

Stability Versus Maneuverability of Non-holonomic Differential Drive Mobile Robot: Focus on Aggressive Position Control Applications

Kaustav Mondal, Brent Wallace, Armando A. Rodriguez

Abstract—This paper presents a novel control centric dynamic modeling analysis focused on the relationship between stability and maneuverability of non-holonomic differential drive robots. The impact of specific vehicle design parameters on stability, lateral and longitudinal maneuverability of robot are examined over a broad range of forward motion operating conditions. The central objective is to determine whether the directional instability created by placing the center of gravity (c.g.) behind wheel-axle, aids in the performance of a robot executing aggressive cornering maneuvers. To this end, the paper explores two outer-loop position control applications, (1) Trajectory tracking using Lyapunov based method, (2) Minimum-time maneuvering of racetrack using Model Predictive Control (MPC) strategy. A hierarchical inner-outer loop control architecture with a weighted \mathcal{H}^∞ mixed sensitivity based inner-loop velocity tracking system, is presented for the same. The advantages and disadvantages of proposed modeling approach and associated control relevant performance trade-offs are demonstrated through simulations in discrete time.

I. INTRODUCTION: MOTIVATION AND CONTRIBUTIONS

With mobile robots becoming increasingly ubiquitous in everyday life, they are expected to be more agile in their directional maneuvering capability to push the boundaries of their performance. This is definitely a necessity when the robot is working in an uncertain dynamic environment, for example, in human-robot interaction based applications where it may need to rapidly alter the trajectory to avoid collisions due to an unexpected obstacle in the path. Other applications may include tasks such as trajectory tracking of predefined trajectories with sharp corners, minimize lap-times around a track where vehicles employ aggressive cornering strategies to maximize progress around apex, playing robot soccer etc. Exploiting the instability of a vehicular system to enhance maneuverability has long been a tradition. This idea is widely used in modeling of combat jets by placing center of gravity(c.g.) behind center of pressure to create a pitch-up instability and hence make the jet more maneuverable[1]. Also, multi-legged robots are designed to have “straight walk” instability which increases their turning maneuverability[2]. Using similar ideas, in this paper, we examine the lateral maneuverability of a differential drive non-holonomic mobile robot by placing the c.g. behind wheel axle, making the robot directionally unstable. To our knowledge, this has not been addressed in the literature.

K.Mondal is a Ph.D. student in School of Elect., Computer & Energy Eng. (ECEE), Arizona State University (ASU), Tempe, AZ; Brent Wallace is a BS student in ECEE; Dr. A.A. Rodriguez aar@asu.edu is a Professor in ECEE, ASU. This work has been supported, in part, by National Science Foundation (NSF) Grant No. 1565177. Any opinion, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of NSF.

Our paper sheds light on two critical questions, 1) How does directional instability affect the lateral and longitudinal maneuverability of the robot, 2) How does directional instability affect the performance of robot in applications involving high bandwidth position control. In this paper we focus on two outer-loop position control applications, 1) trajectory tracking [3], 2) minimum time maneuvering around racetrack [4]. First, a 4th order two-input two-output (TITO) linear parameter varying (LPV) state space model is presented, capturing the dynamic behavior of robot at varying operating points (v_{eq}, ω_{eq}) . Dynamic analysis is performed on a family of plants with varying c.g. position to understand how the steady state control effort and stability of longitudinal and lateral dynamics of the robot change with variation in c.g. position. For simulation purposes, the implemented parametric motion model consists of the kinematic model and an inner-loop (v, ω) low-level velocity tracking system using a 2×2 decentralized MIMO PI controller, designed using weighted \mathcal{H}^∞ mixed sensitivity approach[11, 12]. To perform trajectory tracking, a high-level Lyapunov based nonlinear position control law [5] is implemented on a family of parametric motion models of robot with varying c.g. position and guidelines to design controller gains are presented. The impact of directional (in)stability on tracking errors are studied. To minimize lap-time around a track [4], a model predictive control (MPC) strategy is used because of its ability to systematically include linear/nonlinear constraints on state and control signals pertaining to real world scenarios [10]. A linearly constrained convex quadratic programming problem (LCQP) is solved to generate the necessary control signals to maximize progress (minimize time) around the track in a given prediction horizon [4]. The impact of directional (in)stability of the robot on its ability to perform aggressive cornering maneuvers and time-taken to traverse around track are analyzed and control relevant trade-offs are studied. The remainder of the paper is structured as follows. Section II discusses the kinematic model, dynamic model and control relevant dynamic model analysis of the robot. Section III discusses the (v, ω) tracking inner-loop control system. Section IV discusses outer-loop position control applications. Section V summarizes the paper.

II. MODELING OF DIFFERENTIAL DRIVE ROBOT

In this section, we present dynamic modeling and analysis of non-holonomic robot. The nonlinear kinematic and dynamic model and their linear parameter varying (LPV) approximate representations under forward motion operating conditions $(v_{eq} > 0, \omega_{eq} \geq 0)$ are analyzed. The impact of

critical vehicle design specifications on dynamic properties of the robot have also been addressed.

A. Kinematic Model

The nonlinear kinematic model of a non-holonomic robot is a simple unicycle type model and defined as [6] : $\dot{x} = v \cos \theta$, $\dot{y} = v \sin \theta$, $\dot{\theta} = \omega$ where $(x, y) \in \mathbb{R}^2$, $\theta \in \mathbb{R}_{[-\pi, \pi]}$ is the pose of the robot in inertial frame. The inputs (v, ω) denote the translational and angular velocity of the robot, respectively. By linearizing this model about time varying local operating points and further discretizing using Euler method, we obtain the linear parameter varying (LPV) state-space representation of the nonlinear kinematic model[14]

$$\mathbf{x}_K(k+1) = A_K(k)\mathbf{x}_K(k) + B_K(k)\mathbf{u}_K(k) + E_K(k) \quad (1)$$

$$A_K = \begin{bmatrix} 1 & 0 & -v_r(k) \sin \theta_r(k) \Delta t \\ 0 & 1 & v_r(k) \cos \theta_r(k) \Delta t \\ 0 & 0 & 1 \end{bmatrix}, B_K = \begin{bmatrix} \cos \theta_r(k) \Delta t & 0 \\ \sin \theta_r(k) \Delta t & 0 \\ 0 & \Delta t \end{bmatrix}$$

$$E_K = \begin{bmatrix} v_r(k) \theta_r(k) \sin \theta_r(k) \Delta t \\ -v_r(k) \theta_r(k) \cos \theta_r(k) \Delta t \\ 0 \end{bmatrix}, \mathbf{x}_K = \begin{bmatrix} x(k) \\ y(k) \\ \theta(k) \end{bmatrix}, \mathbf{u}_K = \begin{bmatrix} v(k) \\ \omega(k) \end{bmatrix}$$

where $\mathbf{x}_r(k) = [x_r(k) \ y_r(k) \ \theta_r(k)]^T$, $\mathbf{u}_r(k) = [v_r(k) \ \omega_r(k)]^T$ are time varying local operating points about which the nonlinear kinematic model is linearized, Δt is the sampling period and k is the sampling instant.

B. Dynamic Model

The nonlinear dynamic model of non-holonomic robot is a force-to-velocity vehicle model described as a function of (v, ω) and motor torques $\tau_{r,l}$ ¹ applied to wheels. Using motor torques in terms of aggregate force $F = \frac{\tau_r + \tau_l}{r}$, aggregate torque $\tau = d_w \frac{\tau_r - \tau_l}{r}$ as inputs (r, d_w defined in Table I) and (v, ω) as states, the nonlinear dynamic model [7] is :

$$\left(m + \frac{2I_w}{r^2}\right) \dot{v} - m_c d \omega^2 = F, \quad \left(I + \frac{d_w^2}{2r^2} I_w\right) \dot{\omega} + m_c d \omega v = \tau$$

In this model, ω^2 and ωv nonlinearities are observed as long as $d \neq 0$ where d is the distance of robot center of gravity (c.g.) from wheel axle along the longitudinal axis of robot. The rest of the parameters used in the model are defined in Table I. From the nonlinear dynamic model, we obtain the linear model $P_{(F, \tau) \rightarrow (v, \omega)}$ linearized about (v_{eq}, ω_{eq}) :

$$\begin{bmatrix} \dot{v} \\ \dot{\omega} \end{bmatrix} = \begin{bmatrix} 0 & \frac{2m_c d \omega_{eq}}{\hat{I}} \\ -\frac{m_c d \omega_{eq}}{\hat{I}} & -\frac{m_c d v_{eq}}{\hat{I}} \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} + \begin{bmatrix} \frac{1}{\hat{m}} & 0 \\ 0 & \frac{1}{\hat{I}} \end{bmatrix} \begin{bmatrix} F \\ \tau \end{bmatrix} + \begin{bmatrix} \frac{m_c d \omega_{eq}^2}{\hat{m}} - \frac{2m_c d \omega_{eq}}{\hat{I}} \\ \frac{m_c d v_{eq} \omega_{eq}}{\hat{I}} \end{bmatrix} \quad (2)$$

