SERBIAN JOURNAL OF ELECTRICAL ENGINEERING
Vol. 18, No. 1, February 2021, 1-14

UDC: 007.52:004.94 **DOI: https://doi.org/10.2298/SJEE2101001K**

End-Effector Cartesian Stiffness Shaping - Sequential Least Squares Programming Approach

Nikola Knežević[1], Branko Lukić[1], Kosta Jovanović[1], Leon Žlajpah[2], Tadej Petrič[2]

Abstract: Control of robot end-effector (EE) Cartesian stiffness matrix (or the whole mechanical impedance) is still a challenging open issue in physical human-robot interaction (pHRI). This paper presents an optimization approach for shaping the robot EE Cartesian stiffness. This research targets collaborative robots with intrinsic compliance – serial elastic actuators (SEAs). Although robots with SEAs have constant joint stiffness, task redundancy (null-space) for a specific task could be used for robot reconfiguration and shaping the stiffness matrix while still keeping the EE position unchanged. The method proposed in this paper to investigate null-space reconfiguration's influence on Cartesian robot stiffness is based on the Sequential Least Squares Programming (SLSQP) algorithm, which presents an expansion of the quadratic programming algorithm for nonlinear functions with constraints. The method is tested in simulations for 4 DOF planar robot. Results are presented for control of the EE Cartesian stiffness initially along one axis, and then control of stiffness along both planar axis – shaping the main diagonal of the EE stiffness matrix.

Keywords: Cartesian Stiffness Control, Robot Redundancy, Physical Human-Robot Interaction, Sequential Last Squares Programming.

1 Introduction

In contrast to the rigid industrial robots, the robots driven by serial elastic actuators SEA are intended to work in human proximity [1], as service [2] or as medical aid devices [3] and for physical Human-Robot Collaboration [4]. All these robots somehow interact with humans; thus, safety is of the most importance. Covering robots with soft materials was one of the first and primitive methods for safe interaction. Lather on, safe interaction is achieved thought out implementation of control algorithms such as impedance control [5]. Unfortunately, soft materials could not achieve satisfying safety standards for not disrupted work in human proximity [6], while active controller could not react in

[1]School of Electrical Engineering, University of Belgrade Bulevar kralja Aleksandra 73, 11020 Belgrade, Serbia; E-mails: knezevic@etf.rs, branko@etf.rs, kostaj@etf.rs.
[2]Jožef Stefan Institute, Jamova cesta 39, 1000 Ljubljana, Slovenia; E-mails: leon.zlajpah@ijs.si, tadej.petric@ijs.si

time to absorb high amplitude impacts [7] that could lead to faster wear and tear of mechanical parts in stiff joint robots.

These shortcomings can be overcome with the use of compliant actuators. There are two types of compliant actuators: serial elastic actuators (SEAs) with constant stiffness [3, 6, 8]; variable stiffness actuators (VSAs) with a variety of realization [9, 10]. VSA presents the MIMO system; where beside position, actuator stiffness can be adjusted as well. Safe physical human-robot interaction (pHRI) is very important, thus there is a lot of effort invested and still investing in developing compliant actuators [11] mechanical designs and in finding suitable control strategies [12 − 16]. All for the purpose of enabling wider use of robots in the industrial environment and developing home service robots.

The focus of this paper is on passive end-effector (EE) Cartesian stiffness control of task redundant robots with SEAs. Exploiting the task redundancy through the robot reconfiguration (null-space movement) will allow certain control of EE stiffness [17 − 19]. The problem can occur when demanded EE stiffness matrix is not feasible and some of the elements of the commanded stiffness matrix are out of the physical boundaries. In this case, the robot movement in null space will present a trade-off between stiffness matrix elements for some selected optimization criteria. Lukić et al. in his work [18] presenting a methodology for finding robot pose that will satisfy the desired EE position and Cartesian stiffness dividing problem on two main tasks. The first task is to trace Cartesian position reference and the second task is to optimize EE Cartesian stiffness behavior. Because of using the gradient-based method, this optimization task can't guarantee the global optimum. This (sub)optimal solution can be further combined and complement with active compliant to achieve the desired EE stiffness matrix [19]. Finding the optimal solution (robot pose), in general, is an optimization of nonlinear function with constraints, where constraints are reflected in the limitation of joint positions and achieving the desired Cartesian position. A similar problematic for VSA is presented in [17, 19], where optimization is achieved through the adjusting joints stiffnesses. Optimal joint stiffnesses are computed with is the Karush-Kuhn-Tucker (KKT) theorem [17, 19], also known as KKT condition, for the nonlinear function optimization with constraints. The KKT condition is an algorithm for the optimization of nonlinear functions with constraints that seeks solution over the entire domain where the function is defined. In paper [20] authors present methodology and experimental test for detecting and reacting to a collision between a robot manipulator and a human. Margini et al. [21] propose two control schemes for impedance and direct force control base on contact location. In all these papers some control over Cartesian stiffness needs to be achieved. Because of that, we propose a methodology for the optimization of Cartesian stiffness behaviour to achieve desired Cartesian stiffness. In the end, the goal is to optimize robot pose during movement where it is expected that robot in consecutive (discrete) intervals on

