Experiments in Feedback Linearized Iterative Learning-Based Path Following for Center-Articulated Industrial Vehicles*

Lukas G. Dekker[†], Joshua A. Marshall[‡]

Offroad Robotics

Queen's University

Kingston, ON Canada K7L 3N6

{lukas.dekker, joshua.marshall}@queensu.ca

+1 613 533-2921

Johan Larsson
Epiroc Rock Drills AB
Klerkgatan 21, SE-701 91
Örebro, Sweden
johan.larsson@epiroc.com
+46 19-670 72 62
January 24, 2019

*This work was supported in part by the Natural Sciences and Engineering Research Council of Canada (NSERC) under project RGPIN-2015-04025, the Swedish Knowledge Foundation (KK-stiftelsen) under project 20150282, and by Epiroc Rock Drills AB (Sweden).

[†]The first author is now with MacLean Engineering, 1000 Raglan St, Collingwood, ON L9Y 3Z1 Canada.

[‡]Corresponding author. This work was performed while the second author was a Visiting Professor with the Center for Applied Autonomous Sensor Systems (AASS) at Örebro University, Örebro, Sweden.

Abstract

This paper describes the design, industrial application, and field testing of a technique for autonomous wheeled-vehicle path following that uses iterative learning control (ILC) in a feedback linearized space. One advantage of this approach is that ILC is used without having to employ approximate linearization at every time step. The main contribution of this paper is the unique field experiments that employed two large industrial-scale center-articulated underground mining vehicles. The described field work not only tested the underlying technique on commercial vehicles, but also presents a method for parallel speed learning, wherein the speed is adjusted over subsequent learning trials to improve cycle productivity. Finally, presented are field results for an approach to pre-learning through simulation prior to deployment in the field in order to reduce the initial path-following errors.

1 Introduction

For many industry applications, such as in mining and construction, large center-articulated vehicles (e.g., see Figure 1) are often used to haul materials from one place to another in challenging workplace conditions (Marshall, Barfoot, & Larson, 2008). What is more, these vehicles often repeatedly drive the same route,

over and over again, thus providing an opportunity for autonomous vehicles to potentially improve their performance with every pass. This paper presents results from field experiments in a representative environment on the application of an autonomous wheeled-vehicle path following technique that uses iterative learning control (ILC) in a feedback linearized (FBL) space—which we refer to as ILC+FBL—to achieve both reductions in path-following errors and improvements in vehicle speed over multiple passes (or trials) of the same route.

The work presented herein builds on our preliminary work on wheeled-vehicle path following (Dekker, Marshall, & Larsson, 2017) and on the work of Ostafew, Schoellig, and Barfoot (2013). The baseline control strategy used in this paper follows the work of Marshall et al. (2008), wherein a two-timescale control architecture is utilized. Thus, path following is conducted by an outer-loop control and navigation system and low-level control of vehicle dynamics (i.e., steering rate and vehicle speed regulation) is executed by a suite of inner-loop controllers. By using this architecture for such large industrial-scale vehicles, limitations or destabilization of path-following performance may occur as a result of low inner-loop bandwidth. This is expected due to the slow actuator and vehicle dynamics of such large vehicles. Attempts have been made to account for this limited low-level bandwidth such as a “look ahead” patch implemented in Section 4.2 of the

work by Marshall et al. (2008) that uses heading error feedback from a point ahead (in time) along the desired route, or through the use of model predictive control (Ostafew, Schoellig, Barfoot, & Collier, 2016).

By its nature, ILC allows for similar anticipatory controller behavior because the entire error signal from the previous run is available, thus errors from any point along the desired route can be considered when computing corrections to the next control input at a given discrete step (D. Wang & Ye, 2005). As a result, controller design for path following can be conducted by using a kinematic model while ILC iteratively corrects inputs to account for unmodeled vehicle dynamics, environmental disturbances (e.g., wheel slip or rough ground), and extreme path geometries. Many works have made attempts to account for such environmental disturbances (Nayl, Nikolakopoulos, & Gustafsson, 2012; Ishigami, Nagatani, & Yoshida, 2006; Helmick, Cheng, Clouse, Matthies, & Roumeliotis, 2004; Balakrishna & Ghosal, 1995) and propose control schemes to account for wheel slip in rough unmodeled terrains for vehicles such as the LHD used in mining applications as well as planetary rovers. Wheel slip is accounted for in two ways. One approach is to attempt to model the terrain/wheel interaction and how, dynamically, the vehicle may respond in the case of wheel slip. The path following controller is then designed based on this dynamic slip model and thus controller

design becomes a more complex task. The second approach is to use localization methods to determine when slip has occurred, and if so a temporary desired path is formulated that directs the vehicle back to the original desired path. Conversely, as proposed in this paper, the implementation of an ILC augmentation allows for similar slip compensation (Ostafew et al., 2013) by learning from previous trials where these slips may occur and for correcting steering inputs based on these slip errors. Thus, no additional vehicle modeling or path alteration is necessary.

In the work presented by this paper, ILC+FBL is simplified by designing the iterative learning controller in a fully feedback linearized error space. As a result, the error system does not change over time, thus facilitating controller design. In addition, this ILC+FBL framework is efficient due to the fact that learned corrections are applied to the transformed control input prior to conducting only a single back-computation to the nonlinear system control input (steering rate in this case). As its primary contribution, this paper presents field results from the application of a parallel speed learning algorithm (ISLC), similar to speed scheduling achieved by Ostafew, Schoellig, Barfoot, and Collier (2014) but heavily based on the work by Boe and Hannaford (1998). Validation of these schemes was achieved first through simulation trials and then through field tests by using two dynamically different industrial-scale mining vehicles outfitted with teach-and-repeat,

navigation, localization and control functionality through sensor systems similar to those described by Marshall et al. (2008). The vehicles used in the work presented by this paper are the Epiroc ST14 and ST18 underground loaders, shown in Figure 1.

1.1 Motivation and Additional Related Work

Path following involves tracking a desired path (or route) irrespective of time. There are many works that study path following and present successful results (e.g., recently Normey-Rico, Alcalá, Gómez-Ortega, & Camacho, 2001; Paden, Cap, Yong, Yershov, & Frazzoli, 2016; del Río, Jiménez, Sevillano, Vicente, & Balcells, 2001; Harmouche, Laghrouche, & Chitour, 2014; Lapierre, Soetanto, & Pascoal, 2006; Pakshin, Emelianova, Emelianov, Gałkowski, & Rogers, 2016, and many others). On the other hand, ILC has seen many applications in the realm of industrial manufacturing as a result of the inherently repetitive nature of the required tasks. Thus, ILC is a well established control methodology with many practical applications in this field. Bristow, Tharayil, and Alleyne (2006) give a recent survey of ILC designs and applications, and some examples include tracking control for chain conveyors (Barton, Lewin, & Brown, 2000) and manipulator-type robots (D. Wang & Ye, 2005; Elci, Longman, Phan, Juang, & Ugoletti, 1994; Wallén,

(a) The ST14 loader has a carrying capacity of 14 tonnes.

(b) The ST18 loader has a carrying capacity of 18 tonnes.

Figure 1: Epiroc Scooptram (ST) underground loaders at the Kvarntorp underground mine near Örebro, Sweden. 8

Norrlöf, & Gunnarsson, 2008; van den Berg et al., 2010; Zhao, Lin, Xi, & Guo, 2015). Others have made noteworthy contributions in, for example, healthcare applications (Freeman, 2015, 2017). Other challenging applications of ILC arise in the field of tracking control for mobile robotic ground vehicles, yet successful results have been achieved regarding the trajectory tracking problem (Moore & Bahl, 1999; Shao & Chen, 2010; H. Wang, Dong, & Wang, 2016; Xi, Li, & Xiao, 2014; Kang, Lee, & Han, 2005; Chen & Moore, 2002). In these examples, experimental validation was conducted through simulation, with few results from real autonomous vehicle trials in representative environments.

