Path Tracking Control for Tracked Based Machines Using Sensor Fusion

P. Saeedi, D. Kusalovic, P. Jacobsen, K. Ardron, P. Lawrence and D. G. Lowe
Department of Electrical and Computer Engineering,
Department of Computer Science, University of British Columbia
Vancouver, BC, V6T 1Z4, Canada
parvanes@ece.ubc.ca, peterl@ece.ubc.ca, lowe@cs.ubc.ca

Abstract

This paper describes a control system using different sensor modalities for moving a tracked vehicle (an excavator) from a starting position to a goal position. The system includes a path planner, a path-tracking controller, a cross-coupling controller and a vision based slippage controller. Experiments are conducted to test and verify the performance of the presented system. An analysis of the results shows that improvement is achieved in both path-tracking accuracy and slippage control problems.

1 Introduction

Tracked vehicles are widely used in industries such as forestry, construction and mining. These machines are used for a variety of tasks, such as lifting and carrying loads, excavation and ground leveling. Autonomous controls for driving or assisting humans in operating these machines can potentially improve the operational safety and efficiency. Much research has gone toward controlling vehicle movement to reduce human interaction when the vehicle is performing a task [1] [8]. The most common level of automation, for these type of vehicles, is achieved by teleoperation, in which the operator controls the vehicle remotely [6] [10]. The ultimate goal is to have a completely autonomous vehicle by eliminating the need for constant, low level, human guidance. Achieving this goal in natural environments requires planning every movement, to avoid any obstacles and to locate the vehicle at each time with respect to a global coordinate system. With the application of a good control scheme, the effect of human error can be minimized or completely removed, and more consistent operation of the vehicle can be achieved to increase efficiency.

The main goal of this work has been to move a tracked vehicle safely from a starting point to a target point in an unstructured outdoor environment. To achieve this goal, several subsystems such as path planning, path tracking and 3D motion tracking, were designed and implemented. Since for tracked vehicles the slippage phenomenon is one of the most important sources of inaccuracy, a slippage detection and correction mechanism was also designed and implemented.

2 Track Control System Overview

A block diagram of the system is shown in Figure 1. The human operator, at the top control level, selects a desired goal posture. This position, along with a map of the environment are passed to a path planner unit. The path planner designs a collision free optimal path from the current posture of the vehicle. By comparing the current position of the vehicle (from the flowrate sensors) with the path to be followed, the reference translational and rotational velocities for the mobile robot are computed. The vehicle is either commanded to continue driving along the path, when on the desired path, or made to converge back to the desired path if it has strayed. The cross-coupling motion controller (CCMC) controls the heading error

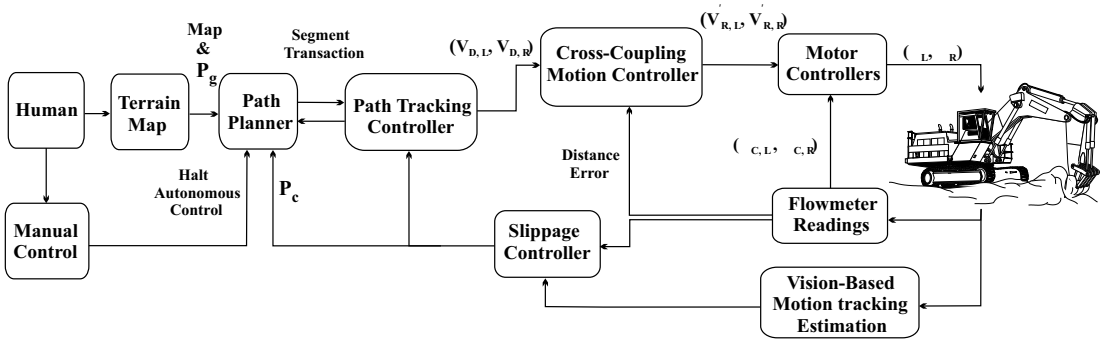

Figure 1: Track control system block diagram.

directly, using the left and right track reference speeds, as well as the accumulated distance error between the two tracks. The vision-based tracking system processes consecutive trinocular stereo sets of images to detect and track correspondences to the most stable points in the environment. Through measuring the motion of these features, the $3D$ trajectory of the camera, and hence, the trajectory of the vehicle is estimated. If the estimated trajectory by the vision system differs substantially from that of the dead-reckoning sensors, for the same time period, a slippage value is calculated in the slippage controller. The slippage value is used to compute a scaling factor to reduce the left or right track reference speeds until the slippage is eliminated.