Here, $\hat{m} = m + \frac{2I_w}{r^2}$, $\hat{I} = I + \frac{d_w^2}{2r^2} I_w$ are constants. It is useful to observe in (2) that if $d = 0$, matrix A is zeroed and (2) becomes a simple decoupled gain-integrator model. If $d \neq 0$, then depending on the position of the c.g. w.r.t. wheel axle (Figure 1), the dynamic model may be directionally stable ($\lambda_{1,2} < 0$, $d > 0$) or unstable ($\lambda_{1,2} > 0$, $d < 0$) where $\lambda_{1,2} = \left(-\frac{c_3}{2} \pm \sqrt{\frac{c_3^2}{4} - c_1 c_2}\right) d$, $c_1 = \frac{2m_c \omega_{eq}}{\hat{m}}$, $c_2 = \frac{m_c \omega_{eq}}{\hat{I}}$,

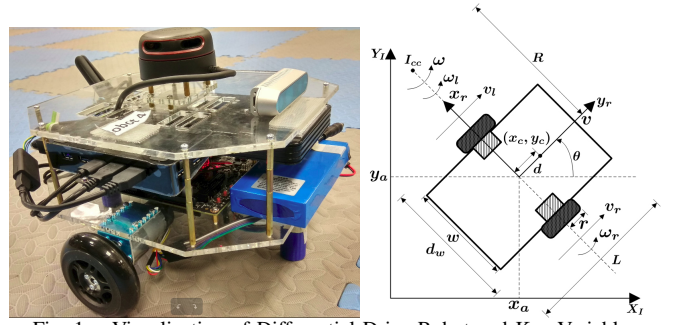


Fig. 1. Visualization of Differential Drive Robot and Key Variables

$c_3 = \frac{m_c v_{eq}}{\hat{I}}$. For our nominal plant (Figure 1), c.g. is very close to wheel axle ($d \approx 0$), as seen in Table I.

In this paper, we assume that torques on each wheel $\tau_{r,l}$ are generated by identical armature controlled dc motors with input voltages $e_{a,r,l}$. We use the same motor model presented in [8] : $e_{a,r,l} = l_a \frac{di_a}{dt} + r_a i_a + e_b$ (Armature Equation), $e_b = k_b k_g \omega_{r,l}$ (Back EMF Equation), $\tau_{r,l} = k_t k_g i_a$ (Torque Equation), $I_w \dot{\omega}_{r,l} + \beta \omega_{r,l} = \frac{\tau_{r,l}}{k_g^2}$ (Load Equation) where $e_{a,r,l}$ are the motor input voltages, i_a is the armature current, e_b is the back emf, $\omega_{r,l}$ are the angular wheel speeds and $\tau_{r,l}$ are the torques applied to the wheels. The motor parameters are given in Table I. It is important to note that the above motor model alters the properties (moves poles to the left) of dynamic model in (2) because the motors “wrap” around the vehicle model; i.e. through a wheel speed gearing-damping feedback path and a gearing-back-emf feedback path as described in [8]. In other words, adding the motor dynamics has a stabilizing effect on (2). Using the linear dynamic model (2) and motor dynamics shown above, we obtain the two-input two-output (TITO) LPV fourth order state space model $P_{(\bar{e}_a, \Delta e_a) \rightarrow (v, \omega)}$ describing the dynamics of our differential drive robot near time varying operating point (v_{eq}, ω_{eq}) [9]:

$$\dot{\mathbf{x}}_p = A_p \mathbf{x}_p + B_p \mathbf{u}_p + E_p, \quad \mathbf{y}_p = C_p \mathbf{x}_p \quad (3)$$

$$A_p = \begin{bmatrix} \frac{-2\beta k_g^2}{\hat{m} r^2} & \frac{2m_c \omega_{eq} d}{\hat{m}} & \frac{k_t k_g}{\hat{m} r} & \frac{k_t k_g}{\hat{m} r} \\ \frac{-m_c v_{eq} d}{\hat{I}} & \left(-\frac{m_c v_{eq} d}{\hat{I}} - \frac{d_w^2 k_g^2 \beta}{2I r^2}\right) & \frac{k_t k_g d_w}{2I r} & -\frac{k_t k_g d_w}{2I r} \\ -\frac{k_b k_g}{l_a r} & -\frac{k_b k_g d_w}{2l_a r} & -\frac{r_a}{l_a} & 0 \\ \frac{k_b k_g}{l_a r} & \frac{k_b k_g d_w}{2l_a r} & 0 & -\frac{r_a}{l_a} \end{bmatrix}$$

$$B_p = \begin{bmatrix} 0_{2 \times 2} \\ \frac{1}{l_a} M \end{bmatrix} \quad C_p = [I_{2 \times 2} \quad 0_{2 \times 2}] \quad E_p = \begin{bmatrix} \frac{m_c d \omega_{eq}^2}{\hat{m}} - \frac{2m_c d \omega_{eq}}{\hat{I}} \\ \frac{m_c d v_{eq} \omega_{eq}}{\hat{I}} \\ 0_{2 \times 1} \end{bmatrix}$$

where $\mathbf{x}_p = [v \ \omega \ i_{a_r} \ i_{a_l}]^T$, $\mathbf{u}_p = [\bar{e}_a \ \Delta e_a]^T$, (i_{a_r}, i_{a_l}) represents the armature current, $(\bar{e}_a, \Delta e_a)$ represents the average and difference of motor input voltages (e_{a_r}, e_{a_l}) , respectively, M^2 is the mapping between (e_{a_r}, e_{a_l}) and $(\bar{e}_a, \Delta e_a)$. Instead of using traditional input voltage (e_{a_r}, e_{a_l}) , we use $(\bar{e}_a, \Delta e_a)$ as inputs to (3) because \bar{e}_a and Δe_a are seen as natural control inputs for v and ω , respectively. Furthermore, if c.g. is designed to lie on wheel axle ($d = 0$), (3) becomes fully decoupled allowing us to use decentralized controllers

¹Throughout the paper, we use subscripts r and l to denote right and left for wheel and motor quantities, respectively

²Throughout the paper, we use $M = \begin{bmatrix} 1 & 0.5 \\ 1 & -0.5 \end{bmatrix}$ to denote the mapping between (e_{a_r}, e_{a_l}) and $(\bar{e}_a, \Delta e_a)$

$K(s) = k_i(s)I_{2 \times 2}$, $i = 1, 2$ to control longitudinal ($P_{\bar{e}_a \rightarrow v}$) and lateral ($P_{\Delta e_a \rightarrow \omega}$) dynamics of the robot independently and this is advantageous from a control design perspective.

C. Dynamic Model Analysis

In this subsection, we analyze the effect of vehicle design specifications on dynamic properties of non-holonomic robot. Vehicle design begins by determining critical mission capabilities. For applications involving aggressive cornering maneuvers such as 1) high velocity trajectory tracking, 2) minimum lap-time around racetrack, it would be advantageous to design a directionally unstable $P_{(F,\tau) \rightarrow (v,\omega)}$ in (2) because directional instability leads to 1) more lateral maneuverability, 2) less chances of motor saturation (to be discussed in Section IV). As we saw in (2), position of c.g. w.r.t wheel axle impacts the stability of $P_{(F,\tau) \rightarrow (v,\omega)}$. Thus, the designer can fine-tune his/her design by shifting the c.g. in front of wheel axle ($d > 0$) to make (2) directionally stable (sluggish) or shifting the c.g. behind wheel axle ($d < 0$) to make (2) directionally unstable (more maneuverable). After adding motor dynamics to (2), the point of marginal stability in (3) is moved to $d = -\epsilon$, $\epsilon \in \mathbb{R}_{>0}$ since motor dynamics has a stabilizing effect on (2). To perform stability analysis on (3) ($P_{(F,\tau) \rightarrow (v,\omega)}$ + motor dynamics), stability regions are plotted for a fixed grid of d (position of c.g.) and v_{eq} (translational velocity operating point) while keeping $\omega_{eq} = 1.2$ rad/sec. From Figure 2, it is evident that stability depends on parameters d and v_{eq} with $P_{(\bar{e}_a, \Delta e_a) \rightarrow (v,\omega)}$ being always stable when $d \geq 0$. For $d < 0$, the plant is stable at lower translational velocity ($v_{eq} < 1.35$ m/sec). For $v_{eq} \geq 1.35$ m/sec, there exists a $d = -\rho$, $\rho \in \mathbb{R}_{\geq 0}$ corresponding to marginal stability of $P_{(\bar{e}_a, \Delta e_a) \rightarrow (v,\omega)}$ in (3). Also, stability region shrinks; i.e. $\rho \rightarrow 0$ as v_{eq} increases. Therefore, depending on c.g. position (d) and operating point (v_{eq}, ω_{eq}), (3) may be stable/sluggish or unstable (more maneuverable).

