



HAL
open science

A Practical Method to Speed-Up the Experimental Procedure of Iterative Learning Controllers

Oktaý Koçan, Augustin Manecy, Charles Poussot-Vassal

► To cite this version:

Oktaý Koçan, Augustin Manecy, Charles Poussot-Vassal. A Practical Method to Speed-Up the Experimental Procedure of Iterative Learning Controllers. IROS 2018, Oct 2018, Madrid, Spain. pp.6411-6416, <10.1109/IROS.2018.8594025>. <hal-02329295>

HAL Id: hal-02329295

<https://hal.science/hal-02329295v1>

Submitted on 23 Oct 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



HAL Authorization

A Practical Method to Speed-up the Experimental Procedure of Iterative Learning Controllers

Oktaý Koçan¹, Augustin Manecy¹, Charles Poussot-Vassal¹

Abstract—This paper proposes a practical approach for fastening the lengthy experimental processes that may occur with iterative learning control (ILC) upto a certain level using simple low order identified models. The traditional practice in ILC experiments is to update the ILC signal by directly using the experimental data after each run of the process which corresponds to one ILC update per one run. When considered from the point of experimental time, even conducting a moderate number of ILC updates can take quite long with this procedure. Since an accurate linear model can adequately represent the actual system upto a certain amplitude and/or frequency of the desired reference, we propose that the experiment length can be reduced by updating the ILC signal via predicted system data until the limits of the linear model. This approach allows one to carry out large number of ILC updates while not needing to carry out the same amount of real experiments. Consequently, a significant number of experiments that would be needed for achieving the same results can be skipped with a simulation approach. The efficiency of the proposed method was tested through experimentation with three different UAV reference trajectories and the results demonstrated that it is possible to attain significant amount of tracking precision in several flight experiments.

I. INTRODUCTION

One can think of iterative learning control (ILC) as to simply learn from the previous trial errors of a process (e.g. trajectory or movement). The repetitiveness of the process under same initial conditions is a must for ILC since it improves the output tracking by filtering the repeated error. This filtered error is used to change the system input in the following trial in order to reach a better tracking.

An ILC can significantly improve a system's behaviour when it is used in conjunction with traditional control techniques such as feedback control. This is a result of the efficiency of ILC in completing the deficiencies of traditional controllers when it comes to repeating errors and lags in the output. Although a well designed feedback controller can successfully deal with non-repeating disturbances and noise, its control signal does not depend on the repetitiveness of the process and stays the same for each system run which gives the same amount of repeating non-zero tracking error [1]. In such a case, ILC provides with an advantage of detecting the repeating errors between process iterations and eliminating them gradually as iterations continue.

Another reason that makes ILC an appealing method is that it can easily be integrated to an existing system. Unlike adaptive controllers and neural network controllers which modify the control parameters in the system, ILC only modifies the input signal [2]. Therefore, one can simply add the ILC to an already designed closed-loop system by way of an open loop connection.

One can notice that the majority of the recent ILC publications is related to an optimization-based ILC update method even though there exist many other ILC update methods such as P-type, D-type, PD-type, fractional-order, inversion-based ILCs. This preference can be seen as a result of the efficiency and the promissive nature of the optimization based approaches. In general, the optimization based ILC methods are grounded on solving a convex optimization problem that is defined by an objective function to be minimized under some constraint equations. The main goal here can be seen as making the tracking error of the system as small as possible by varying the system input. A sufficient understanding on the application of optimization based ILC methods can be gained through [3]–[7]. Moreover, the Ph.D theses [8] and [9] are quite useful for having a deeper insight on the mathematical theory as well as for reviewing derivations of convergence conditions in general.

The outline of this paper hereafter is as follows: Section II explains the idea behind our method; Section III gives insight to the details of the control architectures we used and designed; Section IV shows the experimental setup including the data flow and initialisation processes; Section V represents our experiment results and analyses; and finally Section VI is dedicated for the conclusions.

II. UAV ORIENTED PROBLEM STATEMENT

A general practice when conducting ILC experiments with UAVs is to perform one ILC signal update per flight of a given trajectory. In other words, each update is done utilizing the experimental data of the previous flight. Although it is straightforward to follow such an update procedure, one can argue about the efficiency of this approach when the number of experiments that are needed to be done to get the stabilizing ILC signal increases. Moreover, the procedure of experimentation with UAVs can become tedious considering the factors such as the time spent for the experimental setup, the length of the chosen flight trajectory, the requirement of battery change and any unexpected problem that may occur along the flight and cause a need of rerun. Under these conditions, even conducting 20 experiments may take long, not to mention the cases in which the number of experiments reaches more than 100 as in [10]. On the other hand, it is usually easy to obtain a linear dynamic model of a UAV for a given trajectory. If the task is to reach a certain tracking performance, using this linear model with ILC can make it much faster to approach the desired performance when ILC is applied in a non-traditional way by using predicted iteration data. The linear model can be utilized to predict the future flight data and to carry out large number of ILC iterations which would then allow one to largely improve the ILC signal without needing to actually perform real flight experiments. What makes this practical approach interesting is that one can

¹DTIS, *Info, Processing and Systems*. ONERA, *The French Aerospace Lab*, Toulouse, France. <first-name>.<lastname>@onera.fr

highly improve the tracking accuracy by using simple linear models such as UAV models identified with a few number of poles and zeros.

