

Multiple-Lap Path Tracking for an Autonomous Race Vehicle via Iterative Learning Control

Nitin R. Kapania^{1,2} and J. Christian Gerdes¹

Abstract—Iterative learning control has been successfully used for several decades to improve the performance of control systems that perform a single repeated task. Using information from prior control executions, learning controllers gradually determine open-loop control inputs whose reference tracking performance can exceed that of traditional feedback-feedforward control algorithms. This paper considers iterative learning control for a previously unexplored field: autonomous racing. Racecars are driven multiple laps around the same sequence of turns while operating near the physical limits of tire-road friction, where steering dynamics become highly nonlinear and transient, making accurate path tracking difficult. However, because the vehicle trajectory is identical for each lap in the case of single-car racing, the nonlinear vehicle dynamics and unmodelled road conditions are repeatable and can be accounted for using iterative learning control, provided the tire force limits have not been exceeded. This paper describes the design and application of proportional-derivative (PD) and quadratically optimal (Q-ILC) learning algorithms for multiple-lap path tracking of an autonomous race vehicle. Simulation results are used to tune controller gains and test convergence, and experimental results are presented on an Audi TTS race vehicle driving several laps around Thunderhill Raceway in Willows, CA at lateral accelerations of up to 8 m/s^2 . Both control algorithms are able to correct transient path tracking errors and improve the performance provided by a reference feedforward controller.

I. INTRODUCTION

Iterative learning control (ILC) solves the control problem of tracking a specific reference command that is repeated many times. By using tracking error information from prior attempts, ILC techniques are used to gradually determine the open-loop control inputs that cause the system output to track a desired reference with minimal tracking error.

ILC algorithms work best when the exogenous system signals (i.e. disturbances and reference inputs) are constant from iteration-to-iteration. As a result, iterative learning controllers have frequently been applied in relatively structured robotics and automation environments, with recent publications considering piezoelectric positioning [1], robotic arm tracking [2], and microdeposition [3]. However, iterative learning control has recently expanded into applications outside of the traditional automation and process control setting. Chen and Moore [4] proposed a simple iterative learning scheme in 2006 to improve path-following of a ground vehicle with omni-directional wheels, where double integration of the previous feedback input was used to improve the feedforward signal. In 2013, Sun et al. proposed an iterative learning

controller for overspeed protection of high-speed trains [5]. Purwin and Andrea synthesized an iterative controller using least-squares methods to aggressively maneuver a quadrotor unmanned aerial vehicle from one state to another [6]. Iterative learning control has also extended into the neuromuscular and biological domains, with Rogers et al. [7] developing an ILC algorithm for robotically assisted stroke rehabilitation.

This paper presents autonomous race car piloting as a previously unexplored application area for iterative learning control. Race car drivers must drive multiple laps around the same sequence of turns on a closed track, while operating near the physical limits of tire-road friction to minimize lap times. At the limits of tire friction, the steering dynamics associated with lateral tracking of the desired path become highly nonlinear, and difficult-to-measure disturbances such as bank, grade and local friction variation of the road surface have a large effect on the transient dynamics of the vehicle. These factors make development of a suitable feedforward steering controller challenging. However, because operation of the race vehicle occurs over multiple laps, with the reference road curvature unchanging from lap-to-lap, the unknown transient disturbances and vehicle dynamics tend to be constant from lap to lap and can therefore be accounted for via iterative learning control. A notable exception occurs when the vehicle has significantly understeered due to lack of front tire force availability. In this case, additional steering will have no impact on path tracking.