This paper examines the path-following problem and thus, is heavily based on the methodology employed by Ostafew et al. (2013) who combine ILC with feedback linearization (ILC+FBL) for path-following control of a rugged differential drive wheeled vehicle in an outdoor environment. ILC+FBL is not a new concept; it has been applied to both manipulator-type mechanisms (Marino & Tomei, 2009; Freeman, 2015; Pakshin, Emelianova, Emelianov, Gałkowski, & Rogers, 2016) and mobile robots (Ostafew et al., 2013). Novelty in this paper, however, is demonstrated by applying ILC in the fully feedback-linearized space (slightly different from the approach by Ostafew et al. (2013)) and in applying, for a mobile robot, a parallel learning algorithm (Boe & Hannaford, 1998) to optimize vehicle speed

(i.e., reduce trial time) while simultaneously correcting steering inputs. Most importantly, this paper presents rigorous industrial-scale field results that demonstrate the application of ILC+FBL for wheeled-vehicle path following and also shows a method for conducting learning using pre-trial simulations of ILC+FBL to compute correction factors passed from a simulator to a real vehicle.

1.2 About this Paper

Section 2 briefly outlines the underlying FBL-based path-following controller that is augmented by ILC. Section 3 then presents the controller design and implementation process for three ILC methods, including a base iterative learning augmentation to path following (ILC+FBL), a parallel iterative speed learning controller (ISLC+FBL) and a method to conduct iterative learning trials in a simulation environment prior to any real vehicle path-following runs. Finally, Section 4 presents the results of extensive field trials in an underground mine by using two center-articulated underground vehicles.

2 Path Following Control Background

In this work, ILC+FBL augments a baseline path-following controller with an ILC block. For completeness, this section briefly describes the underlying FBL-based

controller, which is similar to the method outlined by Marshall et al. (2008).

2.1 Vehicle Kinematics

In our experiments (see Section 4) we apply ILC+FBL to an underground mining vehicle called a load-haul-dump (LHD) machine, which is a industrial-scale, hydraulically-steered, center-articulated vehicle used for excavating and moving raw materials. Figure 1 shows photos of the two vehicles used in our experiments (14-tonne and 18-tonne payload machines). These vehicles differ in size, and thus dynamics, but share similar center-articulated kinematics, allowing for the same controller design for each vehicle. We reasonably contend that the concepts presented here are also applicable to other types of wheeled vehicles.

Referring to Figure 2(a), the system configuration is $\mathbf{q} = (x, y, \theta, \phi)$, where $(x, y) \in \mathbb{R}^2$ is the position of the front component F , $\theta \in \mathbb{S}^1$ is the heading angle of the front component, and $\phi \in \mathbb{S}^1$ is the steering (articulation) angle. The forward vehicle speed and the steering rate at F are $v \in \mathbb{R}$, and $\omega \in \mathbb{R}$, respectively (i.e.,

Figure 2: LHD vehicle configuration and error definitions for path following.

$\omega(t) \equiv \dot{\phi}(t) := \frac{d}{dt}\phi(t)$). Thus, a kinematic (nonholonomic) model of the vehicle is

$$\dot{\mathbf{q}} = \begin{bmatrix} \cos \theta & 0 \\ \sin \theta & 0 \\ \frac{-\sin \phi}{\ell(\phi)} & \frac{-\ell_R}{\ell(\phi)} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix}, \quad (1)$$

where, for simplicity, we write $\ell(\phi) := \ell_R + \ell_F \cos \phi$. Positive values of v and ω correspond to forward movement and steering to the right, respectively.

2.2 Baseline Controller

Following Marshall et al. (2008) and others, to achieve path following we define two errors as depicted in Figure 2(b). The lateral error ε_L is the distance from F to

the tangent to the desired path at the closest discrete point along the path and the heading error ε_H is the difference between the angle (in the configuration space) of this instantaneous tangent and the vehicle heading θ . In our experiments, the error states $(\varepsilon_L, \varepsilon_H)$ were repeatedly computed (estimated) by a UKF-based underground navigation system described in detail by Marshall et al. (2008).

Without loss of generality, it is assumed the tangent to the desired path instantaneously corresponds with the x -axis shown in Figure 2(a). Thus, $\varepsilon_L \equiv y$ and $\varepsilon_H \equiv \theta$ and the error dynamics can be written, using (1), as

$$\dot{\varepsilon}_L = v \sin \varepsilon_H \quad (2a)$$

$$\dot{\varepsilon}_H = \frac{-v \sin \phi}{\ell(\phi)} - \frac{\ell_R}{\ell(\phi)} \omega. \quad (2b)$$

Then, by well-known techniques for feedback linearization of the defined error states (Isidori, 1995), consider the nonlinear change of coordinates

$$z_1 := \varepsilon_L \quad (3a)$$

$$z_2 := v \sin \varepsilon_H. \quad (3b)$$

Thus, in matrix form, the error dynamics (2) are a second-order linear system

$$\begin{bmatrix} \dot{z}_1 \\ \dot{z}_2 \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}}_{\mathbf{A}} \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} + \underbrace{\begin{bmatrix} 0 \\ 1 \end{bmatrix}}_{\mathbf{B}} \eta, \quad (4)$$

where $\eta \in \mathbb{R}$ is the transformed control input in the feedback linearized space is

$$\eta = \frac{-v^2 \sin \phi \cos \varepsilon_H}{\ell(\phi)} - \frac{\ell_R v \cos \varepsilon_H}{\ell(\phi)} \omega. \quad (5)$$

Therefore, to stabilize the second-order error system (4), a simple PD-type controller is applied, where the transformed control input is of the form $\eta = \mathbf{K}\mathbf{z}$, where $\mathbf{K} = [k_P \ k_D]$ and k_P and k_D are proportional and derivative gains, respectively, and for asymptotic stability we must have $k_P, k_D < 0$. Finally, the actual vehicle steering input ω is back-computed by inverting the transformed input (5) to obtain

$$\omega = \frac{-v \sin \phi}{\ell_R} - \frac{\ell(\phi)}{\ell_R v \cos \varepsilon_H} \eta. \quad (6)$$

Note that, from (6), the heading error is limited to the open set $\varepsilon_H \in (-\frac{\pi}{2}, \frac{\pi}{2})$.

It is worth mentioning that other control methodologies for path following could also be used in conjunction with ILC+FBL. For example, one could choose to augment a model-predictive control (MPC) scheme (e.g., Klančar & Škrjanc, 2007; Ostafew et al., 2016) rather than the baseline controller described here.

3 ILC+FBL Designs

This section outlines three versions of ILC+FBL, where we first develop the base ILC control algorithm and verify its stability. Following from this, a second par-

allel iterative speed learning control (ISLC) law is described, which adjusts the desired forward vehicle speed along the path. Finally, ILC+FBL is applied in a pre-trial simulation environment before sending corrected control inputs to the vehicle. Validation by experimental field trials for these three methods is then provided in Section 4.

3.1 Base ILC Setup

In our implementation of ILC+FBL, a discrete-time ILC formulation is added to augment the FBL path-following controller by applying additive corrections to the control input (in this case, the steering rate). These corrections are applied for a specific desired path (route) and learning is conducted for each new desired route. The corrections are computed offline following each run based on the performance of the previous trial. We perform ILC in the feedback-linearized space defined by the coordinates transformation (3), as seen in Figure 3. This is slightly different from others who compute ω directly as the feed-forward signal (e.g., Ostafew et al., 2013). In our case, the learned corrections are applied to the transformed input (5) before performing a nonlinear transformation back to the steering input ω . Thus, we keep the single back-computation (6) unchanged over

Figure 3: Block diagram of ILC+FBL algorithm with equation references to the controller development outlined in Sections 2 and 3. Note that the dashed arrows to the ILC block imply learning occurs offline, after each iteration.

time; i.e.,

$$\omega = \frac{-v \sin \phi}{\ell_R} - \frac{\ell(\phi)}{\ell_R v \cos \varepsilon_H} (\eta + \tilde{\eta}), \quad (7)$$

where $\tilde{\eta}$ is the computed additive correction.

