

Soft Robotics: What Cartesian Stiffness Can We Obtain With Passively Compliant, Uncoupled Joints?

A. Albu-Schäffer, M. Fischer, G. Schreiber, F. Schoeppe and G. Hirzinger

German Aerospace Center - DLR

Institute for Robotics and Mechatronics

82230 Wessling, Germany

Email: [Alin.Albu-Schaeffer, Max.Fischer, Guenter.Schreiber, Florian.Schoeppe]@dlr.de

Abstract—In the field of service robotics, whole arm contact with an unstructured environment or human beings becomes a major issue. Therefore soft robots, which means robots with passively (or mechanically) compliant joints, become more and more important. In this work we analyze what Cartesian stiffness at the tool center point one can achieve with a passively compliant, redundant robot with variable joint stiffness. We restrict this work to the special case of uncoupled joint stiffness only, as coupling of joint stiffness seems to be mechanically difficult to realize. Finally we discuss a Cartesian controller which incorporates the compliance of the joints and ensures the correct stiffness behavior also for high displacements from the desired position.

I. INTRODUCTION

In the field of service robotics, compliance and reactivity to unforeseen contacts in a priori unknown environments with the whole arm, not only with the tool center point behind a force torque sensor (FTS), are a major issue. Today there are basically three ways to establish compliance: (1) Use a FTS, possibly in combination with a passively compliant device close to it; (2) build a stiff robot with joint torque sensors and implement compliant behavior through control at joint level; or (3) construct a *soft robot* with mechanically (or passively) compliant joints with variable joint stiffness. The stiffness can be varied by an additional actuator in each joint.

In situations where control and precision in the position domain are less important and the focus is on compliance, soft robots (3) become more and more interesting, as they:

- cause less impact on their environment, especially on human beings, in the case of contact,
- are less prone to structural damage in case of contact, especially in non-nominal situations (power failure; controller failure; sensor failure etc.). This holds especially true for small and lightweight structures as e.g. artificial hands or humanoid robots.
- yield better compliant performance, as they inherently and instantaneously show a compliant behavior, whereas (1) and (2) can only react compliantly with a time delay through the loop *sensor data processing, control and actuation*.

There are already a few interesting examples of robots realizing an adjustable mechanical joint stiffness [14], [6], [9]. Concerning safety aspects of passively compliant robots, [17], [2] made some important contributions. In this work we would like to give a theoretical analysis and an optimization solution for the problem of realizing a desired Cartesian stiffness at the tool center point (TCP) of

a passively compliant robot and touch some control issues that result from drawbacks of passively compliant robots.

II. PROBLEM STATEMENT

With a soft robot, we would like to realize a *desired Cartesian stiffness* at the tool tip. While for some applications, the adjustment of the stiffness on joint level may be sufficient, a specification of Cartesian stiffness is required for various tasks such as assembly of parts or sliding along surfaces. This desired Cartesian stiffness should be realized in a manner, that a force (or torque) input along specified principal stiffness axes should result in a linear position (orientation) displacement along these axes also for a larger displacement from the initial position.

To realize this behavior, we can use the *mechanically variable joint stiffnesses* and - in case of a redundant robot - the *null space* of the robot [12].

Additionally it needs to be stated that when talking about passively compliant robots, we imply that the robot's joints are decoupled in the sense of compliance. Of course, one could establish coupling of joint stiffnesses mechanically¹, but this is beyond the scope of this paper. This leads to the first constraint in this work:

Diagonal Joint Stiffness Matrix: *Joint stiffness matrices (\mathbf{K}_J) have no coupling terms, they are always of diagonal shape in this work.*

For a general robotic application it is most natural to specify the desired stiffness behaviour of the robot in Cartesian coordinates. The desired values would result from a task description with respect to the TCP. The user may specify a desired constant stiffness matrix $\mathbf{K}_C = -\frac{\partial \mathbf{f}}{\partial \mathbf{x}} \in \mathbb{R}^{m \times m}$ as the relation between the Cartesian wrench \mathbf{f} and the Cartesian displacement \mathbf{x} . Here, m is the number of Cartesian degrees of freedom (DoF). On the other hand, the robot will be able to adjust the stiffness $\mathbf{K}_J = -\frac{\partial \boldsymbol{\tau}}{\partial \mathbf{q}} \in \mathbb{R}^{n \times n}$ of its n joints, where $\boldsymbol{\tau}$ is the joint torque and \mathbf{q} is the joint position. Hence the Cartesian stiffness has to be transformed to the joint space using a mapping \mathcal{T} : $\mathbf{K}_J = \mathcal{T}(\mathbf{K}_C)$, an approach known as stiffness control. This transformation can be computed as follows:

$$\begin{aligned} \mathbf{K}_J &= -\frac{\partial \boldsymbol{\tau}}{\partial \mathbf{q}} = -\frac{\partial (\mathbf{J}(\mathbf{q})^T \mathbf{K}_C \Delta \mathbf{x})}{\partial \mathbf{q}} \\ &= \mathbf{J}(\mathbf{q})^T \mathbf{K}_C \mathbf{J}(\mathbf{q}) - \frac{\partial \mathbf{J}(\mathbf{q})^T}{\partial \mathbf{q}} \mathbf{K}_C \Delta \mathbf{x} \quad (1) \end{aligned}$$

¹This kind of coupling is realized in the human limbs.

\mathbf{J} denotes the manipulator Jacobian $\mathbf{J}(\mathbf{q}) = \frac{\partial \mathbf{f}(\mathbf{q})}{\partial \mathbf{q}}$, where $\mathbf{f}(\mathbf{q})$ is the forward kinematics mapping. $\Delta \mathbf{x} = \mathbf{x}_d - \mathbf{x}$ is the Cartesian displacement between the desired and the actual position. If the stiffness is computed around the equilibrium position, (i.e. $\Delta \mathbf{x} = \mathbf{0}$) then (1) reduces to

$$\mathbf{K}_J = \mathbf{J}(\mathbf{q})^T \mathbf{K}_C \mathbf{J}(\mathbf{q}). \quad (2)$$

This relation was generally used in the early work on stiffness control [11]. The importance of the second term in (1), which reflects the position dependent change of the Jacobian is pointed out in [7], [3]. Notice that the mapping \mathcal{T} and the matrix \mathbf{K}_J in (1),(2) have only local meaning and will change with the robot configuration (see also sec. V).