This paper is further divided as follows. Section II introduces a linear model for the planar vehicle dynamics of a race car following a fixed reference path. Because the transfer function between the steering wheel input and the vehicle's path deviation is open-loop unstable, a stabilizing lanekeeping controller is added to the steering system and the closed loop dynamics are represented in the commonly used "lifted domain". Section III presents a PD-type iterative learning controller with a low-pass filter used to speed up convergence. Gain tuning and stability at low lateral accelerations are shown using lifted domain techniques, while nonlinear simulations are used to predict a desirable tracking response in the presence of high lateral acceleration. Section IV presents a quadratically optimal (Q-ILC) iterative learning controller, which has the benefit of explicitly accounting for changes in vehicle speed along the race track. Section V presents experimental data of both controllers implemented on an Audi TTS race vehicle at combined lateral/longitudinal accelerations of up to 8 m/s^2 .

¹Department of Mechanical Engineering, Stanford University, Stanford, CA 94305, USA

²Corresponding author: nkapania@stanford.edu

II. VEHICLE DYNAMICS AND PROBLEM OVERVIEW

For this paper, the objective of piloting an autonomous vehicle along a fixed race track in minimum time is divided into separate lateral and longitudinal vehicle control problems. The lateral controller uses the steering wheel input to track a desired “racing-line”, shown in Fig. 1a. The racing line is frequently represented by a path curvature function $\kappa(s)$ parametrized by distance along the track (Fig. 1b). The longitudinal controller tracks a desired speed profile that keeps the vehicle at a specified lateral-longitudinal acceleration magnitude, typically near the limits of tire-road friction (Fig. 1c). This paper focuses on learning the desired steer angle command δ_L over multiple laps in order to accurately track the reference path at high speeds, and we therefore assume minimum time velocity and curvature profiles have been computed using methods published in [8], and that tracking of the velocity profile is handled by a separate controller [9].

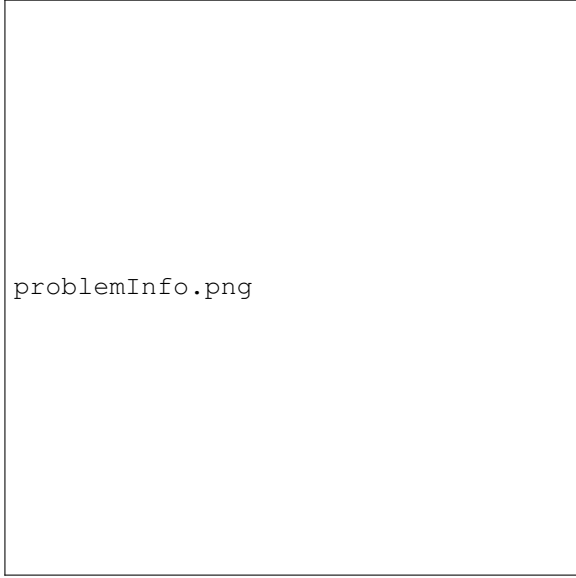


Fig. 1. (a) Overhead plot of racing line for Thunderhill Raceway, Willows CA, USA. (b) Racing line represented by curvature along distance traveled. (c) Velocity profile to keep vehicle at combined lateral/longitudinal acceleration of 8 m/s².

A. Lateral Vehicle Dynamics

Fig. 2 shows a schematic of a vehicle following a path with curvature profile $\kappa(s)$. The lateral deviation of the vehicle from the desired path (e) is the measured output for the ILC algorithms. Since iterative learning controllers determine appropriate *feedforward* control inputs over several iterations, it is necessary for the open-loop dynamics between the control input and control output to be asymptotally stable. For the case of active steering control, the transfer function between the vehicle steer angle (δ) and the lateral path error e is characterized by two poles at the origin, requiring the addition of a stabilizing feedback controller.