3.1.1 Discrete Spatially-Indexed ILC

ILC is normally implemented by indexing over discrete time steps (Bristow et al., 2006) such that the system state is given in discrete-time as $\mathbf{q}(k) := \mathbf{q}(t = kT), k = 0, 1, 2, \dots$ (e.g., see Figure 4), where T is the time step. However, in the path-following case, we cannot define errors as a function of time because the ve-

hicle will surely have a different pose at the same time for every subsequent trial. Thus, we define i as the time index when the i -th desired path point is reached and spatially index our system at these path points in the x - y plane, similar to the approach of (Ostafew et al., 2013). It is also worth mentioning that this formulation is similar to—but not quite the same as—the point-to-point ILC problem, which seeks to track a reference signal at only a subset of reference trajectory points; e.g., see Chu, Freeman, and Owens (2015); Freeman, Cai, Rogers, and Lewin (2011) for recent and comprehensive treatments.

The system state is then given in discrete time by $\mathbf{q}(i) := \mathbf{q}(t = i\tilde{T}), i = 0, 1, 2, \dots$, where we define $\tilde{T} = nT$. Here we let n be the number of discrete time steps k between each path point, which may be variable. If $0 < T \ll 1$ and n is small enough, then $0 < \tilde{T} \ll 1$ which is reasonable in our case, assuming that learning occurs at typical tramming speeds (i.e., v is not very low or null).

Recalling (4), the discrete linearized error system becomes

$$\begin{bmatrix} z_1(i+1) \\ z_2(i+1) \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & \tilde{T} \\ 0 & 1 \end{bmatrix}}_{\mathbf{F}} \begin{bmatrix} z_1(i) \\ z_2(i) \end{bmatrix} + \underbrace{\begin{bmatrix} 0 \\ \tilde{T} \end{bmatrix}}_{\mathbf{G}} \eta(i) \quad (8a)$$

$$\psi(i) = \underbrace{\begin{bmatrix} 1 & 0 \end{bmatrix}}_{\mathbf{H}} \mathbf{z}(i), \quad (8b)$$

where ψ is the error (output) over which we have chosen to learn (in this case, we

Figure 4: Desired and driven paths showing closest points and path indexing.

have selected the lateral path-following error ε_L).

This formulation ignores vehicle states at discrete steps between path points. However, we contend that the assumption that n is small enough such that $0 < \tilde{T} \ll 1$ is sufficient for the purposes of controller design (since it plays no part in the final design). A more complete treatment is described by Ostafew et al. (2013), although the result is the same.

3.1.2 Error System Model

The goal of ILC+FBL is to learn the feed-forward control inputs $\tilde{\eta}(i)$ over trial iterations indexed by $j = 0, 1, 2, \dots, J - 1$. We apply this additive correction to the real-time control inputs. Thus, employing $\eta = \mathbf{Kz}$ and (8), we use the error model

$$\mathbf{z}_j(i+1) = \mathbf{F}\mathbf{z}_j(i) + \mathbf{G}(\eta_j(i) + \tilde{\eta}_j(i)) \quad (9a)$$

$$= \underbrace{(\mathbf{F} + \mathbf{G}\mathbf{K})}_{\tilde{\mathbf{F}}} \mathbf{z}_j(i) + \mathbf{G}\tilde{\eta}_j(i). \quad (9b)$$

Now, we assume the initial condition $\mathbf{z}_j(0) = \mathbf{z}_0$ for all j and let q be the forward discrete time-shift operator, such that $q\mathbf{z}(i) = \mathbf{z}(i+1)$. Thus, by Z-transforms and by using (8) and (9), the system is written as

$$\psi_j(i) = \underbrace{\mathbf{H}(q\mathbf{I} - \tilde{\mathbf{F}})^{-1}\mathbf{G}}_{P(q)} \tilde{\eta}_j(i) + \underbrace{\mathbf{H}\tilde{\mathbf{F}}^i}_{d(i)} \mathbf{z}_0. \quad (10)$$

Because this system is a second-order state model, we have a relative degree of $m = 2$ and therefore, following the formulation in Bristow et al. (2006), for an N -sample system we have that

$$\tilde{\eta}_j(i), i = 0, 1, 2, \dots, N-1 \quad (11a)$$

$$\psi_j(i), i = 2, 3, 4, \dots, N+1. \quad (11b)$$

Assuming a desired system output $\psi_d \equiv 0$ for path following, we write the performance error signal as

$$e_j(i) = \psi_d(i) - \psi_j(i) \quad (12a)$$

$$= -\psi_j(i), i = 2, 3, 4, \dots, N+1. \quad (12b)$$

3.1.3 Lifted Form

The error system described in Section 3.1.2 can be written in the *lifted form* (Bristow et al., 2006)

$$\mathbf{y}_j = \mathbf{P}\tilde{\mathbf{h}}_j + \mathbf{d} \quad (13a)$$

$$\mathbf{e}_j = -\mathbf{y}_j, \quad (13b)$$

where, for each learning iteration j ,

$$\mathbf{y}_j = \begin{bmatrix} \psi_j(2) \\ \psi_j(3) \\ \vdots \\ \psi_j(N+1) \end{bmatrix}, \quad \tilde{\mathbf{h}}_j = \begin{bmatrix} \tilde{\eta}_j(0) \\ \tilde{\eta}_j(1) \\ \vdots \\ \tilde{\eta}_j(N-1) \end{bmatrix}, \quad (14a)$$

$$\mathbf{P} = \begin{bmatrix} p_2 & 0 & \dots & 0 \\ p_3 & p_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ p_{N+1} & p_N & \dots & p_2 \end{bmatrix}, \quad \mathbf{d} = \begin{bmatrix} d(2) \\ d(3) \\ \vdots \\ d(N+1) \end{bmatrix}, \quad (14b)$$

and the entries of $\mathbf{P}$ are $p_i = \mathbf{H}\tilde{\mathbf{F}}^{i-1}\mathbf{G}$. Note here that because ILC is employed within the feedback linearized space, approximate linearization is not necessary at each discrete step and thus it is also computationally cheap since the system model $\mathbf{P}$ is time-independent.

3.1.4 Anticipatory Learning Algorithm

Consider the commonly-used ILC law (Bristow et al., 2006)

$$\tilde{\mathbf{h}}_{j+1} = \mathbf{Q}(\tilde{\mathbf{h}}_j + \mathbf{L}\mathbf{e}_j), \quad (15)$$

which is a simple proportional law, where $\mathbf{Q}$ is referred to as the *Q-filter* and $\mathbf{L}$ is the *learning function*. The *Q-filter* is often referred to as a “forgetting factor” because it is used to discount non-repeated disturbances which increases robustness. Alternatively, the learning function can be thought of as a gain determining how aggressively the controller will alter the control input. This proportional law is causal and thus provides little learning benefit as it lacks the ability to “look ahead” along the path (Bristow et al., 2006).

To introduce anticipatory behavior, a phase lead ILC controller, as used by D. Wang and Ye (2005), is employed. This follows from the proportional-type algorithm and has the lifted form

$$\tilde{\mathbf{h}}_{j+1} = \mathbf{Q}(\tilde{\mathbf{h}}_j + \tilde{\mathbf{L}}\mathbf{e}_j), \quad (16)$$

where $\tilde{\mathbf{L}}$ is the phase lead learning function expressed as

$$\tilde{\mathbf{L}} = \begin{bmatrix} 0 & \dots & 0 & k_p & 0 & \dots & 0 \\ \vdots & \ddots & & \ddots & k_p & \ddots & \vdots \\ & & & & & \ddots & 0 \\ & & & & & \ddots & k_p \\ & & & & & & 0 \\ \vdots & & & & & \ddots & \vdots \\ 0 & \dots & & & \dots & & 0 \end{bmatrix}. \quad (17)$$

Here k_p is a proportional gain and $\tilde{\mathbf{L}}$ is non-causal (non-zero entries appear above the main matrix diagonal). For one discrete step the learning law (16) is

$$\tilde{\eta}_{j+1}(i) = k_q(\tilde{\eta}_j(i) + k_p e_j(i + u)), \quad (18)$$

where the positive integer $u \geq 1$ is the phase lead, representing the number of discrete steps to look ahead on the error signal from the previous iteration, and k_q and k_p are proportional gains. Thus, for the next iteration $j + 1$, the input at the same discrete step i is a function of the previous input and the error signal from u path points ahead in the previous iteration j .

