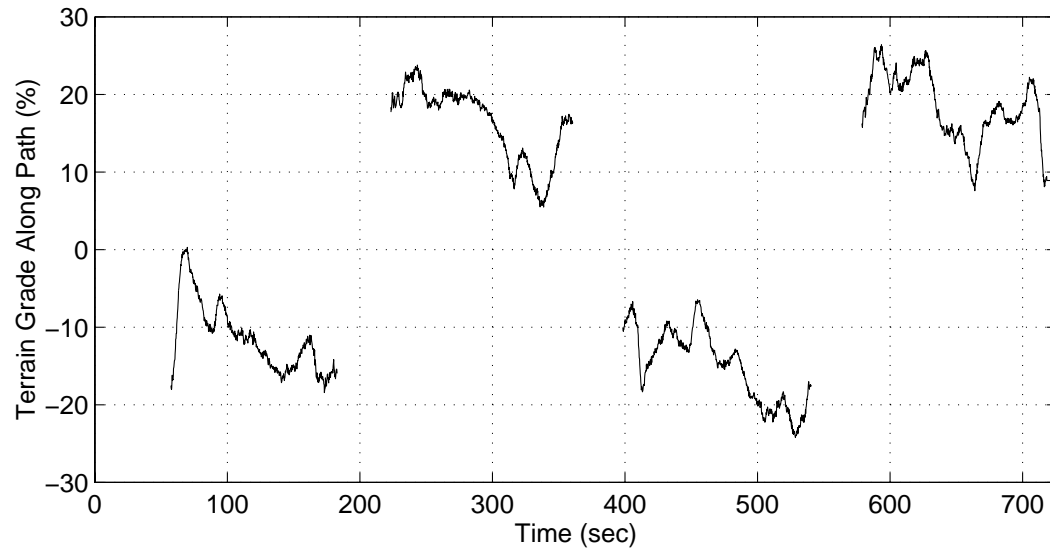


Figure 12.12: Terrain grade for row guidance results shown in Fig. (12.11).

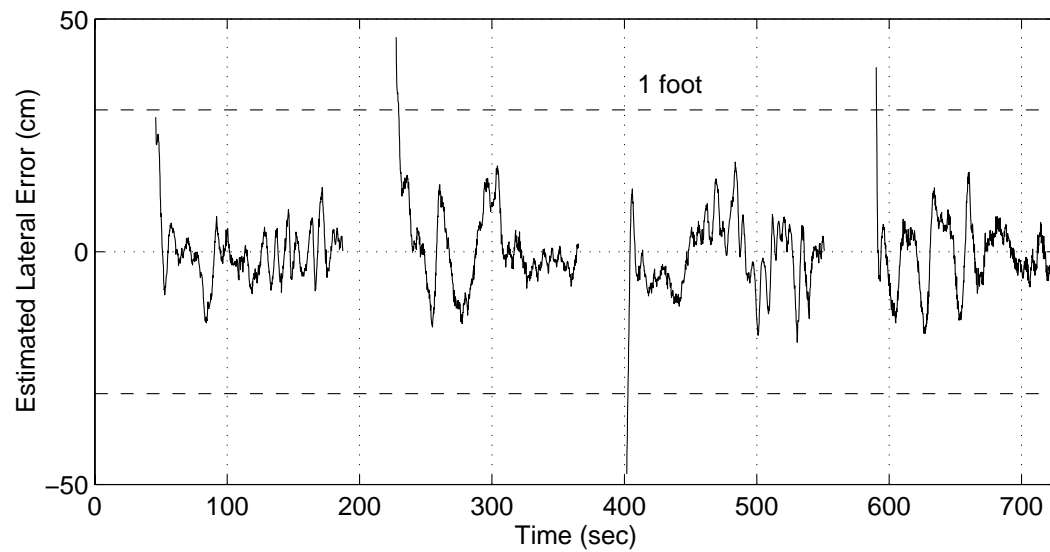


Figure 12.13: Row guidance on sloped terrain with slope-adjusted bias estimation: - 0.8 cm mean, 6.4 cm standard deviation (one-sigma). Terrain slope for these tests is shown in Fig. (12.14).

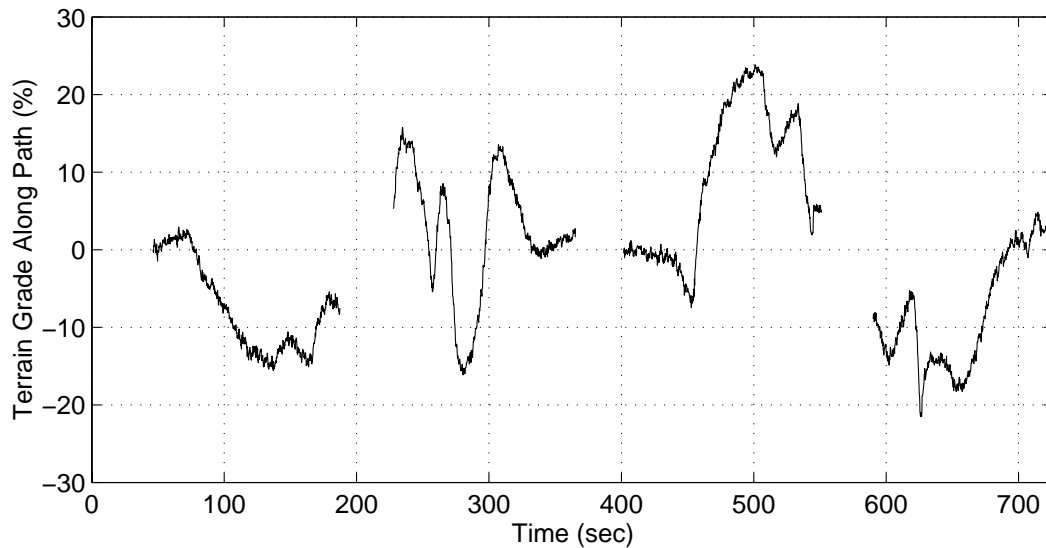


Figure 12.14: Terrain grade for row guidance results shown in Fig. (12.13).

The mean and standard deviation in tracking error for control on flat terrain, human control, control without bias estimation, control with standard bias estimation, and control with slope-adjusted bias estimation are summarized in Tbl. 12.1. Slope-adjusted bias estimation shows a slight improvement over standard bias estimation. Both schemes essentially zero out the mean tracking error as expected. Interestingly, the standard deviation on near-flat terrain is only roughly 50% lower than the standard deviation on sloped terrain: using slope-adjusted bias estimation on steep terrain yielded results almost at the level of those on flat terrain. The human driver performed much worse than either bias estimation scheme. The accurate attitude information available from the GPS attitude sensor made it possible to accurately calculate the terrain's slope; without the high accuracy offered by the GPS attitude system, the results presented here might not have been realized.