A lookahead controller provides the stabilizing feedback command for this paper. The “lookahead error” is defined

BikeModelSchematic.png

Fig. 2. Left: Schematic of bicycle model. Right: Diagram showing tracking error e , heading error $\Delta\Psi$, lookahead distance x_{LA} , and lookahead error e_{LA} .

by

$$e_{LA} = e + x_{LA}\Delta\Psi \quad (1)$$

Where $\Delta\Psi$ is the vehicle heading error and x_{LA} is the lookahead distance, typically 5-20 meters for autonomous driving. The resulting feedback control law is

$$\delta_{FB} = -k_P e_{LA} \quad (2)$$

with proportional gain k_P . The control law (2) is a natural extension of potential field lanekeeping, as described by Rossetter et al. in [10], which also provides heuristics for selecting k_P and x_{LA} . Desirable stability properties over significant tire saturation levels are demonstrated in [11].

With the feedback controller added, closed loop dynamics of the lateral path deviation are dependent on three other states: vehicle sideslip β , yaw rate r and heading error $\Delta\Psi$. For controller development and testing, these dynamics are given by the planar bicycle model:

$$\dot{\beta} = \frac{F_{yf} + F_{yr}}{mU_x} - r \quad \dot{r} = \frac{aF_{yf} - bF_{yr}}{I_z} \quad (3a)$$

$$\dot{e} = U_x(\beta + \Delta\Psi) \quad \dot{\Delta\Psi} = r - U_x\kappa \quad (3b)$$

Where U_x is the vehicle forward velocity and F_{yf} and F_{yr} the front and rear lateral tire forces. The vehicle mass and yaw inertia are denoted by m and I_z , while the geometric parameters a and b are shown in Fig. 2.

As automotive racing frequently occurs near the limits of tire force saturation, lateral tire force is modeled using the nonlinear Fiala brush tire model, assuming a single

coefficient of friction and a parabolic force distribution [12]. The lateral tire forces are functions of the front and rear tire slip angles α_F and α_R .

$$F_y = \begin{cases} -C_\alpha \tan \alpha + \frac{C_\alpha^2}{3\mu F_z} |\tan \alpha| \tan \alpha \\ -\frac{C_\alpha^3}{27\mu^2 F_z^2} \tan^3 \alpha, & |\alpha| < \arctan\left(\frac{3\mu F_z}{C_\alpha}\right) \\ -\mu F_z \operatorname{sgn} \alpha, & \text{otherwise} \end{cases} \quad (4)$$

where μ is the surface coefficient of friction, F_z is the normal load, and C_α is the tire cornering stiffness. The linearized tire slip angles are given by

$$\alpha_F = \beta + \frac{ar}{U_x} - \delta \quad (5a)$$

$$\alpha_R = \beta - \frac{br}{U_x} \quad (5b)$$

B. Linear Time Varying Model in the Lifted Domain

While the nonlinear tire model presented in (4) captures the effect of tire saturation at the limits of handling, established methods for design and analysis of iterative learning controllers requires a linear system description. With the assumption of low lateral acceleration, a simple linear tire model is given by,

$$F_y = -C_\alpha \alpha \quad (6)$$

The lateral error dynamics of the vehicle in response to a *feedforward* steer angle input (recall that there is also a feedback steer angle δ_{FB} in the loop) can then be represented by the continuous, linear time varying (LTV) state space model,

$$\dot{x}(t) = A_c(t)x + B_c(t)\delta_L + d_c(t) \quad (7a)$$

$$e = C_c x(t) \quad (7b)$$

$$x = [e \ \Delta\Psi \ r \ \beta]^T \quad (7c)$$

Where δ_L is the learned steering input. The LTV framework is chosen to account for the variation in vehicle velocity U_x along the track (Fig. 1). While the longitudinal dynamics are not explicitly accounted for in (3a), allowing for the state matrices to vary with measured values of U_x enables more accurate controllers and analysis. The time-varying matrices A_c , B_c , and C_c are given by

