Multi-Sensor Track Controls for Excavator Based Machines

P. Saeedi, P.D. Lawrence, D. G. Lowe and P. Jacobsen
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 multi-sensor control system for motion control of a tracked vehicle (an excavator). The system includes several controllers that collaborate to move the mobile vehicle on a predefined path. A fuzzy logic path-tracking controller estimates the rotational and translational velocities for the vehicle to move along the predesigned path. A cross-coupling controller corrects the possible orientation error that may occur when moving along curvatures. A vision-based motion tracking system finds the 3D motion of the vehicle as it moves in the working environment. Finally, a specially designed slippage controller detects and corrects slippage by comparing the motion through reading of flowmeters and the vision system. Experiments are conducted to test and verify 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, such as excavator-type machines, are widely used in industries such as forestry, construction and mining. These machines are used for a variety of tasks, such as carrying loads, digging ground, straight traction, and ground leveling. Autonomous controls for driving or assisting humans in operating these machines can improve the operation safety and efficiency. Much research has gone toward controlling vehicle movement so that only partial or no operator interaction is required while the vehicle is performing a task [4] [15]. The most common level of automation, for these type of vehicles, is achieved by teleoperation, in which the operator controls the vehicle remotely [11] [18]. 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 obstacle 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.

Numerous methods are developed to track trajectories and paths outdoors [10] [21] [5] [16]. Some of these methods implement non-linear trajectory control algorithms using the difference between the actual and virtual reference positions. Others accomplish the task by generating error vectors from the lateral displacement and heading errors. Path tracking is also performed using feedforward compensation for the steering mechanism to provide anticipatory control of the steering lag.

The main goal of this work is to move a tracked vehicle along a known path in an unstructured outdoor environment. This paper is organized as follows. First a short overview of the system is presented in Section 2. The design and implementation of the path tracking controller is described in Section 3. The cross-coupling motion controller is studied in Section 4. The motor controller is addressed in Section 5. Section 6 represents the vision based motion tracking system. The slippage controller is discussed in Section 7. Experimental results are represented in Section 8, and conclusions and future work are outlined in Section 9.

2 Track Control System Overview

A block diagram of the semi-autonomous track control system is shown in Figure 1.

The input to the system is the predefined path provided either by an operator or a path planning controller. At each time, a path segment is passed to the path tracking controller. By comparing the current position of the vehicle (obtained incrementally from

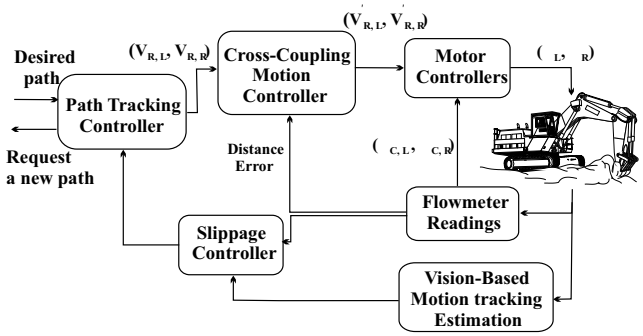

Figure 1: Track control system block diagram.

the rate flow 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. When the vehicle reaches the end of the current segment, the path tracking controller requests the next path segment.

The cross-coupling motion controller (CCMC) controls the heading error 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. Using stereo algorithm and the $2D$ displacements of similar image features in different frames, the $3D$ trajectory of the camera, and hence, 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 Tracking Controller

Figure 2 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.

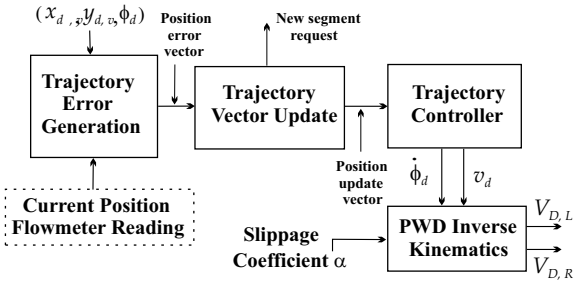

Figure 2: Path Tracking Controller block diagram.

