Robotica (2017) volume 35, pp. 1939–1957. © Cambridge University Press 2015
doi:10.1017/S0263574715000120

A 3-PRS parallel manipulator for ankle rehabilitation: towards a low-cost robotic rehabilitation

Marina Vallés†, José Cazalilla†, Ángel Valera†*, Vicente Mata‡, Álvaro Page§ and Miguel Díaz-Rodríguez¶

†Departamento de Ingeniería de Sistemas y Automática, Universidad Politécnica de Valencia, Valencia 46022, Spain (e-mail: jcazalilla@ai2.upv.es, mvalles@ ai2.upv.es).
‡Centro de Investigación de Tecnología de Vehículos, Universidad Politécnica de Valencia, Valencia 46022, Spain (e-mail: vmata@mcm.upv.es).
§Departamento de Física Aplicada, Universidad Politécnica de Valencia, Valencia 46022, Spain (e-mail: alvaro.page@ibv.upv.es).
¶Departamento de Tecnología y Diseño, Facultad de Ingeniería, Universidad de los Andes, Mérida 5101, Venezuela (e-mail: dmiguel@ula.ve).

(Accepted February 10, 2015. First published online: March 13, 2015)

SUMMARY

This paper presents the design, kinematics, dynamics and control of a low-cost parallel rehabilitation robot developed at the Universitat Politècnica de Valencia. Several position and force controllers have been tested to ensure accurate tracking performances. An orthopedic boot, equipped with a force sensor, has been placed over the platform of the parallel robot to perform exercises for injured ankles. Passive, active-assistive and active-resistive exercises have been implemented to train dorsi/plantar flexion, inversion and eversion ankle movements. In order to implement the controllers, the component-based middleware Orocos has been used with the advantage over other solutions that the whole scheme control can be implemented modularly. These modules are independent and can be configured and reconfigured in both configuration and runtime. This means that no specific knowledge is needed by medical staff, for example, to carry out rehabilitation exercises using this low-cost parallel robot. The integration between Orocos and ROS, with a CAD model displaying the actual position of the rehabilitation robot in real time, makes it possible to develop a teleoperation application. In addition, a teleoperated rehabilitation exercise can be performed by a specialist using a Wiimote (or any other Bluetooth device).

KEYWORDS: Parallel robots; Robot control; Force control; Motion control; Rehabilitation robotics; Control engineering computing.

1. Introduction

The development of mechatronic devices for applying forces or controlling human motions is not new in biomechanics. Indeed, there are several precedents such as in surgery, rehabilitation and, to a lesser extent, in functional assessment and trials for diagnostic support areas.[1,2] In the field of rehabilitation, the idea is to develop a device that mimics the work done by the patient along with the physiotherapist during the rehabilitation session.

The quality of healthcare and the clinical rehabilitation process improve when using robotic devices to assist physiotherapists, because it can aid by developing evidence-based therapy (deliver the optimal therapy to a particular patient by monitoring his or her biomedical variables). In addition, a

* Corresponding author. E-mail: giuprog@ai2.upv.es

robotic device can increase the productivity of therapists. In this sense, robotic rehabilitation systems allow patients to perform a wide range of self-administered tasks from passive repetitive actions to functional activities and from assistive tasks to those providing opposition. These systems allow patients to train repetitively and intensively and provide physiotherapists with tools that allow them to treat patients with minimal supervision.

Recent rehabilitation devices have been reviewed in.[3,4] They include a broad range of application such as systems for gait training,[5–7] modified isokinetic tables,[8], systems for rehabilitation of the upper limb (revised in[9]) or active orthoses.[10–12]

In the particular case of the rehabilitation of the lower limb, devices have been developed to generate ankle motions, usually intended for neurological rehabilitation,[13–15] although other applications are intended for ankle sprains.[16] The characteristics of the exercises to be performed in each case are very different, which means that the robot can be configured to suit different needs.[17]

On the other hand, in order to reproduce these movements, a great effort and a very precise control of movements are needed. This control should cover both position and effort aspects when performing each type of exercise. These features make parallel robots more advantageous than other solutions.[1]

The first device proposed for ankle rehabilitation is the Rutgers Ankle.[18] The device is a six Degree-of-Freedom (6-DOF) Parallel Robot consisting of a mobile platform and a fixed based connected by several open kinematic chains. The 6-DOF allows the ankle joint, which is called the ankle joint complex, to move within the range of the ankle motion. However, ankle rehabilitation does not necessarily require 6-DOF. For this reason, parallel robots with 3-DOF and 4-DOF are proposed in.[16] In addition, the range of motion may vary depending on the patient's ankle, which is why a reconfigurable device with respect to the range of motion of each patient's ankle is proposed in.[17] In the references presented so far, the ankle joint complex is modeled as a spherical joint. Recently, the 2-DOF ankle model was proposed based on the functional aspects of the ankle kinematic model. This ankle model was then used to develop a tripod parallel robot.[19]

In addition to parallel robot-based devices, wearable 3-DOF and 4-DOF parallel robots are proposed in[14] and.[20] In these two papers, the mobile platform is linked to the foot while the fixed platform is attached to the lower extremity. Due to the fact that parallel robots have a small workspace and suffer from singular configurations, robot configurations with actuation redundancy have been proposed.[21] However, actuation redundancy has the disadvantage of making them more expensive.

More recent examples of parallel robots used in ankle rehabilitation are the one proposed in,[13] where selection and design of the control algorithms are based on the analysis of the rehabilitation protocol taking into account the dynamics of the system and the dynamics of the interaction between the human and the robot. In[22] a robot is proposed for dynamic posturography studies, where multi-axial perturbations are required. An analysis of accuracy, workspace range and dynamic performance on the control of roll, pitch and yaw angles is achieved. In the above case, no force sensor measurement was taken into account.

In addition to finding a suitable mechanical solution, it is very important to provide adaptability in terms of the type of exercises that can be performed. This involves not only an appropriate kinematic and dynamic design but also developing control systems that are capable of monitoring both movements and forces and integrating the whole system into an easily configurable interface for medical staff.

In this project, we propose a new prototype of parallel robot for ankle rehabilitation with a high level of versatility: patient adaptation, different types of exercises and ease of use for medical personnel.