$$A_c(t) = \begin{bmatrix} 0 & U_x(t) & 0 & U_x(t) \\ 0 & 0 & 1 & 0 \\ \frac{-ak_F C_F}{I_z} & \frac{-ak_F x_L A C_F}{I_z} & \frac{-(a^2 C_F + b^2 C_R)}{U_x(t) I_z} & \frac{b C_R - a C_F}{I_z} \\ \frac{-k_F C_F}{m U_x(t)} & \frac{-k_F x_L A C_F}{m U_x(t)} & \frac{b C_F - a C_R}{m U_x(t)^2} - 1 & \frac{-(C_F + C_R)}{m U_x(t)} \end{bmatrix} \quad (8)$$

$$B_c(t) = [0 \ 0 \ \frac{a C_F}{I_z} \ \frac{C_F}{m U_x(t)}]^T \quad (9)$$

$$C_c = [1 \ 0 \ 0 \ 0] \quad (10)$$

The disturbance d_c , is assumed constant from lap to lap and is given by

$$d_c(t) = [0 \ -\kappa U_x(t) \ 0 \ 0]^T \quad (11)$$

The next step is to discretize (7) by the controller sample time T_s , resulting in the discrete time system,

$$x_j(k+1) = A(k)x_j(k) + B(k)\delta_j^L(k) \quad (12)$$

$$e_j(k) = Cx_j(k) \quad (13)$$

Where $k = 1 \dots N$ is the time sample index, and $j = 1 \dots M$ is the number of iterations (i.e. the number of laps around the track). Development and analysis of the iterative learning controllers in the next section will be made easier by representing the system dynamics in the “lifted-domain”, where the inputs and outputs are stacked into arrays and related by matrix multiplication, as follows:

$$\mathbf{e}_j = \mathbf{P}\delta_j^L + \mathbf{d}_j \quad (14a)$$

$$\delta_j^L = [\delta_j^L(0) \dots \delta_j^L(N-1)]^T \quad (14b)$$

$$\mathbf{e}_j = [e_j(1) \dots e_j(N)]^T \quad (14c)$$

Where the elements of the $N \times N$ matrix \mathbf{P} are given by,

$$p_{lk} = \begin{cases} 0 & \text{if } l < k \\ CB(k) & \text{if } l = k \\ CA(l)A(l-1) \dots A(k)B(k) & \text{if } l > k \end{cases} \quad (15)$$

Note that for the case where U_x is constant, we have a linear time invariant (LTI) system given by only N elements, with $p(k) = CA^{k-1}B$, and the resulting lifted-domain matrix \mathbf{P} is Toeplitz.

III. CONTROLLER DESIGN

With the representation of the steering dynamics for a given lap represented by (14), the next step is to design algorithms that determine the learned steering input δ_{j+1}^L for the next lap, given the error response from the completed lap \mathbf{e}_j . A common framework for iterative learning algorithms is to choose δ_{j+1}^L as follows [13],

$$\delta_{j+1}^L = \mathbf{Q}(\delta_{L,j} - \mathbf{L}\mathbf{e}_j) \quad (16)$$

Where \mathbf{Q} is the $N \times N$ filter matrix, and \mathbf{L} is the $N \times N$ learning matrix. In following sections, the matrices \mathbf{Q} and \mathbf{L} will be obtained by designing a PD type iterative learning controller as well as a quadratically optimal (Q-ILC) learning controller.

A. Proportional-Derivative Controller

The proportional-derivative iterative learning controller computes the steering addition $\delta_j^L(k)$ at a given time index k based on the error $e_{j-1}(k)$ and the change in error at the same time index from the previous lap,

$$\delta_{j+1}^L(k) = \delta_j^L(k) - k_p e_{j-1}(k) - k_d(e_{j-1}(k) - e_{j-1}(k-1)) \quad (17)$$

Where k_p and k_d are the proportional and derivative gains. In the lifted domain representation from (16), the resulting learning matrix \mathbf{L} is given by

$$\mathbf{L} = \begin{bmatrix} -(k_p + k_d) & 0 & & \\ k_d & \ddots & & \\ 0 & k_d & -(k_p + k_d) & \end{bmatrix} \quad (18)$$

