Redundancy resolution in tasks with parametrizable uncertainty

Citation for published version (APA):

Djordjevic, G., Kostic, D., Rasic, M., & Surdilovic, D. (2001). Redundancy resolution in tasks with parametrizable uncertainty. *Journal of Automatic Control*, 11(2), 83-88. [https://doi.org/10.1016/0196-6553\(83\)90118-9](https://doi.org/10.1016/0196-6553(83)90118-9)

DOI:

[10.1016/0196-6553\(83\)90118-9](https://doi.org/10.1016/0196-6553(83)90118-9)

Document status and date:

Published: 01/01/2001

Document Version:

Publisher's PDF, also known as Version of Record (includes final page, issue and volume numbers)

Please check the document version of this publication:

- A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
- The final author version and the galley proof are versions of the publication after peer review.
- The final published version features the final layout of the paper including the volume, issue and page numbers.

[Link to publication](#)

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license above, please follow below link for the End User Agreement:

www.tue.nl/taverne

Take down policy

If you believe that this document breaches copyright please contact us at:

openaccess@tue.nl

providing details and we will investigate your claim.

Redundancy Resolution in Tasks with Parameterizable Uncertainty

Goran Đorđević¹, Dragan Kostić², Milan Rašić³, Dragoljub Šurdilović⁴

¹ Johns Hopkins University, Biomedical Engineering, 720 Rutland Ave, Baltimore, MD 21205 -2196 [dordeuic@bme.jhu.edu]

² a [dkostic@wfw.wtb.tue.nl]

³ University of Niš, Faculty of Electronic Eng. 18000 Niš, Yugoslavia

[mrasic@elfak.ni.ac.yu]

⁴ Fraunhofer IPK, Pascal str. 8-9, D-10587 Berlin, Germany

[Dragoljub.Surdilovic@ipk.fhg.de]

Abstract

Redundant robots motion planning and control in uncertain task is addressed by model-based approach. Instead of control and adapt the environment to the robot or apply a complex visual servoing system, we have modeled the redundancy resolution (RR) on the parameter spaces that quantify uncertainties of the task. A modeling tool was Successive Approximations (SA). It provides very advantageous properties: small computational effort and small model size, accurate output, extrapolation, and generalization across parameter set obtained by random addressing of the model. The task discussed is press loading with typical two-dimensional uncertainties in pick-up and unloading locations. The robot used is 4 DOF planar robot. The SA-based models of redundancy resolution in the 2D parameter spaces are highly efficient: for more than 30 times less computational efforts resulted in a zero end-point errors, regardless of the task uncertainty.

1 Introduction

One of the costs of robot application in industry is adaptation of its workspace to become predictable and structured. This is usually very complex and expensive task. Solution to the problem can be found in visual servoing of the robot. Besides it requires very expensive image processing system, visual servoing algorithms not robust for industrial applications. Another important issue that arises in a case of unpredictable environment is that sometimes robot is required to produce some inner motion, without changing its primary task, as a response to some changes in the workspace. Hence, truly flexible system should have a kinematically redundant robot. This fact, besides managing of the unpredictable workspace, opens another research subject: on-line redundancy resolution (RR) that covers both, motion planning and motion control. Oppositely to off-line planning and optimal control of redundant robot, on-line redundancy resolution is a serious problem. Generally speaking, on-line optimal control will use local optimal algorithms, which are suboptimal. They are also computationally too complex that can hardly be implemented even on modern robot control architectures.

Solution to on-line control of redundancy in the presence of task uncertainties can be found, besides visual servoing, in specific modeling of the robot in the task. This

model will be parameterized in accordance with a set of parameters that cover all of the task's uncertainties. Parameters are chosen heuristically, having in mind that their values should be easily estimated, using typical industrial sensors, such as encoders, tachos, reflexive optocouplers, etc. Consequently, the model of the robot in parameterized task will be designed off-line, based on all of available control knowledge on the task. As an example we use the model of the robot that avoids obstacle of uncertain but parameterizable position. If that model is of small computational complexity, it becomes a promising candidate for on-line control.