As mentioned in the introduction, the robot has the ability to adjust mechanically the stiffness in each joint, what means that a diagonal joint stiffness is available: $\mathbf{K}_J = \text{diag}\{k_{Ji}\} \in \mathbb{R}^{m \times m}$. In general, \mathbf{K}_J in (1) is a full matrix, what means that only an approximation of the desired Cartesian stiffness can be achieved.

Problem statement: *Given a desired Cartesian robot configuration $\mathbf{x}_d \in \mathbb{R}^m$ and a desired Cartesian stiffness matrix $\mathbf{K}_C \in \mathbb{R}^{m \times m}$ for that configuration, find the optimal joint position \mathbf{q}_d and the diagonal joint stiffness matrix $\mathbf{K}_J \in \mathbb{R}^{m \times m}$ which provides the best approximation of \mathbf{K}_C under the constraint $\mathbf{x}_d = \mathbf{f}(\mathbf{q}_d)$.*

The implementation of the second term in (1) using the passive joint compliance seems difficult, since it needs displacement $\Delta \mathbf{x}$ and this in turn would require the fast and continuous change of the joint stiffness during motion, what is in contrast to the main design idea. Therefore, eq. (2) will be used for computation of the desired passive joint compliance. In sec. V an approach will be introduced, which enables the combination of the variable joint stiffness with a slower Cartesian control loop. This slower loop may also implement the remaining part of the stiffness transformation (1).

The inverse problem of computing the resulting Cartesian stiffness matrix $\mathbf{K}_C = \mathcal{T}^{-1}(\mathbf{K}_J)$ for a given \mathbf{K}_J can be easiest solved by considering compliance matrices, which are the inverses of the stiffness matrices ($\mathbf{C}_C = \mathbf{K}_C^{-1}$ and $\mathbf{C}_J = \mathbf{K}_J^{-1}$), under the assumption of a positive definite \mathbf{K}_C and \mathbf{K}_J . The same reasoning as in (1), (2) leads to

$$\mathbf{C}_C = \mathbf{J}(\mathbf{q}) \mathbf{C}_J \mathbf{J}(\mathbf{q})^T \quad (3)$$

and hence to

$$\mathbf{K}_C = (\mathbf{J}(\mathbf{q}) \mathbf{K}_J^{-1} \mathbf{J}(\mathbf{q})^T)^{-1}. \quad (4)$$

Remark: *The same result can be obtained directly from (2) by using a pseudoinverse matrix $\mathbf{J}(\mathbf{q})^+$ satisfying $\mathbf{J} \mathbf{J}^+ = \mathbf{I}$ in order to compute*

$$\mathbf{K}_C = \mathbf{J}(\mathbf{q})^{+T} \mathbf{K}_J \mathbf{J}(\mathbf{q})^+. \quad (5)$$

The pseudoinverse is generally given by $\mathbf{J}(\mathbf{q})^+ = \mathbf{A}^{-1} \mathbf{J}(\mathbf{q})^T (\mathbf{J}(\mathbf{q}) \mathbf{A}^{-1} \mathbf{J}(\mathbf{q})^T)^{-1}$, with \mathbf{A} being a positive definite matrix. Notice that for the given problem, the mapping between the spaces of torques and the space of (infinitesimal) displacements is uniquely defined by \mathbf{K}_J^{-1} . Therefore, $\mathbf{A} = \mathbf{K}_J$ has to be chosen as a metric tensor in the pseudoinverse. The usage of an arbitrary positive definite matrix \mathbf{A} would lead to a wrong stiffness transformation.

III. ANALYSIS OF ROBOTS WITH COMPLIANT UNCOUPLED JOINTS

A. Analysis of a 2 DoF planar finger

In order to get a first feeling for our problem, the simplest, 2DoF case of a planar finger will be analyzed in this section. Suppose, the robot is fixed at a given configuration. We want to determine the limitations imposed on the Cartesian stiffness by choosing a diagonal joint stiffness $\mathbf{K}_J = \text{diag}\{k_{J1}, k_{J2}\}$. Eq. (5) becomes in this case

$$\begin{aligned} \mathbf{K}_C &= \mathbf{J}^{+T} \begin{pmatrix} k_{J1} & 0 \\ 0 & k_{J2} \end{pmatrix} \mathbf{J}^+ \\ &= \alpha \cdot \mathbf{J}^{+T} \begin{pmatrix} 1 & 0 \\ 0 & \beta \end{pmatrix} \mathbf{J}^+ \end{aligned} \quad (6)$$

where the notations $\alpha = k_{J1}$ and $\beta = \frac{k_{J2}}{k_{J1}}$ have been used. This can be rewritten in the following form:

$$\begin{aligned} \mathbf{K}_C &= \alpha \mathbf{P}_{J^+}(\beta)^T \begin{pmatrix} \lambda_1(\beta) & 0 \\ 0 & \lambda_2(\beta) \end{pmatrix} \mathbf{P}_{J^+}(\beta) \\ &= \mathbf{P}_{J^+}(\beta)^T \begin{pmatrix} \alpha \lambda_1(\beta) & 0 \\ 0 & \alpha \lambda_2(\beta) \end{pmatrix} \mathbf{P}_{J^+}(\beta) \end{aligned} \quad (7)$$

where $\lambda_1(\beta)$, $\lambda_2(\beta)$ are the eigenvalues which determine the stiffness along the principal stiffness axes.² These axes are given by the matrix $\mathbf{P}_{J^+}(\beta)$ which contains the eigenvectors of \mathbf{K}_C . As a direct consequence of (7) it turns out that *the direction of the principal axes and the ratio of the stiffness values along these axes can not be chosen independently from each other.* Both the eigenvalues $\lambda_1(\beta)$, $\lambda_2(\beta)$ and the matrix $\mathbf{P}_{J^+}(\beta)$ are parameterized by the same scalar β . α is linearly scaling both eigenvalues, keeping their ratio and the principal directions constant.