An important consideration in choosing the gains k_p and k_d is achieving a monotonic decrease in the path tracking error on every lap. This property, known as monotonic convergence, occurs if the following condition is met [13]:

$$\gamma \triangleq \bar{\sigma}(\mathbf{PQ}(I - \mathbf{LP})\mathbf{P}^{-1}) < 1 \quad (19)$$

Where $\bar{\sigma}$ is the maximum singular value. In this case, the value of γ provides an upper bound on the decrease in the tracking error norm from lap to lap, i.e.

$$\|e_\infty - e_{j+1}\|_2 \leq \gamma \|e_\infty - e_j\|_2 \quad (20)$$

where e_∞ is the converged path tracking error. The monotonic stability condition differs from its weaker counterpart of asymptotic stability in that we are guaranteed that the path tracking error on the first lap (i.e. with no added learning input) is the worst case performance.

Fig. 3 shows values of γ for both an unfiltered PD controller ($\mathbf{Q} = I$), and for a PD controller with a 2 Hz low pass filter. The γ values are plotted as a contour map against the controller gains k_p and k_d . Addition of the low-pass filter assists in achieving controller monotonic stability by removing oscillations that are frequently generated by iterative learning controllers when trying to remove small reference tracking errors after several iterations. Since the filtering occurs when generating a control signal for the next lap, the filter \mathbf{Q} can be zero-phase.

However, testing for linear stability is insufficient controller design given that racing frequently occurs near the limits of vehicle handling, when the vehicle dynamics are described by the nonlinear equations of motion presented in Section II. To test the PD controller feasibility, the vehicle tracking performance over multiple laps is simulated using the path curvature and speed profile shown in Fig. 1. Simulated results for the root-mean-square (RMS) tracking error are shown in Fig. 4 for both the linear state dynamics prescribed by (13) and the nonlinear dynamics model given by (3) and (4). The results indicate that as the vehicle corners closer to the limits of handling, the tracking performance of the ILC degrades relative to the expected performance given by the linear model, but can still be expected to converge over relatively few iterations.



Fig. 3. Values of convergence bound γ vs. k_p and k_d for PD iterative learning controller with (left) no filtering and (right) with a 2 Hz low-pass filter. Lower values of γ correspond to faster convergence. Shaded regions correspond to systems with monotonic stability.

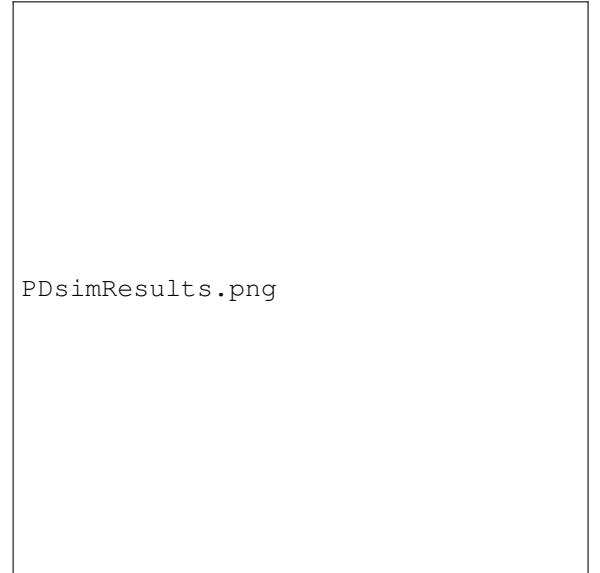


Fig. 4. Simulated results for root-mean-square path tracking error for PD iterative learning control at several values of vehicle lateral acceleration. Controller gains are $k_p = .05$, $k_d = .05$. Results from a nonlinear vehicle dynamics simulation are compared to results from the linear vehicle model.

QILCsimResults.png

Fig. 5. Simulated results for root-mean-square path tracking error for Q-ILC at several values of vehicle lateral acceleration, with $T = R = I$ and $S = 100I$. Results from a nonlinear vehicle dynamics simulation are compared to results from the linear vehicle model.