its trajectory has a very "close" configuration that doesn't have to be globally optimal, but rather locally optimal. Thus, that is why an optimization algorithm that is more gradient-based, such as quadratic programming in form for nonlinear function: Sequential Least Squares Programming (SLSQP) algorithm is selected [22].

Section 2 presents a mathematical model for EE Cartesian stiffness and proposed optimization norm. SLSQP algorithm is described in Section 3, results are presented in Section 4 while conclusions and final remarks are pointed in Section 5.

2 Model of 4-DOF Planar Manipulator and Cartesian Stiffness Optimization

To optimize the robot's EE stiffness, the kinematic model of the manipulator is needed. Fig. 1 depicts the 4-DOF planar manipulator and its referent joint angles orientations.

Cartesian position of the robot EE, from Fig. 1, is defined with x, y coordinates positions, and θ orientation as follows:

$$x = l_1 \sin(q_1) + l_2 \sin(q_1 + q_2) + l_3 \sin(q_1 + q_2 + q_3) + l_4 \sin(q_1 + q_2 + q_3 + q_4), \quad (1)$$

$$y = l_1 \cos(q_1) + l_2 \cos(q_1 + q_2) + l_3 \cos(q_1 + q_2 + q_3) + l_4 \cos(q_1 + q_2 + q_3 + q_4), \quad (2)$$

$$\theta = q_1 + q_2 + q_3 + q_4, \quad (3)$$

where l_1, l_2, l_3, l_4 represents the lengths of links and q_1, q_2, q_3, q_4 represents the joint positions. The Jacobian matrix for the EE of 4-DOF planar manipulator from Fig. 1 is defined as

$$J = \begin{bmatrix} \frac{\partial x}{\partial q_1} & \frac{\partial x}{\partial q_2} & \frac{\partial x}{\partial q_3} & \frac{\partial x}{\partial q_4} \\ \frac{\partial y}{\partial q_1} & \frac{\partial y}{\partial q_2} & \frac{\partial y}{\partial q_3} & \frac{\partial y}{\partial q_4} \\ \frac{\partial \theta}{\partial q_1} & \frac{\partial \theta}{\partial q_2} & \frac{\partial \theta}{\partial q_3} & \frac{\partial \theta}{\partial q_4} \end{bmatrix}, \quad (4)$$

while joint's stiffness matrix K_j has a diagonal matrix form

$$K_j = \begin{bmatrix} k_{j_{11}} & 0 & 0 & 0 \\ 0 & k_{j_{22}} & 0 & 0 \\ 0 & 0 & k_{j_{33}} & 0 \\ 0 & 0 & 0 & k_{j_{44}} \end{bmatrix}, \quad (5)$$

where $k_{j_{ii}}$ is *i*-th joint stiffness.

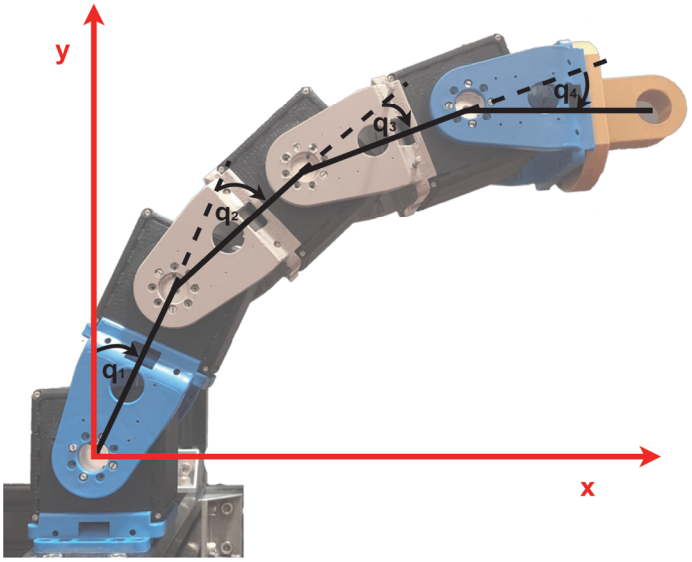

Fig. 1 – *The 4 DOF planar manipulator configuration with clockwise joint orientation.*

