

Iterative Learning Control of Spacecraft Proximity Operations Based on Confidence Level

Steve Ulrich* and Kirk Hovell†

Carleton University, Ottawa, Ontario K1S 5B6, Canada

This paper addresses the problem of trajectory tracking control for small free-flyer spacecraft in proximity operations. The proposed approach consists in a current output-feedback iterative learning control law that combines a causal output feedback and a non-causal feed-forward learning component. A new confidence level factor is introduced in the algorithm to carefully, and iteratively, transition learning from a conservative to a confident process. Simulations and experiments performed at Carleton University's Spacecraft Robotics and Control Laboratory demonstrate the performance of the new confidence-based learning approach in a spacecraft robotic inspection maneuver scenario. Results indicate that the iterative learning controller is successful at achieving accurate tracking performance for a fast trajectory without saturating the low-thrust actuators onboard the spacecraft.

I. Introduction

DRIVEN by recent on-orbit servicing and orbital debris removal initiatives, such as the German Orbital Servicing mission, the Defense Advanced Research Projects Agency's Phoenix program and Robotic Servicing of Geosynchronous Satellites program, the European Space Agency's Clean Space program, as well as NASA's Restore-L mission, the space industry has undergone a paradigm shift toward the development of intelligent robotic spacecraft. These have significantly higher levels of autonomy, and are equipped with one or more dexterous robotic arms capable of physically interacting with uncooperative objects in space, including depleted rocket engines and decommissioned communication satellites. Such advanced missions concepts often require robotic spacecraft to autonomously perform *proximity operations*; that is, rendezvous, inspect, and capture an uncooperative and possibly tumbling target. A key to successfully and safely accomplish these complex maneuvers is the capability to track the desired relative motion between the robotic spacecraft and the target object with a high level of accuracy and precision, regardless of external perturbations and other dynamics uncertainties. Indeed, given the relatively short time-scale and small separations during proximity operations, these adverse effects can significantly reduce the performance and safety of trajectory tracking actions.

In this context, this paper develops a novel learning-based control scheme for fast and precise trajectory tracking for small free-flyer robotic spacecraft operating in close proximity, and validates it through experiments at Carleton University's Spacecraft Proximity Operation Testbed (SPOT). The main feature of the new control scheme is its capability to transition from conservative to confident learning while minimizing its computational requirement, making it highly suitable for safe and efficient operations.

I.A. Related Work

In proximity operations, the spacecraft would typically be close enough to employ the Clohessy-Wiltshire equations¹ to model the relative dynamics. Furthermore, when the time scale is significantly less than the orbital period of the target spacecraft, and the distance between both vehicles is relatively small, a simple double-integrator model can be applied.² Being linear, these dynamics representations are particularly attractive for feedback control purposes, which is why the literature proposes numerous linear feedback control laws to regulate or track a prescribed relative trajectory. For example, laboratory demonstrations

* Assistant Professor, Department of Mechanical and Aerospace Engineering, 1125 Colonel By Drive. Senior Member AIAA.

† Graduate Student, Department of Mechanical and Aerospace Engineering, 1125 Colonel By Drive. Student Member AIAA.

of simple linear control laws such as proportional-derivative and linear quadratic regulator were performed at the Massachusetts Institute of Technology,^{3,4} and an experimental validation of a linear quadratic model predictive control for the rendezvous and docking of two spacecraft platforms was recently conducted.⁵ Nonlinear control methodologies have also been designed and evaluated for their ability to improve the trajectory tracking performance compared to linear feedback controllers. Some of these techniques include adaptive feedback linearization,⁶ sliding mode control,⁷ and more recently, state-dependent Riccati equation-based control,⁸ as well as direct adaptive control.⁹

Amongst proximity operation maneuvers, it is notable that inspection maneuvers often consist in following the same bounded relative trajectory repeatedly, such as a circle or an ellipse, under very similar operating conditions and external perturbations. All aforementioned control schemes are *causal* controllers, i.e., feedback controllers, with the result that applying these approaches to track the same trajectory in a repeated manner would yield the same tracking error on each pass, as these causal controllers do not exploit error and control input signals from previous iterations. In other words, these existing schemes do not exploit past experience to increasingly improve tracking performance. Alternatively, data-based controllers that can track pre-defined, repetitive, commands with increasing accuracy, through updating control inputs by learning from previous executions, may be well suited for this application. More specifically, iterative learning control (ILC) algorithms that improve tracking performance by incorporating error information into the control input for subsequent iterations can be efficiently used to address the problem of precise trajectory tracking for repetitive proximity operation maneuvers. The new control approach developed in this paper falls in the area of ILC.

The idea of ILC was first widely popularized in the work of Arimoto et al.,¹⁰ and since then, it has found many successfull practical applications in manufacturing, industrial robotic manipulations, and chemical processing, where mass production on an assembly line inherently entails repetitive tasks. See Bristow et al.¹¹ and Ahn et al.¹² for comprehensive surveys on the topic. However, space-based applications are more rare. Porter and Mohamed^{13,14} investigated a model-based ILC scheme and its indirect adaptive version where the unknown plant matrices are identified in real-time. An adverse consequence of on-line identification procedures is their increased computational burden, which could rule out the use of such approaches for small free-flyer robotic spacecraft with limited computational resources. An optimal feedback control loop combined to a learning feedforward control loop was proposed by Wu et al.¹⁵ for a perturbed satellite formation keeping application. Their approach consisted of regulating the relative position over multiple orbits. However, the initial conditions were not reset to the same values on each trial; a violation to the fundamental initial condition requirement to be satisfied for ILC schemes. This initial condition requirement difficulty inherent to formation flying was overcome by Ahn et al.¹⁶ via the application of a robust periodic learning control law which was developed previously by Xu and Qu.¹⁷ More recently, Wu et al.¹⁸ designed an ILC approach for attitude tracking for a satellite with a repeating ground track orbit. Their feedback control law is used to ensure stability and compensate for non-repeating disturbances, and the ILC scheme is used to reject the effects of the repeating disturbances, such as atmospheric drag, gravity gradient, magnetic torque, and solar radiation pressure.

I.B. Contributions

The contributions of this paper to the field of learning-based control is twofold.