| Row No. | | Auto, Flat Terrain | Human Control | Auto, No Bias Estimation | Auto, Standard Estimation | Auto, Slope-Adjusted Estimation |
|------------|------------------------------|--------------------|---------------|--------------------------|---------------------------|---------------------------------|
| 1 | <i>mean</i> | 0.4 | 5.9 | -18.6 | -1.0 | -1.0 |
| | <i>1-σ</i> | 3.4 | 16.2 | 8.4 | 5.0 | 4.8 |
| 2 | <i>mean</i> | 0.3 | 3.7 | 29.0 | 2.3 | -0.6 |
| | <i>1-σ</i> | 2.0 | 20.9 | 13.2 | 8.3 | 7.2 |
| 3 | <i>mean</i> | 0.7 | 10.2 | -26.8 | -0.1 | -1.0 |
| | <i>1-σ</i> | 7.3 | 30.4 | 14.9 | 9.4 | 7.1 |
| 4 | <i>mean</i> | 0.1 | 5.3 | 36.5 | 2.5 | -0.6 |
| | <i>1-σ</i> | 3.1 | 29.2 | 15.5 | 11.3 | 6.6 |
| <i>Avg</i> | <i>mean</i> | 0.4 | 3.6 | 27.7 | 0.9 | -0.8 |
| | <i>1-σ</i> | 4.0 | 24.1 | 13.0 | 8.5 | 6.4 |

Table 12.1: Experimental row tracking performance on sloped terrain. The mean value for the tests run without bias estimation is the average *absolute* mean. All results are in centimeters.

CHAPTER 13

Vehicle Control on Contours

Farming along a hill's contours can reduce top soil erosion and improve irrigation efficiency by reducing water run-off. Therefore, accurate contour farming may be important on sloped terrain. When contour farming, most tractor drivers will drive the first pass as close to the contour as they can. Every subsequent pass is then made one implement width away from the previous one. If the slope of the terrain changes along the trajectory, each successive pass will diverge from the true contour. Once the driver has judged the divergence to be too much, a new contour pass is driven and the process repeated. The gaps between batches of passes are filled in separately. This research sought to create a control algorithm capable of accurately controlling the tractor along that first contour pass. Accuracy was judged by the mean and standard deviation of the height error of the vehicle's control point.¹

13.1 Control Algorithms

Accurate tractor control along straight rows on sloped terrain was experimentally demonstrated in Ch. 12. If the tractor's location along a desired contour could be linearized about that contour, the same linear controller used for row control could then be used to control the tractor along the contour. Although contours are usually nonlinear, realistic contours were assumed to curve relatively slowly so that future information about how the contour line was changing ahead would have little impact

¹The control point was located on the ground directly beneath the center of the rear axle.

on the current control signal. In other words, a controller based only on current error information could be used. This assumption was important because *it allowed contour control without the use of a terrain map or other source of terrain information.*

13.2 Specifying Contour Trajectories

Because user flexibility was important, contour trajectories were specified only by a vertical distance above the reference station antenna. This meant that a controller would attempt to track that particular contour indefinitely without regard for field boundaries. Therefore, the control algorithms presented here are not a replacement for a human operator, but rather a tool to assist the operator in accurate contour farming. It was assumed that the human operator would want to offset the original automatic contour manually; therefore, this chapter did not address contour offsetting.

For the vehicle model in Eqns. (3.43–3.45), accurate contour control involved calculating the vehicle’s tracking error, yaw error, and steer angle error. Once the vehicle’s linearized state had been estimated, the full-state linear control law

$$u = -K(\hat{x} - x_r) \quad (13.1)$$

was used to generate the control signal where the gains K were calculated using a LQR control law. The difficult part of the contour control algorithm involved estimating the vehicle’s state. To accomplish this task, vehicle attitude information was used to calculate contour direction and subsequently yaw error. Height information from the GPS position sensor was then used to calculate the contour tracking error. Even on terrain considered steep for farming, the slope angle was relatively shallow, and small measurement noises in attitude and position created large uncertainties in the vehicle’s state. Bias estimation was used to zero out the tracking error that stemmed from the inevitable crabbing of the vehicle necessary to track the contour.

13.2.1 Measuring Yaw Error

To calculate yaw error, the attitude information available from the GPS system was used to generate a unit vector q out of the tractor’s roof. The direction of steepest

descent ψ_ζ of the projection of q onto the North-East plane could be calculated from q 's horizontal components:

$$\tan \psi_\zeta = \frac{q_2}{q_1} \quad (13.2)$$

where the subscripts 1 and 2 denote north and east, respectively. The vehicle's actual heading was then used to measure the yaw error:

$$\psi_e = \psi - \tan^{-1} \left(\frac{q_2}{q_1} \right) \pm \frac{\pi}{2} \quad (13.3)$$

where the factor $\pi/2$ was either added or subtracted depending on whether or not the contour rotated clockwise or counter-clockwise.

13.2.2 Measuring Steering Error

The reference steer angle was assumed to be constant. Although this assumption held true only if the terrain slope was constant, the slope was assumed to vary slowly enough that the reference steer angle variations could be assumed small. Experiments showed that the adjusting the process noise variance of the disturbances acting on the steer angle bias absorbed what little error was introduced by this assumption.

13.2.3 Measuring Tracking Error

Once the unit vector q had been calculated, the terrain slope angle ζ could be calculated from

$$\tan \zeta = \frac{\sqrt{q_1^2 + q_2^2}}{q_3} \quad (13.4)$$

Note that this slope direction is also orthogonal to the trajectory because, by definition, the contour is perpendicular to the direction of steepest descent.

This information was then used to translate height error h into tracking error d . In a 3-2-1 coordinate frame

$$d = \frac{h}{\sin \zeta} \quad (13.5)$$

$$= \frac{h}{\sqrt{1 - \cos^2 \theta \cos^2 \phi}} \quad (13.6)$$

13.3 Measurement Variances

An accurate estimate of the yaw and tracking errors could improve controller performance by filtering out measurement noise. Therefore, approximations were needed for the measurement uncertainties in tracking error and desired yaw as a function of known variances in height and attitude noise.

13.3.1 Tracking Error Measurement Variance

For tracking error, the above requirement translates into finding

$$E \left[\left(d - \hat{d} \right)^2 \middle| \hat{h}, \hat{\Theta}^v \right] \quad (13.7)$$

where \hat{d} is the estimated tracking error. A first-order multi-variable Taylor series expansion [Hil76] of d about \hat{d} using values of $\hat{\phi}$, $\hat{\theta}$, and \hat{h} can be used to obtain an approximation for $d - \hat{d}$:

$$\begin{aligned} d \cong & \frac{\hat{h}}{\sqrt{1 - \cos^2 \hat{\theta} \cos^2 \hat{\phi}}} + \left[\left(h - \hat{h} \right) \frac{\partial}{\partial h} + \left(\phi - \hat{\phi} \right) \frac{\partial}{\partial \phi} \right. \\ & \left. + \left(\theta - \hat{\theta} \right) \frac{\partial}{\partial \theta} \right] \left(\frac{h}{\sqrt{1 - \cos^2 \theta \cos^2 \phi}} \right) \end{aligned} \quad (13.8)$$

$$\begin{aligned} d - \hat{d} = & \left(h - \hat{h} \right) \frac{1}{\sqrt{1 - \cos^2 \hat{\theta} \cos^2 \hat{\phi}}} \\ & - \left(\phi - \hat{\phi} \right) \frac{\hat{h} \cos^2 \hat{\theta} \cos \hat{\phi} \sin \hat{\phi}}{\left[1 - \cos^2 \hat{\theta} \cos^2 \hat{\phi} \right]^{3/2}} \\ & - \left(\theta - \hat{\theta} \right) \frac{\hat{h} \cos \hat{\theta} \cos^2 \hat{\phi} \sin \hat{\theta}}{\left[1 - \cos^2 \hat{\theta} \cos^2 \hat{\phi} \right]^{3/2}} \end{aligned} \quad (13.9)$$