B. Quadratically Optimal Controller

An alternate approach to determining the learned steering input δ_j^L is to minimize a quadratic cost function for the next lap:

$$J_{j+1} = e_{j+1}^T T e_{j+1} + \delta_{j+1}^{L T} R \delta_{j+1}^L + \Delta_{j+1}^T S \Delta_{j+1} \quad (21)$$

Where $\Delta_{j+1} = \delta_{j+1}^L - \delta_j^L$ and the $N \times N$ matrices T , R , and S are weighting matrices. This formulation allows the control designer to weight the competing objectives of minimizing tracking error, control effort, and change in the control signal from lap to lap. While constraints can be added to the optimization problem, the unconstrained problem in (21) can be solved analytically [13] to obtain desired controller and filter matrices:

$$\mathbf{Q} = (P^T T P + R + S)^{-1} (P^T T P + S) \quad (22a)$$

$$\mathbf{L} = (P^T T P + S)^{-1} P^T T \quad (22b)$$

An advantage to the quadratically optimal control design over the simple PD controller is that the controller matrices \mathbf{Q} and \mathbf{L} take the linear time-varying dynamics \mathbf{P} into account. This allows the iterative learning algorithm to take into account changes in the steering dynamics due to changes in vehicle velocity. However, a disadvantage is that determining new \mathbf{Q} and \mathbf{L} matrices every lap requires resolving (22), which can be computationally expensive for fast sampling rates.

Fig. 5 shows simulated results for the quadratically optimal ILC controller. The results are similar to those in Fig. 4 in that the predicted controller performance is worse when the simulation accounts for nonlinear tire dynamics. However,

shelleyPic.png

Fig. 6. Autonomous Audi TTS with electronic power steering, brake booster, and throttle by wire.

the simulation still shows a rapid decrease in path tracking error over the first ten laps, with RMS errors on the order of 8-9 cm at lateral accelerations of .8 g .

IV. EXPERIMENTAL RESULTS

Experimental data for both controllers was collected over multiple laps at Thunderhill Raceway, a 3 mile paved racetrack in Willows, CA, with track boundaries shown in Fig. 1a. The experimental testbed is an autonomous Audi TTS equipped with an electronic power steering motor, active brake booster, and throttle by wire Fig. 6. Vehicle and controller parameters are shown in Table 1.

An integrated Differential Global Positioning System (DGPS) and Inertial Measurement Unit (IMU) is used to obtain vehicle state information, and a localization algorithm determines the lateral path tracking error e , heading error $\Delta\Psi$, and distance along the desired racing line. The steering controller updates at 200 Hz, and the iterative steering corrections are calculated after each lap from data downsampled to 10 Hz. The corrections are then applied as a function of distance along the track using an interpolated lookup table. For safety reasons, a steady-state feedforward steering algorithm [14] is also applied to keep the tracking error on the first lap below 1 m.

Fig. 7 shows the applied iterative learning signals and resulting path tracking error over four laps using the PD learning algorithm. The car is driven aggressively at peak lateral/longitudinal accelerations of 8 m/s^2 . On the first lap, despite the incorporation of a feedforward-feedback controller operating at a high sampling rate, several transient spikes in tracking error are visible due to the underdamped tire dynamics near the limits of handling. However, the

TABLE I
VEHICLE PARAMETERS

Parameter	Symbol	Value	Units
Vehicle mass	m	1500	kg
Yaw moment of inertia	I_z	2250	$\text{kg} \cdot \text{m}^2$
Front axle to CG	a	1.04	m
Rear axle to CG	b	1.42	m
Front cornering stiffness	C_F	160	$\text{kN} \cdot \text{rad}^{-1}$
Rear cornering stiffness	C_R	180	$\text{kN} \cdot \text{rad}^{-1}$
Lookahead Distance	x_{LA}	15.2	m
Lanekeeping Gain	k_{LK}	.053	rad m^{-1}
Lanekeeping Sample Time	t_s	.005	s
ILC Sample Time	T_s	.1	s
PD Gains	k_p and k_d	.02 & .4	rad m^{-1}
Q-ILC Matrix	T and R	I	-
Q-ILC Matrix	S	$100 I$	-

iterative learning algorithm is able to significantly attenuate these transient spikes over just two or three laps. Similar qualitative results occur for the quadratically optimal ILC.

