Trajectory Tracking and Stair Climbing Capability Assessment for a Skid-Steered Mobile Robot

Chandrakanth R. Terupally[*], J.Jim Zhu[Δ] and Robert L. Williams II[↑]

Abstract— **This paper presents derivation of a trajectory tracking and stair climbing stabilization controller for a 4-wheel driven skid-steered wheeled mobile robot (SSWR). The robot vehicle is modeled with six degrees of freedom (6DOF) rigid body equations which are later simplified for planar and pitched movement. An efficient control algorithm, called Trajectory Linearization Control (TLC), is used to tackle the challenges posed by nonlinearities of the model. In TLC, state dynamics are linearized along the trajectory being tracked. Kinematics and dynamics of the vehicle are controlled individually by feedback loops, where the former constitutes the outermost loop. Simulation results promise trajectory tracking with desired accuracy. For the test prototype with current physical configuration, an upper limit is found on steepness of staircase it can safely climb. 3-sigma Monte Carlo analysis is used to show robustness of the controller to parametric perturbations.**

I. Introduction

Skid steered wheeled robots (SSWRs) constitute a genre of nonholonomic vehicles that lack a steering column and are controlled by differential steer mechanism. Absence of a steering column makes them mechanically robust and simple to control. Compared to tracked vehicles, SSWRs have two major advantages. Firstly, an SSWR with similar dimensions and weight would dissipate lesser energy in skidding turns compared to its tracked counterpart, owing to lesser area of surface contact. Secondly, SSWRs have lesser tendency to slip on stair edges and sway from direction of maximum ascent, as wheels tend to lock up with stairs.

Caracciolo et al. [1] approached planar tracking problem for an SSWR by constraining the instantaneous centre of rotation to between the two axles and used dynamic feedback linearization to reduce the control problem to mere kinematics. *Kozilowski et al.* [2] used *Dixon's* [3] algorithms based on Lyapunov's approach. Their technique relied on backstepping through intensive use of coordinate transformations making the algorithm quite complicated. Global exponential stability for transformed states was achieved by tracking an exponentially decaying tunable oscillator signal. This methodology overcame Brockett's condition and regulation problem was realized in contrast to *Caracciolo et al.* and the present paper.

For stair climbing, *Martens and Newman* [4] experimented with an algorithm by maintaining direction of maximum ascent during the climb by compensating for gravity-induced drift. A proportional control law was implemented with experimentally determined bandwidth and gain. *Steplight et.al* [5] developed a kinematics-based PD controller aided with a suite of sensors, used one at a time based on a rule, to estimate stair profile. *Xiong and Matthies* [6] used robust stair edge detection techniques using a single camera to estimate heading angle and centre position. They experimented stair climbing under various types of stairs in varied lighting conditions and shadows to prove robustness of their algorithm. Their work was extended by *Helmick et.al.* [7], who used a full-fledged sensor suite with accelerometers, vision, sonar etc. with enhanced estimation and control laws to detect and estimate stair profile. They claim to have achieved climbing speeds over 1½ feet/sec.

This paper aims to develop a unified controller for planar tracking and autonomous stair climbing based on 6DOF dynamic model, which is briefed in section II. The 6DOF vehicle model is simplified for planar tracking and modifications are made to the planar model to account for pitch angle during stair climbing. Section III discusses derivation of control law, followed by evaluation of controller performance by means of numerical simulations. Results presented for planar tracking show desirable performance with almost zero steady state tracking error. Robustness of the controller to parametric perturbations is investigated with 3-sigma Monte-Carlo simulations. Of 100 simulations with randomly perturbed physical parameters and friction coefficients, 96 yielded stable results. Simulation of stair climb showed that with the current configuration of the prototype robot, only a 20° staircase or a 30° ramp could be safely climbed up at a speed of 0.3m/s. This unsatisfactory limit calls for design modifications to the prototype robot.

Regulation problem is not addressed, as it is a concern of the highest level of control – the mission trajectory planner, to plan a path for the robot. Optimality decisions will be made at the same level.

II. Dynamic Modeling

Physical constants are defined and their values for the prototype robot are presented below.

a, distance of CG from front axle of robot = 0.198 m.
b, distance of CG from rear axle of robot = 0.2275 m.
c, half the track width = 0.294 m.
d, distance from instantaneous center of rotation (geometric center by choice) of robot to CG, -0.0148 m.
h, height of CG above plane of surface of contact.
I_{xx}, moment of inertia about body *x*-axis = 11 Kgm^2.
I_{yy} moment of inertia about body *y*-axis = 6.68 Kgm^2.
I_{zz} moment of inertia about bqody *z*-axis = 12.79 Kgm^2.
I_{xy} cross moment of inertia = -0.4 Kgm^2

[*] School of EECS, Ohio University.
[Δ] School of EECS, Ohio University, Correspondence email: zhuj@ohio.edu.
[↑] Department of Mechanical Engineering, Ohio University.