Squaring Eqn. (13.9) and taking the expected values of the resulting terms with the assumption that sensors are unbiased,

$$\begin{aligned}
 E \left[\left(d - \hat{d} \right)^2 \middle| \hat{d}, \hat{\Theta}^v \right] &= \sum_h \frac{1}{1 - \cos^2 \hat{\theta} \cos^2 \hat{\phi}} \\
 &+ \sum_{\phi} \frac{\hat{h}^2 \cos^4 \hat{\theta} \cos^2 \hat{\phi} \sin^2 \hat{\phi}}{\left[1 - \cos^2 \hat{\theta} \cos^2 \hat{\phi} \right]^3} \\
 &+ \sum_{\theta} \frac{\hat{h}^2 \cos^2 \hat{\theta} \cos^4 \hat{\phi} \sin^2 \hat{\theta}}{\left[1 - \cos^2 \hat{\theta} \cos^2 \hat{\phi} \right]^3} \quad (13.10)
 \end{aligned}$$

where \sum_h , \sum_{ϕ} , and \sum_{θ} are the variances in the measurements of height, roll, and pitch. This variance approximation shows how small angles in roll and pitch (*i.e.*, terrain that is only slightly sloped) can lead to large measurement uncertainties. In Fig. (13.1), the effect of terrain slope on tracking uncertainty can be seen for identical height uncertainties. As predicted by Eqn. (13.10), on flat ground, any uncertainty in height will lead to an infinite uncertainty in tracking error. However, this scenario is unrealistic: if the terrain is flat, the farmer has no reason to use a contouring algorithm.

13.3.2 Yaw Error Measurement Variance

An expression for the heading error measurement variance can be derived using the same procedure:

$$\begin{aligned}
 E \left[\left(\psi_e - \hat{\psi}_e \right)^2 \right] &= \sum_{\phi} \frac{\sin^2 \hat{\theta}}{\left[1 - \cos^2 \hat{\theta} \cos^2 \hat{\phi} \right]^2} \\
 &+ \sum_{\theta} \frac{\cos^2 \hat{\theta} \cos^2 \hat{\phi} \sin^2 \hat{\phi}}{\left[1 - \cos^2 \hat{\theta} \cos^2 \hat{\phi} \right]^2} + \sum_{\psi} \quad (13.11)
 \end{aligned}$$

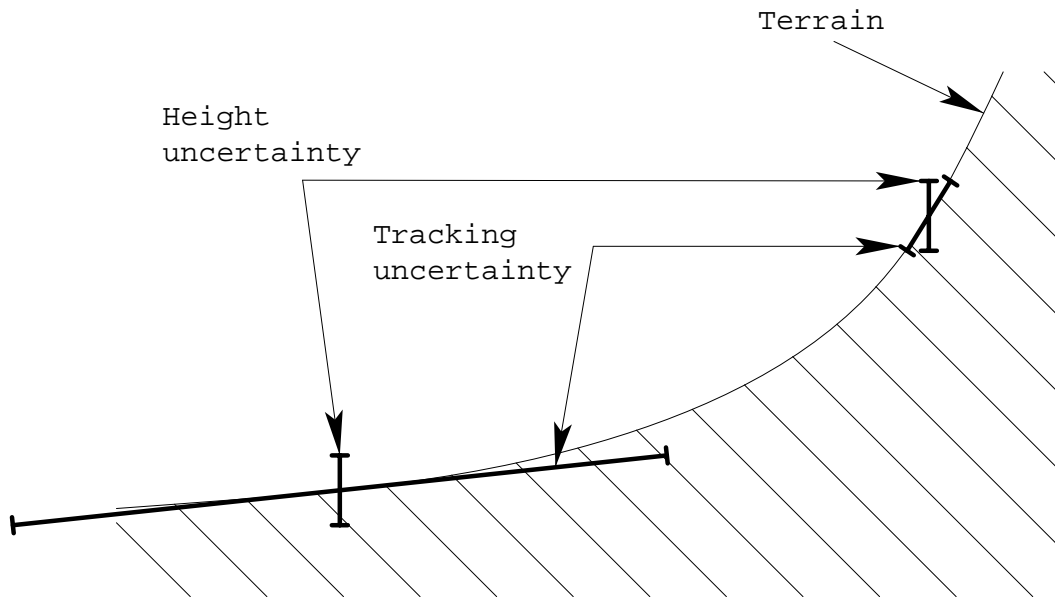


Figure 13.1: Effect of terrain slope angle on tracking uncertainty. The viewer is standing on the ground looking at a cross-section of the tractor's path.

13.4 Experimental Results

Tractor configuration, implement, and speed in the contour tests were identical to conditions in the sloped-terrain tests detailed in Ch. 12. Slope-adjusted bias estimation was used to zero out steady-state errors. Experimental results for the height error for a contour 15 meters above the reference station are shown in Fig. (13.2). The mean height error was 0.5 cm and the standard deviation 4.3 cm. Figs. (13.3–13.4) show a time history of the estimated measurement variances in yaw and tracking error. Even though the attitude and position sensors were very accurate, the relatively shallow slope angle (average about 7° (14% grade), maximum of 13° (23% grade) for this contour) created large uncertainties in tracking error measurements of about 50 cm and in yaw error measurements of about 15° (one-sigma).

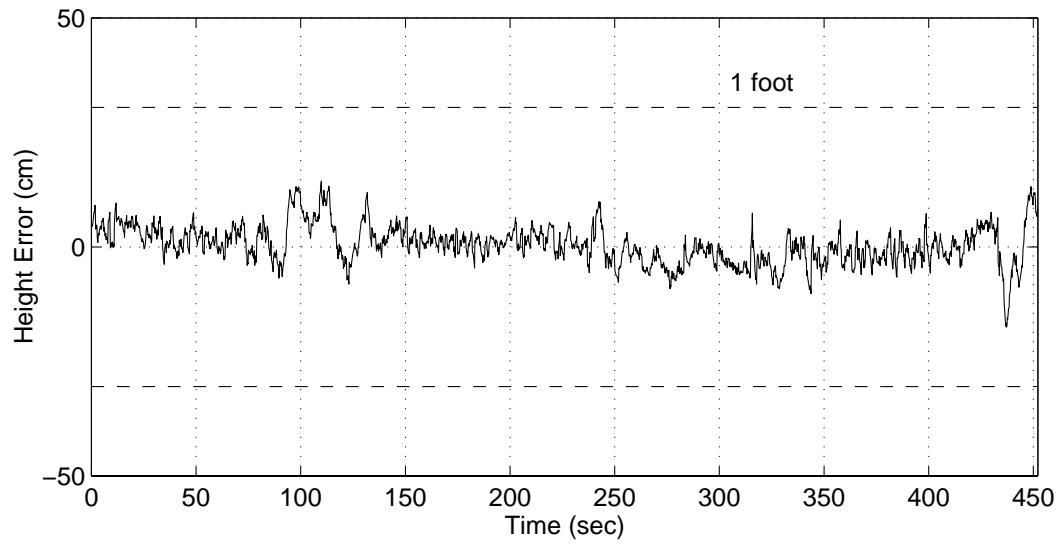


Figure 13.2: Height error for automatic control along a contour: 0.5 cm mean, 4.3 cm standard deviation (one-sigma). Dashed lines are one foot.