Stability for this control law can be shown by examining the error dynamics. By combining the lifted form system (13) and the learning algorithm (16), the error

propagation can be expressed as

$$\mathbf{e}_{j+1} = \underbrace{\mathbf{Q}(\mathbf{I} - \mathbf{P}\tilde{\mathbf{L}})}_W \mathbf{e}_j + (\mathbf{Q} - \mathbf{I})\mathbf{d}. \quad (19)$$

Thus, for stability one must choose the eigenvalues of W such that the spectral radius $\rho(W) < 1$ by an appropriate choice of $\tilde{\mathbf{L}}$.

3.2 Iterative Speed Learning Control

The objective of the ILC+FBL technique described in Section 3.1 is to reduce the path-following errors. However, along with reduced errors there may be the potential to improve (i.e., usually increase) the vehicle speed if some path-following error can be tolerated. This is particularly useful in applications where productivity is of interest (e.g., hauling more material per unit time with mining or construction vehicles). In this section we formulate an iterative speed learning controller (ISLC) that attempts to simultaneously (i.e., in parallel to error minimization) improve the vehicle speed along the desired path. This algorithm starts with the base ILC algorithm and adds a second law of the form

$$\tilde{\mathbf{v}}_{j+1} = \mathbf{Q}_s(\tilde{\mathbf{v}}_j + \mathbf{L}_s \mathbf{e}_{s,j}), \quad (20)$$

where $\tilde{\mathbf{v}}$ is the vector of desired forward vehicle speeds to be altered, $\mathbf{Q}_s$ is a speed learning Q-filter, $\mathbf{L}_s$ is a speed learning function (of the same non-causal phase

lead form in Section 3.1.4), and $\mathbf{e}_{s,j}$ is the defined speed learning error.

In this case, we apply learned alterations directly to the desired forward vehicle speed as opposed to applying correction factors because there is no base controller prescribing an initial vehicle speed to be corrected. Instead, a constant $\bar{v}$ is chosen for the entire path and, from this, changes are made along the desired path to form a variable speed profile. From preliminary ILC trials and based on the work by Boe and Hannaford (1998), there is an intuitive relationship between forward vehicle speed, v , and path-following performance; namely, as v increases, the performance decreases (i.e., ε_L increases). Thus, the error is defined as

$$\mathbf{e}_{s,j} = \mathbf{e}_t - |\mathbf{y}_j|, \quad (21)$$

where $\mathbf{e}_t$ is a tunable error threshold and $\mathbf{y}_j$ is the error system output. In this case the algorithm drives $\mathbf{e}_{s,j} \rightarrow 0$ as $j \rightarrow \infty$. However, in contrast to minimizing the path-following error, by defining the error as (21) the desired vehicle speed, $\bar{v}$ is increased when errors are low (and vice versa) until the error threshold $\mathbf{e}_t$ converges. In this way, an “allowable” error is introduced in order to ramp up the forward vehicle speed when it is considered safe (i.e., low path-following error ε_L) (Boe & Hannaford, 1998). Thus, the algorithm for one discrete step is

$$\bar{v}_{j+1}(i) = k_{q_s}(\bar{v}_j(i) + k_{p_s}e_{s,j}(i + u)), \quad (22)$$

where

$$e_{s,j} = e_t - |\psi_j(i)|. \quad (23)$$

This new parallel combination of iterative learning control laws is combined with the base FBL algorithm (thus, ISLC+FBL) to correct the steering inputs for improving path-following performance while simultaneously increasing the desired vehicle speed (at the expense of allowing some increase in path-following errors). Thus, a performance balance between these two opposing objectives is necessary and can be achieved through appropriate tuning of the learning functions $\mathbf{L}$ and $\mathbf{L}_s$ and Q-filters $\mathbf{Q}$, and $\mathbf{Q}_s$ for ILC and ISLC, respectively. This is demonstrated in practice by the field trial results in Section 4.4.3.

3.3 Pre-Trial Simulation Learning

One disadvantage to relying on ILC in the field to reduce the path-following errors is that this takes time—in our experiments, at least a few iterations. If the initial path-following errors at trial $j = 1$ are too large (e.g., unsafe or may cause damage to the vehicle and/or its environment) then one option is to conduct pre-trial simulation learning. Although simulation will not provide the rich experiences obtainable from actual trials in the target environment, replaying taught routes (as per Marshall et al., 2008) may serve to partially incorporate some aspects of

the environment and geometry of the path to be followed through ILC, thus reducing the initial errors prior to learning on a real vehicle. Others have also recognized this deficiency and have similarly suggested the use of predictive trials to accelerate convergence; e.g., see (Chu, Owens, & Freeman, 2016).

Using the taught route, we implemented a computationally cheap kinematic simulation of the ILC+FBL algorithm for the Epiroc ST14 Scooptram in MATLAB, which we then integrated with Epiroc’s offline “route manager” application (a MATLAB-based application). This simulation discretely propagates the vehicle pose forward based on inputs (v, ω) passed to the system model given by (1). By doing this, some of the steering rate corrections provided by the ILC algorithm can be pre-computed offline. Following simulation, the pre-learned array of correction factors can be passed to the real vehicle where “fine tuning” can be achieved by running the ILC+FBL for additional trials on the target platform and in the target environment. This technique is demonstrated in practice through the field experiments described in Section 4.3.

4 Underground Field Experiments

This section describes extensive field testing of the ILC-based path-following algorithms described in Section 3 on an industrial-scale autonomous vehicle. We

begin by providing some background about the experimental setup, including descriptions of the method for “teaching” routes (see Section 4.1 and Marshall et al., 2008), baseline controller gain selection (Section 4.2), and preliminary simulations (Section 4.3) prior to a presentation of the field trial results (Section 4.4).

4.1 Route Profiling

The prescribed desired path—what we refer to as a *route profile*—was initially taught by manually tele-operating each vehicle along a route. Although this taught path is collision-free and kinematically feasible, it is not guaranteed to be optimal in any specific sense, as compared with with one that might have been generated using a path planning algorithm. For this paper, the chosen route consisted of two rounded 90-degree corners in order to present the greatest challenge to the ILC+FBL algorithm, testing its robustness, as well as providing an accurate representation of a possible path required by an operating mine. This route is depicted in the LiDAR-generated occupancy grid map output by the localization system (Marshall et al., 2008) shown in Figure 5. For consistency, this two-turn route was chosen to be the route in all conducted learning trials, both simulated and for underground trials (ILC, ISLC, and Pre-trial Simulation ILC) and, although not identical for both the ST14 and ST18, the routes for each vehicle followed the

Figure 5: 2D occupancy grid map generated by the localization system (a LiDAR-IMU-odometry fusion) showing the desired route (taught manually) through the underground space as well as the start and end points (bottom and top of the image, respectively).

same two-turn structure.

4.2 Controller Gain and Parameter Selection

The baseline controller gains (see Section 2.2) were selected in order to maintain separation between the inner- and outer-loop control bandwidths (ω_I and ω_O , respectively) in our two-timescale path-following control architecture. Details on the selection of k_P and k_D for the baseline controller are also outlined by Marshall et al. (2008) but briefly, k_P and k_D were selected as

$$k_P = -\omega_O^2 \quad (24a)$$

$$k_D = -2\zeta\omega_O \quad (24b)$$

where $k_P, k_D < 0$ and ζ is a damping ratio which, in the path-following case, is selected to achieve an over-damped transient response ($\zeta \geq 1$).

