Springer

KSME

Journal of Mechanical Science and Technology 29 (7) (2015) 2685~2691
www.springerlink.com/content/1738-494x(Print)/1976-3824(Online)
DOI 10.1007/s12206-015-0515-1

# Experimental studies of control concepts for a parallel manipulator with flexible links †

Markus Burkhardt[1], Robert Seifried[2] and Peter Eberhard[1,*]

[1]*Institute of Engineering and Computational Mechanics, University of Stuttgart, Pfaffenwaldring 9, Stuttgart, Germany*
[2]*Institute of Mechanics and Ocean Engineering, Hamburg University of Technology (TUHH), Eißendorfer Straße 42, Hamburg, Germany*

(Manuscript Received December 5, 2014; Revised April 30, 2015; Accepted April 30, 2015)

**Abstract**

Control of flexible multibody systems, such as flexible manipulators, is a challenging task. This is especially true if end-effector trajectory tracking is aspired. On the one hand, these systems require a large number of generalized coordinates to describe their dynamical behavior accurately. On the other hand, only a small subset of these values can be measured or reconstructed on-the-fly. Hence, it is difficult, if not nearly impossible, to use a state controller. In addition, flexible systems are underactuated, i.e. they possess less control inputs than generalized coordinates. In case of a non-collocated output controller, which is the case for end-effector trajectory tracking, the closed loop of the system might lose passivity and is non-minimum phase. In order to achieve end-effector trajectory tracking, exact and approximate feed-forward controls can be applied. In this work, two different versions of such concepts are compared experimentally. These model-based concepts are computed off-line and they supply, next to the required input values, a $C^1$-continuous solution of the complete state vector which can be used for feedback control. If the system is non-minimum phase, a two-sided boundary value problem has to be solved and the solution includes a pre-actuation as well as a post-actuation phase. While the exact method incorporates all dynamical effects of the flexible multibody system, the approximate concepts neglect certain implications, for example the dynamical effects due to the flexibility. In addition to the presentation of the theoretical basics of the control approaches and the underlying models, this contribution addresses some of the crucial obstacles, which have to be overcome for the operation of the test bench, e.g., signal conditioning, state reconstruction and friction compensation. Since the installed sensors do not allow the direct measurement of the end-effector position, image tracking is used to judge the quality of the different control approaches.

*Keywords*: Experimental studies; Feed-forward control; Flexible multibody system; Servo constraints

## 1. Introduction

In this contribution, different control approaches for end-effector trajectory tracking of a parallel manipulator with highly flexible links are compared experimentally. The considered approaches are based on different model-based feed-forward control designs [1] which are supported by additional feedback controllers.

The paper is organized in the following way. Sec. 2 presents the parallel manipulator with flexible links, which is used to benchmark the two feed-forward controls. Next to the hardware components, an overview of the process control is given. In Sec. 3, the forward and inverse dynamics of the mechanical system are presented. The concept of servo constraints is used to include the trajectory tracking problem into the system dynamics. The solution of these augmented equations of motion, which has to be obtained from a boundary value problem, provides the feed-forward control trajectories. Sec. 4 is dedicated to the control concept and the signal processing needed to perform the experiments on the given hardware. In Sec. 5, the techniques needed to compare the quality of the proposed feed-forward control concepts are presented. Next to the used image tracking approach, the obligatory post-processing steps are introduced. In addition, the results of the experimental studies are presented and a comparison of the presented alternatives is given. The conclusions in the final section form the completion of this contribution.

## 2. A parallel manipulator with flexible links

The experimental studies are carried out on a custom-made parallel manipulator with flexible links. This test stand can be used for a wide variety of control tasks, because the hardware as well as the software is highly adjustable. In the following, a short overview of the available hardware components and the

*Corresponding author. Tel.: +49 711 685 66389, Fax.: +49 711 685 66400
E-mail address: peter.eberhard@itm.uni-stuttgart.de
† This paper was presented at the Joint Conference of the 3rd IMSD and the 7th ACMD, Busan, Korea, June, 2014. Recommended by Guest Editor Sung-Soo Kim and Jin Hwan Choi