To our best knowledge, the practical approach presented in this work was not introduced by other authors in the ILC literature.

III. PHILOSOPHY AND CONTROL DESIGN

In this section, we present the design of our cascade control architecture with an aim of motivating its usage. We assume that operator can give an arbitrary repetitive sequence of position setpoints. As this sequence can be discontinuous and not achievable by the UAV, we first systematically filter this raw sequence by means of a 3rd order filter (see Section III-A). The resulting filtered sequence thus suits our dynamic objectives by becoming continuous and twice differentiable. Following that, a position feedback controller is used to compute the attitude and total thrust references. Thrust of each rotor is then adjusted in order to track the references thanks to an attitude controller and mixer matrix.

The position controllers (one per axis) were designed to achieve robustness objectives rather than to achieve high tracking performances. Consequently, they lead to lags and poor tracking when dynamical position sequences are given by an operator. Considering that the operator's position sequences are repetitive, an ILC algorithm was added to achieve the best possible tracking performances for a given trajectory by compensating for the weak performance of the position feedback controller.

A. Reference trajectory & Pre-filter

As mentioned previously, a pre-filter is used to smooth the raw repetitive trajectories in order to make them continuous along attitude references. This operation allows some time for the UAV to reach the desired position at the beginning of the reference trajectory which is initialized according to the UAV's initial state. Moreover, we implement one pre-filter per axis which allows us to adjust the DC-gains differently and to produce only a small phase delay without having any change in amplitude w.r.t the initial trajectory given by the operator. The pre-filter's transfer function can be described as:

$$\frac{(\cdot)_{ref}(s)}{(\cdot)_{raw.ref}(s)} = \frac{G(\cdot)}{(\tau_{1(\cdot)}s + 1)(\tau_{2(\cdot)}s + 1)(\tau_{3(\cdot)}s + 1)} \quad (1)$$

where (\cdot) can be X , Y or Z , the ' ref ' and ' $raw.ref$ ' signify the filtered and unfiltered references, respectively and $\tau_{1(\cdot)}$, $\tau_{2(\cdot)}$ and $\tau_{3(\cdot)}$ define the time responses which were chosen to be less than 2s. Thus, the first reference trajectory for the UAV is selected as an eight-shaped path at 1 meter altitude with a period of $T = 7.5$ s. The eight-shape is obtained by combining two sinusoidal signals on X and Y -axes with same amplitudes ($A_{X,Y} = 1$ m) but different frequencies; i.e $f_X = 0.5f_Y$.

B. The feedback control

The horizontal and vertical positions of the UAV are each controlled by a position and a speed feedback loops. The controller design was done utilizing an *LQ method* with a linearized model of the UAV with an aim of achieving an overshoot of $< 5\%$ for the position tracking (X, Y, Z), a limited command noise and good robustness properties. The main drawback of

this feedback controller is, as illustrated later, its "slow" time response (see Fig. 4 and 5). It takes approximately 5.5s to reach a reference position, which introduces delay when tracking fast varying trajectories.

In Fig. 1, one can observe the architecture of the position closed-loop system. The *LQ* horizontal controllers deliver the attitude references α_x^* and α_y^* in the *OptiTrack* frame which are then converted in Euler angles references ϕ^* and θ^* . The vertical *LQ* controller generates thrust reference δT^* to which a nominal thrust u_{T0} is added to compensate for the weight of the UAV. Furthermore, Euler angle references and thrust references are tracked by the *PX4* controllers which generate setpoints for the motors (ω_{r_i}). The position and speed of the UAV are then measured by the optitrack system and used by the *LQ* controllers. Hence, the closed-loop system considered by ILC consists of the transfer functions between the position references X^* , Y^* and Z^* that are assigned to position controllers and the measured position outputs \bar{X} , \bar{Y} and \bar{Z} :

$$G_X(s) = \frac{\bar{X}(s)}{X^*(s)}, G_Y(s) = \frac{\bar{Y}(s)}{Y^*(s)}, G_Z(s) = \frac{\bar{Z}(s)}{Z^*(s)}. \quad (2)$$

