Improving the Accuracy of Industrial Robots via Iterative Reference Trajectory Modification

Luigi Biagiotti, *Member, IEEE,* Lorenzo Moriello, *Member, IEEE,* Claudio Melchiorri, *Member, IEEE,*

***Abstract*—In this paper, a novel repetitive control scheme is presented and discussed. The general framework is the control of repetitive tasks of robotic systems or, more in general, of automatic machines. The key idea of the proposed scheme consists in modifying the reference trajectory provided to the plant in order to compensate for external loads or unmodeled dynamics that cyclically affect it. By exploiting the fact that uniform B-spline trajectories can be generated by means of dynamic filters, the trajectory planning phase has been integrated within a repetitive control scheme able to modify in real-time the reference signal in order to nullify the tracking errors occurring at the desired via-points. Because of this mechanism, the control scheme is very suitable for the application to industrial plants with off-the-shelf, unmodifiable controllers. Experimental results obtained with a standard industrial manipulator both in joint- and in work-space show the effectiveness of the proposed method.**

***Index Terms*—Repetitive Control, Learning Algorithms, Iterative Methods, Robotic Manipulators, B-spline Trajectories.**

I. Introduction

In practical applications, desired tasks are often repetitive or cyclic in nature. This is particularly true in industrial robotics and in automatic machines, where many tasks simply imply the continuous repetition of a given path. From a control point of view, it is therefore required to track and/or reject a periodic exogenous signal that can be considered known since it refers to planned trajectories or disturbances whose cycle time is easily measurable or known in advance. In order to improve the tracking accuracy, Repetitive Control (RC) represents a simple and effective method, since it aims at cancelling tracking errors over repetitions, by learning from previous iterations. RC was first developed by Inoue et al. [1], [2] to improve the control of the power supply in a proton synchrotron accelerator, but soon was applied to many other different systems. Many surveys, see e.g. [3], [4], report the successful use of RC in a number of applications, such as high accuracy trajectory tracking of servomechanism, torque vibration suppression in motors, noise cancellation in power supply, industrial robotics, and so on.

In this paper, a novel repetitive control scheme is presented. The scheme is based on a proper modification of the reference trajectory for the plant, which is supposed to be already controlled by an off-the-shelf controller. A similar idea has been already proposed in the continuous-time domain in [5], where a two-degrees-of-freedom local control, and a plug-in RC is used to update the reference trajectory. The novelty of this paper lies in the assumption that the reference trajectories are defined by spline functions, which are de-facto the standard tool used in the industrial field for planning complex motions interpolating a set of given via-points [6]. Thanks to the possibility of generating B-spline trajectories by means of dynamic filters [7], in the seminal works [8] and [9] the trajectory planner has been inserted inside a control loop that modifies in real-time the control points of the B-spline curve so that the tracking error at the desired via-points converges to zero. The mechanism for the control points modification induces a discrete-time repetitive control acting on the plant, along with the trajectory generator, which works at a very low rate. Therefore, the proposed control scheme, which has been directly developed in the discrete-time domain, is characterized by a very low computational complexity. Moreover, the application of this control scheme is independent on the particular control law of the plant, which is seen as a servo-system able to track a spline curve. As a consequence, the proposed approach can be easily implemented as an outer feedback control loop, that provides a reference input for the robotic system, able to reduce the error at the given via-points below the level which is already guaranteed by the standard robot controller. The implementation of the proposed method is also supported by very weak stability conditions that, for a given plant, only depend on the duration of the trajectory and that, de facto, are always met in case of position-controlled robot manipulators and more generally in position-controlled electro-mechanical systems.

In recent years, the use of B-spline functions combined with learning mechanisms, such as RC and ILC (Iterative Learning Control) and similar, has been widely adopted with the purpose of reducing the complexity of the resulting controller and increasing the robustness of the system. Note that B-spline functions are a particular case of basis functions, introduced into the ILC to reduce the dimensionality of the input/output spaces of the controlled plant, by describing command and output signals as a linear combination of a relatively small number of these basis functions [10]. The functions depend on the original reference signal and on the system dynamics, or are user-defined; for instance, in [11], [12], the basis functions are chosen starting from the plant dynamics; in [13], [14], the derivatives of the reference trajectory (position, velocity, acceleration, etc.) are considered; in [15], the basis functions are defined as Legendre polynomials. B-spline functions are another notable example of user-defined functions. They have

L. Biagiotti is with the Department of Engineering "Enzo Ferrari", University of Modena and Reggio Emilia, Strada Vignolese 905, 41125 Modena, Italy, e-mail: luigi.biagiotti@unimore.it.

C. Melchiorri and L. Moriello are with the Department of Electrical, Electronic and Information Engineering "Guglielmo Marconi", University of Bologna, Viale Risorgimento 2, 40136 Bologna, Italy, e-mail: {claudio.melchiorri, lorenzo.moriello}@unibo.it.

been used in [16], [17] to parameterize a feedforward control term for improving the position control of robotic manipulators. The weights defining the spline are adapted on the basis of the tracking error with a learning algorithm, leading to the control scheme called Desired Compensation Learning Law (DCLL) [18]. In [19], the output trajectory of the plant, in this case a nanomanipulation system, is decomposed in a number of shaping primitives represented by B-spline functions and then an iterative learning algorithm is applied in order to determine the control input associated to each primitive. In [20], a spline function is used to parameterize a signal which is added to the reference input for a servo drive and its parameters are determined by means of a particle swarm based repetitive compensator with the purpose of reducing the tracking error from one iteration to the next one. Finally, the so called B-Spline Networks (BSN), which are a particular implementation of Neural Networks that utilize B-spline basis functions to store information, are combined with iterative learning mechanism to improve the tracking performances of mobile robots [21], inverters [22], [23], piezoelectric actuators [24], linear reciprocating vapor compressors [25], etc.

With respect to these methods, in which the B-spline functions are a way to describe in a concise manner the control signal, in the proposed approach the B-splines represent the form of the reference signals to be tracked, and therefore, on the one hand, they are subject to a number of constraints descending from the specific application, such as the order, the number of control points and the duration of the knot spans, but, on the other hand, they have a clear geometrical meaning, especially when they are defined in the robot workspace.

Moreover, in the works cited above the order of the spline is generally limited to small values because of the computational burden due to the B-spline evaluation process while the high efficiency of the overall algorithm, due to the use of dynamic filters for B-spline generation, is one of the most important feature of the proposed approach.

The paper is organized as follows. In Sec. II a general overview of the filters for B-spline generation is given both in the continuous- and in the discrete-time domain. Then, in Sec. III the proposed RC approach, based on B-spline filters, is illustrated and a convergence analysis is provided. This part of the paper incorporates the contributions of the conference papers [8], [9], with corrections and additions. In particular, the effects of the proposed RC scheme on the trajectory provided to the robot are further analyzed according to an ILC fashion, in which the index denoting the current repetition is explicitly reported. In fact, as often mentioned in the literature (see [4], [26] among many others) Repetitive Control and the Iterative Learning Control are nearly equivalent design philosophies apart the settings of the initial conditions for each trial. In our case, the RC formalism leads to a simpler control scheme and to a straightforward stability analysis but the ILC perspective provides a very intuitive insight into the behavior of the proposed control method.

Finally, the main contribution of this work concerns the experimental validation of the proposed controller on a multi degrees-of-freedom industrial manipulator both in the joint-space and in the workspace (Sec. IV). In particular, in order to take into account the trajectory in the workspace, the RC scheme has been adapted with respect to the basic scheme by inserting the inverse kinematics function between the filter for B-spline trajectory generation and the robot, controlled at the joint level.

Final conclusions are reported in Sec. V.

II. B-spline curves and B-spline filters for Set-point generation

In a number of practical applications the reference signal for dynamical systems is defined by using spline functions that interpolate a set of desired via-points $q_i^\star$, $i = 0, \dots, n-1$ at time instants t_i. By assuming a B-spline form of the trajectory, i.e.

$$q(t) = \sum_{i=0}^{n-1} p_i \, B_i^d(t), \qquad t_0 \leq t \leq t_{n-1} \tag{1}$$

where $B_i^d(t)$ is a B-spline basis function of degree d, the control points p_i must be computed by imposing interpolation conditions on the given data points $q_i^\star$, see [6]. Note that, as shown in Fig. 1, the control points alone determine the geometric shape of the B-spline curve, which represents a sort of smooth approximation of the so-called control polygon.

A. B-spline evaluation

In order to evaluate the B-spline (1) for a given value $t \in [t_0, t_{n-1}]$ it is necessary to compute the basis functions $B_i^d(t)$ via numerical procedures which are usually based on recursion. In [7] uniform B-spline trajectories, i.e. B-splines characterized by an equally-spaced distribution of the knots t_i i.e. $t_{i+1} - t_i = T \;\; i = 0, \dots n-2$, have been generated by feeding a chain of d (continuous-time) dynamic filters defined as

$$M(s) = \frac{1 - e^{-sT}}{Ts}$$

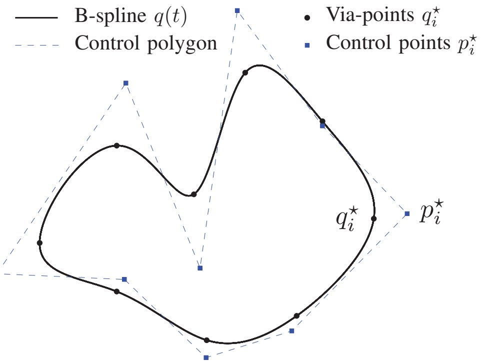


Fig. 1. Two-dimensional B-spline trajectory interpolating a set of via-points $q_i^\star$.