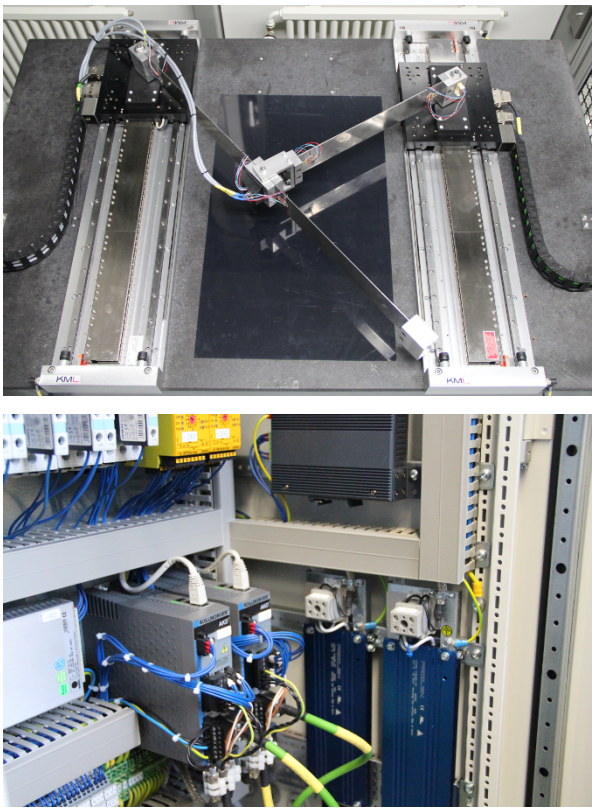


Fig. 1. Experimental setup of the parallel manipulator and the control box.

process control is given.

### 2.1 Experimental setup

The experimental platform, which can be seen in Fig. 1, consists of two identical linear direct drives which are mounted on a granite plate. Two revolute joints, each located on top of one direct drive, define the rotational axes of two highly flexible links with different length. An additional revolute joint connects the middle of the long link with the end of the short link. An additional mass which is attached to the free end of the long link defines the end-effector point. Hence, this parallel manipulator permits a non-redundant movement of the end-effector in the horizontal plane. The drive positions are measured with optical encodes. Five full bridges with four strain gauges each are attached to the elastic links. Three of these five bridges can be employed simultaneously to capture the elastic deformations.

### 2.2 Process control

The process control is carried out with a Speedgoat performance real-time target machine running the Mathworks xPCtarget kernel, which is called Simulink Real-Time since Matlab R2014a. Fig. 2 shows the flowchart of the process control. The embedded system implements the safety logic as well as the observers and controllers needed for the control

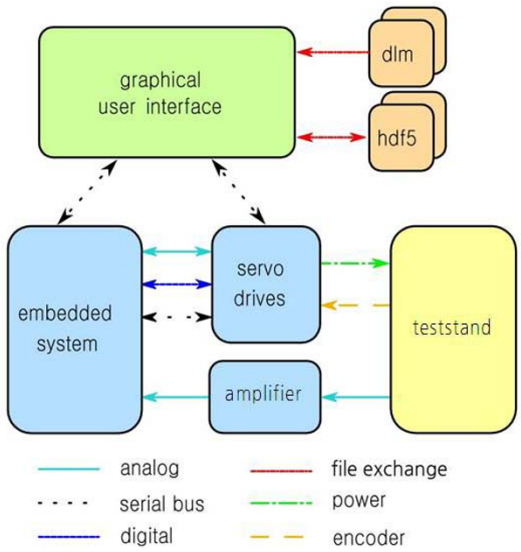



Fig. 2. Flowchart of the process control.

tasks. The current controllers are realized in hardware by means of two distinct Kollmorgen AKD servo drives, which offer high-bandwidth current-loops. The communication between the embedded system and the servo drives is carried out with digital and analog I/O-lines as well as with a serial bus according to the CANopen protocol following *IEC 61800-7-301*.

The human-machine interface is realized in software as a graphical user interface (GUI). The GUI is based on C/C++ and heavily relies on the application framework Qt4 [2]. It communicates with the embedded system via a proprietary TCP/IP-based protocol which is provided as a C-API by xpcTarget. The openly published communication protocol Modbus [3] is used to interact with the servo drives. The file management is carried out using the hierarchical data format (HDF5), which allows an efficient data exchange with Matlab. The target machine runs a multi-rate model to accommodate the different tasks and communication types. The basic sample time of 1.666 kHz is used for the analog I/O-lines, the control loop and the CANopen process data objects (PDO). This rather slow sampling rate is due to the limited numbers of messages that can be transmitted within a certain time interval. The digital I/O-lines and the safety logic are processed with a sampling rate of 500 Hz. These different sample times are executed in parallel on designated cores.

## 3. Forward and inverse dynamics

The control concepts that are used in this work require a detailed model of the parallel manipulator. Due to the fact that the time constant of the electrical part is very small compared to the mechanical part, the dynamics of the electrical part can be neglected. This assumption is valid for many electromechanical systems [4] and is adopted in this work. Therefore, it is only necessary to state the forward dynamics and the in-

verse dynamics of the mechanical subsystem. In addition, the friction forces, which act on the sliders, do not have to be included in the dynamical model. This can be justified by the fact, that the online computed friction compensation is capable to purge these effects, at least in theory.