For ILC+FBL and ISLC+FBL, the Q-filter and learning function gains, k_q and k_p , respectively, were selected to ensure stability of (19), as per Section 3.1.4. In this case, initial tuning was validated on a simulated model (written in the Julia programming language—see more in Section 4.3) of a center-articulated vehicle with modeled steering dynamics before applying the controller to the actual vehicles. For ILC, Q-filter tuning is usually conducted by initializing k_q as unity and slowly decreasing until a desired performance robustness condition is achieved

(Bristow et al., 2006). Here, no Q-filtering was required for the ILC algorithm and thus, the filter was set to unity. The ISLC algorithm, however, showed improved response with the use of some Q-filtering. The learning gains for each algorithm were selected to achieve stable yet aggressive learning behavior (i.e., convergence after minimal learning iterations). Table 1 summarizes the values chosen for each gain for the real-vehicle experiments.

The phase lead value $u \in \mathbb{Z}^+$ from (18) was selected to provide anticipatory behavior for the ILC algorithms and should account for the forward vehicle speed (within the typical LHD speed range of 0 to 5 m/s) at a given location along the desired path. By preliminary simulation, a simple linear relationship did not provide the necessary anticipatory behaviour at the higher end of the tested speeds thus, a simple exponential relationship was devised to compute this phase lead as a function of the vehicle forward speed v and some tunable parameters

$$u = \lceil mv^a + b \rceil, \quad (25)$$

where, for this case, the selected parameters are given in Table 1. The value of u is discrete because the path-following errors are computed relative to the closest path point and are indexed at this discrete point. This phase lead function allows for a minimum phase lead and an exponential relationship between speed and the desired phase lead, thus allowing the vehicle to look further ahead when travel-

Table 1: Gain and parameter choices for ILC and ISLC implementations.

Gain or parameter	Description	Value
k_p	Learning gain (ILC)	0.40
k_{p_s}	Learning gain (ISLC)	0.85
k_q	Q-filter (ILC)	1.00
k_{q_s}	Q-filter (ISLC)	0.98
m	Phase lead parameter	2.0
a	Phase lead parameter	1.4
b	Phase lead parameter	3.0
e_t	Lateral error threshold (ISLC)	0.2 m

ling faster. Finally, the error threshold value was selected to be $e_t = 0.2$ m because this level of path-following error was deemed tolerable for such a large-scale vehicle. Additionally, this threshold provided stable increases in forward vehicle speed in the simulation environment prior to testing on the real vehicles.

4.3 Pre-Experiment Simulation Tests

Initial testing was conducted in a simulation environment in order to ensure controller behavior and conduct initial gain tuning for ILC+FBL. This simulation tool—created by using the Julia 0.4.5 language—employed a model of the Epiroc ST14 vehicle, including the vehicle kinematic model, as given in (1), as well as some basic steering dynamics for increased model accuracy, and inner-loop controllers to stabilize the rate inputs v and ω given their desired values (see Section 4.3 in Marshall et al., 2008). This section briefly covers validation of the simulated ST14 model and provides some preliminary results showing a simulated ILC+FBL implementation, used in advance of field testing.

4.3.1 Vehicle Model Validation

The developed simulator discretely propagates the vehicle position and orientation in the configuration space based on the rate inputs v and ω to the kinematic model (1). Furthermore, basic steering dynamics were included to model the behavior of the steering cylinders and the effect of the vehicle mass on the steering rate, given a desired rate ω_d prescribed by the path-following controller.

In order to compare the model against the actual ST14 vehicle, a steering input was generated by manually driving the actual vehicle at a constant forward vehi-

Figure 6: Plot of the modeled and actual steering rate ω output resulting from an identical steering command passed to the simulated and real ST14 vehicle.

cle speed ($v = 4.0$ m/s) on an arbitrary route underground and recording relevant data (steering input and steering rate). This recorded input was then passed to the Julia-based simulator in order to generate a steering rate output and compare the results. Figure 6 shows the steering rate of the simulated (model) and real ST14 vehicle given the recorded steering rate.

It is clear from Figure 6 that the devised model provides a similar output to the real vehicle. The error between the two steering rate signals remains bounded be-

tween approximately ± 10 deg/s. Hence, the parameters used for this simulation were deemed to be a suitable starting point for field testing on a real vehicle.

4.3.2 Simulated ILC+FBL Trials

The ILC+FBL algorithm was first tested by using the simulation model of the Epiroc ST14 loader, which was validated in Section 4.3.1. The simulation was conducted for $J = 10$ learning iterations while path following at a constant desired forward speed of $\bar{v} = 4.0$ m/s on the desired route recorded using an actual vehicle (see Section 4.1). The results of these simulation trials can be seen in Figure 7, which shows the lateral and heading path-following errors along the desired path as well as the maximum and RMS values of these errors for each trial. It is clear that after only $J = 10$ learning iterations, both the lateral and heading errors converged to low values, which is evident from the flat section of the maximum and RMS plots for the last four learning iterations.

4.4 Underground Field Experiments

This section provides a detailed overview of the test equipment (Epiroc ST14 and ST18 autonomous vehicles) and presents experimental field trial data from tests with the ILC, ISLC, and pre-trial simulation learning control algorithms.

(a) Path-following errors along the desired path.

(b) Maximum and RMS path-following errors per learning iteration.

Figure 7: Lateral and heading path-following errors, (ε_L and ε_H , respectively) for 35 $J = 10$ learning iterations at a forward vehicle speed of $v = 4.0$ m/s of the simulated ST14 autonomous vehicle.

4.4.1 Test Platform and Environment

Two vehicles were used for underground field trials to test the performance and robustness of the ILC+FBL and ISLC+FBL algorithms. These included the Epiroc ST14 and ST18 LHD machines, which are diesel-driven and hydraulically-steered at a centre articulation joint (i.e., vehicle frames rotate while wheels stay fixed), shown in Figure 2(a). Technical specifications outlining the size, capacity, and power of these machines are provided in Table 2. These vehicles are equipped with a proprietary CAN-based distributed control system with GUI module and x86-based computing module and five multi-I/O modules distributed on the machine to handle operator/control inputs such as buttons, joysticks, pedals, as well as sensors and outputs (e.g., for temperatures, positions, and hydraulic valve controls). The algorithm code was written using C++ and flashed to this computing module for implementation. For control and estimation, a driveshaft-mounted encoder was used to measure speed (96 pulses per revolution) and the articulation angle was measured by using a 16-bit absolute encoder. The vehicle yaw rate was measured using one of the gyros of an AIMS IMU mounted on the front component of the LHD's body. Finally, two SICK LMS511 laser rangefinders (shown in Figure 1), one rear and one forward-facing, sampled at 75 Hz, were used for mapping and real-time localization purposes.

Table 2: Technical specifications of the Epiroc ST14 and ST18 LHD vehicles.

Vehicle Specification	ST14	ST18
Tramming capacity	14 000 kg	17 500 kg
Approximate weight (empty)	39 000 kg	51 500 kg
Engine	Cummins QSM11	Cummins QSX15
Engine power	250 kW	336 kW
Transmission	Dana T40000 (4 gears)	Dana TE32 (4 gears)
Top speed (zero grade, empty)	8.25 m/s	7.50 m/s
Length	10.825 m	11.227 m
Width (standard bucket)	2.800 m	3.190 m
Height (bucket down)	2.550 m	2.854 m

This research focused on the development of a path-following algorithm based on ILC+FBL, which we compared with the existing FBL-only path-following controller (without look-ahead “patches”) outlined in Section 4.2 of Marshall et al. (2008). Although this previous work pertained to a different vehicle model, the same localization code was ported to the newer ST14 and ST18 vehicles that were used in the research described by this paper. Path-following performance was evaluated based on the output of the on-board UKF-based localization system that uses occupancy grid-based metric maps of the underground environment created during a teaching phase (i.e., where a human operator first drives the route to be re-played by the autonomous vehicle). Details about the teaching—also called route profiling—and metric map-based localization and navigation algorithms are described in Sections 3 and 4, respectively, of Marshall et al. (2008).

