International Journal of Social Robotics (2020) 12:1061–1073
https://doi.org/10.1007/s12369-019-00579-y

ORIGINAL RESEARCH

Simplified Stable Admittance Control Using End-Effector Orientations

Wen Yu[1] · **Adolfo Perrusquía**[1]

Accepted: 17 July 2019 / Published online: 30 July 2019

Abstract
Admittance control is used mainly for human–robot interaction. It transforms forces and torques to the desired position and orientation of the end effector. When the admittance control is in the task space, it needs the Jacobian matrix, while in the joint space, it requires the inverse kinematics. This paper modifies the admittance control using only the orientation components of the end-effector to avoid the calculation of the inverse kinematics and the Jacobian matrix. We use geometric properties, adaptive control and sliding mode control to approximate them. The stability of those controllers is proven. Experiments are presented in real time with a 2-DOF pan and tilt robot and a 4-DOF exoskeleton. The results of the experiment show the effectiveness of the proposed controllers.

Keywords Admittance control · Stability · Sliding mode control

1 Introduction

Human–robot cooperative control is an emerging field in robotics, which provides an interaction between human action and the robot [1]. The main objective of cooperative control is to generate specific tasks to combine human abilities and robot properties [2], for example, co-manipulation [3], haptic operation [4,5], learning from demonstrations [6]. Admittance control is one of the most common methods to make the connection between humans and robots [7].

The admittance control can be considered as a dynamic mapping of forces/torques to the movement (position or velocity). It uses forces/torques to admit a certain number of movements [8]. In general, the relationship between forces/torques and movement is imposed by a spring-mass damper system. The parameters of this system represent the robot's ability to follow the path given by the human operator [1].

Normal admittance control requires inverse kinematics in joint space,[1] so that the movement of the robot is linearized and decoupled globally [8,9]. The main problem with inverse kinematics is that their solutions (if they exist) are coupled by the position of the end-effector and kinematic parameters. The output trajectories of the admittance model are practically decoupled, which causes singularity problems in the joint solutions. The trajectories generated from the admittance control are in task space.[2] Although the inverse kinematics can be avoided, the output of the admittance model can not be applied to the robot joints. It requires the Jacobian to transform the control signals of the task space into the joint space [10]. The main problem of the Jacobian is that it depends on joint measures and kinematic parameters that are not always available, even more can make the closed-loop system unstable, because it presents the same problem of inverse kinematics, i.e., the Jacobian is coupled and the output of the admittance model is decoupled.

The classical admittance control has two loops [11]: the inner loop is a joint space control loop that uses the robot model and a proportional-derivative (PD) controller to improve the robustness against the modeling error. The external control loop is the admittance model that trans-

✉ Wen Yu
yuw@ctrl.cinvestav.mx

[1] Departamento de Control Automático, CINVESTAV-IPN (National Polytechnic Institute), Av.IPN 2508, 07360 Mexico City, Mexico

[1] Joint space is defined by a vector whose components are the translational and angular displacements of each joint of a robotic link.

[2] Task space (or Cartesian space) is defined by the position and orientation of the end effector of a robot.

forms the input forces into position/velocity movements. Both joint space and task space need the complete dynamic model of the robot. The modeling error hinders the precise realization of the desired admittance dynamics. This is the "accuracy/robustness dilemma in impedance control" [11]. There are some modeling error compensation methods for the admittance control, such as the adaptive algorithms [4,12], however, they all need a structure model of the dynamics of the robot [13,14]. It is difficult to estimate all the dynamics when the robot has many joints [15,16]. The gravity torques are the main factors of losing accuracy in the closed-loop system, therefore it is very important to compensate them [8,17]. Compensators without a model, such as neural networks [10], fuzzy control [18,19], robust control [20], do not need the mathematical models, but their control precisions are poor [11].

PID is one of the simplest model-free controllers that does not require dynamic knowledge [21]. For the admittance control, the PID control can be used to establish the cooperative application between the robot and the operator [8,17]. However, the PID control has transient performance problems and instability, caused by the integrator.

This paper proposes new admittance control methods to solve the following problems in the classical admittance control in human–robot cooperation tasks:

1. Classical admittance control in joint space needs the inverse kinematics.
2. Classical admittance control in task space needs the Jacobian matrix.
3. Classical admittance control is a model-based controller.
4. The PID admittance control has an integrator that destroys the performance of the control.

The key idea of our methods is that we only use a simplified model with the end effector orientations. The admittance model uses a force/torque sensor as the human–robot interface. The input forces are decoupled, while the positions of the end-effector are coupled. Both the input torques and the orientations of the simplified model are decoupled. These good properties can simplify our design process and avoid the complete solution of the inverse kinematics and the Jacobian matrix for this type of applications.

The new admittance controllers are designed in joint and task spaces. Semi-global asymptotic stability and global asymptotic stability are proven by sliding mode and adaptive compensations, respectively. The admittance controllers do not require inverse kinematics and Jacobian matrix. Geometric relations are applied to approximate them. Comparisons of real-time experiments between classical and proposed admittance controllers are presented using a 2-DOF (degree of freedom) pan and tilt robot and a 4-DOF exoskeleton.

2 Admittance Control in Joint Space

2.1 Human–Robot Cooperation with Admittance Control

The cooperation between humans and robots in joint space can be divided into two loops, see Fig. 1. The internal loop is the control of joint position, which forces each joint of the robot to follow the desired angles q_r. The classical position control algorithm is PD control. The external loop is the admittance model that generates the desired joint angles q_r. This model receives the force/torque measurements generated by the human and transforms them into the desired joint angles. The outer loop is closed by the human vision.

Figure 1a shows the classical scheme for human–robot cooperation in joint space, that uses the robot inverse kinematics to calculate q_r from the position components x_r obtained from the human forces/torques F. The position controller is a model-based controller that guarantees perfect position tracking. When the robot model is not available, the controller loses accuracy and the admittance control has bad performance. The inverse kinematics presents application problems since its solutions are coupled while the output position components x_r from the admittance model are decoupled.

Figure 1b shows our proposed method, which avoids the knowledge of the dynamics of the robot, and uses a simple PD control plus a compensator that will be explained in future sections. The inverse kinematics is changed to an orientation model. The main advantage is that it offers a direct relationship between the orientations of the end effector and the joint angles without using the solution of the inverse kinematics.

The impedance of a mechanical system describes the force-position relation by

$$Z(s) = \frac{F(s)}{x_r(s)} = M_a s^2 + B_a s + K_a, \tag{1}$$

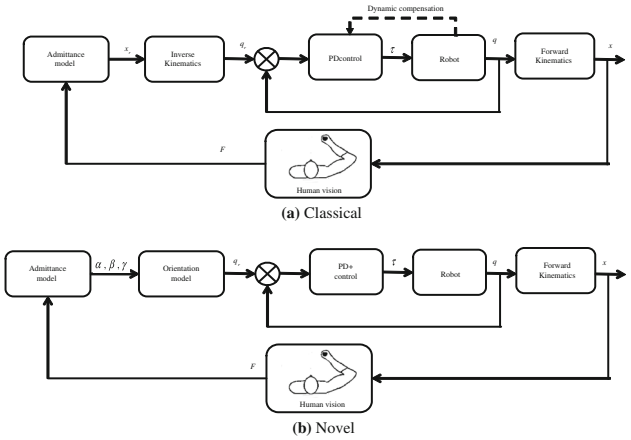

Fig. 1 Human–robot cooperation joint space schemes

where $F(s)$ is the force applied on the environment, $x_r(s)$ represents the position where the mechanical system contacts with the environment, $Z(s)$ is the environment impedance, M_a, B_a and K_a are the inertia, viscosity and stiffness of the mechanical system. The admittance of a mechanical system is the inverse relation of the impedance,

$$\frac{x_r(s)}{F(s)} = \frac{1}{M_a s^2 + B_a s + K_a}. \tag{2}$$

In a human–robot cooperation task, if the robot end-effector does not have contact with the environment, $F(s)$ denotes the external force/torque vector applied by the operator. $x_r(s)$ becomes the desired reference for the inner loop control. $F(s) \in R^m$, m is the dimension of the output vector of the force/torque sensor. $M_a, B_a, K_a \in R^{m\times m}$. Obviously, $M_a\ddot{x}_r + B\dot{x}_r + K_a x_r = F$. Usually, the desired reference $x_r(s)$ has six components as

$$\begin{bmatrix} X \\ Y \\ Z \\ \alpha \\ \beta \\ \gamma \end{bmatrix} = \frac{1}{M_a s^2 + B_a s + K_a} \begin{bmatrix} f_x \\ f_y \\ f_z \\ \tau_x \\ \tau_y \\ \tau_z \end{bmatrix}, \tag{3}$$

where X, Y, Z are the positions of the robot end-effector, α, β, γ are the end-effector orientations, f_x, f_y, f_z are the forces of the force/torque sensor, τ_x, τ_x, τ_z are the torques of the force/torque sensor.