### *3.1 Forward dynamics*

The mechanical part of the parallel manipulator is modeled as a flexible multibody system using the floating frame of reference approach. This approach is suitable for bodies that undergo a large working motion and experience only small elastic deformations. The large nonlinear motion of the body is described with absolute coordinates of a reference frame, whereas the deformation is described relative to this moving reference frame. For the integration into the multibody system, the absolute coordinates of the reference frame are expressed in minimal coordinates. In this contribution, the equations of motion are derived by applying the Newton-Euler equations together with D'Alembert's principle. For systems with a tree-like topology and $n_f$ degrees of freedom, this formalism yields the equations of motion in terms of a minimal set of generalized coordinates $\mathbf{y}$ which are linearly independent. However, if there are kinematic loops in the system, it might be impossible or disadvantageous to find an appropriate set of independent coordinates. In this case, the kinematic loops have to be cut open. This yields additional $n_c$ degrees of freedom, that have to be eliminated after the equations of motion are stated. Hence, the equations of motion are stated in terms of the dependent coordinates $\mathbf{z}$. Finally, algebraic constraint equations $\mathbf{c}(t,\mathbf{z})=\mathbf{0}$ and Lagrange multipliers $\boldsymbol{\lambda}$ are used to close the kinematic loops again. Thus, the set of differential-algebraic equations which describes the constrained dynamics of a general multibody system can be stated as

$$\mathbf{M}(\mathbf{z})\ddot{\mathbf{z}} = \mathbf{f}(\mathbf{z},\mathbf{z}) + \mathbf{B}(\mathbf{z})\mathbf{u} + \mathbf{C}(\mathbf{z})^{\mathrm{T}}\boldsymbol{\lambda}, \tag{1a}$$

$$\mathbf{0} = \mathbf{c}(\mathbf{z}). \tag{1b}$$

The dynamics stated in Eq. (1a) of the open-loop system is characterized by the generalized mass matrix $\mathbf{M}$, which is symmetric positive definite, and the generalized force vector $\mathbf{f}$, which contains the centrifugal, Coriolis and gyroscopic forces as well as those applied forces, which are not regarded as control inputs $\mathbf{u}$. For input-affine systems, the input matrix $\mathbf{B}$ is introduced. The transposed Jacobian matrix of the constraints $\mathbf{C}$ associates the Lagrange multipliers with the system dynamics. All needed computations are carried out with the research code Neweul-M$^2$ [5], which derives the equations of motion symbolically.

The topology of the parallel manipulator is shown in Fig. 3. The red parts are the sliders, which are modeled as rigid bodies with a mass of 6.3 kg each. The long link is modeled as an elastic body, which is obtained by a linear FE-model and reduced with a hybrid model-order reduction, which combines the classical Craig-Bampton approach with modern Gramian-matrix-based reduction techniques [6]. Afterward, Rayleigh damping is added. The short arm is regarded as a rigid body. Hence, the vector of dependent coordinates $\mathbf{z}$ consists of the drive positions $s_1$ and $s_2$, the angles $\alpha_1$ and $\alpha_2$ and the six elastic coordinates $\mathbf{q}_e$, which are assigned to the six mode shapes, which are shown in Fig. 4.

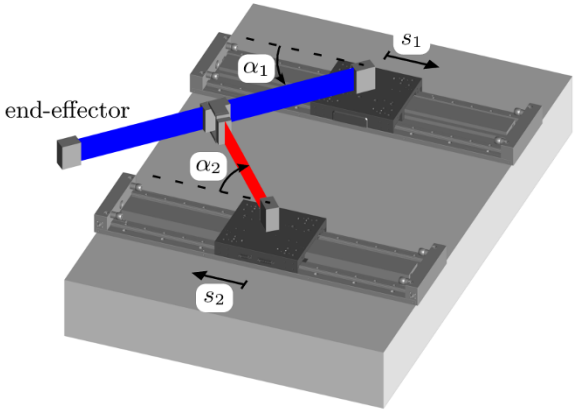



Fig. 3. Schematic model of the parallel manipulator.