Figure 1 displays an example for this result. The Cartesian stiffness matrices are described by three parameters: the eigenvalues λ_1 and λ_2 and the angle θ between the coordinate system of the TCP and the coordinate system defined by the principal stiffness axes. The implementable stiffness matrices \mathbf{K}_C form a two dimensional manifold in this three dimensional space, parameterized by k_{J1} and k_{J2} . The intersection of this manifold with any horizontal plane ($\theta = \theta_0$) is a straight line, due to the previously mentioned property. Fig. 2 gives a two dimensional representation of the results. It displays the Cartesian stiffness values which can be obtained with a diagonal upper bounded joint stiffness matrix as a function of the rotation angle θ between the TCP and the principal axes. One can see that there are regions for θ which cannot be chosen.

These results reveal a severe limitation of the method, since independent stiffness values along the different axes are crucial for the use of stiffness control in most applications. It turns out that special attention has to be paid to the design of the manipulator and to the choice of the optimal configuration and stiffness transformation, in order to obtain good results in practice.

An obvious solution to the problem is to increase the number of joints of the planar finger. Since the joint stiffness

²In this simple example only the two dimensional case of translational stiffness is considered. Since the configuration space is here a subset of \mathbb{R}^2 , the Euclidian norm in \mathbb{R}^2 can be naturally considered for the forces and displacement spaces and the concepts of orthogonality, principal stiffness axes, etc. are thus well defined. In the general, six dimensional Cartesian case, the intuitive generalization of these concepts has to be treated with care.

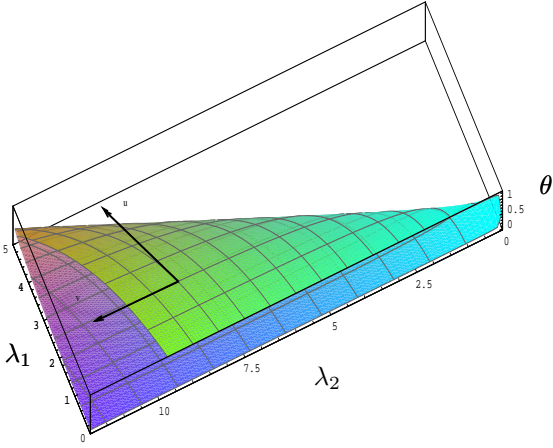


Fig. 1. Achievable stiffness with a 2 DoF robot

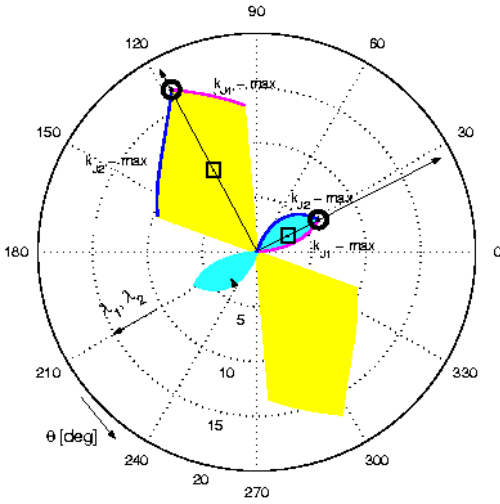


Fig. 2. Achievable Cartesian stiffness for $k_{J1}, k_{J2} \in (0, k_{\max}]$. Darker leaves: values for λ_1 , lighter leaves: values for λ_2 . The principal axes of stiffness are drawn for $\theta = 27.6713^\circ$, $\beta = 1$, what results in $\lambda_2/\lambda_1 = 2.6283$. The values for $\alpha = 1$ are marked with a circle, while the values for $\alpha = 0.5$ are indicated by a rectangle.

in (2) is symmetric, we need three independent variables to parameterize it. Adding a third joint without passive compliance would already introduce an additional degree of freedom through the null-space motion, which may solve the problem. This would be the minimal requirement concerning the number of independent parameters. If the joint has also a variable stiffness, a further tuning parameter is available. Nevertheless, the surjectivity of the mapping \mathcal{T}^{-1} for only diagonal joint matrices is still not guaranteed. These results obtained for this simple planar case give the motivation for a detailed analysis and a solution for the general 6 DoF case.

B. Specification of Cartesian Stiffness in 3D

1) *Structure of Cartesian Stiffness Matrices:* An arbitrary Cartesian stiffness matrix can be decomposed in four blocks: The translational and rotational stiffness part \mathbf{K}_T and \mathbf{K}_R , resp. and two coupling matrices, \mathbf{K}_{TR} and \mathbf{K}_{RT} . \mathbf{K}_T and \mathbf{K}_R can be written as diagonal stiffness matrices λ_T and λ_R in a rotated coordinate system with the rotations

\mathbf{P}_T and \mathbf{P}_R in the translational and rotational stiffness dimensions, respectively.

$$\begin{aligned} \mathbf{K}_C &= \begin{pmatrix} \mathbf{K}_T & \mathbf{K}_{RT} \\ \mathbf{K}_{TR} & \mathbf{K}_R \end{pmatrix} \begin{pmatrix} \mathbf{P}_T^T \lambda_T \mathbf{P}_T & \mathbf{K}_{RT} \\ \mathbf{K}_{TR} & \mathbf{P}_R^T \lambda_R \mathbf{P}_R \end{pmatrix} \\ &= \begin{pmatrix} \mathbf{P}_T^T & 0 \\ 0 & \mathbf{P}_R^T \end{pmatrix} \begin{pmatrix} \lambda_T & \bar{\mathbf{K}}_{RT} \\ \bar{\mathbf{K}}_{TR} & \lambda_R \end{pmatrix} \begin{pmatrix} \mathbf{P}_T & 0 \\ 0 & \mathbf{P}_R \end{pmatrix} \end{aligned} \quad (8)$$

In order to specify a desired Cartesian stiffness \mathbf{K}_{CD} it is sufficient (and customary) for many practical applications to specify (1) translational and rotational stiffnesses in the same (task relevant) coordinate system \mathbf{P} and to (2) omit coupling terms between translational and rotational stiffnesses (although one might think of some ‘‘screwing’’ applications where these couplings could make sense).

$$\mathbf{K}_{CD} = \begin{pmatrix} \mathbf{P}^T \lambda_T \mathbf{P} & 0 \\ 0 & \mathbf{P}^T \lambda_R \mathbf{P} \end{pmatrix} \quad (9)$$

Remark: We will use here this restricted way of specifying the stiffness for purpose of simplicity and intuitiveness. However, this restriction is not required in any further step of the optimization process and is exploited only in the definition of the intuitive error measure in sec. IV. This measure is used only for the visualization and interpretation of the optimization results. Hence, any way of specifying a complete p.d. Cartesian stiffness matrix (e.g. based on twist and wrench representation, or using a potential function in order to generate a force field [8], [4], [16], [13]) may be used instead.