2.2 Admittance Control in Joint Space Without Inverse Kinematic

Since x_r is in human space , i.e., task space, it should be transformed into joint space as

$$\begin{aligned} &\dot{x}_r = J(q)\dot{q}_r, \quad \dot{q}_r = J^{-1}(q)\dot{x}_r \\ &q_r = invk\,(x_r)\,, \end{aligned} \tag{4}$$

where q_r is the desired joint angles, $J(q)$ is the Jacobian of the robot, $invk\,(\cdot)$ is the inverse kinematic of the robot. The dynamics for an n-link robot is,

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) = \tau \tag{5}$$

where $q \in R^n$ represents the link positions, $M\,(q) \in R^{n\times n}$ is the inertia matrix, $C\,(q,\dot{q}) \in R^{n\times n}$ is centrifugal and Coriolis matrix, $G\,(q) \in R^n$ is vector of gravitational torques, $\tau \in R^n$ is the control torque vector for the joints. The classical PD control is

$$\tau = -K_p\,(q - q_r) - K_v\,(\dot{q} - \dot{q}_r) \tag{6}$$

where q_r is obtained from the admittance control (3) and (4) and $K_p, K_v \in R^{n\times n}$ are the proportional and derivative matrices gains, respectively.

This paper uses a simplified admittance control in joint space, that uses the end-effector orientations α, β, and γ in (3). For the basic relations of robot orientation and joint angles, we can avoid the use of the robot inverse kinematics, $invk\,(\cdot)$ in (4). As it is a cooperative task between the human and the robot, the robot must move slowly, so that the next position is closed to the previous one

$$q(t+1) = q(t) + q_r(t), \tag{7}$$

where $q(t)$ is the current joint position, $q(t+1)$ is the next joint position, q_r corresponds to the desired joint position obtained from the admittance model and the inverse kinematics.

Since $q(t+1) \approx q(t)$, then q_r must be as small as possible, which is easy to obtain by increasing the parameters of the admittance model. The position q_r is obtained by the robot inverse kinematics. However, we can avoid it by using an orientation model in the current position of the robot. This orientation model is reliable since the joint displacements are small and helps to obtain a direct relation between the orientations and the joint angles without using the inverse kinematics of the robot.

From robot geometric relations, each orientation component in the current pose can be expressed as a linear combination of the joint angles

$$\alpha = \sum_{i=1}^{n}(c_i q_i + d_i),\ \ \beta = \sum_{i=1}^{n}(c_i q_i + d_i),\ \ \gamma = \sum_{i=1}^{n}(c_i q_i + d_i) \tag{8}$$

where c_i and d_i are offsets of the orientation that depends of the robot configuration, q_i is the i-th joint angle, $q = [q_1 \cdots q_n]^\top$.

When the DOF is more than 2 and less than 7, i.e., $6 \geq n \geq 3$, we can use robot geometric relations to calculate the the orientations in function of the joint angles as in (8). We use one of the following methods to generate the desired joint angles q_i from the orientations:

1. The three orientations (α, β, γ) are obtained from the force-sensor torques (τ_x, τ_y, τ_z). The first three joint angles are of these three orientations. The other joint angles can be estimated from the forces of the force sensor (f_x, f_y, f_z), so that the linear combination of the orientation (8) is satisfied .
2. The operator can first move the robot to the desired position with the first three joint angles, then the operator move the robot to the desired orientation angles by chang-

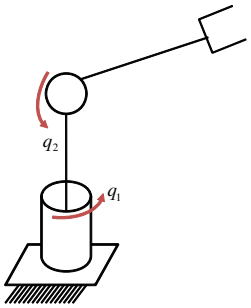

Fig. 2 2-DOF robot configuration

ing the orientation components. This two-step method is applied in many industrial robots. It is similar with the linear combination of the orientation (8).

When $n < 3$, the above approach can be used directly, because the three orientations are uncoupled and have direct relationships with the angles of the joint. So the two orientations are also decoupled. When we have more than six DOF, $n \geq 7$, the proposed method can not be used directly, since there are only six force/torque components, i.e., $m = 6$. For this case, we need additional assumptions, such as fix the extra joint at a certain value or use one force/torque component for two joint angles, to generate another reference.

Now we show the calculation method for different configurations.

2-DOF Robot

A typical 2-DOF robot is the pan and tilt robot which is shown in Fig. 2. The relation between the orientations and the joint angles are

$$\begin{aligned} \alpha &= \frac{\pi}{2} \\ \beta &= -q_2 \\ \gamma &= q_1 \end{aligned} \tag{9}$$

This is an one to one mapping. The admittance control (3) generates three orientations and three positions in task space. Two orientations in (9) generate two desired angles for the joint space control.

4-DOF Robot

A typical 4-DOF robot is shown in Fig. 3. The relation between the orientations and the joint angles are

$$\begin{aligned} \alpha &= \frac{\pi}{2} - q_3 \\ \beta &= -q_2 - q_4 \\ \gamma &= q_1. \end{aligned} \tag{10}$$

There exists a linear combination in the β orientation in (10). First we divide the orientation β into two terms $\beta = \beta_{q_2} + \beta_{q_4}$. Then we use both torque and force at Y direction to generate

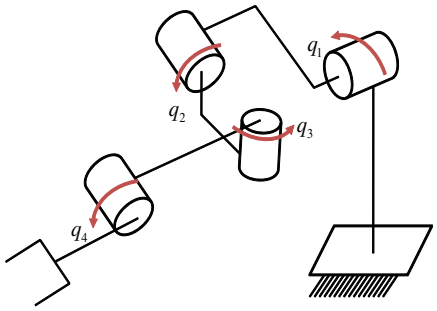

Fig. 3 4-DOF robot configuration

β_{q_2} and β_{q_4}. The joint positions from the force/torque sensor are

$$\begin{aligned} q_3 &= \frac{\pi}{2} - \alpha, \quad \alpha = admit\,(\tau_x) \\ q_1 &= \gamma, \quad \gamma = admit\,(\tau_z) \\ q_2 &= -\beta_{q_2}, \quad \beta_{q_2} = admit\,(\tau_y) \\ q_4 &= -\beta_{q_4}, \quad \beta_{q_4} = admit\,(f_y) \end{aligned} \tag{11}$$

where $admit\,(\cdot)$ is the admittance model (3).

5-DOF Robot

The orientations of the 5-DOF robot end-effector have the following relations with the joint angles

$$\begin{aligned} \alpha &= \frac{\pi}{2} - q_3 \\ \beta &= -q_2 - q_4 \\ \gamma &= q_1 + q_5 \end{aligned} \tag{12}$$

So β and γ have the linear combinations of the joint angles. We can use the similar method as the 4-DOF robot to separate the orientations into two terms and use additional force components as

$$\begin{aligned} q_3 &= \frac{\pi}{2} - \alpha, \quad \alpha = admit\,(\tau_x) \\ q_2 &= \beta_{q_2}, \quad \beta_{q_2} = admit\,(\tau_y) \\ q_1 &= \gamma_{q_1}, \quad \beta_{q_1} = admit\,(\tau_z) \\ q_4 &= -\beta_{q_4}, \quad \beta_{q_4} = admit\,(f_y) \\ q_5 &= \gamma_{q_5}, \quad \gamma_{q_5} = admit\,(f_z) \end{aligned} \tag{13}$$

where $admit\,(\cdot)$ is the admittance model (3).

