IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Vancouver, BC, Canada, pp. 4568-4575, 2017.

# Forward Dynamics Compliance Control (FDCC): A New Approach to Cartesian Compliance for Robotic Manipulators

Stefan Scherzinger[1], Arne Roennau[1] and Rüdiger Dillmann[2]

*Abstract*— **Compliant end effectors in robotics are an important prerequisite for the field of object manipulation and environment interactions. However, current manipulators are usually stiff position-controlled systems, generating the need to add compliance. In this work, we present Forward Dynamics Compliance Control (FDCC), a new threefold control concept that realizes Cartesian compliance through combining Admittance, Impedance and Force Control into one control strategy. We close the control loop only through a force-torque sensor, allowing a system independent and decoupled configuration of the end effector compliance. As a key component in FDCC, we leverage forward dynamics simulations of a virtual model to directly map Cartesian inputs to joint control commands, leading to excellent stability in singularities. Experiments on three different robotic manipulators verify the key advantages of this approach.**

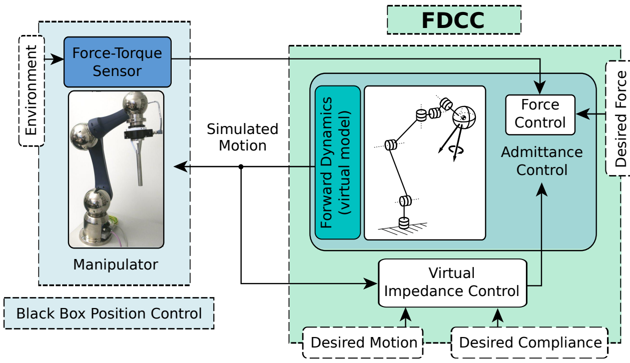



Fig. 1: Forward Dynamics Compliance Control (FDCC) with its key components and interfaces.

## I. INTRODUCTION

Robotic manipulators must be prepared to *yield* contact forces, as they emerge in a multitude of contact dominated tasks, such as object manipulation of all kinds. Despite this crucial necessity, however, the vast majority of deployed manipulators are stiff systems with high gain position control. Intensive research has been done, to impose a compliant behavior onto those systems in form of add-ons.

In this work, we refer with the term *compliance* to the broader and general ability of a robotic manipulator to *yield* external forces. There are inherently compliant mechanical devices to realize this behavior, such as early implementation of the Remote Center Compliance (RCC) [1], also referred to as Passive Compliance, with a variety of more recent implementations [2]. There exist also various realizations in software control, which constitute an imposed control to mimic a certain mechanical behavior, originally referred to as Active Compliance [3]. In these control principles, the contact forces are mostly regulated either *directly* through Force Control [3], Hybrid Position/Force Control [4],[5], Admittance Control [6],[7], or *implicitly* through Impedance Control [6], and Stiffness Control [8].

While Force Control and Hybrid Position/Force Control handle effort and motion in different subspaces, Impedance and Admittance Control embrace the idea, that motion and effort are so inherently woven together, that they are best controlled at the same time [6]. Usually position, velocity and acceleration dependent error compensation is used to reflect the behavior of springs, dampers and masses and to comprise linear second order equations to describe the wanted behavior in Cartesian space. This aspect has been generalized by Virtual Model Control (VMC) [9], which also attaches virtual elements to the system to solve the equations to obtain torque control commands. Especially more recent research on Impedance Control has showed significant results [10],[11], and has reached industrial maturity [12],[13].

Yet, position-controlled systems still represent the majority of robotic manipulators currently available, for which those methods are not directly applicable. In this work, we propose a new add-on control strategy that will extend the capabilities of these classical robots, as is depicted in Fig. 1. We consider the position control as a black box, which is the lowest possible level to interact with the system to impose a behavior in an outer control loop. For these requirements, Admittance Control is the most promising, since other control principles require the robots to have joint torque interfaces, which is why methods, basing on the idea of the Computed Torque Method [14],[15] are not applicable.

To illustrate the problem, let's take as a starting point for Cartesian compliance the equation of motion of a one dimensional linear spring damper mass element to express a relation between end effector target motion $x_0$ and external effort $F^{ext}$:

$$F^{ext} = m(\ddot{x} - \ddot{x}_0) + d(\dot{x} - \dot{x}_0) + c(x - x_0) \qquad (1)$$

The goal here is to compute a time dependent motion $x(t)$ in order to control the applied force on the system (Admittance Control). For the general case, a set of joint motion $\boldsymbol{q}(t)$

[1]Stefan Scherzinger and Arne Roennau are with FZI Research Center for Information Technology, Haid-und-Neu-Str. 10-14, 76131 Karlsruhe, Germany stefan.scherzinger@fzi.de arne.roennau@fzi.de

[2]Rüdiger Dillmann is with IAR Institute for Anthropomatics and Robotics, HIS Humanoids and Intelligence Systems Lab, KIT Karlsruhe Institute of Technology, Adenauerring 2, 76131 Karlsruhe, Germany ruediger.dillmann@kit.edu

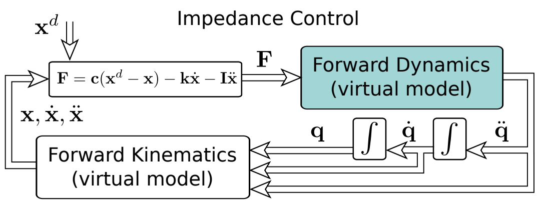



Fig. 2: Schematic illustration of the Impedance Control component

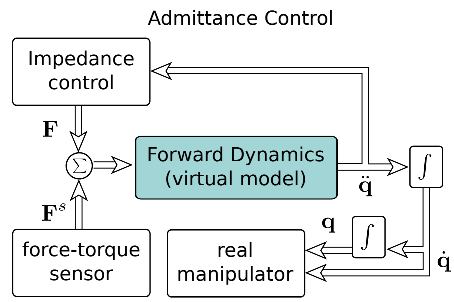



Fig. 3: Schematic illustration of the Admittance Control component, which uses the system input $\boldsymbol{F}$ from the impedance control from Fig. 2.

must be found, which would effect this 6D motion $\boldsymbol{x}(t)$, which is essentially trying to solve this problem from an inverse perspective.