I_{xz} cross moment of inertia = -0.22 Kgm^2.
I_{yz} cross moment of inertia = 0.9543 Kgm^2.
K_m, motor torque constant = 0.74 Nm/A.
K_e, motor back-emf constant = 0.8433 Vs/rad.
r, radius of wheel = 0.1778 m.
R_a, motor armature resistance = 0.22 ohms.
m, mass of the robot =115.3 kg.
n, motor gear ratio = 1.
u_2, u_3 front left and front right applied motor voltages.
τ, torque produced by the motor.

A. 6DOF Model

The following four equations summarize the 6DOF model derived in [8].

Kinematics of translational motion:

$$\begin{bmatrix} \dot{X} \\ \dot{Y} \\ \dot{Z} \end{bmatrix} = \begin{bmatrix} C_\theta C_\psi & S_\phi S_\theta C_\psi - C_\phi S_\psi & C_\phi S_\theta C_\psi + S_\phi S_\psi \\ C_\theta S_\psi & S_\phi S_\theta S_\psi + C_\phi C_\psi & C_\phi S_\theta S_\psi - S_\phi C_\psi \\ -S_\theta & S_\phi C_\theta & C_\phi C_\theta \end{bmatrix} \begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix} \quad (1)$$

Dynamics of translational motion are given by

$$\begin{aligned} \dot{v}_x &= \tfrac{1}{m} F_x - \omega_y v_z + \omega_z v_y \\ \dot{v}_y &= \tfrac{1}{m} F_y - \omega_z v_x + \omega_x v_z \\ \dot{v}_z &= \tfrac{1}{m} F_z - \omega_x v_y + \omega_y v_x \end{aligned} \quad (2)$$

Kinematics of rotational motion are given by

$$\begin{aligned} \dot{\phi} &= \omega_x + \omega_y \sin\phi \tan\theta + \omega_z \cos\phi \tan\theta \\ \dot{\theta} &= \omega_y \cos\phi - \omega_z \sin\phi \\ \dot{\psi} &= \omega_y \sin\phi \sec\theta + \omega_z \cos\phi \sec\theta \end{aligned} \quad (3)$$

Dynamics of rotational motion are given by

$$\begin{aligned} \dot{\omega}_x &= I_{xy}^x \omega_x \omega_y + I_{yz}^x \omega_y \omega_z + g_l^x M_x + g_n^x M_z \\ \dot{\omega}_y &= I_{xx}^y \omega_x^2 + I_{zz}^y \omega_z^2 + I_{xz}^y \omega_x \omega_z + g_m^y M_y \\ \dot{\omega}_z &= I_{xy}^z \omega_x \omega_y + I_{yz}^z \omega_y \omega_z + g_l^z M_x + g_n^z M_z \end{aligned} \quad (4)$$

where,

$\dot{X}, \dot{Y}, \dot{Z}$ are velocities measured in world frame; ϕ, θ, ψ are Euler angles roll, pitch and yaw respectively; v_x, v_y, v_z are vehicle velocities measured in body frame; $\omega_x, \omega_y, \omega_z$ are angular velocities of the vehicle about *x,y,z* axes (body axes) respectively; F_x, F_y, F_z denote forces acting on center of gravity along *x,y,z* axes respectively; M_x, M_y, M_z respectively denote total moment about body x, y and z-axes; I_{xy}^x denotes coefficient of cross moment of inertia occurring in $\omega_x \omega_y$ term in the equation for $\dot{\omega}_x$ and g_z^x denotes coefficient of M_z in equation for $\dot{\omega}_x$. Detailed expressions for each of them can be found in [13].