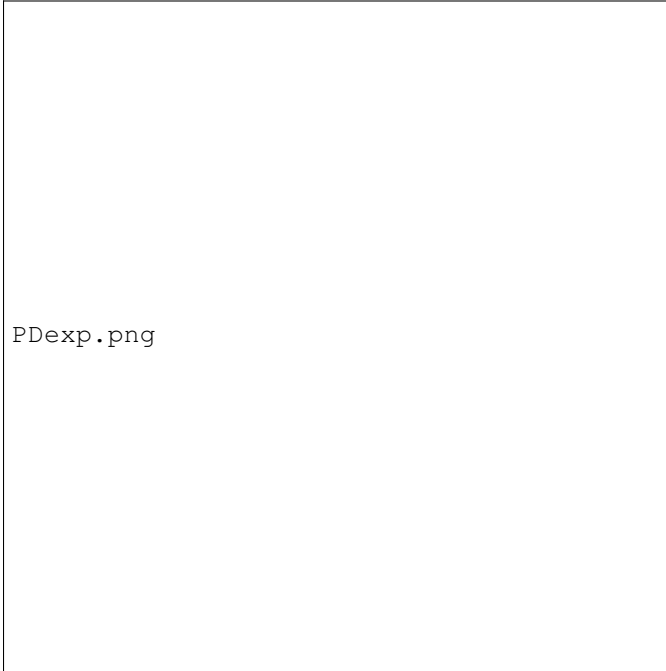


Fig. 7. Experimental results for path tracking error with PD learning controller, at peak lateral accelerations of 8 m/s^2 .

In Fig. 8, results are shown for versions of the velocity profile in Fig. 1 that are scaled to achieve different vehicle accelerations. The results show that at low vehicle accelerations, when vehicle dynamics are accurately prescribed by a linear tire model, the feedback-feedforward controller is able to keep the vehicle close to the desired path, leaving little room for improvement through iterative learning control. However, as the speed profile becomes more aggressive, the path tracking degrades in the presence of highly transient tire dynamics, and iterative learning control can be effectively used to obtain tight path tracking path over two or three laps of racing. In practice, the performance of both the PD



Fig. 8. Experimental results for both controller types over a variety of lateral accelerations.

algorithm and quadratically optimal algorithms are similar, and an important observation is that the RMS tracking error increases slightly from lap-to-lap at the end of some tests. While not predicted in simulation, this behavior is not unreasonable given unmodelled sensor noise and disturbances that vary from lap to lap. More refined tuning of the gain matrices may be able to prevent this RMS error increase, or the ILC algorithm can be stopped after several iterations once the tracking performance is acceptable.

V. CONCLUSION

This paper demonstrates the application of iterative learning control (ILC) methods to achieve accurate steering control of an autonomous race car over multiple laps. Two different algorithms, proportional-derivative (PD) and quadratically optimal (Q-ILC) learning control are tested in simulation and then used to experimentally eliminate path tracking errors caused by the highly transient nature of the vehicle lateral dynamics near the limits of tire-road friction. Both learning algorithms provide comparable lap-to-lap tracking performance, although the PD method is computationally fast enough to run in real time. Two clear limitations of the presented iterative learning controllers present avenues for future work. While representing the vehicle dynamics with a linear, time-varying model allows for the quadratically optimal ILC algorithm to account for varying longitudinal speed, a better approach is to linearize the vehicle dynamics at each point on the track and create an affine time-varying model whose path tracking error can be minimized. Furthermore, applying a steering wheel input to eliminate lateral errors will work only if the vehicle is near the limits of handling, but has not fully saturated the available tire force and entered a limit understeer condition. In this case, a separate controller must be developed to modify the

racing line and velocity profile for future laps in order to reduce the vehicle cornering forces.