First, a learning algorithm that combines a causal feedback controller to a non-causal learning signal is developed. The new algorithm is designed with real-time implementation considerations in mind. As such, it is developed as a simplified version of the existing parallel current-iteration scheme,^{19–21} by neglecting the so-called Q - and L - filters which are part of the original algorithm. Doing so alleviates the difficulty in design and tuning those two filters while also reducing the computational burden to implement the learning algorithm on a real-time microprocessor. One problem that may arise in practical scenarios with the original, or even the simplified, parallel current-iteration scheme is actuator saturation. This is because the previous control input signal is directly used in a feedforward configuration (or first passed through a low-pass filter to smooth out its noise content) resulting in a very fast, steep learning of the pre-defined trajectory. This type of learning model where the previous control input is used *as is* to calculate the current control input is herein referred to as *confident learning*. While this type of learning will result in a rapid decrease in the tracking error, it may also rapidly result in actuator saturation, which in turn may yield to a decrease in tracking performance with consecutive iterations. For this reason, the simplified parallel current-iteration learning scheme is further modified by introducing the *confidence level factor* into the algorithm. This new parameter

allows the control of the confidence level in the control input signal associated with the previous iteration. A simple heuristic to increase this parameter in a linear fashion, from low to high, with consecutive trials is provided. This has the beneficial effect of transitioning the algorithm from conservative to confident learning, thus resulting in a learning process that better meets the need for safe and efficient operations. In contrast to disturbance observer-based ILC algorithms^{22,23} which are known to also increase the level of confidence of the learning process through the estimation of an increasingly accurate model of the disturbances with successive iterations, the proposed approach is model-free. As such, it does not require any state and/or disturbance on-line predictors.

Second, the derived confidence level-based learning scheme is thoroughly applied in laboratory experiments for robotic spacecraft platforms to achieve fast and accurate trajectory tracking while in close proximity. This allows the testing of the new learning controller under practical, real-time conditions (e.g., measurement noise, limited computation power, control actuation limits, delays, and quantization of signals). Based on the authors' knowledge, no previous work experimentally investigated the use of ILC for spacecraft applications.

To summarize, the new learning algorithm combined with its experimental validation on spacecraft robot platforms represent the original contributions of this work.

The remaining of this paper is organized as follows. Section II develops the new iterative learning control law based on confidence level. Section III provides a simple proximity operation application example to illustrate the performance of the new learning algorithm. Section IV describes the laboratory setup and discusses the experiment results. Finally, Section V provides a conclusion.

II. Iterative Learning Control Based on Confidence Level

In this section, a new ILC scheme that makes use of a feedback controller and a feedforward control signal which is modified according to the confidence level associated with the previous iteration is developed. Typical ILC schemes use open-loop control action only, which cannot compensate for nonrepeating disturbances. Thus, in most practical implementations, a current-feedback strategy is preferred where a feedback controller is used in combination with ILC. In most cases, a typical causal feedback controller can be designed without ILC, and then, ILC can be implemented in a feedforward fashion without modifying the feedback controller. As explained by Bristow et al.,¹¹ ILC can be combined with a feedback loop in two ways: (1) in series, where the ILC control input is applied to the desired trajectory, and (2) in parallel, where the ILC control input and feedback control input are combined. The feedback controller modifies the input/output dynamics with regard to the ILC depending on the particular configuration.

The parallel architecture, adopted for this work for its inherent implementation simplicity, directly alters the feedback control inputs by adding feedforward signals for improved tracking performance. As the ILC converges, the feedback controller applies less effort. One advantage of this configuration is that the individual control contributions from the ILC and feedback controller are easily separable in this architecture. For example, the ILC can be disabled whenever non-repeating reference trajectories are used.

In this section, a widely-used parallel ILC law is first simplified to ease its real-time implementation on small spacecraft platforms, and then the resulting simplified ILC law is furthermore modified by introducing a scaling factor to the feedforward control input signal based on the confidence level of the current iteration.

II.A. Simplified Parallel ILC Design

Perhaps one of the most widely-used ILC learning algorithm is^{19–21}

$$\mathbf{u}_{j+1}(k) = \mathbf{Q}(\mathbf{q})[\mathbf{u}_j(k) + \mathbf{L}(\mathbf{q})\mathbf{e}_j(k+1)] \quad (1)$$

where $k \in \mathbb{N}^0 = \{0, 1, \dots, n-1\}$ denotes the discrete-time index and where $n \in \mathbb{N} < \infty$ is the trial length in discrete-time steps, that is, $n = t_f/\Delta t$ with Δt and t_f being the sampling time and final time of the trial in seconds, respectively, and where $j \in \mathbb{N}$ denotes the positive iteration index. The LTI dynamics $\mathbf{Q}(\mathbf{q})$ and $\mathbf{L}(\mathbf{q})$ are defined as the Q - and L -filter, respectively. The Q -filter is most often a zero-phase finite-impulse-response Gaussian low-pass filter which adds robustness and high-frequency noise filtering, whereas the L -filter is a function that acts on the basis of the previous iteration tracking error.

In a parallel current-feedback ILC, an additional term, $\mathbf{C}(\mathbf{q})\mathbf{e}_{j+1}(k)$, is added to the learning component

in the current iteration, that is

$$\mathbf{u}_{j+1}(k) = \mathbf{Q}(\mathbf{q})[\mathbf{u}_j(k) + \mathbf{L}(\mathbf{q})\mathbf{e}_j(k+1)] + \mathbf{C}(\mathbf{q})\mathbf{e}_{j+1}(k) \quad (2)$$

In other words, this algorithm is identical to the algorithm Eq. (1) combined with a feedback controller in the parallel architecture. Equation (2) can be written concisely by separating the current-iteration ILC signal into feedforward and feedback components as

$$\mathbf{u}_{j+1}(k) = \mathbf{w}_j(k) + \mathbf{u}_{j+1}^{fb}(k) \quad (3)$$

where the feedforward component, denoted $\mathbf{w}_j(k)$, is given by

$$\mathbf{w}_j(k) = \mathbf{Q}(\mathbf{q})[\mathbf{u}_j(k) + \mathbf{L}(\mathbf{q})\mathbf{e}_j(k+1)] \quad (4)$$

and where the feedback component, denoted $\mathbf{u}_{j+1}^{fb}(k)$, is

$$\mathbf{u}_{j+1}^{fb}(k) = \mathbf{C}(\mathbf{q})\mathbf{e}_{j+1}(k) \quad (5)$$

Assuming a simple linear causal feedback control law can be derived for the nominal plant, the greatest difficulty in designing the parallel ILC controller in Eq. (2) is determining the Q - and L - filters. The ILS theory does not provide any means of systematically selecting these filters, although Ahn et al.¹² do give some recommendations. To overcome this design complexity and to further decrease the number of required operations needed to for real-time implementation, the parallel learning law is herein intuitively simplified by setting $\mathbf{w}_j(k) = \mathbf{u}_j(k)$. This way, the simplified algorithm has the advantage of requiring less computation power.