6-DOF Robot

For a 6-DOF robot, the orientations satisfy

$$\begin{aligned} \alpha &= \frac{\pi}{2} - q_3 - q_6 \\ \beta &= -q_2 - q_4 \\ \gamma &= q_1 + q_5 \end{aligned} \tag{14}$$

Here all orientations have linear combinations of the joint angles. Each of them are divided into two terms, then

$$\begin{aligned} q_3 &= \frac{\pi}{2} - \alpha_{q_3}, \quad \alpha_{q_3} = admit\,(\tau_x) \\ q_6 &= -\alpha_{q_6}, \quad \alpha_{q_6} = admit\,(f_x) \\ q_2 &= -\beta_{q_2}, \quad \beta_{q_2} = admit\,(\tau_y) \\ q_4 &= -\beta_{q_4}, \quad \alpha_{q_4} = admit\,(f_y) \\ q_1 &= \gamma_{q_1}, \quad \gamma_{q_1} = admit\,(\tau_z) \\ q_5 &= \gamma_{q_5}, \quad \gamma_{q_5} = admit\,(f_z) \end{aligned} \tag{15}$$

where $admit\,(\cdot)$ is the admittance model (3).

2.3 Stable Admittance Control with Adaptive Compensation

In order to assure stability, the following model-based admittance control is needed [9,22]

$$\tau = M(q)J^{-1}(q)(u_i - \dot{J}(q)\dot{q}) + C(q,\dot{q})\dot{q} + G(q) \tag{16}$$

where $u_i = \ddot{x}_r - K_v(\dot{x} - \dot{x}_r) - K_p(x - x_r)$ is the proportional-derivative (PD) control, K_p, $K_v \in R^{n\times n}$ denote the gains of the controller, x and x_r are defined as in Fig. 1. This classical admittance controller is robust with respect to modeling error by using an inner position control u_i. However, it needs the complex model of the robot (5).

In this paper, we use an adaptive algorithm to solve the model problem [23]. The robot dynamics (5) can be linearly parameterized as [24]

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) = Y_s\Theta \tag{17}$$

where $Y_s = Y(q,\dot{q},\ddot{q}) \in R^{n\times p}$ is a regressor of known nonlinear functions, $\Theta \in R^p$ is the parameter vector. When M, C, and G are unknown, (17) can be written as

$$\hat{M}(q)\ddot{q} + \hat{C}(q,\dot{q})\dot{q} + \hat{G}(q) = Y_s\hat{\Theta} \tag{18}$$

where $\hat{M}$, $\hat{C}$, $\hat{G}$ and $\hat{\Theta}$ are estimates of M, C, G and Θ, respectively.

The expression (18) can be also written as

$$\tilde{M}(q)\ddot{q} + \tilde{C}(q,\dot{q})\dot{q} + \tilde{G}(q) = Y_s\tilde{\Theta} + d \tag{19}$$

where $\tilde{M} = \hat{M} - M^*$, $\tilde{C} = \hat{C} - C^*$, $\tilde{G} = \hat{G} - G^*$, and $\tilde{\Theta} = \hat{\Theta} - \Theta^*$, d is the bounded unknown modeling error $\|d\| \le \bar{d}$, M^*, C^*, G^*, and Θ^* are unknown optimal parameters.

The following properties will be used to prove stability [24,25],

1. The inertia matrix $M(q)$ is symmetric positive definite, and there exists a number $\beta_1 > 0$ such that

$$0 < \lambda_m\{M(q)\} \le \|M(q)\| \le \lambda_M\{M(q)\} \le \beta_1 \tag{20}$$

where $\lambda_M\{A\}$ and $\lambda_m\{A\}$ are the maximum and minimum eigenvalues of the matrix A. The norm $\|A\| = \sqrt{\lambda_M(A^TA)}$ stands for the induced Frobenius norm.

2. For the centripetal and Coriolis matrix $C(q,\dot{q})$, there exists a number $\beta_2 > 0$ such that

$$\|C(q,\dot{q})\| \le \beta_2\|\dot{q}\| \tag{21}$$

and $\dot{M}(q) - 2C(q,\dot{q})$ is skew-symmetric, i.e.

$$\dot{q}^T\left[\dot{M}(q) - 2C(q,\dot{q})\right]\dot{q} = 0 \tag{22}$$

The norm $\|b\|$ of the vector $b \in R^n$ stands for the vector Euclidean norm.

3. For the gravitational torques vector $G(q)$, there exists a number $\beta_3 > 0$ such that

$$\|G(q)\| \le \beta_3 \tag{23}$$

We define the auxiliary tracking error Ω as

$$\Omega = (\dot{q} - \dot{q}_r) + \Lambda\,(q - q_r) \tag{24}$$

where $\Lambda = \Lambda^T > 0$. The PD control in (6) becomes $\tau = -K_s\Omega$, $K_s \in R^{n\times n}$ denotes the matrix gain of the PD control. With the adaptive estimation (18), the admittance control (16) is simplified as

$$\tau = -K_s\Omega + Y_s\hat{\Theta} \tag{25}$$

The closed-loop system is (5) and (25)

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) = -K_s\Omega + Y_s\hat{\Theta}$$

Using (19), the closed-loop system is

$$M(q)\dot{\Omega} + (C(q,\dot{q}) + K_s)\Omega = Y_{s_1}\tilde{\Theta}_{s_1} - Y_{s_2}\Theta_{s_2} \tag{26}$$

where $\Theta = \Theta_{s_1} + \Theta_{s_2}$, $Y_s = Y_{s_1} + Y_{s_2}$, $Y_{s_1}\tilde{\Theta}_{s_1} = \tilde{G}(q)$, $Y_{s_2}\Theta_{s_2} = M(q)\ddot{q}_s + C(q,\dot{q})\dot{q}_s - d$, $\dot{q}_s = \dot{q}_r - \Lambda(q - q_r)$, and $\ddot{q}_s = \frac{d}{dt}\dot{q}_s$. The PD control with adaptive gravity compensation is

$$\tau = -K_s\Omega + Y_{s_1}\hat{\Theta}_{s_1} \tag{27}$$

The following theorem gives the stability proof of the admittance control with adaptive compensation (27).

Theorem 1 *Consider the robot dynamic (5) controlled by (27), if K_s satisfy the following condition:*

$$\lambda_m(K_s) \geq \bar{k} \tag{28}$$

where $\lambda_m(K_s)$ is the minimum eigenvalue of K_s, $\bar{k}$ is the estimated upper bound of $\|Y_{s_2}\Theta_{s_2}\|$, and the parameters are updated as

$$\frac{d}{dt}\hat{\Theta}_{s_1} = -K_\Theta^{-1} Y_{s_1}^T \Omega \tag{29}$$

then the closed-loop system is semi-globally asymptotically stable.