3.1 Trajectory Error Generation

The trajectory error generation estimates an error vector in the vehicle frame using the desired destination and the current position (Figure 3). The desired destination is provided in the system input, while the current trajectory is estimated through the readings of the resolvers.

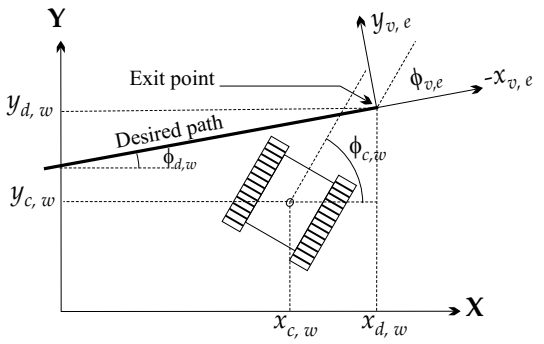

Figure 3: Trajectory error generation.

The generated trajectory error can be expressed by

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

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. Equation 1 represents the required positional correction before reaching to the end of the current path segment.

3.2 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.

3.3 Trajectory Controller

The rotational and translational velocities of the vehicle, $(V_d, \dot{\phi}_d)$, are controlled by this controller. The translational control is performed by determining the desired translational velocity of the vehicle from the changes in $x_{v,e}$ 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

$$y_{v,e} = 0\,, \quad \frac{dy_{v,e}}{T_s} = 0\,, \quad \phi_{v,e} = 0\,, \quad \frac{d\phi_{v,e}}{T_s} = 0 \tag{2}$$

3.4 Translational Velocity Controller

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

$$\frac{dv_{d,v}}{T_s} = \begin{cases} 0, & \text{if } v = +V_{max} \text{ and } v < v_{des} \\ 0, & \text{if } v = -V_{max} \text{ and } v > v_{des} \\ -A_v, & \text{if } v > v_{des} \\ A_v, & \text{if } v < v_{des} \end{cases} \tag{3}$$

$$v = \sqrt{\left(\frac{dx_{v,e}}{T_s}\right)^2 + \left(\frac{dy_{v,e}}{T_s}\right)^2} \tag{4}$$

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 [20] are adopted:

$$v_{d,s}(x_{v,e}) = \text{sign}(x_{v,e}).\sqrt{2.A_{Stop}.|x_{v,e}|} \tag{5}$$

$$\frac{dv_{d,v}}{T_s} = \begin{cases} 0, & \text{if } v_{d,s}(x_{v,e}) = v_{des} \\ A_{Stop}, & \text{if } v < v_{d,s}(x_{v,e}) \\ -A_{Stop}, & \text{if } v > v_{d,s}(x_{v,e}) \end{cases} \tag{6}$$

3.5 Rotational Velocity Controller

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

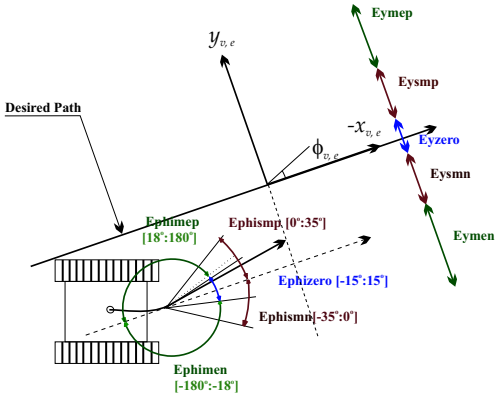

Figure 4: Rotational velocity controller input space.

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 [17]. In order to have smooth output changes, the membership functions are equally distributed as shown in Figure 5-a. The output space mapping function rules are represented in Figure 5-b.