However, all solutions that rely on Inverse Kinematics will have to take special measures, see e.g. [16],[17] to assure stability in the vicinity of singularities. Without this, system instability has been observed [10]. General damping approaches exist [18],[19] to regulate extensive joint speeds, but these approaches cannot make up for an inherent problem.

This requires a change of perspective, which is why we leave inverse kinematics aside and build upon fast forward dynamics simulations instead, creating the option to fuse multiple control concepts.

## II. CONTROL CONCEPT

In our approach, we combine three control principles, i.e. Impedance Control, Admittance Control and Force Control into one new control strategy. In contrast to the approach from [20], we do not switch between either Admittance or Impedance Control depending on different criteria, but merge both principles. At the core of this new control strategy, we build upon the idea to use dynamics simulations to directly control robotic manipulators via virtual and measured forces, acting on their end effectors. As a key component to this concept, we propose and evaluate the usage of Forward Dynamics as a highly suitable *solver* to map effort from task space to motion commands in joint space, and to enable an easy integration of all three control concepts. By using simulation as a crucial part of our control, we embrace the idea to mimic 'how the system would behave', which is as close as an imposed compliance by software can get to mechanical, passive devices.

### A. Impedance Control

The Impedance Control presented here is in contrast to classical implementations pure virtual. It uses the resulting motion from the forward dynamics simulation without including feedback from the physical robot, see Fig. 2. The obtained joint accelerations $\ddot{\boldsymbol{q}}$ are integrated twice and mapped with forward kinematics into Cartesian accelerations $\ddot{\boldsymbol{x}}$, velocities $\dot{\boldsymbol{x}}$ and positions $\boldsymbol{x}$ respectively, where they are compared to the desired motion $\ddot{\boldsymbol{x}}^d, \dot{\boldsymbol{x}}^d, \boldsymbol{x}^d$. The resulting effort of each individual component, i.e. spring, damper, and mass, is

$$\boldsymbol{F}^{sp} = \boldsymbol{c}\Delta\boldsymbol{x} = \boldsymbol{c}(\boldsymbol{x}^d - \boldsymbol{x}) \tag{2}$$

$$\boldsymbol{F}^{da} = -\boldsymbol{k}\dot{\boldsymbol{x}} \tag{3}$$

$$\boldsymbol{F}^{in} = -\boldsymbol{I}\ddot{\boldsymbol{x}} \tag{4}$$

Where $\boldsymbol{F}^{sp}$ denote the forces and torques of the virtual spring, and $\boldsymbol{F}^{da}$ and $\boldsymbol{F}^{in}$ denote the forces and torques generated in the virtual damper and through accelerating the apparent mass of the end effector. With $\boldsymbol{c}, \boldsymbol{k}$ and $\boldsymbol{I}$ beeing the 6D stiffness, damping and inertia matrix according to:

$$\boldsymbol{c} = \begin{bmatrix} \boldsymbol{c}_r & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{c}_t \end{bmatrix}, \boldsymbol{c}_r = [c_{rx}, c_{ry}, c_{rz}]^T, \boldsymbol{c}_t = [c_{tx}, c_{ty}, c_{tz}]^T \tag{5}$$

$$\boldsymbol{k} = \begin{bmatrix} \boldsymbol{k}_r & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{k}_t \end{bmatrix}, \boldsymbol{k}_r = [k_{rx}, k_{ry}, k_{rz}]^T, \boldsymbol{k}_t = [k_{tx}, k_{ty}, k_{tz}]^T \tag{6}$$

with the respective translational and rotary entries. The 6D inertia matrix is

$$\boldsymbol{I} = \begin{bmatrix} I_{xx} & -I_{xy} & -I_{xz} & \\ -I_{yx} & I_{yy} & -I_{yz} & \boldsymbol{0} \\ -I_{zx} & -I_{zy} & I_{zz} & \\ & \boldsymbol{0} & & \boldsymbol{m} \end{bmatrix}, \boldsymbol{m} = \begin{bmatrix} m & 0 & 0 \\ 0 & m & 0 \\ 0 & 0 & m \end{bmatrix} \tag{7}$$

The negative signs in $\boldsymbol{F}^{sp}$ and $\boldsymbol{F}^{in}$ show that these forces always oppose motion. For those forces we chose to remove the dependency on the desired motion $\dot{\boldsymbol{x}}^d, \ddot{\boldsymbol{x}}^d$, to prevent overshoots and to achieve a smoother motion in the open loop control phases. Motion characteristics can, however, be commanded indirectly through the characteristics of $\boldsymbol{x}(t)$ over time.

### B. Admittance Control

Beside the sensor measurements $\boldsymbol{F}^s$, the resulting forces as computed by the virtual Impedance Control are used as inputs, see Fig. 3. Integrating measurements from the force-torque sensor effectively closes the overall control loop with the real hardware, whenever the sensor detects contacts. Instead of computing the reaction motion due to these forces in task space and later express them in joint space, forward dynamics simulation is applied to compute both steps at the same time. This reaction motion is directly commanded to the manipulator, whose result is again perceived indirectly by the sensor.