All underground field trials were conducted at Epiroc’s Kvarntorp underground test mine, near Örebro, Sweden. This is a controlled underground environment with light barriers to cordon off the automation area and provide a safe operating zone for the autonomous vehicles. Supervisory control and monitoring was conducted from an enclosed room in the automation area that contained an Epiroc Operator Station (OPS) required to interface with the autonomous vehicles via an installed wireless network. The drifts (a.k.a., tunnels) in the test mine were ini-

tially established as part of an ore producing mine and therefore provide a realistic representation of an operating mine.

4.4.2 ILC+FBL Trials

Both the ST14 and ST18 vehicles were used to test the robustness of the ILC+FBL algorithm, which provided two vehicle platforms with differing masses and dimensions (see Table 2). Using two parametrically differing platforms implies that the algorithm should also apply to dynamically changing parameters on one vehicle (i.e., adding mass in the bucket when hauling). All trials involved running the ILC+FBL for $J = 10$ learning iterations on one previously taught route (see Section 4.1). Constant forward vehicle speeds were prescribed for the individual trials; these were: 2.0, 3.0, and 4.0 m/s. In this paper, only the results for the 4.0 m/s trial are presented because the same learning behavior was present in the results at all forward vehicle speeds.

Figures 8 and 9 show the path-following performance of the ILC+FBL algorithm running on the ST14 and ST18 vehicles, respectively, for the highest tested forward vehicle speed of $v = 4.0$ m/s. Recall that this forward speed was prescribed as constant for the entire desired path and is maintained even in the 90-degree corners. The top plots, Figures 8(a) and 9(a), demonstrate the path-

following errors along the length d of the desired path. For the first iteration, which represented a trial conducted solely by applying FBL (i.e., no ILC corrections applied), as expected there were spikes in both lateral and heading errors upon reaching the 90-degree corner sections of the desired route (at approximately 40 m and 100 m along the desired test path).

Following $J = 10$ learning iterations using ILC+FBL, however, there is a large decrease in the lateral and heading errors for both the ST14 and ST18 vehicles. The bottom plots, Figures 8(b) and 9(b), show the maximum and root mean squared (RMS) errors calculated from the error signals for each learning iteration. The obvious downward trend in these plots demonstrates the ability of the ILC+FBL algorithm to apply corrections and iteratively reduce the errors along the desired path until convergence at a comparatively low level. It is important to note that, because the RMS errors follow the same downward trend, the maximum path-following errors (i.e., in the corners) are not improved by ILC+FBL at the expense of introducing errors elsewhere along the desired path. Furthermore, note that our ILC error system model in (10) contains only the lateral error (i.e., $\psi = z_1 = \varepsilon_L$) yet we see learning benefit applied to both lateral and heading tracking errors. This can be explained by the inherent relationship between ε_L and ε_H shown by the error dynamics of the system in (2). To achieve a repeatable starting position, the

Table 3: Percent reduction in maximum and RMS errors for $J = 10$ learning iterations by using ILC+FBL on both ST14 and ST18 autonomous vehicles.

Error Type	ST14	ST18
Lateral error (ε_L)		
Maximum	93.9%	89.2%
RMS	96.7%	92.9%
Heading error (ε_H)		
Maximum	79.5%	65.4%
RMS	82.2%	68.1%

base FBL path following controller was used to bring the vehicle back to the initial position following each learning iteration. The error spike at the beginning of the 10th learning iteration is likely the result of a change in ground conditions at the start point and does not change the learning results in the rest of the trial. Table 3 shows calculated percent reductions in the maximum and RMS path-following errors for the ST14 and ST18 vehicles.

(a) Path-following errors along desired path.

(b) Maximum and RMS path-following errors per learning iteration.

Figure 8: Lateral and heading path-following errors (ε_L and ε_H , respectively) for
42

$J = 10$ learning iterations at a forward vehicle speed of $v = 4.0$ m/s for the ST14.

(a) Path-following errors along desired path.

(b) Maximum and RMS path-following errors per learning iteration.

Figure 9: Lateral and heading path-following errors (ε_L and ε_H , respectively) for
43
 $J = 10$ learning iterations at a forward vehicle speed of $v = 4.0$ m/s for the ST18.

4.4.3 ISLC+FBL Trials

The Epiroc ST18 vehicle was used to test the performance of the ISLC+FBL path-following algorithm (see Section 3.2), which updates corrections to the steering input computed by the FBL controller while simultaneously altering the desired speed profile along the path in order to allow the vehicle to drive at increased speeds based on the previous iteration's tracking performance (in this case, the lateral error ε_L). To test ISLC+FBL, the algorithm was run for $J = 20$ learning iterations at a starting speed of $\bar{v} = 2.0$ m/s.

Figure 10 shows the path-following performance achieved by the ISLC+FBL algorithm. Here, similar to the ILC+FBL results in Section 4.4.2, we observed large initial spikes in lateral and heading errors over the corners of the desired path (at approximately 40 m and 100 m along the desired path). However, these errors were significantly lower at 2.0 m/s forward speed compared with at 4.0 m/s; $\varepsilon_L \approx 0.7$ m compared to $\varepsilon_L \approx 2.0$ m at $v = 4.0$ m/s. This confirms the assertion that the path-following performance of FBL is inversely related to the forward vehicle speed. After only five learning iterations, however, the path-following errors decreased significantly, to less than 0.2 m of lateral error and to within an approximately 5 degree error bound on heading error. Note the simultaneous increase in vehicle speed (bottom plot of Figure 10) because, by five learning iterations,

inflection points appeared corresponding to the straight portions of the desired route (i.e., where errors were low). The error signals after 10 and 20 learning iterations showed similar error outputs and remained approximately bounded below $\varepsilon_L < 0.3$ m, which was expected because the error threshold e_t in the ISLC algorithm (23) was set to 0.2 m. Furthermore, the forward vehicle speed continued to increase with additional learning iterations and a smooth speed profile began to form after 10 and 20 learning iterations, thus demonstrating optimized “slow downs” in the 90-degree corners and higher speeds (up to $v \approx 4.5 - 5.0$ m/s) in the straight sections. After 20 learning iterations, the learned speed profile shows speeds greater than 4.0 m/s in the straight sections but never reaches more than 4.0 m/s in the corners (approximately 35 and 100 m). The fixed speed ILC trials show errors of a similar magnitude at 4.0 m/s in these corner regions, but only after 10 learning iterations at that speed. In this ISLC case, the errors are continuously maintained approximately at the threshold value and thus, a lower cornering speed is computed.

Figure 11 demonstrates that the ISLC+FBL algorithm successfully increases forward vehicle speeds while maintaining an allowed level of path-following performance. The top two plots of maximum and RMS heading and lateral error show the initial learning process, converging to a relatively stable value where we

observed the maximum lateral error to fluctuate around 0.2 m and the maximum heading error at around 5 degrees for all $J = 20$ learning iterations. At the same time, the forward vehicle speeds increased, as expected, demonstrating a balance between speed and allowed path-following errors. The maximum lateral error for the last learning iteration $j = 20$ is greater than at iteration $j = 10$ due to this fluctuation (resulting from speed increases) and would be accounted for again if subsequent learning iterations were conducted. However, once a desired speed profile is achieved the ISLC algorithm can be ceased. Again, we note the convergence of the RMS errors and maximum errors. The bottom plot shows the total iteration time for each learning trial and demonstrates a the time-performance improvement as a result of the increases in forward vehicle speed. From a constant speed of $\bar{v} = 2.0$ m/s, the ISLC+FBL algorithm provides a 40.7% reduction in trial time, which—in the case of application in an operating mine environment—could yield significant production benefits.