3 Path Planning Controller

There are two main objectives associated with the design of the path planner. First, it must find an obstacle free path, from a starting point to a goal point. Second, it has to minimize soil disturbance, in forestry applications for example, while operating heavy machinery outdoors. The inputs to the system are the elevation map of the work space, as well as cost constraints. The obstacles are defined as either objects that block a specific path, or as terrain that is not traversable due to mechanical constraints of the machine (for example steep slopes). In terms of efficiency, the shortest path length is considered to be the main criterion. Most likely, a shorter path can be executed faster and consumes less fuel. The resolution of the environment grid, $n \times m$, and the type of cost functions are determined in the set up process by the operator.

3.1 Cost Map Generation

One of the requirements of the path planner is to optimize the found path, based on several conditions such as length, slope and soil disturbance. However, employing a multiple objective search can significantly impact the required memory and increase running time dramatically. Also, some of the constraints in the cost functions may be mutually conflicting. For these reasons, for every existing path a cost value is associated with a weighted sum of individual constraints [2]. A maximum of six weight coefficient combinations is allowed in the design. The cost function associated with every traversable arc in the map is a scalar calculated from:

$$c(m,n) = \lambda_1 . \text{Arc length} + \lambda_2 . \frac{\text{Elevation change}}{\text{Arc length}} \tag{1}$$

$$\text{Arc length} = \sqrt{(x_m - x_n)^2 + (y_m - y_n)^2 + (z_m - z_n)^2} \quad , \quad \text{Elevation change} = |z_m - z_n| \tag{2}$$

The relationship between coefficients λ_1 and λ_2 can be expressed by:

$$0 \leq \lambda_1 \ , \ \lambda_2 \leq 1 \qquad , \qquad \lambda_1 + \lambda_2 = 1 \tag{3}$$

3.2 Search for the Best Solution

The accuracy and efficiency of the path planning controller is mainly dependent on the search method's convergence and computational complexity. The A* approach was chosen for its robustness in off-road route planning [5]. The algorithm begins at a start node and then iteratively checks surrounding nodes for their accessibility. When a traversable node is visited for the first time, a backpointer to the previous node, as well as the cost of the path traveled so far, is estimated and assigned to the node. The grid map of the workspace confines the angle changes with respect to the previous arc to 0^o, 45^o, 90^o, 135^o and 180^o. To avoid sharp angle changes, which can cause undesirable disruption of delicate forest soil, two strategies are considered.

First, sharp turns are banned (rotation angles ≥ 90). Whenever a new path is discovered, it is checked for any sharp angles with the two node's predecessors. If any sharp angle exists, then an obstacle is reported and the backpointer is not switched. The second strategy minimizes the number of turns in general. This is enforced by employing one more test and comparing the number of turns and switching to a neighboring node when a new path is discovered . One of the safety features included in the system is implemented by expanding the edges of found obstacles through grid node displacement. This safety feature is paid for by the possibility of not finding a solution even when one may exist. To overcome any system failure due to that, a second search is conducted after the obstacles' borders shrink back to their original positions.

4 Path Tracking Controller

Figure 2-a represents the block diagram of the path tracking controller. This controller maintains the mobile vehicle on a desired path. The inputs to this controller are the current position and orientation of the vehicle, and the desired path segments. Using these two pieces of information, the controller estimates the motion needed to reach the end of the desired path segment. Upon reaching the end of the desired path segment, this controller transmits a path segment request for the next path segment to be traversed.