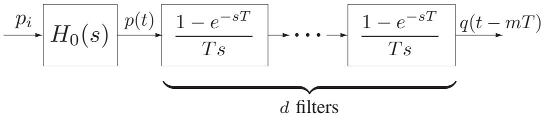


Fig. 2. System composed by d mean filters and by a zero-order hold $H_0(s)$ for the computation of continuous-time B-spline trajectories of degree d.

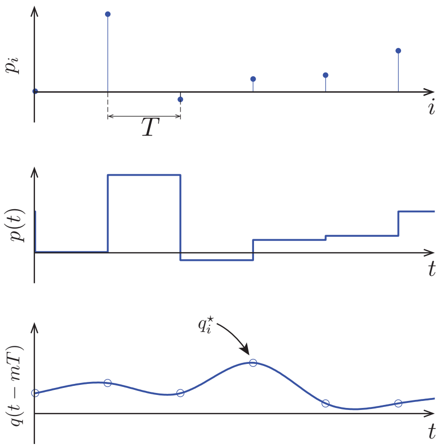


Fig. 3. Control points sequence p_i defining a cubic B-spline and related reference trajectory $q(t-mT)$ obtained with the dynamic filter of Fig. 4 with $d=3$ (and accordingly $m=2$).

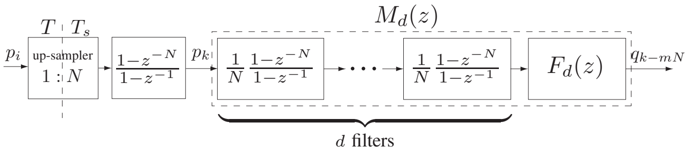

$M_d(z)$
T
T_s
p_i
up-sampler
$1:N$
$\frac{1-z^{-N}}{1-z^{-1}}$
p_k
$\frac{1}{N}\frac{1-z^{-N}}{1-z^{-1}}$
$\frac{1}{N}\frac{1-z^{-N}}{1-z^{-1}}$
$F_d(z)$
q_{k-mN}
d filters

Fig. 4. System $M_d(z)$ composed by d moving average filters and by the FIR filter $F_d(z)$ defined in (2) for the generation of discrete-time B-spline trajectories of degree d.

with the staircase signal $p(t)$ obtained by maintaining the value of each control point p_i for the entire period $iT \leq t < (i{+}1)T$. See the scheme of Fig. 2 and the signals shown in Fig. 3, where the generation of a cubic B-spline is considered. Note that $p(t)$ is obtained by applying a zero-order hold to the train of impulses of amplitude p_i. Moreover, it is worth noticing that the output trajectory is delayed with respect to the application of control points of $m\,T$ seconds, where $m=\frac{d+1}{2}$. For computer controlled systems equipped with digital controllers with sampling period T_s, the B-spline reference trajectory must be computed at time-instants kT_s. It is therefore necessary to discretize the filter of Fig. 2. By z-transforming the chain of d filters $M(s)$ with a zero-order hold the system of Fig. 4 is obtained. Besides the initial zero-order hold, the discrete-time filter $M_d(z)$ for the evaluation of B-splines of degree d is composed by d moving average filters and by the additional FIR filter $F_d(z)$ defined as

$$F_d(z) = \frac{z^{-1}Q_{d-1}(z)}{d!}, \tag{2}$$

with the polynomial

$$Q_r(z) = c_{r,0} + c_{r,1}z^{-1} + \ldots + c_{r,r-1}z^{-(r-1)} + c_{r,r}z^{-r} \tag{3}$$

whose coefficients can be computed in a recursive way as

$$\begin{aligned} c_{r,0} &= c_{r,r} = 1 \\ c_{r,r-i} &= c_{r,i} = c_{r-1,r-i-1}\cdot(i+1) + c_{r-1,r-i}\cdot(r-i+1), \\ & \qquad i = 1,\ldots,\left[\tfrac{r}{2}\right] \end{aligned}$$

being $[\cdot]$ the integer part operator.

In Tab. I the expression of the FIR filter $F_d(z)$ is reported for several values of the B-spline degree d. Note that filter $F_d(z)$ can be written as $F_d(z) = z^{-m}\,\tilde{F}_d(z)$ where $\tilde{F}_d(z)$ is a zero-phase filter, i.e. a filter characterized by $\arg\{\tilde{F}_d(e^{j\omega T})\} = 0$, $\forall\,\omega \in [0,\infty)$.

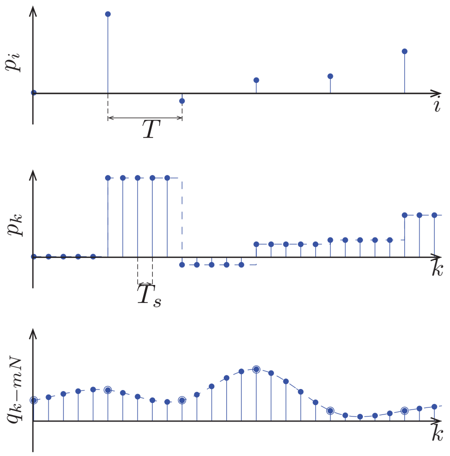


Fig. 5. Control points sequence p_i defining a cubic B-spline and related reference trajectory q_{k-mN} with $m=2$ obtained with the dynamic filter of Fig. 4.

The sequence p_i of the control points is transformed in the staircase sequence p_k, with sampling time T_s, by means of an upsampling operation with replication

$$p_k = p_i, \qquad k = iN, iN+1, \ldots, (i+1)N-1 \tag{4}$$

where N denotes the ratio, supposed to be an integer, between T and T_s. The samples of the B-spline sequence are then generated by the filter denoted by $M_d(z)$ and coincide with the value of the continuous-time trajectory at time instants kT, i.e. $q_k = q(kT)$, see Fig. 5.

In the multi-dimensional case, that is with vectorial control points, the spline curve can be evaluated by considering a filter like the one in Fig. 4 for each component of the vector p_i.

B. Control points computation

The control points p_i are computed by imposing the interpolation conditions on the via-points at the time-instants defined by knots which for uniform B-spline are multiple of the fundamental period T, i.e.

$$q(iT) = q_i^\star, \qquad i = 0,\ldots,n-1. \tag{5}$$

$F_1(z) =$	z^{-1}
$F_2(z) =$	$\frac{1}{2}z^{-1} + \frac{1}{2}z^{-2}$
$F_3(z) =$	$\frac{1}{6}z^{-1} + \frac{4}{6}z^{-2} + \frac{1}{6}z^{-3}$
$F_4(z) =$	$\frac{1}{24}z^{-1} + \frac{11}{24}z^{-2} + \frac{11}{24}z^{-3} + \frac{1}{24}z^{-4}$
$F_5(z) =$	$\frac{1}{120}z^{-1} + \frac{26}{120}z^{-2} + \frac{66}{120}z^{-3} + \frac{26}{120}z^{-4} + \frac{1}{120}z^{-5}$

TABLE I
EXPRESSION OF THE FILTER $F_d(z)$ FOR DIFFERENT VALUES OF d.

If for instance cubic B-splines are considered, (5) can be written as

$$q(iT) = \frac{1}{6}p^\star_{i-1} + \frac{4}{6}p^\star_i + \frac{1}{6}p^\star_{i+1} = q^\star_i, \quad i = 0, \dots, n-1. \quad (6)$$

See [7] for more details. Since it is supposed that the motion is repetitive, the so-called periodic splines must be considered, i.e. spline functions characterized by the boundary conditions

$$q^{(j)}(t_0) = q^{(j)}(t_{n-1}), \qquad j = 1, \dots, d-1. \qquad (7)$$

The conditions (6) and (7) lead to the linear system

$$\begin{bmatrix} 4 & 1 & 0 & & & \cdots & 0 & 1 \\ 1 & 4 & 1 & 0 & & \cdots & & 0 \\ 0 & 1 & 4 & 1 & 0 & \cdots & & 0 \\ \vdots & & & \ddots & & & & \vdots \\ 0 & & \cdots & 0 & 1 & 4 & 1 & 0 \\ 0 & & \cdots & & 0 & 1 & 4 & 1 \\ 1 & 0 & \cdots & & & 0 & 1 & 4 \end{bmatrix} \begin{bmatrix} p^\star_0 \\ p^\star_1 \\ p^\star_2 \\ \vdots \\ p^\star_{n-3} \\ p^\star_{n-2} \\ p^\star_{n-1} \end{bmatrix} = \begin{bmatrix} 6\,q^\star_0 \\ 6\,q^\star_1 \\ 6\,q^\star_2 \\ \vdots \\ 6\,q^\star_{n-3} \\ 6\,q^\star_{n-2} \\ 6\,q^\star_{n-1} \end{bmatrix} \qquad (8)$$

whose solution provides the sequence of control points $p^\star_i$ defining the interpolating B-spline. As well-known the definition of the interpolating B-spline is a global problem, that can be performed only when the entire set of via-points is provided. However, it is possible to approximate this global mapping between via-points $q^\star_i$ and control points $p^\star_i$ within a smaller set of data. In fact, the relation (6) between $q^\star_i$ and $p^\star_i$ can be written as a (discrete-time) dynamic system[1], e.g.

$$\frac{P(z_N)}{Q(z_N)} = \frac{6}{z_N + 4 + z_N^{-1}} \qquad (9)$$

for cubic B-splines. Although (9) represents an unstable system, it can be used for computing the sequence $p^\star_i$ from $q^\star_i$ by approximating its impulse response with a FIR filter defined by

$$H(z_N) = \sum_{n=-r}^{r} h(n)\; z_N^{-n} \qquad (10)$$

with the coefficients $h(n)$ that for $d = 3$ can be computed as

$$h(n) = \frac{1-\alpha}{1+\alpha}\alpha^{|n|} \qquad (11)$$

where $\alpha = -2 + \sqrt{3}$ is the stable pole of (9). Note that the value of $h(n)$ becomes extremely small as $|n|$ grows. This means that for the computation of the control points $p^\star_k$, only the weights of the via-points close to $q^\star_k$ are important, while, from a practical point of view, the others can be neglected with consequent small approximation errors. For instance, the choice $r = 4$ guarantees an approximation error with respect to the exact solution of (8) smaller than 0.5% (for more details see [8], [27]). It is worth noticing that from (11) it descends that $h(n) = h(-n)$, therefore $H(z_N)$ is a zero-phase filter. Moreover, the filter $H(z_N)$ is not causal and it is necessary to introduce a delay equal to r to make it feasible, that is

$$H'(z_N) = z_N^{-r}\, H(z_N) = \sum_{n=0}^{2r} h(n-r)\; z_N^{-n}. \qquad (12)$$

By feeding the filter $H'(z_N)$ with the ordered sequence of via-points $q^\star_i$ it is possible to obtain online the control points $p^\star_i$ defining the interpolating B-spline at the price of a r samples delay and a small approximation error.