2) *Number of Required DoFs:* In this section, an estimation of the minimal number of DoF is given, which is required for the implementation of an arbitrary Cartesian stiffness matrix with a passively compliant robot. The number of degrees of freedom needed to describe an arbitrary Cartesian stiffness in 3D is found by looking at the symmetrical³ 6×6 stiffness matrix \mathbf{K}_C : There are 21 (upper right or lower left triangular matrix) parameters to be specified. For the passively compliant robot with diagonal joint stiffness we have two independent variables for each joint (joint angle and joint stiffness), but we need 6 DoFs to achieve a specified TCP pose. Thus we get from n joints

$$DoF_{\text{stiffness}} = 2 \cdot n - 6$$

DoFs to realize \mathbf{K}_C . This means that we need a 14 DoF robot to get at least the degrees of freedom needed to obtain 21 ‘‘stiffness’’ DoFs, which - of course - doesn’t guarantee surjectivity of the mapping from joint positions and stiffnesses to Cartesian stiffness.

3) *Limitations for the Cartesian stiffness of robots with passively compliant joints:* There are two main sources of errors for the Cartesian stiffness we identified so far:

- Due to the diagonal form of the \mathbf{K}_J , the desired Cartesian stiffness can not be realized exactly.
- Because of the locality of the Jacobian the desired behavior is only local: A larger deflection due to a given force will yield a significant position error if the Jacobian and as a result the joint stiffness is not updated.

The first problem will be tackled in the next section and the second problem in sec. V.

³As mentioned in sec. II, we are considering here the case of small deflections, where the symmetry holds. For the treatment of high deflections see sec. V.

IV. THE TCP STIFFNESS OPTIMIZATION PROBLEM

BestLocalStiffnessApprox

As stated above, there is a very high probability that a desired Cartesian stiffness \mathbf{K}_{CD} cannot be achieved, mostly due to the lack of DoFs and - if we use a robot with at least 14 DoFs - due to the non-surjective mapping from joint stiffnesses to Cartesian stiffnesses. Therefore we can only find a good approximation instead of a perfect solution by formulating the problem as an optimization problem: *Mimize*

$$\text{norm}(\mathbf{K}_{CD}, \mathbf{K}_C(\mathbf{K}_J, \mathbf{q}))$$

subject to

$$\mathbf{f}(\mathbf{q}) = \mathbf{x}_d \quad (10)$$

This formulation raises three questions: (1) What function can we use as a suitable **norm**, (2) how hard is the optimization problem and (3) what tools can we use?

A. Choosing the appropriate norm

Obviously, the result of the optimization task stated above will depend on the type and the weighting of the norm used to calculate the optimization error. The choice of a norm (and of a corresponding weighting) should be seen rather as a degree of freedom in the design of the solution: by choosing a norm, one specifies in which sense the resulting approximation should be optimal, similar to the procedure in any optimal control problem. In particular applications, the choice of a norm may result from the specific problem formulation. However, we are interested here in an application independent solution, so different norms will be evaluated in terms of computation cost and intuitiveness of the parameterisation.

1) *Induced L_2 norm*: Assuming the Euclidian norm for the twist space, one could choose the induced matrix norm $\|\mathbf{K}_C\|_2 = \lambda_{\max}(\mathbf{K}_C)$, with $\lambda_{\max}(\mathbf{K}_C)$ being the maximal eigenvalue of \mathbf{K}_C . However, this solution is computationally expensive (compared to IV-A.3), not very intuitive (compared to IV-A.2) and questionable from geometrical point of view [10].

2) *Intuitive norm*: We tried to identify a set of intuitive error qualities and came up with an error vector

$$\Delta = (\Delta_T^{Rot}, \Delta_R^{Rot}, \Delta_T^{Rel}, \Delta_R^{Rel}, \Delta_K)^T \quad (11)$$

where Δ_T^{Rot} and Δ_R^{Rot} mean the angle axis rotations between the main axes coordinate systems of the desired and resulting translational and rotational stiffnesses, respectively,

$$\Delta_T^{Rel} = \frac{1}{3} \sum_{i=1}^3 \frac{|\lambda_{Ti} - \tilde{\lambda}_{Ti}|}{\frac{1}{2}(\lambda_{Ti} + \tilde{\lambda}_{Ti})}$$

$$\Delta_R^{Rel} = \frac{1}{3} \sum_{i=1}^3 \frac{|\lambda_{Ri} - \tilde{\lambda}_{Ri}|}{\frac{1}{2}(\lambda_{Ri} + \tilde{\lambda}_{Ri})}$$

are a measure for the relative errors of the stiffnesses (i.e. eigenvalues) λ of the translational and rotational dimensions and Δ_K is a measure for the coupling between translational and rotational stiffnesses. For Δ_K we simply use the largest absolute value of the elements of \mathbf{K}_{RT} and \mathbf{K}_{TR} (see eq. 8).