The DC gain of longitudinal dynamics $P_{\bar{e}_a \rightarrow v}$ and lateral dynamics $P_{\Delta e_a \rightarrow \omega}$ gives insight into control effort required to perform a maneuver at steady state. In Figure 3, the DC gain is plotted for a family of $P_{\Delta e_a \rightarrow \omega}$ with varying position of c.g. ($d \in \mathbb{R}_{[-6,6]}$), varying translational velocity operating point $v_{eq} \in \mathbb{R}_{[0,5]}$ and fixed $\omega_{eq} = 1.2$ rad/sec. It is interesting to see that for each v_{eq} , DC gain of $P_{\Delta e_a \rightarrow \omega}$ increases as d decreases until it reaches the critical value at $d = -\rho$, $\rho \in \mathbb{R}_{>0}$ corresponding to marginal stability of (3) beyond which DC gain analysis is undefined. This means that as d decreases to $d = -\rho$, the increase in DC gain of $P_{\Delta e_a \rightarrow \omega}$ leads to less control effort Δe_a required to perform a cornering maneuver at steady state. Also, comparing DC gains of $P_{\Delta e_a \rightarrow \omega}$ corresponding to any two c.g. positions at an equal distance ϵ ($\epsilon \in \mathbb{R}_{[0,\rho]}$) and on opposite sides of wheel axle ($d = 0$), we notice that as v_{eq} increases, the DC gain decreases for $d = \epsilon$ and increases for $d = -\epsilon$. From this observation, we conclude that control effort Δe_a required by a robot with $d = \epsilon$, increases as robot performs a particular cornering maneuver at a higher velocity. Similarly, lesser control effort Δe_a is required to perform the same maneuver if the robot is designed with

$d = -\epsilon$. From all these observations, we conclude that it is to our advantage to design a robot with c.g. behind wheel axle for performing aggressive position control maneuvers. The variation of parameter d and v_{eq} does not have any significant effect on the stability and frequency response of longitudinal dynamics $P_{\bar{e} \rightarrow v}$.

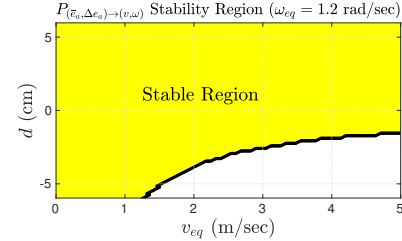


Fig. 2. $P_{(\bar{e}_a, \Delta e_a) \rightarrow (v, \omega)}$ Stability Region

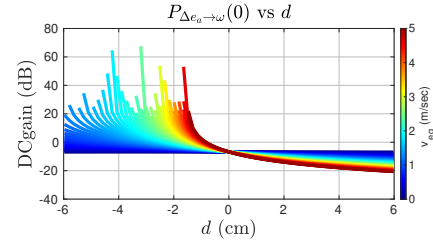


Fig. 3. DC gain of $P_{\Delta e_a \rightarrow \omega}$

In this paper, we focus our attention on a stable and an unstable dynamic model (represented by P_1 and P_2) linearized about $(v_{eq}, \omega_{eq}) = (2, 1.2)$, whose c.g. are at 4 cm and -4 cm w.r.t. wheel axle, respectively. The lateral dynamics of P_2 is unstable due to presence of RHP pole at 1.15. The singular values of the models are illustrated in Figure 4. The minimum singular values of P_1 , P_2 (corresponding to velocity v -channel) at low frequencies, are near -20 dB which suggests the need of an aggressive controller in v -channel ($\sigma_{min}(PK) \geq 40$ dB at low frequencies) to ensure good command following and sensor noise attenuation. Also, the maximum singular value (corresponding to ω -channel) of P_2 at low frequencies, is close to 10 dB, 20 dB more than $\sigma_{max}(P_1)$. This suggests that P_1 would require a far more aggressive controller (in ω -channel) than P_2 to achieve comparable properties at plant output. From this, we note that steady state control action Δe_a for P_1 would be greater than that of P_2 while executing cornering maneuvers, making P_1 (stable/sluggish) more susceptible to motor saturation.

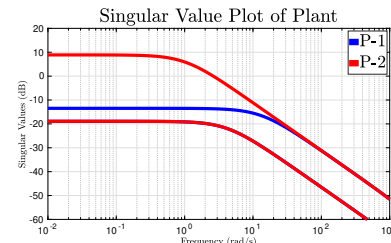


Fig. 4. Singular Values of $P_{(\bar{e}_a, \Delta e_a) \rightarrow (v, \omega)}$

III. INNER LOOP CONTROL STRATEGY

To show control relevant trade-offs associated with change in c.g. position w.r.t. dynamic models P_1 , P_2 , we use a parametric motion model consisting of kinematic model in

TABLE I
NOMINAL PLANT PARAMETER VALUES

Par	Definition	Nominal Value
m	Mass of fully loaded vehicle, $m = m_c + 2m_w$	3400 g
m_c	Mass of base (excludes wheels-motors)	2760 g
m_w	Mass of single wheel-motor combination	320 g
I_w	Wheel+motor moment of inertia about axle	$1.67 \mu\text{kgm}^2$
I	Total inertia	52.1 g m^2
r	Radius of wheels	4.2 cm
l	Length of robot chassis/base	28 cm
w	Width of robot chassis/base	25 cm
d_w	Distance between two wheels (at midpoint)	$\approx 25 \text{ cm}$
d	Distance c.g. lies forward of wheel axles	2.5 cm
l_a	Armature inductance	1.729 mH
r_a	Armature resistance	3.01Ω
k_b	Back EMF constant	9.5 mV/rad/s
k_t	Torque constant	9.5 mNm/A
k_g	Motor-wheel gear (down) ratio	50
β	Speed damping constant	$3.29 \mu\text{Nms}$

(1) and a MIMO inner-loop closed loop system $[v, \omega] = P_{(v_c, \omega_c) \rightarrow (v, \omega)} [v_c \ \omega_c]$ representing the low-level velocity (v) and angular velocity (ω) tracking systems. In this paper, we use a hierarchical inner-outer loop control strategy with a 2×2 decentralized PI controller $K(s) = k_i(s)I_{2 \times 2}$, $i = 1, 2$ in the inner-loop, designed using a non-standard (r, d_i) based weighted \mathcal{H}^∞ mixed sensitivity optimization problem [11, 12]. The optimization in (4) is solved using MATLAB's *Hinfstruct* command [13] which uses non-smooth optimization techniques to design fixed structure PI controllers. In (4), (S_e, T_e) and (S_c, T_c) are sensitivity, complementary sensitivity at output and input loop breakpoints, respectively, KS_e is the control action, PS_c is the transfer function matrix from input disturbance to output, K_{PI} is the controller space of all internally stabilizing 2×2 decentralized MIMO PI controllers and $W_{1,2,3}$ are \mathcal{RH}^∞ frequency domain weighting matrices used to shape closed loop properties

$$K = \arg\left\{ \min_{K \in \mathcal{K}_{PI}} \gamma \left\| \begin{bmatrix} W_1 S_e & W_1 P S_c \\ W_2 K S_e & W_2 T_c \\ W_3 T_e & W_3 P S_c \end{bmatrix} \right\|_{\mathcal{H}^\infty} < \gamma \right\} \quad (4)$$