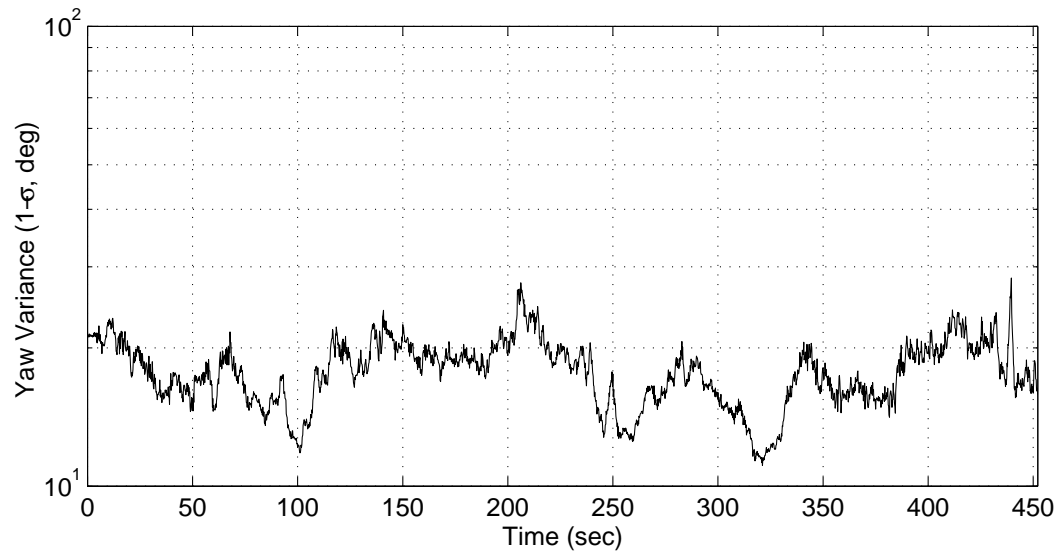


Figure 13.3: Yaw error measurement variance for automatic control along a contour.

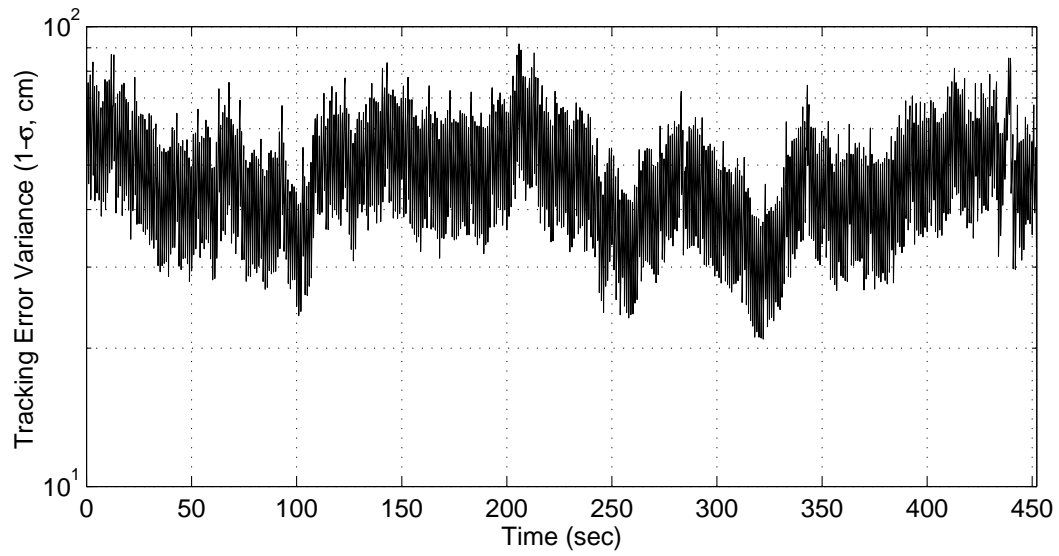


Figure 13.4: Tracking error measurement variance for automatic control along a contour.

CHAPTER 14

Conclusions and Future Research

This research has expanded prior agricultural vehicle control results by developing and demonstrating high-precision control algorithms that steer a general front-wheel-steered vehicle along arcs, spirals, curves, and contours to sub-ten-centimeter precision. Farmers may use these non-linear trajectory “building blocks” to smoothly join linear trajectories, creating a seamless *global* trajectory. These results facilitate farming operations previously impossible to an automatic controller:

- Fields bordered by irregular boundaries such as streams or roads can now be worked with curves that the farmer can easily specify.
- Non-linear trajectories that avoid in-field obstacles can be generated and tracked so that precision farming operations can be continuous despite in-field telephone poles or electrical towers.
- Fields irrigated with center-pivot irrigation systems can be farmed using the spiral trajectory introduced in this research.
- Rows can be tracked on steep terrain without suffering a large performance penalty thanks to an improved bias estimation algorithm that accounts for the sloped terrain’s effects on the biases.
- Contours can be farmed automatically to centimeter-level precision without the aid of a terrain database.

The new model introduced in this research can adapt to a wide variety of factors that can alter the vehicle's response. With the model's ability to identify the effects of factors such as changing front-wheel traction, farmers will not need to specify changes in ballast, tires, or implement. Instead, the model will be able to quickly adjust based on the observed changes in performance so that the controller can maintain high tracking precision.

This research has also forged tools for future researchers so that the envelope of autofarming systems' capabilities can be further expanded:

- The new three-dimensional towed implement model developed here would allow engineers to develop towed implement controllers.
- With the ability to automatically control the tractor along non-linear trajectories, engineers can incorporate paths created by path-planning algorithms, algorithms which could help automate the task planning process.

Finally, the error analysis introduced here to describe the effects of poor attitude measurements on control point position accuracy are general: they can be applied anytime an engineer measures a point's location in a rotating body based on a position measurement taken elsewhere on the body. This analysis also demonstrated the need for precise attitude measurements in autofarming.

14.1 Summary of Results

14.1.1 Vehicle Modeling

New three-dimensional nonlinear models of a tractor and a tractor-trailer have been developed. A variety of time-varying factors influenced the dynamics, and these factors were accommodated by additional states. In simulation, a first-order nonlinear filter suffered little performance penalty compared to a computationally expensive second-order filter. Experimental data revealed that the first-order filter converged on an important parameter—the proportional relationship between steer angle and

yaw rate—from a poor initial guess to the known value in under a minute with a steady-state accuracy of approximately 10%.

14.1.2 Analysis of the Lever Arm Correction

The vector correction necessary to translate the GPS position measurement on the roof of the tractor to the tractor's control point on the ground beneath the center of the rear axle was a nonlinear function of an uncertain tractor attitude. An approximation for the additional uncertainty introduced by the correction was derived based on a first-order series expansion of the correction. Monte Carlo simulation results showed that the additional uncertainty grew rapidly as attitude uncertainty increased. These results were used to support the claim that an accurate attitude estimate was as important to control point position measurement as an accurate position measurement at the roof antenna.