### C. Force Control

The third component, the Force Control, is very similar to the admittance control part. The only difference is that desired forces $\boldsymbol{F}^d$ are included into the sum from Fig. 3. Consequently, the Force Control proposed here is only theoretically exact, as long as the target positions reflect the exact moment when contact with the environment was established. A downside to this is that any offset will cause a partial compensation by the virtual spring forces and will distort the accuracy of the Force Control. However, this effect is small as long as the localization of the system is within reasonable accuracy. As an example, a localization error of 5 mm, causes with a stiffness of 1000 N/m an error in the target force of 5 N. We furthermore see an advantage in the simplicity of this superposition and argue that for many tasks the exerted forces by a manipulator do not have to be mathematically exact but rather should be controlled inside safe margins. In addition, when stochastically working controllers are used on top of Forward Dynamics Compliance Control, such as applications from the field of machine learning, the ability of exerting exact forces down to sub-Newton precision is less relevant. If however, exact Force Control is desired, we suppose to subtract the part $\boldsymbol{F}_{\parallel}^{sp}$ of the virtual spring that is collinear with the desired forces $\boldsymbol{F}^d$:

$$\boldsymbol{F}^{net} = \boldsymbol{F}^s + \boldsymbol{F}^d + \boldsymbol{F}^{in} + \boldsymbol{F}^{da} + \boldsymbol{F}^{sp}_{reduced} \tag{8}$$

$$\boldsymbol{F}^{sp}_{reduced} = \boldsymbol{F}^{sp} - \boldsymbol{F}^{sp}_{\parallel} \tag{9}$$

$$\boldsymbol{F}^{sp}_{\parallel} = [\boldsymbol{f}^{sp}_{\parallel}, \boldsymbol{m}^{sp}_{\parallel}]^{\intercal} \tag{10}$$

$$\boldsymbol{f}^{sp}_{\parallel} = \frac{\boldsymbol{f}^{sp} \cdot \boldsymbol{f}^d}{\|\boldsymbol{f}^d\|} \frac{\boldsymbol{f}^d}{\|\boldsymbol{f}^d\|} \tag{11}$$

$$\boldsymbol{m}^{sp}_{\parallel} = \frac{\boldsymbol{m}^{sp} \cdot \boldsymbol{m}^d}{\|\boldsymbol{m}^d\|} \frac{\boldsymbol{m}^d}{\|\boldsymbol{m}^d\|} \tag{12}$$

In which $\boldsymbol{f}^{sp}, \boldsymbol{f}^d, \boldsymbol{m}^{sp}$ and $\boldsymbol{m}^d$ denote the 3D-vectors of forces and torques of the virtual spring and the desired force.

### D. Combined Control Architecture

We achieve the combination of all three control concepts by merging them over their common interface to the forward dynamics solver, which is graphically highlighted both in Fig. 2 and Fig. 3. The whole concept is depicted in Fig. 4. Both open-loop and closed-loop control are used in this approach, which inherently apply, depending on whether the robotic manipulator is in free motion or in contact with its environment. The closed loop component is covered by the Admittance Control, which leverages the forward dynamics solver to circumvent the Inverse Kinematics problem, and to compute reaction motions from measurements of the force-torque sensor on the end effector. Those reaction motions are themselves directly processed in the Impedance Control, which deliberately ignores the joint state feedback of the real robotic system, and instead compares this simulated motion to the desired motion input. Through a virtual spring, damper and mass, this motion is mapped to a regulating 6D force, which is superimposed to the measurements of the sensor.

While in unconstrained motion this leads to an unaffected system behavior and covers the open loop control, the measurements form the sensor close the loop whenever the robotic manipulator is in contact with its environment.

In the third control component, Force Control, a 6D target force is applied in parallel to the already mentioned forces. All forces and torques appearing on the end effector are joined into a net force, and cause the reaction motion of the system by applying forward dynamics simulation, as graphically highlighted in Fig. 4 as a common solver pipeline. This iterative process runs individually and completely detached from the position control cycle on the real Robot.

The simulated motion is then directly given to the robotic position control, adhering the systems specific control cycle, which typically lies a magnitude under the FDCC sampling rate.

By the very nature of forward dynamics simulations, we obtain the resulting reaction motion on acceleration level first, and, after two integrations, we obtain smooth control commands for the joints of the manipulator, which make the usage of noise filters redundant.

All input to the forward dynamics solver is the external force $\boldsymbol{F}^{ext} = \boldsymbol{F}^c$, which is controlled to zero by a PD controller for each dimension. In the ideal case, the net force $\boldsymbol{F}^{net}$, encompassing the target forces, the forces measured by the sensor, and all forces that result from virtual motion in the impedance control, is controlled to zero and the system is in energetic equilibrium at all times, including in the middle of motion and during possible contacts with the environment. For the PD controllers,

$$F_i^c = -k_{p,i} F_i^{net} - k_{d,i} \dot{F}_i^{net} \;\;, \;\; i = 1 \dots 6 \tag{13}$$

is used for all six Cartesian dimensions $i$, where $k_{p,i}$ and $k_{d,i}$ are the proportional gains and derivative gains respectively. Note that no integral gain is necessary, since any offset between desired pose $\boldsymbol{x}^d$ and virtual pose $\boldsymbol{x}$ will generated a restoring force in the virtual spring, which will lead to corrective motions in the virtual model. As long as the kinematic model of the manipulator is known with sufficient accuracy, this control will eliminate steady state errors on the real system.