Based on the design specifications, a 3-DOF $\underline{\text{P}}$RS parallel robot was conceptualized (see Fig. 1). A platform type was developed instead of a wearable robot. The platform type allows to account for the weight of the patient, so proprioceptive exercise can be performed. The prototype has two rotational DOF; thus, the main rehabilitation exercise: Dorsi/Plantar flexion exercise and Eversion/Inversion can be performed. In addition, the robot has one translational DOF that accounts for the height of the patient while sitting on a chair, making it adaptable in a vertical direction. A force sensor has been added to allow a feed forward signal to determine the strength achieved by the patient. This makes it possible to implement different controllers that will be used depending on the patient activity required (active or passive motions). Another major advantage of this project is how the controllers are implemented. The software design technique is based on modular programming. The functionalities of the program are separated into independent and interchangeable modules. The controller's modules can be loaded and executed depending on the rehabilitation exercises required.

https://doi.org/10.1017/S0263574715000120 Published online by Cambridge University Press

This allows the functionality of the robot to be changed dynamically. In this way, the physiotherapist may adapt the therapy based on how the patient is recovering. Furthermore, a remote control module has been put forward, meaning that a specialist can use a WiiMote (or any other Bluetooth device with accelerometers) to perform an "e-rehabilitation".

The paper is organized as follows: the parallel robot design is shown in Section 2. Section 3 deals with the model-based position and force control schemes. Section 4 presents the robot hardware and software architecture. Experiments and results with the ankle rehabilitation robot are shown in Section 5. Finally, conclusions are presented in Section 6.

2. The 3-DOF Parallel Manipulator

2.1. *Physical description of the low-cost PM*

As mentioned before, a 3-DOF spatial PM was used to address the design problem. The robot consists of three kinematic chains; each chain has a $\underline{\text{P}}$RS configuration (P, R, and S stand for prismatic, revolute, and spherical joint, respectively). The underlined format ($\underline{\text{P}}$) indicates the actuated joint. The choice of the $\underline{\text{P}}$RS configuration was guided by the need to develop a low-cost robot with 2-DOF of angular rotation in two axes (Dorsi/Plantar Flexion and Eversion/Inversion) and 1-DOF of translation motion (height). In,[23] a complete description of the mechatronic development process of the PM is presented.

The physical system consists of three legs connecting the moving platform to the base. Each leg is driven by a direct drive ball screw (prismatic joints). A coupler bar is connected to the ball screw with a revolute joint. A spherical joint connects the upper part of the coupler to the moving platform. The lower part of the ball screws are perpendicularly attached to the platform's base. The ball screws are distributed on the base in an equilateral triangular configuration. The ball screw transforms the rotational movement of the motor into linear motion.

The motors in each leg are brushless DC servomotors equipped with power amplifiers. The actuators are Aerotech BMS465 AH brushless servomotors. The specifications of these motors are a stall torque of 2.86 N.m and a peak torque of 11.43 N.m, both continuous. The lead of the ball screw is of 20 mm. Aerotech BA10 power amplifiers operate the motors. The control system was developed on an industrial PC.

2.2. *Kinematic and dynamic model*

For modeling purposes, mobile reference systems have been attached to the robot links using Denavit–Hartenberg's notation, a detailed explanation can be found in ref. [23]. Figure 2 shows a kinematic diagram of the robot. Nine generalized coordinates are used to model the robot (q_i, where $i = 1 \ldots 9$). The active coordinates q_1, q_6, and q_8 are associated with the actuated prismatic joints ($\underline{P}$). The passive coordinates q_2, q_7, and q_9 are associated with the revolute joints (R), and coordinates q_3, q_4, and q_5 correspond to only one of the spherical joints (S, located at P_1 in Fig. 2). The spherical joint has been modeled by means of three mutually perpendicular rotational joints.

The forward position problem is solved using the geometric approach. In Fig. 2, based on the rigid body assumption, the length between points P_i and P_j is constant and equal to l_m; thus, the following equations hold

$$\Psi_1\,(q_1, q_2, q_6, q_7) = \|(\vec{r}_{A_1B_1} + \vec{r}_{B_1P_1}) - (\vec{r}_{A_1A_2} + \vec{r}_{A_2B_2} + \vec{r}_{B_2P_2})\| - l_m = 0, \tag{1}$$

$$\Psi_2\,(q_1, q_2, q_8, q_9) = \|(\vec{r}_{A_1B_1} + \vec{r}_{B_1P_1}) - (\vec{r}_{A_1A_3} + \vec{r}_{A_3B_3} + \vec{r}_{B_3P_3})\| - l_m = 0, \tag{2}$$

$$\Psi_3\,(q_6, q_7, q_8, q_9) = \|(\vec{r}_{A_1A_3} + \vec{r}_{A_3B_3} + \vec{r}_{B_3P_3}) - (\vec{r}_{A_1A_2} + \vec{r}_{A_2B_2} + \vec{r}_{B_2P_2})\| - l_m = 0, \tag{3}$$

where $\vec{r}_{ij}$ is a position vector from the ith to the jth point.

In the forward position problem, the position of the actuators (q_1, q_6, and q_8) is given. Thus, the values of q_2, q_7, and q_9 are found by solving the nonlinear system (1)–(3). The Newton–Raphson (N–R) numerical method is chosen to solve this nonlinear system. The method converges quite quickly when the initial guess is close to the desired solution.[24] Once those coordinates are obtained, the location of points P_i can be found. With the coordinates of P_i the rotation matrix between the mobile platform and the fixed base is easily obtained. The remaining generalized coordinates (q_3, q_4, and q_5) are found in a straightforward manner from the rotation matrix.

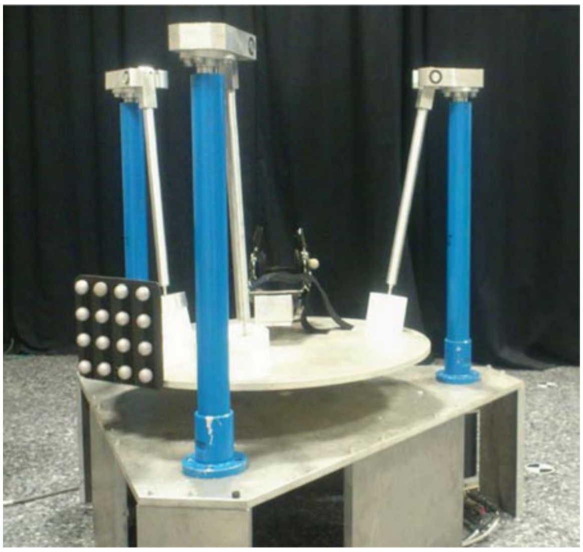

Fig. 1. The 3-PRS low-cost parallel manipulator.

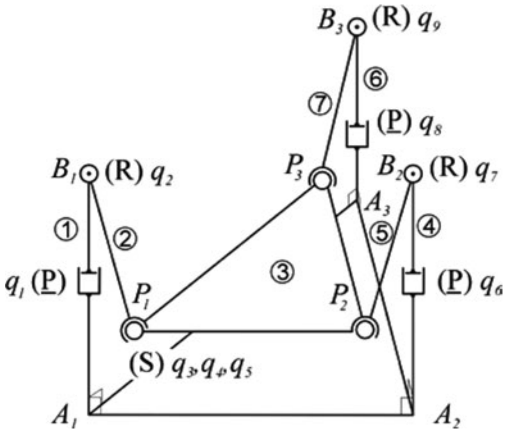

Fig. 2. Kinematic diagram of the 3-P̲RS parallel manipulator, type of joints and generalized coordinates.