The Cartesian stiffness matrix K_c is the function of the joint stiffness matrix and Jacobian matrix

$$K_c = (J(q)K_j^{-1}J(q)^T)^{-1}, \tag{6}$$

where K_c is a symmetric 3×3 matrix and q is a 4-dimensional joint position vector. Based on the desired Cartesian stiffness matrix K_{cd} and achieved a Cartesian stiffness matrix K_c, the performance is given as follows

$$H = \left\| K_{cd} - K_c \right\|, \tag{7}$$

Where H is a user-selected matrix norm (i.e. Euclidean or Frobenius norm).

The task of the optimization algorithm is to minimize norm H, which value is being considered as the performance index. For simplicity of the problem, optimization along one axis is initially considered, followed by the simultaneous optimization of both elements of the Cartesian stiffness matrix main diagonal. As the objective function (function over which minimizing is performed) weighted Frobenius [17, 26, 27, 28] norm is used

$$H = \sqrt{A(K_{cd_{11}} - K_{c_{11}})^2 + B(K_{cd_{22}} - K_{c_{22}})^2}, \tag{8}$$

where coefficients A and B are weight elements used to prioritise one axis stiffness over the other. $K_{cd_{11}}$, $K_{cd_{22}}$, $K_{c_{11}}$ and $K_{c_{22}}$ are corresponding coefficients of the desired and achieved EE stiffness matrices that corresponds with stiffness along x-axis and y-axis, respectively. At the end of the optimization process, the EE position needs to be the same as the initial position. Thus, optimization constraints regarding the EE position must be included.

3 Optimization Using Sequential Least Square Programming

Cartesian stiffness shaping can be conducted using the SLSQP algorithm [22], which is developed for nonlinear optimization problems with constraints. In general, the gradient descent method can be used for nonlinear optimization also, even this method is very simple for implementation it is rarely used in practice due to slow convergence. Also, Newton's method can be used but if a user can afford memory and time resources [23]. The SLSQP algorithm uses the Han-Powell quasi-Newton [24] method for solving nonlinear optimization in combination with Quadratic Programming [25]. It is an iterative method in which both objective and constraints function need to be triple continuously differentiable. The method generates steps by reducing the nonlinear optimization problems (NLP) to Quadratic Programming (QP) subproblems. In this way, the time needed for convergence is reduced. The general form of the problem is given as follows

$$\min f(p), \text{ over } p \in R^n, \tag{9}$$

$$\text{subject to } h(p) = 0, \tag{10}$$

$$g(p) \le 0, \tag{11}$$

where $f : R^n \to R$ is an objective function, function $h : R^n \to R^m$ and $g : R^n \to R^z$ describe the equality and inequality constraints, n represents the number of variables over which optimization is performed, m is the number of equality, and z is the number of inequality constraints function. The NLP for a given iteration $p[k]$ is reduced to the QP subproblems, and then the next iteration $p[k+1]$ is obtained from the QP solution. This construction is done in such a way that the sequence $p[k]$ converges to a local minimum $f(p^*)$ of the NLP (9-12) as $k \to \infty$, where $*$ refers to the variable related to the local minimum point. In the presence of constraints, the analysis and implementation of NLP are much more complex.

The points that satisfy equality and inequality constraints are called a feasible set of the NLP and are represented with

$$\mathcal{F} = \{p \in R^n \,|\, h(p) = 0, g(p) \le 0\}. \tag{12}$$

The Lagrangian function of NLP is defined as

$$\mathcal{L}(p,\lambda,\mu) = f(p) + \lambda^T h(p) + \mu^T g(p)\,, \tag{13}$$

where vectors $\lambda \in R^m$ and $\mu \in R^z$ are referred to as Lagrangian multipliers.