14.1.3 Control on Arcs and Spirals

Several types of nonlinear trajectories were identified as necessary components for a practical autofarming system: arcs, spirals, curves, and contours. The nonlinear vehicle model was simplified for nominal conditions to a low-order model suitable for use in a real-time control algorithm. The model was transformed into polar coordinates. For arc trajectories, the model was linearized about nominal conditions, and experimental data revealed that despite rough ground, the controller allowed the Model 8400 tractor's control point to track an arc to a mean of -0.28 cm and a standard deviation of 3.43 cm as measured by CPDGPS. Spiral trajectories increased or decreased one implement width per revolution and could be applied to fields irrigated by center-pivot irrigation. The polar analysis developed for arcs was extended to spirals, whose radius of curvature was no longer constant. The Model 7800 tractor tracked a large spiral trajectory while pulling a towed implement under full engine load to a mean tracking error of -0.22 cm and a standard deviation of 5.27 cm.

14.1.4 Control on Curves

Curve trajectories, which could arise from irregular field boundaries, were interpolated from discrete input position data using cubic splines. Several issues such as determining the vehicle's location along the curve, calculating the rate of change of curve-dependent parameters, and predicting future vehicle locations along the curve were addressed. Four controllers were derived for vehicle curve tracking. The first controller, a linear local-error controller, adapted the polar analysis to curve trajectories, linearizing the model about the current curve conditions. This controller tracked curves to a mean of 1.99 cm and a standard deviation of 9.07 cm. The second controller, though also linear, previewed a finite amount of the curve. The well-known discrete linear quadratic tracking algorithm used this future state information to track a curve to a mean of 3.44 cm and a standard deviation of 6.69 cm. The third control algorithm, a nonlinear algorithm based only on local information, used feedback linearization to linearize the nonlinear vehicle dynamics exactly, splitting the control signal into a feedforward term and a feedback term. This algorithm tracked a curve to within -0.01 cm mean and 4.12 cm standard deviation. Finally, nonlinear optimization techniques, though too computationally expensive to run in real-time, were examined as a possible future control solution. This fourth and final curve control algorithm was nonlinear and previewed a finite amount of future curve information, which was used to derive general expressions for the Jacobian and Hessian of the tracking error in terms of curve parameters. In simulation, this control algorithm tracked a realistic curve under ideal conditions to within virtually zero tracking error.

14.1.5 Control on Sloped Terrain and Contours

To control the tractor accurately on sloped terrain, the GPS attitude information was used to calculate slope information, which was then fed into a variant of bias estimation termed "slope-adjusted bias estimation". On steep slopes of up to 28% grade, the tracking error along straight rows while pulling a heavy implement was -0.8 cm mean and 6.4 cm standard deviation, a 25% improvement over control on the same trajectory using standard bias estimation. The accurate attitude information

| Trajectory | Mean Tracking Error (cm) | Standard Deviation (cm) |
|-----------------------|-----------------------------|----------------------------|
| <i>Arcs</i> | < 1 | $3 - 4$ |
| <i>Spirals</i> | < 1 | $5 - 6$ |
| <i>Curves</i> | $0 - 4$ | $4 - 10$ |
| <i>Rows on slopes</i> | < 1 | $6 - 7$ |
| <i>Contours</i> | < 1 | $4 - 5$ |

Table 14.1: Summary of experimental tracking performance for all trajectory types.

was also used to develop a tractor controller that could track a contour without prior knowledge of the terrain. On steep terrain, the contour tracking control algorithm held the contour to a mean error in height of 0.5 cm and a standard deviation of 4.3 cm.

Tbl. 14.1 summarizes experimental results for controller tracking performance along all the trajectories introduced in this research. The values listed are representative of what a farmer might expect not under ideal conditions, but under realistic conditions.

14.2 Future Research

One of the difficulties encountered in graduate school was deciding when to stop researching and start writing. Each new avenue of research has revealed more avenues, each more interesting than the last. This project has been especially interesting because there is an immediate commercial need for automatically controlled tractors. This need virtually guarantees that autofarming systems will be commercialized and that industry and academia will address many of the research issues listed below:

1. *Higher-order modeling.* Higher-order vehicle dynamics were assumed negligible when creating the tractor and implement models, and experiment showed that these assumptions were valid at the relatively low operating speeds encountered in high-precision agriculture. Preliminary control experiments at high speed, however, showed that the neglected dynamics should no longer be assumed

negligible, and that a tractor model that incorporates higher-order dynamics would be a key component of a high-speed control algorithm. Members of the farming community who have witnessed this research have often expressed a desire not to increase the tracking ability of a tractor for a particular speed, but to increase the tractor's forward speed while achieving the same accuracy as a good human operator would driving at "normal" speed. In other words, the quality of the steering control will dominate the choice of tractor speed [Bot82]. In some high-precision applications, a tractor's forward speed is now limited by the operator's driving ability, not the tractor's ability to pull the implement through the soil. Therefore, better control at high speeds will be a large benefit.

2. *Real-time system identification and adaptive control.* Real-time identification of K_δ and various biases is just the beginning of real-time identification of changing vehicle parameters. Tires, ballasting, and even the temperature of the hydraulic fluid in the steering actuator all affect the vehicle model. Real-time system identification techniques offer improved estimation and control through more accurate identification of model parameters or even the model itself.
3. *Real-time identification of disturbances.* As mentioned in Ch. 3, the absence of a tractor suspension system meant the tractor was very susceptible to ground disturbances. Similarly, controllers could easily be driven unstable by a sudden increase in ground disturbances. If the disturbance strength was known in real-time, the controller could be tuned to maximize performance while still maintaining stability. A real-time disturbance identification technique could provide such information. This technique would be particularly valuable at higher speeds where controller instability could even become dangerous.
4. *Towed implement control on nonlinear trajectories.* Time did not permit research into controlling a point on the implement along nonlinear trajectories. In practical applications where high accuracy is desired, arcs and spirals will have large radii, and the difference in tracking performance between controlling a point on the tractor vs. a point on the implement will be small. On curved trajectories, however, rapidly changing curves will present a control challenge

when controlling a towed implement. The problem is complicated by bounds on control effort, steer angle, and the angle between the implement and the tractor. Such a controller would be of significant value to farmers.