The inverse position analysis consists of finding the actuated generalized coordinates given the roll, the pitch angles, and the heave of the reference system attached to the mobile platform. From these values, the coordinates of points P_i can be obtained, so that, following,[25]

$$\vec{r}_{A_1P_1} - q_1 \cdot \vec{u}_{A_1B_1} = l_m \cdot \vec{u}_{B_1P_1}. \tag{4}$$

In Eq. (4), $\vec{u}_{ij}$ is a unit vector in the direction of points i and j. Analytical expression for the generalized coordinate q_1 is obtained by squaring both sides of Eq. (4). A similar procedure can be applied for the other two limbs.

Moreover, as is well known, the relationship between the velocities of the joint coordinates and the position and orientation of the end effector of the robot can be obtained through the Jacobian matrix.

To obtain this matrix, in Fig. 2 the following vectorial expression holds

$$\vec{p} + \vec{b}_i = \vec{a}_i + q_1 \cdot \vec{u}_{A_iB_i} + l_r \cdot \vec{u}_{B_iP_i}. \tag{5}$$

where l_r is the length of the coupler link, $\vec{p}$ is the position vector of the end effector, $\vec{b}_i$ is the position vector between the end effector and the spherical joint i. Finally, $\vec{a}_i$ define the position of the prismatic joint with regard to the global reference system. Deriving and developing Eq. (5), the following expression is obtained

$$\vec{u}_{B_iP_i}\cdot\vec{v}_p+\vec{u}_{B_iP_i}\cdot\left(\vec{\omega}_p\times\vec{b}_i\right)=\vec{u}_{B_iP_i}\cdot\dot{q}_i\cdot\vec{u}_{A_iB_i}. \tag{6}$$

Applying the above equation to each of the robot's legs or chains

$$\begin{bmatrix}\vec{u}_{B_1P_1}\cdot\vec{u}_{A_1B_1} & 0 & 0\\ 0 & \vec{u}_{B_2P_2}\cdot\vec{u}_{A_2B_2} & 0\\ 0 & 0 & \vec{u}_{B_3P_3}\cdot\vec{u}_{A_3B_3}\end{bmatrix}\cdot\begin{bmatrix}\dot{q}_1\\ \dot{q}_6\\ \dot{q}_8\end{bmatrix}=J_q\cdot\vec{\dot{q}}, \tag{7}$$

$$\begin{bmatrix}\vec{u}^{\,T}_{B_1P_1} & \left(\vec{b}_1\times\vec{u}_{B_1P_1}\right)^T\\ \vec{u}^{\,T}_{B_2P_2} & \left(\vec{b}_2\times\vec{u}_{B_2P_2}\right)^T\\ \vec{u}^{\,T}_{B_3P_3} & \left(\vec{b}_3\times\vec{u}_{B_3P_3}\right)^T\end{bmatrix}\cdot\begin{bmatrix}\dot{x}\\ \dot{y}\\ \dot{z}\\ \omega_x\\ \omega_y\\ \omega_z\end{bmatrix}=J_x\cdot\vec{\dot{X}}. \tag{8}$$

In Eq. (8) x-y-z stand for components of the end-effector position, $\vec{p}$, and $\omega_x-\omega_y-\omega_z$ are the components of the angular velocity of the platform.

However not all variables of vector $\vec{X}$ are independent. The relationship between the generalized and the end-effector coordinates is given by

$$\vec{\dot{q}}=\mathrm{J}_{\mathrm{q}}^{-1}\cdot\mathrm{J}_{\mathrm{x}}\cdot\begin{bmatrix}\mathrm{I} & 0\\ 0 & \mathrm{G}\end{bmatrix}\cdot\begin{bmatrix}\frac{\delta\mathrm{x}}{\delta\mathrm{z}} & \frac{\delta\mathrm{x}}{\delta\gamma} & \frac{\delta\mathrm{x}}{\delta\beta}\\ \frac{\delta\mathrm{y}}{\delta\mathrm{z}} & \frac{\delta\mathrm{y}}{\delta\gamma} & \frac{\delta\mathrm{y}}{\delta\beta}\\ 1 & 0 & 0\\ 0 & 1 & 0\\ 0 & 0 & 1\\ \frac{\delta\alpha}{\delta\mathrm{z}} & \frac{\delta\alpha}{\delta\gamma} & \frac{\delta\alpha}{\delta\beta}\end{bmatrix}\cdot\begin{bmatrix}\dot{\mathrm{z}}\\ \dot{\gamma}\\ \dot{\beta}\end{bmatrix}, \tag{9}$$

where I is the 3×3 Identity matrix and G is a matrix relating the generalized coordinates and the angular velocity of the platform.

The development of a rehabilitation robot needs to implement and test dynamic robot control schemes. The motion dynamic controller requires the equation of motion to be described as follows

$$M\left(\vec{q},\vec{\Phi}\right)\cdot\vec{\ddot{q}}+\vec{C}\left(\vec{q},\vec{\dot{q}},\vec{\Phi}\right)\cdot\vec{\dot{q}}+\vec{G}\left(\vec{q},\vec{\Phi}\right)=\vec{\tau}. \tag{10}$$

In Eq. (10), M stands for the system mass matrix, $\vec{C}$ is the matrix grouping the centrifugal and Coriolis terms, $\vec{G}$ is the vector corresponding to gravitational terms and $\vec{\tau}$ is the vector of generalized forces. It is worth noting that Eq. (10) is valid only when the system is modeled through a set of independent generalized coordinates. In this paper, a coordinate partitioning procedure has been considered in order to allow the system to be modeled by a set of independent generalized coordinates grouped together in the vector $\vec{q}=\left[q_1\,q_6\,q_8\right]^T$.

Equation (10) can be rewritten (see ref. 26, 27) as follows

$$\begin{bmatrix}K_{rb} & K_r & K_f\end{bmatrix}\cdot\begin{bmatrix}\vec{\Phi}_{rb}\\ \vec{\Phi}_r\\ \vec{\Phi}_f\end{bmatrix}=\vec{\tau}. \tag{11}$$

In Eq. (11), $\vec{\Phi}_{rb}$, $\vec{\Phi}_r$, $\vec{\Phi}_f$ are the vectors grouping the rigid body, rotor and friction parameters. K_i is the part of the regressor matrix determining the linear relationship between the corresponding parameters (rigid body, rotor, and friction) and the generalized forces.