The (v, ω) tracking system of both models (P_1, P_2) are designed to have similar bandwidths, for a fair comparison. $K(s)$ is generated for each design (D-I and D-II) corresponding to P_1, P_2 , respectively by solving (4) followed by brute force enumerations. The structure of controller used is as follows: $K_{11} = g_1(s+z_1)/s$, $K_{22} = g_2(s+z_2)/s$. A roll-off term $\left(\frac{100}{s+100}\right)$ separately added to the PI controllers to ensure high frequency noise attenuation. The directional instability in P_2 (RHP pole at 1.15 for $P_{\Delta e_a \rightarrow \omega}$) imposes a lower bound of 11.5 rad/sec ("factor of 10 rule") on inner-loop bandwidth of ω -channel for D-II. Further restrictions on achievable bandwidth due to sensor noise, motor saturation and ZOH sample delay are taken into consideration. The gains for $K(s)$ are $(g_1 = 10.8, z_1 = 12.5, g_2 = 10, z_2 = 16.5)$ for D-I and $(g_1 = 10.8, z_1 = 12.5, g_2 = 8.5, z_2 = 3.5)$ for D-II. Figure 5 depicts closed loop responses of both designs at output. The closed loop properties are summarized in Table II where (v_{ts}, ω_{ts}) , (v_{tr}, ω_{tr}) and (B_v, B_ω) are the settling time, rise time and bandwidth of (v, ω) inner-loop closed loop system, respectively. The results in Table II are desirable for both designs with D-I outperforming D-II. D-II has better steady state command following and less control

action than D-I in ω -channel, as shown in Figure 5. The step response in v -channel experiences 16.97% overshoot for both designs. However, the step response in ω -channel experiences 4.19% and 17% overshoot for D-I and D-II, respectively. The overshoot in D-II is due to presence of RHP pole at 1.15 [16]. Therefore, by designing an unstable robot we pay a price in terms of relatively high overshoot in the ω -channel.

The (v, ω) closed loop tracking system is implemented in discrete time in Section IV-A and used as part of the prediction model in the MPC optimization (26) in Section IV-B. Hence, we present the discrete state-space representation of inner-loop closed loop system $P_{(v_c, \omega_c) \rightarrow (v, \omega)}$ given by

$$\begin{aligned} \mathbf{x}_d(k+1) &= A_d \mathbf{x}_d(k) + B_d \mathbf{u}_c(k) + E_d, \quad \mathbf{y}_d(k) = C_d \mathbf{x}_d(k) \\ A_d &= e^{A_{cl} \Delta t} & B_d &= A_{cl}^{-1} (A_d - I) B_{cl} \\ C_d &= C_{cl} & E_d &= A_{cl}^{-1} (A_d - I) E_{cl} \end{aligned} \quad (5)$$

where $\mathbf{x}_d = [\mathbf{x}_p \ \mathbf{x}_k]^T$, \mathbf{x}_p and \mathbf{x}_k represent states of dynamic model and controller, respectively, $\mathbf{u}_c = [v_c \ \omega_c]^T$, $(A_{cl}, B_{cl}, C_{cl}, D_{cl})$ are the continuous time LTI state-space matrices of the (v, ω) closed-loop tracking system, Δt is the sampling period and k is sampling instant. Tustin's method [18] is chosen as the method of discretization because it preserves the (in)stability of system during discretization.

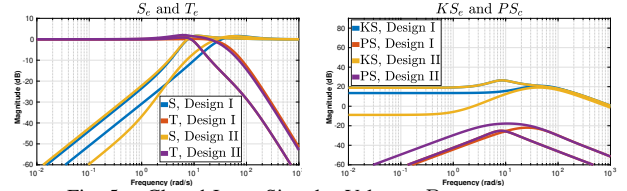


Fig. 5. Closed Loop Singular Values : $P_{(v_c, \omega_c) \rightarrow (v, \omega)}$

TABLE II

CLOSED LOOP PROPERTIES FOR DESIGNS I AND II						
Design	\bar{S}_e	\bar{S}_c	\bar{T}_e	\bar{T}_c	$\bar{K}S_e$	$\bar{P}S_c$
D-I	1.65	1.65	1.47	1.49	26.49	-21.96
D-II	1.68	1.66	2.01	1.65	26.51	-17.74
	v_{ts}	ω_{ts}	v_{tr}	ω_{tr}	B_v	B_ω
D-I	0.87	0.18	0.16	0.051	9.02	18.26
D-II	0.87	0.65	0.16	0.053	9.05	19.05

IV. OUTER LOOP CONTROL STRATEGIES

In this section, we demonstrate the impact of c.g. position w.r.t wheel-axle on maneuverability of robot by comparing performance of P_1 and P_2 for two outer-loop position control applications, 1) trajectory tracking of predefined trajectory, 2) minimum-time maneuvering around racetrack. By comparing the trajectory tracking performance of P_1 and P_2 , we show how designing a robot with c.g. behind wheel-axle leads to greater maneuverability and less tracking errors. We also demonstrate superiority of P_2 over P_1 in performing aggressive cornering maneuvers, thereby reducing lap-times around a racetrack. Control-relevant trade-offs are also presented.

A. Trajectory Tracking

The problem of tracking a pre-defined reference trajectory can be thought of as a leader-follower formation control problem. Given that (x_r, y_r, θ_r) and (x, y, θ) describe the time varying inertial pose of the leader and follower, respectively, it follows that $(x_r - x, y_r - y, \theta_r - \theta)$ define the pose

errors in inertial frame. From this, the error is transformed to the frame of leader (virtual robot) using transformation matrix \mathcal{R}_θ [3] to obtain (x_e, y_e, θ_e) , as shown in (6).

$$\begin{bmatrix} x_e \\ y_e \\ \theta_e \end{bmatrix} = \begin{bmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_r - x \\ y_r - y \\ \theta_r - \theta \end{bmatrix} \quad (6)$$

It is noted that $(v = \sqrt{\dot{x}^2 + \dot{y}^2}, \omega = \dot{\theta})$ denote the velocity, angular velocity of the follower and $(v_r = \sqrt{\dot{x}_r^2 + \dot{y}_r^2}, \omega_r = \kappa_r(s_r)v_r)$ denote the velocity, angular velocity of the leader (virtual robot) where $\kappa_r(s_r)$ is the curvature of reference trajectory parameterized as a function of arc length s_r traveled by the virtual robot. Using the kinematic model in (1) and tracking error in (6), we obtain the nonlinear error dynamics [3, 5] defined as :

$$\dot{x}_e = \omega y_e - v + v_r \cos \theta_e \quad (7a)$$

$$\dot{y}_e = -\omega x_e + v_r \sin \theta_e \quad (7b)$$

$$\dot{\theta}_e = \omega_r - \omega \quad (7c)$$

From (6-7), assuming perfect (infinite bandwidth) inner-loop velocity tracking system, we obtain the nonlinear outer-loop longitudinal and lateral position tracking control law [3] in (8) and (9), respectively. The associated nonlinear closed loop system is shown in (10-12)

$$v_c = v_r \cos \theta_e + k_x x_e \quad (8) \quad \dot{x}_e = -k_x x_e + \omega y_e \quad (10)$$

$$\omega_c = \omega_r + k_y v_r y_e + k_\theta v_r \sin \theta_e \quad (9) \quad \dot{y}_e = -\omega x_e + v_r \sin \theta_e \quad (11)$$

$$\dot{\theta}_e = -k_y v_r y_e - k_\theta v_r \sin \theta_e \quad (12)$$

where $k_x, k_y, k_\theta > 0$ are the controller gains. To perform stability analysis of closed loop system in (10-12) about equilibrium $\mathbf{q}_e = [x_e \ y_e \ \theta_e]^T = \mathbf{0}$, we use a positive definite Lyapunov function $V = \frac{1}{2}x_e^2 + \frac{1}{2}y_e^2 + \frac{1}{k_y}(1 - \cos \theta_e)$. By applying Lyapunov stability theorem [15], we show that $\dot{V} = -k_x x_e^2 - \frac{k_\theta v_r}{k_y} \sin^2 \theta_e \leq 0 \ \forall \ v_r > 0$. This ensures local stability. Further, using Barbalat's lemma [15], we prove that $\mathbf{q}_e = \mathbf{0}$ is locally asymptotically stable. By linearizing (10-12) about $\mathbf{q}_e = \mathbf{0}$, we obtain (13)

$$\dot{\mathbf{q}}_e = \mathbf{A} \mathbf{q}_e, \quad \mathbf{A} = \begin{bmatrix} -k_x & \omega_r & 0 \\ -\omega_r & 0 & v_r \\ 0 & -k_y v_r & -k_\theta v_r \end{bmatrix} \quad (13)$$

with the characteristic equation of matrix \mathbf{A} as $\Phi(s) = \det(s\mathbf{I} - \mathbf{A}) = s^3 + a_2 s^2 + a_1 s + a_0$ where $a_2 = k_x + k_\theta v_r$, $a_1 = k_y v_r^2 + k_\theta k_x v_r + \omega_r^2$, $a_0 = k_x k_y v_r^2 + k_\theta v_r \omega_r^2$. Using $k_x, k_y, k_\theta, v_r > 0$, we see that $a_0, a_1, a_2 > 0$ and $a_2 a_1 > a_0$. Hence, using Routh's stability criterion, $\mathbf{q}_e = \mathbf{0}$ is locally exponentially stable (LES).