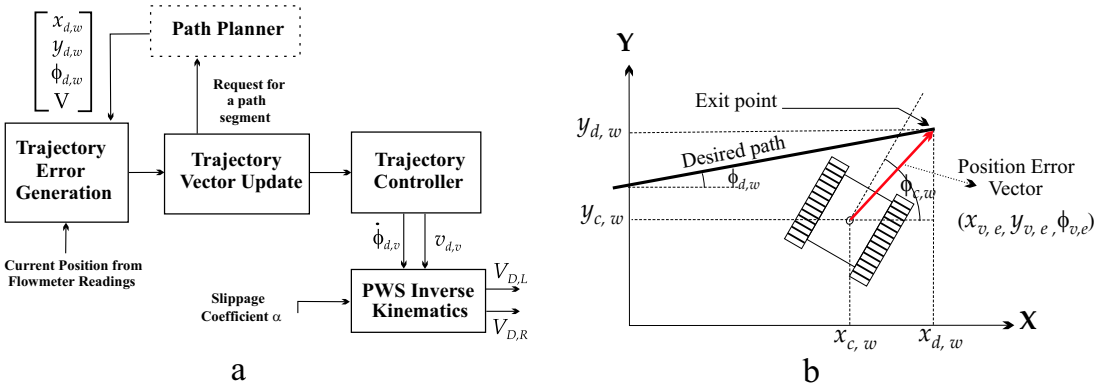

Figure 2: a-Path Tracking Controller block diagram. b-Trajectory error generation.

4.1 Trajectory Error Generation

The trajectory error generation estimates an error vector in the vehicle frame using the desired destination and the current position (Figure 2-b). The desired destination is provided in the system input, while the current trajectory is estimated through the readings of the flowrate sensors. The generated trajectory error can be expressed by

$$\begin{bmatrix} x_{v,e} \\ y_{v,e} \\ \phi_{v,e} \end{bmatrix} = \begin{bmatrix} cos(\phi_{d,w}) & sin(\phi_{d,w}) & 0 \\ sin(\phi_{d,w}) & -cos(\phi_{d,w}) & 0 \\ 0 & 0 & -1 \end{bmatrix} \cdot \begin{bmatrix} x_{d,w} - x_{c,w} \\ y_{d,w} - y_{c,w} \\ \phi_{d,w} - \phi_{c,w} \end{bmatrix} \quad (4)$$

where sub-indices (d, w) and (c, w) represent the destination and current position in world coordinate frame, and (v, e) the vehicle error frame.

4.2 Trajectory Vector Update

The trajectory vector update is responsible for requesting a new segment when the end of the current segment is reached. This is achieved when the value of $x_{v,e}$ approaches zero. At each moment, the value of $x_{v,e}$ is carefully observed to ensure that an error in $y_{v,e}$ or $\phi_{v,e}$ would have no influence on the segment direction. The trajectory vector update also derives the error changes of $\frac{dx_{v,e}}{T_s}$, $\frac{dy_{v,e}}{T_s}$ and $\frac{d\phi_{v,e}}{T_s}$ that are used in the trajectory controller.

4.3 Trajectory Controller

The rotational and translational velocities of the vehicle, $(V_{d,v}, \dot{\phi}_{d,v})$, are controlled by this controller. The translational control is performed by determining the desired translational velocity of the vehicle from $x_{v,e}$ and $y_{v,e}$. The orientation control is performed by using $y_{v,e}$ and $\phi_{v,e}$ and their derivatives. The goal is to converge the inputs to zero.

4.3.1 Translational Velocity Control

Regardless of the position and orientation, this controller is responsible for maintaining a desired acceleration during the starting time and velocity change when converging to a desired translational velocity. Here, the sum of changes in x and y of the vehicle are measured to estimate the actual velocity of the vehicle. The control rules for converging to, or maintaining, a desired action can be expressed as

$$\frac{dv_{d,v}}{T_s} = \begin{cases} 0, & \text{if } v = +V_{max} \text{ and } v < v_{des} \\ 0, & \text{if } v = -V_{max} \text{ and } v > v_{des} \\ -A_v, & \text{if } v > v_{des} \\ A_v, & \text{if } v < v_{des} \end{cases} \tag{5}$$