Proof We proposed the following Lyapunov function

$$V = \frac{1}{2}\Omega^T M(q)\Omega + \frac{1}{2}\tilde{\Theta}_{s_1}^T K_\Theta \tilde{\Theta}_{s_1} \tag{30}$$

The first term corresponds to the kinetic energy of the closed-loop system, the second one is the adaptive term, $K_\Theta \in R^{p\times p}$ denote its matrix gain. The time derivative of (30) along (26) is

$$\begin{aligned}\dot{V} &= \Omega^T M(q)\dot{\Omega} + \frac{1}{2}\Omega^T \dot{M}(q)\Omega + \tilde{\Theta}_{s_1}^T K_\Theta \left(\frac{d}{dt}\tilde{\Theta}_{s_1}\right) \\ &= -\Omega^T K_s \Omega + \tilde{\Theta}_{s_1}^T \left(K_\Theta \left(\frac{d}{dt}\tilde{\Theta}_{s_1}\right) + Y_{s_1}^T \Omega\right) \\ &\quad - \Omega^T Y_{s_2}\Theta_{s_2}\end{aligned} \tag{31}$$

Because $\frac{d}{dt}\hat{\Theta}_{s_1} = \frac{d}{dt}\tilde{\Theta}_{s_1} - \frac{d}{dt}\Theta_{s_1}^* = \frac{d}{dt}\tilde{\Theta}_{s_1}$, from (31) and the adaptation law (29), the time derivative of V reduces to:

$$\begin{aligned}\dot{V} &= -\Omega^T K_s \Omega - \Omega^T Y_{s_2}\Theta_{s_2} \\ &\leq -\lambda_m(K_s)\|\Omega\|^2 + \|\Omega\|\|Y_{s_2}\Theta_{s_2}\|\end{aligned}$$

Using the properties (20) and (21) on $\|Y_{s_2}\Theta_{s_2}\|$ becomes

$$\|Y_{s_2}\Theta_{s_2}\| \leq \|M(q)\|\|\ddot{q}_s\| + \|C(q,\dot{q})\|\|\dot{q}_s\| + \bar{d} \leq \bar{k}$$

where $\bar{k} = f(\ddot{q}_s, \dot{q}_s, \dot{q}, \beta_i)$ is a state dependent function. Therefore, $\dot{V}$ is:

$$\dot{V} \leq -\lambda_m(K_s)\|\Omega\|^2 + \|\Omega\|\bar{k} \tag{32}$$

This implies that there exists a large enough gain K_s such that $\lambda_m(K_s) \geq \bar{k}$ and Ω converges into a set-bounded $\mu = \frac{\bar{k}}{\lambda_m(K_s)}$ as $t \to \infty$. So the time derivative of the vector parameters error $\left(\frac{d}{dt}\tilde{\Theta}_{s_1}\right)$ converge into a set-bounded ϵ when $\Omega \to \mu$ as $t \to \infty$. □

The PD control with adaptive compensation (27) requires a large K_s to cancel the uncertainty $Y_{s_2}\Theta_{s_2}$. This can be also removed by a more simple compensator, sliding mode control. PD control with sliding mode compensation is

$$\tau = -K_s\Omega - K_m sgn(\Omega) \tag{33}$$

where $K_m \in R^{n\times n}$ is the sliding mode matrix gain, $sgn(x) = [sgn(x_1), sgn(x_2), \ldots, sgn(x_n)]^T$. The following theorem gives the stability proof of the admittance control with the sliding mode compensation (33).

Theorem 2 *Consider the robot dynamic (5) controlled by (33), if the following conditions are satisfied*

$$\begin{aligned}K_s &> 0 \\ K_m &\geq \bar{k},\end{aligned} \tag{34}$$

where $\bar{k}$ is the estimated upper bound of $\|Y_s\Theta\|$, then the closed-loop system is globally asymptotically stable.

Proof We proposed the following Lyapunov function

$$V(\Omega) = \frac{1}{2}\Omega^T M(q)\Omega \tag{35}$$

Taking the time derivative of (35), and $sgn(\Omega)\Omega = \|\Omega\|$

$$\begin{aligned}\dot{V} &= \Omega^T M(q)\dot{\Omega} + \frac{1}{2}\Omega^T \dot{M}(q)\Omega \\ &= -\Omega^T K_s\Omega - K_m\|\Omega\| - \Omega^T Y_s\Theta \\ &\leq -\lambda_m(K_s)\|\Omega\|^2 - (K_m - \|Y_s\Theta\|)\|\Omega\|\end{aligned}$$

Using the properties (20)–(23) on $\|Y_s\Theta\|$

$$\|Y_s\Theta\| \leq \|M(q)\|\|\ddot{q}_s\| + \|C(q,\dot{q})\|\|\dot{q}_s\| + \|G(q)\| \leq \bar{k}$$

With the condition (34),

$$\dot{V} \leq -\lambda_m(K_s)\|\Omega\|^2 \leq 0 \tag{36}$$

By LaSalle Lemma, Ω converges to zero as $t \to \infty$, and the closed-loop system is globally asymptotically stable. □

3 Admittance Control in Task Space

The admittance control (3) gives the positions and orientations of the end-effector in task space. If we design a controller in task space, we will not use the inverse kinematics of the robot $invk(\cdot)$ as (4). The task space scheme is similar to Fig. 1, where it differs in the use of the Jacobian matrix instead of the inverse kinematics, also note that the

Jacobian is in the inner loop and not in the outer loop as the inverse kinematics.

The dynamic (5) in joint space can be written in task space (Cartesian space) as,

$$M_x\ddot{x} + C_x\dot{x} + G_x = f_\tau \tag{37}$$

where $x \in R^m$ is the vector of the end-effector position and orientation in Cartesian space, $M_x = J^{-T}M(q)J^{-1}$, $C_x = J^{-T}C(q,\dot{q})J^{-1} - M_x\dot{J}J^{-1}$, $G_x = J^{-T}G(q)$, $f_\tau = J^{-T}\tau$, J is the Jacobian of the robot defined in (4). M_x denotes the mass matrix, C_x stands for the Cartesian centripetal and Coriolis forces matrix, G_x is the gravity forces vector.

When the control f_τ in (37) is applied to the robot dynamics, it needs to be transformed into control torque as

$$\tau = J^T f_\tau \tag{38}$$

Here we uses two types of Jacobian: analytical Jacobian J_a and geometric Jacobian J_g [26]. The analytical Jacobian J_a is obtained directly by differentiating the forward kinematic as in $J(q)$ in (4), and J in (37), so $J_a = J(q) = J$. The geometric Jacobian J_g considers the geometric relation, it can be represented as $J_g = \left[J_{gv}^T, J_{g\omega}^T\right]^T$, where J_{gv} is the linear velocity Jacobian and $J_{g\omega}$ is the angular velocity Jacobian.

The Jacobian gives the mapping from joint space to task space and has two components, the linear velocity Jacobian and the angular velocity Jacobian. The linear velocity Jacobian requires kinematics parameters of the robot and joint measures. While the angular velocity Jacobian only requires joint measures.

The relation between the geometric and analytical Jacobians is [26]

$$J_a = \begin{bmatrix} I & 0 \\ 0 & T(O)^{-1} \end{bmatrix} J_g \tag{39}$$

where $T(O)$ is a rotation matrix of the orientation components, $O = \left[\alpha, \beta, \gamma\right]^T$.

For the end-effector, the linear velocity component of the analytical Jacobian is the same as the geometric Jacobian. In this paper, we will use the angular velocity component of the analytical Jacobian J_a. Taking the time derivative of (8)

$$\dot{\alpha} = \sum_{i=1}^{n} c_i\dot{q}_i \quad \dot{\beta} = \sum_{i=1}^{n} c_i\dot{q}_i \quad \dot{\gamma} = \sum_{i=1}^{n} c_i\dot{q}_i \tag{40}$$

So the relation between the time derivative of the orientations and the joint velocities only lies in c_i. The angular velocity component of the analytical Jacobian can be expressed as

$$\left[\dot{\alpha}, \dot{\beta}, \dot{\gamma}\right]^T = J_{a\omega}\dot{q} \tag{41}$$

where $J_{a\omega}$ is the angular velocity component of the analytical Jacobian J_a which is scalar with c_i in each orientation.

We will use $J_{a\omega}^T$ (or $\hat{J}_{a\omega}^T$) to calculate the control torque as in (38). Now we give the estimation method for different configurations.