For the feedback component, a proportional-derivative law acting on the current iteration is proposed, that is

$$\mathbf{u}_{j+1}^{fb}(k) = \mathbf{k}_p \mathbf{e}_{j+1}(k) + \mathbf{k}_d \frac{(\mathbf{e}_{j+1}(k) - \mathbf{e}_{j+1}(k-1))}{\Delta t} \quad (6)$$

where \mathbf{k}_p and \mathbf{k}_d respectively denote the proportional and derivate control gain matrix, and where the trajectory tracking error for iteration $j+1$ is denoted by $\mathbf{e}_{j+1}(k)$. In view of the above, the resulting simplified ILC control input for iteration $j+1$ is given by

$$\mathbf{u}_{j+1}(k) = \mathbf{u}_j(k) + \mathbf{k}_p \mathbf{e}_{j+1}(k) + \mathbf{k}_d \frac{(\mathbf{e}_{j+1}(k) - \mathbf{e}_{j+1}(k-1))}{\Delta t} \quad (7)$$

II.B. Confidence Level-Based Feedforward Signal

One problem that may arises in any practical implementations of either the original or the simplified version of the parallel ILC control law is actuator saturation. Indeed, if the first iteration (causal-only) does not provide adequate tracking performance, the actuation efforts for the second iteration will have to be greatly increased to compensate for the undesirable behavior experienced during the previous iteration. For example, inadequate performance exhibited during the first iteration may be characterized by large steady-state errors due to a-priori plant dynamics models that are not sufficiently accurate, thereby resulting in poorly-designed feedback control laws. After only a few iterations, actuator saturation may occur, and any additional iterations will decrease the tracking performance, eventually leading to divergence of the closed-loop system.

To overcome this inherent problem related to practical implementations, a *confidence level factor*, denoted $0 < \beta_{j+1} \in \mathbb{R} < 1$, is introduced in the learning algorithm. This scalar parameter represents the current confidence in the previous control input signal, and its selection is intuitive; 1 being associated with excellent tracking performance exhibited during the previous iteration (perfect confidence), and 0 with poor performance during the previous iteration (not trustworthy). With the confidence level factor, the previously-simplified ILC law becomes

$$\mathbf{u}_{j+1}(k) = \beta_{j+1} \mathbf{u}_j(k) + \mathbf{k}_p \mathbf{e}_{j+1}(k) + \mathbf{k}_d \frac{(\mathbf{e}_{j+1}(k) - \mathbf{e}_{j+1}(k-1))}{\Delta t} \quad (8)$$

Note that $\beta_{j+1} = 1$ simplifies the algorithm to Eq. (7) and that $\beta_{j+1} = 0$ implies that the non-causal feedforward signal is ignored, with the latter case indicating that a re-design of the feedback controller may

be appropriate. While the confidence level factor can be kept constant for all iterations, it is advisable to update it at each iteration, to reflect the fact that, as the plant iteratively learns to track the desired outputs occurs with increasing performance, more confidence can be placed on the previous control inputs. In other words, it is desirable to increase β_{j+1} with the iteration number, so that learning transitions from a conservative to a confident process, ultimately reaching a point where the causal part of the algorithm almost vanishes after several iterations. A heuristic update scheme of the confidence level factor that accomplishes this is herein proposed as

$$\beta_{j+1} = j/N \quad (9)$$

where $N \in \mathbb{N}$ denotes the number of trials. This update corresponds to a linear increase in confidence level with consecutive iterations. Note that other update schemes could also be used to control the learning rate of the algorithm, such as exponential laws. The proposed parallel ILC scheme based on confidence level is illustrated in Fig. 1.

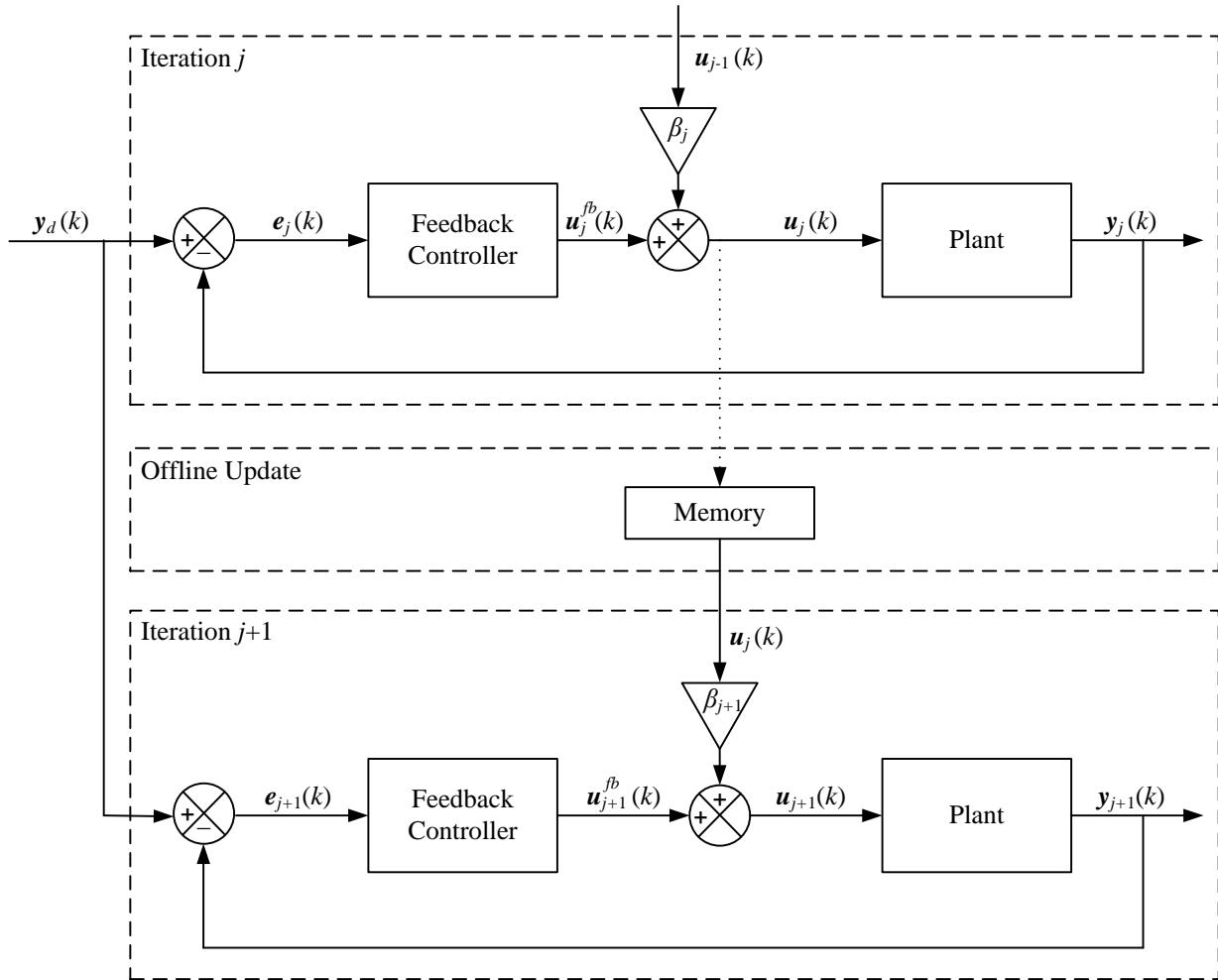


Figure 1. Block-scheme diagram of the proposed simplified parallel iterative learning control law based on confidence level.