B. Planar Motion

For planar motion, we have

$$\theta = \phi = 0,\ \omega_x = \omega_y = 0, M_x = M_y = 0, \dot{v}_z = 0,\ v_z = 0 \quad (5)$$

$$\tfrac{1}{m} F_x \gg \omega_y v_z \text{ and } \tfrac{1}{m} F_y \gg \omega_z v_x \ . \quad (6)$$

Simplifying (1)-(4) using (5) and (6) yields a model as presented in [1] and [2], which can be summarized as

$$\dot{p} = \mathcal{S}\, q \quad (7)$$

$$\dot{q} = -\underline{M}^{-1}\underline{C} q - \underline{M}^{-1}\underline{R} + \underline{M}^{-1}\underline{B}\tau \quad (8)$$

where,

$p = \begin{bmatrix} X & Y \end{bmatrix}^T$ is position vector and $q = \begin{bmatrix} v_x & \omega_z \end{bmatrix}^T$ is velocity vector;

$$\mathcal{S}(t) = \begin{pmatrix} \cos\psi & d\sin\psi \\ \sin\psi & -d\cos\psi \end{pmatrix}.$$

Derivation and detailed expressions for the matrices $\underline{M}, \underline{C}, \underline{R}, \underline{B}$ are presented in [2, 9].

C. Introducing Pitch into the Planar Model

Planar dynamics presented in previous subsection are modified to introduce static effects of stair/ramp climbing. Changes in normal reactions at wheels due to non-zero pitch are the only aspect considered. Body roll and rate of change of pitch are neglected assuming that the vehicle is on a transversally level (unbanked) ramp/staircase and is heading in the direction of maximum ascent. A comprehensive model of stair-climb with dynamic effects of body roll and pitch is reserved for future work.

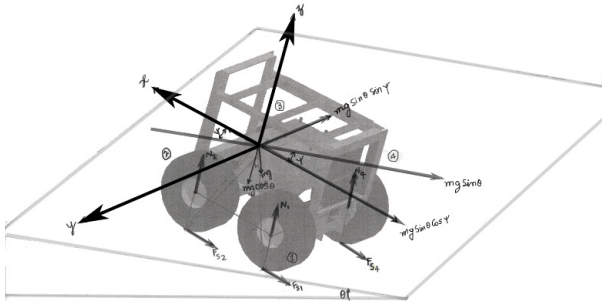

Figure 1: Free body diagram of the vehicle on an inclined plane (Courtesy: J. Litter [10])

The situation differs from planar (i.e. zero pitch) condition in that the normal reactions are distributed unequally on wheels and, in addition, there is a component of weight pulling the robot downwards the incline.

From the geometry of Figure 1 (best viewed at 250% zoom), writing torque equations about CG at surface point of contact of each wheel,

$$\vec{r}_{io} \times \vec{\mathcal{Z}} + \sum_{\substack{j=1 \\ j \neq i}}^{4} \vec{r}_{io} \times N_j = 0, \forall i = 1,2,3,4 \quad (9)$$

where $\vec{r}_{io}$ is the vector from the point of surface contact of i^{th} wheel to origin (CG); θ is measured positive for ascent; N_j is the normal reaction at i^{th} wheel; $\vec{\mathcal{Z}}$ is the resultant

force vector at CG, and

$$\bar{Z} = -(m\dot{v}_x + mg\sin\theta\cos\psi)\hat{x} - (mg\sin\theta\sin\psi)\hat{y} - (mg\cos\theta)\hat{z}$$

The set of equations in (9) is degenerate. A particular solution would exist if it were assumed that the vehicle is heading in the direction of maximum ascent (i.e., zero body-roll angle). In such a case, the front wheels (and so will the back wheels) share an equal amount of reaction force.

i.e., $N_1 = N_4$ and $N_2 = N_3$

This assumption renders *y* component of $\bar{Z}$ to zero. The solution to the set of equations would now be

$$N_1 = N_4 = \left(mgb\cos\theta + m\dot{v}_x h + mgh\sin\theta\right)\Big/2(a+b)$$

$$N_2 = N_3 = \left(mga\cos\theta - m\dot{v}_x h - mgh\sin\theta\right)\Big/2(a+b).$$

These expressions are substituted in the planar model's equations for frictional forces to account for non-zero pitch angle.

Compared to vehicle on a ramp, stability on staircase is worsened by the bumping effect when wheels encounter stair edges, which in turn introduces random errors in heading angle leading to a natural tendency to sway away from direction of steepest ascent to zero ascent. On a staircase with an inclination 'θ', it can be shown that the pitch angle variation due to wheel engaging the stairs is approximately $\delta\theta = 2\arcsin\left(r\frac{1-\sin\theta}{a+b}\right)$, assuming that step height and wheel radius are approximately equal.

III. Controller Design

Consider the time-varying nonlinear dynamic system

$$\dot{\xi}(t) = f(\xi(t), \mu(t), \sigma(t))$$
$$\eta(t) = h(\xi(t), \mu(t), \sigma(t)) \quad (10)$$

where $\xi(t)$ is *n* x 1 state vector, $\mu(t)$ is *p* x 1 vector of input, $\eta(t)$ is *m* x 1 output vector of the system, and $\sigma(t)$ is a set of time-varying parameters influencing the system behavior.