These equations state that the output of the translational velocity controller, $v_{d,v}$, changes by comparing the current positional change with the desired translational velocity. It depends not only on a prespecified desired velocity, but also on the current values of $y_{v,e}$ and $\phi_{v,e}$. Therefore, the translational velocity of the vehicle decreases when not on the specified path. This may happen when a path segment changes or track slippage occurs, and when a turn is attempted. At each turning point the translational velocity has to be decreased to prevent the vehicle from overshooting when converging to a new path. In order to ensure a stop at an exact stop position with a specific deceleration, the following rules [12] are adopted:

$$v_{d,s}(x_{v,e}) = \text{sign}(x_{v,e}).\sqrt{2.A_{Stop}.|x_{v,e}|} \tag{6}$$

$$\frac{dv_{d,v}}{T_s} = \begin{cases} 0, & \text{if } v_{d,s}(x_{v,e}) = v_{des} \\ A_{Stop}, & \text{if } v < v_{d,s}(x_{v,e}) \\ -A_{Stop}, & \text{if } v > v_{d,s}(x_{v,e}) \end{cases} \tag{7}$$

4.3.2 Rotational Velocity Control

This controller forces the vehicle to move on a desired path by controlling the rotational velocity, $\dot{\phi}_{d,v}$. For this purpose, two fuzzy PD-regulators, one for $y_{v,e}$ and its derivative and another for $\phi_{v,e}$ and its derivative, are implemented. The input space of the fuzzy-PD controllers are $y_{v,e}$, $\frac{dy_{v,e}}{T_s}$, $\phi_{v,e}$ and $\frac{d\phi_{v,e}}{T_s}$. Variables y and ϕ are divided into five membership functions (MFs) and $\frac{dy_{v,e}}{T_s}$ and $\frac{d\phi_{v,e}}{T_s}$ into three. Similarly the universe of y, dy and $d\phi$ are partitioned. The output space for this controller is defined by variable $\dot{\phi}_{d,v}$ with a defuzzication of the centroid-based method [9]. In order to have smooth output changes, the membership functions are equally

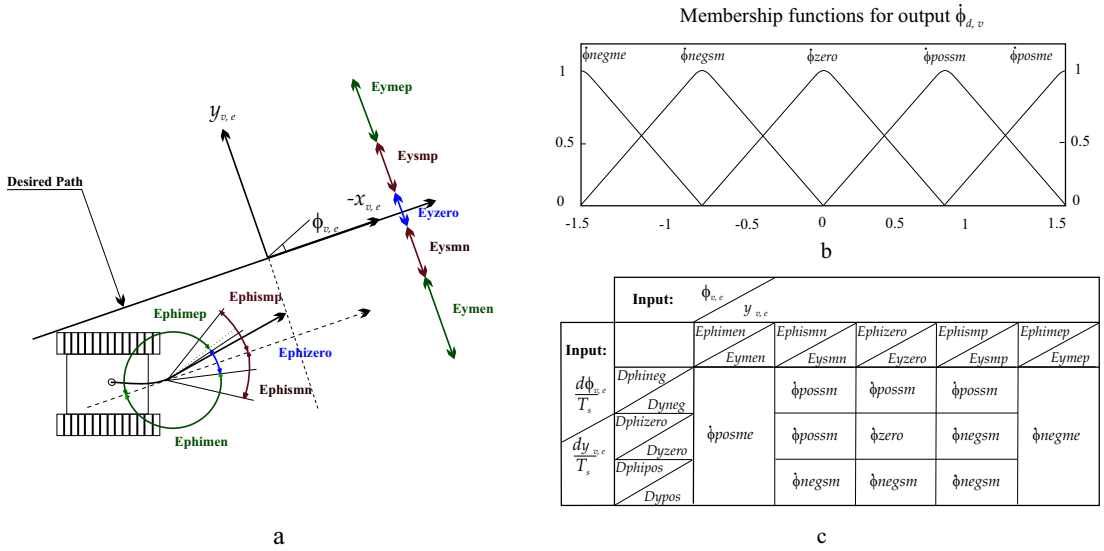

Figure 3: Rotational velocity controller input (a) and output (b and c) spaces.

distributed as shown in Figure 3-b. The output space mapping function rules are represented in Figure 3-c.