III. Illustrative Example

In this section, the new learning controller is validated in numerical simulations for the problem of accurate trajectory tracking in proximity operations. First, the output tracking problem is defined, and then continuous linear time-invariant dynamics models for spacecraft relative motion as well as their discretized versions are presented. Finally, simulation results are presented for a 50-m circular inspection trajectory

with two different designs of the confidence level factor β_{j+1} .

III.A. Problem Statement

The output tracking control objective consists of ensuring that the actual position of an inspector spacecraft with respect to a target object tracks as closely as possible a pre-defined, desired, repetitive inspection trajectory around the object, without requiring any precise knowledge of the actual plant physical parameters, and with increasing accuracy through iterative update of the control inputs by learning from previous executions.

To quantify this control objective, a three-dimensional output tracking error for iteration $j + 1$, denoted $\mathbf{e}_{j+1}(k) \in \mathbb{R}^3$, is defined as

$$\mathbf{e}_{j+1}(k) \triangleq \mathbf{y}_d(k) - \mathbf{y}_{j+1}(k) \equiv \begin{bmatrix} \tilde{x}_{j+1}(k) \\ \tilde{y}_{j+1}(k) \\ \tilde{z}_{j+1}(k) \end{bmatrix} \quad (10)$$

where $\mathbf{y}_d(k) = [x_d(k) \ y_d(k) \ z_d(k)]^T \in \mathbb{R}^3$ denotes the desired relative position components in the Local-Vertical-Local-Horizontal (LVLH) reference frame, denoted by \mathcal{F}_L and defined with its origin located at the target spacecraft, with its \vec{L}_x unit vector in the direction of the target position vector, its \vec{L}_z unit vector perpendicular to the orbital plane, and its \vec{L}_y unit vector completing the right-handed orthogonal frame. Under the assumption that the relative position between both spacecraft is available for feedback control purposes, $\mathbf{y}_{j+1}(k) \in \mathbb{R}^3$ denotes the $(j + 1)^{\text{th}}$ -iteration discretized output of the plant which is given by

$$\mathbf{y}_{j+1}(k) = [x_{j+1}(k) \ y_{j+1}(k) \ z_{j+1}(k)]^T \quad (11)$$

where $x_{j+1}(k), y_{j+1}(k), z_{j+1}(k) \in \mathbb{R}$ are the actual relative position components in LVLH.

III.B. Continuous Linear Time-Invariant Dynamics

Defining the states as $\mathbf{x}(t) = [x \ y \ z \ \dot{x} \ \dot{y} \ \dot{z}]^T \in \mathbb{R}^6$, and the control input forces as $\mathbf{u}(t) = [f_x \ f_y \ f_z]^T \in \mathbb{R}^3$, allows the linearized relative dynamics applicable to close proximity operations to be described as a square state-space system of the form

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t), \quad \mathbf{y}(t) = \mathbf{C}\mathbf{x}(t) \quad (12)$$

where the appropriately-dimensioned time-invariant input and output matrices, \mathbf{B} and \mathbf{C} , are given by

$$\mathbf{B} = \begin{bmatrix} \mathbf{0}_{3 \times 3} \\ \frac{1}{m} \mathbf{I}_{3 \times 3} \end{bmatrix}, \quad \mathbf{C} = \begin{bmatrix} \mathbf{I}_{3 \times 3} & \mathbf{0}_{3 \times 3} \end{bmatrix} \quad (13)$$

with $m \in \mathbb{R}$ denoting the mass of the inspector spacecraft. The system matrix $\mathbf{A} \in \mathbb{R}^{6 \times 6}$ in Eq. (12) depends on the formulation employed to model the relative dynamics. If the target object orbit is assumed to be circular (or near-circular), and the separation distance is relatively small compared to their orbital radii, the nonlinear equations of relative motion simplify to the well-known Clohessy-Wiltshire (CW) equations¹ and the system matrix is

$$\mathbf{A}_{\text{CW}} = \begin{bmatrix} \mathbf{0}_{3 \times 3} & & & \mathbf{I}_{3 \times 3} \\ 3n^2 & 0 & 0 & 0 & 2n & 0 \\ 0 & 0 & 0 & -2n & 0 & 0 \\ 0 & 0 & -n^2 & 0 & 0 & 0 \end{bmatrix} \quad (14)$$

where $n \in \mathbb{R}$ denotes the orbital mean motion given by

$$n = \sqrt{\mu/a^3} \quad (15)$$

with $a \in \mathbb{R}$ denoting the semi-major axis of the target spacecraft orbit and $\mu \in \mathbb{R}$ denoting the gravitational parameter of the Earth ($398,600 \text{ km}^3/\text{s}^2$).

As shown in Eq. (14), the in-plane (x - y) and out-of-plane (z) motions are decoupled, with the in-plane coupling terms correspond to Coriolis and gravity forces, while the out-of-plane is an undamped oscillation with the orbital mean motion. As noted by Fehse,² these terms can be considered low-frequency perturbations to be compensated by a feedback controller, which means a double-integrator model that is independent for each axis can be used. Also, it can easily be shown that in close proximity, that is, when the maximum separation is on the order of a few tens of meters, these coupling terms are at least one order of magnitude smaller than the remaining double-integrator terms in the CW equations. This simplification is also justified whenever the spacecraft are maneuvering on a faster time scale than the orbital mean motion, n . As explained by Paluszak and Thomas,²⁴ this is evident because the bode plots of the double-integrator superimposed on the CW equations are identical in frequencies higher than n . With the double-integrator simplification, the system matrix \mathbf{A} becomes independent of n , and reduces to

$$\mathbf{A}_{\text{int}} = \begin{bmatrix} \mathbf{0}_{3 \times 3} & \mathbf{I}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} \end{bmatrix} \quad (16)$$

III.C. Discretized Linear Time-Invariant Dynamics

In practice, however, control inputs and plant outputs are respectively fed and measured at fixed time steps. To capture this, a discrete-time representation of the plant dynamics Eq. (12) is obtained, and the resulting time-invariant difference equations are given by

$$\mathbf{x}_j(k+1) = \mathbf{A}_D \mathbf{x}_j(k) + \mathbf{B}_D \mathbf{u}_j(k), \quad \mathbf{y}_j(k) = \mathbf{C}_D \mathbf{x}_j(k) \quad (17)$$