Define the nominal trajectories by

$$\dot{\bar{\xi}}(t) = f(\bar{\xi}(t), \bar{\mu}(t), \sigma(t))$$
$$\bar{\eta}(t) = h(\bar{\xi}(t), \bar{\mu}(t), \sigma(t)) \quad (11)$$

where an over-bar indicates nominal value of the quantity, $\bar{\eta}(t)$ is the desired output being tracked, $\bar{\mu}(t)$ is the nominal control input required, and $\bar{\xi}(t)$ is the nominal state vector.

Define the state, input and output errors respectively as

$$\tilde{\xi}(t) = \xi(t) - \bar{\xi}(t),\ \tilde{\mu}(t) = \mu(t) - \bar{\mu}(t),\ \tilde{\eta}(t) = \eta(t) - \bar{\eta}(t). \quad (12)$$

Taking time derivative of the state error vector and eliminating $\xi(t)$, $\mu(t)$, $\eta(t)$, and $\sigma(t)$ using (12) yields the tracking error dynamics

$$\dot{\tilde{\xi}} = f(\tilde{\xi} + \bar{\xi}, \tilde{\mu} + \bar{\mu}, \sigma) - f(\bar{\xi}, \bar{\mu}, \sigma) := F(\tilde{\xi}, \tilde{\mu}, \bar{\xi}, \bar{\mu}, \sigma)$$
$$\tilde{\eta} = h(\tilde{\xi} + \bar{\xi}, \tilde{\mu} + \bar{\mu}, \sigma) - h(\bar{\xi}, \bar{\mu}, \sigma) := H(\tilde{\xi}, \tilde{\mu}, \bar{\xi}, \bar{\mu}, \sigma) \quad (13)$$

The tracking error dynamics are stabilized by designing a PI controller using PD-eigenstructure placement techniques [11, 12, 13].

Open loop control $\bar{\mu}(t)$ and nominal state $\bar{\xi}(t)$ could be computed with a perfect inverse system that yields identity on cascade with the actual system. In general, this is neither possible nor desirable. Hence, a pseudo inverse system is designed to estimate the nominal control input required (to achieve desired output) and PI control is used to drive the tracking errors to zero. Pseudo inverse (or equivalently, nominal control) is expected to do a fair job in keeping open loop control errors small enough so that linearization of (10) about origin (i.e. zero errors) is valid at any given point of time.

Time variable has been dropped to avoid clutter, but quantities in the above equations are not constrained to be time-invariant. Also, it is assumed that the parameter set $\sigma(t)$ of the system is known and does not require to be estimated or errors accounted for.

Equation (13) when linearized, the state, input and output would be linear approximations of the actual errors. Hence, corresponding variables are changed while writing the linearized tracking error dynamics as:

$$\dot{x}(t) = A\left(\bar{\xi}(t), \bar{\mu}(t), \sigma\right)\tilde{x}(t) + B\left(\bar{\xi}(t), \bar{\mu}(t), \sigma\right)\bar{u}(t)$$
$$y(t) = C\left(\bar{\xi}(t), \bar{\mu}(t), \sigma\right)\tilde{x}(t) + D\left(\bar{\xi}(t), \bar{\mu}(t), \sigma\right)\bar{u}(t) \quad (14)$$

where $x(t), u(t), y(t)$ are linearized approximations of the error state $\tilde{\xi}(t)$ the error control input $\tilde{\mu}(t)$ and the output error $\tilde{\eta}(t)$ respectively, and *A, B, C, D* (with dependencies are dropped to avoid clutter) are Jacobian matrices in the Taylor expansion of (13) at $\left[\tilde{\xi}, \tilde{\mu}\right] = \left[0, 0\right]$.