We used this error vector as is as an intuitive, human readable error representation. Instead of making a norm

out of this error vector (which basically means a weighted sum and implies the selection of a proper set of weighting factors), we used a different norm for the optimization algorithm. This also eases the optimization task dramatically:

3) **Weighted Frobenius- or Schur Norm.**: Let $\mathbf{A} = (a_{ij})$ be a $n \times n$ matrix. Let $\mathbf{G} = (g_{ij})$ (*weight matrix*) be a $n \times n$ matrix of positive scalars. The **weighted Frobenius- or Schur norm** of \mathbf{A} is defined by:

$$\|\mathbf{A}\|_F^G := \left\{ \sum_{i,j} g_{ij} |a_{ij}|^2 \right\}^{\frac{1}{2}}$$

By transforming \mathbf{G} to diagonal form (see eq. 9) we can choose weights for the coupling terms (\mathbf{K}_{RT} and \mathbf{K}_{TR}) (see eq. 8), for the errors of the translational and rotational stiffness coordinate system orientation (off-diagonal elements of \mathbf{K}_T and \mathbf{K}_R , resp.) and the translational and orientational stiffness values (diagonal elements of \mathbf{K}_T and \mathbf{K}_R , resp.).

An ad-hoc approach to the optimization problem would be to use a nonlinear optimization algorithm and optimize joint stiffnesses and the nullspace joint position simultaneously. Looking at the problem more in detail, it turns out that a separate optimization of joint stiffnesses and nullspace joint position makes sense.

B. Joint Stiffness Optimization

Let \mathbf{k}_J be the vector of joint stiffnesses of the diagonal matrix \mathbf{K}_J . By choosing the Frobenius norm as **norm** in the general optimization problem statement (eq. 10), we can assign a unique optimal joint stiffness to a given robot pose for a desired Cartesian stiffness:

$$\mathbf{k}_J^{Best}(\mathbf{q}) = \left\{ \mathbf{k}_J, \|\mathbf{K}_C(\mathbf{q}, \mathbf{k}_J) - \mathbf{K}_{CD}\|_F^G \rightarrow Min! \right\}$$

By searching the extremal of $\mathbf{k}_J(\mathbf{q})$ (i.e. differentiating the Frobenius norm w.r.t. the joint stiffnesses)

$$\frac{\partial}{\partial \mathbf{k}_{Ji}} \|\mathbf{K}_C(q_1, \dots, q_n, k_{J1}, \dots, k_{Jn}) - \mathbf{K}_{CD}\|_F^G = 0$$

we get a linear equation system with the unknowns k_{Ji} , $i \in \{1, \dots, n\}$ (since k_{Ji} enter quadratically the Frobenius norm):

$$\mathbf{A} \cdot \mathbf{k}_J = \mathbf{b}$$

The matrix \mathbf{A} and the right hand side \mathbf{b} depend on the robot pose \mathbf{q} , the desired Cartesian stiffness \mathbf{K}_J and the weight matrix G . So we can calculate \mathbf{k}_J^{Best} as

$$\mathbf{k}_J^{Best}(\mathbf{q}) = \mathbf{A}^{-1} \cdot \mathbf{b} \quad (12)$$

This extremely efficient joint stiffness optimization step can now be used as a sub-procedure in the following, unfortunately much more complicated optimization step for the nullspace of the robot.

C. Nullspace Optimization

At a given Cartesian position within the workspace of a redundant robot, there are an infinite number of joint configurations reaching the respective TCP position. The resulting space is the well-known nullspace.

So the question arises, whether there are optimal configurations meeting our desired Cartesian stiffness. Figure (3) shows a plot of the norm $B(\mathbf{q}) = \|\mathbf{K}_C^{Best}(\mathbf{q}) - \mathbf{K}_{CD}\|_F^G$

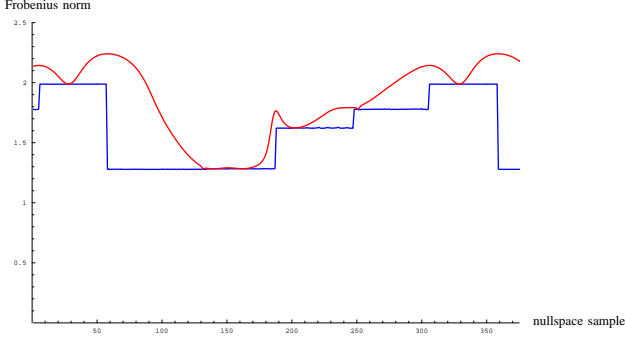


Fig. 3. Achievable stiffness quality during a walk through the nullspace of a 7 DoF robot; joint stiffness optimization only

optimized by (12), which was generated by equally sampling the nullspace of a seven DoF robot.

As one can see, the optimization function contains local minima. The darker line in figure (3) points out the reachable local minima from the position within the nullspace. Optimization is done by a gradient search along the following gradient:

$$\text{grad } B(\mathbf{q}^i) = \lim_{\lambda \rightarrow 0} \left(\sum_{j=1}^k \frac{(B(\mathbf{q}^i + \lambda \mathbf{n}_j) - B(\mathbf{q}^i))}{\lambda} \cdot \mathbf{n}_j \right)$$

whereas $N = \{\mathbf{n}_1, \dots, \mathbf{n}_k\}$ represents the nullspace of \mathbf{J} with the dimension $k = DoF - 6$ and \mathbf{q}^i is the position at sample i in the nullspace.

Due to the existence of the local minima, well-known local optimization methods, like the Moore-Penrose pseudoinverse, will lead to very suboptimal results. Therefore the optimization should be performed in less real-time critical planning layer, where other nonlinear optimization methods can be utilized. For the sake of fast computability, a cluster

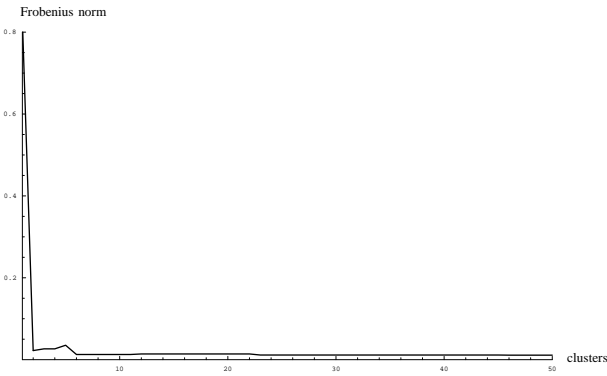


Fig. 4. Number of nullspace clusters vs. Cartesian stiffness error method was investigated, where the nullspace is sampled by a few number of representants. With those representants, the nullspace optimization is performed, and the minimum is selected. On an investigation about the required number of representants, one sees in figure (4), that the practical number of representants should be somewhere between 10 and 100 for a four dimensional nullspace (10 DoF robot). Figure (5) shows the achievable quality during a random walk through the nullspace of a 10 DoF Robot using the previously described methods.