Assuming a spacecraft mass of $m = 20 \text{ kg}$, a semi-major axis $a = 7,200 \text{ km}$, and a sampling frequency $f = 1/\Delta t = 240 \text{ Hz}$ results in the following discretized system and input matrices for the double-integrator model

$$\mathbf{A}_{D_{\text{int}}} = \begin{bmatrix} 1 & 0 & 0 & 0.0042 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0.0042 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0.0042 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{B}_{D_{\text{int}}} = 1e-3 \begin{bmatrix} 0.004 & 0 & 0 \\ 0 & 0.004 & 0 \\ 0 & 0 & 0.004 \\ 0.2083 & 0 & 0 \\ 0 & 0.2083 & 0 \\ 0 & 0 & 0.2083 \end{bmatrix}$$

and for the CW model

$$\mathbf{A}_{D_{\text{CW}}} = \begin{bmatrix} 1 & 0 & 0 & 0.0042 & 1.79e-8 & 0 \\ 0 & 1 & 0 & -1.79e-8 & 0.0042 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0.0042 \\ 0 & 0 & 0 & 1 & 8.61e-6 & 0 \\ 0 & 0 & 0 & 8.61e-6 & 1 & 0 \\ 0 & 0 & -4.45e-9 & 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{B}_{D_{\text{CW}}} = 1e-3 \begin{bmatrix} 0.004 & 1.25e-9 & 0 \\ -1.25e-9 & 0.004 & 0 \\ 0 & 0 & 0.004 \\ 0.2083 & 8.97e-7 & 0 \\ -8.97e-7 & 0.2083 & 0 \\ 0 & 0 & 0.2083 \end{bmatrix}$$

In the following subsection, the feedback control parameter of the new learning controller will be tuned in simulations for the double-integrator model, whereas the CW model will be used to assess the performance of the controller.

III.D. Simulation Results

Numerical simulations in MATLAB/Simulink were performed to assess the effectiveness of the proposed confidence level-based simplified ILC approach. In this example, a 50-m radius circular inspection trajectory in the along track and cross-track ($\tilde{L}_y - \tilde{L}_z$) directions must be tracked over multiple repetitions. As mentioned earlier, the plant to which the controller is applied corresponds to the discretized CW dynamics formulation. The dynamics was initialized with a non-zero offset in the along-track direction, denoted y_0 and equal to -50 m. The desired trajectories and initial conditions represent a free-flyer robotic spacecraft that is initially at rest with respect to the target object and is then commanded to perform a fly-around inspection maneuver around the target in the $\tilde{L}_y - \tilde{L}_z$ plane in 300 s, with a constant angular velocity. Note that the selection of this desired relative trajectory does not take into account any fuel consumption considerations. Nevertheless, it demonstrates the capability of the new ILC scheme to track fast and demanding trajectories that may be required in proximity operations. Furthermore, a control input force limit of ± 2.5 N was imposed along each axis, to illustrate the main benefits of employing a learning process that transitions from conservative to confident, i.e., with $\beta_{j+1} = j/N$, compared to a confident approach, with $\beta_{j+1} = 1$. The learning law was initialized with $\mathbf{u}_0 = \mathbf{0}_{N \times 1}$ and $\beta_1 = 0$, and the feedback parameters were chosen as

$$\mathbf{k}_p = \begin{bmatrix} 0.172 & 0 & 0 \\ 0 & 0.172 & 0 \\ 0 & 0 & 0.100 \end{bmatrix}, \quad \mathbf{k}_d = \begin{bmatrix} 4 & 0 & 0 \\ 0 & 4 & 0 \\ 0 & 0 & 3 \end{bmatrix}$$

To illustrate the robustness of the proposed ILC scheme to dynamics modelling errors, these feedback parameters were tuned in numerical simulations to achieve a satisfactory response when applied to the discretized dynamics Eq. (17) with the double-integrator dynamics matrices $\mathbf{A}_{D_{\text{int}}}$ and $\mathbf{B}_{D_{\text{int}}}$, and a desired radius of 1 m. Then, the same ILC scheme was applied to the discretized dynamics, but this time with Clohessy-Wiltshire matrices, $\mathbf{A}_{D_{\text{CW}}}$ and $\mathbf{B}_{D_{\text{CW}}}$ and a desired radius of 50 m, without re-tuning the control parameters. This way, the robustness of the ILC controller to unmodeled dynamics can be assessed.

Results obtained by using the proposed learning controller are provided in Figs. 2 and 3 for both tuning cases of β_{j+1} . These figures report the circular trajectory and the three-dimensional average position error defined as

$$\frac{1}{n} \sum_{k=1}^n \sqrt{\tilde{x}(k)^2 + \tilde{y}(k)^2 + \tilde{z}(k)^2}$$

In the first iteration, as depicted in both figures, the actual trajectory is far off, with an average positon error of 3.3 m. In both design cases, despite the large initial offset, the spacecraft learns to track the reference trajectory over the next nine iterations. In Fig. 2, this is demonstrated through the corresponding evolution of the average position tracking error, which rapidly settles around 0.5 m. This illustrates the expected behavior of a confident learning controller, which considers the previous control input signal, with no modification, into the calculation of the current control input. However, as shown in this figure, the controller stops to learn after the third iteration. This is due to the actuation limits, which prevent further increase in control actuation. In addition, as slight decrease in performance is observed. Specifically, from the third to the tenth iteration, the average position error steadily increases from 0.4984 m to 0.5217 m.

On the other hand, with the more conservative approach where the confidence level is linearly increased, the controller keeps learning iteratively from its past experience; modifying its learning behaviour from conservative ($\beta_2 = 0.1$) to confident ($\beta_{10} = 0.9$). Although the final average position error at the tenth iteration is nearly identical, it is important to recall that using the linear update law (conservative design) does not result in actuator saturation. This yields safer and more efficient operations. In addition, as it will be shown in the following section, employing a constant confidence level factor $\beta_{j+1} = 1$ under real practical situations further deteriorates the tracking performance.