5. *Collision avoidance.* With the possibility of more than one automatic tractor in the field at any given time, some sort of traffic management system is needed to avoid collisions.
6. *Formation driving.* A completely autonomous tractor presents numerous safety challenges such as obstacle detection and avoidance. An intermediate step towards autonomous tractors could involve human oversight of driverless tractors. A human driver, armed with a remote kill switch, could monitor several autonomous tractors working nearby. Such a scenario would be practical if the vehicles worked as a team in formation. The “master” vehicle could be either human-controller or automatic. Therefore, control algorithms are needed that would force a “slave” vehicle to maintain a prescribed offset from the master vehicle.
7. *Nonlinear optimization routines for real-time control.* The rapid rise of inexpensive desk-top computers means that engineers will have access to increased real-time computing power. Nonlinear optimization algorithms previously thought impractical for real-time control are becoming feasible, and research is needed to develop these algorithms for real-time control.
8. *Path optimization algorithm(s).* Path optimization algorithms are needed to optimize a tractor’s route across the field according to some user-specified performance metric. The algorithm should take into account the vehicle’s dynamic constraints as well as obstacles, field boundaries, and possibly other vehicles in the field. Ideally, the user would specify costs or penalties on such factors as overlap, underlap, total working time, and fuel usage. Once the desired vehicle and implement combination has been specified, the path planning algorithm would specify a global path that would minimize a cost function created from the user-specified penalties. This global path would be composed of lines, arcs,

spirals, and curves, and would be uploaded into the vehicle's computer prior to starting work.

14.3 Implications

CPDGPS offers researchers economical and precise position and attitude sensors. The cost of desk-top computing power is falling rapidly, and researchers now have the power to run sophisticated control algorithms in real-time. These two factors imply several trends:

1. Commercial autonomous vehicles will soon be relatively common, beginning first in agriculture.
2. Computationally intensive control algorithms will become practical real-time control solutions.
3. Industry interest in key fields such as vehicle modeling, real-time system identification and adaptive control, nonlinear optimization, and path generation will increase.

Eventually, vehicles may become fully autonomous. With the removal of the driver, vehicles could be completely redesigned to omit driver-related constraints such as a cab or good road visibility. Farms all over the world may one day see multiple driverless tractors working a single field through the night, cooperating to efficiently accomplish tasks. Tractors will communicate with each other and the farm management office, relaying back information such as yield data, fuel level, soil moisture content, and work time remaining. Raw material usage will be significantly reduced as overlap is eliminated. Specialized irrigation techniques will become more cost-efficient and irrigation water usage will decrease. The computer and GPS hardware to facilitate all of this already exists. This research has provided some of the fundamental algorithms needed to make this vision a reality.

APPENDIX A

Test Equipment

Two Deere and Co. tractors were used during the course of this research: a Model 7800 and a Model 8400. Both tractors used the same positioning and attitude hardware and software. All controls software was written at Stanford.

A.1 Deere and Co. Model 7800

The first test platform used in this research was the Deere and Co. Model 7800 farm tractor shown in Fig. (2.1). As tested, the 7800 had the following characteristics:

- 108 kW (145 hp, measured at power-take-off shaft) 7.6L turbocharged in-line 6-cylinder diesel engine.
- 19 forward gears with forward speed of up to 35 km/h (22 mph).
- Four-wheel drive.
- Air-filled row-crop tires.
- 1,000 kg front ballast, 1,000 kg rear-axle ballast.
- Unballasted weight of 6,500 kg (14,300 lbs).
- Length of 4.4 m (173 in), axle width of 2.8 m (110 in), and height of 3.0 m (116 in).



Figure A.1: Deere and Co. Model 8400 farm tractor.

A.2 Deere and Co. Model 8400

The second test platform used in this research was a Deere and Co. Model 8400 farm tractor shown in Fig. (A.1). As tested, the 8400 had the following characteristics:

- 168 kW (225 hp, measured at power-take-off shaft) 8.1L turbocharged in-line 6-cylinder diesel engine.
- 16 forward gears with forward speed of up to 35 km/h (22 mph).
- Four-wheel drive.
- Air-filled row-crop tires.
- 1,000 kg front ballast, 1,000 kg rear-axle ballast.
- Unballasted weight of 8,500 kg (18,700 lbs).
- Length of 5.2 m (206 in), axle width of 3.0 m (118 in), and height of 3.0 m (120 in).

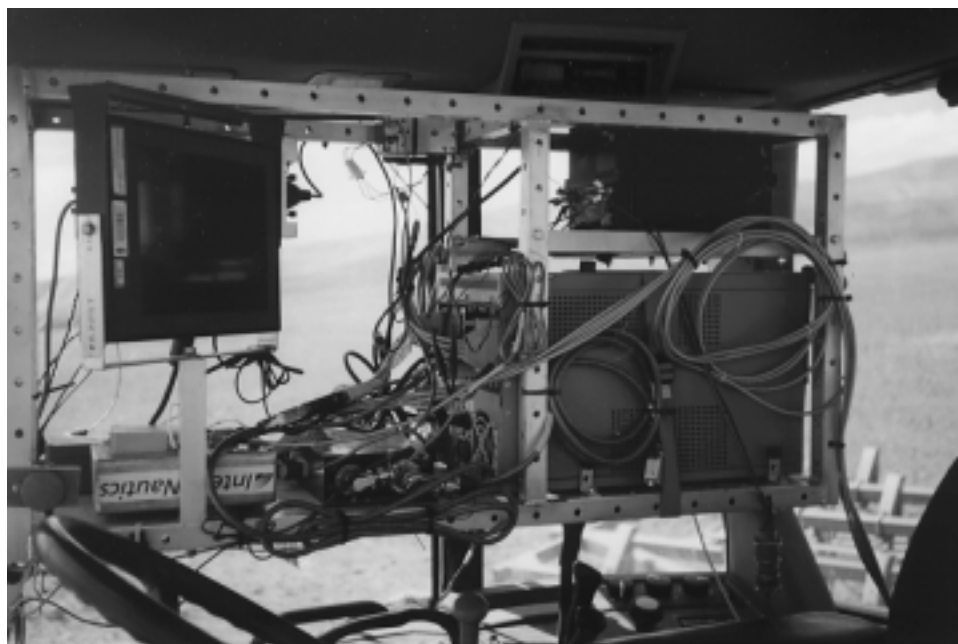


Figure A.2: Equipment mounted inside the 7800.

A.3 Sensor and Actuator Hardware

The IntegriNautics Corporation provided CPDGPS hardware and software to measure the master antenna's location relative to the reference station antenna at 5 Hz. Measurement noise was on the order of 1-1.5 cm (one-sigma) in horizontal and 2-3 cm (one-sigma) in vertical position. The reference station, built by Stanford with IntegriNautics' GPS hardware and software, transmitted GPS corrections to the tractor through Pacific Crest radio modems at 9,600 baud. Trimble Navigation's TANS Vector provided the tractor computer with attitude measurements at 10 Hz to a resolution of 0.1 degrees (one-sigma) using an array of four single-frequency GPS antennas mounted to the roof of the cab. Mounted inside the cab of the 7800 was a 100 MHz PC using the Lynx real-time operating system. Fig. (A.2) shows the rack mounted in the interior of the 7800's cab. The flat-panel display is mounted on the upper left with the GPS receiver beneath it. The TANS Vector sits underneath the wiring, while the computer is located on the lower right. The power supply sits above the