4.4 PWS Inverse Kinematics

The Power Wheel Steering (PWS) Inverse Kinematic for the conversion from the trajectory controller output, $[v_{d,v}, \dot{\phi}_{d,v}]$, to the desired translational velocities of the right and left tracks, $[V_{D,L}, V_{D,R}]$, are defined by:

$$\begin{bmatrix} V_{D,L} \\ V_{D,R} \end{bmatrix} = \begin{bmatrix} 1-\alpha_L \\ 1-\alpha_R \end{bmatrix} \cdot \begin{bmatrix} v_{d,v} - \dot{\phi}_{d,v} . \frac{D_{track}}{2} \\ v_{d,v} + \dot{\phi}_{d,v} . \frac{D_{track}}{2} \end{bmatrix} \tag{8}$$

D_{track} is the distance between tracks (for our excavator it measures as 1.275m). α_R and α_L are the detected slippage coefficients, bounded by [0 1].

5 Cross-Coupling Motion Control

A tracked vehicle moves by the rotation of two steel or rubber tracks that are driven and controlled by two independent motors. The steering action is accomplished by the difference in speed of the two tracks. The velocities for such a vehicle can be expressed by [11]

$$\dot{x} = \frac{v_R + v_L}{2} sin\theta \quad , \quad \dot{y} = \frac{v_R + v_L}{2} cos\theta \quad , \quad \dot{\theta} = \frac{v_R - v_L}{D_{track}} \tag{9}$$

where x, y and θ show the position and the heading of the vehicle in the world coordinates, $\dot{x}$ and $\dot{y}$ describe the translational velocities, and $\dot{\theta}$ represents the angular velocity. The linear velocities of the left and right tracks are represented by v_L and v_R and D_{track} is the distance between the two tracks. Several external and internal sources can effect the accuracy of the vehicle's motion. The heading error describes the robot's orientation error, while the tracking error shows the distance between the actual and desired vehicle positions. The heading error is the most disturbing, since it can increase the tracking error over time. Also, since each track works in an individual loop and receives no information about the other, when a disturbance happens in one loop and causes an error in the motion of the vehicle, it is corrected only in

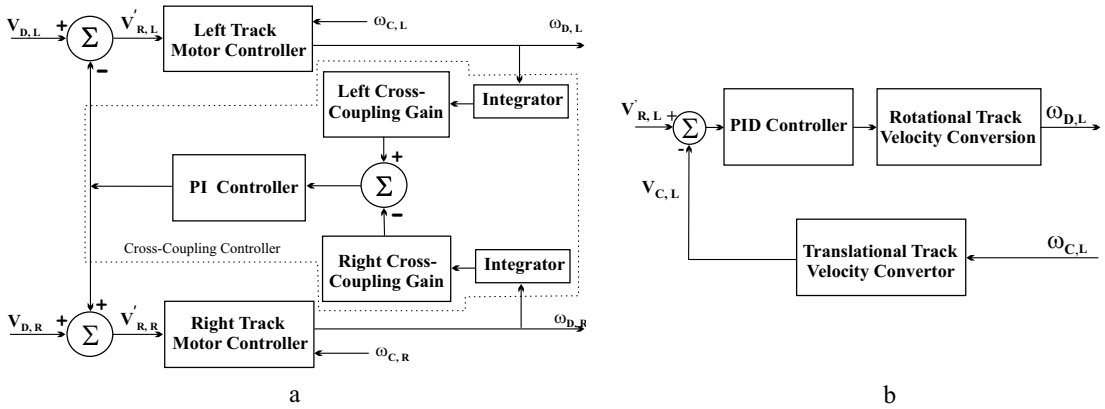

Figure 4: a-Cross-Coupling Controller block diagram. b-Block diagram of the motor controller.

its own loop, while the other loop caries on as before. The cross-coupling controller regulates orientation error of the vehicle by exchanging feedback information between the control loops [4].

The idea of cross-coupling control is based on calculation of the actual error, multiplying that by a controller gain, and feeding this error back to the individual loops [3], Figure 4-a. The controller gains for the left and right track are computed from:

$$C_L = 1 \quad , \quad C_R = \frac{v_L}{v_R} = \frac{R - \frac{D_{track}}{2R}}{R + \frac{D_{track}}{2R}} \tag{10}$$

where R is the radius of the circle path.