Active constraints are an index set for $p \in R^n$ when

$$\mathcal{I}_{ac} = \{i \in \{1,\ldots,z\} \,|\, g_i(p) = 0\}\,. \tag{14}$$

Let $f(p^*)$ be a local minimum of the NLP, then the condition

$$g_i(p^*)\mu_i^* = 0\,, \quad 1 \le i \le z\,, \tag{15}$$

$$\mu_i^* > 0\,, \quad i \in \mathcal{I}_{ac}\,, \tag{16}$$

is called strict complementary slackness at p^*.

Matrix G is defined with the derivate of active constraints as $G(p) = (\nabla h_1(p),\ldots,\nabla h_m(p),\nabla g_{i_1}(p),\ldots,\nabla g_{i_j}(p))$, were ∇ represent the gradient operator and j representing the number of active constraints.

For $f(p^*)$ that is a local minimum of the NLP assume existing of λ^* and μ^* such that

$$\nabla\mathcal{L}\left(p^*,\lambda^*,\mu^*\right) = \nabla f(p^*) + \nabla h(p^*)\lambda^* + \nabla g(p^*)\mu^* = 0\,. \tag{17}$$

If (17) holds true, then it is called the first-order necessary optimality condition.

Assume that the following conditions are satisfied:

– The columns of $G(p^*)$ are linearly independent,

– Strict complementarity slackness holds at p^*,

– The Hessian of the Lagrangian with respect to p is positive definite on the null space of $G(p^*)^T$,

then (9) in combination with these conditions are called the second-order enough optimality conditions of the NLP. These optimality conditions guarantee that p^* is the point of a local minimum of NLP, and that the Lagrangian multipliers λ^* and μ^* are uniquely determined.

After defining the constraints that need to be satisfied for accepting some value as a minimum, it is necessary to construct the QP for each iteration step. The QP subproblems reflect the local properties of the NLP with respect to the current iterate $p[k]$. Therefore, the objective function f is replaced with its local quadratic approximation

$$f(p) \approx f(p[k]) + \nabla f(p[k])(p - p[k]) + \frac{1}{2}(p - p[k])Hf(p[k])(p - p[k]), \quad (18)$$

and constraint function g and h are replaced with their local affine approximations:

$$g(p) \approx g(p[k]) + \nabla g(p[k])(p - p[k]), \quad (19)$$

$$h(p) \approx h(p[k]) + \nabla h(p[k])(p - p[k]), \quad (20)$$

where $\nabla f(p)$ representing the gradient of function and $Hf(p)$ Hessian of function. In our use case, function *f* represent norm defined in (9), and variable *p* represent robot joint position vector $q = [q_1, q_2, q_3, q_4]^T$. Equality constraints are given with (1), (2) and (3), while inequality constraints are defined with joints positions motions.

4 Simulation Results

For validating this optimization technique, testing is conducted on a simulation model for the 4-DOF planar manipulator. Robot links lengths are set as $l_1 = l_2 = l_3 = l_4 = 0.2$ m. In the beginning, we optimize Cartesian stiffness along one axis only. After that, optimization of both elements on the Cartesian stiffness matrix main diagonal is performed. In all experiments, all joint stiffness is set to 500Nm/rad.

4.1 One axis optimization

The initial joints and EE positions are provided to the optimization algorithm. The joints stiffness and robot pose (through the Jacobian matrix) are in direct relation to achievable EE Cartesian stiffness, thus the computation of the range of achievable EE Cartesian stiffness is necessary.

For each of the two EE positions, $P1(x, y) = (0.07574\,\text{m}, -0.236\,\text{m})$ and $P2(x, y) = (0.16849\,\text{m}, -0.20115\,\text{m})$, Cartesian stiffness optimization is performed over the *y*-axis, while desired EE Cartesian stiffness values are picked from the feasible region.

Table 1 presents results from 4 experiments. Experiments were conducted in two different points of robot workspace with some initial configuration. There are also presented resulting robot configuration, achieved norm, and Cartesian stiffness. The optimization process was successful in finding the robot configuration that will ensure desired EE Cartesian stiffness while keeping the EE position unchanged.

Fig. 2 presents initial and final robot configuration for the point P2 when goal stiffness is set to 12500 N/m.

Table 1

Results for EE Cartesian stiffness matrix optimization along the y-axis.