C. Structure of the used ILC algorithm

A norm-optimal ILC (NO-ILC) was chosen as the ILC update method for the UAV experiments. The method is based on the minimization of a quadratic cost function (J) which may vary with different requirements. In our case, the J is defined in the following form:

$$J(u_{i+1}) = e_{i+1}^T W_e e_{i+1} + u_{i+1}^T W_u u_{i+1} + \lambda[(u_{i+1} - u_i)^T (u_{i+1} - u_i) - \delta], \quad (3)$$

where i is the iteration index of ILC; $e_{i+1} \in \mathbb{R}^N$ is the system's current tracking error; $u_{i+1} \in \mathbb{R}^N$ is the system's current input; $u_i \in \mathbb{R}^N$ is the system's previous input; $\gamma \in \mathbb{R}^1$ is the constraint on the input increment; $\lambda \in \mathbb{R}^1$ is the Lagrange multiplier and $W_e \in \mathbb{R}^{N \times N}$; and $W_u \in \mathbb{R}^{N \times N}$ are the respective weighting matrices (Here, one should assume that N is the length of the vectors e_{i+1} , u_{i+1} and u_i which is determined by the selected simulation and step times).

The NO-ILC that is applied in this paper was designed according to the procedure provided in [8]. In broad strokes, the procedure is based on the analytical solution of a cost function that yields an ILC update equation from which one can extract the structures of the ILC filters $Q \in \mathbb{R}^{N \times N}$ and $L \in \mathbb{R}^{N \times N}$ having the forms given in (5) and (6). The cost function is written by considering the system's current output tracking error (e_{i+1}), current input (u_{i+1}) and current input difference ($u_{i+1} - u_i$). Then the optimization is done via the method of *Lagrange multipliers*. Furthermore, the convergence of the ILC algorithm is determined by the tuning of two scalar parameters: $\rho > 0$ and $\lambda > 0$ which are the weight on the input and the Lagrange multiplier, respectively. They have to be selected in a way that $\|Q\|_2 < 1$ and $\|QL\|_2 \leq 0.5/\sqrt{\rho + \lambda}$. The NO-ILC performance can be seen as a matter of trade-off between these two parameters.

The NO-ILC update equation (4) and filters (5)-(6) are written

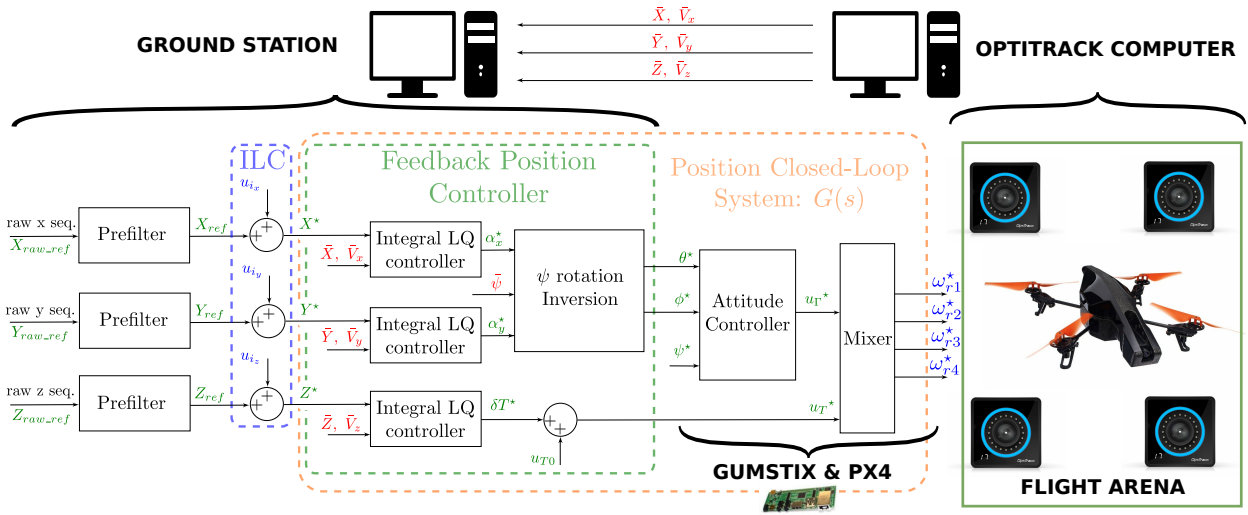


Fig. 1. NO-ILC experimental setup and control architecture.

as given below:

$$u_{i+1} = Qu_i + QLe_i, \quad (4)$$

$$Q = ((\lambda + \rho)I + G^T G)^{-1}(\lambda I + G^T G), \quad (5)$$

$$L = (\lambda I + G^T G)^{-1}G^T, \quad (6)$$

where G is the lifted-matrix (see in [2]) of the closed-loop system's impulse response.