6 Motor Controller

The rotational velocity of each track is controlled by a PID regulator based upon the desired track translational velocity, $V'_{R,L}$ or $V'_{R,R}$, specified by the cross-coupling controller and feedback, $\omega_{C,L}$ or $\omega_{C,R}$, from the corresponding track velocity sensor, flow sensor, (Figure 4-b).

7 Vision-Based Motion Tracking

We developed a vision-based 3D trajectory tracking system [7] suitable for an autonomous robot vehicle. The system includes a head with three on-board CCD cameras, which can be mounted anywhere on the mobile vehicle. For this work, the camera looks down and straight ahead of the excavator. By processing consecutive trinocular sets of precisely aligned and rectified images, the local 3D trajectory of the vehicle in an unstructured environment can be tracked. The system does not rely on any prior knowledge of the environment or any specific landmark in the scene. Further, the scene is mostly constructed of rigid objects, although if there are a few small moving objects the system still relies on static information. The motion of the robot is also assumed to be limited in acceleration. The system consists of the following components:

1. Feature Extraction: meaningful features from the scene are detected.
2. Stereo Vision: a 3D representation of the extracted features within the scene is obtained from a trinocular set of stereo images.
3. Feature Tracking: the features are matched using a multi-stage matching process.
4. Motion Estimation: relative motion of the camera is estimated in a reference frame.

5. Position Refinement: the 3D world feature locations are refined by combining all previous geometric measurements of the same features.

8 Slippage Controller

One major problem with tracked robots such as excavators is the track slippage during sudden starts or stops, and over various types of surfaces. Slippage usually occurs when one or both tracks loses traction with the ground surface, which makes the readings of the dead-reckoning position odometry erroneous. Moreover, when different amounts of slippage occur between left and right tracks, the tracked vehicle does not follow the appropriate curvature of the desired path. With the use of only local, dead-reckoning sensors, the interaction between machine tracks and the ground, and thus slippage, cannot be detected. To overcome this problem, a novel, vision-based 3D motion tracking system is integrated with the existing excavator sensors.

Slippage is now detected by comparing the track distance traveled in a specified time period, as measured by the track flow sensors, with that estimated by the vision system. Instead of relying on the dead-reckoning sensor for measuring the traveled distance, a flowmeter sensor is used. This sensor is used because readings of odometers for outdoor uneven surfaces are often not very reliable. The individual left and right track distances, measured by vision system:

$$\Delta d_{L,vision} = V_{excav,vision}\Delta T - \frac{D_{track}}{2}\dot{\phi}_{excav,vision}\Delta T \tag{11}$$

$$\Delta d_{R,vision} = V_{excav,vision}\Delta T + \frac{D_{track}}{2}\dot{\phi}_{excav,vision}\Delta T \tag{12}$$

Here ΔT represents the time between the two frames, 50ms. During a typical instance of slippage, the distance measured by the respective track hydraulic flowmeter is larger than that estimated by the vision system. The flowmeter readings for individual left and right tracks are $d_{L,flow}$ and $d_{R,flow}$. The track slippage, therefore, can be defined by:

$$\alpha_i = Slip_i = \frac{\Delta d_{i,flow} - \Delta d_{i,vision}}{\Delta d_{i,vision}} \times 100 \tag{13}$$

Here the subscript i can be either L or R for the right or left track respectively. As the slippage value varies in magnitude between 0% to 100%, it represents conditions of zero track slippage.

9 Experimental Results

Figure 5 demonstrates the performance of the system in moving the vehicle on a desired path. In this experiment, the vehicle moves along the dotted line, from starting point A to the goal point B. The vehicle goes through slippage at the beginning (solid line). This slippage changes the orientation of the vehicle in an unwanted direction. Through slippage detection and control, the orientation is corrected shortly after. The displayed results are the average values for three individual executions. The accumulated positional error along x and y directions are less than 7% with a small heading error of 2%.