Adding an integrator to the linear system in (16),

$$\dot{x}(t) = A(t)x(t) + B(t)u(t)$$
$$y(t) = C(t)x(t) + D(t)u(t) \quad (15)$$
$$\dot{\zeta}(t) = x(t)$$

Now a state feedback PI control law is implemented as

$$u(t) = -K_P(t)x(t) - K_I(t)\zeta(t) \quad (16)$$

Substituting (16) in (15) and rewriting in matrix form,

$$\begin{bmatrix}\dot{x}(t)\\ \dot{\zeta}(t)\end{bmatrix} = \begin{bmatrix}A(t) - B(t)K_P(t) & -B(t)K_I(t)\\ I & O\end{bmatrix}\begin{bmatrix}x(t)\\ \zeta(t)\end{bmatrix}$$
$$u(t) = -\begin{bmatrix}K_P(t) & K_I(t)\end{bmatrix}\begin{bmatrix}x(t)\\ \zeta(t)\end{bmatrix} \quad (17)$$

Using PD-eigenstructure assignment [11, 12, 13] for the system in (17), for constant PD-eigenvalues, the gains $K_P(t)$ and $K_I(t)$ can be evaluated for a desired polynomial

$$\lambda^n + \alpha_{n-1}\lambda^{n-1} + \ldots + \alpha_1\lambda + \alpha_0$$

where *n* is the order of the system and α_i for *i = 0* to *n-1* are coefficients of the polynomial.

Figure 2 depicts a schematic representation of TLC.

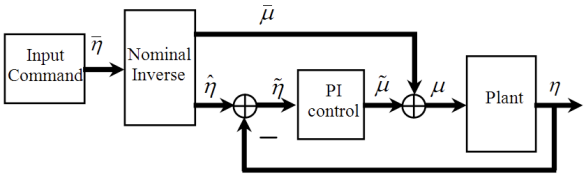

Figure 2: Trajectory linearization control schematic

A. Kinematics controller

The procedure outlined in the preceding section is applied to the kinematics system of (7) with $p(t)$ as output. For the closed loop system

$$\begin{bmatrix} \dot{x}(t) \\ \dot{\zeta}(t) \end{bmatrix} = \begin{bmatrix} A_K(t) - B_K(t)K_{KP}(t) & -B_K(t)K_{KI}(t) \\ I_2 & O_2 \end{bmatrix} \begin{bmatrix} x(t) \\ \zeta(t) \end{bmatrix}, \quad (18)$$

the expressions for PI gains K_{KP}, K_{KI} are evaluated using PD-eigenvalue placement techniques. Detailed expressions can be found in [9].

Inverse Kinematics

A stable and causal pseudo inversion in this case is straightforward due to absence of zero dynamics. Inverse system is described by the equations:

$$\bar{q}(t) = \bar{S}^{-1}(t)\dot{\bar{p}}(t) \quad (19)$$

where $\dot{\bar{p}}(t)$ is obtained from $\bar{p}(t)$ using a pseudo-differentiator whose bandwidth is dictated by that of the input $\bar{p}(t)$.

B. Dynamics controller

The closed loop state equation with state feedback becomes

$$\begin{bmatrix} \dot{x}(t) \\ \dot{\zeta}(t) \end{bmatrix} = \begin{bmatrix} A_D(t) - B_D(t)K_{DP}(t) & -B_D(t)K_{DI}(t) \\ I_2 & O_2 \end{bmatrix} \begin{bmatrix} x(t) \\ \zeta(t) \end{bmatrix}. \quad (20)$$

K_{DP}, K_{DI} are derived using PD-eigenvalue techniques. Their exact expressions (evaluated using symbolic toolbox of Matlab®) are presented in [9].

Inverse Dynamics

The state (which is the output too) and input are decoupled in forward dynamics of the robot as described by (8). It is, hence, possible to write a stable and causal pseudo-inverse relation by simple rearrangement of terms as

$$\begin{bmatrix} \bar{u}_2 \\ \bar{u}_3 \end{bmatrix} = \frac{1}{K_1}\underline{B}^{-1}\left(\underline{M}\begin{bmatrix} \dot{\bar{q}}_1 \\ \dot{\bar{q}}_2 \end{bmatrix} + \underline{\bar{C}}\begin{bmatrix} \bar{q}_1 \\ \bar{q}_2 \end{bmatrix} + K_2\underline{B}\begin{bmatrix} \bar{q}_1 - c\bar{q}_2 \\ \bar{q}_1 + c\bar{q}_2 \end{bmatrix} + \underline{\bar{R}} \right) \quad (21)$$

where, $\dot{\bar{q}}$ is pseudo derivative of $\bar{q}$.

TLC relies on an important assumption that errors are small, without which linearization may be invalid. A large step command may be sufficient to invalidate this assumption. It is hence desired that all commands input to the controller-plant system be smooth. This is achieved by forcing all input commands through a low pass filter called 'command shaping' whose bandwidth depends on the trajectories the robot is expected to track. Command shaper, typically a low pass filter, would make a step command into a smooth trajectory that becomes nominal input to the controller. The model used for simulations presented in this paper had such a filter in inverse kinematics and inverse dynamics blocks. Time varying bandwidth filter has been found to be advantageous over a constant bandwidth one in applications where commands are expected to have wide range of bandwidths [11, 12, 13].