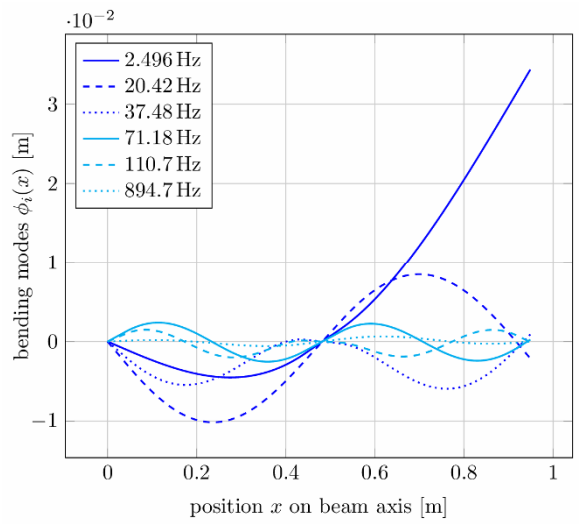



Fig. 4. Mode shapes of the isolated flexible body.

The kinematic loop due to the joint connecting the links is stated as the difference of the two position vectors $\mathbf{r}_{j,1}$ and $\mathbf{r}_{j,2}$. Hence, the two-dimensional constraint equations $\mathbf{c}$ on position, velocity, and acceleration level are expressed as

$$\mathbf{c}(\mathbf{z}) = \mathbf{r}_{j,2}(\mathbf{z}) - \mathbf{r}_{j,1}(\mathbf{z}) = \mathbf{0}, \tag{2a}$$

$$\dot{\mathbf{c}}(\mathbf{z},\dot{\mathbf{z}}) = \frac{\partial \mathbf{r}_{j,2}}{\partial \mathbf{z}}\dot{\mathbf{z}} - \frac{\partial \mathbf{r}_{j,1}}{\partial \mathbf{z}}\dot{\mathbf{z}} = (\mathbf{J}_{j,2} - \mathbf{J}_{j,1})\dot{\mathbf{z}} = \mathbf{C}\,\dot{\mathbf{z}} = \mathbf{0}, \tag{2b}$$

$$\ddot{\mathbf{c}}(\mathbf{z},\dot{\mathbf{z}},\ddot{\mathbf{z}}) = \mathbf{C}\ddot{\mathbf{z}} + \left(\frac{\partial \mathbf{J}_{j,2}\dot{\mathbf{z}}}{\partial \mathbf{z}} - \frac{\partial \mathbf{J}_{j,1}\dot{\mathbf{z}}}{\partial \mathbf{z}}\right)\dot{\mathbf{z}} = \mathbf{C}\,\ddot{\mathbf{z}} + \mathbf{c}'' = \mathbf{0}. \tag{2c}$$

Next to the reaction forces, which are provided by the Lagrange multipliers, the applied forces, namely the driving forces arising from the product of the force constants $K_{m,i}$ and

the motor currents $i_i$, which act on the sliders have to be considered. Due to the friction compensation, the friction forces do not have to be considered in this model.

### *3.2 Inverse dynamics*

Based on the dynamical model of the parallel manipulator, the inverse dynamics can be stated in a straight-forward way. By using the concept of servo constraints [7], the forward dynamics can be extended with additional $n_s$ constraint equations $\mathbf{s}(t,\mathbf{z})=\mathbf{0}$ describing the trajectory-tracking problem. In contrast to classical constraints, the control inputs $\mathbf{u}$ instead of the Lagrange multipliers are used to enforce the servo constraints. The solution of the so-called internal dynamics, which describes the dynamics that fulfills both the classical and the servo constraints, provides the feed-forward control. Depending on the stability of the internal dynamics, which depends on the stability of the system dynamics as well as on the input-output combination [8], a two-point boundary value has to be solved to obtain a bounded solution. A detailed description of this approach can be found in Ref. [9]. Due to the separation of the overall dynamics in two parts, namely the driven and the internal dynamics, it is not necessary for the solution of the boundary-value problem to specify the friction forces which act on the drives, since they do not affect the internal dynamics. The dry friction for example, which has to be modeled as a non-smooth force and, therefore, impedes the numerical solution, can be neglected in the solution phase. Instead, these friction forces are computed algebraically in a post-processing step or online in the form of a friction compensation [10]. The inverse dynamics of the parallel manipulator can be stated as a set of differential-algebraic equations according to

$$\mathbf{M}(\mathbf{z})\ddot{\mathbf{z}}=\mathbf{f}(\mathbf{z},\mathbf{z})+\mathbf{B}(\mathbf{z})\mathbf{u}+\mathbf{C}(\mathbf{z})^{\mathrm{T}}\boldsymbol{\lambda}, \tag{3a}$$

$$\mathbf{0}=\mathbf{c}(\mathbf{z}), \tag{3b}$$

$$\mathbf{0}=\mathbf{s}(t,\mathbf{z}). \tag{3c}$$

For end-effector trajectory tracking, the servo constraints can be expressed as the difference between the actual end-effector position in the plane $\mathbf{r}_{\mathrm{ee}}$ and the desired position $\mathbf{r}_{\mathrm{des}}$ yielding the servo constraints on position, velocity, and acceleration level