ACKNOWLEDGMENT

This research is supported by the National Science Foundation Graduate Research Fellowship Program (GRFP). The authors would like to thank the members of the Dynamic Design Lab at Stanford University and the Audi Electronics Research Lab.

REFERENCES

- [1] D. Huang, J.-X. Xu, V. Venkataramanan, and T. Huynh, "High-performance tracking of piezoelectric positioning stage using current-cycle iterative learning control with gain scheduling," *IEEE Trans. on Industrial Electronics*, vol. 61, no. 2, pp. 1085–1098, 2014.
- [2] C. T. Freeman and Y. Tan, "Iterative learning control with mixed constraints for point-to-point tracking," *IEEE Trans. on Control Systems Technology*, vol. 21, no. 3, pp. 604–616, 2013.
- [3] D. J. Hoelzle, A. G. Alleyne, and A. J. W. Johnson, "Basis task approach to iterative learning control with applications to micro-robotic deposition," *IEEE Trans. on Control Systems Technology*, vol. 19, no. 5, pp. 1138–1148, 2011.
- [4] Y. Q. Chen and K. L. Moore, "A practical iterative learning path-following control of an omni-directional vehicle," *Asian Journal of Control*, vol. 4, no. 1, pp. 90–98, 2002.
- [5] H. Sun, Z. Hou, and D. Li, "Coordinated iterative learning control schemes for train trajectory tracking with overspeed protection," *IEEE Trans. on Automation Science and Engineering*, vol. 10, no. 2, pp. 323–333, 2013.
- [6] O. Purwin and R. DAndrea, "Performing and extending aggressive maneuvers using iterative learning control," *Robotics and Autonomous Systems*, vol. 59, no. 1, pp. 1–11, 2011.
- [7] E. Rogers, D. H. Owens, H. Werner, C. T. Freeman, P. L. Lewin, S. Kichhoff, C. Schmidt, and G. Lichtenberg, "Norm-optimal iterative learning control with application to problems in accelerator-based free electron lasers and rehabilitation robotics," *European Journal of Control*, vol. 16, no. 5, pp. 497–522, 2010.
- [8] P. A. Theodosis and J. C. Gerdes, "Generating a racing line for an autonomous racecar using professional driving techniques," in *Dynamic Systems and Control Conference*, 2011, pp. 853–860.
- [9] K. Kritayakirana, "Autonomous vehicle control at the limits of handling," Ph.D. dissertation, Stanford University, 2012.
- [10] E. J. Rossetter and J. C. Gerdes, "A study of lateral vehicle control under a virtual force framework," in *Proc. International Symposium on Advanced Vehicle Control, Hiroshima, Japan*, 2002.
- [11] K. L. Talvala and J. C. Gerdes, "Lanekeeping at the limits of handling: Stability via lyapunov functions and a comparison with stability control," in *Dynamic Systems and Control Conference*, 2008, pp. 361–368.
- [12] H. B. Pacejka, *Tire and Vehicle Dynamics*, 3rd ed. Butterworth-Heinemann, 2012.
- [13] D. A. Bristow, M. Tharayil, and A. G. Alleyne, "A survey of iterative learning control," *Control Systems, IEEE*, vol. 26, no. 3, pp. 96–114, 2006.
- [14] N. R. Kapania and J. C. Gerdes, "An autonomous lanekeeping system for vehicle path tracking and stability at the limits of handling," in *12th International Symposium on Advanced Vehicle Control*, 2014.