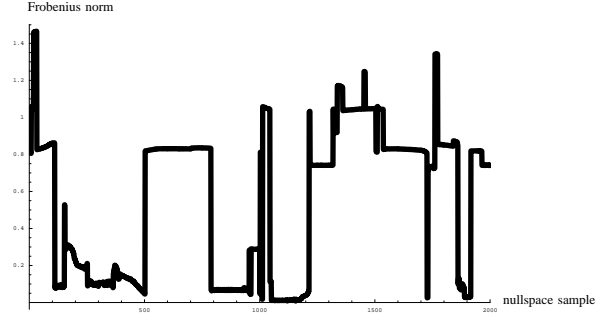


Fig. 5. Achievable stiffness quality during a random walk through the nullspace of a 10 DoF robot; joint stiffness and nullspace optimization

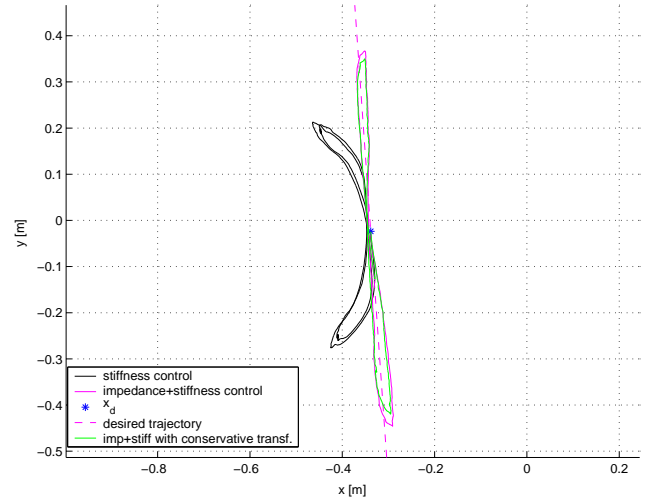


Fig. 6. Measurements for stiffness control and the version with an additional Cartesian impedance control: x-y plot. The stiffness in the dotted direction should be low, in the other directions high. The displacement is produced by a human user applying a force at the TCP.

V. CONTROL

As mentioned in sec. II, the stiffness mapping from joint space to Cartesian space is valid only locally. A first choice for implementing a Cartesian stiffness may be to use (2) for computing the desired joint stiffness matrix \mathbf{K}_J and command this stiffness, together with the desired position $\mathbf{q}_d = f^{-1}(\mathbf{x}_d)$ to the variable compliant joint. The torque in the joint springs would then be

$$\boldsymbol{\tau} = \mathbf{K}_J(f^{-1}(\mathbf{x}_d) - \mathbf{q}) - \mathbf{D}_J \dot{\mathbf{q}}, \quad (13)$$

where $\mathbf{D}_J \dot{\mathbf{q}}$ is a damping term, inherently available in the joint springs by design or added by control. This method leads however, at least for higher displacements $\Delta \mathbf{x}$ from the desired position, to substantial errors in the stiffness matrix. The reason for these errors is the local character of (1). The last term in (1) does not completely eliminate this error, because $\frac{\partial(\mathbf{K}_C \Delta \mathbf{x})}{\partial \mathbf{q}} = \mathbf{K}_C \mathbf{J}(\mathbf{q})$ is also valid only locally. Here, the Jacobian is implicitly assumed to be constant, despite of being multiplied by a big displacement $\Delta \mathbf{q}$. This error can be seen in fig. 6, where the joint stiffness is emulated on the DLR light-weight robot [5] by control. For the direction of low desired stiffness, the error increases with the displacement. For comparison, the results with an additional slower Cartesian loop, as will be presented in the next section, are also displayed.

A. Combination of the passive variable joint stiffness with a Cartesian controller

In this section, a method of dividing the impedance control problem into two levels is presented [1], which can make use of the elastic joint design concept:

- The first level is the level of the physical, variably tunable joint stiffness, equivalent to an infinite bandwidth control. It should have the form (13), while accounting for the local character of (1).
- The second layer is a Cartesian control level, (with a typical sampling rate of 1 to 6ms for the DLR robots), in which $\mathbf{J}(\mathbf{q})$, $\mathbf{f}(\mathbf{q})$ and a Cartesian controller may be computed. This controller should be designed to implement those parts which can not be assigned to the first level.⁴

In the following, the index ${}_C$ will be used for signals which are measured or computed in the Cartesian loop. Furthermore, for the difference signals, the notations $\Delta\mathbf{q}_C = \mathbf{q}_d - \mathbf{q}_C$ and $\Delta\mathbf{q}_J = \mathbf{q}_C - \mathbf{q}$ are used. Considering the above remarks, one can write:

$$\begin{aligned}\Delta\mathbf{x} &= \mathbf{x}_d - \mathbf{x} \approx \mathbf{x}_d - \mathbf{x}_C - \left. \frac{\partial \mathbf{x}}{\partial \mathbf{q}} \right|_{\mathbf{q}=\mathbf{q}_C} (\mathbf{q} - \mathbf{q}_C) = \\ &= \Delta\mathbf{x}_C + \mathbf{J}(\mathbf{q}_C)\Delta\mathbf{q}_J\end{aligned}\quad (14)$$

$$\mathbf{J}^T(\mathbf{q}) \approx \mathbf{J}^T(\mathbf{q}_C) + \Delta\mathbf{q}_J^T \left. \frac{\partial \mathbf{J}^T(\mathbf{q})}{\partial \mathbf{q}} \right|_{\mathbf{q}=\mathbf{q}_C} \quad (15)$$