To design tracking controller gains (k_x, k_y, k_θ) , we exploit the linear behavior of nonlinear closed loop system in (10-12) about operating point $\mathbf{q}_e = \mathbf{0}$. We use the same strategy used in [3], where the longitudinal and cascaded lateral tracking control system [3] are decoupled by assuming tracking error to be negligible. Under such assumptions, they can be approximated by a first-order and second-order system, respectively. Using this information, the characteristic equation $\Phi(s)$ in (13) takes the form $\Phi(s) = (s + 2\zeta_1 \omega_{n_1})(s^2 + 2\zeta_2 \omega_{n_2} s + \omega_{n_2}^2)$ where $s = -2\zeta_1 \omega_{n_1}$

captures the characteristics of first order longitudinal tracking control system and $s^2 + 2\zeta_2 \omega_{n_2} s + \omega_{n_2}^2 = 0$ captures the characteristics of second order lateral tracking control system [17]. By comparing the coefficients of s , we obtain the solution: $k_x = 2\zeta_1 \omega_{n_1}$, $k_y = \frac{\omega_{n_2}^2 - \omega_r^2}{v_r^2}$ and $k_\theta = \frac{2\zeta_2 \omega_{n_2}}{v_r}$. To design a good hierarchical inner-outer trajectory tracking control system, the bandwidths B_v, B_ω of inner-loop (v, ω) tracking system should be 5-10 times greater than the bandwidth B_{lon}, B_{lat} of corresponding outer-loop longitudinal and lateral position control system, respectively [3, 16]. For the longitudinal control system, this condition ($v \approx v_c$) is satisfied by choosing gain k_x according to guideline in (14) where $B_{lon} = k_x$. To achieve good angular velocity tracking properties ($\omega \approx \omega_c$) in lateral position control system, we choose $\zeta_2 = \frac{1}{\sqrt{2}}$ and bandwidth ω_{n_2} ($B_{lat} = \omega_{n_2}$) is chosen to follow the guideline in (15).

$$0 < k_x < \frac{B_v}{5} \quad (14) \quad 0 < \omega_{n_2} < \frac{B_\omega}{5} \quad (15)$$

We also note that (v_r, ω_r) of reference trajectory should satisfy (15) to ensure good closed-loop tracking performance (bandwidth ω_{n_2} depends on v_r, ω_r ; $\omega_{n_2} = \sqrt{\omega_r^2 + k_y v_r^2}$). Hence, this puts an upper bound on the aggressiveness of feasible reference trajectories. To study the effect of c.g. position on tracking performance, we consider a family of 3 “ \hookrightarrow ” shaped reference trajectories of varying cornering curvature ($\kappa_r^1 = \frac{\pi}{5}$, $\kappa_r^2 = \frac{\pi}{4}$, $\kappa_r^3 = \frac{\pi}{3}$, in rad/m) and $v_r = 2$ m/sec $\forall t$. The outer-loop tracking controller (8-9) is implemented on the LPV motion models of P_1 and P_2 consisting of kinematic model in (1) and inner-loop velocity tracking system in (5), depending on operating points (x_{eq}, u_{eq}) representing current state and control, in discrete time. Simulations are performed in MATLAB [20]. Using the inner-loop bandwidths B_v, B_ω (Table II) and controller design guidelines in (14-15), the outer-loop position controller gains are : $k_x = 1.8$, $k_y = 2.6$, $k_\theta = 2.7$, $\zeta_1 = \frac{1}{\sqrt{2}}$, $\omega_{n_1} = 1.27$, $g = 2.6$, $\zeta_2 = \frac{1}{\sqrt{2}}$, $\omega_{n_2} = 3.8$. For simulation purposes, the sampling frequency f_s is chosen as 15 Hz to satisfy the condition $2\pi f_s \geq 10 \max\{B_v, B_\omega\}$, in accordance with Shannon-Nyquist theorem[18].

Figure 6 depicts the comparison of trajectory tracking performances of Design I, II corresponding to dynamic models P_1, P_2 , respectively. We try to answer two questions, 1) Does designing a directionally unstable robot enhance directional maneuverability? 2) If so, how does enhanced directional maneuverability help track trajectories having aggressive cornering maneuvers? First, in Figure 6, we notice that the position of c.g. does not have any significant effect on longitudinal tracking error x_e . We also see that the lateral tracking errors $(\|y_e\|_{\mathcal{L}^\infty}, \|\theta_e\|_{\mathcal{L}^\infty})$ for Design II is less than Design I for each κ_r , supporting our claim that robot becomes more maneuverable and incurs less lateral tracking errors when it is directionally unstable. Further observation shows that the rate of change of $(\|y_e\|_{\mathcal{L}^\infty}, \|\theta_e\|_{\mathcal{L}^\infty})$ w.r.t. κ_r ($\frac{\Delta\|y_e\|_{\mathcal{L}^\infty}}{\Delta\kappa_r}, \frac{\Delta\|\theta_e\|_{\mathcal{L}^\infty}}{\Delta\kappa_r}$) is less for Design II than Design I. This shows that, even though the lateral tracking errors may increase with increase in

κ_r , designing the robot to be directionally unstable may limit this increase. From voltage plots, we observe that the control input of lateral dynamics $P_{\Delta e_a \rightarrow \omega}$ represented by Δe_a ($e_{a_r} - e_{a_l}$) is less for Design II than I for each κ_r during cornering. This shows that, depending on the direction of turn, voltage $e_{a_r,l}$ of motor on the outside is less for Design II than Design I and less likely to saturate during an aggressive cornering maneuver. Lastly, the rate of change of $e_{a_r,l}$ w.r.t κ_r ($\frac{\Delta e_{a_r,l}}{\Delta \kappa_r}$) is less for Design II than I during the cornering maneuver. All these observations support the idea that directionally unstable robot is more maneuverable than a directionally stable (sluggish) robot. However, we note that Design II experiences more overshoot than Design I in ω -channel, which increases as κ_r increases. This is expected since P_2 has a RHP pole at 1.15 in lateral dynamics ($P_{\Delta e_a \rightarrow \omega}$) [16]. A prefilter may be used to reduce the overshoot, at the cost of introducing lag into the system.

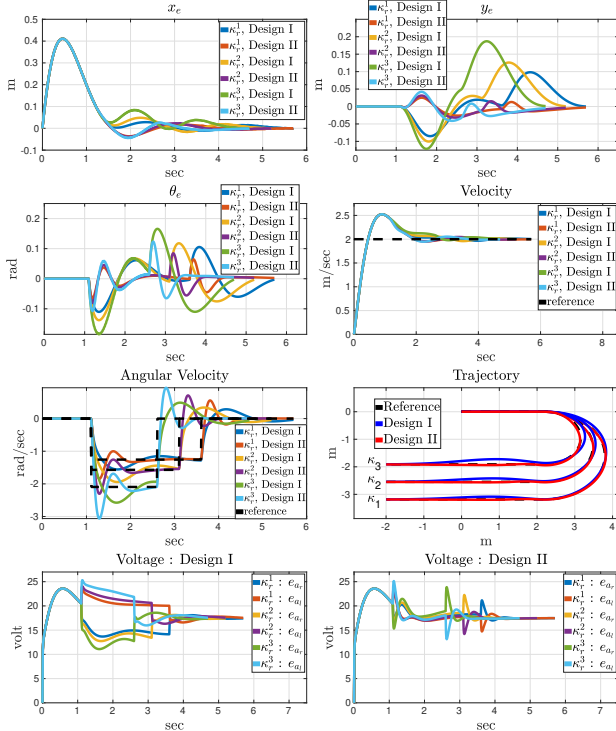


Fig. 6. Tracking Results: Variation of κ_r , $\kappa_r^1 = \frac{\pi}{5}$, $\kappa_r^2 = \frac{\pi}{4}$, $\kappa_r^3 = \frac{\pi}{3}$
B. Minimum Time Maneuvering Around Track

In this subsection, we present an outer-loop MPC based position control strategy (Figure 7) to minimize time taken to travel around a racetrack [4]. In this, we show that maneuverability of the robot plays a vital role in keeping the lap-time of robot to a minimum. As discussed in [4], maximizing the progress about center line of the racetrack can be an efficient way to avoid non-convexity of the original minimum-time around racetrack problem [4, 21]. Following this, we solve a linearly constrained convex quadratic programming problem (LCQP) at each time-instant to generate the necessary control input to maximize progress around track in a given prediction horizon. The first control signal (v_c, ω_c) of the optimal control sequence generated by the QP is fed into the (v, ω) tracking inner-loop control system in (5) as

reference commands and the process is repeated by shifting the horizon one time step into the future.