Experiment	Robot position	Initial joint config. [rad]	Resulting joint config. [rad]	Norm value	Stiffness: achieved (desired)
1	P1	–2 –1 3.6518 1.524	–3.1373 –1.4574 1.6627 2.5183	1.38e–4	10000 (10000)
2	P1	–2 –1 3.6518 1.524	–2.6899 –1.5276 2.4282 2.9360	7.09–5	8500 (8500)
3	P2	1.04719 0.62831 1.57079 1.0471	2.9539 –1.2305 1.9664 2.7833	0.002	9500 (9500)
4	P2	1.04719 0.62831 1.57079 1.0471	1.0223 1.9019 1.8759 –2.5368	0.0022	12500 (12500)

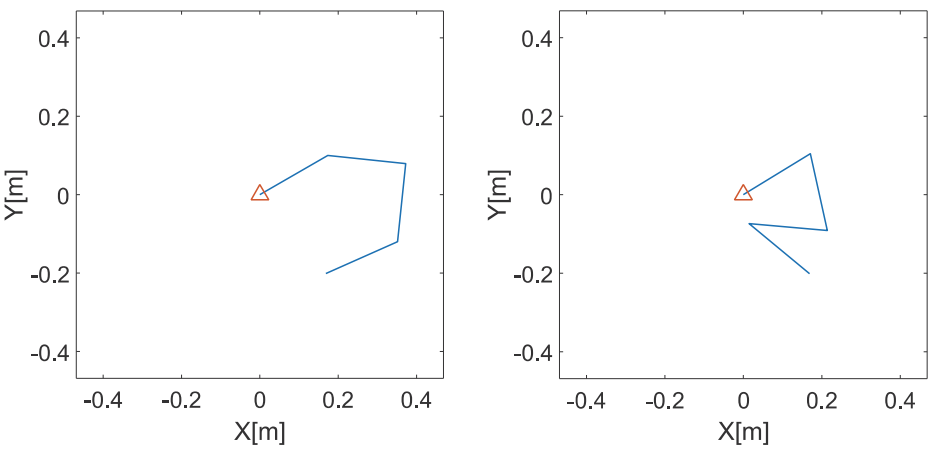

Fig. 2 – *Robot referent position P2 in fourth experiment (Table 1).*
Left: Initial robot configuration before optimization;
Right: resulting robot configuration after optimization.

4.2 Multiple axes optimization

When multiple axes optimization is performed and the desired Cartesian stiffness matrix is not feasible, then the algorithm needs to a trade-off between optimization over multiple axes to find a combined local minimum based on defined criteria (norm *H*). The simulation is performed in points, *P*1 and *P*2, same as in one axis optimization. In addition to one axis optimization, the desired Cartesian stiffness over x axis is added to the algorithm, also picked from its feasible region. Also, four simulations were run where both axes have the same weighted factor, and results are provided in **Table 2**. In simulations for position *P*1, the desired EE Cartesian stiffness was not achieved as in the previous attempts. Besides this, the achieved norm is one that corresponds to the local minimum of the optimization process. On Fig. 3 is shown combined norm for axis *x* and *y* norm as functions of q_1 and q_2. This is because configuration has 4 DOF, 2 needs for achieving the desired EE Cartesian position, and 2 DOF are left for additional optimization form both axes. Thus, for a feasible EE position for each set of q_1 and q_2, the q_3 and q_4 are uniquely defined.

Table 2

Results for Cartesian stiffness matrix optimization along x and y-axis (without cross elements of stiffness matrix).

Experiment	Robot position	Initial joint config.	Resulting joint config.	Norm value	Stiffness: Achieved (desired)
1	P1	–2 –1 3.6518 1.524	–2.6745 –1.5182 2.5449 2.9280	492.33	20417; 9262 (20000; 9000)
2	P1	–2 –1 3.6518 1.524	–2.5496 –1.8051 2.8173 3.1416	165.25	18663; 14027 (18500; 14000)
3	P2	1.04719 0.62831 1.57079 1.0471	0.3720 1.2165 1.7136 0.2087	0.0027	10000; 14000 (10000; 14000)
4	P2	1.04719 0.62831 1.57079 1.0471	–1.3597 2.3813 1.7190 –0.2143	0.002	12000; 14000 (12000; 14000)

Fig. 3 represent achievable Cartesian stiffness of 4 DOF planar manipulator with fixed EE position in the function of first two joint positions (in this case, Cartesian stiffness in exact EE position depends on the position of first two joints while the position of third and fourth joints depends on EE position). The white region represents infeasible robot configuration, while the yellow color on graphic present robot configuration whit the highest norm value and dark blue color represent the lowest norm values.