Suggested approach follows several steps. First, the task is parameterized regarding expected uncertainties. On the chosen parameter space, the motion planning and global optimal redundancy resolution is computed, producing parameterized joint motions. The modeling procedure on the parameter space, the Successive Approximations [2,3,4], will be applied to parameterized sets of joint motions. The result will be a set of parameterized models, one for each joint. Finally, the robot control system, equipped with that kind of feed-forward model, on-line accepts estimated values of the changes in environment, and produces the output. If we have modeled redundancy, the output of the model will be a set of optimal joint motions, which resolve the redundancy. As a consequence of the SA procedure applied in modeling, the model will be of small size, computationally highly efficient, it will have analytical output and can be even used in generation of the paths never applied as examples for modeling [2,4].

The paper is organized as follows. Next section gives some basics on how to apply the SA procedure for modeling of the robot in uncertain task. The third section gives detailed explanation of the SA-based model in the case of redundant robot in press loading task. The concluding remarks are given in the last section.

2 Successive Approximations based robot modeling in uncertain tasks

The SA procedure yields a parametric model of some ordered data set, by an successive repetition of polynomial least-squares fitting procedure. It generalizes, i.e. interpolates and extrapolates in dimensions of the chosen parameter space. The output of the model is a polynomial with coefficients as functions of chosen parameters.

In previous research, the SA procedure has been shown very efficient in modeling of robot kinematics [2,3,4], specifically in RR. The SA-based RR model is obtained by compilation of parameterized joint motions, after the redundancy has been resolved. Taken as a direct one, the model gives a joint motion which maintains applied RR procedure, given the external motion by means of parameters used for model generation. The small errors obtained and, the most of all, the enormously reduced complexity of computations, has suggested potentials of this method for modeling of kinematics [2]. Another important issue is generalization across the parameters that enables the extension of the model to the tasks, or paths, never used as examples. Hence, the model becomes domain-general, not task-general. One typical example is the model built on rectilinear segments and parameterized only by point-rotation transformation. By specific interpretation of the model it is achieved that the same model can be used for resolving a much more complex paths like rosette, belonging to the domain [2].

In the case of task-general model, the parameters have different meaning: they describe the uncertainties of the task. In this Section we elaborate the SA procedure for task-specific case only to enable easier understanding of the Example in the next Section. Given the path Π , it can be considered as variable of the parameters that describe uncertainty of the workspace that alter the shape of the path. Generally, if a position is demanded only, the path can be affected by uncertain ending-point, uncertain starting-point, or uncertain via-points, Fig. 1.

Fig. 1. Typical task uncertainties in ending-point, via-point and starting-point.

Suppose that path Π of the robot's tip is associated with temporal representation by izochronous set of consecutive points $\Pi(i\Delta t)$, $i=0, \dots, n_t$, along which the robot's tip position and orientation is defined by vector

$$\mathbf{X}(i\Delta t) = [x \ y \ z \ \alpha \ \beta \ \gamma]^T, \quad i=0, \dots, n_t \quad (1)$$

The task parameter ρ_1 , for example velocity variation of some object within robots workspace, is defined on the interval $\rho_1 \in [\rho_{1\min}, \rho_{1\max}]$. Parameterization of the path Π is obtained by nonlinear function $F_1(\Pi(t), \rho_1)$, producing the similar path $\Pi_{\rho_1}(i\Delta t)$. Due to easier procedure later in approximations we scaled the time interval $[0, t_{nt}] \mapsto [-1, 1]$. After applying a transformation to the path, a new, parameterized path is obtained $\Pi_{\rho_1}(\tau)$.

The computation of inverse kinematics, specially if the redundancy exists, is notoriously cumbersome task. It requires a lot of expert knowledge and, on the other side, they are very computationally expensive. Both of problems will be solved if we use a parameterized solutions of redundancy and build a parametric model. The complexity of RR

algorithm will not influence the complexity later in model-based computation.

On the parameterized path, we apply appropriate RR algorithm, which gives

$$F_{1,i}(\Pi(\tau), \rho_1) = \Pi_{\rho_1,i}(t) \xrightarrow{IK} \mathbf{q}_{\rho_1,i}(\tau), \quad (2)$$

$$\rho_{1,i} = \rho_{1\min} + i\Delta\rho_1, \quad \Delta\rho_1 = \frac{\rho_{1\max} - \rho_{1\min}}{n_1}$$

where $\mathbf{q} = [q_1 \dots q_n]^T$, n is number of robot DOF, and i states for i -th example due to task parameter $\rho_1 = \rho_{1,i}$.