IV. EXPERIMENTAL SETUP

The flight experiments were carried out in a flight arena equipped with *OptiTrack* motion capture system. The UAV that was used during the experiments is a *Parrot AR. Drone 2.0*. As it was previously mentioned, a cascade control scheme was chosen. The fast and robust attitude regulation loop is implemented in the *Pixhawk PX4* autopilot which computes motors' references to track attitude set-points. The guidance and navigation loops are implemented in a more powerful calculator, the *Gumstix*, yielding attitude references to the *textitpixhawk* to track desired position trajectories. The Fig. 1 shows the experimental environment in a schematic way. The specific role of each component can be detailed as below:

- 1) **Ground station:** a Simulink model can both sends high level position reference to the *Gumstix* or execute its own guidance law (position feedback and ILC) and directly send attitude references to the *Gumstix*.
- 2) **Wi-Fi Link:** Wi-Fi has been chosen for the data transfer between the drone, the ILC computer and *OptiTrack* system.
- 3) **Gumstix Board:** It is the bridge between the data received via Wi-Fi and the *PX4* autopilot. The *Gumstix* can run its own guidance law or directly receive attitude commands coming from the ground station.
- 4) **Pixhawk PX4:** joined to the *Gumstix* by a serial link, the *PX4* is the actual autopilot of the drone, where attitude control takes place.

A. Data flow of ILC

The ILC experiments were carried out following a procedure that is different than the traditional way. Normally, one uses the

experimental data from the previous flight in order to obtain the input signal in the next flight and this leads to one ILC update per one experiment. This becomes a problematic issue when one needs to do large number of iterations to reach a desired reference. We propose that it is possible to overcome this problem by using a closed-loop system identification process together with an ILC algorithm. The data flow procedure applied during the experiment is given in Fig. 2. It can be seen that the first flight includes no ILC signal, i.e. the UAV flies with its own internal closed-loop controllers that are adjusted to have low performance. The data from the first flight is used to identify the UAV dynamics in X , Y and Z -axes. Then, ILC makes 300 iterations by predicting the positions for next flight. The signal from the last iteration is then used to modify the reference signal of the system. After applying the same steps for 3 times, this approach reaches its limit and the identification does not improve the trajectory further. At this point, the update process is switched back to the traditional way where the ILC begins directly using the error from the real flights and this process yields one update per flight.

The reason why the proposed procedure can provide a rapid correction of closed-loop lags and errors from model imperfections by means of only simulated ILC iterations in several flights at the beginning is because the reidentification process of the linear UAV model after each flight allows us to obtain a close behaviour to the desired tracking. Basically, what ILC does is to refine the UAV linear model as of the first flight. One can think of each newly performed flight with the current ILC signal as a new operating point which is closer to the desired reference trajectory. Thus, the forthcoming ILC signal needs to include the information of the last operating point to better the tracking and this is the point where the reidentification is made use of. In this way, the simulation based ILC approach improves the tracking until the limits of the linear UAV model. Accordingly, it can be said that when ILC is switched back to the traditional update, the remaining errors are due to the disturbances which cannot be approximated by our linear model.

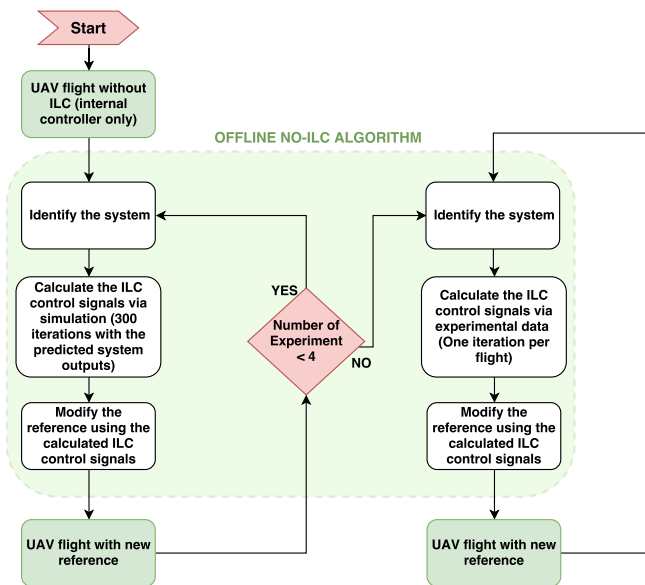


Fig. 2. Data flow procedure for NO-ILC experiment

B. Initialization of the ILC

The initialization of the ILC was done according to the data provided in Table I. The values of ρ and λ were selected with respect to the two convergence conditions given in Section III-C. The low value set for ρ indicates that a high convergence rate is aimed. Besides, identification algorithm is asked for continuous transfer functions of low complexity (3 poles and no zeros) and it can be noticed that the initial UAV state is 'hovering' at 1m altitude.