$$\mathbf{s}(t,\mathbf{z})=\mathbf{r}_{\mathrm{ee}}(\mathbf{z})-\mathbf{r}_{\mathrm{des}}(t)=\mathbf{0}, \tag{4a}$$

$$\dot{\mathbf{s}}(t,\mathbf{z},\dot{\mathbf{z}})=\frac{\partial\mathbf{r}_{\mathrm{ee}}}{\partial\mathbf{z}}\dot{\mathbf{z}}-\dot{\mathbf{r}}_{\mathrm{des}}=\mathbf{S}\,\dot{\mathbf{z}}+\mathbf{s}'=\mathbf{0}, \tag{4b}$$

$$\ddot{\mathbf{s}}(t,\mathbf{z},\dot{\mathbf{z}},\ddot{\mathbf{z}})=\mathbf{S}\,\ddot{\mathbf{z}}+\frac{\partial\mathbf{S}\,\dot{\mathbf{z}}}{\partial\mathbf{z}}\dot{\mathbf{z}}-\ddot{\mathbf{r}}_{\mathrm{des}}=\mathbf{S}\,\ddot{\mathbf{z}}+\mathbf{s}''=\mathbf{0}. \tag{4c}$$

Based on Eqs. (2c) and (4c), the system inputs and the Lagrange multipliers can be expressed as

$$\begin{bmatrix}\boldsymbol{\lambda}\\ \mathbf{u}\end{bmatrix}=-\begin{bmatrix}\mathbf{C}\mathbf{M}^{-1}\mathbf{C}^{\mathrm{T}} & \mathbf{C}\mathbf{M}^{-1}\mathbf{B}\\ \mathbf{S}\mathbf{M}^{-1}\mathbf{C}^{\mathrm{T}} & \mathbf{S}\mathbf{M}^{-1}\mathbf{B}\end{bmatrix}^{-1}\begin{bmatrix}\mathbf{C}\mathbf{M}^{-1}\mathbf{f}+\mathbf{c}''\\ \mathbf{S}\mathbf{M}^{-1}\mathbf{f}+\mathbf{s}''\end{bmatrix}, \tag{5}$$

if the inverse of the matrix exists. Then, the differentiation index of the servo control problem is three and there remain $n_{\mathrm{f}}-(n_{\mathrm{c}}+n_s)$ free coordinates. In nonlinear control theory, this correlates with a relative degree of two. In the next step, the driven dynamics is eliminated and the internal dynamics is stated as a set of ordinary differential equations for the independent states. For the flexible manipulator, a possible basis for the independent states consists of the elastic coordinates and their derivatives. The dependent coordinates, which still appear in the dynamics, have to be computed with a root search. Based on the provided end-effector trajectories and the solution of the internal dynamics, it is possible to provide the trajectories of the dependent coordinates $\mathbf{z}(t)$ and their time derivatives as well as for the system inputs $\mathbf{u}(t)$ and the Lagrange multipliers $\boldsymbol{\lambda}(t)$ that fulfill both types of constraints on a fixed time interval. These form the feed-forward control. Therefore, this approach can be regarded as dynamical trajectory planning. For the application on the real-time target system, the internal dynamics are solved off-line and the computed trajectories are interpolated with a constant sample time of 0.6 ms, which corresponds to the sampling time of the control loops, and are exported as HDF5-files.

## 4. Control concept and signal conditioning

For the stabilization of the pre-computed solution, two distinct cascade controllers for the drive positions are implemented. These well-established controllers offer the possibility to feed-forward the positions, velocities as well as the currents, which accords with the requirements of the overall control concept. Fig. 5 shows the topology of the control loop. The outer loop is a proportional gain, which amplifies the position error and feeds the middle loop. The velocity loop is carried out as a PI-controller and the desired velocity is feed-forwarded. The inner loop, which consists of the electrical subsystem $\sum_{\mathrm{e}}$ and a proportional gain, is realized in the servo controllers.