Now, on the parameterized motions $\mathbf{q}_{\rho_1,i}(\tau)$ set we apply fitting procedure on each of elements of the set $\mathbf{q}_{\rho_1,i}$, $i=1, \dots, n_1$, with polynomials ${}^i P_{n_\alpha}({}^i \alpha_j; \tau)$, $i=0, \dots, n_1$. This step gives a matrix of coefficients

$$A = \begin{bmatrix} {}^0 P_{n_\gamma} & {}^1 P_{n_\gamma} & \dots & {}^{n_1} P_{n_\gamma} \\ {}^0 \alpha_0 & {}^1 \alpha_0 & \dots & {}^{n_1} \alpha_0 \\ \vdots & \vdots & \ddots & \vdots \\ {}^0 \alpha_{n_\alpha} & {}^1 \alpha_{n_\alpha} & \dots & {}^{n_1} \alpha_{n_\alpha} \end{bmatrix} \quad (3)$$

The number of rows n_1 of the matrix A is often to large if we want to use all necessary examples in modeling. Besides, this set of coefficients can not be used between i th and $(i+1)$ th example, since it is not included in examples set.

So, we suggest a generalization of coefficients ${}^i \alpha_j$, keeping j constant, $\{{}^i \alpha_0, {}^i \alpha_1, \dots, {}^i \alpha_{n_\alpha}\}$, by fitting polynomials ${}^i P_{n_\beta}({}^i \beta_j; \rho_1)$, with task parameter ρ_1 as variable.

This produces another set of coefficients

$$\left. \begin{matrix} {}^0 P_{n_\beta} : \\ {}^1 P_{n_\beta} : \\ \vdots \\ {}^{n_\alpha} P_{n_\beta} : \end{matrix} \right\} B = \begin{bmatrix} {}^0 \beta_0 & \dots & {}^0 \beta_{n_\beta} \\ {}^1 \beta_0 & \dots & {}^1 \beta_{n_\beta} \\ \vdots & \ddots & \vdots \\ {}^{n_\alpha} \beta_0 & \dots & {}^{n_\alpha} \beta_{n_\beta} \end{bmatrix} \quad (4)$$

Matrix B now has $(n_\alpha + 1)(n_\beta + 1)$ elements and it is, generally, of the smaller size than matrix A , $(n_\alpha + 1)(n_1 + 1)$.

Another step is incorporation of a new task parameter defined by a nonlinear transformation

$$F_{2,j}((F_{1,i}(\Pi(\tau), \rho_1), \rho_2) = \Pi_{\rho_{21,i}}(t) \xrightarrow{IK} \mathbf{q}_{\rho_{21,i}}(\tau) \quad (5)$$

For example, it can define another source of task uncertainties, like relative position of pick-up point. For every single transformation made by changing the parameter $\{\rho_{2,j}, j=0, \dots, n_2\}$, where $\rho_2 \in [\rho_{2\min}, \rho_{2\max}]$, a set of coefficients ${}^i \beta_j$, like in matrix B , is obtained. To uniquely identify each of the coefficients we will denote them by ${}^{i,j} \beta_k$, $i=0, \dots, n_\alpha, j=0, \dots, n_\beta, k=0, \dots, n_2$. Besides it is a big set of $(n_\alpha + 1)(n_\beta + 1)(n_2 + 1)$ elements, it can not ad-

dress the transformations made by parameter values $\rho_{2,j}$ and $\rho_{2,j+1}$. To solve both problems we again determine coefficients of fitting polynomial which approximate $\{^{0,0}\beta_0, \dots, ^{n_\alpha, n_\beta}\beta_{n_\beta}\}$. The polynomials are now denoted as $^{i,j}P_{n_\gamma}({}^{i,j}\gamma_k; \rho_2)$, where ρ_2 is variable, and $^{i,j}\beta_k$, $i=0, \dots, n_\alpha$, $j=0, \dots, n_\beta$, $k=0, \dots, n_\gamma$. All these $^{i,j}\gamma_k$ coefficients can be organized in 3D matrix

$$\Gamma = \begin{matrix} & \dots & ^{0,0}\gamma_{n_\gamma} & \dots & ^{0,n_\beta}\gamma_{n_\gamma} & \dots \\ \begin{matrix} ^{0,0}\gamma_0 & \dots & ^{0,n_\beta}\gamma_0 & \dots & \vdots \\ \vdots & \ddots & \vdots & \dots & \vdots \\ ^{n_\alpha,0}\gamma_0 & \dots & ^{n_\alpha,n_\beta}\gamma_0 & \dots & \dots \end{matrix} & \dots & \dots & \dots & \dots & \dots \end{matrix} \quad (6)$$