3.6 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:

$$\begin{bmatrix} V_{D,L} \\ V_{D,R} \end{bmatrix} = \begin{bmatrix} 1-\alpha_L \\ 1-\alpha_R \end{bmatrix} \cdot \begin{bmatrix} v_{d,v} - \dot{\phi}_{d,v}.\frac{D_{track}}{2} \\ v_{d,v} + \dot{\phi}_{d,v}.\frac{D_{track}}{2} \end{bmatrix} \tag{7}$$

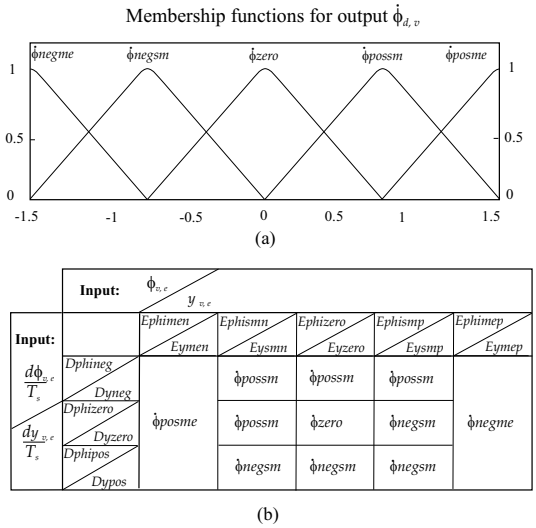

Figure 5: Rotational velocity control output space.

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].

4 Cross-Coupling Motion Control

A tracked vehicle, such as an excavator moves by the rotation of two rubber tracks that are driven and controlled by two independent motors. The steering action for such a vehicle is accomplished by the difference in speed of the two tracks. The velocities for such a vehicle can be expressed by [19]

$$\dot{x} = \frac{v_R + v_L}{2} sin\theta \ , \ \dot{y} = \frac{v_R + v_L}{2} cos\theta \ , \ \dot{\theta} = \frac{v_R - v_L}{D_{track}} \tag{8}$$

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 its own loop, while the other loop caries on as before. The cross-coupling controller regulates the orientation error $e\phi$ of the vehicle by exchanging feedback information between the two control loops [8].

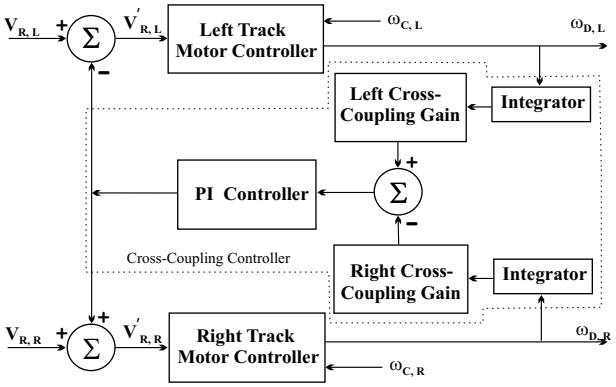

Figure 6: Cross-Coupling Controller block diagram.

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 [7] [6], Figure 6. In straight line motions the orientation error $e\theta$ is represented by

$$e\theta = \Sigma\Delta e\theta = \frac{d_R - d_L}{D_{track}} \tag{9}$$

where d_R and d_L are the total left and right track displacement for the current segment. In order to permit the vehicle to move along curved paths, the center of the robot must move along a circle of radius R. The corresponding track speeds are

$$v_L = V(R - \frac{D_{track}}{2R}) \ , \ v_R = V(R + \frac{D_{track}}{2R}) \tag{10}$$

where V is the translational velocity of the vehicle. From these two equations we can derive the relationship between the left and right track velocities to define two coefficients

$$C_L = 1 \ , \ C_R = \frac{v_L}{v_R} = \frac{R - \frac{D_{track}}{2R}}{R + \frac{D_{track}}{2R}} \tag{11}$$

These two coefficients are used as left and right cross coupling gains, Figure 6. To track a general nonlinear trajectory, the radius of the curvature is calculated at each sampling period and substituted into Equation 11.