### E. Interfaces

In order to describe robotic tasks in task space, the system must provide user interfaces, which receive commands, while maintaining its compliant behavior for the end effector. FDCC provides interfaces to simultaneously command both desired motion $\boldsymbol{x}^d$ and desired force $\boldsymbol{F}^d$ with respect to an arbitrary end effector coordinate system $\{x, y, z\}$, as illustrated in Fig. 5. This is possible through the conjunction of virtual Impedance Control and Admittance Control. The characteristics of the compliant behavior can be adjusted through parameters for the virtual inertia, dampers and springs.

## III. FORWARD DYNAMICS MODELS

An ideal Cartesian Robot control tries to be perfectly homogeneous in all its six dimensions. Yet, the end effector

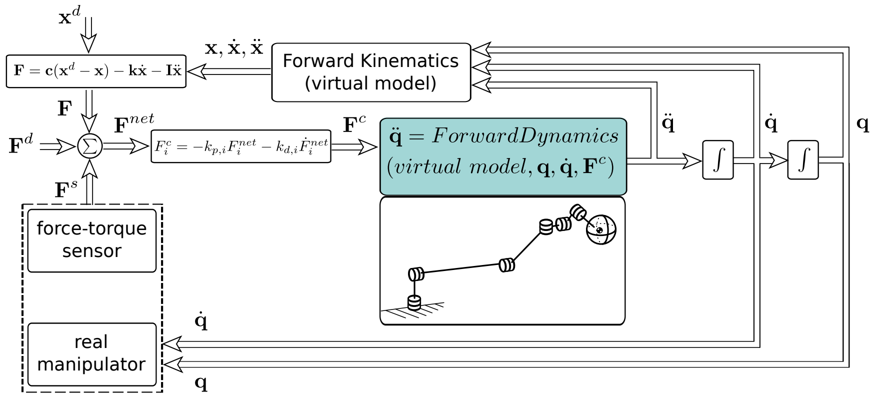



Fig. 4: Control architecture of FDCC. The three control concepts Impedance Control, Admittance Control and Force Control are combined. As the central element, Forward Dynamics of a specially designed virtual model to achieve manipulator-agnostic behavior is leveraged.

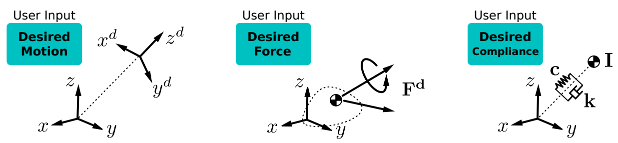



Fig. 5: User control interfaces of Forward Dynamics Compliance Control (FDCC)

always belongs to a kinematic chain with constraints. Especially singularities show the strong non-linearity of the mapping from task space to joint space. Our goal is to create a solution that is as joint configuration independent as possible, but does not enforce solutions in joint space for given Cartesian motion, if the kinematic structure is not able to account for it.

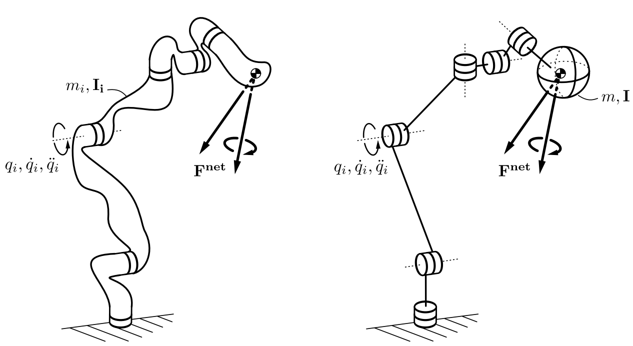



Fig. 6: Modelling the manipulator (left) as a massless kinematic chain with a freely chosen end effector inertia (right)

We achieve this with taking the exact kinematics of the robotic manipulator we want to control as a starting point. This is necessary to assure that the real system can theoretically execute the motion as simulated. The second step encompasses a central idea behind the here proposed control scheme: we build a dynamic model, while reducing the inertias of all links to a minimum. Only the last link is given a mass and inertia, as is illustrated in Fig. 6. The parameters $\boldsymbol{c}, \boldsymbol{k}$ and $\boldsymbol{I}$ from (2) to (4) can be set freely to reflect any wanted behavior. The idea behind this is to reduce all possible dynamic effects of the rest of the chain to a minimum, making the end effector the dominant element that governs the simulated motion.

Position-controlled manipulators inherently account for gravity, and hence, outer control loops do not need to compensate those effects. In order not to drift with the virtual model, gravity is removed from the system, which leads in combination with the damper related forces from (3) to a rigid body, connected to a set of massless links and joints, floating through a viscous fluid. The idea is then simply to apply the overall acting forces on this model, compute its reaction motion, and give those as control commands to the robotic manipulator. The output of the forward dynamics solver are accelerations in joint space, which represent the reaction motion due to the controlled net force. Obtaining the accelerations first, gives smooth velocities and positions after integration. In addition, the concentrated inertia acts as a low-pass filter, which is robust to sensor noise. Thus, using forward dynamics enables both high sensitivity and at the same time intolerance to sensor noise.

In addition, in singularities, external forces lose their immediate effect of generating motion in the joints of the virtual model. The (infinite) stiff mechanical structures of the virtual manipulator links then account for the external loads. No acceleration can occur on the orthogonal joint axis. In this aspect, the solution presented here is contrary to inverse kinematics approaches: while in singularities kinematic approaches lead to high joint motion, forward dynamics leads to no joint motion at all, and thus provides superior stability.