From Eq. (11), a set of base parameters corresponding to the complete base parameter model can be obtained. However, not even those parameters can always be identified properly. Thus, the reduced

https://doi.org/10.1017/S0263574715000120 Published online by Cambridge University Press

model containing only the relevant parameters will be obtained through a process that considers the robot's leg symmetries, the influence on the dynamic behavior of the robot, the statistical significance of the parameters identified and the physical feasibility of the parameters.[27]

The dynamic terms of the equation of motion for the actual parallel robot based on relevant parameters can be written as follows

$$M\left(\vec{q}\right)\cdot\vec{\ddot{q}} = \begin{bmatrix} J_1 & 0 & 0 \\ 0 & J_2 & 0 \\ 0 & 0 & J_3 \end{bmatrix} \cdot \begin{bmatrix} \ddot{q}_1 \\ \ddot{q}_6 \\ \ddot{q}_8 \end{bmatrix} + \begin{bmatrix} M_{11}(\vec{q}) & M_{12}(\vec{q}) & M_{13}(\vec{q}) \\ M_{21}(\vec{q}) & M_{22}(\vec{q}) & M_{23}(\vec{q}) \\ M_{31}(\vec{q}) & M_{32}(\vec{q}) & M_{33}(\vec{q}) \end{bmatrix} \cdot \begin{bmatrix} \Omega_1 \\ \Omega_2 \\ \Omega_3 \end{bmatrix}, \tag{12}$$

$$C\left(\vec{q},\vec{\dot{q}}\right)\cdot\vec{\dot{q}} = \begin{bmatrix} F_{v_1}\dot{q}_1 + F_{c_1}\mathrm{sign}\left(\dot{q}_1\right) \\ F_{v_2}\dot{q}_6 + F_{c_2}\mathrm{sign}\left(\dot{q}_6\right) \\ F_{v_3}\dot{q}_8 + F_{c_3}\mathrm{sign}\left(\dot{q}_8\right) \end{bmatrix} + \begin{bmatrix} C_{11}(\vec{q},\vec{\dot{q}})\, C_{12}(\vec{q},\vec{\dot{q}})\, C_{13}(\vec{q},\vec{\dot{q}}) \\ C_{21}(\vec{q},\vec{\dot{q}})\, C_{22}(\vec{q},\vec{\dot{q}})\, C_{23}(\vec{q},\vec{\dot{q}}) \\ C_{31}(\vec{q},\vec{\dot{q}})\, C_{32}(\vec{q},\vec{\dot{q}})\, C_{33}(\vec{q},\vec{\dot{q}}) \end{bmatrix} \cdot \begin{bmatrix} \Omega_1 \\ \Omega_2 \\ \Omega_3 \end{bmatrix}, \tag{13}$$

$$G\left(\vec{q}\right) = g \cdot \begin{bmatrix} G_{11}(\vec{q})\, G_{12}(\vec{q})\, G_{13}(\vec{q}) \\ G_{21}(\vec{q})\, G_{22}(\vec{q})\, G_{23}(\vec{q}) \\ G_{31}(\vec{q})\, G_{32}(\vec{q})\, G_{33}(\vec{q}) \end{bmatrix} \cdot \begin{bmatrix} \Omega_1 \\ \Omega_2 \\ \Omega_3 \end{bmatrix}. \tag{14}$$

Where J_i are the actuator relevant parameters, F_{Ci} and F_{Vi} are the friction base parameters, and Ω_i are the rigid body base parameters considered as relevant ones. M_{ij}, C_{ij}, and G_{ij} are the elements of the inertial, Coriolis and gravity matrices that depend on the Ω_i parameters.

3. Robot Position and Force Control Schemes

3.1. Model-based position control schemes

In recent years, the passivity-based approach to robot control has gained a lot of attention. This approach solves the robot control problem by exploiting the robot system's physical structure, and specifically its passivity property. The design philosophy of these controllers is to reshape the system's natural energy in such a way that the tracking control objective is achieved.[28]

For the tracking problem, the kinetic and potential energy must be modified as required in passivity-based controllers. The general expression of the controllers is[23]

$$\tau_c = M(q)\cdot a + C(q,\vec{\dot{q}})\cdot\bar{v}_1 + G(q) - K_p\cdot e - K_d\cdot v_2. \tag{15}$$

Where a, v_1, v_2, and e varies according to the type of controller (see Table I). In all these controllers, the control law has two parts: compensation of robot dynamics and a proportional and derivative controller.

Table I. Passivity-based tracking controllers.

Controller	A	v_1	e	v_2
Paden, Panja[29]	$\ddot{q}_d$	$\dot{q}_d$	$q - q_d$	$\dot{e}$
Slotine, Li[30]	$\ddot{q}_r$	$\dot{q}_r$	0	$\dot{e} + \Lambda_1 e$
Sadegh, Horowitz[31]	$\ddot{q}_r$	$\dot{q}_r$	$q - q_d$	$\dot{e} + \Lambda_1 e$

where $\vec{\dot{q}}_r = \vec{\dot{q}}_d - \Lambda_1\vec{e}$, Λ_1 is a symmetric positive definite matrix and $\vec{q}_\mathrm{d}$ is the desired position.

The first controller is a variation of the classic PD with a gravity compensation regulator. The second is a tracking controller based on the sliding mode theory. In the last case, some modifications have been made to the control law and the energy function. It allows the system's asymptotic stability to be shown using the Lyapunov theory.

https://doi.org/10.1017/S0263574715000120 Published online by Cambridge University Press

3.2. Robot force control

In this work, different types of force control strategies have been developed and used. The first type is based on explicit force control, which consists of following the force reference value using feedback (and feedforward, if necessary). Usually, the explicit control is restricted to a linear control,[32] as is the case of the classic PID force controller in Eq. (16).

$$F = K_p \cdot (F_{ref} - f) + K_I \cdot \int (F_{ref} - f)dt + K_d \cdot \frac{d}{dt}(F_{ref} - f), \tag{16}$$

where F_{ref} and f are the force reference and the force measured, respectively. F is the force control action.

The effect of the three parameters of a PID controller is well known and has been widely addressed in the literature.[33] The integral term assures zero tracking error. The function of the derivative term is to damp the system. Nevertheless, the force control application has some specific problems that may require modifications of the classic PID controller. Firstly, in constrained motion, the dynamics of the system depend on the characteristics of the environment. Basically, this means that the parameters of the controller must be re-tuned for every application. Sometimes the characteristics of the environment are not known in advance. On the other hand, it may be demonstrated that the integral term can make the system unstable. One solution may be to use a force feedforward term instead of the integrator[34]

$$F = F_{ref} + K_p \cdot (F_{ref} - f) + K_d \cdot \frac{d}{dt}(F_{ref} - f). \tag{17}$$

The second type of controller implemented in this work is based on the impedance control proposed in.[35] In this case, the purpose is not to follow a reference value, but to ensure the desired dynamic behavior of the system (which is known as mechanical impedance). Typical specified dynamics correspond to a second order spring-damper system determined by the following expression

$$Z(s) = \frac{F(s)}{v(s)} = \frac{M \cdot s^2 \cdot X + B \cdot s \cdot X + K \cdot X}{s \cdot X} = M \cdot s + B + \frac{K}{s}, \tag{18}$$

where M, B, and K are the mass, damping and stiffness parameters and s the Laplace transform operator. These parameters determine the closed-loop system behavior with respect to contact force F.