5 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 feed back from the corresponding velocity sensor directly attached to the track axle. The derivation of the vehicle's current translational and rotational velocities, $V_{Cur,v}$, $\dot{\phi}_{Cur,v}$, from the measured rotational odometry velocities, $\omega_{C,L}$, $\omega_{C,R}$, are described by

$$\begin{bmatrix} V_{Cur,v} \\ \omega_{Cur,v} \end{bmatrix} = \begin{bmatrix} \omega_{C,R}.\frac{R_{Odo}}{2} + \omega_{C,L}.\frac{R_{Odo}}{2} \\ \dot{\phi}_{C,R}.\frac{R_{Odo}}{D_{Odo}} - \dot{\phi}_{C,L}.\frac{R_{Odo}}{D_{Odo}} \end{bmatrix} \quad (12)$$

Here R_{Odo} represents the radius of the resolvers and D_{Odo} denotes the distance between the odometry wheels. The vehicle motion transformation from the measured velocities to the position change is given by

$$\begin{bmatrix} dx_{Cur,W} \\ dy_{Cur,W} \\ d\phi_{Cur,W} \end{bmatrix} = \begin{bmatrix} cos(\phi_{Cur,W}) & 0 \\ sin(\phi_{Cur,W}) & 0 \\ 0 & 1 \end{bmatrix} . \begin{bmatrix} V_{Cur,v} \\ \dot{\phi}_{Cur,v} \end{bmatrix} .T_s \quad (13)$$

where $\phi_{Cur,W}$ denotes the Cartesian world angle from the position integrator, and T_s, the sampling period.

6 Vision-Based Motion Tracking

The vision system includes a head with three CCD cameras, mountable anywhere on the mobile vehicle. By processing consecutive trinocular sets images, the local 3D trajectory of the vehicle in an unstructured environment can be tracked [14]. The system does not rely on any prior knowledge of the environment or any specific landmark in the scene. We assume that 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 vehicle's motion is assumed to be limited in acceleration.

6.1 Feature Extraction

Although globally all the points in a scene convey some information about the motion, locally not all the pixel correspondences on the scene image carry valuable motion information. Also processing the entire existing image pixels is a time consuming process and includes ambiguity. Therefore, we work with discrete points, corners, of the scene with maximum information content. Schmid and Mohr [1] evaluate several corner detectors. They show that Harris and Stephens corner detector [3] outperforms other methods with a higher geometric stability and a larger independency from imaging conditions. Therefore, Harris and Stephens' corner detector is implemented that involves shifting a circular patch of the image in different directions. If the patch includes a corner, then shifting along all directions results in large changes. A corner can be detected when a minimum number of changes produced by any of the shifts is large enough. Each corner's quality is measured by a corner response R,

$$R = Det(M) - K(T_r(M))^2 \quad (14)$$

$$M = \begin{bmatrix} X^2 \otimes W & XY \otimes W \\ XY \otimes W & Y^2 \otimes W \end{bmatrix}, \; X \approx \frac{\partial I}{\partial x} \quad Y \approx \frac{\partial I}{\partial y} \quad (15)$$

6.2 Stereo Vision

The 3D world coordinates relative to the camera for each corner are computed using stereo algorithm. Our camera system captures a set of three images which are precisely aligned horizontally and vertically [13]. Candidate feature correspondences for the overlapping regions in the three stereo images are found and the Normalized Sum of Square Differences is computed for each pair of match candidates. The best match candidate is found by disparity sum minimization using the multiple-baseline algorithm [12]. In addition to the epipolar constraint, agreement between the horizontal and vertical disparities is employed, which eliminates unstable features, particularly those due to shadow effect. For areas of the reference image that are common in only one of the horizontal or vertical images, the Fua [2] method is employed.

6.3 Feature Tracking

Corresponding $3D$ features are tracked from one frame to the next in this section. There is no assumption or prediction about the value or direction of the motion. Therefore, for each corner a simple search routine is applied that finds all possible match candidates in the vicinity of the predicted position in the next image frame. Accordingly, NSSD is employed for each pair of match candidates. In order to cover all motions, a search scope of 27×27 pixels is employed.

6.4 Motion Estimation

Having a set of corresponding corners between each two consecutive images, motion estimation becomes the problem of optimizing a 3D transformation that projects the world corners, constructed from the first image, onto the second image. Therefore, the Newton's error minimization is implemented. To minimize the probability of converging to a false local minimum, we look for outliers and eliminate them during the iteration process. At each iteration a correction vector x is computed that is subtracted from the current estimate ($P^{(i)}$), resulting in a new estimate ($P^{(i+1)}$).

$$P^{(i+1)} = P^{(i)} - x \quad (16)$$

Given a vector of error measurements between the world $3D$ features and their projections, we find the x that minimizes this error. The effect of each element of correction vector x_i on error measurement e_i is the multiplication of error partial derivative with respect to that parameter to the same correction vector

$$Jx = e \qquad \text{where} \qquad J_{ij} = \frac{\partial e_i}{\partial x_j} \tag{17}$$

J is the Jacobian matrix and e_i presents the error between the predicted location of the object and actual position of the match found in image coordinates [9].