IV. Numerical Simulations

Performance of the controller is evaluated for planar tracking and stair climbing by means of numerical simulations in Matlab-Simulink. To simulate performance of the tracking controller, the robot is commanded to trace the curve,

$X = t,\ Y = -.002t^3 + .1t^2 + .1t$, where t is time variable.

Figure 3 depicts performance graphs of the controller in nominal conditions. Subplot[1] (1, 1) shows a bird's eye view of path traced by the robot, which nearly overlaps with the commanded curve. Subplot (1,2) shows tracking error. It first increases because the commanded trajectory starts with a velocity of 1.005m/s, where as the robot starts from rest. With only one integrator in the kinematics loop, this curve should show a quadratic rise[2]; but as command shaping filter gains (of both outer and inner loops) are increased as time progresses, the deviation decreases later on. Heading angle is closely tracked and catches up almost instantaneously as shown by subplot (3,1).

Initial motor voltages (subplot (2,2)) and currents (subplot (2,3)) are well within their practical limits. With Roboteq motor controllers (AX2550) used on the test robot, 110A can be delivered to each motor. Subplot (2,2) shows that near the end of tracking session, motor voltages approach their supply limit of 24v (with currently installed batteries on the test prototype) as the robot pushes its speed beyond what it was built for (15km/h or ~4m/s).

Left and right motor currents (and hence torques) as seen in subplot (2,3) change directions for the commanded trajectory. This makes sense because the robot is initially commanded to turn to its left and later towards its right. This is exactly the same test trajectory *Caracciolo et al.* used in [1]. However, their torque curves didn't show such a behavior and did not cross zero during the course, which appears to be in error.

Robustness evaluation via Monte-Carlo simulations

To assess robustness of the controller to parametric perturbations, Monte-Carlo simulations were run programmatically in a loop with random dispersions in physical parameters and friction coefficients based on an assumed normally distributed statistical model. Table 1 shows various parameters with their nominal values and assumed standard deviations. Figure 4 shows performance plots resulting from one hundred simulations for planar tracking. The parameters were normally dispersed about their nominal value by three times their (assumed) standard

[1] *Subplot (i,j)* refers to the plot in row *i* and column *j* in the figure.
[2] The commanded curve has nonzero rate of acceleration (a jerk).