TABLE I
NO-ILC INITIALIZATION PARAMETERS

Sample time, T_s	0.02 sec.
Simulation time, T_{sim}	37.5 sec.
Discretization method	T_{ustin}
Identified models in X, Y and Z	$n_{poles} = 3; n_{zeros} = 0$
Initial states	$[0 \ 0 \ -1]^T \in \mathbb{R}^{3 \times 1}$
Initial system input	$[x_{ref} \ y_{ref} \ z_{ref}] \in \mathbb{R}^{N \times 3}$
Initial ILC input	$[0 \ 0 \ 0] \in \mathbb{R}^{N \times 3}$
Number of ILC iterations	300
Weight on the error, W_e	$\rho I \in \mathbb{R}^{N \times 3}$
Weight on the system input, W_e	$I \in \mathbb{R}^{N \times 3}$
ρ	0.001
λ	0.1

V. EXPERIMENTAL RESULTS AND ANALYSES

In this section, the tracking results for an eight-shaped reference trajectory are presented first and then the same procedure is extended to two more references to further test the efficiency of the designed ILC procedure. These references consist of a square-shaped trajectory and an inclined elliptical trajectory. All the flight experiments were conducted with respect to the data flow and the initialization procedures given in Section IV-B except the readjustments applied in the simulations time and the initial UAV altitude.

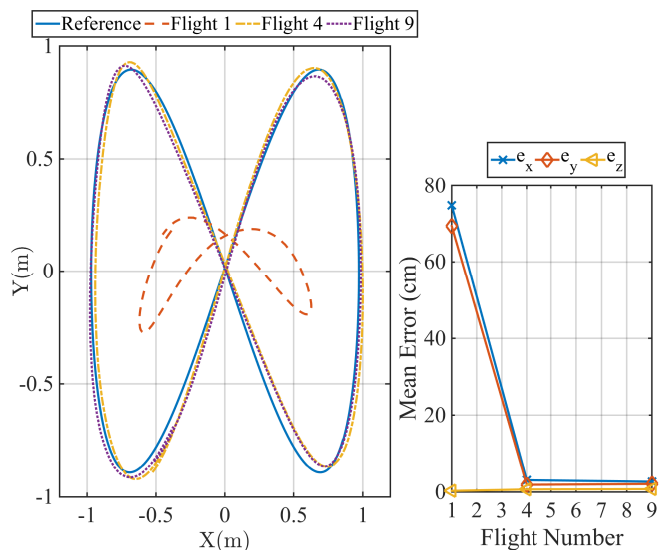


Fig. 3. Tracking results for the eight-shaped reference

A. Eight-shaped trajectory tracking results

For the eight-shaped trajectory, the experimentation was done until the ninth flight considering the fact that the tracking precision was not improving further. The tracking results obtained with the NO-ILC algorithm are depicted in Fig. 3. The ILC update for the first 4 flights are based on the predicted data obtained through the identified models of the system. On the other hand, the ILC used directly the real flight data for the remaining 5 flights. It can be observed that the reference tracking almost does not improve as of the fifth flight. This is due to the fact that it was possible to reach the maximum tracking performance with ILC using the identified system models by the end of the fourth flight. Moreover, one can also see in Fig. 3 how the mean position error in each axis changed along the flight experiments. It is straightforward to conclude that a sufficiently accurate tracking performance can be obtained in only 4 flight experiments.

On the identified transfer functions: After each flight, we identify the UAV transfer functions on each axis. Since the translational speeds, and the attitude angles become more aggressive as the ILC improves the tracking, the identified linear transfer functions are slightly different after each flight. This simply highlights the utility of identifying a new transfer function after each flight. As shown in Fig. 4, the identified transfer function of the X-axis evolves only for a small amount along the flights whereas the differences are relatively larger on the Y-axis (Fig. 5). This can be found reasonable since it is harder to follow the trajectory on the Y-axis that is of a frequency twice as the one on X-axis.