Note, that it is not necessary to model the dynamics close to the real system. In fact, it is more advantageous to choose the concentrated mass on the end effector completely independent of the actual robot masses to achieve a nearly identical behavior in most workspace configurations, using different robots.

### A. Manipulator-Agnostic Behavior

As described before, the control loop involving the real manipulator is only closed through the end effector force-torque sensor. Since no direct feedback from the joints of the hardware has an immediate influence, the controller can be designed decoupled from the real system with almost arbitrary characteristics. Advantages are, e.g. that aperiodic damping can be pre-adjusted in conjunction with the environment, regardless of the underlying manipulator, making the compliance we propose here a true add-on compliance.

### B. Simulating Reaction Motion

The before described system can be modelled with the general equations of motion for a rigid body system in joint coordinates:

$$\boldsymbol{H}(\boldsymbol{q})\,\ddot{\boldsymbol{q}} + \boldsymbol{C}\left(\boldsymbol{q}, \dot{\boldsymbol{q}}, \boldsymbol{F}^{ext}\right) = \boldsymbol{\tau} \tag{14}$$

With $\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}}$ denoting the generalized coordinates, $\boldsymbol{H}$ denoting the generalized inertia matrix, $\boldsymbol{C}$ holds gravitational forces, centrifugal and coriolis terms, and also includes external forces $\boldsymbol{F}^{ext}$, other than $\boldsymbol{\tau}$. $\boldsymbol{\tau}$ denote the generalized forces, acting in the joints of the manipulator. Leaving gravity out and asuming a fully underactuated (floating) system, (14) becomes

$$\boldsymbol{H}(\boldsymbol{q})\,\ddot{\boldsymbol{q}} + \boldsymbol{C}\left(\dot{\boldsymbol{q}}, \boldsymbol{F}^{ext}\right) = \boldsymbol{0} \tag{15}$$

There exist various ways of solving theses equations for the joint accelerations $\ddot{\boldsymbol{q}}$. In this work, we apply the *Articulated Body Algorithm* from [21], which solves the forward dynamics problem recursively in $\mathcal{O}(N)$ complexity. $N$ being the number of bodies of the system. We chose this algorithm, because it uses a compact notation for both forces and torques using 6-D vector representation. Nevertheless, any algorithm is suitable that solves the equations of motion of a rigid body system according to

$$\ddot{\boldsymbol{q}} = ForwardDynamics\left(model, \boldsymbol{q}, \dot{\boldsymbol{q}}, \boldsymbol{F}^{ext}\right) \tag{16}$$

Especially for redundant manipulators, it is necessary to add additional damping terms in $\boldsymbol{F}^{ext}$ that apply to each (nearly) massless link of the chain to prevent motion in the null space when the end effector is already at rest. Note that for algorithmic reasons, the masses cannot be set to zero exactly, because they would provoke infinite joint accelerations. However, values in the neighborhood of few grams per link are suitable.

## IV. IMPLEMENTATION

The controller architecture from Fig. 4 was implemented in C++ using ROS [22] as middleware. For resolving the forward dynamics of the model, we built upon the *Rigid Body Dynamics library* [23] from Felis, which provides an efficient implementation of the Articulated Body Algorithm [21] from Featherstone.

## V. EXPERIMENTAL RESULTS

We evaluated the FDCC on three robotic manipulators of highly different kinematics and dynamics: The Schunk LWA4P, the Universal Robot UR10 and the KUKA KR16, see Table I. As force-torque sensors, the Robotiq FT 150 and the ATI Delta were used. The Forward Dynamics Compliance Control ran on a Core-i7 with 8 GB ram on a 64-Bit Linux-based operating system, and at a sampling rate of 1 KHz.

TABLE I: Manipulators and their characteristics

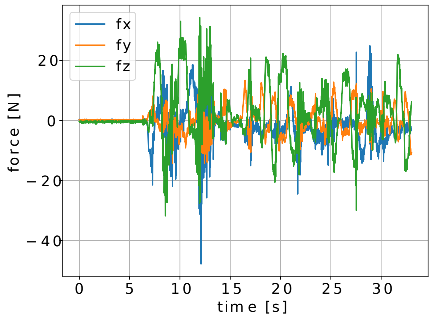
| | Lwa4p | UR10 | KR16 |
|---|---|---|---|
| Axes | 6 | 6 | 6 |
| Max. Payload [kg] | 6 | 10 | 16 |
| Weight [kg] | 15 | 17 | 235 |
| Reach [mm] | 730 | 1300 | 1611 |
| Repeatability [mm] | $\pm 0.15$ | $\pm 0.1$ | $\pm 0.05$ |
| Control frequency [Hz] | 100 | 125 | 83.3 |

Throughout all experiments and alike for all manipulators, the controller gains from (13) were constant, to underline the ability to achieve a robot-agnostic behavior with FDCC. Note, that although the virtual model for each robot differed in the underlying kinematics, they could be controlled with the same dynamics parameters, e.g. an end effector mass of $1\,\mathrm{kg}$ in the experiment from Fig. 9, although one of the physical robots weighed an oder of magnitude more than the others.