Even if cubic B-splines have been used throughout the paper, the expressions of the algebraic system and of the FIR filter $H(z_N)$ for off-line and online computation of the control points defining quintic B-splines, i.e. $d = 5$, have been reported in the appendix because of their importance for the applications.

III. Tracking of B-spline curves and asymptotic perfect tracking via Iterative Learning modification of the control points

The reference trajectory generated by the discrete B-spline filter is then provided to the plant, as illustrated in Fig. 6. Since this scheme has a standard feedforward cascade structure without feedback control actions, with the only purpose of generating arbitrarily complex trajectories for the plant $G(z)$, the capabilities of $G(z)$ to track such inputs are implicitly assumed. Therefore, the system $G(z)$ is assumed to be a controlled plant, with a standard closed-loop structure, whose frequency response is characterized by a typical low-pass behavior with a static gain as close as possible to the unity. In order to follow the input signal accurately, the bandwidth of $G(z)$ must be large enough [28], and in particular larger than the maximum spectral components of the reference input.

A great advantage of using the linear filter $M_d(z)$ for generating B-spline curves consists in the straightforward spectral characterization of the resulting trajectory. The magnitude of the frequency response of the discrete-time B-spline generator of Fig. 4, including the initial interpolator,

$$\tilde{M}_d(z) = \frac{1 - z^{-N}}{1 - z^{-1}} M_d(z),$$

is given by

$$\tilde{M}_d(e^{j\omega T_s}) = \tilde{F}_d(e^{j\omega T_s}) \left[\frac{\text{sinc}\left(\frac{\omega}{\omega_0}\right)}{\text{sinc}\left(\frac{\omega}{\omega_s}\right)}\right]^{d+1} e^{-j\omega\, mT}, \qquad \omega \leq \frac{\omega_s}{2}$$

where sinc$(\cdot)$ denotes the normalized sinc function defined as $\text{sinc}(x) = \frac{\sin(\pi x)}{\pi x}$ and $\omega_0 = \frac{2\pi}{T}$, $\omega_s = \frac{2\pi}{T_s}$. Note that $\tilde{M}_d(e^{j\omega T_s})$ is characterized by a pure delay of mT seconds. The zero-phase FIR filter $\tilde{F}_d(e^{j\omega T_s})$ has a standard low-pass behavior, therefore $\tilde{M}_d(e^{j\omega T_s})$ is a low-pass filter as well and its magnitude decreases rather quickly as ω grows, especially for high values of d. In Fig. 7 the magnitude of the frequency response of the B-spline filter $\tilde{M}_d(z)$ is shown for $d = 1, 3, 5$.

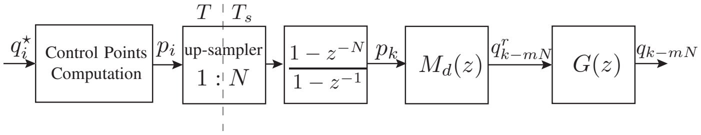


Fig. 6. Set-point definition by means of a B-spline filter for a (controlled) discrete-time plant $G(z)$.

[1] The expressions depending on z_N are referred to the sampling time T = $N\,T_s$.

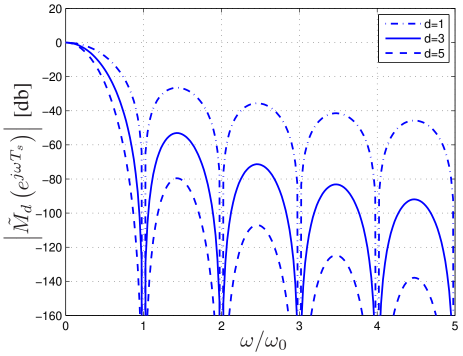


Fig. 7. Magnitude of the frequency response of the B-spline filter $\tilde{M}_d(z)$ for $d = 1, 2, 3$ ($N = 50$).

In case of cubic or quintic B-splines, that are rather standard in the robotic field, the spectrum components of the reference trajectory q^r_{k-mN} at the output of this filter are significant only in the frequency range $[0,\, \omega_0]$, while the reduction of the components for $\omega > \omega_0$ is at least of two order of magnitude (-40 db). As a rule of thumb, in order to obtain a good tracking performance the controlled plant $G(z)$ must have a cutoff frequency $\omega_c \gg \omega_0$, and accordingly

$$G(e^{j\omega T_s}) \approx 1 \text{ for } \omega \leq \frac{2\pi}{T} = \omega_0 \ll \omega_c. \tag{13}$$

However, even though (13) is valid, the tracking error $e = q - q^r$ between plant output and reference B-spline trajectory can be significant, because of modeling errors and external disturbances. For instance, in a robotic manipulator equipped with a standard decentralized control, each motor is affected by gravitational and dynamic coupling terms that are usually neglected in the control design although they can significantly deteriorate the tracking performance of the system [29].

A. Plug-in Repetitive Control

Since it is assumed that the structure of the internal controller of the plant cannot be modified by the user, as in an industrial robots, the proposed approach aims at improving the tracking precision during repetitive tasks by modifying the reference trajectory. As a B-spline curve is completely determined by the position of its control points, the modification of the trajectory can be obtained by directly acting on them, e.g. by means of the control scheme reported in Fig. 8. This scheme is obtained by inserting the trajectory generator, including the filter for control points computation, and the controlled plant $G(z)$ in a discrete-time control loop that, on the basis of the interpolation error $\tilde{q}_i = q^\star_i - q_i$, modifies in real-time the control points sequence (denoted by p^r_i) from the initial value $p^\star_i$. It is a typical dual rate system with the feedback loop running at a sampling period T, considerably higher than the period T_s of the trajectory generator and of the controlled plant $G(z)$. According to internal model principle [30], it is straightforward to conclude that the presence in the loop-function of the term

$$\frac{1}{1 - z_N^{-n}} \tag{14}$$

assures asymptotic perfect tracking of any periodic signal with period n and therefore the tracking error at the given via-points $q^\star_i$ asymptotically vanishes. This is a basic results of Repetitive Control in the discrete-time domain [31], but it requires that the feedback loop is stable. In order to analyze the stability of the scheme running with sampling period T, let's consider the block-scheme representation of Fig. 9 obtained from the control scheme of Fig. 8 after some formal manipulations. The discrete-time transfer function $[\tilde{M}_d\, G]^T(z_N)$ represents the transfer function $\tilde{M}_d(z)\, G(z)$, modeling the trajectory generator and the plant, re-sampled with the period T. Note that this scheme has a quite standard repetitive control structure whose stability can be inferred by analyzing its characteristic equation, i.e.

$$1 + \frac{z_N^{-n}}{1 - z_N^{-n}} K_p\, H(z_N)\, z_N^m\, [\tilde{M}_d\, G]^T(z_N) = 0. \tag{15}$$

By following the approach proposed in [31], it is possible to see that the asymptotic stability of (15) is equivalent to the stability of the feedback system with loop-transfer function

$$L(z_N) = z_N^{-n}\Big(K_p\, H(z_N)\, z_N^m\, [\tilde{M}_d\, G]^T(z_N) - 1\Big).$$

Therefore, by applying the Nyquist criterion it descends that all the poles of (15) are within the unit circle if and only if the polar plot of $L(e^{j\omega T})$ for $-\frac{\pi}{T} \leq \omega \leq \frac{\pi}{T}$ does not encircle or touch the critical points -1. This can be assured by imposing that

$$\left|K_p\, H(e^{j\omega T})\, e^{j\omega\, mT}\, [\tilde{M}_d\, G]^T(e^{j\omega T}) - 1\right| < 1, \quad \omega \leq \frac{\pi}{T}. \tag{16}$$

The condition (16) requires that the complex function $K_p\, H(e^{j\omega T})\, e^{j\omega\, mT}\, [\tilde{M}_d\, G]^T(e^{j\omega T})$ lies within the circle of unit radius centered in $1+j0$, $\forall \omega \leq \frac{\pi}{T}$, see Fig. 10. In nominal conditions, when the plant is able to track the reference B-spline with negligible errors, and therefore (13) is met, the transfer function $[\tilde{M}_d\, G]^T(z_N)$ can be approximated as $[\tilde{M}_d\, G]^T(z_N) \approx \tilde{M}_d^T(z_N)$. In this case, the transfer function $\tilde{M}_d(z)$ of the B-spline filter, re-sampled with period T, describes the relationship between control points and interpolated via-points with, as already remarked, an additional delay of mT seconds, i.e.