B. Amplitude analysis for eight-shaped trajectory tracking

After obtaining a satisfactory tracking performance for an eight-shaped trajectory with the characteristics given in Section III-A an amplitude analysis was performed in order to find the limitation of the used ILC algorithm. In different words, the same procedure was applied using the reference trajectories with higher amplitudes but the same frequency as before. In this process, a simulator of the UAV was used instead of carrying

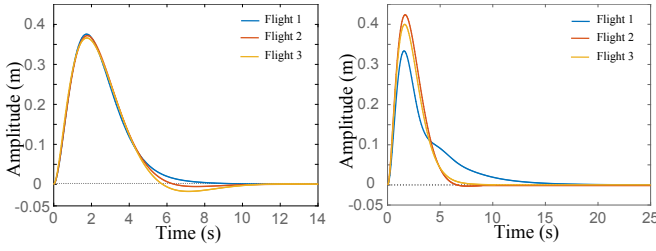


Fig. 4. Evolution of the impulse response of the identified transfer function of the X-axis with the flights

out real flight experiments. Three simulations were shown for each amplitude considering the fact that the simulator output became invariant after three flight simulations. The results of this amplitude analysis were given on Table II in terms of the absolute values of the mean and maximum position errors in each axis. Looking at the error values in general, one may observe that increments in reference amplitude cause larger mean and maximum errors. Until $A_4 = 1.75\text{m}$, the errors in Z -axis show a different fashion along the runs; i.e. they increase whereas the errors in X and Y -axes decrease as aimed. Then, at $A_5 = 2.00\text{m}$, the mean errors in X and Y -axes first decrease and then slightly increase while the errors in Z -axis keep the same trend. The issue observed in Z -axis can be seen as an outcome of higher pitch and roll angles that were taken by the UAV for improving the accuracy of the lateral trajectory which accordingly caused a compromise of altitude accuracy. Hence, one can say that the ILC algorithm for the eight-shaped trajectory starts to become unsatisfactory as of $A_4 = 1.75\text{m}$.

TABLE II
MEAN POSITION ERROR VS. REFERENCE AMPLITUDE (IN CM)

Frequency, $f = 0.1333\text{Hz}$									
Reference Amplitude	Mean error	Run 1	Run 2	Run 3	Max. error	Run 1	Run 2	Run 3	
$A_1 = 1.00$	\bar{e}_x	81.98	3.366	3.400	$e_{x\max}$	128.6	6.192	5.683	
	\bar{e}_y	80.29	3.504	1.292	$e_{y\max}$	126.3	7.831	3.363	
	\bar{e}_z	0.091	1.235	1.357	$e_{z\max}$	0.181	2.127	2.363	
$A_2 = 1.25$	\bar{e}_x	102.7	4.082	4.084	$e_{x\max}$	161.3	7.687	6.985	
	\bar{e}_y	100.3	5.114	1.773	$e_{y\max}$	157.8	11.14	4.770	
	\bar{e}_z	0.142	2.001	2.260	$e_{z\max}$	0.284	3.435	3.864	
$A_3 = 1.50$	\bar{e}_x	123.4	4.757	4.678	$e_{x\max}$	193.8	9.166	8.150	
	\bar{e}_y	120.2	6.899	2.363	$e_{y\max}$	189.3	14.69	6.361	
	\bar{e}_z	0.204	3.039	3.582	$e_{z\max}$	0.409	5.220	6.062	
$A_4 = 1.75$	\bar{e}_x	144.3	5.399	5.140	$e_{x\max}$	226.4	10.67	9.455	
	\bar{e}_y	140.1	8.819	3.122	$e_{y\max}$	220.7	18.34	7.897	
	\bar{e}_z	0.277	4.405	5.244	$e_{z\max}$	0.557	7.669	9.288	
$A_5 = 2.00$	\bar{e}_x	165.2	5.907	8.713	$e_{x\max}$	259.0	12.49	14.91	
	\bar{e}_y	160.0	10.26	15.26	$e_{y\max}$	252.2	21.24	32.30	
	\bar{e}_z	0.362	6.503	14.27	$e_{z\max}$	0.729	11.41	27.33	

C. Square-shaped trajectory

A square-shaped trajectory is much harder to follow with precision compared to the eight-shaped trajectory if the UAV is required to fly continuously without stopping at the corners of the square. In different words, at the corners of the square