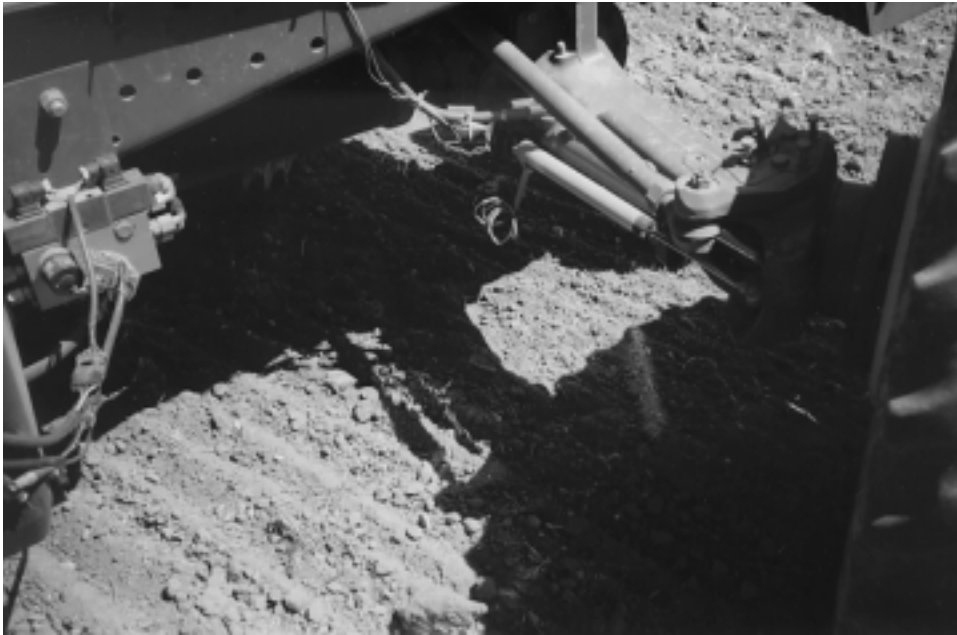


Figure A.3: Electrohydraulic valve (left) and steering angle potentiometer (right) on the 7800.

computer. An Orthman hydraulic valve, shown on the left in Fig. (A.3), actuated the front wheels, and a potentiometer measured the front-wheel angle, both interfacing with the tractor computer through a Motorola MC68HC11 microprocessor.

Bibliography

- [A⁺97] A. Alessandri et al. An Application of the Extended Kalman Filter for Integrated Navigation in Mobile Robots. In *Proceedings of the 1997 American Control Conference*, pages 527–531, Jun 1997. Albuquerque, New Mexico.
- [AM91] B. D. O. Anderson and J. B. Moore. *Optimal Control, Linear Quadratic Methods*. Prentice-Hall, Inc., 1 edition, 1991.
- [B⁺96] P. Bolzern et al. Dynamic Model of a Two-Trailer Articulated Vehicle Subject to Nonholonomic Constraints. *Robotica*, 14(4):445–450, 1996.
- [B⁺97] T. Bell et al. Realistic Autofarming, Closed-Loop Tractor Control Over Irregular Paths Using Kinematic GPS. In *Proceedings of the 9th World Congress of the International Association of Institutes of Navigation*, 1997. November 18-21, 1997, Amsterdam, The Netherlands.
- [B⁺98a] T. Bell et al. Automatic Tractor Row and Contour Control on Sloped Terrain Using Carrier-Phase Differential GPS. In *4th International Conference on Precision Agriculture*, 1998. To be published.
- [B⁺98b] T. Bell et al. Realistic Autofarming, Closed-Loop Tractor Control Over Irregular Paths Using Kinematic GPS. *Journal of Navigation*, 51(3):327–335, 1998. Published by the Royal Institute of Navigation, London, U.K.
- [BDW95] B. Barshan and H. F. Durrant-Whyte. Inertial Navigation Systems for Mobile Robots. *IEEE Transactions on Robotics and Automation*, 11(3):328–342, Jun 1995.

- [Ber95] D. P. Bertsekas. *Nonlinear Programming*. Athena Scientific, 1995.
- [BH75] A. E. Bryson and Y. Ho. *Applied Optimal Control*. Taylor and Francis, 1975.
- [Bot82] D. J. Bottoms. The Tractor Driver's Steering Control Task. *Ergonomics*, 25(1):31–39, 1982.
- [Bra91] R. M. Brach. Vehicle Dynamics Model for Simulation on a Microcomputer. *International Journal of Vehicle Design*, 12(4):404–419, 1991.
- [CS84] D. A. Crolla and H. B. Spencer. Tractor Handling During Control Loss on Sloping Ground. *Vehicle System Dynamics*, 13:1–17, 1984.
- [DW97] A. W. Divelbliss and J. T. Wen. Trajectory Tracking Control of a Car-Trailer System. *IEEE Transactions on Control Systems Technology*, 5(3):269–278, May 1997.
- [F⁺90] G. F. Franklin et al. *Digital Control of Dynamic Systems*. Addison-Wesley Publishing Co., Inc., 2 edition, 1990.
- [G⁺81] P. E. Gill et al. *Practical Optimization*. Academic Press, 1981.
- [G⁺86] P. E. Gill et al. *User's Guide for NPSOL (Version 4.0): A Fortran Package for Non-Linear Programming*. Stanford University, Systems Optimization Lab, Department of Operations Research, Stanford, CA, 94305, 4.0 edition, Jan 1986.
- [God98] J. Godhavn. Nonlinear Tracking of Underactuated Surface Vessels. In *Proceedings of the 1997 American Control Conference*, 1998.
- [GP90] B. Guenter and R. Parent. Computing the Arc Length of Parametric Curves. *IEEE Computer Graphics and Applications*, pages 72–78, May 1990.
- [Gre88] D. T. Greenwood. *Principles of Dynamics*. Prentice Hall, Inc., 2 edition, 1988.

- [GS93] R. Gourdeau and H. M. Schwartz. Adaptive Control of Robotic Manipulators Using an Extended Kalman Filter. *Journal of Dynamic Systems, Measurement, and Control*, 115:203–208, Mar 1993.
- [H⁺97] T. Hashizume et al. Path Tracking Controller for Tractor with Multiple Trailers. In *Proceedings of the 1997 23rd Annual International Conference on Industrial Electronics, Control, and Implementation, IECON*, volume 3, pages 1269–1274, 1997.
- [Hil76] F. B. Hildebrand. *Advanced Calculus for Applications*. Prentice Hall, Inc., 2 edition, 1976.
- [HJ77] W. E. Holley and A. E. Bryson Jr. Wind Modeling and Lateral Control for Automatic Landing. *Journal of Spacecraft*, 14(2):65–72, Feb 1977.
- [HJ80] P. J. Hartley and C. J. Judd. Parameterization and Shape of B-spline Curves for CAD. *Computer-Aided Design*, 12(5):235–238, Sept 1980.
- [Joh97] P. Johnson. Boggabilla GPS Boosts Efficiency. *The Land*, (3884), Sept 4 1997.
- [Jun97] J. L. Junkins. Adventures on The Interface of Dynamics and Control. *Journal of Guidance, Control, and Dynamics*, 20(6):1058–1071, 1997.
- [K⁺71] P. G. Kaminski et al. Discrete Square Root Filtering, A Survey of Current Techniques. *IEEE Transaction on Automatic Control*, 16(6):727–736, Dec 1971.
- [K⁺98] M.-Joon Kim et al. Autopilot Design for BTT Missiles Using Receding Horizon Predictive Control Scheme. In *Proceedings of the 1997 American Control Conference*, 1998.
- [Kas95] N. J. Kasdin. Discrete Simulation of Colored Noise and Stochastic Processes and $1/f^\alpha$ Power Law Noise Generation. *Proceedings of the IEEE*, 83(5):802–827, May 1995.