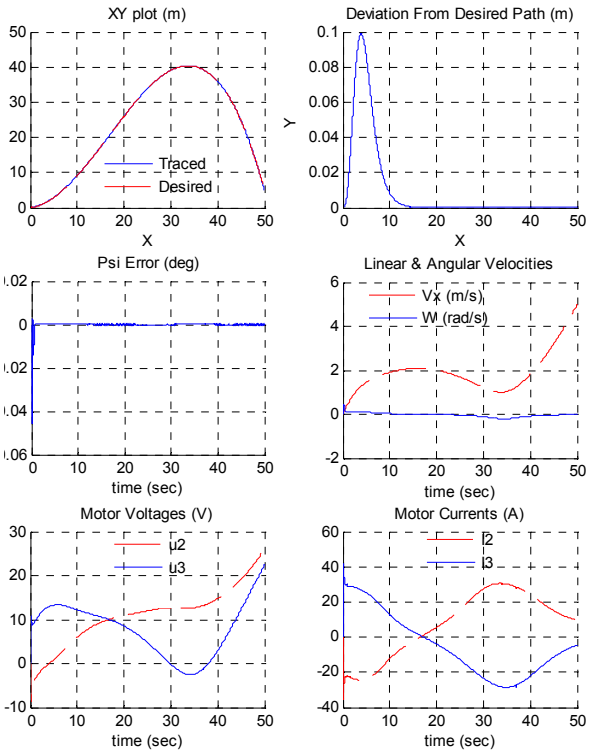

Figure 3: Nominal performance graphs

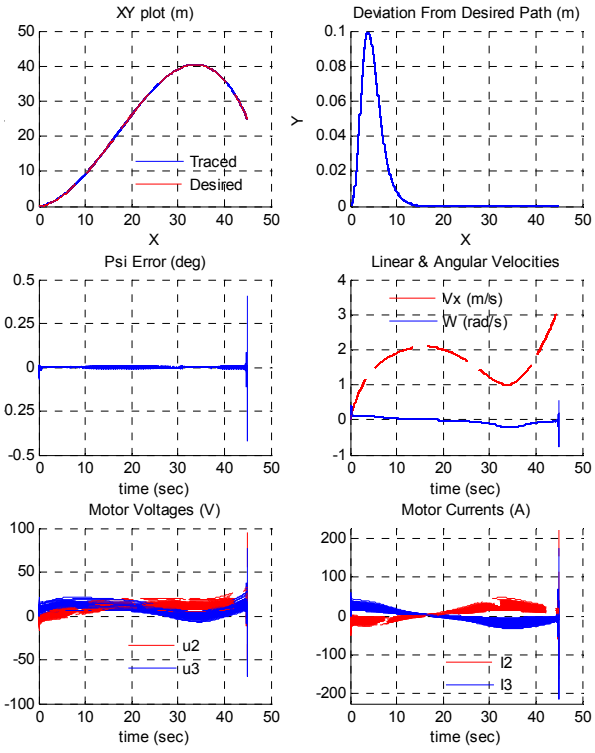

Figure 4: 3-Sigma Monte-Carlo simulations simulation

deviation. Friction profiles were smoothly varied with time, within preset limits randomly unique to each simulation.

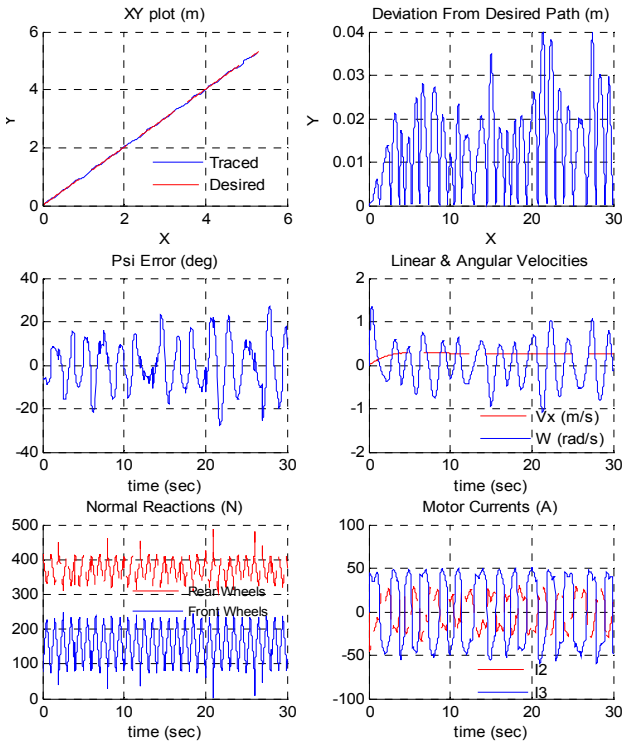

Figure 6: Performance graphs for stair climb-up

TABLE I
STANDARD DEVIATIONS OF PHYSICAL PARAMETERS

Parameter	Nominal Value	Standard Deviation
a	0.2275 m	5mm
b	0.1980 m	5mm
c	0.2940 m	5mm
d	-0.0147 m	---
mul	0.6000	0.05
mus	0.0600	0.005
m	115.30 kg	1.00 kg
I	12.79 kg	0.50 kg
r	0.1778 m	1mm
h	0.2769 m	1mm
Ke	0.84 Vs/rad	0.04 Vs/rad
Km	0.75 Nm/A	0.04 Nm/A
Ra	0.22 ohms	0.02 ohms

An exhaustive analysis of logged data and performance plots revealed that, out of one hundred simulations, four went unstable (seen as spikes at end of simulation time). In the first instance of instability, motor torque constant K_m was perturbed by ~30%, which is unrealistic for normal operating conditions. Second instance of instability was caused by a hike in wheel radius *'r'* by over 6cms (~34%). Such a variation is unusually high even if improper tire inflation is considered. In the third instance, moment of inertia was perturbed by approximately 20%. Using perturbation analysis, this parameter was found to be critical for stability at high speeds. A perturbation of 70% in armature resistance resulted in fourth instance of instability. These cases suggest that the perturbation models need

further validation with practical situations.

Stair Climbing Simulations

Figure 6 shows nominal performance simulating the robot climbing up the stairs. Pitch angle has been chosen to be 20° plus a disturbance[3] of ±10° to account for bouncing effect. Simulations show that the robot tips over with any pitch angle more than this. A uniform random disturbance within ±10° is added to heading (yaw angle) to account for random errors introduced due to slipping of tires on stair edges.