The states needed for the feedback controller are the drive positions $s_i$ and velocities $\dot{s}_i$. The positions of the mechanical subsystem $\sum_{\mathrm{m}}$ are acquired from the servo drives and transmitted as CANopen PDOs. In order to reconstruct the velocities, a Kalman filter (KF) [12] is applied. The underlying model of the filter is a zero-mean first-order Markov model [13], which is borrowed from target tracking applications [14]. The main advantage of this filter is its simplicity and the consequential efficiency in real-time applications. In addition, the assumptions of the filter regarding the smoothness of the trajectories consort with a feed-forward control. The discrete-time state-space representation of the model is

$$\mathbf{x}_{k+1}=\begin{bmatrix}1 & T & \frac{1}{\alpha^2}(-1+\alpha T+e^{-\alpha T})\\ 0 & 1 & \frac{1}{\alpha}(1-e^{-\alpha T})\\ 0 & 0 & e^{-\alpha T}\end{bmatrix}\mathbf{x}_k+\mathbf{u}_k, \tag{6a}$$

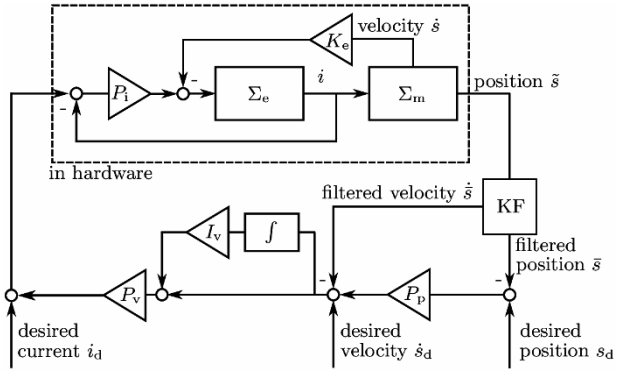



Fig. 5. Control loop.

$$y_k = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \mathbf{x}_k + v_k, \tag{6b}$$

in which the discrete state vector $\mathbf{x}_k$ includes the position, velocity, and acceleration of one drive each. The system dynamics, which consists of two integrators and a stable pole associated with the acceleration, is driven by the discrete time white noise sequence $\mathbf{u}_k$, which is assumed to be normally distributed with zero mean and covariance $\mathbf{Q}_k$. Since the position signals are acquired from an optical encoder and processed digitally, the measurement noise covariance $R_k$ of the output $y_k$ is almost negligible. Therefore, the main purpose of the filter is the robust reconstruction of the velocity. These are the only signals that are needed for the feedback controllers.

## 5. Experimental studies

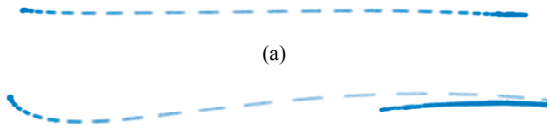
Based on the presented models and control approaches, two different implementations are benchmarked. The first implementation is based on the full dynamical model of the elastic system and the second one is based on the corresponding rigid model. While the first concept requires the solution of a boundary value problem, see Sec. 3.2, the second concept can be solved algebraically, see e.g. Ref. [15]. For both scenarios, the end-effector trajectory is supposed to cover a distance of 0.282 m within 0.5 s following a straight line. The used feedback controllers are parameterized identically and in both cases, the positions, velocities, and currents are feed-forwarded. The friction compensation adds, in dependency of the motion of the drives, a constant current to the pre-computed inputs.

### *5.1 Image tracking*

In order to compare the two implementations, the position of the end-effector point is tracked with a camera. To obtain the best possible contrast, all reflecting parts of the end-effector are covered with felt. The designated point and an additional point are highlighted with reflector foil. The distance between these emphasized points is used for the conversion from pixels to meters. Compared to a designated measure, these two points remain in the image plane, which avoids additional conversions. The used camera is capable to record 60 frames per seconds with a resolution of 1280 x 720 pixels. The left image of Fig. 6 one shows the brightness distribution if the end-effector is at rest. In this case, the image matches the shape of the reflector foil. However, if the end-effector is moving, the image is blurred. This is caused by the exposure time of the camera and the rather fast movement of the end-effector. The image on the left of Fig. 6 shows such movement. In addition, the exposure time also affects the brightness of the pixels. A darker color in the Light-Bertlein color map indicates that the area was exposed for a longer period of time than a lighter color, i.e. a darker color indicates a slower motion. Fig. 7 shows the brightness distribution of all frames merged into one image for both scenarios.

(a) (b)

Fig. 6. Brightness distributions of the end-effector point at rest (a); and during motion (b).



(a)

(b)

Fig. 7. Brightness distribution of all frames merged into one image for the elastic (a); and the rigid (b) feed-forward control.