6.5 Position Refinement

Sensor noise and quantization associated with the image can each introduce slight displacements at feature locations within the image. As the camera navigates in its environment, most of the features fall into the camera field of view for a period of several frames. Detection of a feature in each frame by itself provides an additional information about that feature. Also, as the camera becomes closer to a feature, or as the features move from the image sides to its center, the 3D accuracy of the feature can improve dramatically. Therefore, combining the measurement for a feature with all its previous associated information reduces the uncertainty of that feature. In our system a positional covariance is associated with each observed feature using a Kalman filter generation. Each filter is updated using new information for the same feature over time.

$$w_i^{-1} = w_{i-1}^{-1} + A_i^T V_i^{-1} A_i \tag{18}$$

$$x_i = x_{i-1} + k_i(b_i - A_i x_{i-1}) \tag{19}$$

$$k_i = w_i A_i^T V_i^{-1} \tag{20}$$

In these equations i represents the frame number. w_i is the uncertainty in the estimation of x corresponding to frame i, k_i denotes the filter gain, b_i indicates the current measurement of the feature, V_i represents the covariance matrix of the errors and A_i is the identity matrix. We keep track of the features for a while, 6 consecutive frames, and then retire them.

7 Slippage Controller

One major problem with track 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 lose traction with the ground, which makes the readings of the dead-reckoning position odometer errornous. Moreover, when different amounts of slippage occur between left and right tracks, the track vehicle does not follow the appropriate curvature of the desired path. This creates path tracking, and/or position estimation errors. The individual track slippage condition is detected by comparing the track distance traveled during a time period, as measured by the track flowmeter sensor, with that calculated by the vision system. At each time, the vision system provides six motion parameters $(\Delta X, \Delta Y, \Delta Z, \Delta\phi_x, \Delta\phi_y, \Delta\phi_z)$. In this work, only $\Delta X, \Delta Y$ and $\Delta\phi_z$ are of interest, since these were the only motions measurable by the individual track hydraulic flowmeters. Figure 7 illustrates the motion of the center of the excavator and the motion of the camera system between two image frames.

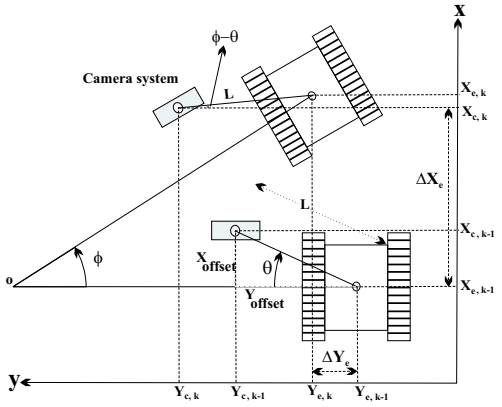

Figure 7: Excavator and camera in world coordinate.

$$\text{Assuming that} \qquad X_{e,k-1} = 0 \ , \ Y_{e,k-1} = 0 \tag{21}$$

the excavator and camera motion from frame $k-1$ to k in world coordinate system can be represented by