4.4.4 Pre-Trial Simulation ILC+FBL

To test the potential benefit of conducting pre-trial simulation learning in a simulated environment (see Section 3.3) we ran field trials of ILC+FBL on the Epiroc ST14 for $J = 6$ learning iterations on the route from Section 4.1 at a forward vehicle

Figure 10: Path-following performance at distances along the desired path for $J = 20$ learning iterations with the ST18 running ISLC+FBL from an initial constant speed of $\bar{v} = 2.0$ m/s.

Figure 11: Maximum and RMS path-following errors per trial and total trial time for $J = 20$ learning iterations with ST18 while learning speed profile.

speed of $v = 4.0$ m/s. These baseline real-world learning trials were essentially a repetition of the results presented in Section 4.4.2 and are referred to as the *Vehicle Learning* (VL) dataset. From Figure 12, we see the same error response (reduction) from ILC+FBL as before, where initially, higher errors appeared in the corners of the route, which reduced to low errors ($\epsilon_L \approx 0.25$ m) after approximately $J = 6$ learning iterations of ILC+FBL. The same number of iterations were conducted in the simulated environment and the error outputs were recorded for each trial. This data is referred to as the *Simulation Learning* (SL) dataset. The maximum and RMS errors for these trials are shown in Figure 12(b). It is clear that the performance of ILC+FBL for SL matches closely the performance of VL in terms of maximum and RMS errors per learning iteration because the plots show very similar learning convergence curves.

Finally, following each learning iteration conducted in the simulated environment, the computed array of correction factors $\tilde{\eta}$ was stored and passed to the real vehicle and the errors recorded for this run. This dataset is denoted the *Vehicle Response from Simulated Learning* (V-SL) dataset. In this case, each time the learned correction factors were passed to the real vehicle, there was no prior learning applied. In other words, the only added learning benefit comes from the learned correction factors after j learning iterations. The correction factors after each sim-

ulated trial were passed to and run on the real vehicle in order to establish a similar learning curve of maximum and RMS error for comparison. For example, in Figure 12(b), iteration $j = 4$ of V-SL corresponds to running four learning iterations in the simulation, and passing this fourth version of $\tilde{\eta}$ to the real vehicle to observe the results in practice. Again, in this case, we observed similar learning convergence to the VL and SL responses, with slightly higher errors at the end for V-SL. This is reflected by the error plot in Figure 12(a), which shows slightly more error in the final ($J = 6$) V-SL iteration than for VL.

Table 4 shows the differences in maximum and RMS error between the real vehicle learning trials (VL) and the error output from passing simulated correction factors to the vehicle (V-SL). It is clear that there is little difference between conducting learning in simulation or with the actual vehicle for the first four iterations where the differences in maximum lateral error remain below 0.07 m. Iterations 5 and 6, however, demonstrate slightly higher differences in lateral error (up to 0.304 m). This is not surprising because we expect that one of the significant benefits of employing ILC for this application is to incorporate controller adjustments in response to the unmodeled and challenging nature of the operating environment (e.g., rough terrain, significant vehicle dynamics, and other disturbances).

Table 4: Differences in maximum and RMS errors per learning iteration between the real vehicle learning trials (VL) and the error output from passing simulated correction factors to the vehicle (V-SL). A small difference implies that there was very little difference between learning in simulation in comparison with learning in the actual operating environment.

Error Type	1	2	3	4	5	6
Lateral error (ε_L) [m]						
Maximum	0.000	0.051	0.063	0.054	0.159	0.304
RMS	0.000	0.050	0.127	0.114	0.130	0.142
Heading error (ε_H) [degrees]						
Maximum	0.000	0.003	0.015	0.013	0.014	0.017
RMS	0.000	0.001	0.000	0.004	0.008	0.011

(a) Path-following errors along desired path.

(b) Maximum and RMS path-following errors per learning trial.

Figure 12: Lateral and heading errors ($\varepsilon_L, \varepsilon_H$) for pre-simulation trials comparing
52
simulated and real learning trial performance for $J = 6$ learning iterations.

5 Conclusion

To summarize, this paper presents the application of an anticipatory ILC+FBL algorithm for wheeled-vehicle path following to an center-articulated industrial vehicle in a representative environment. A parallel speed learning algorithm was demonstrated (ISLC), which optimizes the desired forward speed profile based on path-following errors in order to increase the vehicle speed while maintaining a maximum allowable lateral error. Finally, ILC+FBL learning was conducted in a simulation environment prior to field trial runs in order to reduce the initial errors associated with learning trials on an actual autonomous vehicle.

Validation of the proposed control schemes was achieved through experimental trials by using two industrial-scale autonomous vehicles in real-world operating conditions. These field trials demonstrated an implementation of ILC+FBL, and its proposed variants. After $J = 10$ learning trials at a high speed of $v = 4.0$ m/s, we observed reductions in maximum errors in the range 89 – 94% for lateral ε_L and 65 – 80% for heading ε_H by using ILC+FBL. Furthermore, the ISLC algorithm provided approximately 40% reduction in total trial time, while keeping path-following errors below a specified threshold. Finally, pre-trial simulation learning was shown to be successful since after $J = 6$ learning iterations in the simulator, the difference in maximum lateral error from vehicle response

compared to real vehicle trials was approximately 0.3 m. Thus, pre-trial learning conducted in a simulator is also a promising option.

Future work should include a more extensive controller design for the ISLC algorithm because in the current work ISLC relied heavily on observations of the steering rate as well as vehicle error as a function of vehicle speed. A formal stability analysis could be conducted for this algorithm, for completeness. Finally, ILC by its nature is highly situational, and thus additional experimental trials could be completed for various desired routes, including different environmental disturbances (e.g., material spills from the bucket), varying loads in the bucket, and altered path geometries. In this work, the desired path was generated by manual tele-operation to provide a kinematically feasible route. There is an obvious opportunity for future experimental trials and evaluation of this control scheme on paths generated by, for example, a optimal path-planning algorithm.

References

Balakrishna, R., & Ghosal, A. (1995, Feb). Modeling of slip for wheeled mobile robots. *IEEE Transactions on Robotics and Automation*, 11(1), 126-132. doi: 10.1109/70.345944

- Barton, A. D., Lewin, P. L., & Brown, D. J. (2000). Practical implementation of a real-time iterative learning position controller. *International Journal of Control*, 73(10), 992-999. doi: 10.1080/002071700405941
- Boe, F., & Hannaford, B. (1998). On-line improvement of speed and tracking performance on repetitive paths. *IEEE Transactions on Control Systems Technology*, 6(3), 350-358. doi: 10.1109/87.668036
- Bristow, D. A., Tharayil, M., & Alleyne, A. G. (2006, June). A survey of iterative learning control. *IEEE Control Systems*, 26(3), 96-114. doi: 10.1109/MCS.2006.1636313
- Chen, Y. Q., & Moore, K. L. (2002). A practical iterative learning path-following control of an omni-directional vehicle. *Asian Journal of Control*, 4(1), 90-98. doi: 10.1111/j.1934-6093.2002.tb00335.x
- Chu, B., Freeman, C. T., & Owens, D. H. (2015, May). A novel design framework for point-to-point ilc using successive projection. *IEEE Transactions on Control Systems Technology*, 23(3), 1156-1163. doi: 10.1109/TCST.2014.2356931
- Chu, B., Owens, D. H., & Freeman, C. T. (2016, May). Iterative learning control with predictive trial information: Convergence, robustness, and experimental verification. *IEEE Transactions on Control Systems Technology*, 24(3), 1101-1108. doi: 10.1109/TCST.2015.2476779