The torque produced by an ideal Cartesian stiffness can be written as:

$$\begin{aligned}\tau_{dK} &= \mathbf{J}^T(\mathbf{q})\mathbf{K}_C\Delta\mathbf{x} \approx \mathbf{J}^T(\mathbf{q}_C)\mathbf{K}_C\Delta\mathbf{x}_C + \\ &+ \mathbf{J}^T(\mathbf{q}_C)\mathbf{K}_C\mathbf{J}(\mathbf{q}_C)\Delta\mathbf{q}_J + \\ &+ \Delta\mathbf{q}_J^T \left. \frac{\partial \mathbf{J}^T(\mathbf{q})}{\partial \mathbf{q}} \right|_{\mathbf{q}=\mathbf{q}_C} \mathbf{K}_C\Delta\mathbf{x}_C + \\ &+ \Delta\mathbf{q}_J^T \left. \frac{\partial \mathbf{J}^T(\mathbf{q})}{\partial \mathbf{q}} \right|_{\mathbf{q}=\mathbf{q}_C} \mathbf{K}_C\mathbf{J}(\mathbf{q}_C)\Delta\mathbf{q}_J\end{aligned}\quad (16)$$

The first term corresponds to the impedance controller on Cartesian control level. The second term corresponds to a stiffness controller, as described in [11]. Here, in contrast to (13), the joint stiffness acts only locally, in the vicinity of the last Cartesian position. This helps overcoming the slower Cartesian sampling rate. The third term corresponds to the correction term in (1), which ensures the conservativeness of the mapping. Finally, the fourth term depends on the square of the small displacement $\Delta\mathbf{q}_J$ and consequently has no practical significance.

Two alternatives may be considered for implementing (16), having in mind that in our robot, only a diagonal \mathbf{K}_J is available:

- If the commanded \mathbf{K}_C corresponds already to a diagonal matrix on joint level after the optimization from sec.IV, the Cartesian controller should implement the first and third term in (16). Depending on the implementation of the low level joint control, the

⁴Of course there is an additional control module, which has to take care of the low level control for the variable stiffness actuator. This includes the command of the motors to adjust the stiffness as well as the setting of the equilibrium point for the springs \mathbf{q}_d [15]. This part is not subject of the current paper.

torque may be commanded directly or in form of a desired joint displacement, using the known value of the elasticity. $\mathbf{J}^T(\mathbf{q}_C)\mathbf{K}_C\mathbf{J}(\mathbf{q}_C)$ is also computed in every Cartesian cycle and represents a constant desired stiffness, which is commanded to the joint for the duration of this step. The desired position during this step is \mathbf{q}_C .

- If an arbitrary \mathbf{K}_C is commanded, then also the off-diagonal terms of the stiffness controller have to be implemented by the software.

VI. SIMULATION RESULTS

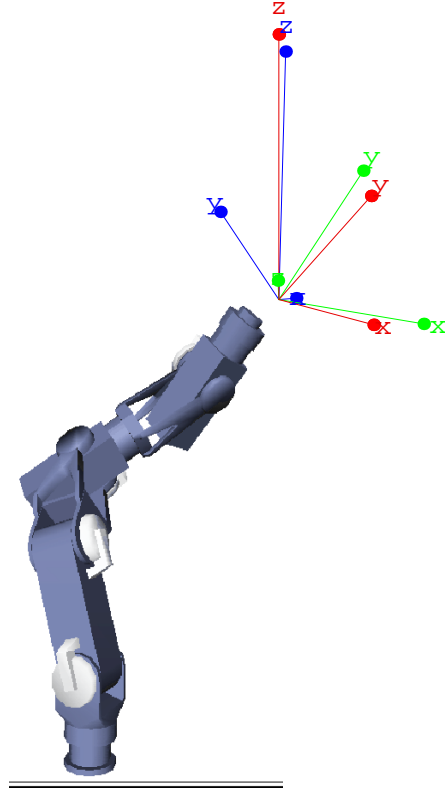


Fig. 7. Desired and resulting Cartesian stiffness for a 7 DoF robot. Medium gray (red) line: desired stiffness (identical frames and values were chosen for the translational and rotational part); light gray (green) frame: resulting Cartesian stiffness, translational part; dark gray (blue) frame: resulting Cartesian stiffness, rotational part.

Typical results of the method described in sec. IV for finding the best approximation for \mathbf{K}_C are shown in fig. 7 and fig. 8. A fixed Cartesian desired configuration and the same \mathbf{K}_C are commanded for a 7 DoF arm (fig. 7) and for the same arm mounted on a three DoF platform, leading to a 10 DoF system (fig. 8). The following weighting matrix was used for the Frobenius norm:

$$\mathbf{G} = \begin{pmatrix} 3 & 30 & 30 & 3 & 3 & 3 \\ 30 & 3 & 30 & 3 & 3 & 3 \\ 30 & 30 & 3 & 3 & 3 & 3 \\ 3 & 3 & 3 & 10 & 100 & 100 \\ 3 & 3 & 3 & 100 & 10 & 100 \\ 3 & 3 & 3 & 100 & 100 & 10 \end{pmatrix} \quad (17)$$

It can be seen that different values have been chosen for the different sub-matrices in (8). This is related to the different scaling of rotations and translations but is also used to weight the various errors from (11) corresponding to the requirements of a specific application. By increasing the