The matrix Γ has $(n_\alpha + 1)(n_\beta + 1)(n_\gamma + 1)$ coefficients which is much smaller than $(n_\alpha + 1)(n_\beta + 1)(n_2 + 1)$. The procedure above can be applied to some new task parameter, defined in the same fashion like the former two.

The computation of the model output, i.e. inverse kinematic solution, goes backwards. By determining the values of the parameters that define uncertainties of the task, ρ_1^* and ρ_2^* , all of the polynomials $^{i,j}P_{n_\gamma}$ are computed for $\rho_2 = \rho_2^*$ and obtained values are associated to coefficients $^i\beta_j^*$. These coefficients are used for computing the polynomials $^iP_{n_\beta}$, each producing the coefficient α_j^* . Coefficients α_j^* determine the polynomial over variable τ

$$\alpha_0^* + \alpha_1^* \tau + \dots + \alpha_{n_\alpha}^* \tau^{n_\alpha} \quad (7)$$

as an approximation of joint motion for one joint. Note that computation of coefficients α_j^* is done prior motion is getting starting and during the motion equation (7) is tabulated only for current time.

A comparison of the SA to other similar methods like Neural Networks and Fuzzy controllers, are discussed in [3]. Here, we just point out that the SA procedure requires generally smaller number of examples for modeling, it produces highly compressed and structurally simplified model, it generalizes well in both interpolation and extrapolation and it produces analytical output.

Besides benefits of SA based modeling, two important issues remain to be investigated. First, convergence of the modeling procedure should be guaranteed by some criterion. Here, we made decisions on the model dimension mostly by heuristics. Second, a possibility of inverse model addressing imposes truly flexible and efficient model, and should be further investigated.

3 Example

We discuss a task of press loading along with two subtasks, Fig. 2, with three major objects in robot's environment: press, conveyor, and unloading table. Several typical uncertainties can be expected. First, on a conveyor, a pick-up location is uncertain due to motion direction of the particle, and changeable velocity of conveyor. Fixed pick-up point will require additional feeder, new sensor set and very robust control. Second, unloading position can be also unpredictable. If we expect some other tasks to be fulfilled, like obstacle avoidance, specified orientation of the gripper, etc., the robot applied should be redundant. So, we have chosen a 4DOF jointed, planar robot, with lengths $l_1 = 0.65$, $l_2 = l_3 = 0.4$ and $l_4 = 0.2$. Given the position and orientation of the tip, it has one redundant DOF.

Instead of expensive vision system, several sensors are engaged to coordinate the robot in the environment. First, we detect the velocity of the conveyor either by tacho or incremental encoder with velocity estimation. Second, reflective optical sensor array positioned above the conveyor enables precise detection of the part on the conveyor. Two information, velocity and sensor index number, are sufficient to anticipate piece location and define pick-up position in which the robot gripper should be positioned and properly oriented. On the table, encoder as sensor gives an information used in table control and unloading location of the robot's tip.

Fig. 2. Sketch of the robot, conveyor, press and unloading table. The path $\Pi: A \rightarrow B \rightarrow C \rightarrow A \rightarrow C \rightarrow D \rightarrow A$ is given by solid line.

In this particular task, we parameterize four subtasks: pick-up task from the conveyor ($A \rightarrow B$), press loading task ($B \rightarrow C$), press unloading task ($C \rightarrow D$), and return to ready position task ($D \rightarrow A$). This task may be rather cumbersome to program, particularly if some optimization should be applied in trajectory planning that will engage the redundancy, and should be further altered according to task uncertainties. The parameterization of the task is the key issue of flexibility of the RR model.