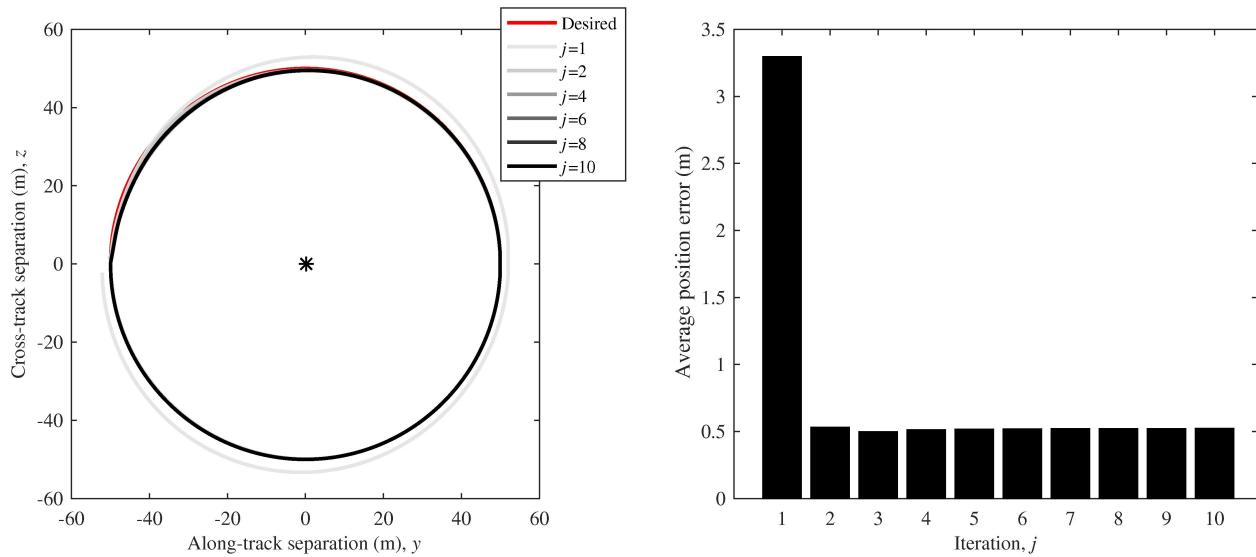


Figure 2. Simulated learning of a 50-m circular trajectory with $\beta_{j+1} = 1$. The spacecraft out-of-plane (yz -plane) motion, and the average position error, are depicted for different iterations.

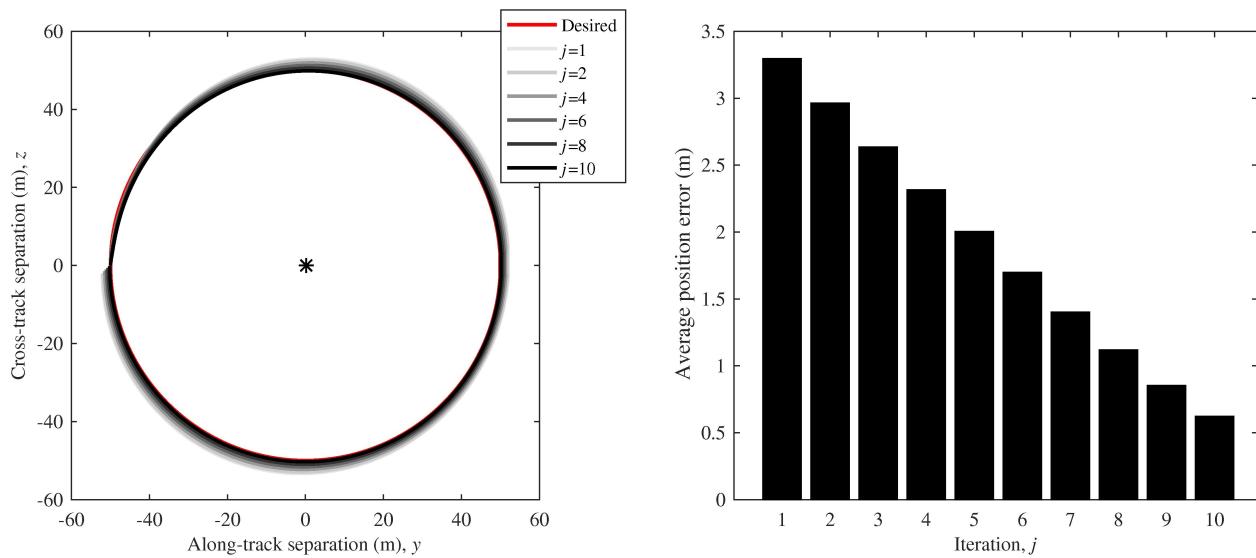


Figure 3. Simulated learning of a 50-m circular trajectory with $\beta_{j+1} = j/N$. The spacecraft out-of-plane (yz -plane) motion, and the average position error, are depicted for different iterations.

IV. Experimental Validation

Experiments were conducted using a chaser spacecraft platform floating on air bearings to track a circular trajectory around a stationary target object. A series of 10 trials were performed for both β_{j+1} design approaches. During the experiments, ground truth data (position and angle) was recorded and average position errors were calculated after each trial for analysis purposes.

IV.A. Experiment Setup

Experiments were conducted at Carleton University's Spacecraft Robotics and Control Laboratory, using the Spacecraft Proximity Operations Testbed (SPOT). Specifically, SPOT consists of two air-bearing spacecraft platforms operating in close proximity on a 2.4 m \times 3.7 m granite surface (see Fig. 4). The use of air bearing on the platforms reduces the friction to a negligible level. Due to surface slope angles of 0.0026 and 0.0031 deg along both directions, residual gravitational acceleration of 0.439 and 0.525 mm·s⁻² perturbs the dynamics of the floating platforms along the x and y direction, respectively. Both platforms are 0.3 \times 0.3 \times 0.3 m in dimensions and are actuated by compressed air at 80 psi expelled through eight miniature air nozzles distributed around each platform, thereby providing full planar control authority. The structure consists of an aluminum frame with four corner rods on which three modular decks are stacked. To protect the internal components, the structure is covered with semi-transparent acrylic panels. Figure 5 shows a picture of a fully assembled SPOT spacecraft platform. Each thruster generates 0.25 N of thrust and is controlled at a frequency of 10 Hz by a pulse-width modulation scheme using solenoid valves. Pressurized air for the thrusters and the air bearing flotation system is stored onboard in a single air cylinder at 4500 psi. When the air cylinder is full, the mass of a platform is 17 kg.



Figure 4. Spacecraft Proximity Operation Testbed at Carleton University's Spacecraft Robotics and Control Laboratory.