$$\tilde{M}_d(z_N) = \frac{Q(z_N)}{P(z_N)} z_N^{-m}.$$

On the other hand, the transfer function $H(e^{j\omega T})$ approximates the relationship between via-points and control points. Therefore,

$$H(e^{j\omega T})\, e^{j\omega\, mT}\, \tilde{M}_d^T(e^{j\omega T}) \approx 1, \quad \omega \leq \frac{\pi}{T} \tag{17}$$

and the condition (16) is valid for $0 < K_p < 2$. Note that the value $K_p = 1$ maximizes the robustness of the RC scheme, since in nominal conditions the polar plot of loop function $L(e^{j\omega T})$ is a point located at the center of the stability region shown in Fig. 10. If the condition (13) is not satisfied and therefore even in nominal conditions (i.e. without external periodic disturbances) the tracking of the reference trajectory is not satisfactory, it is necessary to consider $[\tilde{M}_d\, G]^T(z_N) \approx \tilde{M}_d^T(z_N)\, \Delta M_d(z_N)$ where the term $\Delta M_d(z_N)$ takes into account the effects, that cannot be neglected, of the plant $G(z)$ on the reference input. Note that $\Delta M_d(z_N) \neq G^T(z_N)$ because

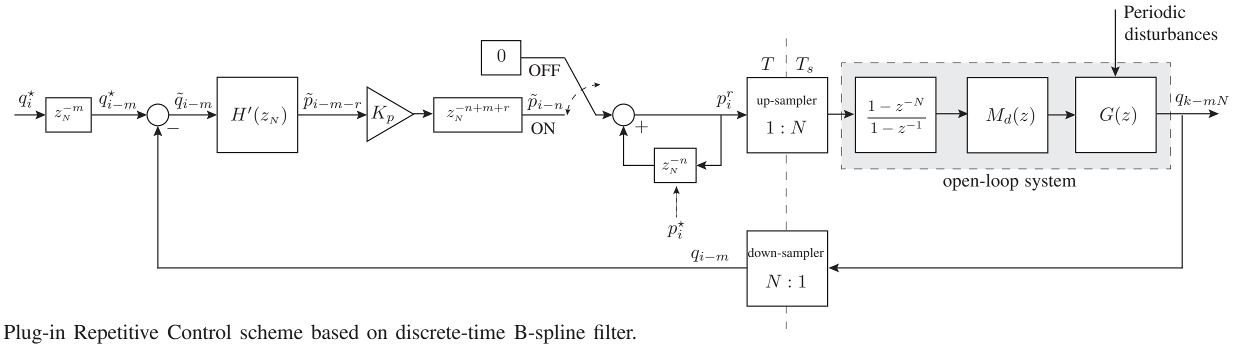


Fig. 8. Plug-in Repetitive Control scheme based on discrete-time B-spline filter.

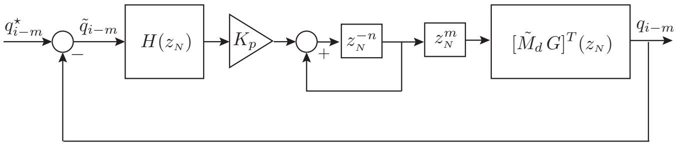


Fig. 9. Equivalent block-scheme representation of Repetitive Control in Fig. 8.

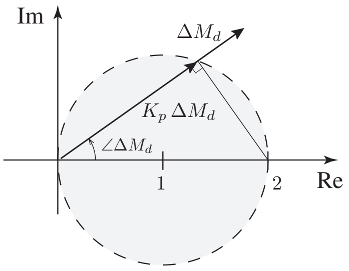


Fig. 10. Stability region of RC for $K_p\,\Delta M_d(e^{j\omega T})$.

of aliasing phenomena. In fact, the down-sampling operation does not commute, that is $[\tilde{M}_d\, G]^T(z_N) \neq \tilde{M}_d^T(z_N)\, G^T(z_N)$ [32]. However, $\Delta M_d(z_N)$ is a direct consequence of $G(z)$ and therefore describes similar phenomena; for instance, if $G(z)$ involves a gain reduction/increase also $\Delta M_d(z_N)$ does involve a similar gain variation, or if $G(z)$ causes a phase delay also $\Delta M_d(z_N)$ is characterized by a negative phase shift.

In this case, condition (16) becomes

$$\left| K_p\,\Delta M_d(e^{j\omega T}) - 1 \right| < 1, \quad \omega \leq \frac{\pi}{T}. \tag{18}$$

As shown in Fig. 10, where the stability region for the function $K_p\,\Delta M_d(e^{j\omega T})$ is shown, the gain of $\Delta M_d(e^{j\omega T})$ can be compensated by a proper choice of the free parameter K_p. Simple geometrical considerations lead to the following stability condition

$$0 < K_p < \frac{2\,\cos(\angle\Delta M_d(e^{j\omega T}))}{|\Delta M_d(e^{j\omega T})|}, \quad \forall\omega \leq \frac{\pi}{T} \tag{19}$$

where $\angle$ denotes the function that returns the phase angle of a complex number. Obviously, (19) can be satisfied only if $-\pi/2 < \angle\Delta M_d(e^{j\omega T}) < \pi/2$. If the phase shift due to the plant exceeds $\pm\pi/2$, the stability of the RC cannot be assured, but a good feedback controller and a proper choice of the trajectory (in particular a proper choice of the time-distance T between the via-points) should ever prevent this possibility. Since, in an industrial robotic system the controller can not be (easily) modified, the stability condition determines the minimum duration of the B-spline trajectory, which depends on the minimum allowable value of T. In particular T must be adapted to the controller bandwidth in order to verify (13).

B. Iterative Learning analysis of the proposed control scheme

In order to understand the effects of the control scheme of Fig. 8 on the resulting reference trajectory, an analysis based on the Iterative Learning perspective can be useful. By a simple inspection of the scheme, it descends that the control points p_i defining the spline trajectory are updated according to the law

$$p_{i,j}^r = p_{i,j-1}^r + K_p\, H(z_N)\tilde{q}_{i-m,j-1} \tag{20}$$

where the index j denotes the current cycle, and z_N is the time-shift operator. Therefore, the i-th control points is modified on the basis of the error $\tilde{q}_{i-m}$ between desired position $q_{i-m}^\star$ and the actual position q_{i-m} (delayed by m samples), transformed in an error in the "control points space" by means of the filter $H(z_N)$. By combining (20) with the I/O relationship of the plant with the B-spline generator, i.e.

$$q_{i-m,j} = [\tilde{M}_d\, G]^T(q)\; p_{i,j}^r$$

it is possible to derive the equation of the iteration error dynamics, that in the z-domain is

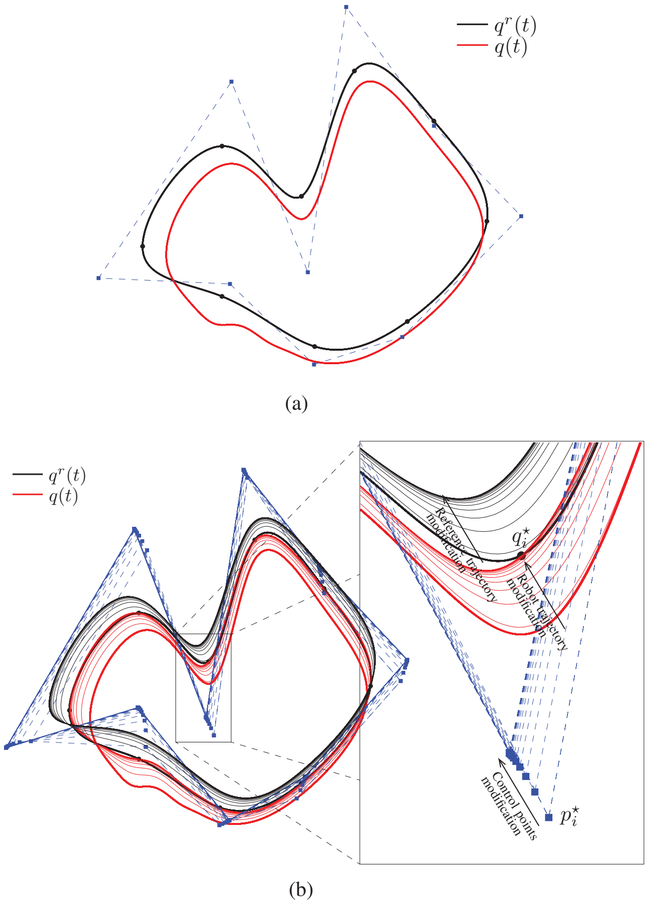
$$\tilde{Q}_j(z_N) = \Big[1 - [\tilde{M}_d\, G]^T(z_N) K_p\, H(z_N)\Big]\tilde{Q}_{j-1}(z_N) \tag{21}$$

where $\tilde{Q}_j(z_N) = \mathcal{Z}_T\{\tilde{q}_{i,j}\}$ is the Z-transform of the sequence obtained from $\tilde{q}_j(t)$ with sampling time T. Therefore, the ILC system is stable and the error $\tilde{q}_{i,j}$ asymptotically vanishes only if $\big[1 - [\tilde{M}_d\, G]^T(z_N) K_p\, H(z_N)\big]$ is a contraction mapping [33], that is

$$\sup_{\omega\in[-\pi/T,\pi/T]} \left| 1 - K_p\, H(e^{j\omega T})\, e^{j\omega\, mT}\, [\tilde{M}_d\, G]^T(e^{j\omega T}) \right| < 1\,. \tag{22}$$

Note that stability condition (22) is equivalent to (16), but equation (21), that leads to (22), provides some additional information, e.g. about the transient. If the condition (17) is satisfied, the choice $K_p = 1$ guarantees that the error $\tilde{q}_{i,j}$ converges to zero after one single cycle for any value of the initial error $\tilde{q}_{i,0}$ [34], while smaller values of K_p can aid to satisfy (22), or equivalently (16), but, on the other hand, they cause a slower convergence of the error to zero.