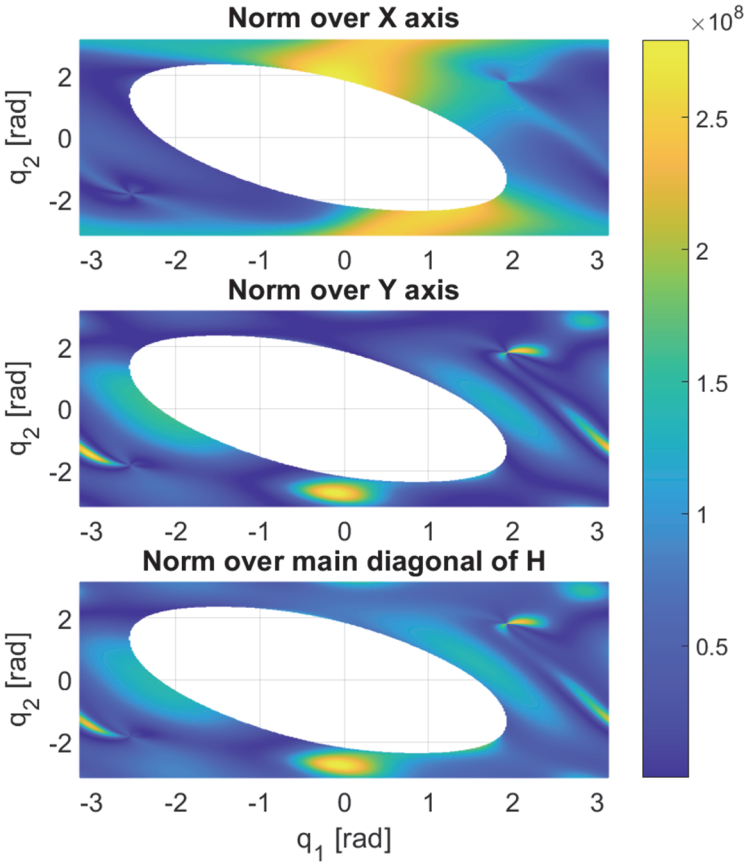

Fig. 3 – *Norm over x-axis (top), norm over y-axis (middle), and combined norm over x and y axis (bottom), where darker blue colour presents norm that is closer to zero. Stiffness along both axes have the same priority (A=B=1).*

From Fig. 3 is obvious that differences in norm exist when different axis optimization activates. Also, it can be shown that this robot kinematic configuration norm possesses more than one local minimum and the result of the optimization algorithm could compute one that is not global. If the right initial configuration was chosen algorithm can succeed to find a global minimum. Also,

from Fig. 3 correct initial configuration can be chosen in order to ensure global minimum optimization. Although the problem was stated for the 4-DOF robot defining the position of first and second joint and EE position the third and fourth joint was uniquely defined. If $q_1 = -1.5\,\text{rad}$ and $q_2 = 2.5\,\text{rad}$, the optimization algorithm will finish with the norm $H = 0.00027$.

4.2 Weighted multiple axes optimization

In order to favorize one axis stiffness over another, weight coefficients values need to be changed. If the coefficient has a higher value, then the stiffness along the corresponding axis is more favored. Optimization with different weighted coefficients, for the configuration used in the first simulation from **Table 2**, is obtained. The results are provided in **Table 3**. It can be noticed that increasing the values of the weight coefficient along one axis will make achieved stiffness along that axis closer to the desired one.

Table 3

Results for Cartesian stiffness matrix optimization along x and y-axis (without cross elements of stiffness matrix) with different weighted coefficients.

Experiment	Robot position	Initial joint config.	Weighted coef.	Norm value	Stiffness: Achieved (desired)
1	P1	–2 –1 3.6518 1.524	A=1 B=1	6.63e+3	18626; 14357 (12000;14000)
2	P1	–2 –1 3.6518 1.524	A=16 B=1	1.3e+3	12023; 15303 (12000;14000)
3	P1	–2 –1 3.6518 1.524	A=1 B=16	6.66e+3	18659; 14053 (12000;14000)

5 Conclusion

The paper introduces Sequential Least Squares Programming algorithm as an efficient solution to the EE stiffness shaping of the robot with passive compliant actuators. The presented algorithm is demonstrated for utilization in the real-time control/shaping of the EE stiffness due to short convergence time. Considered the algorithm as an optimization technique, which minimize deviations of the desired EE stiffness (by the null-space reconfiguration) while

keeping desired EE position, the stiffness shaping is successfully achieved for exact desired stiffness matching in the achievable stiffness range or as a shaping of the stiffness ellipse which depicts relative ratio of the stiffness in perpendicular axes. However, because of its property to finding local minimum the different initial configuration can lead to a non-global or different solution as a compromise to its short convergence time.