The motion of both platforms is measured in real-time through four active LEDs on each platform, which are tracked by an 8-camera PhaseSpace motion capture system. This provides highly accurate ground truth position and attitude data. All motion capture cameras are connected to the PhaseSpace server, itself connected to the ground station computer. This computer then communicates, via a wireless user datagram protocol, ground truth information to the platforms' onboard computers, which consist of Raspberry Pi-3 running the Raspbian Linux operating system. Based on their respective position and attitude data the platforms perform feedback control, calculating their required thrust to maneuver autonomously through the pulse-width-modulation activation of their solenoid valves. The ground station computer also receives real-time telemetry data (i.e., any signals of interest, as specified by the user) from all onboard computers, for post-experiment analysis purposes. Additional sensors are also available on-board the platforms, including a three-axis inertial measurement unit comprising of an accelerometer and a gyroscope, as well as a pair of

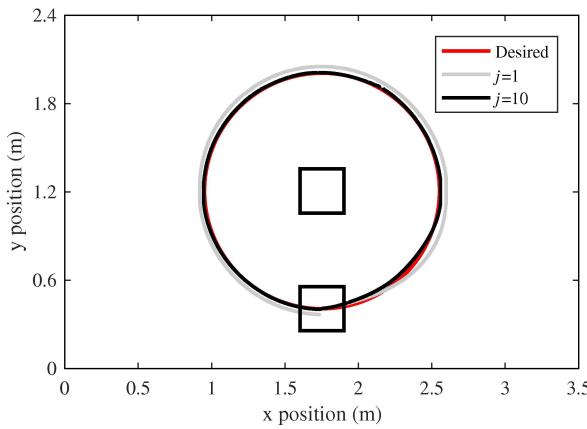
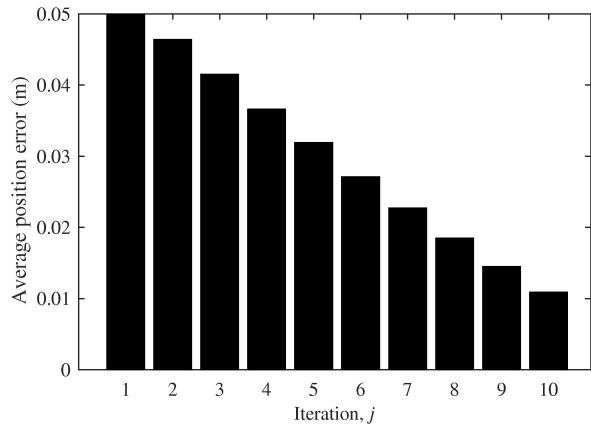


Figure 6. Simulated learning of a 1-m circular inspection trajectory with $\beta_{j+1} = j/N$. The planar trajectory and average position error are depicted for different iterations.



calibrated monochrome stereo cameras and a miniature laser rangefinder. None of these sensors were used in this work, as they as primarily meant to do research in the area of computer vision systems.

A MATLAB/Simulink numerical simulator that recreates the friction-free, double-integrator dynamics and emulates the different on-board sensors and actuators is first used to design and validate the control laws. Once the performance in simulations is satisfactory, the control software is converted into C/C++ with the Simulink Embedded Coder tool, compiled and then executed on the platforms' Raspberry Pi-3 computers.

IV.B. Experimental Results

For the experiment, after both platforms have been positioned arbitrarily on the table by the user, they stay at rest for 20 s to ensure the motion capture system has properly acquired the LED markers' and resolved the position and attitude ground truth data. Then, for the next 40 s, both platforms maneuver through PD control to their desired initial x - y position and attitude. At 60 s into the experiment, the chaser robot (black platform) is commanded to perform a 0.8-m radius, counter-clockwise, circular inspection trajectory around the stationary target object (red platform) in 60 seconds. Throughout the trajectory, the orientation of the chaser spacecraft is controlled with a PD controller such that its vision sensors are always pointing at the target. The chaser maneuvers autonomously via the proposed learning controller by opening the solenoid valves regulating the airflow to its air nozzles. The experiment is repeated 10 times for each tuning of the confidence level factor β_{j+1} . After the completion of each trial, the chaser platform returns to the same initial position, ready to initiate another iteration of the trajectory. A video of a sample iteration can be found at <https://goo.gl/B7Y96s>.

The results obtained from the high-fidelity simulator and from the experimental testbed for $\beta_{j+1} = j/N$ are provided in Figs. 6 and 7, respectively. These figures illustrate the circular inspection trajectory for the first and the tenth iteration (to avoid cluttering the figures), and provide the evolution of the average position error. It is notable that the simulated and experimental results are very similar. As expected, the proposed confidence-based learning controller keeps learning iteratively from its past trials by modifying its learning behaviour from conservative to confident. As a result, the controller iteratively increases its tracking performance over the ten trials without saturating its low-thrust air nozzles. Altogether, these experimental results successfully demonstrate the functionality and performance of the new learning algorithm developed in this paper to practically track fast and demanding trajectories in a safe and efficient manner.



Figure 5. Picture of a SPOT spacecraft platform.

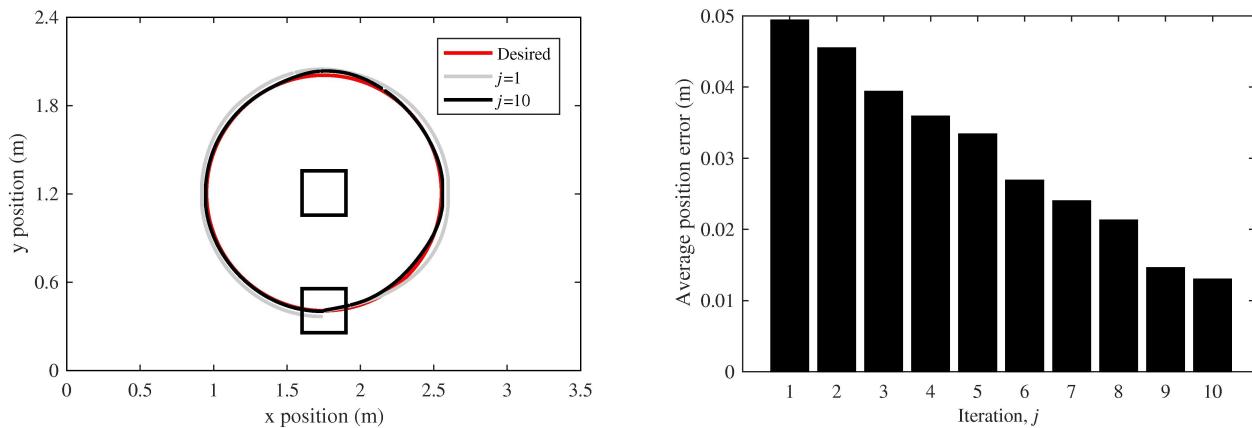


Figure 7. Experimental learning of a 1-m circular inspection trajectory with $\beta_{j+1} = j/N$. The planar trajectory and average position error are depicted for different iterations.