2-DOF Robot

The 2-DOF robot is shown in Fig. 2. The relation between the orientations and the joint angles in the geometric Jacobian is

$$J_{g\omega}(q) = \begin{bmatrix} 0 & \sin(q_1) \\ 0 & -\cos(q_1) \\ 1 & 0 \end{bmatrix} \tag{42}$$

This Jacobian is easy to compute and requires joint measures. We take the time derivative of (9) to obtain the angular velocity of the analytical Jacobian as in (41)

$$\dot{\alpha} = 0, \quad \dot{\beta} = -\dot{q}_2, \quad \dot{\gamma} = \dot{q}_1$$

So

$$\begin{aligned} J_{a\omega} &= T(O)^{-1}J_{g\omega} \\ &= \begin{bmatrix} \cos(q_1) & \sin(q_1) & 0 \\ -\sin(q_1) & \cos(q_1) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & \sin(q_1) \\ 0 & -\cos(q_1) \\ 1 & 0 \end{bmatrix} \\ &= \begin{bmatrix} 0 & 0 \\ 0 & -1 \\ 1 & 0 \end{bmatrix} \end{aligned} \tag{43}$$

Notice the the analytical Jacobian in (43) has similar structure as the geometric Jacobian in (42).

4-DOF Robot

The 4-DOF robot is shown in Fig. 3, which has the following angular velocity Jacobian:

$$J_{g\omega} = \begin{bmatrix} 0 & s_1 & -c_1s_2 & c_1c_2s_3 + c_3s_1 \\ 0 & -c_1 & -s_1s_2 & c_2s_1s_3 - c_1c_3 \\ 1 & 0 & c_2 & s_2s_3 \end{bmatrix} \tag{44}$$

where $s_i = \sin(q_i)$, $c_i = \cos(q_i)$, $i = 1, 2, 3, 4$. Using the same method as the 2-DOF,

$$\dot{\alpha} = \dot{q}_3, \quad \dot{\beta} = -\dot{q}_2 - \dot{q}_4, \quad \dot{\gamma} = \dot{q}_1$$

So

$$J_{a\omega} = \begin{bmatrix} 0 & 0 & -1 & 0 \\ 0 & -1 & 0 & -1 \\ 1 & 0 & 0 & 0 \end{bmatrix} \tag{45}$$

5-DOF Robot

Taking the time derivative of (12), the angular velocity component of the analytical Jacobian is:

$$J_{a\omega} = \begin{bmatrix} 0 & 0 & -1 & 0 & 0 \\ 0 & -1 & 0 & -1 & 0 \\ 1 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{46}$$

6-DOF Robot

Taking the time derivative of (15), the angular velocity component of the analytical Jacobian is:

$$J_{a\omega} = \begin{bmatrix} 0 & 0 & -1 & 0 & 0 & -1 \\ 0 & -1 & 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \tag{47}$$

In order to avoid the coupling, we only use the orientation of the robot. However, when the orientations are linear combinations of more than two joint angles as in (8), there are multiple solutions of the joint angles. To solve this problem, one of the following methods can be used:

1. Modify the parameter c_i, such that all joint movements contribute to the orientations. For example, the 6-DOF robot the angular velocity component of the analytical Jacobian is changed as

$$\hat{J}_{a\omega} = \begin{bmatrix} 0 & 0 & -1 & 0 & 0 & 0.1 \\ 0 & -0.75 & 0 & -0.25 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{48}$$

   Where $\hat{J}_{a\omega}$ is an approximation of $J_{a\omega}$.
2. As in joint space case, we can divide the Jacobian in two parts: $\hat{J}_{a\omega_1}$ only includes the positions of the joint angles, $\hat{J}_{a\omega_2}$ only has the orientation of joint angles. For a 6-DOF robot,

$$\hat{J}_{a\omega_1} = \begin{bmatrix} 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad \hat{J}_{a\omega_2} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \tag{49}$$

   Clearly, we do not need the inverse of the Jacobian. The admittance control (16) in task space becomes

$$\begin{aligned} f_\tau &= M_x u_x + C_x \dot{x} + G_x \\ u_x &= \ddot{x}_r - K_v(\dot{x} - \dot{x}_r) - K_p(x - x_r) \end{aligned} \tag{50}$$

This controller does not require the inverse of the Jacobian. However, it has the same problems as the joint space controller, it requires the task space model (37).

This paper uses a task space admittance control. In order to avoid the estimation of the dynamic model, we use PID control as

$$\begin{aligned} f_\tau &= -K_s \Omega_x \\ \Omega_x &= (\dot{x} - \dot{x}_r) + \Lambda (x - x_r) - \xi \\ \dot{\xi} &= K_i (x - x_r) \end{aligned} \tag{51}$$

where K_s is the matrix gain of the PD control, K_i denote the integral matrix gain, the auxiliary tracking error Ω_x is defined as

$$\Omega_x = (\dot{x} - \dot{x}_r) + \Lambda (x - x_r) \tag{52}$$

where $\Lambda = \Lambda^T > 0$.

The stability of this admittance PID control has been proven in our previous paper [10,27]. Nevertheless, PID control may reduce bandwidth of the closed loop system. Similar with the joint space, we may use sliding mode compensation to avoid the integration action. In order to decrease the chattering problem in the normal sliding mode control (33), we use the following second-order sliding mode compensation to replace the integrator of the PID control (51).

$$\begin{aligned} f_\tau &= -K_s \Omega_x - k_1 \|\Omega_x\|^{1/2} sgn(\Omega_x) + \xi \\ \dot{\xi} &= -k_2 sgn(\Omega_x) \\ \Omega_x &= (\dot{x} - \dot{x}_r) + \Lambda (x - x_r) \end{aligned} \tag{53}$$

where k_1, k_2 are the sliding mode gains to be designed. The closed loop system with the controller (53) is

$$\begin{aligned} M_x \dot{\Omega}_x + (C_x + K_s)\Omega_x &= \xi - Y_x \Theta_x - k_1 \|\Omega_x\|^{1/2} sgn(\Omega_x) \\ \dot{\xi} &= -k_2 sgn(\Omega_x) \end{aligned}$$

where $Y_x \Theta_x$ is the parameterization of (37), $\dot{x}_s = \dot{x}_d - \Lambda(x - x_r)$.

The following theorem gives the stability and finite time convergence of the PD control with sliding mode compensation.

Theorem 3 *If the sliding mode* (53) *satisfy*

$$k_1 > \bar{k}_x, \quad k_2 > \sqrt{\frac{2}{k_1 - \bar{k}_x}} \frac{(k_1 - \bar{k}_x)(1 + p)}{(1 - p)} \tag{54}$$

where p is some chosen constant, $0 < p < 1$, $\bar{k}_x$ is the upper bound of $Y_x \Theta_x$, then the tracking error Ω_x is stable, and it converge to zero in finite time, i.e.,

$$\|\Omega_x\| \to 0$$

Proof Consider the next Lyapunov function

$$V = \frac{1}{2}\zeta^T P \zeta$$

where $\zeta = [\|\Omega_x\|^{1/2} sgn(\Omega_x), \xi]^T$, $P = \frac{1}{2}\begin{bmatrix} 4k_2 + k_1^2 & -k_1 \\ -k_1 & 2 \end{bmatrix}$. It is continuous everywhere but not differentiable at $\Omega_x = 0$. So

$$V = 2k_2\|\Omega_x\| + \frac{\xi^2}{2} + \frac{1}{2}(k_1\|\Omega_x\|^{1/2} sgn(\Omega_x) - \xi)^2$$