In Fig. 11 the mechanism is illustrated with an example



(a)

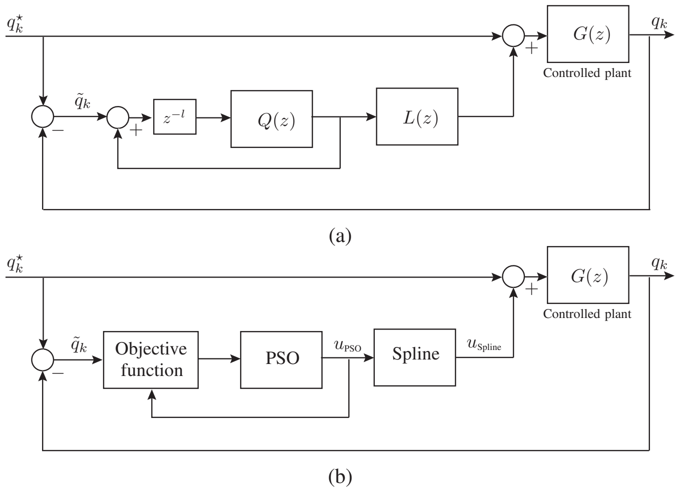
(b)

Fig. 11. Example of tracking of a 2-D B-spline trajectory (a) and trajectory modification via ILC mechanism (b): the solid back line is the reference trajectory $q^r(t)$ while the solid red line is the actual trajectory $q(t)$ performed by the plant.

based on a 2-D B-spline trajectory. The initial tracking error of the plant with its internal control has been assumed very large only for the sake of clarity, see Fig. 11(a). For the same reason, a very small value of K_p has been considered. In this way, the asymptotic approach of the output trajectory to the reference trajectory results very slow and it is possible to appreciate the modification of trajectory and via-points, as highlighted in Fig. 11(b).

C. A comparative evaluation of the proposed control scheme

In order to better highlight the advantages of the proposed controller, its performances have been compared with those of well-settled repetitive control based techniques that share the same design philosophy, namely the possibility to be applied to controlled plants as a plug-in module that does not require modifications on the basic off-the-shelf controller. In particular, the so-called *plug-in repetitive control* [35]–[38], shown in Fig. 12(a), has been considered. Additionally, the scheme recently proposed in [20], where an additive signal, parameterized as a uniform B-spline, is added to the given reference trajectory, has been taken into account, see Fig. 12(b). In this latter case, the controller has not only a plug-in structure but it is also based on the iterative modification of a B-spline function. The main difference with respect to the approach proposed in this paper is that the modification of



(a)

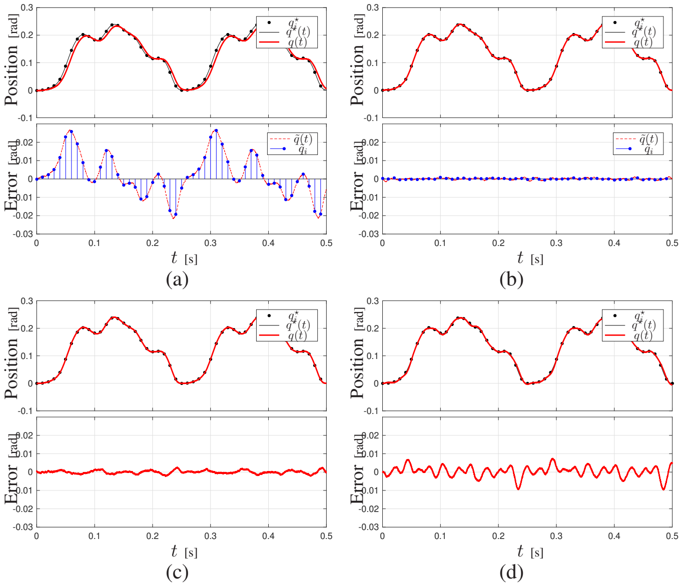
(b)

Fig. 12. Plug-in repetitive control scheme (a) and particle swarm based repetitive spline compensator (b).

the B-spline is obtained by using a Particle Swarm Optimizer (PSO) applied to a cost function that depends on the tracking error and on the control signal.

The plant considered in the simulations is the servo drive with a standard current/velocity/position control architecture used to evaluate the PSO repetitive spline compensator in [20] (the model and the controller are available in [39]), and also the reference trajectory comes from the same paper. Since, the proposed control scheme is based on a B-spline reference trajectory interpolating a set of via-points, the trajectory in [20] has been uniformly sampled in order to obtain the via-points. In Fig. 13(a), the trajectory is shown along with the tracking error without any RC mechanism. The duration of a single cycle is $T_{tot} = 0.5$ s, and the control points are 50. The sampling time of the servo drive is $T_s = 0.1$ ms, while the knot span T is 0.01s. Accordingly $N = 100$.

The application of the B-spline based RC proposed in this



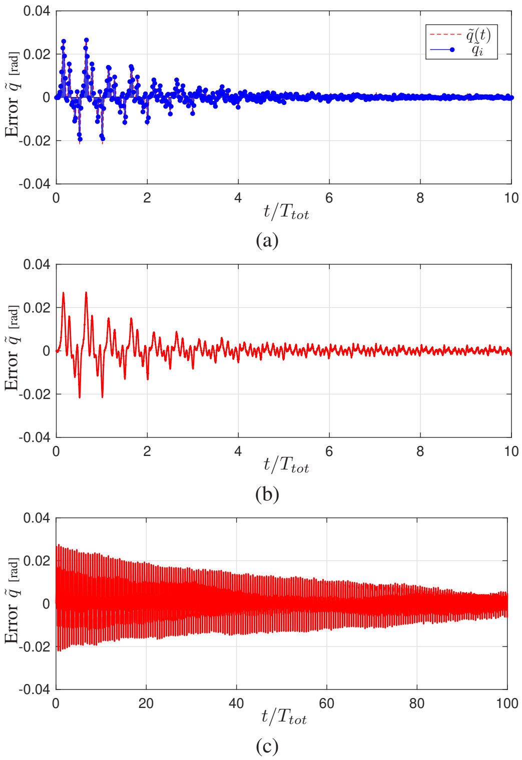
(a)

(b)

(c)

(d)

Fig. 13. Reference trajectory and tracking error without RC (a). Tracking error with the B-spline based RC -iteration #10- (b), the plug-in RC -iteration #10- (c) and the PSO based RC -iteration #100- (d).



(a)

(b)

(c)

Fig. 14. Comparison of the error decay obtained with the B-spline based RC (a), the plug-in RC (b) and the PSO based RC (c).

paper (with $K_p = 0.5$) leads to a quick reduction of the tracking error, as highlighted in Fig. 14(a). After nine cycles the error $\tilde{q}_i$ at the knots has practically vanished, see Fig. 13(b). In general, the overall tracking error has been considerably reduced. Similar results are obtained with the plug-in RC, working at T_s (therefore N times faster than the B-spline based RC), see Fig. 14(b) and Fig. 13(c). In this case the final level of the error is influenced by the choice of the filter $Q(z)$, which has been assumed as a standard low-pass filter, while the decay rate of the error depends on the filter $L(z)$, that in the simulation is a simple gain equal to K_p. Therefore, besides the lower sampling rate, the integration of the B-spline filter in the RC loop simplifies the tuning of the controller, because the free parameters are only K_p and T, and guarantees that the tracking errors goes exactly to zero (at least at the knots), since the presence of the filter $Q(z)$ is not necessary for the stability of the control loop.

The PSO based RC modifies the B-spline curve on the basis of the overall tracking error (and not only at the knots) and for this reason it exhibits a higher robustness with respect to noise and aperiodic disturbances. However, the tracking error converges to zero very slowly, although the (numerous) free parameters that characterize the scheme have been selected by the authors of the paper and are therefore supposed to be optimized. From Fig. 14(c) and Fig. 13(d) it is quite evident that after 99 iterations the error level is higher than the error level of the B-spline based RC after 9 cycles. Finally, from a computational point of view, the PSO based RC is rather demanding, because of the optimization procedure, the spline computation, etc., and its implementation may be unpractical. On the contrary, the computational burden of proposed B-spline based scheme is very low, as it can be roughly assessed by considering the time required by a 5 seconds simulation, namely $t_{sim} = 1.201383\,\mathrm{s}$ for the B-spline based RC, $t_{sim} = 2.679974\,\mathrm{s}$ for the plug-in RC and $t_{sim} = 33.3352\,\mathrm{s}$ for the PSO based RC.

IV. Application of the RC scheme to an industrial manipulator

In order to show the importance for applications of the proposed approach, an extensive experimental activity has been performed. In a real scenario involving an industrial manipulator the proposed control can be used according two different schemes and purposes:

a) the iterative modification of the robot trajectories defined in the joint-space is obtained on the basis of the measurements provided by the proprioceptive sensors of the robot, i.e. motors encoders;
b) the robot trajectories are directly defined in the workspace and are modified on the basis of an external sensor that detect the position of the end-effector in the 3-D space, i.e. a RGB-D camera [40].

In case a), the goal of the repetitive control is improving the robot precision by compensating the errors that the internal controller of the robot is not able to correct.

In case b) the external sensor allows the compensation of errors that are not sensed by the motors encoders, e.g. position errors due to the elasticity of the transmission chain or to the flexibility of the links, or errors due to misalignments between the reference frame of the robot base and the reference frame in which the desired trajectory is defined. Note that the proposed control scheme based on a low-rate outer loop is very attractive in this second scenario, since the integration of many external sensors may be affected by some bottlenecks, due to the intrinsic properties of the sensor itself, like in the case of cameras, or to the time required for elaboration and data transmission, that limit the minimum sampling time.