The description of the different methods for implementing the impedance control lies outside the scope of this paper, but typically they are a combination of linear feedback and inverse dynamics, both described in.[36] An interesting alternative is proposed in,[37] where sliding control is established.

Figure 3 shows the implementation of the force control architecture developed in this work in more detail. The controller receives the position and velocity references for the robot axes, as well as the force reference that the robot shall apply to the human/environment. The actual force in the robot end effector is measured by the platform-mounted force sensor. The specific force control algorithm is programmed into the "*Force Controller*" block. Finally, the new position and velocity references are computed by means of the Jacobian matrix J of Eq. (9).

The proposed architecture provides several advantages. On the one hand, a very accurate force control can be established. On the other hand, as it is an open, flexible architecture, any force control algorithm can be implemented.

4. Robot Control Architecture

4.1. Robot hardware architecture

In order to implement the control architecture for the parallel robot, an industrial PC has been used. It is based on a high-performance 4U Rackmount industrial system with 7 PCI slots and 7 ISA slots. It has a 2.5 GHz Intel® Pentium® Core 2 Quad/Duo processor and 4 GB SDRAM.

The industrial PC is equipped with 2 Advantech™ data acquisition cards: a PCI-1720 and a PCL-833.The PCI-1720 card has been used for supplying the control actions for each parallel robot

https://doi.org/10.1017/S0263574715000120 Published online by Cambridge University Press

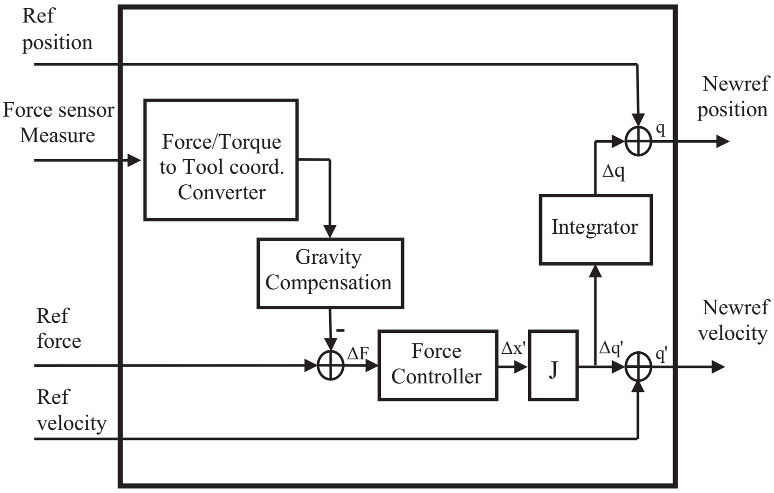

Fig. 3. Structure of the force control.

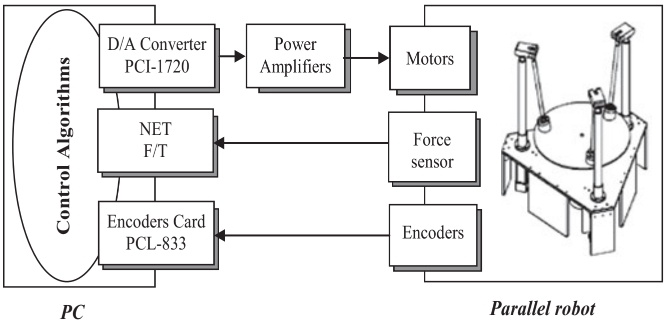

Fig. 4. Robot control architecture.

actuator. It provides four 12-bit isolated digital-to-analog outputs for the Universal PCI 2.2 bus. It has multiple output ranges (0∼5 V, 0∼10 V, ± 5 V, ± 10 V), programmable software and an isolation protection of 2500 VDC between the outputs and the PCI bus.

The PCL-833 card is a 4-axis quadrature encoder and counter add-on card for an ISA bus. The card includes four 32-bit quadruple AB phase encoder counters, an onboard 8-bit timer with a wide range time-based selector and it is optically isolated up to 2500 V.

In order to establish the force control, the robot has been equipped with the Delta SI-330-30 ATI sensor. This is a sensor with 6-DOFcapable of measuring forces and torques in the XYZ axes using a monolithic instrumented transducer. The maximum range of forces is ± 3700 N for X and Y, and ± 10000 N for the Z axis. The maximum range of torque is ± 270 N.m for X and Y, and ± 400 N.m for the Z axis. In order to transmit the signals from the sensor to the control unit (industrial PC in this case) there are three options: data acquisition card, F/T controller or Ethernet communication system. In this work, the last option has been implemented. The NET F/T system provides Ethernet/IP and CAN bus communication interfaces and is compatible with standard Ethernet. This device can be easily connected to any local area network, allowing more than 7000 Hz frequency for measurement of the six components, ensuring real-time communication.

Figure 4 shows the control architecture based on an industrial PC developed for this study.

The programming language used to control the parallel robot is C++. The PC is equipped with the Linux Ubuntu operating system, patched with Xenomai (a real-time kernel). Therefore, real-time characteristics are available.

Because the control architecture is based on an industrial PC, it has two main advantages: first, it is totally open and gives a powerful platform for programming high-level tasks based on the Ubuntu 12.04 operating system. Thus, any controller and/or control technique can be programmed and implemented, such as automatic trajectory generation, control based on external sensing using

https://doi.org/10.1017/S0263574715000120 Published online by Cambridge University Press

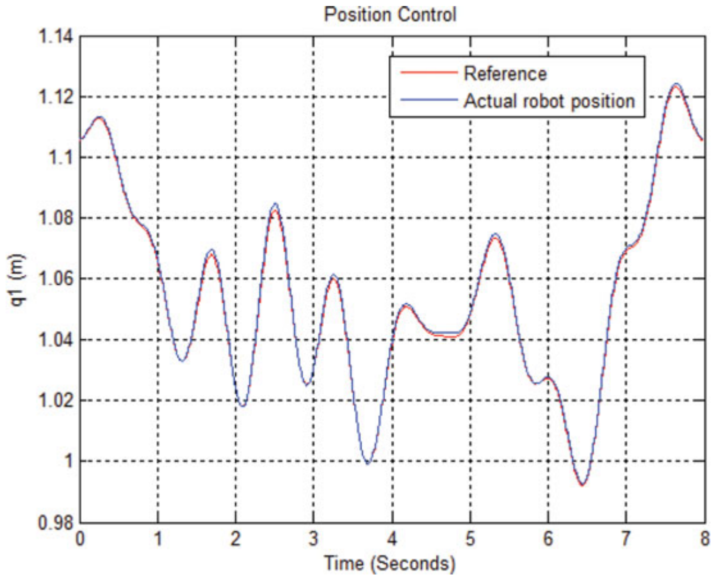

Fig. 5. q1 robot position.

a force sensor or artificial vision, etc. The second advantage is the low cost: the total cost of the hardware (the computer and the industrial data acquisition cards) does not exceed $ 2000. In addition, the operating system and all the development and programming tools are free software, so this does increase the cost of the control system.