### *5.2 Post-processing*

Based on these images, the trajectory of the end-effector is identified. In a first step, the positions of the end-effector point and the additional marker are obtained frame-by-frame. Due to the fact that multiple pixels are exposed by the light reflected by the foil, it is possible to interpret the brightness of the frame as a two-dimensional non-normalized probability distribution. In order to improve the accuracy, it is important to filter the image to remove noise and to isolate the two peaks from each other. Otherwise, the size of the image would have a great influence on the expected value. The filtering is done with a sliding-neighborhood filter. The expected values of these distributions, which equal the components $x$ and $y$ of the position vector $\mathbf{r}_{pix}$ of the designated markers in the horizontal plane, can be obtained by

$$p(i,j) = \frac{b(i,j)}{\sum_{i\in M}\sum_{j\in N} b(i,j)}, \tag{7a}$$

$$x = \sum_{i\in M}\sum_{j\in N} ip(i,j), \tag{7b}$$

$$y = \sum_{i\in M}\sum_{j\in N} jp(i,j). \tag{7c}$$

Here, the normalized probability $p(i,j)$ of the $(i,j)$-th pixel is

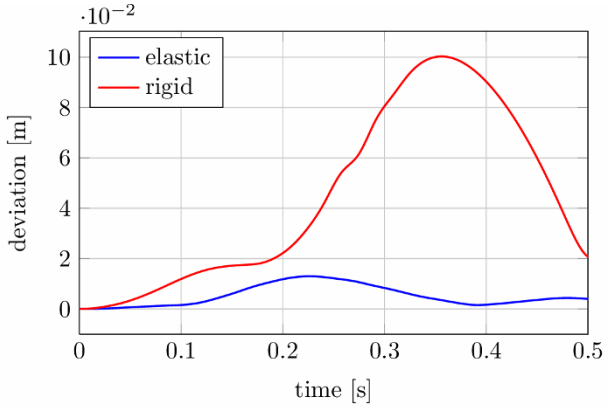



Fig. 8. Reconstructed end-effector deviations of the two implementations.

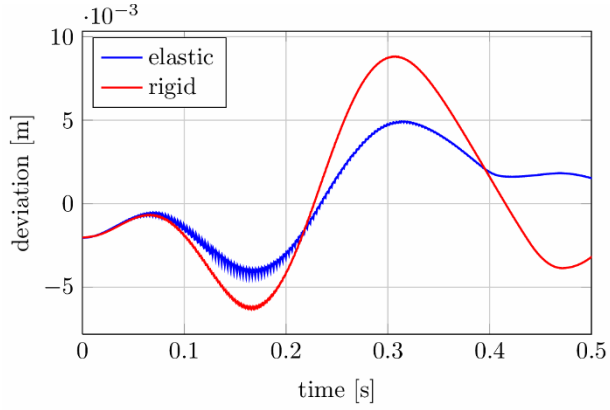



Fig. 9. Tracking errors of the direct drives for the two presented implementations.

obtained from the brightness $b(i, j)$, which ranges from *0* to *255*. Due to the filtering, the summation can be performed on the subsets $M$ and $N$, which are the neighborhoods of the maximum value of the brightness distribution. Outside of these sets, the brightness is zero.

In a next step, the conversion from pixels to meters is carried out with the conversion factor

$$f_c = \frac{|\Delta|}{|\Delta_{\text{pix}}|}, \tag{8}$$

where $\Delta$ labels the difference of the position vectors of the two designated markers. The values with the index "Pix" are measured in pixels and the index-free values are measured in meters. After the conversion of the reconstructed position vectors, the data is resampled using spline interpolation, aligned with the desired values and cropped. In a last step, the measured and processed trajectories are rotated and translated in such a way, that the initial conditions of the desired and measured trajectories match.

### *5.3 Interpretation of the experimental results*

The processed measurement data sets permit to compare the different feed-forward control implementations. To compare the two approaches, the deviations of the end-effector from the desired trajectory are regarded. Fig. 8 shows the tracking errors of both approaches over time. Due to the fact that the post-processing was carried out in such way that the initial conditions of the end-effector match the desired values, there are no deviations at t = 0. Consistent with Fig. 7, the trajectory of the feed-forward approach, which is based on the elastic model, shows a much smaller error than the second approach which is based on the corresponding rigid model. The maximum errors of the flexible and the rigid feed-forward implementation are 1.3 cm and 10 cm, respectively.

The errors shown in Fig. 8 do not only result from inaccurate models. In fact, the control deviations cannot be regarded as negligible. Therefore, it is reasonable to investigate the tracking errors of the drive positions as well.

Fig. 9 shows the position-tracking errors of the first drive for both approaches. Both graphs exhibit errors in the initial conditions as well as in the movement. These errors are mainly caused by the friction forces, which act on the drives and by the elastic deformations of the long link that affect the drive positions during the movement. Due to the feedback controllers and the friction compensation, this tracking error is stabilized. Again, it can be stated the flexible approach yields smaller tracking errors, because this approach results in smaller elastic deformations and therefore less disturbances. These errors in the drive positions are, in case of the flexible feed-forward control implementation, the main source of errors in the end-effector tracking. It should also be noted, that at this point the elastic deformation is not included in the feedback control. Thus, errors in the elastic deformation cannot be compensated yet.