Since $k_1 > 0, k_2 > 0$, V is positive definite, and

$$\lambda_m(P)\|\zeta\|^2 \leq V \leq \lambda_M(P)\|\zeta\|^2 \tag{55}$$

Here $\|\zeta\|^2 = \|\Omega_x\| + \|\xi\|^2$. The time derivative of V is

$$\dot{V} = -\frac{1}{\|\Omega_x\|^{1/2}}(\zeta^T Q_1 \zeta - \|\Omega_x\|^{1/2} Y_x \Theta_x Q_2^T \zeta) \tag{56}$$

where $Q_1 = \frac{k_1}{2}\begin{bmatrix} 2k_2 + k_1^2 & -k_| \\ -k_1 & 1 \end{bmatrix}$, $Q_2 = \begin{bmatrix} 2k_2 + \frac{k_1^2}{2} \\ -\frac{k_1}{2} \end{bmatrix}$. Using the next inequality

$$\dot{V} \leq -\frac{1}{\|\Omega_x\|^{1/2}}\zeta^T Q_3 \zeta \tag{57}$$

where

$$Q_3 = \frac{k_1}{2}\begin{bmatrix} 2k_2 + k_1^2 - \left(\frac{4k_2}{k_1} + k_1\right))\bar{k}_x & -(k_1 + 2\bar{k}_x) \\ -(k_1 + 2\bar{k}_x) & 1 \end{bmatrix} \tag{58}$$

with the condition (54), $Q_3 > 0$, $\dot{V}$ is negative definite. From (55)

$$\|\Omega_x\|^{1/2} \leq \|\zeta\| \leq \frac{V^{1/2}}{\lambda_m^{1/2}(P)} \tag{59}$$

So

$$\dot{V} \leq -\frac{1}{\|\Omega_x\|^{1/2}}\zeta^T Q_3 \zeta \leq -\gamma V^{1/2} \tag{60}$$

where $\gamma = \frac{\lambda_{\min}^{1/2}(P)\lambda_{\min}(Q_3)}{\lambda_{\max}(P)} > 0$. Because the solution of the differential equation $\dot{y} = -\gamma y^{1/2}$ is

$$y_t = \left[y_0 - \frac{\gamma}{2}t\right]^2$$

y_t converges to zero in finite time and reach to zero after $t = \frac{2}{\gamma}y_0$. Using comparison principle for (60), when $V(\zeta_0) \leq y_0$, $V_t \leq y_t$. So V_t (or Ω_x) converges to zero after $T = \frac{2}{\gamma}V_0^{1/2}(\zeta_0)$. □

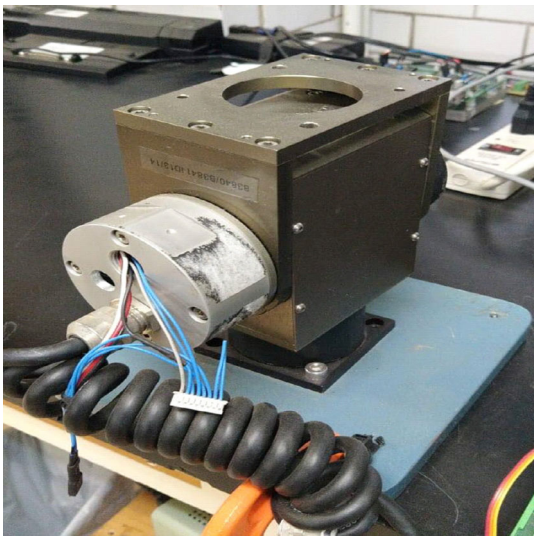

Fig. 4 2-DOF Pan and Tilt robot

4 Experiment Results

In order to test our simplified stable admittance controllers in joint space and task space, we use two robots: the 2-DOF pan and tilt robot shown in Fig. 4 and the 4-DOF exoskeleton shown in Fig. 5. Both robots are controlled by the human operator with the Schunk force/torque (F/T) sensor (see Fig. 6). The real time environment is the Simulink and Matlab 2012. The communication protocol is the controller area network (CAN bus), which enables the PC to communicate with the actuators and the F/T sensor. For both joint and task space, the controller gains are tuned manually until a satisfactory response is obtained.

At each time step, the (human) user applies a certain amount of force/torque to perform random movements in the position of the robot through the admittance and orientation models. It is compared classic model-based admittance control with our proposed controllers (adaptive and first order sliding compensation in the joint space, PID and second order sliding compensation in task space). The modeling error is

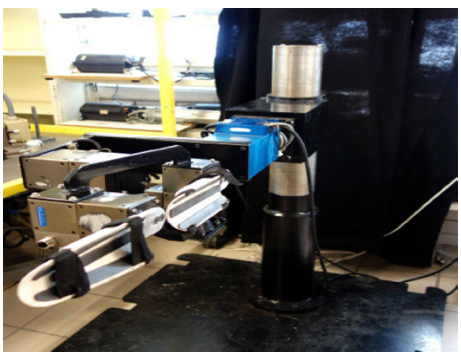

Fig. 5 4-DOF exoskeleton robot

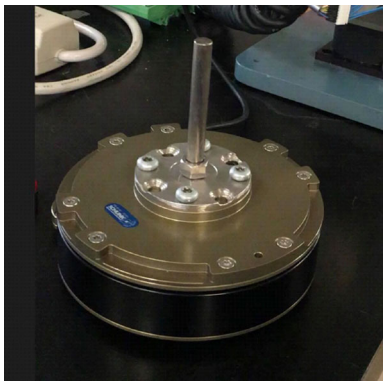

Fig. 6 Schunk force/torque sensor

Table 1 2-DOF robot: joint space control gains

Gain	Classical PD	Adaptive PD	Sliding PD
K_p	diag{50×10^2}	–	–
K_v	diag{100}	–	–
Λ	–	diag{90}	
K_m	–	–	diag{0.15}
K_Θ	–	1×10^{-3}	–
K_s	–	diag{0.9}	

assumed as a disturbance in the control loop. The dynamics of the robot is partially unknown, in addition, the robots present a high friction in each joint, which is also considered as a disturbance.

We first test the joint space controllers. The controller gains are tuned manually until we get a satisfactory response. The proposed controller gains of the 2-DOF robot are given in Table 1

Here diag{val} denotes a diagonal matrix of $n \times n$ with diagonal values val for joint space and $m \times m$ for task space. The admittance model is designed and proposed such that (7) is satisfied

$$admit(\cdot) = \frac{1}{M_{a_i} s^2 + B_{a_i} s + K_{a_i}}$$

where $M_{a_i} = 1$, $B_{a_i} = 140$, $K_{a_i} = 4000$ with $i = 1, 2, 3$. The best scenario is when the robot dynamics is known and the system does not have any disturbance, then both controllers achieve the tracking task as it is shown in the simulation study of the Fig. 7.

The worst case is when the robot dynamics is unknown and there exists disturbances, where classical PD cannot achieve the control task accurately as it is shown in the experimental results of the Fig. 8.

We do the same methodology for the 4-DOF robot. The proposed control gains for the 4-DOF exoskeleton are given in Table 2.

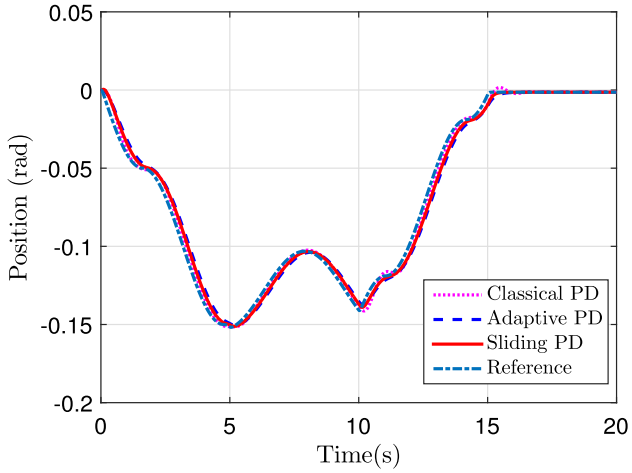

Fig. 7 2-DOF simulation results: joint position q_1

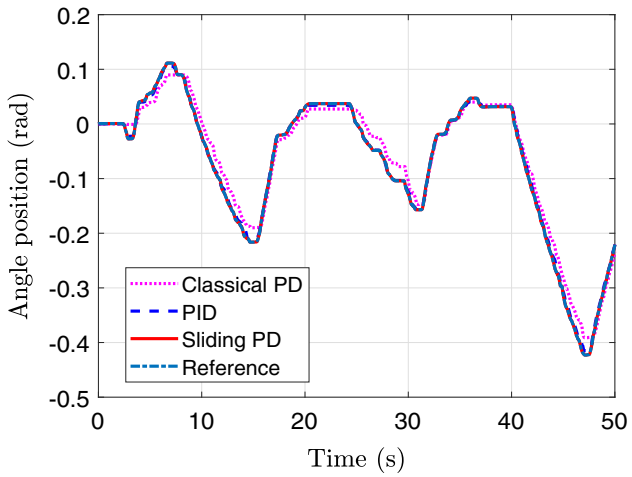

Fig. 8 2-DOF experiments results: joint position q_2

Table 2 4-DOF exoskeleton robot: joint space control gains

Gain	Classical PD	Adaptive PD	Sliding PD
K_p	diag{4×10^3}	–	–
K_v	diag{10}	–	–
Λ	–	diag{90}	
K_m	–	–	diag{1}
K_Θ	–	diag{0.1}	–
K_s	–	diag{3}	diag{2}

The admittance model is the same as above, with $i = 1, 2, 3, 4$. We only present the flexion-extension results of the shoulder and elbow because they present gravitational torques effect. The best scenario results are given in the simulations of Fig. 9.