4.2. *Robot software architecture*

It is well-known that a very difficult aspect of creating a robotic prototype is the software architecture. In recent years, component-based software development has been increasing, with the aim of achieving an execution in a distributed way, as well as the reusability of the code developed. In order to implement different robot controllers, the real-time middleware Orocos[38] (Open Robot Control Software) has been used. Because the Orocos environment provides component-based software development, it involves the following advantages that have been used in this project:

- Modular design and structure.
- Fully reusable code and modules.
- Modules configurable and reconfigurable during both setup and running time.
- Distributed execution of the modules, improving total execution time.

Due to the modular design and structure, when a number of modules are implemented and a control scheme is required, it is as simple as inserting the necessary modules to configure them, making connections with each other and making them run. Therefore, because the different control schemes have common parts, as several modules have been developed, these modules are reused to implement different controllers.

Note that, although it can be a complicated task at first, the development of component-based software makes the programmer's job easier in the end because if a module works correctly in one particular scheme, it will certainly work correctly in another control scheme. Therefore, besides the advantages discussed above, this approach minimizes the chance of programming errors in the implementation of any of the modules.

Different position and force controllers for ankle rehabilitation have been implemented and tested in the parallel robot using Orocos. Figure 5 shows the reference and the robot q_1 position. The results point to the conclusion that the controller was able to generate very good tracking performance, providing a mean error of −7.61e-4 m. Figure 6 shows the reference and the force applied by the robot for the Z-axis.

Orocos is one of the best motion control frameworks available at the moment, but it presents certain constraints when trying to achieve something other than the control itself. One of the solutions

https://doi.org/10.1017/S0263574715000120 Published online by Cambridge University Press

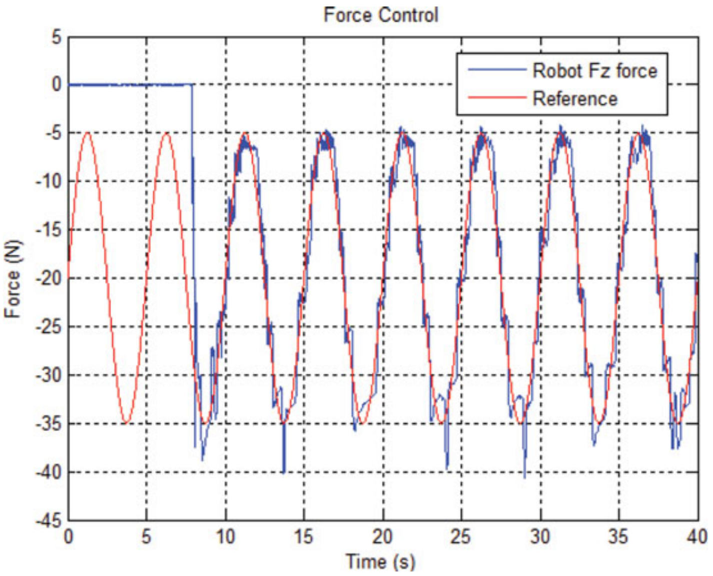

Fig. 6. Robot force.

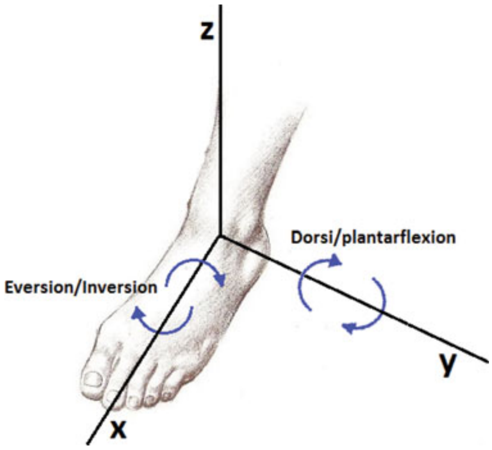

Fig. 7. Ankle movements.

is ROS,[40] that has been designed as a conglomeration of various tools organized in packages. Each package or "stack" may contain libraries, executables or scripts and a manifest which defines the dependencies on other packages and meta information about the package itself.

A ROS package called *rtt_ros_integration* allows for Orocos components to connect to the ROS network. This way they can both publish and subscribe to all the available streamed topics, being both middlewares fully compatible.

What is important is that while ROS has many tools and functionalities that are useful in the development of robotic applications, Orocos provides a solid core for the main control scheme in real time. In other words, ROS and Orocos complement each other, broadening the range of solutions they can offer as standalone platforms.

5. Ankle Rehabilitation Robot

The robot presented in Section 2 is the one used to show how the robot operates in order to carry out a rehabilitation of the lower limbs, in particular, the ankle. As can be seen in Fig. 7, four of the possible movements of the ankle are plantar/dorsiflexion, inversion and eversion, which are represented by gamma and beta.

https://doi.org/10.1017/S0263574715000120 Published online by Cambridge University Press

Table II. Ankle and robot range of motion.

Type of movement	Max. human ankle	Max. parallel robot
Dorsiflexion	20.3°–29.8°	50°
Plantarflexion	37.6°–45.8°	50°
Eversion	15.4°–25.9°	50°
Inversion	22°–36°	50°

Table III. Ankle and force sensor range of torque.

Type of movement	Max. human ankle passive movement	Max. force sensor
Dorsiflexion	–33.1 ± 16.5 N·m	± 270 N·m
Plantarflexion	40.1 ± 9.2 N·m	± 270 N·m
Eversion	–48.1 ± 12.2 N·m	± 270 N·m
Inversion	34.1 ± 14.5 N·m	± 270 N·m

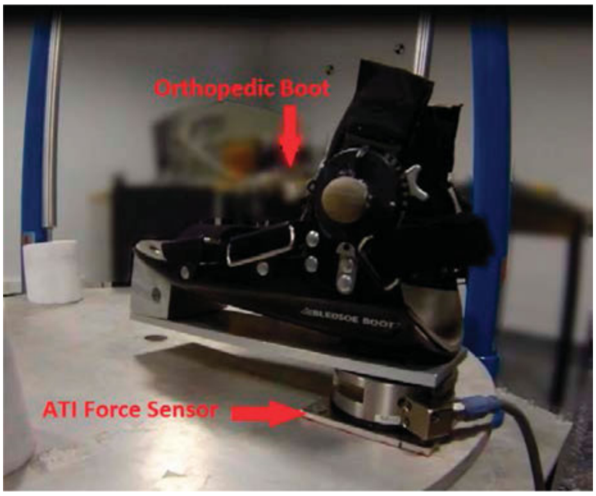

Fig. 8. Orthopedic boot.