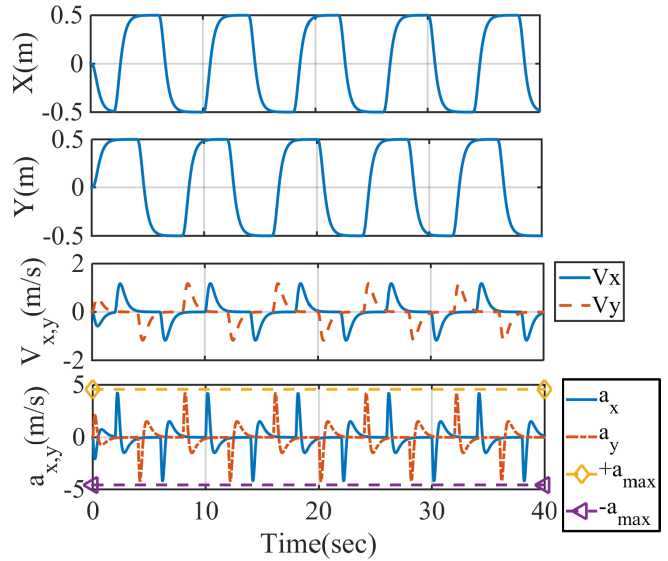


Fig. 6. Filtered square reference trajectory

the UAV is asked to perform a sudden change in its direction of movement which is a strenuous control task. Moreover, a prefiltering process similar to the one in Section III-B was also implemented on the square trajectory before beginning the ILC experimentation. Thus, the reference trajectory was made more real-like from the point of UAV dynamics and the thrust level could be kept below the limit of saturation. Fig. 6 depicts the filtered square reference positions for X and Y axes as well as the corresponding velocity and acceleration curves. One can observe that the demanded acceleration from the UAV was kept within a range defined by a limit of $\pm 25^\circ$ in pitch and roll angles. In order to avoid saturations at sharp corner turns, this limit was made more strict compared to the one set for the eight-shaped reference, i.e. $\pm 35^\circ$.

The square trajectory has a period of 8s and it was simulated for 40s which corresponds to 5 loops. Fig. 7 demonstrates the evolution of the reference tracking according to the predefined data flow procedure and the plotted data represents only the fifth loop of the UAV's trajectory. It can be figured out that the most of the improvement in the tracking was obtained during the first 4 flights where the ILC algorithm used the predicted flight data. The rest of the flights were carried out with the real flight data using the traditional ILC update process and the precision of the tracking could be slightly improved.

D. Elliptical inclined trajectory

The final trajectory that was analysed is an ellipse with 0.8m semi-major axis, 0.4m semi-minor axis and 15° inclination. The altitude of the UAV was initially set to 1.5m and the UAV starts its flight from the center of the ellipse. Furthermore, the period of the ellipse and the flight duration were chosen to be 4s and 20s, respectively.

Due to the inclination of the reference, the UAV needs to allocate some of its total allowable thrust for altitude increment which limits the reachable accelerations for the maneuvers on the lateral plane. Accordingly, the reference was filtered in order to be sure that the demanded manoeuvres stay below the saturation limits. Fig. 8 demonstrates the positions, velocities and