$$\Delta X_e = X_{e,k} - X_{e,k-1} = X_{e,k} \tag{22}$$

$$\Delta Y_e = Y_{e,k} - Y_{e,k-1} = Y_{e,k} \tag{23}$$

$$\Delta X_c = X_{c,k} - X_{c,k-1} = \Delta X_{vision} \tag{24}$$

$$\Delta Y_c = Y_{c,k} - Y_{c,k-1} = \Delta Y_{vision} \tag{25}$$

The coordinate of the camera center is expressed by

$$\Delta X_{c,k-1} = X_{e,k-1} + X_{offset} = X_{offset} \tag{26}$$

$$\Delta Y_{c,k-1} = Y_{e,k-1} + Y_{offset} = Y_{offset} \tag{27}$$

$$\Delta X_{c,k} = X_{e,k} + Lsin(\phi - \theta) \tag{28}$$

$$\Delta Y_{c,k} = Y_{e,k} + Lcos(\phi - \theta) \tag{29}$$

$$L = \sqrt{X^2{}_{offset} + Y^2{}_{offset}} = 0.891m \tag{30}$$

The motion of the camera can be rewritten as

$$\Delta X_{vision} = X_{e,k} + Lsin(\phi - \theta) - X_{offset} \tag{31}$$

$$\Delta Y_{vision} = Y_{e,k} + Lcos(\phi - \theta) - Y_{offset} \tag{32}$$

Changes in the position of excavator can be expressed by the estimated motion by the vision system as

$$X_{e,k} = \Delta X_{vision} + X_{offset} - Lsin(\Delta\phi_{vision} - \theta) \quad (33)$$

$$Y_{e,k} = \Delta Y_{vision} + Y_{offset} - Lcos(\Delta\phi_{vision} - \theta) \quad (34)$$

With the assumption of a constant speed for the excavator between two consecutive frames, the excavator translational and rotational speeds are estimated by

$$V_{exc,vision} = \frac{\sqrt{\Delta X_{vision}^2 + \Delta Y_{vision}^2}}{\Delta T} \quad (35)$$

$$\omega_{exc,vision} = \frac{\Delta\phi_{vision}}{\Delta T} \quad (36)$$

Here ΔT represents the time between the two frames, 50ms. The individual left and right track distances, measured by the vision system can be estimated by

$$\Delta d_{L,vision} = V_{exc,vision}\Delta T - \frac{B}{2}\omega_{exc,vision}\Delta T \quad (37)$$

$$\Delta d_{R,vision} = V_{exc,vision}\Delta T + \frac{B}{2}\omega_{exc,vision}\Delta T \quad (38)$$

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 track slippage, therefore, can be defined by

$$\alpha_i = Slip_i = \frac{\Delta d_{i,flow} - \Delta d_{i,vision}}{\Delta d_{i,vision}} \times 100 \quad (39)$$

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 to total track slippage.

8 Experimental Results

The performance of our system is evaluated through several experiments, some of which are demonstrated here. Figure 8 shows the result of the trajectory error generation for a typical path segment change. In this experiment the vehicle navigates from its current position ($x_{cur,v}$=75cm, $y_{cur,v}$=50cm) to a desired location of ($x_{cur,v}$=140cm, $y_{cur,v}$=0cm). Figures 8-b, -c and -d represent the corresponding changes in $x_{v,e}$, $y_{v,e}$ and $\phi_{v,e}$.

Figure 9 represents the performance of the slippage controller in conjuction with the rest of our system. In this experiment, the right track goes through a 25 degree slippage, solid line. The system automatically adjusts the speed of the left track, and as soon as the slippage is passed, the system goes back to normal values.

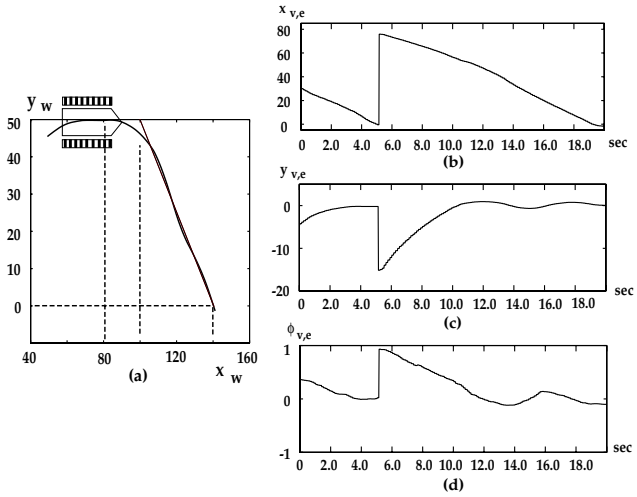

Figure 8: Trajectory error in a segment change.