In Fig. 15, the 60 via-points, and the interpolating workspace trajectory, used in all the experiments reported in this section are shown. Obviously, in the *scenario a*, based on joint-space trajectories, the reference trajectory is built by firstly applying the inverse kinematics function of manipulator to the given via-points.

A. *Scenario a*

In order to experimentally evaluate the proposed method the setup shown in Fig. 16 has been arranged. The system is composed by a Comau Smart5 Six industrial robotic arm, a C4G Controller and a standard PC with an Intel Core 2 Duo 2.4 GHz processor. The Comau Smart5 Six is a 6 DOF robot with anthropomorphic structure with a payload of 6 Kg. The robot is driven by the C4G Controller that performs both the position/velocity control and the power stage management with current control of each joint. Moreover, the C4G controller implements a software option called “C4G open” that allows the integration of the robot control unit with an external personal computer in order to develop a

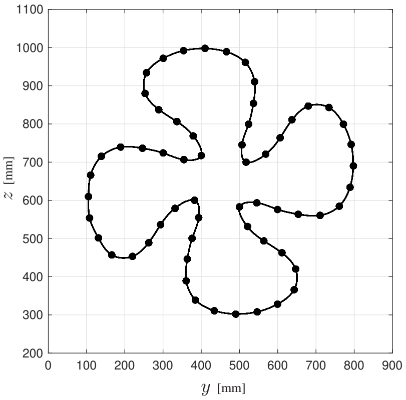


Fig. 15. Reference trajectory interpolating a set of via-points disposed on the plane $y - z$ used to experimentally validate the proposed control scheme both in the robot joint-space and in the robot workspace.

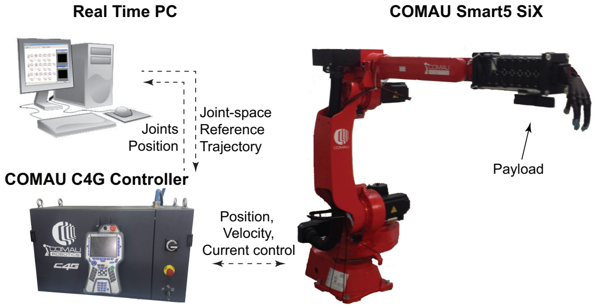
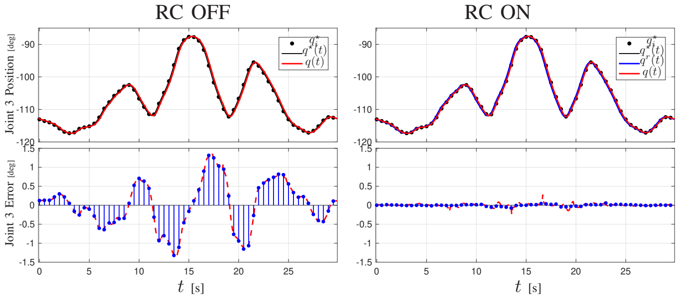


Fig. 16. Experimental setup based on a Comau Smart5 Six industrial manipulator.

complex control system at a higher hierarchical level. The C4G Open architecture is based on a real time communication on Ethernet network between the controller and the PC, equipped with the real-time operating system RTAI-Linux on a Ubuntu NATTY distribution, that allows the trajectory generator/RC loop to run with a sampling period $T_s = 1$ ms. For the design of the control scheme and of trajectory generator, the MatLab/Simulink/RealTime Workshop environment has been used. The knot span of the uniform trajectory has been set to $T = 0.5$ s, and therefore its total duration is $T_{tot} = 30$ s.

For the sake of clarity, the behavior of only one robotic joint (the third) has been initially analyzed. In Fig. 17(a) the performance of the original robotic system, without RC, is shown. As can be seen, the third joint is affected by a quite evident tracking error, due to both the dynamic coupling with the other joints and the 3 Kg payload represented by the UBHand IV robotic hand [41].

In Fig. 18 the tracking performance of the third joint is presented when the RC is switched on. It is worth noticing that, as soon as the RC is activated, the error $\tilde{q}_i$ at the given via-points $q_i^\star$ starts decreasing and vanishes in a few iterations. More generally, the entire error between the planned B-spline and curve tracked by the robot considerably decreases. This reduction is quite evident if the tracking error obtained without RC during a trajectory period, shown in Fig. 17(a), is compared with the error obtained after application of the controller, see Fig. 17(b) where a detail of the trajectory



(a)

(b)

Fig. 17. Tracking performance of the third joint during a trajectory cycle without (a) and with (b) RC (iteration #6).

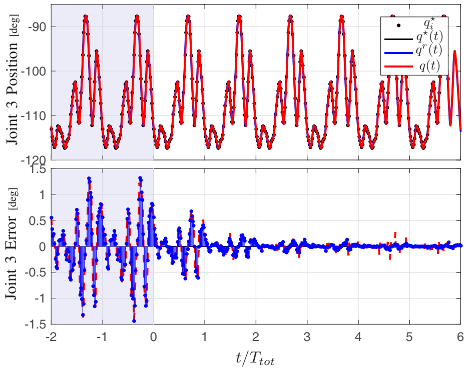


Fig. 18. Response of the robotic system when the RC is activated ($t = 0$). Errors at sampling instants T are highlighted in blue, while the red line denotes the overall tracking error.

tracking with RC (after 5 cycles) is shown. In particular, a reduction of the maximum tracking error of about 80% can be observed, while the error at knots goes practically to zero.

Similar considerations are valid for all the robot joints, as shown in Fig. 19 where the errors obtained during the tracking of the trajectory in Fig. 15, with the RC mechanism activated, are reported.

B. Scenario b

In this scenario, the cyclic motion of Fig. 15, directly defined in the robot workspace by means of a uniform B-spline interpolating a set of via-points, is considered. Additionally, an external sensor, that is a simple vision system based on ASUS Xtion PRO Live RGB-D camera, has been integrated into the robotic setup described in Sec. IV-A, see Fig. 20. The camera, which is disposed in front of the robot, detects the position of a marker located at the robot end-effector with a resolution of about 1 mm. Note that the precision of the camera, which is a low cost device, is lower than the precision of the industrial robot (whose repeatability is 0.05 mm) but the proposed experiment is only a proof of concept aiming at demonstrating how real applications can benefit from the RC scheme. Besides the poor resolution of the camera and the tracking error that affects the robot with the original controller (which can be supposed periodic along a cyclic trajectory), the main source of error is non-perfect calibration of the camera. In fact, the position and the angular orientation of the camera around its visual axis are certainly affected by an error, and in

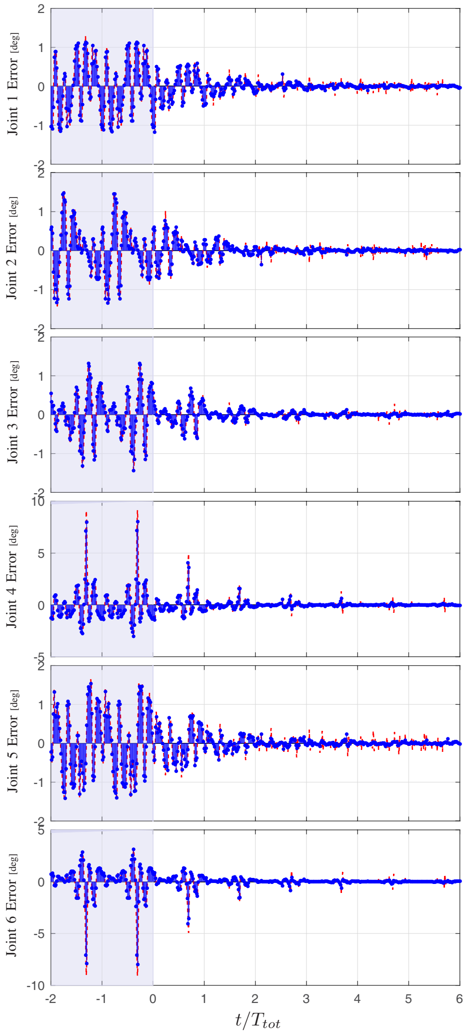


Fig. 19. Error decay for each joint of the robot Comau Smart5 Six during the execution of the trajectory shown in Fig. 15 after the activation of the RC mechanism ($t = 0$).

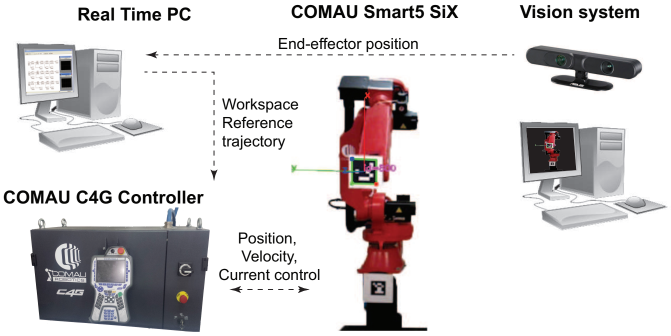


Fig. 20. Experimental setup based on the Comau Smart5 Six industrial manipulator with an external RGB-D sensor.

this application, only a rough calibration has been performed. Note that, being the calibration errors constant, they can be considered periodic.