The two task parameters in the first and the second subtasks are Conveyor Cruise Velocity parameter, $p_{CCV} \in [k_{\min}, k_{\max}]$, and Index of Activated Sensor parameter, $p_{IAS} \in [0, l_{\max}]$. The parameter p_{CCV} is defin-

ed on the interval of velocities and scales up a minimal velocity $v = v_{\min} p_{CCV}$. The parameter p_{IAS} determines the position of the piece along the width of the conveyor $l = l_{\min} + \Delta_{IAS} p_{IAS}$. So, the parameterized position of the piece within pick-up zone, shaded rectangle in Fig. 2, is determined by

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 0.75 \\ -1.1 \end{bmatrix} + \begin{bmatrix} -v_{\min} p_{CCV} t_f \\ \Delta_{IAS} p_{IAS} \end{bmatrix} \quad (8)$$

The two parameters are used in trajectories definition on A→B and B→C subtasks.

Variations of the press unloading and return to ready position subtasks are achieved by means of two parameters: Radial Displacement $p_{RD} \in [0, r_{\max}]$, which defines position relative to the center of the unloading table, and where $r_{\max}$ is maximal radial displacement, and Azimuth Displacement $p_{AD} \in [\alpha_{\min}, \alpha_{\max}]$ parameter, where $(\alpha_{\min}, \alpha_{\max})$ are normalized minimal and maximal angular displacement from table's mediane which defines orientation angle from the specified direction (see Fig. 2). Hence, the parameterized position of the unloading table is defined by

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 0.75 \\ 0.5 \end{bmatrix} + (0.1 + p_{RD}) \begin{bmatrix} \sin(p_{AD} \cdot 10^\circ) \\ \cos(p_{AD} \cdot 10^\circ) \end{bmatrix} \quad (9)$$

The two parameters, defining the pie-like shaded region in Fig. 2, efficiently parameterize the segments C→D and D→A of the path Π . On the parameterized task trajectories, we apply an appropriate RR procedure, which: a) keeps end-effector orientation as specified, b) resolves the redundancy to enable time-optimal motion between the corner points and c) enable cyclicity of joint motions along the closed path Π .

In the pick-up zone we choose 11×11 examples to be used in modeling procedure, regarding the two parameters, $p_{CCV} \in \{1, \dots, 11\}$, and $p_{IAS} \in \{0, \dots, 10\}$, with sensor resolution $\Delta_{IAS} = 0.02 \text{ m}$. Finally, there are 121 examples in parameter space along both, A→B and B→C segments of the path Π . In the unloading task, RR is modeled on a set of examples, defined by $p_{AD} \in \{-5, \dots, 5\}$, and $p_{RD} \in \{1, \dots, 8\}$. It means that on the unloading table parameterization gives 11×8 = 88 examples on both, C→D and D→A segments of the path. We chose all these numbers of examples based on empirical evaluation which avoids overgeneralization as well undergeneralization later in modeling, without intention to find the best solution.

The solution of RR is organized as Jacobian pseudoinverse augmented by additional criteria of orientation and cyclicity. On the segment A→B, and B→C, the redundancy is resolved by pseudoinverse, involving feedback loops to cease computational errors [1,5]:

$$\dot{\mathbf{q}} = \mathbf{J}^T (\mathbf{J}\mathbf{J}^T)^{-1} (\dot{\mathbf{x}} + k_e \mathbf{e}), \quad (10)$$

where $\mathbf{e}$ is an error vector consisting of two subvectors of Cartesian and angular errors. Along segment D→A solution (10) is augmented with a term that in the ready-position point, A, guarantees the same manipulator configuration as

at the beginning of motion A→D. In such a way cyclicity condition is satisfied at the end of each end-effector's traversal A→...→A. Note that above RR is very awkward to be performed on-line. The main objective of the Example is to find the model that successfully interpolate and extrapolate parameterized joint motions obtained by RR resolution along the path Π , but with much less computational complexity than used by augmented pseudoinverse.

Now, the SA procedure starts with normalization in time to the $[-1, 1]$ interval, followed by generation of multidimensional coefficient set of fitting polynomials, Eq. (3, 4, 6), keeping the approximation error on cross-evaluation examples below the limit. The mean-squared error (MSE) defined as

$$\Delta err = \frac{1}{2} \sqrt{\Delta x^2 + \Delta y^2}, \quad (11)$$

where $\Delta x = \max_i |x_{d,i} - x_{SA,i}|$, $\Delta y = \max_i |y_{d,i} - y_{SA,i}|$, is used for evaluation of the approximation quality. Index d, i states for desired trajectory, and index SA, i states for the trajectory computed by the SA model.