Subplot (1,2) shows a maximum deviation of 40cm from the median of staircase during the climb. This could be decreased by placing the controller eigenvalues farther to the left. Performance is achieved as a trade-off with energy consumption. Logic was implemented in the model to stop simulation if reaction-force on front wheels becomes negative at any point. Such situation did not arise in this 30s simulation (at 21s, the reaction was still slightly positive).

Mass and moment of inertia when individually perturbed by 1kg and 0.5kgm^2 respectively, did not challenge stability. Low speed of climb (0.3m/s) also aids stability of the vehicle. Height of CG above ground critically influences the limit of grade. An increase of 5mm in height of CG caused the robot to tip over amid a 30sec climb up. A 20° grade is too low for a typical indoor staircase and suggests that improvement is needed in mechanical design of the vehicle.

V. Conclusion

A unified controller for planar tracking as well as stair climbing of a SSWR has been presented. Simulation results show that the TLC controller is well suited to the application yielding desirable results with required robustness to physical parameters. Kozilowski *et al.* in [2] opined that a control algorithm relying on linearizing the dynamic equations is not possible due to unknown lateral skidding forces. This paper, on contrary, shows it is indeed possible.

This work will be extended to improve fidelity of the dynamic model and incorporate body roll and pitch dynamics in control design. A comprehensive evaluation of robustness of the controller to singular perturbations in addition to regular perturbations should be performed.

Acknowledgement

This work was sponsored by Ohio University, Athens. The authors thankfully acknowledge support and suggestions of Dr. Frank van Graas and Dr. Maarten Uijt de Haag, School of Electrical Engineering & Computer Science, Ohio University.

References

[1] Caracciolo L., De Luca A. and Iannitti S., "Trajectory tracking control of a four-wheel differentially driven mobile robot", *IEEE Int. Conf. Robotics and Automation*, Detroit, MI, 1999, pp. 2632–2638

[2] Krzysztof Kozłowski and Dariusz Pazderski, "Modeling And Control Of A 4-Wheel Skid-Steering Mobile Robot", *Int'l. J. Applied. Math. Computer Science*, 2004, vol. 14, No. 4, pp477–496.

[3] Dixon.W.E, Dawson.D.M., Zergeroglu.E, Behal.H, *Nonlinear Control of Wheeled Mobile Robots*, Springer, London, 2001.

[4] Martens J.D. and Newman W.S., "Stabilization of a mobile robot climbing stairs", *Proc. IEEE International Conference on Robotics and Automation,* Page(s): 2501 -2507 vol.3, 1994.

[5] Steplight.S, Egnal.G, Jung.S-H, Walker.D.B, Taylor.C.J, Ostrowski.J.P. "A Mode-Based Sensor Fusion Approach to Robotic Stair-Climbing", *Proc. 2000 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2000, Page(s): 1113 – 1118, vol. 2.

[6] Yalin Xiong and Larry Matthies, "Vision-Guided Autonomous Stair Climbing", *Proc. 2000 IEEE International Conference on Robotics and Automation*, Apr 24-28, 2000, pg.1842 -1847 vol.2.

[7] Helmick D.M., Roumeliotis S.I., McHenry M.C., Matthies L., "Multi-sensor, high speed autonomous stair climbing", *Intelligent Robots and System, 2002. IEEE/RSJ International Conference on, 2002*, Volume: 1, page(s): 733- 742 vol.1

[8] Robert Nelson, *Flight Stability and Automatic Control*, 2nd edition, McGraw Hill

[9] Chandrakanth Terupally, "Trajectory tracking control and stair climbing stabilization of a skid–steered mobile robot", masters' thesis, Dept of Electrical Engineering and Computer Science, Ohio University, Athens, Nov 2006.

[10] J.Litter, "Mobile robot for search and rescue", masters' thesis, Dept. Mechanical Eng. Ohio University, Athens, June 2004.

[11] C. M. Mickle, "Nonlinear Tracking Control Using A Robust Differential-Algebraic Approach", Ph. D. Dissertation, Dept. of Electrical and Computer Engineering, Louisiana State University, Baton Rouge, Louisiana, Dec. 1998.

[12] M.C. Mickle, Rui Huang, J. Zhu, "Unstable, Non-minimum Phase, Nonlinear Tracking by Trajectory Linearization Control," *Proceedings, 2004 IEEE Conference on Control Applications*, *pp. 812-818,* Taipei, Taiwan, Sept. 2004.

[13] R. Huang, M. C. Mickle, and J. Zhu, "Nonlinear Time-varying Observer Design Using Trajectory Linearization," *Proceedings, 2003 American Control Conference*, pp. 4772-4778, Denver, CO, June, 2003.

[3] Disturbance was modeled as a triangular wave.