- [Kre93] E. Kreyszig. *Advanced Engineering Mathematics*. John Wiley & Sons, Inc., 7 edition, 1993.
- [Lat91] J.-C. Latombe. *Robot Motion Planning*. Kluwer Academic Publishers, 1991.
- [Lau79] A. J. Laub. A Schur Method for Solving Algebraic Riccati Equations. *IEEE Transactions on Automatic Control*, pages 913–912, 1979.
- [LS95] F. L. Lewis and V. L. Syrmos. *Optimal Control*. John Wiley and Sons, Inc., 2 edition, 1995.
- [M⁺92] Y. Mohamoud et al. Optimum Center-Pivot Irrigation System Design with Tillage Effects. *Journal of Irrigation and Drainage Engineering*, 118(2):291–305, 1992.
- [M⁺95] J. Mononen et al. Development of a Control System for Autonomous Land Vehicles. *SAE Technical Paper Series*, (952150), 1995.
- [Mai97] R. Mailer. Innovative Navigation System puts Farmers on the Right Track. *Australian Grain*, 7(3):3–6, Jun–Jul 1997.
- [May82] P. S. Maybeck. *Stochastic Models, Estimation, and Control*, volume 2. Academic Press, 1982.
- [N⁺96] T. Nieminen et al. Development of a Control System for Autonomous Agricultural Vehicles, 1996. Paper 96A-094, AgEng '96, Madrid, Spain.
- [N⁺97a] N. Noguchi et al. Development of a Tillage Robot Using a Position Sensing System and a Geomagnetic Direction Sensor. In *Proceedings of the 1997 ASAE Annual International Meeting*, 1997. ASAE Paper No. 973090, written for presentation at the 1997 ASAE Annual International Meeting, Minneapolis, Minnesota, August 10–14, 1997.
- [N⁺97b] N. Noguchi et al. Development of an Agricultural Mobile Robot using a Geomagnetic Direction Sensor and Image Sensors. *Journal of Agricultural Engineering Research*, 67:1–15, 1997.

- [Nas97] J. Nason. Boggabilla GPS Cuts Farm Costs. *Queensland Country Life*, Aug 21 1997.
- [Nel89] W. L. Nelson. Continuous Steering-Function Control of Robot Carts. *IEEE Transactions on Industrial Electronics*, 36(3):330–337, Aug 1989.
- [O’C97] M. L. O’Connor. *Carrier-Phase Differential GPS for Automatic Control of Land Vehicles*. PhD thesis, Stanford University, 1997.
- [Owe82] G. M. Owen. A Tractor Handling Study. *Vehicle System Dynamics*, 11:215–240, 1982.
- [Par95] B. W. Parkinson. Class notes for Dynamics course (AA242), Autumn 1995. Stanford University, Department of Aeronautics and Astronautics.
- [Per96] B. Pervan. *Navigation Integrity for Aircraft Precision Landing Using the Global Positioning System*. PhD thesis, Stanford University, March 1996.
- [PF97] W. Pye and K. Fitzpatrick. Robotics in the Field. *Industrial Vehicle Technology*, pages 70–74, 1997.
- [PS96] B. W. Parkinson and J. J. Spilker, editors. *Global Positioning System: Theory and Applications*, volume 163 of *Progress in Astronautics and Aeronautics*. American Institute of Aeronautics and Astronautics, Inc., 1996.
- [Rid98] A. Rider. Private-Public Sector Cooperation can be Key to Research. *Co-operative Research*, 1998. Paper presented on May 12, 1998, at the Diamond Jubilee Conference of the Institute of Agricultural Engineers held at Silsoe College, Cranfield University, Bedford, U.K.
- [RS87] J. F. Reid and S. W. Searcy. Vision-Based Guidance of an Agricultural Tractor. *IEEE Control Systems Magazine*, 7(2):39–43, Apr 1987.
- [S⁺] T. S. Stombaugh et al. Automatic Guidance of Agricultural Vehicles at High Field Speeds. ASAE Meeting Presentation. Paper No. 983110.

- [S⁺95] P. Svestka et al. Exact Motion Planning for Tractor-Trailer Robots. In *Proceedings of the 1995 IEEE International Conference on Robotics and Automation*, volume 3, pages 2445–2450, 1995.
- [SC87] J. L. Speyer and E. Z. Crues. On-Line Aircraft State and Stability Derivative Estimation Using the Modified-Gain Extended Kalman Filter. *Journal of Guidance, Control, and Dynamics*, 10(3):262–268, 1987.
- [SF89] A. Stribersky and P. S. Fancher. The Nonlinear Behavior of Heavy-Duty Truck Combinations with Respect to Straightline Stability. *Journal of Dynamic Systems, Measurement, and Control*, 111(4):577–582, Dec 1989.
- [Sim96] V. Sima. *Algorithms for Linear Quadratic Optimization*, volume 200 of *Pure and Applied Mathematics: A Series of Monographs and Textbooks*. Marcel Dekker, Inc., 1996.
- [SL91] J. J. Slotine and W. Li. *Applied Nonlinear Control*. Prentice Hall, 1991.
- [SO95] D. H. Shin and A. Ollero. Mobile Robot Path Planning for Fine-Grained and Smooth Path Specifications. *Journal of Robotic Systems*, 12(7):491–503, 1995.
- [Spa95] H. Spaeth. *One-Dimensional Spline Interpolation Algorithms*. A K Peters, 1995.
- [Spl76] W. E. Splinter. Center-Pivot Irrigation. *Scientific American*, 234(6):90–99, June 1976.
- [ST82] R. J. Sharpe and R. W. Thorne. Numerical Method for Extracting an Arc Length Parameterization from Parametric Curves. *Computer-Aided Design*, 14(2):79–81, Mar 1982.
- [Ste86] R. F. Stengel. *Optimal Control and Estimation*. Dover Publications, Inc., 1986.

- [T⁺85] B. R. Tennes et al. Microcomputers Automate Tractor Steering. *Hydraulics and Pneumatics*, 38(3):52–55, Mar 1985.
- [Tay82] J. R. Taylor. *An Introduction to Error Analysis*. University Science Books, 1982.
- [Tom93] M. Tomizuka. Feedforward Digital Tracking Controllers for Motion Control Applications. *Advanced Robotics*, 7(6):575–586, 1993.
- [vL78] C. F. van Loan. Computing Integrals Involving the Matrix Exponential. *IEEE Transactions on Automatic Control*, pages 395–404, Jun 1978.
- [Vyn98] K. Vyn. Advanced Technology in Farm Equipment. *SAE Off-Highway Engineering*, pages 99–104, Aug 1998.
- [vZ98] R. P. van Zuydam. Centimetre-Precision Guidance of Agricultural Implements in the Open Field by Means of real Time Kinematic (RTK) GPS. In *4th International Conference on Precision Agriculture*, 1998. To be published.
- [WB98] E. White and R. Buick. Comparing GPS Guidance with Foam Marker Guidance. In *4th International Conference on Precision Agriculture*, 1998. To be published.
- [Won93] J. Y. Wong. *Theory of Ground Vehicles*. John Wiley & Sons, Inc., 1993.