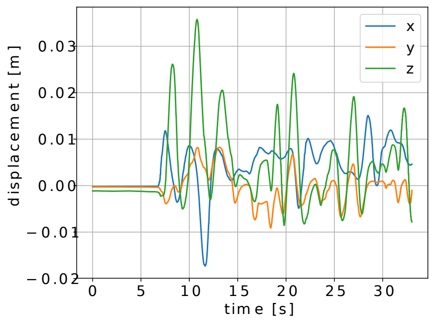
The performance of the Admittance Control part was tested with human interaction. Fig. 7 shows the end effector forces as measured by the ATI Delta on the KR16 during shaking. Although the sensor forces vary heavily, the resulting motion is smooth and stable, which is a direct consequence of obtaining the resulting motion on acceleration level first. Consequently, delay-introducing filters can be omitted.

The Force Control capability was tested with a sliding experiment on a stiff surface of approx. $20\,\mathrm{cm}$, using the UR10 with a special rod, mounted to the end effector. Fig. 8 shows the results. To also cover the effects of contact, the manipulator was started slightely elevated of approx. $5\,\mathrm{mm}$, before the target force was applied. According to the $x$-component of the measured sensor forces, it can be identified when the manipulator is in sliding motion. The offset to zero indicates the presence of friction and vanishes, once the manipulator hovers to its starting pose. Apart from a slight offset and noise, the force is controlled to the target value of $20\,\mathrm{N}$. The offset is a consequence of the superposition with the virtual spring, as discussed in II-C.

Fig. 9 shows the position step responses of the three manipulators. To observe cross correlations in three Cartesian dimensions, the steps were applied successively to the unit axes every four seconds each. It can be seen that the LWA4P shows a less decoupled response than the UR10 or the KR16. However, all manipulators respond very similarly and reach the target displacement without overshooting, as was intended for the experiment.

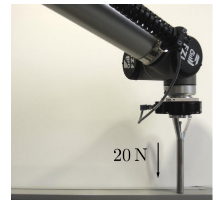



(a) Sensor values during shaking the manipulator

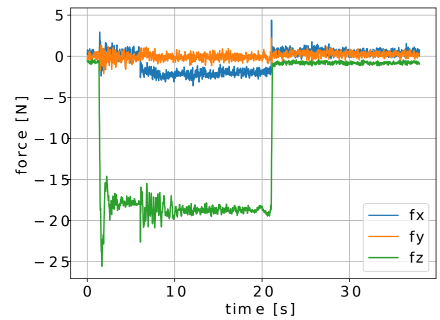



(b) Resulting motion of the end effector

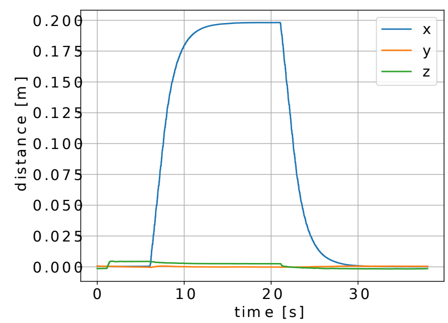
Fig. 7: Admittance Control under human interaction (shaking) on the end effector of the KR16 (a). The varying and noisy input still leads to smooth Cartesian motion (b), after integrating the joint accelerations. The compliance parameters were set to $m = 1\,\mathrm{kg}$, $I_{xx}, I_{yy}, I_{zz} = 0.04\,\mathrm{kgm}^2$, $c_t = 500\,{}^{\mathrm{N}}\!/\mathrm{m}$, $c_r = 30\,{}^{\mathrm{Nm}}\!/\mathrm{rad}$ and $k_t = 310\,{}^{\mathrm{Ns}}\!/\mathrm{m}$, $k_r = 75\,{}^{\mathrm{Nms}}\!/\mathrm{rad}$.




(a) UR10 sliding in contact with a stiff surface




(b) Contact forces measured by the sensor




(c) Distance covered during the sliding motion

Fig. 8: Force Control on a flat stiff surface, using the UR10. In the experiment, the manipulator has an Aluminum rod mounted to its end effector (a) and exerts a force of $20\,\mathrm{N}$ on the underlying surface (b) while sliding horizontally for approx. $20\,\mathrm{cm}$ (c). The compliance parameters were the same as in the experiment from Fig. 7, except for the increased translational damping $k_t = 900\,{}^{\mathrm{Ns}}\!/\mathrm{m}$.

To demonstrate the robustness of FDCC in singularities, we chose the final joint configuration of Fig. 10(a), because it encapsulates both the singularity as induced by the ending workspace, as well as the singularity through the ligning-up of several joints. Fig. 10 shows the results for the LWA4P. It can be observed, that while reaching its geometrical limits and especially under the continuing presence of noisy pulling forces, the system stays stable.

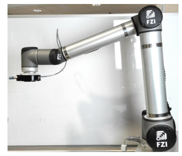
Additional impressions about FDCC are provided in a short video accompanying this paper.

## VI. CONCLUSIONS

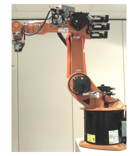
In this paper, we proposed Forward Dynamics Compliance Control (FDCC) as a new means to realize Cartesian add-on compliance on position-controlled manipulators, through integrating virtual Impedance Control on top of combined Force - Admittance Control.