For the sake of simplicity, a fixed orientation of the robot end-effector has been considered, see Fig. 21 where the view of the camera along with the desired trajectory is reported. In the scenario, the via-points are six dimensional vectors $q_i^\star = [x_i^\star, y_i^\star, z_i^\star, \varphi_i^\star, \theta_i^\star, \psi_i^\star]^T$, where parameters $\varphi_i^\star$, $\theta_i^\star$, $\psi_i^\star$ are a minimal representation of the orientation, such as Roll-Pitch-Yaw angles. As a consequence, the control points and the cyclic B-spline trajectory are six dimensional too, and therefore it is necessary to implement a vectorial version of the trajectory generator, and more specifically of the RC scheme. Moreover, since the reference trajectory is defined in the robot workspace and the RC mechanism is performed in the workspace as well, while the robot is controlled at the joint level, the initial RC scheme shown in Fig. 8 has been slightly modified in order to take into account the kinematic transformations of the manipulator. In Fig. 22 the RC control based on the modification of a uniform B-spline reference trajectory defined in the robot manipulator workspace is shown. Note that the only requirement for the success of the RC is that the robot is able to track the planned trajectory in nominal conditions and no additional hypotheses are necessary in order to guarantee the stability of the loop. The gain K_p is assumed equal to the identity matrix, i.e. $K_p = I_6$, and $T = 1$s ($T_{tot} = 60$s). In Fig. 23 the error decay along y and z axis when the RC is activated is shown. In few cycles the RC is able to considerably reduce the error at the via-points. In the example, despite the noise due to the position estimation with the camera, the error is reduced of one order of magnitude. In order to better appreciate the improvement caused by the application of the RC, a single trajectory cycle along the y-axis with and without RC is considered in Fig. 24. By comparing the tracking errors it is clear that the error is reduced not only at the knots iT but also during the inter-samples.

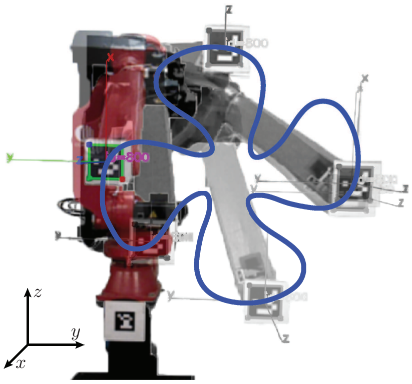


Fig. 21. View of the RGB-D camera and desired trajectory.

Finally, in Fig. 25 the modification of the geometric reference path $q^r(t)$ due to the RC mechanism has been highlighted, by comparing the shape of the reference trajectory $q^r(t)$ and of the actual trajectory $q(t)$, before and after the application of the RC. As already remarked in Sec. III-B, the variation of the control points with respect to their initial value obtained by interpolating the desired via-points produces a deviation of the reference trajectory that compensate periodic errors affecting the system, which in this way is able to follow the desired path with an improved precision.

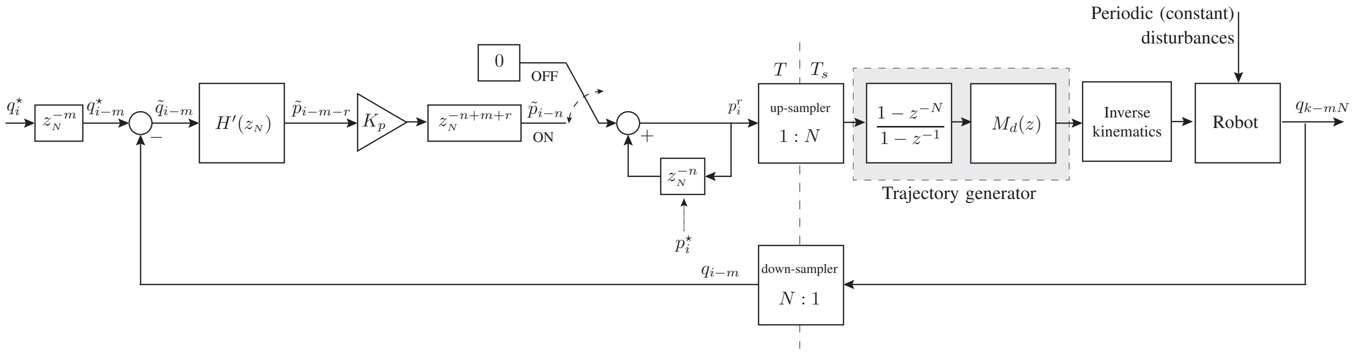


Fig. 22. Repetitive Control scheme for tracking uniform B-spline trajectories defined in the workspace of a robot manipulator.

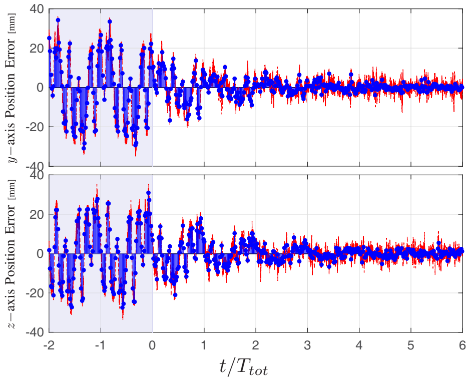
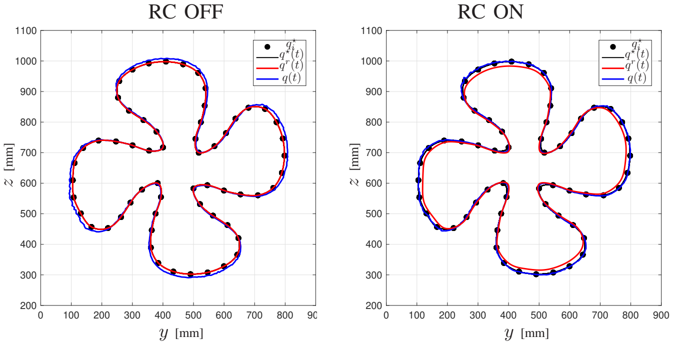
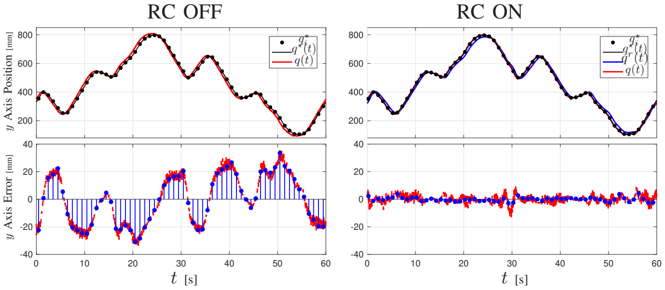


Fig. 23. Error decay in the y and z directions after the activation of the RC mechanism ($t = 0$).



(a) (b)

Fig. 25. $y - z$ planar view of the tracking performance of the system during a trajectory cycle without and with RC (iteration #6).



(a) (b)

Fig. 24. Tracking performance of the system along the y axis during a trajectory cycle without and with RC (iteration #6).

V. Conclusions

In this paper, motion planning and reactive control have been integrated in order to obtain a perfect tracking of a desired set of via-points. By considering tasks performed cyclically, which are quite common in the industrial and robotics field, a B-spline trajectory generation algorithm has been enhanced with a RC-type mechanism that modifies in real-time the control points in order to nullify the tracking error at the desired points. The effectiveness of the proposed approach has been demonstrated both analytically and experimentally. In particular, the tests performed on an industrial manipulator, both in the joint space and in the workspace, have shown that this scheme can be used to enhance the performance of the original position controller of the robot without modification of the controller itself but only by implementing a low-rate outer control loop that modifies the reference trajectory provided to the manipulator. With respect to existing RC schemes, the B-spline based controller offers the following advantages for practical applications:

- straightforward scalability for multi-dimensional servo-systems;
- low sampling rate;
- simple implementation with the need of tuning only the gain K_p (generally equal to 1);
- stability conditions easily met, even in the presence of non-linear elements, e.g. the inverse kinematics function which has been included in the robot workspace RC scheme.

The fact that the perfect tracking is guaranteed only at via-points does not seem a limitation, since the reference trajectory is generally defined by these via-points. Moreover, it is possible to show that the reduction of the error at the via-points also implies a reduction of the overall approximation error of the curve. In general, the so-called inter-sample error depends on the capability of the original plant to track the given curve, and accordingly on the period T between knots and on the spline degree d, see [8] for more details.

Finally, the proposed approach can be used to refine the computation of the control points for a given motion trajectory in order to compensate for cyclic disturbances that characterize the plant. After an initial “training” the modified control points

p_i^r that take into account the dynamic behavior of the plant can be applied without the adaptation mechanism in lieu of the theoretical values $p_i^\star$, computed on the basis of mere geometric interpolation conditions.

Appendix

Control points computation for quintic B-spline trajectories For quintic B-splines, i.e. $d = 5$, the system for the off-line computation of control points $\boldsymbol{p}^\star$ from the via-points $\mathbf{q}^\star$ is

$$\boldsymbol{A}\,\boldsymbol{p}^\star = \mathbf{q}^\star$$

with

$$\boldsymbol{A} = \frac{1}{120}\begin{bmatrix} 66 & 26 & 1 & 0 & & & \cdots & & 0 & 1 & 26 \\ 26 & 66 & 26 & 1 & 0 & & \cdots & & & 0 & 1 \\ 1 & 26 & 66 & 26 & 1 & & \cdots & & & & 0 \\ 0 & 1 & 26 & 66 & 26 & 1 & & & & & \\ \vdots & & & & \ddots & \ddots & \ddots & & & & \vdots \\ & & & & & 1 & 26 & 66 & 26 & 1 & 0 \\ 0 & & & & \cdots & 0 & 1 & 26 & 66 & 26 & 1 \\ 1 & 0 & & & \cdots & & 0 & 1 & 26 & 66 & 26 \\ 26 & 1 & 0 & & \cdots & & & 0 & 1 & 26 & 66 \end{bmatrix}.$$

The coefficients $h(n)$ of the FIR filter $H(z_N)$ for the online computation of the control points are

$$h(n) = c_1\,\alpha_1^{|n|} + c_2\,\alpha_2^{|n|}$$

where α_1 and α_2 are the stable poles of

$$\frac{P(z_N)}{Q(z_N)} = \frac{120}{z_N^2 + 26z_N + 66 + 26z_N^{-1} + z_N^{-2}}$$

defined by

$$\alpha_i = \frac{1}{2}(2 + u_i + \sqrt{4\,u_i + u_i^2}), \qquad i = 1, 2$$

with $u_i = -15 \pm \sqrt{105}$, and the coefficients c_i are

$$c_1 = \frac{\alpha_1(-1+\alpha_1)(-1+\alpha_2)^2}{(\alpha_1-\alpha_2)(-1+\alpha_1\alpha_2)(1+\alpha_1)}$$

$$c_2 = \frac{\alpha_2(-1+\alpha_2)(-1+\alpha_1)^2}{(\alpha_2-\alpha_1)(-1+\alpha_1\alpha_2)(1+\alpha_2)}.$$

References

[1] T. Inoue, S. Iwai, and M. Nakano, "High accuracy control of a proton synchrotron magnet power supply," in *Proc. 8th IFAC World Congress*, vol. 20, 1981, pp. 3137–3142.