On the other hand, several studies have determined the range of motion of the ankle.[41] Obviously, the maximum range is determined by each of the patients,[42] so the exercise must be slightly different (position reference and/or force reference) depending on the patient who is being treated. A patient with a first degree ankle sprain is not the same as another with a third degree ankle sprain. Table II shows the maximum allowable motion of the ankle, both in the X-axis (roll) and the Y-axis (pitch), and the maximum working range of the robot presented above.

Table III shows the maximum torque for a passive movement of a human ankle, as well as the measured range for the force sensor used (described in Section 4.1).

Taking into account the specifications of the motors and the ball screw mentioned in Section 2, forces much greater than 800 N can be supplied for each of the three actuators. Considering the distance (577 mm) between the spherical joints located on the mobile platform, the parallel robot can supply the range of torques required in order to reproduce human ankle movements.

The placement of the boot on the parallel robot platform can be seen in Fig. 8. This boot allows the patient's foot to be attached by Velcro strips. Thus, because of the foot is attached correctly, the rehabilitation exercises can be performed properly. In addition, a force sensor has been placed at the base of the boot to monitor the forces and torques applied. The forces applied to the ankle are monitored using this force sensor, and several exercises to rehabilitate and strengthen injured ankles have been implemented.

The most common injuries are ankle sprains (representing 38% of locomotor system injuries) and these cause stretching or tearing of the ligaments due to a sudden movement in the direction

https://doi.org/10.1017/S0263574715000120 Published online by Cambridge University Press

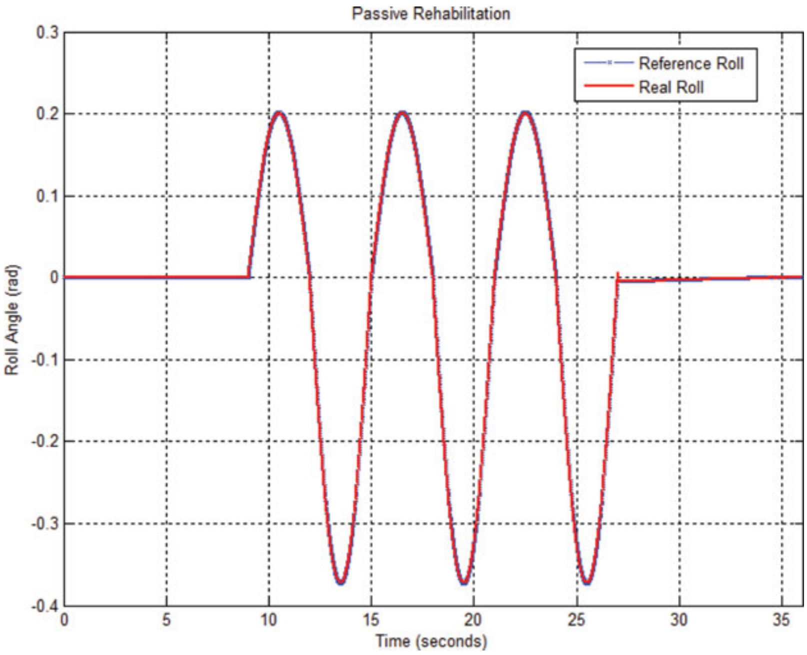

Fig. 9. Passive rehabilitation.

of eversion.[43] In a large percentage of cases, sprains are not treated and rehabilitation is not performed. As a result, between 80% and 90% of sprains can become chronic if the injury is not correctly rehabilitated, leaving ligament instability (which increases over time as more injuries occur). Therefore, proper rehabilitation is necessary.

As for the different exercises that can be performed with this parallel robot, in order to repair or strengthen injured ankles these exercises can be passive or active. Passive exercises are performed without any voluntary movement by the patient, while active exercises are performed with voluntary movement by the patient. In order to show how the robot operates, a series of exercises have been performed with healthy subjects.

5.1. *Passive exercises with the parallel robot*

In passive ankle exercises, the robot is programmed to follow a specific position reference prescribed by a specialist. Thus, using the low-cost parallel robot developed in this project, a number of references have been generated to rehabilitate an injured ankle.

Since the dynamic position control proposed by Paden–Panja has been used (see Table I), the position error is around 0.5 mm, so the movement is very accurate.

This position reference implemented for a passive rehabilitation exercise consists of a sinusoidal signal at a roll frequency of 0.16 Hz, in order to exercise the plantar/dorsiflexion.

The reference and actual position of the roll of the platform can be seen in Fig. 9. The error is imperceptible, so the movement indicated by the specialist is done very accurately.

Furthermore, to follow a reference with high precision, the forces and torques applied to the ankle are being monitored all the time. Figure 10 shows the torque in the X-axis, Y-axis, and Z-axis. Since both plantarflexion and dorsiflexion have been trained in this exercise, the torque in the X–Z-axis is nearly zero, while the Y-axis has significant values. This indicates that the rehabilitation process has been successful.

5.2. *Active-resistive exercises with the parallel robot*

There are several types of active movements, such as strengthening or resistive, in which the patient has to overcome a resistance imposed by the specialist.

Specifically, a resistive application has been proposed in which the aim is to keep the platform of the parallel manipulator in a horizontal position, doing opposed torques to the motion of the platform. A low frequency sinusoidal position reference in roll (and thus strengthens plantarflexion

https://doi.org/10.1017/S0263574715000120 Published online by Cambridge University Press