Fig. 3. Interpolation MSE tested by cross-evaluation, between the examples within pick-up zone A→B.

Along the A→B segment, polynomials of $n_\alpha = 12$, $n_\beta = n_\gamma = 4$ degrees, Eq. (3, 4, 6), see Table 1, are chosen. The MSE on A→B segment are below 0.25 mm, Fig. 3. The number n_α is augmented by 2 which enables approximation polynomial (5) to match the first and the last point of requested path, which makes it perfect in positioning task.

Fig. 4. Interpolated MSE tested by cross-evaluation, between the examples within unloading zone C→D.

The degrees of polynomials used along the B→C segment of the path are given in Table 1. In other words, only 325 coefficients represent the RR model on B→C segment, per joint. The MSE is comparable to that along the A→B segment. The degrees of polynomials adopted for the SA procedure along the C→D segment of the path are given in Table 1, requiring only 3.13Kb of memory for location of joint model. Corresponding MSE is below 0.4 mm, as given in Fig. 4. Finally, on D→A segment, degrees of fitting polynomials are $n_\alpha=11$, $n_\beta=4$, $n_\gamma=8$. These polynomials also provide outstanding fitting, with average of 0.25 mm MSE.

Summarized review of degree of polynomials, along with corresponding size of the model per one robot joint, is given in Table 1, with 'coef.' computed as $(n_\alpha + 1)(n_\beta + 1)(n_\gamma + 1)$.

Table 1. Number of coefficients which make the SA model per one joint of the robot along the parameterized path II.

	A→B	B→C	C→D	D→A
n_α	12	12	9	11
n_β	4	4	4	4
n_γ	4	4	7	8
coef./joint	325	325	400	540
in Kb/joint	2.54	2.54	3.13	4.22

Fig. 5. Operational space errors along A→B* segment. Both errors are smooth, below 0.2 mm, and take zero values at the end of the segment.

A better understanding of SA-based model quality is obtained by comparing the SA-based RR with the RR solution as applied in examples generation. It will be done at an arbitrary points within pick-up region and within unloading region. We point out that RR modeling was performed on discrete set of examples, $p_{CCV} \in \{1, \dots, 11\}$, $p_{IAS} \in \{0, \dots, 10\}$. The obtained model of the RR, due to SA applied, enables generalization of the examples within the two zones. So, to illustrate the quality of the model, we chose one specific point, $B^* = (0.57, -1.0160)$, obtained by taking the values of task parameters $p_{CCV} = 4.5$, and $p_{IAS} = 4.2$. The operational errors are given in Fig. 5. Both errors are below 0.2 mm. Note that errors at the beginning and the end of the segment decrease to zero. Maximal

joint errors are very small, $\max\{\Delta q_i\} < \{1^\circ, 0.5^\circ, 0.5^\circ, 1^\circ\} \times 10^{-2}$.

On the same set of parameters we discuss the trajectory along B*→C segment of the path. Operational errors are below 0.15 mm, and decrease to zero at the end of the segment, Fig. 6. Joint space errors are also very small $\max\{\Delta q_i\} < \{1^\circ, 0.5^\circ, 0.5^\circ, 1^\circ\} \times 10^{-2}$.

Fig. 6. Operational space errors along the B*→C segment. The errors performing smooth variations, zero-boundary values, and are less than 0.15 mm.

Another test-point, $D^* = (0.6885, 0.2706)$, is defined by $p_{AD} = 1.5$, and $p_{RD} = 5.5$, Eq. (9). The errors along trajectories of the C→D* and D*→A segments are evaluated. Joint errors are small as in previous subtasks: $\max\{\Delta q_i\} < \{1^\circ, 0.5^\circ, 0.5^\circ, 1^\circ\} \times 10^{-2}$. The MSE on C→D segment are below 0.1 mm. Again, the errors at the end of the segment decrease to zero, Fig. 7.

Fig. 7. Operational space errors along the C→D* segment. The errors are below 0.1 mm. The errors vanish at the end of the segment.

Finally, the errors along the D*→A segment are below 0.1 mm and 0.05° . By keeping on zero the joint errors at the end of the segment D*→A, the cyclicity of the joint motions is maintained, Fig. 8. The external errors are extremely low, below 0.1 mm. Obtained results indicate a smooth, predictable errors, that guarantee the quality of fitting within the two zones.