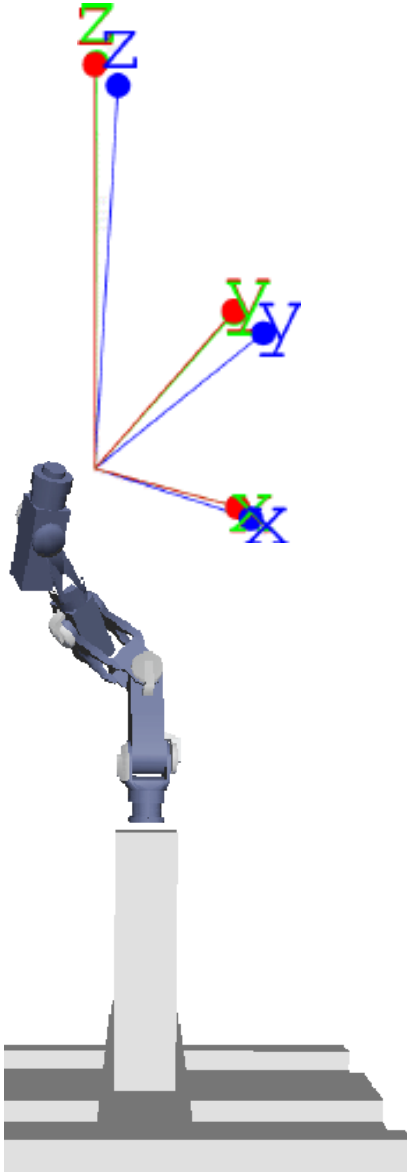


Fig. 8. Desired and resulting Cartesian stiffness for a 10 DoF robot. Medium gray (red) line: desired stiffness (identical frames and values were chosen for the translational and rotational part); light gray (green) frame: resulting Cartesian stiffness, translational part; dark gray (blue) frame: resulting Cartesian stiffness, rotational part.

off-diagonal weighting of K_T and K_R with respect to the diagonal elements, a better alignment of the principal axes is enforced, in detriment to an exact stiffness value along these axes.

The simulations showed that with the 7 DoF arm, the resulting approximation is in general very poor. In contrast, with the same arm mounted on a mobile platform, an acceptable accuracy can be achieved, although this system also does not have sufficient DoFs for an exact solution. The errors for the analyzed example are summarized in table I.

TABLE I
ERRORS OF THE CARTESIAN STIFFNESS APPROXIMATION.

	Δ_T^{Rot}	Δ_R^{Rot}	Δ_T^{Rel}	Δ_R^{Rel}	Δ_K
7 DoF	6.26°	41.83°	75.4 %	55.37 %	0.58
10 DoF	0.61°	6.71°	0.055 %	9.95 %	0.25

VII. CONCLUSIONS

The following conclusions can be drawn from this study: 1.) There are severe limitations concerning the implementation of Cartesian stiffness with an usual (6 or 7 DoF) arm with adjustable diagonal joint compliance. 2.) A mobile system with an arm placed on a vehicle or a humanoid torso has a considerably improved performance. 3.) The optimization problem can be divided into a fast (linear) optimization in the space of the joint stiffness values and a slow (nonlinear) optimization in the null-space of the manipulator. This second part can be computed only offline, in a planning phase for the working posture. 4.) The joint stiffness can be integrated into a Cartesian control scheme to provide the desired stiffness behaviour also for large displacements from the desired position.

Simple applications like “wiping a table” have very different requirements concerning the stiffness accuracy than challenging assembly tasks (e.g., peg in hole). While the paper provides a framework to weight and optimize the various error components, an absolute evaluation of the resulting quality can be done only in the context of the specific task.

REFERENCES

- [1] A. Albu-Schäffer and G. Hirzinger. Cartesian impedance control techniques for torque controlled light-weight robots. *Intl. Conf. on Robotics and Automation*, pages 657–663, 2002.
- [2] A. Bicchi, G. Tonietti and M. Bavaro, and M. Piccigallo. Variable stiffness actuators for fast and safe motion control. *11th International Symposium of Robotics Research (ISRR)*, oct. 2003.
- [3] S. Chen and I. Kao. Simulation of conservative congruence transformation conservative properties in the joint and cartesian spaces. *Intl. Conf. on Robotics and Automation*, pages 1283–1288, 2000.
- [4] E. D. Fasse. On the spatial compliance of robotic manipulators. *Journal of Dynamic Systems, Measurement and Control*, 119:839–844, 1997.
- [5] G. Hirzinger, A. Albu-Schäffer, M. Hähle, I. Schaefer, and N. Sporer. On a new generation of torque controlled light-weight robots. *Intl. Conf. on Robotics and Automation*, pages 3356–3363, 2001.
- [6] H. Kobayashi, K. Hyodo, and D. Ogane. On tendon driven robotic mechanisms with redundant tendons. *Int. J. of Robotics Research*, 17(5):561–571, 1998.
- [7] N. Hogan. Mechanical impedance of single- and multi- articular systems. In J.M. Winters and S. Woo, editors, *Multiple Muscle Systems: Biomechanics and Muscle Organization*, pages 149–163. Springer-Verlag, New York, 1990.
- [8] S. Huang and J. M. Schimmels. The eigenscrew decomposition of spatial stiffness matrices. *IEEE Transactions on Robotics and Automation*, 6(2):146–156, 2000.
- [9] K. Koganezawa and S. Ban. Stiffness control of antagonistically driven redundant d.o.f. manipulator. *IROS*, pages 2280–2285, 2002.
- [10] H. Lipkin and J. Duffy. Hybrid twist and wrench control for a robotic manipulator. *Trans. of the ASME*, 110:138–143, 1988.
- [11] J. K. Salisbury. Active stiffness control of a manipulator in cartesian coordinates. *19th IEEE Conference on Decision and Control*, pages 83–88, 1980.
- [12] F. Schöppe. Mathematische modellierung und numerische simulation von robotern mit passiv nachgiebigen gelenken. Master’s thesis, Tech. Univ. Munich, oct. 2003.
- [13] S. Stramigioli. *Modeling and IPC Control of Interactive Mechanical Systems: A Coordinate-free Approach*. Springer-Verlag, 2001.
- [14] S. Sugano. Human-robot symbiosis. *Workshop on Human-Robot Interaction, ICRA*, 2002.
- [15] G. Tonietti and A. Bicchi. Adaptive simultaneous position and stiffness control for a soft robot arm. *IROS*, pages 1992–1997, 2002.
- [16] M. Zefran and V. Kumar. Affine connections for the cartesian stiffness matrix. In *Intl. Conf. on Robotics and Automation*, Albuquerque, NM, 1997.
- [17] M. Zinn, O. Khatib, B. Roth, and J.K. Salisbury. A new actuation approach for human friendly robot design. *Int. Symp. on Experimental Robotics, Ischia*, 2002.