We showed, how the impedance component could be utilized to design and parametrize desired end effector compliance independently of the underlying hardware. The Admittance Control component proved contact stability and smooth interaction with a soft environment (human arm) and in contact with a stiff environment (workbench surface), on which it successfully exerted a target force during motion. In contrast to the classic approach of enforcing joint motion inversely on the manipulators to fulfil desired Cartesian motion, we went the opposite way and presented with the integration of forward dynamics into the control loop an alternative solution to the inverse kinematics problem. Our central idea to simulate the reaction motion of a generalized, virtual system under the presence of both virtual and measured end effector forces made the fusion of the three control concepts possible. This new principle resulted in smooth and robust behavior in singularities, which makes FDCC a powerful instrument to enhance position-controlled manipulators with Cartesian compliance. It can be used for object manipulation and environment interactions, such as in manufacturing cells or in assembly lines, where position-

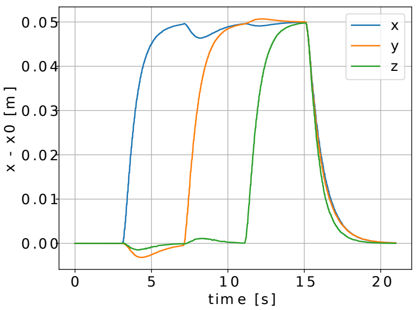
(a) Schunk LWA4P



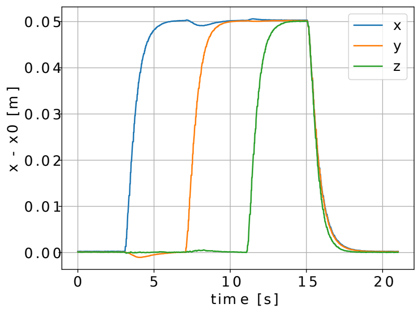
(b) Universal Robots UR10



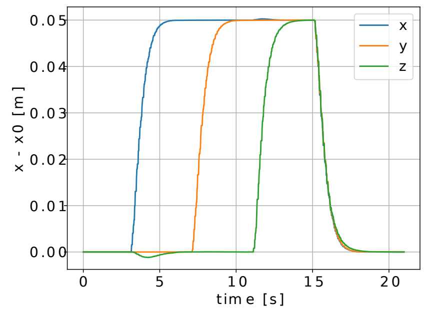
(c) KUKA KR16




(d)

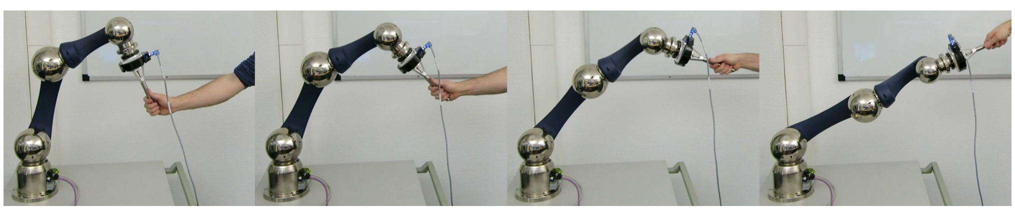



(e)

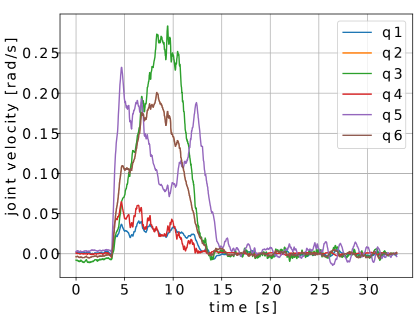



(f)

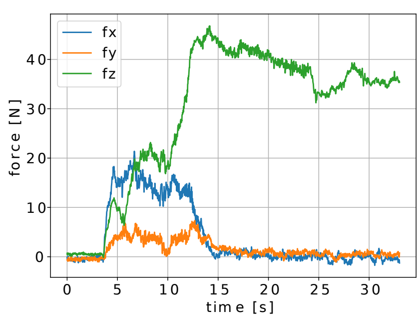
Fig. 9: Comparison of the position step responses (d) - (f) of three different manipulators (a) - (c) to displacements of $50\,\mathrm{mm}$ along the unit axis of each end effector. The compliance parameters were set analog to the values from Fig. 7 with $m = 1\,\mathrm{kg}$, $I_{xx}, I_{yy}, I_{zz} = 0.04\,\mathrm{kgm}^2$, $c_t = 500\,{}^{\mathrm{N}}\!/\!{}_{\mathrm{m}}$, $c_r = 30\,{}^{\mathrm{Nm}}\!/\!{}_{\mathrm{rad}}$ and $k_t = 310\,{}^{\mathrm{Ns}}\!/\!{}_{\mathrm{m}}$, $k_r = 75\,{}^{\mathrm{Nms}}\!/\!{}_{\mathrm{rad}}$.



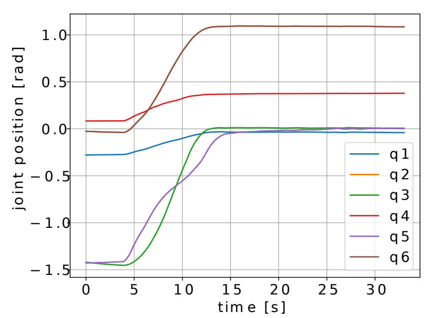
(a) Fully stretching the manipulator into singularity




(b) Commanded joint velocities




(c) Applied forces on the end effector




(d) Measured joint positions

Fig. 10: Testing the control behavior in singularities on the Schunk LWA4P robot. Especially in the fully stretched configuration (a), the commanded velocities are in safe margins (b), while external forces keep acting on the end effector (c). The manipulator behaves stable, as the measured joint positions indicate (d). The compliance parameters for this experiment were set as in the experiment from Fig. 7, i.e. no elevated damping was applied.

controlled robots are common. Future research could expand on high-level manipulation strategies on top of FDCC, exploiting the three different interfaces to simultaneously issue motion, force and compliance commands.

## ACKNOWLEDGMENT

This work was supported in part by the European Community Seventh Framework Program under Grant no. 608849 (EuRoC Project).