- Dekker, L. G., Marshall, J. A., & Larsson, J. (2017, Sept). Industrial-scale autonomous wheeled-vehicle path following by combining iterative learning control with feedback linearization. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems* (p. 2643-2648). doi: 10.1109/IROS.2017.8206089
- del Río, F. D., Jiménez, G., Sevillano, J. L., Vicente, S., & Balcells, A. C. (2001). A path following control for unicycle robots. *Journal of Robotic Systems*, 18(7), 325-342. doi: 10.1002/rob.1027
- Elci, H., Longman, R. W., Phan, M., Juang, J.-N., & Ugoletti, R. (1994, Oct). Discrete frequency based learning control for precision motion control. In *IEEE International Conference on Systems, Man and Cybernetics* (Vol. 3, p. 2767-2773 vol. 3). doi: 10.1109/ICSMC.1994.400292
- Freeman, C. T. (2015, July). Upper limb electrical stimulation using input-output linearization and iterative learning control. *IEEE Transactions on Control Systems Technology*, 23(4), 1546-1554. doi: 10.1109/TCST.2014.2363412
- Freeman, C. T. (2017). Robust ilc design with application to stroke rehabilitation. *Automatica*, 81, 270 - 278. doi: 10.1016/j.automatica.2017.04.016
- Freeman, C. T., Cai, Z., Rogers, E., & Lewin, P. L. (2011, May). Iterative learning control for multiple point-to-point tracking application. *IEEE Transactions on*

- Control Systems Technology*, 19(3), 590-600. doi: 10.1109/TCST.2010.2051670
- Harmouche, M., Laghrouche, S., & Chitour, Y. (2014). Target point-based path-following controller for a car-type vehicle using bounded controls. *International Journal of Robust and Nonlinear Control*, 25(15), 2705-2725. doi: 10.1002/rnc.3225
- Helmick, D. M., Cheng, Y., Clouse, D. S., Matthies, L. H., & Roumeliotis, S. I. (2004, March). Path following using visual odometry for a mars rover in high-slip environments. In *2004 IEEE Aerospace Conference Proceedings* (Vol. 2, p. 772-789 Vol.2). doi: 10.1109/AERO.2004.1367679
- Ishigami, G., Nagatani, K., & Yoshida, K. (2006, Oct). Path following control with slip compensation on loose soil for exploration rover. In *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems* (p. 5552-5557). doi: 10.1109/IROS.2006.282271
- Isidori, A. (1995). *Nonlinear control systems* (3rd ed.; M. Thoma, E. D. Sontag, B. W. Dickinson, A. Fettweis, J. L. Massey, & J. W. Modestino, Eds.). Secaucus, NJ, USA: Springer-Verlag New York, Inc.
- Kang, M. K., Lee, J. S., & Han, K. L. (2005). Kinematic path-tracking of mobile robot using iterative learning control. *Journal of Robotic Systems*, 22(2), 111-121. doi: 10.1002/rob.20052

- Klančar, G., & Škrjanc, I. (2007, June). Tracking-error model-based predictive control for mobile robots in real time. *Robotics and Autonomous Systems*, 55(6), 460–469.
- Lapierre, L., Soetanto, D., & Pascoal, A. (2006). Nonsingular path following control of a unicycle in the presence of parametric modelling uncertainties. *International Journal of Robust and Nonlinear Control*, 16(10), 485-503. doi: 10.1002/rnc.1075
- Marino, R., & Tomei, P. (2009, Aug). An iterative learning control for a class of partially feedback linearizable systems. *IEEE Transactions on Automatic Control*, 54(8), 1991-1996. doi: 10.1109/TAC.2009.2023967
- Marshall, J., Barfoot, T., & Larsson, J. (2008). Autonomous underground tramming for center-articulated vehicles. *Journal of Field Robotics*, 25(6-7), 400–421. doi: 10.1002/rob.20242
- Moore, K. L., & Bahl, V. (1999, Nov). Iterative learning control technique for mobile robot path-tracking control. In *Proceedings of The International Society for Optical Engineering* (Vol. 3838). doi: 10.1117/12.369258
- Nayl, T., Nikolakopoulos, G., & Gustafsson, T. (2012, Sept). Path following for an articulated vehicle based on switching model predictive control under varying speeds and slip angles. In *Proceedings of 2012 IEEE 17th International*

- Conference on Emerging Technologies Factory Automation* (p. 1-7). doi: 10.1109/ETFA.2012.6489635
- Normey-Rico, J. E., Alcalá, I., Gómez-Ortega, J., & Camacho, E. F. (2001). Mobile robot path tracking using a robust PID controller. *Control Engineering Practice*, 9(11), 1209 - 1214. doi: [https://doi.org/10.1016/S0967-0661\(01\)00066-1](https://doi.org/10.1016/S0967-0661(01)00066-1)
- Ostafew, C. J., Schoellig, A. P., & Barfoot, T. D. (2013, Nov). Visual teach and repeat, repeat, repeat: Iterative learning control to improve mobile robot path tracking in challenging outdoor environments. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems* (p. 176-181). doi: 10.1109/IROS.2013.6696350
- Ostafew, C. J., Schoellig, A. P., Barfoot, T. D., & Collier, J. (2014, May). Speed daemon: Experience-based mobile robot speed scheduling. In *2014 Canadian Conference on Computer and Robot Vision* (p. 56-62). doi: 10.1109/CRV.2014.16
- Ostafew, C. J., Schoellig, A. P., Barfoot, T. D., & Collier, J. (2016, January). Learning-based nonlinear model predictive control to improve vision-based mobile robot path tracking. *Journal of Field Robotics*, 33(1), 133–152.
- Paden, B., Cap, M., Yong, S. Z., Yershov, D., & Frazzoli, E. (2016, March). A survey of motion planning and control techniques for self-driving urban vehicles. *IEEE Transactions on Intelligent Vehicles*, 1(1), 33-55. doi: 10.1109/TIV.2016

.2578706

- Pakshin, P., Emelianova, J., Emelianov, M., Gałkowski, K., & Rogers, E. (2016). Dissipativity and stabilization of nonlinear repetitive processes. *Systems & Control Letters*, 91, 14 - 20. doi: 10.1016/j.sysconle.2016.01.005
- Pakshin, P., Emelianova, J., Emelianov, M., Gałkowski, K., & Rogers, E. (2016, Dec). Iterative learning control design based on feedback linearization and nonlinear repetitive process stability theory. In *2016 IEEE 55th Conference on Decision and Control* (p. 5478-5483). doi: 10.1109/CDC.2016.7799110
- Shao, H., & Chen, Y. (2010, June). A modified trajectory tracking algorithm based on ILC for mobile robots. In *2010 International Conference on Intelligent Computing and Cognitive Informatics* (p. 312-315). doi: 10.1109/ICICCI.2010.46
- van den Berg, J., Miller, S., Duckworth, D., Hu, H., Wan, A., Fu, X. Y., ... Abbeel, P. (2010, May). Superhuman performance of surgical tasks by robots using iterative learning from human-guided demonstrations. In *2010 IEEE International Conference on Robotics and Automation* (p. 2074-2081). doi: 10.1109/ROBOT.2010.5509621
- Wallén, J., Norrlöf, M., & Gunnarsson, S. (2008). Arm-side evaluation of ILC applied to a six-degrees-of-freedom industrial robot. *IFAC Proceedings Volumes*, 41(2), 13450 - 13455. (17th IFAC World Congress) doi: 10.3182/

20080706-5-KR-1001.02278

- Wang, D., & Ye, Y. (2005, June). Design and experiments of anticipatory learning control: frequency-domain approach. *IEEE/ASME Transactions on Mechatronics*, 10(3), 305-313. doi: 10.1109/TMECH.2005.848297
- Wang, H., Dong, J., & Wang, Y. (2016, July). Discrete PID-type iterative learning control for mobile robot. *Journal of Control Science and Engineering*, Pages: 7. doi: 10.1155/2016/2320746
- Xi, R., Li, Y., & Xiao, X. (2014, July). Trajectory tracking control for a nonholonomic mobile robot using an improved ILC. In *2014 IEEE International Conference on Information and Automation* (p. 830-835). doi: 10.1109/ICInfA.2014.6932766
- Zhao, Y. M., Lin, Y., Xi, F., & Guo, S. (2015, May). Calibration-based iterative learning control for path tracking of industrial robots. *IEEE Transactions on Industrial Electronics*, 62(5), 2921-2929. doi: 10.1109/TIE.2014.2364800