[2] T. Inoue, M. Nakano, and S. Iwai, "High accuracy control of servomechanism for repeated contouring," in *Proc. 10th Annual Symp. Incremental Motion Contr. Syst. and Device*, 1981, pp. 258–292.

[3] L. Cuiyan, Z. Dongchun, and Z. Xianyi, "A survey of repetitive control," in *Intelligent Robots and Systems, 2004. (IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on*, vol. 2, Sept 2004, pp. 1160–1166 vol.2.

[4] Y. Wang, F. Gao, and F. J. D. III, "Survey on iterative learning control, repetitive control, and run-to-run control," *Journal of Process Control*, vol. 19, no. 10, pp. 1589 – 1600, 2009.

[5] S. Hara, Y. Yamamoto, T. Omata, and M. Nakano, "Repetitive control system: a new type servo system for periodic exogenous signals," *Automatic Control, IEEE Transactions on*, vol. 33, no. 7, pp. 659–668, Jul 1988.

[6] L. Biagiotti and C. Melchiorri, *Trajectory Planning for Automatic Machines and Robots*, 1st ed. Heidelberg, Germany: Springer, 2008.

[7] ——, "B-spline based filters for multi-point trajectories planning," in *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, May 2010, pp. 3065–3070.

[8] L. Biagiotti, L. Moriello, and C. Melchiorri, "A repetitive control scheme based on b-spline trajectories modification," *IFAC-PapersOnLine*, vol. 48, no. 19, pp. 262 – 267, 2015, 11th IFAC Symposium on Robot Control SYROCO 2015.

[9] ——, "A repetitive control scheme for industrial robots based on b-spline trajectories," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Sept 2015, pp. 5417–5422.

[10] J. van de Wijdeven and O. Bosgra, "Using basis functions in iterative learning control: analysis and design theory," *International Journal of Control*, vol. 83, no. 4, pp. 661–675, mar 2010.

[11] K. Hamamoto and T. Sugie, "An iterative learning control algorithm within prescribed input-output subspace," *Automatica*, vol. 37, no. 11, pp. 1803 – 1809, 2001. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0005109801001339

[12] ——, "Iterative learning control for robot manipulators using the finite dimensional input subspace," *IEEE Transactions on Robotics and Automation*, vol. 18, no. 4, pp. 632–635, Aug 2002.

[13] J. Bolder, B. Lemmen, S. Koekebakker, T. Oomen, O. Bosgra, and M. Steinbuch, "Iterative learning control with basis functions for media positioning in scanning inkjet printers," in *2012 IEEE International Symposium on Intelligent Control*, Oct 2012, pp. 1255–1260.

[14] F. Boeren, A. Bareja, T. Kok, and T. Oomen, "Unified ilc framework for repeating and varying tasks: A frequency domain approach with application to a wire-bonder," in *2015 54th IEEE Conference on Decision and Control (CDC)*, Dec 2015, pp. 6724–6729.

[15] M. Q. Phan and J. A. Frueh, "Learning control for trajectory tracking using basis functions," in *Proceedings of 35th IEEE Conference on Decision and Control*, vol. 3, Dec 1996, pp. 2490–2492 vol.3.

[16] N. Sadegh and K. Guglielmo, "Design and implementation of adaptive and repetitive controllers for mechanical manipulators," *IEEE Transactions on Robotics and Automation*, vol. 8, no. 3, pp. 395–400, Jun 1992.

[17] P. Rakprayoon, P. Sanposh, and N. Chayopitak, "Adaptive and repetitive controller for robotic manipulators with slowly updating scheme using b-spline shape function," in *2011 IEEE International Conference on Robotics and Biomimetics*, Dec 2011, pp. 2330–2335.

[18] N. Sadegh and K. Guglielmo, "A new repetitive controller for mechanical manipulators," *Journal of Robotic Systems*, vol. 8, no. 4, pp. 507–529, 1991.

[19] H. Wang and Q. Zou, "B-spline-decomposition-based approach to multi-axis trajectory tracking: Nanomanipulation example," *IEEE Transactions on Control Systems Technology*, vol. 22, no. 4, pp. 1573–1580, July 2014.

[20] K. Basinski, "Particle swarm based repetitive spline compensator for servo drives," *PRZEGLkAD ELEKTROTECHNICZNY*, vol. 1, no. 2, pp. 183–189, feb 2017.

[21] Y. Chen, K. L. Moore, and V. Bahl, "Learning feedforward control using a dilated b-spline network: frequency domain analysis and design," *IEEE Transactions on Neural Networks*, vol. 15, no. 2, pp. 355–366, March 2004.

[22] H. Deng, R. Oruganti, and D. Srinivasan, "Neural controller for ups inverters based on b-spline network," *IEEE Transactions on Industrial Electronics*, vol. 55, no. 2, pp. 899–909, Feb 2008.

[23] M. Malkowski, B. Ufnalski, and L. M. Grzesiak, "B-spline based repetitive controller revisited: Error shift, higher-order polynomials and smooth pass-to-pass transition," in *2015 19th International Conference on System Theory, Control and Computing (ICSTCC)*, Oct 2015, pp. 19–25.

[24] F.-S. Lee, J.-C. Wang, and C.-J. Chien, "B-spline network-based iterative learning control for trajectory tracking of a piezoelectric actuator," *Mechanical Systems and Signal Processing*, vol. 23, no. 2, pp. 523 – 538, 2009.

[25] Z. Lin, J. Wang, and D. Howe, "A learning feed-forward current controller for linear reciprocating vapor compressors," *IEEE Transactions on Industrial Electronics*, vol. 58, no. 8, pp. 3383–3390, Aug 2011.

[26] R. W. Longman, "Iterative learning control and repetitive control for engineering practice," *International Journal of Control*, vol. 73, no. 10, pp. 930–954, 2000.

[27] L. Biagiotti and C. Melchiorri, "Online trajectory planning and filtering for robotic applications via b-spline smoothing filters," in *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, Nov 2013, pp. 5668–5673.

[28] K. Ogata, *Modern control engineering (3rd ed.)*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1997.

[29] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo, *Robotics: Modelling, Planning and Control*, 1st ed. Springer Publishing Company, Incorporated, 2008.

[30] B. Francis and W. Wonham, "The internal model principle for linear multivariable regulators," *Applied Mathematics and Optimization*, vol. 2, no. 2, pp. 170–194, 1975.

[31] M. Tomizuka, T.-C. Tsao, and K.-K. Chew, "Discrete-time domain analysis and synthesis of repetitive controllers," in *American Control Conference, 1988*, June 1988, pp. 860–866.

[32] J. Salt and P. Albertos, "Model-based multirate controllers design," *IEEE Transactions on Control Systems Technology*, vol. 13, no. 6, pp. 988–997, Nov 2005.

[33] D. A. Bristow, M. Tharayil, and A. G. Alleyne, "A survey of iterative learning control," *IEEE Control Systems*, vol. 26, no. 3, pp. 96–114, June 2006.

[34] T. Kavli, "Frequency domain synthesis of trajectory learning controllers for robot manipulators," *Journal of Robotic Systems*, vol. 9, no. 5, pp. 663–680, 1992.

[35] C. Cosner, G. Anwar, and M. Tomizuka, "Plug in repetitive control for industrial robotic manipulators," in *Robotics and Automation, 1990. Proceedings., 1990 IEEE International Conference on*, May 1990, pp. 1970–1975 vol.3.

[36] M.-C. Tsai and W.-S. Yao, "Design of a plug-in type repetitive controller for periodic inputs," *IEEE Transactions on Control Systems Technology*, vol. 10, no. 4, pp. 547–555, July 2002.

[37] Y. R. Teo and A. J. Fleming, "A new repetitive control scheme based on non-causal fir filters," in *2014 American Control Conference*, June 2014, pp. 991–996.

[38] Z. Li, W. Zhang, H. Nakamura, M. Koga, and X. Xu, "Design and analysis of plug-in repetitive controller for zero-following performance," in *The 26th Chinese Control and Decision Conference (2014 CCDC)*, May 2014, pp. 840–846.

[39] B. Ufnalski, "Particle swarm based repetitive spline compensator," MATLAB Central File Exchange, Jun. 2018. [Online]. Available: https://www.mathworks.com/matlabcentral/fileexchange/56448-particle-swarm-based-repetitive-spline-compensator?focused=7185451&tab=function

[40] P. Henry, M. Krainin, E. Herbst, X. Ren, and D. Fox, "Rgb-d mapping: Using depth cameras for dense 3d modeling of indoor environments," in *In the 12th International Symposium on Experimental Robotics (ISER*, 2010.

[41] C. Melchiorri, G. Palli, G. Berselli, and G. Vassura, "Development of the ub hand iv: Overview of design solutions and enabling technologies," *Robotics Automation Magazine, IEEE*, vol. 20, no. 3, pp. 72–81, Sept 2013.