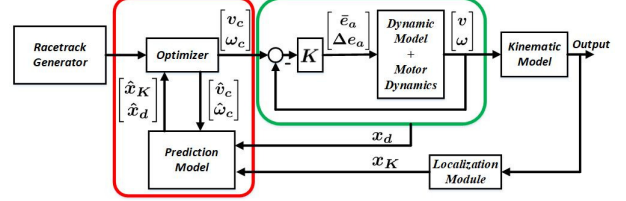


Fig. 7. Inner-Outer Loop Control Structure

To implement the QP, rather than using a single LTI model, we use a successive linearization [22] based LPV prediction model to closely capture the nonlinear behavior of the robot dynamics. This is because the position and orientation of the predicted trajectory can vary significantly while maneuvering about sharp corners on the track. In this paper, we use a discrete time parametric prediction model consisting of the kinematic model in (1) and inner-loop (v, ω) tracking system in (5). It is represented by (16), $\forall k = 1, \dots, H_p$

$$\begin{aligned} \hat{\mathbf{x}}(t+k) &= A\hat{\mathbf{x}}(t+k-1) + B\hat{\mathbf{u}}(t+k-1) + E \quad (16) \\ A &= \begin{bmatrix} A_K & \mathbf{0}_{3 \times 8} \\ \mathbf{0}_{8 \times 3} & A_d \end{bmatrix}, B = \begin{bmatrix} B_K \\ B_d \end{bmatrix}, E = \begin{bmatrix} E_K \\ E_d \end{bmatrix} \\ M_p &= \begin{bmatrix} \mathbf{0}_{2 \times 3} & I_{2 \times 2} & \mathbf{0}_{2 \times 6} \end{bmatrix}, \begin{bmatrix} v \\ \omega \end{bmatrix} = M_p \begin{bmatrix} \hat{\mathbf{x}}_K \\ \hat{\mathbf{x}}_d \end{bmatrix} \\ \hat{\mathbf{x}} &= \underbrace{\begin{bmatrix} x_K & y_K & \theta_K & v & \omega & i_{a_r} & i_{a_l} & x_{k1} & x_{k2} & x_{k3} & x_{k4} \end{bmatrix}^T}_{\hat{\mathbf{x}}_K} \underbrace{\begin{bmatrix} \hat{\mathbf{x}}_d \end{bmatrix}}_{\hat{\mathbf{x}}_d} \end{aligned}$$

where $\hat{\mathbf{x}} = [\hat{\mathbf{x}}_K \ \hat{\mathbf{x}}_d]^T$, $\hat{\mathbf{u}} = [\hat{v}_c \ \hat{\omega}_c]^T$, t is present time instant, H_p is prediction horizon, (A_K, B_K, E_K) represents the state space matrices of LPV kinematic model in (1) and (A_d, B_d, E_d) represents state space matrices of LPV inner-loop (v, ω) tracking system in (5). It is noted that these are parameter varying matrices depending on operating points $(\tilde{\mathbf{x}}, \tilde{\mathbf{u}})$ corresponding to state and control of previous predicted trajectory $\tilde{\mathbf{p}}(t)$. Through iterative substitution, the prediction model is simplified by expressing (16) in terms of $\Delta \hat{\mathbf{u}}$ as the only manipulated variable where $\Delta \hat{\mathbf{u}}(t+k) = \hat{\mathbf{u}}(t+k) - \hat{\mathbf{u}}(t+k-1)$. The modified prediction model is shown in (17) where $\mathbf{x}(0)$ and $\mathbf{u}(0)$ are the current state and control input at the start of prediction horizon, $\hat{\mathbf{X}}(t) = [\hat{\mathbf{x}}(t+1) \dots \hat{\mathbf{x}}(t+H_p)]^T$, $\Delta \hat{\mathbf{U}}(t) = [\Delta \hat{\mathbf{u}}(t+1) \dots \Delta \hat{\mathbf{u}}(t+H_p)]^T$.

$$\hat{\mathbf{X}}(t) = \underbrace{T\mathbf{x}(0) + V\mathbf{u}(0) + W}_{\text{constant term}} + \underbrace{Z}_{\text{variable term}} \Delta \hat{\mathbf{U}}(t) \quad (17)$$

$$\begin{aligned} W &= \begin{bmatrix} E_1 \\ A_2 E_1 + E_2 \\ \vdots \\ \sum_{j=1}^{H_p} \left(\prod_{i=j+1}^{H_p} A_i \right) E_j \end{bmatrix}, V = \begin{bmatrix} B_1 \\ A_2 B_1 + B_2 \\ \vdots \\ \sum_{j=1}^{H_p} \left(\prod_{i=j+1}^{H_p} A_i \right) B_j \end{bmatrix}, \\ T &= \begin{bmatrix} A_1 \\ A_1 A_2 \\ \vdots \\ \prod_{i=1}^{H_p} A_i \end{bmatrix}, Z = \begin{bmatrix} B_1 & 0 & \dots & 0 \\ A_2 B_1 + B_2 & B_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \sum_{j=1}^{H_p} \left(\prod_{i=j+1}^{H_p} A_i \right) B_j & \dots & \dots & B_{H_p} \end{bmatrix} \end{aligned}$$

In (17), $W \in \mathbb{R}^{11H_p \times 1}$, $V \in \mathbb{R}^{11H_p \times 2}$, $T \in \mathbb{R}^{11H_p \times 11}$ and $Z \in \mathbb{R}^{11H_p \times 2H_p}$. For future references, the constant term in (17) is represented by $\mathbf{b}(t)$. The actual control $\hat{\mathbf{U}}(t)$ is defined

in (18) where $\hat{\mathbf{U}}(t) = [\hat{\mathbf{u}}(t+1) \dots \hat{\mathbf{u}}(t+k)]^T$, $I_1 = I_{2 \times 2}$.

$$\hat{\mathbf{U}}(t) = \underbrace{\begin{bmatrix} I_1 \\ I_1 \\ \vdots \\ I_1 \end{bmatrix}}_{I_p} \mathbf{u}(0) + \underbrace{\begin{bmatrix} I_1 & 0 & \dots & 0 \\ I_1 & I_1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ I_1 & I_1 & \dots & I_1 \end{bmatrix}}_{K_p} \Delta \hat{\mathbf{U}}(t) \quad (18)$$

The model of racetrack (Figure 8) is taken from [4]. The center line of track is represented by a differentiable curve $C(s)$ parameterized as a function of arc-length s . For simulation purposes, the center path is discretized by a set of N points represented by arc-length s_i , $i \in 1 \dots N$ where $s_0 = 0$, $s_N = S$ and $T_{C,i} = C(s_i)$. $S \in \mathbb{R}$ is the total length of track. The tangential gradient at point $T_{C,i}$ is $\mathbf{f}_i = \Delta C(s_i)$. The orthogonal gradient at point $T_{C,i}$ is $\mathbf{n}_i = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} \mathbf{f}_i$. For each point $T_{C,i}$ on the center line, there exists points $T_{L,i}$ and $T_{R,i}$ on the left and right of $T_{C,i}$, respectively, at a distance of $\frac{w}{2}$ where w is the width of track. The points $T_{L,i}$ and $T_{R,i}$ $\forall i \in 1 \dots N$ form the left and right boundary points of the track and defined as: $T_{L,i} = C(s_i) + \frac{w}{2} \mathbf{n}_i$, $T_{R,i} = C(s_i) - \frac{w}{2} \mathbf{n}_i$. The region bounded by them forms the track area $T \in \mathbb{R}^2$.