The comparison results of the worst case are given in Fig. 10.

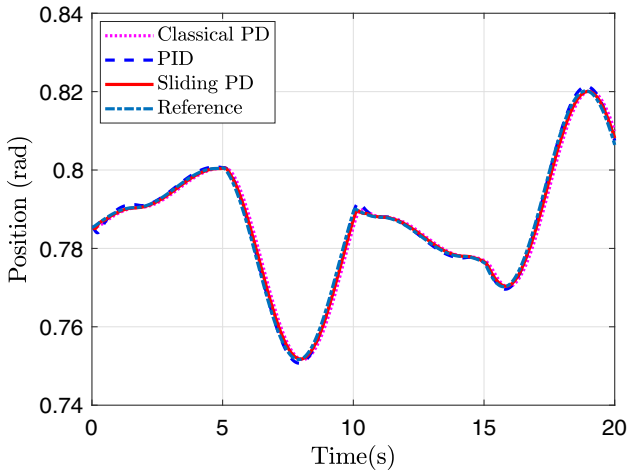

Fig. 9 4-DOF simulation results: joint position q_4

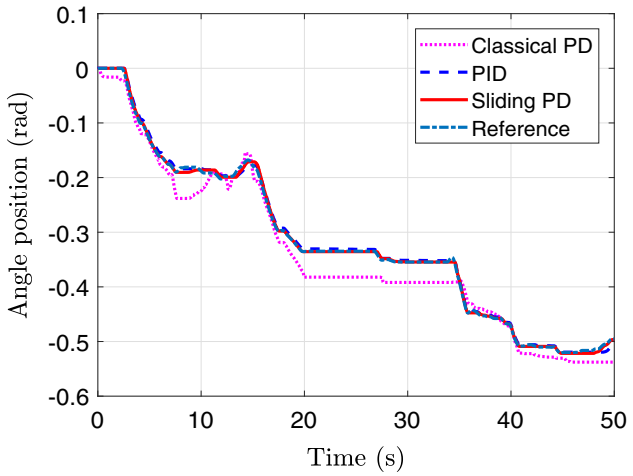

Fig. 10 4-DOF experiment results: joint position q_4

Table 3 2-DOF robot: task space control gains

Gain	Classical PD	PID	Sliding PD
K_p	diag$\{50 \times 10^3\}$	–	–
K_v	diag$\{100\}$	–	–
Λ	–	diag$\{90\}$	
K_i, k_1, k_2	–	diag$\{0.15\}$	
K_s	–	diag$\{0.9\}$	

Then we test the task space controllers. We do the same experiments as in joint space, but instead it is used the angular velocity component of the analytic Jacobian. The proposed control gains for the 2-DOF robot are given in Table 3.

The admittance model is the same as in joint space, with $i = 1, 2, 3$. The experiment results are given in Fig. 11.

For the 4-DOF exoskeleton robot the proposed control gains are given in Table 4.

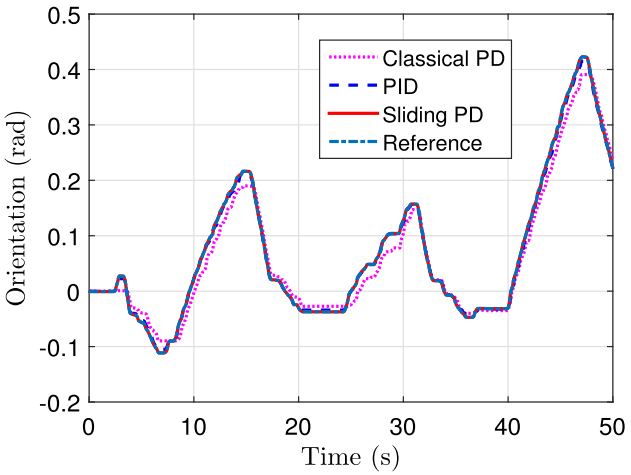

Fig. 11 2-DOF robot experiment: task space results: β Orientation tracking

Table 4 4-DOF exoskeleton robot: task space control gains

Gain	Classical PD	PID	Sliding PD
K_p	diag$\{4 \times 10^3\}$	–	–
K_v	diag$\{10\}$	–	–
Λ	–	diag$\{90\}$	
K_i, k_1, k_2	–	diag$\{1.5\}$	diag$\{0.5\}$
K_s	–	diag$\{3\}$	diag$\{2\}$

The admittance model is the same as in joint space. We use the second method to separate the angular velocity component of the analytical Jacobian (45) as in (49) to avoid multiple joint solutions. The Jacobian approximation is:

$$\hat{J}_{a\omega_1} = \begin{bmatrix} 0 & 0 & -1 & 0 \\ 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \quad \hat{J}_{a\omega_2} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

The comparison results of the worst case scenario are given in Fig. 12.

4.1 Discussion

The results obtained verify our approach. In joint space, when the dynamics of the robot are known and there are no disturbances, the classical PD control can guarantee the tracking of the position, see Figs. 7 and 9. When the dynamics of the robot are unknown and there are disturbances, the classical PD control loses precision, see Figs. 8 and 10, because the robot links has gravitational torques effect.

For the 2-DOF robot, both in the joint space and in the task space, the human operator apply a torque at Y direction, see Figs. 8 and 11. The desired positions are generated from the admittance models. In the joint space, we do not require a

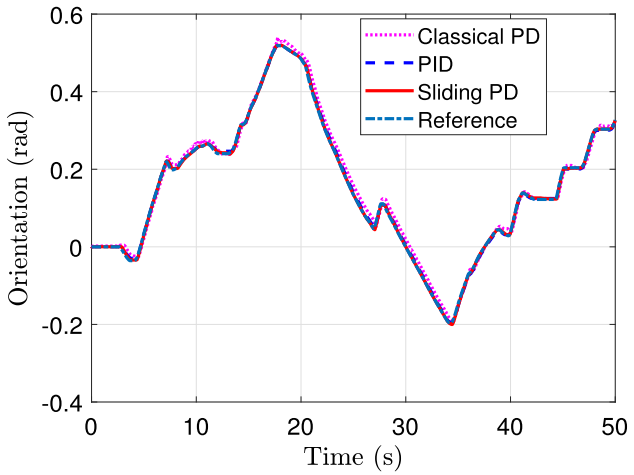

Fig. 12 4-DOF robot experiment: task space results: β Orientation tracking

force component, since the orientation is completely decoupled. In the task space we do not require any other method, since both DOFs are decoupled.

For the 4-DOF robot, a force component is used in joint space to estimate the movement of the robot and obtain the mapping of the force/torque to the joint angles, see Fig. 10. In task space, the Jacobian approach is used to avoid multiple joint solutions, see Fig. 12.

The adaptive and sliding mode PD controllers present good performances for the control task. The adaptive PD control shows good and smooth responses, however, it requires a model structure that complicates the design of the controller when the robot presents many DOFs. On the other hand, the sliding mode PD is a model-free controller that shows a good response and is robust against disturbances, but presents the chattering problem, which is not reliable for the human–robot cooperation task. However, the second order sliding mode (53) overcomes this problem. The PID control is also a model-free controller whose performance is good, and its integral gain must be adjusted carefully to avoid transient performance problems.