## 6 Conclusions

In this contribution, experimental studies regarding two different feed-forward control designs for a parallel manipulator with flexible links are carried out. After a short presentation of the used hardware and software, the forward and inverse dynamics of the flexible multibody system are stated. A quick introduction to the used control concept leads to the discussion of the experimental results. Here, the main idea of image tracking is revised and the two implementations are compared. The feed-forward control approach, which is based on a complex flexible multibody system and obtained by a boundary-value problem, shows obvious improvements in the end-effector deviations compared to the approach, which is based on an corresponding rigid model. Even though there remain deviations in the drive positions as well as in the end-effector tracking, this approach is a promising alternative to classical control concepts.

## Acknowledgment

The authors would like to thank the German Research Foundation (DFG) for their financial support of the project within the Cluster of Excellence in Simulation Technology (EXC 310/2) at the University of Stuttgart and the project SE 1685/3-1.

## References

[1] R. Seifried, *Dynamics of underactuated multibody systems*, Springer, Berlin (2014).

[2] J. Blanchette and M. Summerfield, *C++ GUI Programming with Qt4*, Second Ed., Prentice Hall, Upper Saddle River (2008).

[3] Modbus: Modbus application protocol application V1.1b3, *Available at: http://www.modbus.org/* (2012).

[4] M. W. Spong, S. Hutchinson and M. Vidyasagar, *Robot modeling and control*, John Wiley & Sons, Hoboken (2006).

[5] T. Kurz, P. Eberhard, C. Henninger and W. Schiehlen, From Neweul to Neweul-M$^2$: symbolical equations of motion for multibody system analysis and synthesis, *Multibody System Dynamics*, 24 (1) (2010) 25-41.

[6] M. Burkhardt, P. Holzwarth and R. Seifried, Inversion based trajectory tracking control for a parallel kinematic manipulator with flexible links, *Proc. of the 11$^{th}$ International Conference on Vibration Problems*, Lisbon, Portugal (2013).

[7] W. Blajer and K. Kolodziejczyk, A geometric approach to solving problems of control constraints: theory and a DAE framework, *Multibody System Dynamics*, 11 (4) (2004) 343-364.

[8] C. J. Damaren, Passivity and noncollocation in the control of flexible multibody systems, *Journal of Dynamic Systems, Measurement, and Control*, 122 (1) (2000) 11-17.

[9] R. Seifried, M. Burkhardt and A. Held, Trajectory control of serial and parallel flexible manipulators using model inversion, J. C. Samin and P. Fisette (Eds.), *Multibody Dynamics: Computational Methods and Applications, Computational Methods in Applied Sciences 28*, Springer, Berlin (2013) 53-75.

[10] B. Bona and M. Indri, Friction compensation in robotics: an overview, *Proc. of the 44$^{th}$ IEEE Conference on Decision and Control, and the European Control Conference*, Seville, Spain (2005) 4360-4367.

[11] M. S. Grewal and A. P. Andrews, *Kalman filtering: theory and practice using MATLAB*, John Wiley & Sons, Hoboken, USA (2011).

[12] R. A. Singer, *Estimating optimal tracking filter performance for manned maneuvering*, Target Hughes Aircraft Company (1970).

[13] X. R. Li and V. P. Jilkov, A survey of maneuvering target tracking: dynamic models, *Proc. of SPIE Conference on Signal and Data Processing of Small Targets*, Orlando, USA (2000) 212-235.

[14] A. Ast and P. Eberhard, Flatness-based control of parallel kinematics using multibody systems – simulation and experimental results, *Archive of Applied Mechanics*, 76 (3-4) (2006) 181-197.

**Markus Burkhardt** is a research assistant at the Institute of Engineering and Computational Mechanics at the University of Stuttgart, Germany. His research topics include modeling and control of flexible multibody systems, differential-algebraic equations, and feedforward control.

**Robert Seifried** is head of the Institute of Mechanics and Ocean Engineering at the Hamburg University of Technology. His research interests are in flexible multibody dynamics, mechatronics, nonlinear control of under-actuated multibody systems, optimization, structural vibration, wave propagation and discrete element methods.

**Peter Eberhard** is director of the Institute of Engineering and Computational Mechanics at the University of Stuttgart. His research interests are in multibody dynamics, contact mechanics, optimization, biomechanics, model-order reducetion, and mechatronics as well as various projects with industrial partners.