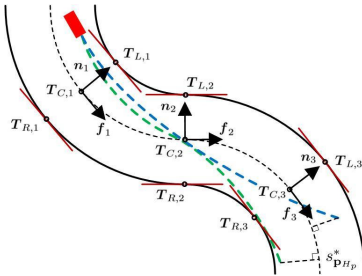


Fig. 8. Track representation according to [4]

To ensure that the predicted robot trajectory does not leave the track, linear constraints are used to confine the robot trajectory within the track area T . The predicted trajectory $\hat{\mathbf{p}}_k \forall k \in 1 \dots H_p$ at previous time step $t-1$ is used to compute the track constraints at present time step t . First, $\hat{\mathbf{p}}_k \forall k \in 1 \dots H_p$ is projected on the center track by solving (19) to obtain the index of point on the center track closest to $\hat{\mathbf{p}}_k$. We use a brute-force method ($O(NH_p)$) to solve (19).

$$j(\hat{\mathbf{p}}_k) = \arg \min_{m \in \{1 \dots N\}} \|T_{C,m} - \hat{\mathbf{p}}_k\|_{\mathcal{L}^2}, \quad \forall k \in 1 \dots H_p \quad (19)$$

Using the index $j(\hat{\mathbf{p}}_k)$, linear track constraints (20-21) are formulated by defining half-spaces at $T_{L,j(\hat{\mathbf{p}}_k)}$ and $T_{R,j(\hat{\mathbf{p}}_k)}$ $\forall k \in 1 \dots H_p$, as shown in Figure 8.

$$n_{j(\hat{\mathbf{p}}_k)}^T \hat{\mathbf{p}}_k - \rho \leq n_{j(\hat{\mathbf{p}}_k)}^T T_{L,j(\hat{\mathbf{p}}_k)}, \quad \forall i = 1 \dots H_p \quad (20)$$

$$-n_{j(\hat{\mathbf{p}}_k)}^T \hat{\mathbf{p}}_k - \rho \leq -n_{j(\hat{\mathbf{p}}_k)}^T T_{R,j(\hat{\mathbf{p}}_k)}, \quad \forall i = 1 \dots H_p \quad (21)$$

To ensure feasibility, the linear track constraints are softened by the use of an additional slack variable $\rho \in \mathbb{R}_{\geq 0}$. The slack variable is included in the objective function (26) as a quadratic term where q_ρ is the slack weight. Using ideas from exact penalty method, q_ρ is a large number so that the track constraints will not be violated unless (26) has no feasible solution to the optimization without slack constraints. By using prediction model (17), the track constraints (20-21) are transformed to affine inequality constraints (22-23) in $(\Delta \hat{\mathbf{U}} \rho)^T$ where $\hat{\mathbf{p}}(t) = C_{xy} \hat{\mathbf{X}}(t)$ represents the

predicted x-y coordinates of robot, $C_{xy} \in \mathbb{R}^{2H_p \times 11H_p}$, $T_L = (T_{L,j(\hat{\mathbf{p}}_1)} \dots T_{L,j(\hat{\mathbf{p}}_{H_p})})^T$, $T_R = (T_{R,j(\hat{\mathbf{p}}_1)} \dots T_{R,j(\hat{\mathbf{p}}_{H_p})})^T$, $N \in \mathbb{R}^{H_p \times 2H_p}$ where $N = \text{diag}(n_{j(\hat{\mathbf{p}}_1)}^T \dots n_{j(\hat{\mathbf{p}}_{H_p})}^T)$.

$$F_1 \begin{pmatrix} \Delta \hat{\mathbf{U}} \\ \rho \end{pmatrix} \leq g_1 \quad (22)$$

$$F_2 \begin{pmatrix} \Delta \hat{\mathbf{U}} \\ \rho \end{pmatrix} \leq g_2 \quad (23)$$

$$F_1 = [NC_{xy}Z \quad -1]$$

$$g_1 = -NC_{xy}\mathbf{b}(t) + NT_L$$

$$F_2 = [-NC_{xy}Z \quad -1]$$

$$g_2 = NC_{xy}\mathbf{b}(t) - NT_R$$

Note that this formulation would be highly inaccurate if the change of predicted position between two consecutive time-steps is large. To prevent failure of optimization in (26), the predicted position $\hat{\mathbf{p}}(t)$ is constrained to lie within a region $L \in \mathbb{R}$ around the previous predicted trajectory $\tilde{\mathbf{p}}(t)$ as shown in (26). The velocity input \hat{v}_c to the inner-loop (v, ω) tracking system is constrained to be 2 m/sec. The angular velocity input $\hat{\omega}_c$ is constrained to lie within a range defined by $\omega_{min} = v\kappa_{min}$, $\omega_{max} = v\kappa_{max}$, $v = 2$ m/sec where $(\kappa_{min}, \kappa_{max})$ are the minimum and maximum curvatures of the track area, respectively. The constraints $\hat{v}_c(t+k) = 2$ m/sec, $\omega_{min} \leq \hat{\omega}_c(t+k) \leq \omega_{max}$, $\forall k = 1 \dots H_p$ are written compactly in (26) using (18). The progress on the track is defined by the arc-length $s(\mathbf{p}_k)$ corresponding to the point on $C(s)$ closest to the robot position $\mathbf{p}_k \forall k \in 1 \dots H_p$.

$$s(\mathbf{p}_k) = \arg \min_{m \in [0, S]} \|C(m) - \mathbf{p}_k\|_{\mathcal{L}^2} \quad (24)$$

The goal is to maximize progress $s(\hat{\mathbf{p}}_{H_p})$ corresponding to the predicted position of robot at sampling instant $k = H_p$, as shown in Figure 8. To avoid nonlinearity while solving (24), we linearize $C(s) \approx T_{C,i} + \mathbf{f}_i(s - s_i)$. Solving (24) at $i = j(\hat{\mathbf{p}}_{H_p})$ using previous predicted trajectory, we obtain

$$s_{\hat{\mathbf{p}}_{H_p}} \approx s_{j(\hat{\mathbf{p}}_{H_p})} + \mathbf{f}_{j(\hat{\mathbf{p}}_{H_p})}^T (\hat{\mathbf{p}}_{H_p} - T_{C,j(\hat{\mathbf{p}}_{H_p})}) \quad (25)$$

Using (17), maximizing $s_{\hat{\mathbf{p}}_{H_p}}$ is equivalent to minimizing $(-\mathbf{f}_{j(\hat{\mathbf{p}}_{H_p})}^T C_{xy}^H Z \Delta \hat{\mathbf{U}}(t))$ where $\hat{\mathbf{p}}_{H_p} = C_{xy}^H \hat{\mathbf{X}}(t)$ is the predicted x-y robot position at $k = H_p$, $C_{xy}^H \in \mathbb{R}^{2 \times 11H_p}$. Combining all state, control and slack variable constraints, we formulate a QP in (26) which computes $\Delta \hat{\mathbf{U}}^*$ that maximizes progress around the track in a given prediction horizon H_p . Matrix $R \in \mathbb{R}^{2H_p \times 2H_p}$ in (26) penalizes any rapid variation in control input.

$$\min_{\Delta \hat{\mathbf{U}}, \rho} \begin{pmatrix} \Delta \hat{\mathbf{U}} \\ \rho \end{pmatrix}^T \begin{pmatrix} R & 0 \\ 0 & q_\rho \end{pmatrix} \begin{pmatrix} \Delta \hat{\mathbf{U}} \\ \rho \end{pmatrix} + \begin{pmatrix} -(C_{xy}^H Z)^T \mathbf{f}_{j(\hat{\mathbf{p}}_{H_p})} \\ 0 \end{pmatrix}^T \begin{pmatrix} \Delta \hat{\mathbf{U}} \\ \rho \end{pmatrix} \quad (26a)$$

$$\text{s.t.} \quad \begin{pmatrix} F_1 \\ F_2 \end{pmatrix} \begin{pmatrix} \Delta \hat{\mathbf{U}} \\ \rho \end{pmatrix} \leq \begin{pmatrix} g_1 \\ g_2 \end{pmatrix}, \rho \geq 0, \quad (26b)$$

$$|C_v(I_p \mathbf{u}_0 + K_p \Delta \hat{\mathbf{U}}) - \mathbf{v}_{max}| \leq \epsilon, \quad (26c)$$

$$\omega_{min} \leq C_\omega(I_p \mathbf{u}_0 + K_p \Delta \hat{\mathbf{U}}) \leq \omega_{max}, \quad (26d)$$