Figure 10 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%.

9 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 track vehicle. Not only does our fuzzy logic tracking control scheme simplify the design, but the existing tolerance in fuzzy logic for dealing with imprecision and uncertainties makes this system more reliable for outdoor environments (especially in natural terrains). Also, the implemented vision system is scene-independent, and does not need any prior information about the scene.

The future work includes the implementation of a dynamic path planning system by employing multiple laser scanners for an even more automated performance.

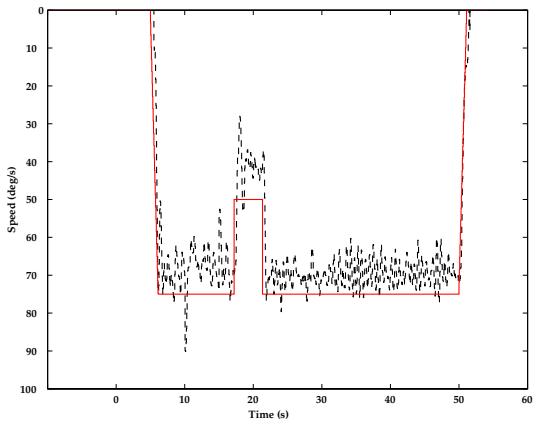

Figure 9: Slippage control results for the right track.

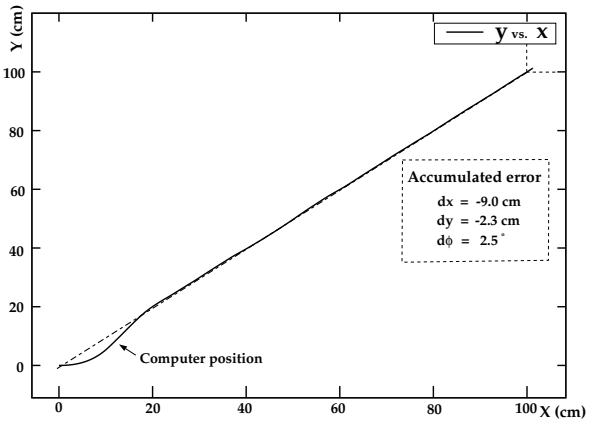

Figure 10: Path tracking and slippage control results.

References

[1] R. Mohr C. Schmid and C. Bauckhage. Comparing and Evaluating Interest Points. *Sixth International Conference on Computer Vision*, pages 230–235, 1998.

[2] Pascal Fua. *A parallel stereo algorithm that produces dense depth maps and preserves image features.* Machine Vision and Applications, Springer-Verlag, 1993.

[3] C. Harris and M. Stephens. A combined corner and edge detector. *Proceeding 4'th Alvey Vision Conference*, pages 147–151, 1988.

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

[5] K.C. Koh and H.S. Cho. A Smooth Path Tracking Algorithm for Wheeled Mobile Robots with Dynamic Constraints. 1999.

[6] 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.

[7] P.K. Kulkarni and K. Srinivasan. Optimal contouring control of multi-axial feed drive servomechanisms. *ASME Journal of Dynamic systems, measurements and control*, pages 140–148, 1989.

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

[9] David G. Lowe. Fitting Parameterized Three-dimensional Models to Images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 441–450, 1991.

[10] M.G. Mehrabi, A. Hemami, and R.M.H. Cheng. Analysis of Steering Control in Vehicles with Two Independent Left and Right Traction Wheels. 1991.

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

[12] M. Okutomi and T. Kanade. A multiple-baseline stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 353–363, 1993.

[13] Point Grey Research. Triclops Stereo Vision System. Technical report, Department of Computer Science, University of British Columbia, Vancouver, www.ptgrey.com, 1997.

[14] 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.

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

[16] S.J. Singh and D.H. Shino. Position Based Path Tracking for Wheeled Mobile Robots. 1989.

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

[18] 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.

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

[20] 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.

[21] Y. Zhang, D. Hong, J.H. Chung, and S.A. Velinsky. Dynamic Model Based Robust Tracking Control of a Differentially Steered Wheeled Mobile Robot. 1998.