10 Conclusions and Future Work

A novel approach is presented in this paper for path tracking with slippage control. The approach is general and can be applied to any tracked vehicle. We employed a multi-cost A^* based path planner that finds the best feasible path using constraints of the shortest length and minimum soil disturbance. The system includes a trajectory controller that employs a non-linear approach for the non-linear problem of the path tracking by employing a fuzzy logic analysis. The existing tolerance in fuzzy logic for dealing with imprecision and uncertainties makes this system suitable for outdoor environments such as natural terrains. A cross-coupling controller has provided an ability to correct tracking errors for the left and right tracks and to minimize the effects of the disturbance on heading direction. Since the desired and the actual translational velocities of the machine can differ due to slippage phenomenon, a novel system was successfully implemented which uses vision and track motion to solve the slippage problem.

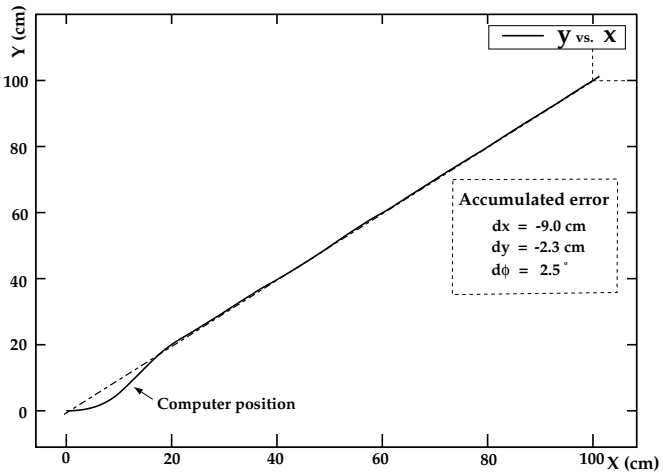

Figure 5: Slippage control results for the right track.

11 Acknowledgments

The authors acknowledge the support of NSERC and IRIS/PRECARN Network of Centers of Excellence.

References

[1] A. Hemami and M. Mehrabi. On the steering control of automated vehicles. *IEEE Conference on Intelligent Transportation System*, pages 266–271, 1997.

[2] R.L. Keeney and H. H. Raiffa. *Decisions with Multiple Objectives: Preference and Value Tradeoffs.* John Wiley & Sons, 1976.

[3] K.Srinivasan and P.K.Kulkarni. Cross-Coupled Control of Biaxial Feed Drive Servomechanisms. *ASME Journal of Dynamic Systems, Measurement and Control*, pages 225–232, 1990.

[4] Y. Koren L. Feng and J. Borenstein. Cross-Coupling Motion Controller for Mobile Robots. *IEEE Control Systems Magazine*, pages 35–43, 1993.

[5] J. Michell. An Algorithmic Approach to Some Problems in Terrain Navigation. *Artificial Intelligence, 37, 171–197*, 1988.

[6] A.D. Nease and E.F. Alexander. Air Force Construction Automation/Robotics. *In Proc. 10th International Symposium on Automation and Robotics in Construction*, 1993.

[7] P. Saeedi, P. Lawrence, and D. Lowe. 3D motion tracking of a mobile robot in a natural environment. *IEEE International Conference on Robotics and Automation*, pages 1682–1687, 2000.

[8] S. Singh. State of the Art in Automation of Earthmoving. *ASCE Journal of Aerospace Engineering, Vol 10, Number 4*, 1997.

[9] M. Jamshidi T. Ross and N. Vadiee. *Fuzzy Logic and Control.* Prentice Hall, 1993.

[10] D. H. Thompson, B. L. Burks, and S. M. Killough. Remote Excavation Using the Telerobotic Small Emplacement Excavator. *Proceedings of the American Nuclear Society Fifth Topical Meeting on Robotics and Remote Systems, Knoxville, Tenn.*, 1993.

[11] C. Ming Wang. Location Estimation and Uncertainty Analysis for Mobile Robots. *IEEE Conference on Robotics and Automation*, pages 1230–1235, 1988.

[12] A. Nilipour Y. Kanayama and C.A. Lelm. A locomotion control method for autonomous vehicles. *IEEE Conference on Robotics and Automation*, pages 1315–1317, 1988.