$$|\hat{\mathbf{p}}(t) - \tilde{\mathbf{p}}(t)| \leq (L \dots L)^T \quad (26e)$$

where $\mathbf{v}_{max} = (2 \dots 2)^T$. We solve (26) on the LPV motion models of P_1 (stable, $d > 0$) and P_2 (unstable, $d < 0$) consisting of kinematic model in (1) and inner-loop (v, ω) tracking system designed in Section III (see Equation (5)). Note that these motion models are varying about operating points $(\mathbf{x}_{eq}, \mathbf{u}_{eq})$ representing the state and control of robot at current time. Simulations are performed in MATLAB [20]

using the function `cplexqp()` from IBM ILOG CPLEX Optimization Studio [19]. Using Shannon-Nyquist theorem [18], to avoid aliasing errors, sampling frequency $f_c = \frac{1}{\Delta t}$ of the MPC controller is chosen to satisfy $2\pi f_c \geq 5 \max\{B_v, B_\omega\}$ where B_v and B_ω (Table II) are bandwidths of the (v, ω) tracking system. By choosing large q_ρ , the predicted trajectory is constrained within the track and stability, feasibility of (26) is ensured but not its performance. Choosing matrix R and trust region L appropriately, limits oscillations in the commands $(v_c, \omega_c)^*$ to inner-loop and subsequently voltage commands to actuators. Lower R and bigger trust region (L) lead to jittery controls. The MPC controller parameters are $f_c = 15$ hz, $q_\rho = 10^6$, $R = 1.5\mathbf{I}_{2H_p \times 2H_p}$, $L = 1.5$ m. We solve (26) for prediction horizon $H_p = \{10 \dots 25\}$ and analyze the performance of P_1 and P_2 , as shown in Figure 9, 10. First, comparing lap-times, we see that lap-times of both plants decrease as H_p increases. We also note that P_2 always takes less time to complete the lap than P_1 for varying H_p . The lap-times of both plants converge as H_p increases which is an indication of the predicted paths of P_1, P_2 approaching a globally optimal path. This is further corroborated by the trajectory plots in Figure 9 for $H_p = 15, 25$. We note that, in both figures, path followed by P_2 is more aggressive than P_1 at corners, an indication of superior maneuverability of P_2 over P_1 . It is to be noted that H_p cannot be arbitrarily increased to improve performance due to increased computational effort (more variables in (26)). Analyzing motor voltage data (Figure 10) for $H_p = 15$, we see that Δe_a required by P_2 is far less than P_1 during cornering maneuvers which ensures less chances of motor voltage saturation for P_2 than P_1 . Similar trends are observed for other prediction horizons.

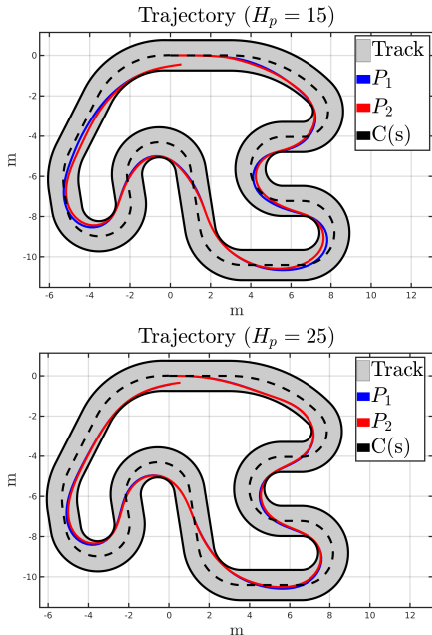


Fig. 9. Maximum Progress Trajectory (P_1 vs P_2)

V. CONCLUSION

In summary, the paper discusses how the c.g. position w.r.t. wheel axle affects the dynamics, stability and directional

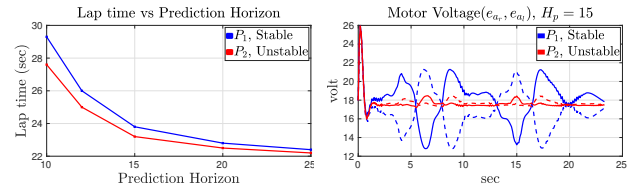


Fig. 10. Lap-time vs Horizon(Left), Motor voltage, $H_p = 15$ (Right)

maneuverability of robot. The paper concludes that a robot designed to have c.g. behind wheel axle ($d < 0$) is more directional maneuverable than a stable design ($d > 0$). The former design experiences less lateral tracking errors (y_e, θ_e) in tracking applications, has quicker lap-times around racetrack, has less chances of motor saturation than its stable counterpart due to enhanced directional maneuverability.

REFERENCES

- [1] J.Echols et al., Fundamental control system design issues for scramjet-powered hypersonic vehicles, AIAA GNC, pp. 1769, 2015.
- [2] S. Aoi et al., Advantage of straight walk instability in turning maneuver of multilegged locomotion: a robotics approach, *Scientific Reports*, vol.6, pp.30199, Nature Publishing Group, 2016.
- [3] C. Low, A trajectory tracking control scheme design for nonholonomic wheeled mobile robots with low-level control systems, IEEE 51st Annual Conf. on Decision and Control (CDC), pp. 536–543, 2012.
- [4] B. Alrifae, J. Maczjewski, Real-time Trajectory optimization for Autonomous Vehicle Racing using Sequential Linearization, 2018 IEEE Intelligent Vehicles Symposium (IV), pp.476–483, IEEE, 2018.
- [5] Y. Kanayama, Y. Kimura, F. Miyazaki, T. Noguchi, A stable tracking control method for an autonomous mobile robot, 1990.
- [6] Y. Zhao, S.L. BeMent, Kinematics, dynamics and control of wheeled mobile robots, IEEE Conf. on Robotics and Automation, 1992.
- [7] R. Dhaouadi, A. Hatab, Dynamic modeling of differential-drive mobile robots using lagrange and newton-euler methodologies: A unified framework, *Advances in Robotics & Auto.*, vol.2, no. 2, pp. 1–7, 2013.
- [8] Z. Li, *Modeling and Control of a Longitudinal Platoon of Ground Robotic Vehicles*, MS Thesis, Arizona State Univ., Tempe, AZ, 2016.
- [9] A.A. Rodriguez et al., Modeling, design & control of low-cost differential-drive robotic ground vehicles: Part I Single vehicle study, IEEE Conf. on Control Tech & App (CCTA), pp. 155–160, 2017.
- [10] K.Mondal et al., Comparison of Kinematic and Dynamic Model Based Linear Model Predictive Control of Non-Holonomic Robot for Trajectory Tracking: Critical Trade-offs Addressed, IASTED International Conference on Mechatronics and Control, 2019.
- [11] K.Mondal, Multivariable control of fixed wing aircrafts, MS Thesis, Dept. EE, ASU, Tempe, AZ, 2015.
- [12] K. Puttannaiah et al., A generalized mixed-sensitivity convex approach to hierarchical multivariable inner-outer loop control design subject to simultaneous input and output loop breaking specifications, American Control Conference (ACC), pp.5632–5637, IEEE, 2016.
- [13] P. Gahinet, P. Apkarian, Decentralized and fixed-structure \mathcal{H}^∞ control in MATLAB, pp.8205–8210, 50th IEEE Conference on Decision and Control and European Control Conference, IEEE, 2011.
- [14] F. Kuhne et al., Mobile robot trajectory tracking using model predictive control, IEEE Latin-American Robotics symposium, vol.51, 2005.
- [15] H.K. Khalil, *Nonlinear Systems*, Prentice Hall, 3rd Ed, 2002.
- [16] S. Skogestad, I. Postlethwaite, *Multivariable Feedback Control: Analysis and Design*, vol.2, Wiley New York, 2007.
- [17] C. Low, Design, implementation, and experimental validation of a cascaded trajectory tracking controller for nonholonomic car-like wheeled mobile robots with velocity and steering controllers in the loops, IEEE Conf. on Control Technology and Applications, 2017.
- [18] C.L. Phillips, H.T. Nagle, *Digital Control System Analysis and Design*, Prentice Hall Press, 2007.
- [19] Cplex, IBM ILOG, V12.1: User's Manual for CPLEX, International Business Machines Corporation, vol.46, no.53, pp.157, 2009
- [20] MathWorks, MATLAB Version 9.5 Release 2018b, 2018.
- [21] Q. Dinh, M. Diehl, An application of sequential convex programming to time optimal trajectory planning for a car motion, CDC, 2009.
- [22] M.A. Henson, Nonlinear model predictive control: current status and future directions, *Computers & Chemical Engineering*, vol.23, no.2, pp.187–202, Elsevier, 1998.