4.2 Limitations and Scope

The current approach is reliable for cooperative tasks between human and robot, where the condition (7) is satisfied in such a way that the simplified orientation model can be applied. When the robot has 7 or more DOF, our approach can not be applied directly because we have a redundant robot with more DOF than the Cartesian degrees. This type of robots requires another methodology for the cooperation between humans and robots that is also the subject of our future work.

When the DOF is equal to or less than 6, our proposal can greatly facilitate the control part of a human–robot cooperation task and helps to focus only on the generation of new approaches, such as in [1,21], where the authors proposed some interesting applications with the human is in contact with the robot and requires a careful design of the admittance model and the use of inverse kinematics or the Jacobian matrix [28,29]. This will be investigated further.

5 Conclusions

In this paper, new admittance controllers are presented in joint space and task space. The controllers only use the orientation components. They do not need the inverse kinematics and the Jacobian of the robot. These controllers also use sliding mode and adaptive compensations to improve tracking accuracy. The stability of the controllers is demonstrated through the analysis of Lyapunov. The proposed controllers are verified using a 2-DOF pan and tilt robot and a 4-DOF exoskeleton. Comparisons are made between classic controllers and the proposed methods.

Compliance with Ethical Standards

Conflict of interest The authors declare that they have no conflict of interest.

References

1. Dimeas F, Aspragathos N (2016) Online stability in human–robot cooperations with admittance control. IEEE Trans Haptics 9(2):267–278
2. Ficuciello F, Villani L, Siciliano B (2015) Variable impedance control of redundant manipulators for intuitive human–robot physical interaction. IEEE Trans Robot 31(4):850–863
3. Bonitz RG, Hsia TC (1996) Internal force-based impedance control for cooperating manipulators. IEEE Trans Robot Autom 12(1):78–89
4. Abdossalami A, Sirouspour S (2008) Adaptive control of haptic interaction with impedance and admittance type virtual environments. In: Symposium on haptic interfaces for virtual environments and teleoperator systems, pp 145–152
5. Kazerooni H, Herm MG (1994) The dynamics and control of a haptic interface device. IEEE Trans Robot Autom 10(4):453–464
6. Garrido J (2015) Aprendizaje por demostración en el espacio articular para el seguimiento de trayectorias aplicado en un exoesqueleto de 4 grados de libertad. Centro de Investigación y Estudios Avanzados del Instituto Politécnico Nacional, México
7. Dohring M, Newman W (2003) The passivity of natural admittance control implementations. In: IEEE international conference on robotics and automation, pp 371–376
8. Yu W, Rosen J, Li X (2011) PID admittance control for an upper limb exoskeleton. In: American control conference, pp 1124–1129
9. Hogan N (1985) Impedance control: an approach to manipulation. J Dyn Syst Measurement Control 107:1–24
10. Yu W, Carmona R, Li X (2013) Neural PID admittance control of a robot. In: American control conference, pp 4963–4968
11. Kang SH, Jin M, Chang PH (2009) A solution to the accuracy/robustness dilemma in impedance control. IEEE ASME Trans Mechatron 14:182–194

12. Chih M, Huang AC (2004) Adpative impedance control of robot manipulators based on function approximation technique, robotica. Cambridge University Press, Cambridge, pp 395–403
13. Lu WS, Meng QH (1991) Impedance control with adaptation for robotic manipulators. IEEE Trans Robot Autom 7(3):408–415
14. Kelly R, Carelli R, Amestegui M, Ortega R (1989) On adaptive impedance control of robots manipulators. IEEE Robot Autom 1:572–577
15. Tee KP, Yan R, Li H (2010) Adaptive admittance control of a robot manipulator under task space constraint. In: IEEE international conference on robotics and automation, pp 5181–5186
16. Singh SK, Popa DO (1995) An analysis of some fundamental problems in adaptive control of force and impedance behavior: theory and experiments. IEEE Trans Robot Autom 11(6):912–921
17. Ferreti G, Magnani GA, Rocco P (2004) Impedance control for elastic joints industrial manipulators. IEEE Trans Robot Autom 20(3):488–498
18. Irawan A, Moktadir M, Tan YY (2015) PD-FLC with admittance control for hexapod robot's leg positioning on seabed. In: IEEE American control conference
19. Kiguchi K, Tanaka T, Fukuda T (2004) Nuero-fuzzy control of a robotic exoskeleton with EMG signals. IEEE Trans Fuzzy Syst 12(4):481–490
20. Mohammadi H, Richter H (2015) Robust tracking/impedance control: application to prosthetics. In: American control conference, pp 2673–2678
21. Yu W, Rosen J (2010) A novel linear PID controller for an upper limb exoskeleton. In: 49th IEEE conference on decision and control, pp 3548–3553
22. Tufail M, de Silva CW (2014) Impedance control schemes for bilateral teleoperation. In: International conference on computer science and education, pp 44–49
23. Perrusquía A, Yu W, Soria A, Lozano R (2017) Stable admittance control without inverse kinematics. In: 20th IFAC world congress (IFAC2017), Toulose
24. Ramírez D, Arturo O, Parra Vega V, Díaz Montiel MG, Pozas Cardenas MJ, Hernández Gómez RA (2008) Cartesian sliding PD control of robots manipulators for tracking in finite time: theory and experiments. In: DAAAM international scientific book, chapter 23, pp 257–272
25. Kelly R, Santibáñez V (2003) Control de Movimiento de Robots Manipuladores. Pearson Prentice Hall, Upper Saddle River
26. Spong MW, Hutchinson S, Vidyasagar M (2004) Robot dynamics and control. Wiley, Eglinton
27. Perrusquía A, Yu W (2019) Task space human–robot interaction using angular velocity Jacobian. In: 2019 international symposium on medical robotics (ISMR)
28. Roy S, Edan Y (2018) Investigating joint-action in short-cycle repetitive handover tasks: the role of giver versus receiver and its implications for human–robot collaborative system design. Int J Social Robot. https://doi.org/10.1007/s12369-017-0424-9
29. Someshwar R, Kerner Y (2013) Optimization of waiting time in HR coordination. In: 2013 IEEE international conference on systems, man, and cybernetics (SMC13), pp 1918–1923

Wen Yu received the B.S. degree in automatic control from Tsinghua University, Beijing, China in 1990 and the M.S. and Ph.D. degrees, both in Electrical Engineering, from Northeastern University, Shenyang, China, in 1992 and 1995, respectively. From 1995 to 1996, he served as a Lecturer in the Department of Automatic Control at Northeastern University, Shenyang, China. Since 1996, he has been with CINVESTAV-IPN (National Polytechnic Institute), Mexico City, Mexico, where he is currently a professor and the chair of the Automatic Control Department. From 2002 to 2003, he held research positions with the Mexican Institute of Petroleum. He was a Senior Visiting Research Fellow with Queen's University Belfast, Belfast, U.K., from 2006 to 2007, and a Visiting Associate Professor with the University of California, Santa Cruz, from 2009 to 2010. He also holds a visiting professorship at Northeastern University in China from 2006. Dr.Wen Yu serves as associate editors of *IEEE Transactions on Cybernetics, Neurocomputing, and Journal of Intelligent and Fuzzy Systems*. He is a member of the Mexican Academy of Sciences.

Adolfo Perrusquía received the B.S. degree in mechatronic engineering from the Interdisciplinary Professional Unit on Engineering and Advanced Technologies of the National Polytechnic Institute (UPIITA-IPN) in 2014, and the M.S. degree in Automatic control from the Automatic Control Department at the Center for Research and Advanced Studies of the National Polytechnic Institute (CINVESTAV-IPN), where he is currently working toward the Ph.D. degree in Automatic Control. His main research of interest focused on robotics, machine learning, mechanisms, nonlinear control, system modeling and system identification.