Another, very important property of SA method is a possibility of extrapolation. We examine the extrapolation in one realistic case: increase of conveyor cruise velocity. We assumed that the CCV is increased by 40% over

maximal, $p_{CCV} \in [10, 14]$, which results in stretching of the pick-up zone. Cross-evaluation of the model shows errors below 1 mm, along both axes (Ox , and Oy). The extrapolation error along Ox axis is shown in Fig. 9.

Fig. 8. Joint space errors along the $D^* \rightarrow A$ segment. The errors are below 0.05° . The errors vanish at the end of the segment, which means that cyclicity of the path Π is maintained.

Fig. 9. Absolute maximal error compared with pseudoinverse solution along Ox axis on the segment $A \rightarrow B^*$ in the case of extrapolation.

Almost the same conclusion could be derived for the unloading segments of the path, but, due to limited space, this result is omitted. So, even in extrapolation case the model actually manages to guide the robot in the right place, maintaining the optimality criteria (time and orientation).

Fig. 10. Joint velocities obtained by simple derivation of polynomials approximating joint motions. To compare, dashed lines represent joint velocities computed by pseudoinverse.

The SA-based RR model provides analytical output in the form of a polynomial approximation of the desired joint motion, one per each joint. Being such, this set could be used in generation of joint velocities with a very small computational effort via method of synthetic division. Velocity histories in the case of $D^* \rightarrow A$ segment of both, pseudo-

inverse and SA-based RR, are given in Fig. 10. Almost perfect matching is achieved.

One of benefits of the SA procedure is reduced computational complexity of approximated joint motions. Generally, joint motions are computed only by operations of addition and multiplication, due to polynomials applied. Expressed in floating point operations (flops) it requires more than order of magnitude flops less than standard procedures of RR, which is burdened by transcendentals. For example, along $D^* \rightarrow A$ segment, it was necessary 3791 flops per one point of the path. At the same time, computation of RR by SA model requires only 104 flops per point of the path, assuming the coefficients α_i^* , Eq. (7) are computed. Thus, the reduction of computational complexity is enormous, 35.315 times. Furthermore, the SA procedure provides joint-oriented models, which means that in a case of decentralized robot control architecture, joint motions can be independently and simultaneously computed. This fact significantly reduces computational time. Roughly estimating, total number of flops divided by the number of joints. Bearing in mind that the errors along trajectories are rather low, generally below 1 mm and that they decrease to zero at the end of each segment (achieved by polynomial fitting), we can conclude that in similar tasks, the SA-based RR solution is more suitable way to handle redundancy in the presence of uncertainties.

4 Conclusion

We propose an efficient method for on-line control of redundant robots when the task is encumbered by uncertainties in position, orientation, velocity etc. The task is parameterized on the set of variables describing uncertainties and the model the RR is build by generalization of examples obtained by varying of the parameters. That parametric model is build off-line. Due to specific nature of the modeling procedure, we have achieved accurate RR, small model size, up to 10 Kb, significantly decreased computational efforts – 35 times less than standard RR procedures, also maintaining the optimality criterion used during examples generation. Maximal error along the segments of the path is less than 0.4 mm, but, by refining of the model, the positioning error is reduced to zero.

References

- [1] J.J. Craig, **Introduction to robotics – Mechanics and Control**, Addison Wesley, Reading MA, 1986.
- [2] G. Đorđević, M. Rašić, D. Kostić, and D. Šurdilović, "Learning of Inverse Kinematics Behavior of Redundant Robot", Proc. of 1999 IEEE ICRA, Detroit, Michigan, May 1999, pp. 3165-3170.
- [3] G. Đorđević, M. Rašić, and D. Kostić, "Representation of Robot Motion Control Skill", IEEE Trans. SMC Part C, Accepted for publication 2000.
- [4] G. Đorđević, V. Potkonjak, M. Rašić, D. Kostić, "Skill-Based Inverse Kinematics of Redundant Robots", 3rd ECPD ICARIAAS, Bremen, Sept. 1997, pp. 301-306.
- [5] Y. Nakamura, **Advanced Robotics: Redundancy and Optimization**, Addison-Wesley, 1991.