In the following research, authors will extend the same approach to EE stiffness shaping of the robots with VSAs, where variable joint stiffness presents additional DOFs that will extend the feasible region of robot EE Cartesian stiffness. Results will be validated in the experimental setup with an industrial robot equipped with a force/torque sensor as a source of perturbation to the presented articulated soft robot. Therefore, the efficiency of the applied methodology would be directly estimated from the applied force and deviation from the EE equilibrium position of the soft robot.

6 Acknowledgment

This research was supported by the Science Fund of the Republic of Serbia, PROMIS, #6062528, ForNextCobot and ARRS project PhRoCiety (N2-0130).

7 References

[1] N. Pedrocchi, F. Vicentini, M. Malosio, L. M. Tosatti: Safe Human-Robot Cooperation in an Industrial Environment, International Journal of Advanced Robotic Systems, Vol. 10, No. 1, January 2013, pp. 1 – 13.

[2] A. De Santis, B. Siciliano, A. De Luca, A. Bicchi: An Atlas of Physical Human-Robot Interaction, Mechanism and Machine Theory, Vol. 43, No. 3, March 2008, pp. 253 – 270.

[3] H. Yu, S. Huang, G. Chen, Y. Pan, Z. Guo: Human-Robot Interaction Control of Rehabilitation Robots with Series Elastic Actuators, IEEE Transactions on Robotics, Vol. 31, No. 5, October 2015, pp. 1089 – 1100.

[4] A. Edsinger, C. C. Kemp: Human-Robot Interaction for Cooperative Manipulation: Handing Objects to One Another, Proceedings of the 16th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN 2007), Jeju, South Korea, August 2007, pp. 1167 – 1172.

[5] N. Hogan: Impedance Control: An Approach to Manipulation: Part II – Implementation, Journal of Dynamic Systems, Measurement, and Control, Vol. 107, No. 1, March 1985, pp. 8 – 16.

[6] S. Robla-Gómez, V. M. Becerra, J. R. Llata, E. González-Sarabia, C. Torre-Ferrero, J. Pérez-Oria: Working Together: A Review on Safe Human-Robot Collaboration in Industrial Environments, IEEE Access, Vol. 5, November 2017, pp. 26754 – 26773.

[7] S. Haddadin, A. Albu-Schäffer, G. Hirzinger: Safe Physical Human-Robot Interaction: Measurements, Analysis and New Insights, Proceedings of the 13th International Symposium on Robotics Research (ISRR), Vol. 66, Hiroshima-shi, Japan, November 2007, pp. 395 – 407.

[8] D. W. Robinson, J. E. Pratt, D. J. Paluska, G. A. Pratt: Series Elastic Actuator Development for a Biomimetic Walking Robot, Proceedings of the IEEE/ASME International Conference on Advanced Intelligent Mechatronics, Atlanta, GA, USA, September 1999, pp. 561 – 568.

[9] G. Grioli, S. Wolf, M. Garabini, M. Catalano, E. Burdet, D. Caldwell, R. Carloni, W. Friedl, M. Grebenstein, M. Laffranchi, D. Lefeber, S. Stramigioli, N. Tsagarakis, M. van Damme, B. Vanderborght, A. Albu-Schaeffer, A. Bicchi: Variable Stiffness Actuators: The User's Point of View, The International Journal of Robotics Research, Vol. 34, No. 6, March 2015, pp. 727 – 743.

[10] B. Vanderborght, A. Albu-Schäffer, A. Bicchi, E. Burdet, D. G. Caldwell, R. Carloni, M. Catalano, O. Eiberger, W. Friedl, G. Ganesh, M. Garabini, M. Grebenstein, G. Grioli, S. Haddadin, H. Hoppner, A. Jafari, M. Laffranchi, D. Lefeber, F. Petit, S. Stramigoli, N. Tsagarakis, M. Van Damme, R. Van Ham, L. C. Visser, S. Wolf: Variable Impedance Actuators: A Review, Robotics and Autonomous Systems, Vol. 61, No. 12, December 2013, pp. 1601 – 1614.