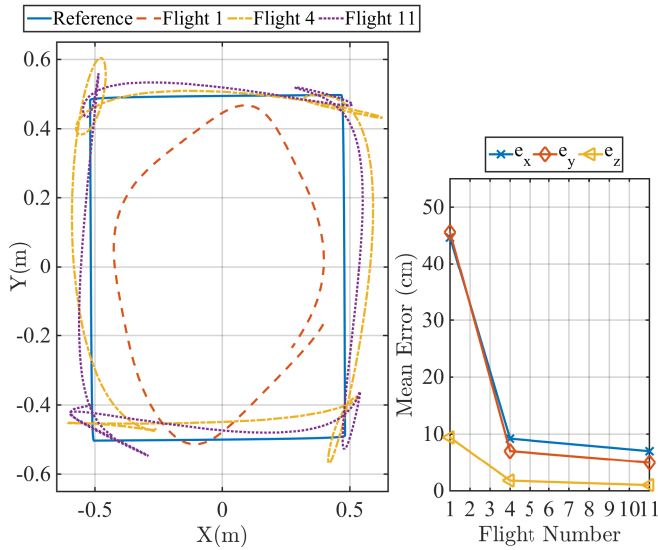


Fig. 7. Tracking results for the square-shaped reference

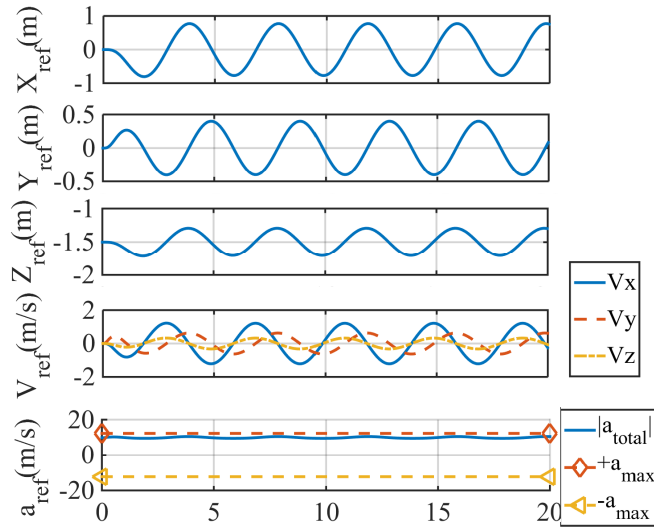


Fig. 8. Filtered elliptical reference trajectory

accelerations for the filtered reference trajectory where one can also observe that the norm of the vector compound of the accelerations in each axis remains between a certain acceleration range. The absolute value of the maximum allowable acceleration of the UAV is 1.25 times the gravitational acceleration, i.e. 12.26m/s^2 . The evolution of the reference tracking is demonstrated in Fig. 9. The experiments were carried out according to the same data flow procedure used before and the represented data correspond to the 5th loop of the trajectory. Similar to the results of other reference trajectories, the largest improvement in tracking was seen to be during the first four flights where ILC used predicted data. Moreover, the seven flights performed afterwards using traditional update could only reduce a small amount of the remaining error.

VI. CONCLUSION

In this paper, we showed a practical procedure that can be utilized to reduce the time spent during ILC experiments. Our

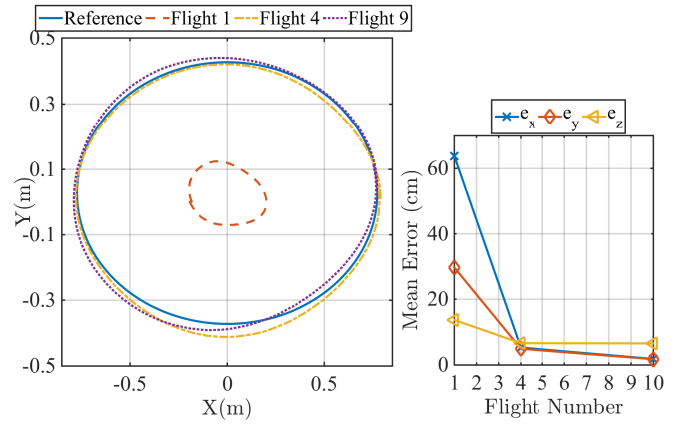


Fig. 9. Tracking results for the elliptical reference

method is based on using predicted flight data when performing the ILC updates. The method showed to be efficient despite the low order linear UAV models used in the update process. The flight experiments with three different trajectories demonstrated that a large amount of tracking performance can be acquired in only 4 flight experiments. Furthermore, using the proposed ILC procedure it was possible to achieve almost the largest agility for our UAV which has already a very limited dynamic performance.

ACKNOWLEDGMENT

We would like to thank Angélique Beaucamp for her assistance at the beginning phase of the experiments. In addition, we thank François Defay for letting us use the flight arena of *ISAE-SUPAERO*.

REFERENCES

- [1] D. Owens and J. Hatonen, "Iterative learning control - an optimization paradigm," *Annual Reviews in Control*, vol. 29, p. 5770, 2005.
- [2] D. Bristow, M. Tharayil, and A. Alleyne, "A survey of iterative learning control: A learning-based method for high-performance tracking control," *IEEE Control Systems Magazine*, vol. 26, no. 3, pp. 96–114, Mar. 2006.
- [3] S. Gunnarsson and M. Norrlof, "On the design of ilc algorithms using optimization," *Automatica*, vol. 37, pp. 2011–2016, May 2001.
- [4] —, "Some aspects of an optimization approach to iterative learning control," in *Proceedings of the 38th Conference on Decision & Control*, Phoenix, Arizona, USA, Dec. 1999, pp. 1581–1586.
- [5] N. Amann, D. Owens, and E. Rogers, "Iterative learning control for discrete time systems with exponential rate of convergence," Centre for Systems and Control Engineering - University of Exeter, Tech. Rep. 95/14, Aug. 1995.
- [6] Y. Chen and K. Moore, "An optimal design of pd-type iterative learning control with monotonic convergence," in *Proceedings of the 2002 IEEE International Symposium on Intelligent Control*, Vancouver, Canada, Oct. 2002, pp. 55–60.
- [7] M. Togai and O. Yamano, "Analysis and design of an optimal learning control scheme for industrial robots: A discrete system approach," in *Proceedings of 24th Conference on Decision and Control*, Ft. Lauderdale, FL, Dec. 1985, pp. 1399–1404.
- [8] M. Norrlof, "Iterative learning control: Analysis, design, and experiments," Ph.D. dissertation, University of Linköping, 2000.
- [9] J. Hatonen, "Issues of algebra and optimality in iterative learning control," Ph.D. dissertation, University of Oulu, 2014.
- [10] S. Lupashin, A. Schollig, M. Sherback, and R. Andrea, "A simple learning strategy for high-speed quadcopter multi-flips," in *IEEE International Conference on Robotics and Automation*, Anchorage, Alaska, USA, May 2010, pp. 1642–1648.