## REFERENCES

[1] Daniel E Whitney. Quasi-static assembly of compliantly supported rigid parts. *Journal of Dynamic Systems, Measurement, and Control*, 104(1):65–77, 1982.

[2] Bram Vanderborght, Alin Albu-Schäffer, Antonio Bicchi, Etienne Burdet, Darwin G Caldwell, Raffaella Carloni, M Catalano, Oliver Eiberger, Werner Friedl, Ganesh Ganesh, et al. Variable impedance actuators: A review. *Robotics and autonomous systems*, 61(12):1601–1614, 2013.

[3] M. T. Mason. Compliance and force control for computer controlled manipulators. *IEEE Transactions on Systems, Man, and Cybernetics*, 11(6):418–432, June 1981.

[4] Marc H Raibert and John J Craig. Hybrid position/force control of manipulators. *Journal of Dynamic Systems, Measurement, and Control*, 102(127):126–133, 1981.

[5] O. Khatib. A unified approach for motion and force control of robot manipulators: The operational space formulation. *IEEE Journal on Robotics and Automation*, 3(1):43–53, February 1987.

[6] N. Hogan. Impedance control - an approach to manipulation. i - theory. ii - implementation. iii - applications. *ASME Transactions Journal of Dynamic Systems and Measurement Control B*, 107:1–24, March 1985.

[7] Homayoun Seraji. Adaptive admittance control: An approach to explicit force control in compliant motion. In *Robotics and Automation, 1994. Proceedings., 1994 IEEE International Conference on*, pages 2705–2712. IEEE, 1994.

[8] J. K. Salisbury. Active stiffness control of a manipulator in cartesian coordinates. In *1980 19th IEEE Conference on Decision and Control including the Symposium on Adaptive Processes*, pages 95–100, Dec 1980.

[9] Jerry Pratt, Peter Dilworth, and Gill Pratt. Virtual model control of a bipedal walking robot. In *Robotics and Automation, 1997. Proceedings., 1997 IEEE International Conference on*, volume 1, pages 193–198. IEEE, 1997.

[10] Alin Albu-Schaffer and Gerd Hirzinger. Cartesian impedance control techniques for torque controlled light-weight robots. In *Robotics and Automation, 2002. Proceedings. ICRA'02. IEEE International Conference on*, volume 1, pages 657–663. IEEE, 2002.

[11] Alin Albu-Schaffer, Christian Ott, Udo Frese, and Gerd Hirzinger. Cartesian impedance control of redundant robots: Recent results with the dlr-light-weight-arms. In *Robotics and Automation, 2003. Proceedings. ICRA'03. IEEE International Conference on*, volume 3, pages 3704–3709. IEEE, 2003.

[12] Clive Loughlin, A Albu-Schäffer, S Haddadin, Ch Ott, A Stemmer, T Wimböck, and G Hirzinger. The dlr lightweight robot: design and control concepts for robots in human environments. *Industrial Robot: an international journal*, 34(5):376–385, 2007.

[13] Rainer Bischoff, Johannes Kurth, Günter Schreiber, Ralf Koeppe, Alin Albu-Schäffer, Alexander Beyer, Oliver Eiberger, Sami Haddadin, Andreas Stemmer, Gerhard Grunwald, et al. The kuka-dlr lightweight robot arm-a new reference platform for robotics research and manufacturing. In *Robotics (ISR), 2010 41st international symposium on and 2010 6th German conference on robotics (ROBOTIK)*, pages 1–8. VDE, 2010.

[14] BR Markiewicz. Analysis of the computed torque drive method and comparison with conventional position servo for a computer-controlled manipulator. 1973.

[15] RH Middletone and GC Goodwin. Adaptive computed torque control for rigid link manipulators. In *Decision and Control, 1986 25th IEEE Conference on*, volume 25, pages 68–73. IEEE, 1986.

[16] Stefano Chiaverini. Singularity-robust task-priority redundancy resolution for real-time kinematic control of robot manipulators. *IEEE Transactions on Robotics and Automation*, 13(3):398–410, 1997.

[17] Yoshihiko Nakamura and Hideo Hanafusa. Inverse kinematic solutions with singularity robustness for robot manipulator control. *ASME, Transactions, Journal of Dynamic Systems, Measurement, and Control*, 108:163–171, 1986.

[18] Charles W Wampler. Manipulator inverse kinematic solutions based on vector formulations and damped least-squares methods. *IEEE Transactions on Systems, Man, and Cybernetics*, 16(1):93–101, 1986.

[19] Stefano Chiaverini, Bruno Siciliano, and Olav Egeland. Review of the damped least-squares inverse kinematics with experiments on an industrial robot manipulator. *IEEE Transactions on control systems technology*, 2(2):123–134, 1994.

[20] Christian Ott, Ranjan Mukherjee, and Yoshihiko Nakamura. Unified impedance and admittance control. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pages 554–561. IEEE, 2010.

[21] Roy Featherstone. The calculation of robot dynamics using articulated-body inertias. *The International Journal of Robotics Research*, 2(1):13–30, 1983.

[22] Morgan Quigley, Ken Conley, Brian Gerkey, Josh Faust, Tully Foote, Jeremy Leibs, Rob Wheeler, and Andrew Y Ng. Ros: an open-source robot operating system. In *ICRA workshop on open source software*, volume 3, page 5. Kobe, 2009.

[23] Martin L Felis. Rbdl: an efficient rigid-body dynamics library using recursive algorithms. *Autonomous Robots*, pages 1–17, 2016.