[11] R. Van Ham, T. G. Sugar, B. Vanderborght, K. W. Hollander, D. Lefeber: Compliant Actuator Designs, IEEE Robotics and Automation Magazine, Vol. 16, No. 3, September 2009, pp. 81 – 94.

[12] G. Buondonno, A. De Luca: Efficient Computation of Inverse Dynamics and Feedback Linearization for VSA-Based Robots, IEEE Robotics and Automation Letters, Vol. 1, No. 2, July 2016, pp. 908 – 915.

[13] M. Trumić, K. Jovanović, A. Fagiolini: Decoupled Nonlinear Adaptive Control of Position and Stiffness for Pneumatic Soft Robots, The International Journal of Robotics Research, February 2020, pp. 1 – 19.

[14] G. Palli, C. Melchiorri, A. De Luca: On the Feedback Linearization of Robots with Variable Joint Stiffness, Proceedings of the IEEE International Conference on Robotics and Automation, Pasadena, CA, USA, May 2008, pp. 1753 – 1759.

[15] V. Potkonjak, B. Svetozarevic, K. Jovanovic, O. Holland: The Puller-Follower Control of Compliant and Noncompliant Antagonistic Tendon Drives in Robotic Systems, International Journal of Advanced Robotic Systems, Vol. 8, No. 5, November 2011, pp. 143 – 155.

[16] B. Lukic, K. Jovanović, T. B. Šekara: Cascade Control of Antagonistic VSA – An Engineering Control Approach to a Bioinspired Robot Actuator, Frontiers in Neurorobotics, Vol. 13, September 2019, pp. 1 – 15.

[17] A. Albu-Schaffer, M. Fischer, G. Schreiber, F. Schoeppe, G. Hirzinger: Soft Robotics: What Cartesian Stiffness Can Obtain with Passively Compliant, Uncoupled Joints?, Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Sendai, Japan, September 2004, pp. 3295 – 3301.

[18] B. Lukić, T. Petrič, L. Žlajpah, K. Jovanović: KUKA LWR Robot Cartesian Stiffness Control Based on Kinematic Redundancy, Proceedings of the 28th International Conference on Robotics in Alpe-Adria-Danube Region (RAAD 2019), Kaiserslautern, Germany, June 2019, pp. 310 – 318.

[19] F. Petit, A. Albu-Schäffer: Cartesian Impedance Control for a Variable Stiffness Robot Arm, Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), San Francisco, CA, USA, September 2011, pp. 4180 – 4186.

[20] S. Haddadin, A. Albu-Schaffer, A. De Luca, G. Hirzinger: Collision Detection and Reaction: A Contribution to Safe Physical Human-Robot Interaction, Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Nice, France, September 2008, pp. 3356 – 3363.

[21] E. Magrini, F. Flacco, A. De Luca: Control of Generalized Contact Motion and Force in Physical Human-Robot Interaction, Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Seattle, WA, USA, May 2015, pp. 2298 – 2304.

[22] D. Kraft: A Software Package for Sequential Quadratic Programming, 1st Edition, DFVLR, Cologne, 1988.

[23] J. Goodman: Newton's Method for Constrained Optimization, Mathematical Programming Vol. 33, No. 2, November 1985, pp. 162 – 171.

[24] H.- S. Chen, M. A. Stadtherr: Enhancements of the Han – Powell Method for Successive Quadratic Programming, Computers & Chemical Engineering, Vol. 8, No. 3-4, 1984, pp. 229 – 234.

[25] A. Floudas, V. Visweswaran: Handbook of Global Optimization – Quadratic Optimization, Edited by R. Horst, P. M. Pardalos, Vol. 2, Springer, Boston, 1995.

[26] A. L. Custódio, H. Rocha, L. N. Vicente: Incorporating Minimum Frobenius Norm Models in Direct Search, Computational Optimization and Applications, Vol. 46, No. 2, June 2010, pp. 265 – 278.

[27] M. J. D. Powell: Least Frobenius Norm Updating of Quadratic Models that Satisfy Interpolation Conditions, Mathematical Programming, Vol. 100, No. 1, May 2004, pp. 183 – 215.

[28] G. Sridharan, W. Yu: Beamformer Design for Interference Alignment Using Reweighted Frobenius Norm Minimization, Proceedings of the IEEE 15th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC), Toronto, Canada, June 2014, pp. 469 – 473.