V. Conclusion

In this paper, a new learning algorithm based on confidence level is developed and applied for spacecraft relative trajectory tracking in proximity operations. The new control scheme represents a simplification of well-known current-feedback approach that combines a causal feedback loop to a non-causal learning loop. The algorithm is further modified by introducing a new parameter referred to as the *confidence level factor*. A simple linear update law for this parameter is provided, which allows the learning process to transition from conservative to confident. The result is a control law that carefully improves its tracking performance over multiple iterations, without saturating the actuators. The iterative learning controller is evaluated in numerical simulations for a spacecraft relative motion tracking under dynamics uncertainties. Results suggested that the controller is efficient at rejecting the effects of the repeating perturbations that arise from dynamics modeling errors arising from neglecting orbital mechanics effects in the double-integrator formulation. Two designs of the confidence level factor are also evaluated and demonstrate that the linear update law does not result in actuator saturation, compared to an over-confident approach. Finally, the proposed learning controller is thoroughly studied in experiments at Carleton University's Spacecraft Robotics and Control Laboratory. Experimental results clearly show the ability of the new learning algorithm to practically track fast and demanding trajectories in a safe and efficient manner.

Acknowledgments

This research was financially supported in part by Ontario Graduate Scholarship, and the Natural Sciences and Engineering Research Council of Canada under the Discovery Grant Program and the Canada Graduate Scholarship CGS M-302098-2015. Useful discussions with Kevin L. Moore from the Colorado School of Mines' College of Engineering and Computational Sciences are also gratefully acknowledged.

References

- ¹Clohessy, W. H. and Wiltshire, R. S., "Terminal Guidance System for Satellite Rendezvous," *Journal of the Aerospace Sciences*, Vol. 27, No. 9, 1960, pp. 653–658.
- ²Fehse, W., *Automated Rendezvous and Docking of Spacecraft*, Cambridge University Press, Cambridge, United Kingdom, 2003.
- ³Nolet, S., *Development of a Guidance, Navigation and Control Architecture and validation Process Enabling Autonomous Docking to a Tumbling Satellite*, Ph.D. thesis, Massachusetts Institute of Technology, Department of Aeronautics and Astronautics, Cambridge, MA, 2007.
- ⁴Fezjic, A., *Development of Control and Autonomy Algorithms for Docking to Complex Tumbling Satellites*, Master's thesis, Massachusetts Institute of Technology, Department of Aeronautics and Astronautics, Cambridge, MA, 2008.
- ⁵Virgili-Llop, J., Zagaris, C., Park, H., II, R. Z., and Romano, M., "Experimental Evaluation of Model Predictive Control and Inverse Dynamics Control for Spacecraft Proximity and Docking Maneuvers," *6th International Conference on Astrodynamics Tools and Techniques*, Darmstadt, Germany, 2016.
- ⁶Subbarao, K. and Welsh, S., "Nonlinear Control of Motion Synchronization for Satellite Proximity Operations," *Journal of Guidance, Control, and Dynamics*, Vol. 31, No. 5, 2008, pp. 1284–1294.

⁷Terui, F., "Position and Attitude Control of a Spacecraft by Sliding Mode Control," *American Control Conference*, Inst. of Electrical and Electronics Engineers, 1998.

⁸Mauro, G. D., Schlotterer, M., Theil, S., and Lavagna, M., "Nonlinear Control for Proximity Operations Based on Differential Algebra," *Journal of Guidance, Control, and Dynamics*, Vol. 38, No. 11, 2015, pp. 1273–1287.

⁹Ulrich, S., Saenz-Otero, A., and Barkana, I., "Passivity-Based Adaptive Control of Robotic Spacecraft for Proximity Operations under Uncertainties," *Journal of Guidance, Control, and Dynamics*, Vol. 39, No. 6, 2016, pp. 1444–1453.

¹⁰Arimoto, S., Kawamura, S., and Miyazaki, F., "Better Operation of Robots by Learning," *Journal of Robotic Systems*, Vol. 1, No. 2, 1984, pp. 123–140.

¹¹Bristow, D. A., Tharayil, M., and Alleyne, A. G., "A Survey of Iterative Learning Control," *IEEE Control Systems Magazine*, Vol. 26, No. 3, 2006, pp. 96–114.

¹²Ahn, H.-S., Moore, K. L., and Chen, Y., *Iterative Learning Control: Robustness and Monotonic Convergence for Interval Systems*, Communications and Control Engineering Series, Springer-Verlag, Berlin, 2007.

¹³Porter, B. and Mohamed, S., "Model-based Iterative Learning Control of Space-Shuttle Manipulator," *AIAA Guidance, Navigation, and Control Conference*, Portland, OR, 1990, AIAA paper 1990-3398.

¹⁴Porter, B. and Mohamed, S., "Adaptive Iterative Learning Control of Robotic Manipulators in Task Space," *AIAA Guidance, Navigation, and Control Conference*, New Orleans, LA, 1991, AIAA paper 1991-2789.

¹⁵Wu, B. L., Poh, E. K., Wang, D. W., and Xu, G. Y., "Satellite Formation Keeping via Real-Time Optimal Control and Iterative Learning Control," *IEEE Aerospace Conference*, Inst. of Electrical and Electronics Engineers, 2009.

¹⁶Ahn, H. S., Moore, K. L., and Chen, Y. Q., "Trajectory-keeping in Satellite Formation Flying via Robust Periodic Learning Control," *International Journal of Robust and Nonlinear Control*, Vol. 20, No. 14, 2010, pp. 1655–1666.

¹⁷Xu, J. X. and Qu, Z. H., "Robust Iterative Learning Control for a Class of Nonlinear Systems," *Automatica*, Vol. 34, No. 8, 1998, pp. 983–988.

¹⁸Wu, B., Wang, D., and Poh, E. K., "High Precision Satellite Attitude Tracking Control via Iterative Learning Control," *Journal of Guidance, Control, and Dynamics*, Vol. 38, No. 3, 2015, pp. 528–533.

¹⁹Moore, K. L., *Iterative Learning Control for Deterministic Systems*, Springer-Verlag, London, 1993.

²⁰Bien, Z. and Xu, J.-X., *Iterative Learning Control: Analysis, Design, Integration and Applications*, Kluwer, Boston, 1998.

²¹Chen, Y. and Wen, C., *Iterative Learning Control: Convergence, Robustness, and Applications*, Springer, London, 1999.

²²Ostafew, C. J., Schoellig, A. P., and Barfoot, T. D., "Conservative to Confident: Treating Uncertainty Robustly Within Learning-Based Control," *IEEE International Conference on Robotics and Automation*, Inst. of Electrical and Electronics Engineers, 2015.

²³Schoellig, A. P., Mueller, F. L., and D'Andrea, R., "Optimization-based Iterative Learning for Precise Quadrocopter Trajectory Tracking," *Autonomous Robots*, Vol. 33, No. 1, 2012, pp. 103–127.

²⁴Paluszek, M. and Thomas, S., "Generalized 3D Spacecraft Proximity Path Planning Using A*," *AIAA Infotech@Aerospace*, Arlington, VA, 2005, AIAA paper 2005-7043.