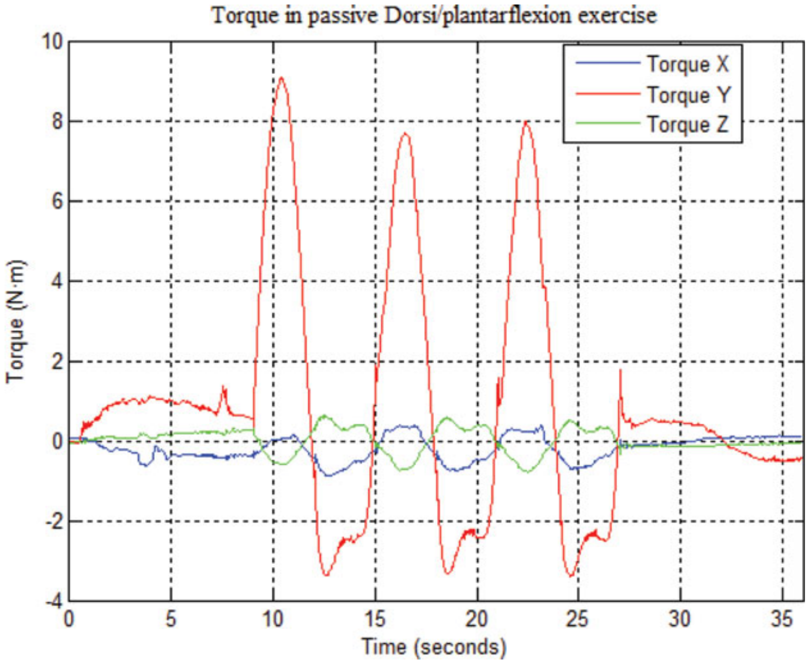

Fig. 10. Torque measured in passive rehabilitation.

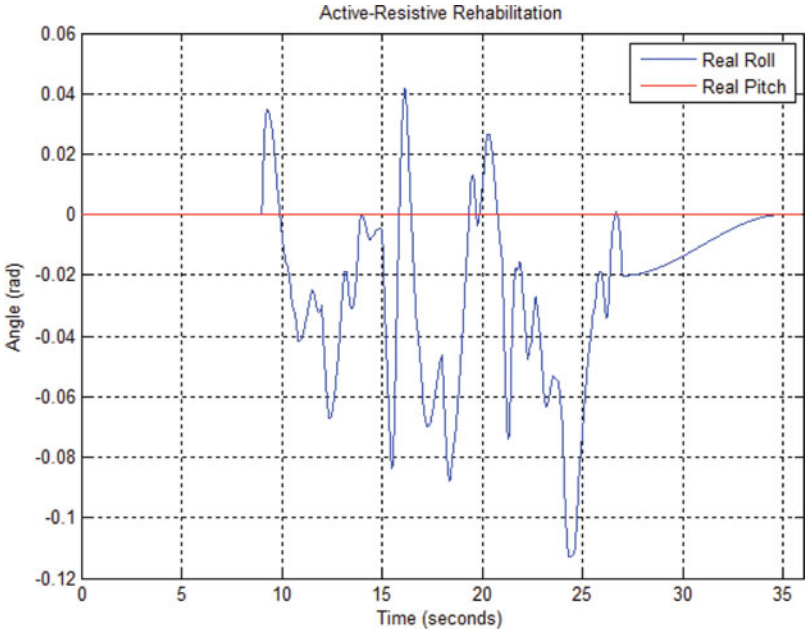

Fig. 11. Active-resistive rehabilitation.

and dorsiflexion) has therefore been implemented. Figures 11 and 12 below show the actual position of the robot (roll and pitch) and the torques measured at the patient's ankle.

As seen in the previous graphs, the patient, by the torques measured in the ankle, is able to account for the sinusoidal reference. Thus, the platform is, generally, kept in a horizontal position (roll = pitch = 0).

5.3. Active-assistive exercises with the parallel robot

The main difference between active-assistive and active-resistive exercises is that in assistive movements the patient is not able to carry out the movement against gravity by him or herself.

https://doi.org/10.1017/S0263574715000120 Published online by Cambridge University Press

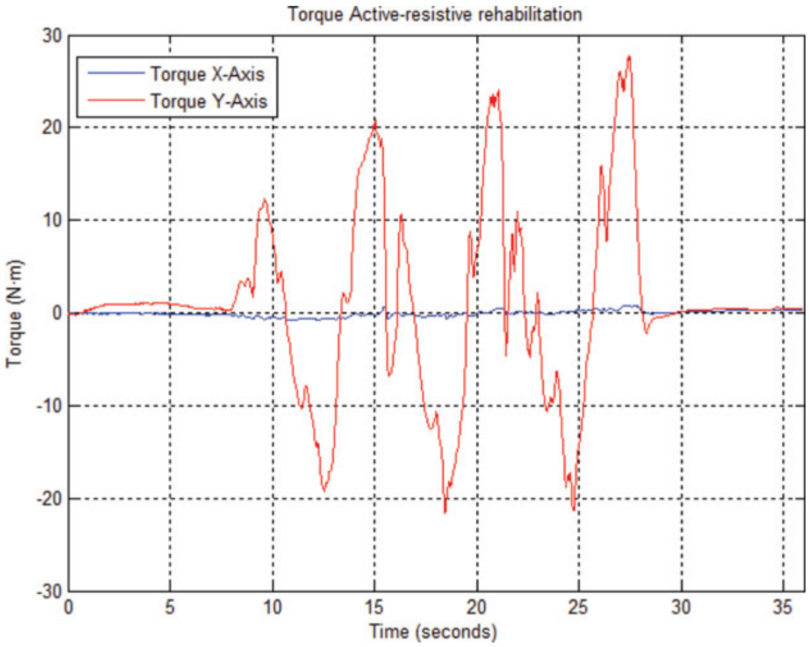

Fig. 12. Forces measured in active resistive rehabilitation.

These types of exercise are usually done at an early stage in the rehabilitation process. For this reason, external help is needed to perform them correctly.

The assistive exercise developed is based on the fact that the platform assists the patient to perform the movement, depending on the torques applied by the injured ankle to the load cell.

Thus,

$$\vec{q}_{Newref} = \vec{q}_{ref} + K_{st} \cdot \vec{T}_{xy}, \tag{19}$$

where $\vec{q}_{ref}$ is the position reference, K_{st} is a stiffness constant, $\vec{T}_{xy}$ is the torque measured in the ankle and $\vec{q}_{Newref}$ is the position reference modified considering these two parameters. The position reference generated supports itself in a horizontal position (Gamma = Beta = 0) and at a constant height (Z = constant) and, depending on the forces read, a new reference position modified is generated dynamically, in order to assist the movement of the injured patient.

As can be seen in Figs. 13 and 14 above, the platform changes its orientation in accordance with the data read by the force sensor. Obviously, depending on the stage of the patient's rehabilitation, movement of the platform will be more or less sensitive to the force measurements made by the ankle.

5.4. Configuration of exercises for each patient using Orocos

As seen in the previous sections, several active and passive rehabilitation exercises have been implemented. One of the main disadvantages in rehabilitation robotics is the difficulty of health personnel learning how the robot works. Therefore, robots in rehabilitation clinics are very expensive and have limited functionality due to the manufacturer's own restrictions.

The low-cost robot presented in this article is controlled using the Orocos component-based middleware. One of the main advantages of Orocos is that the control scheme is developed modularly. Afterwards, once the modules are implemented, they are loaded into the system, configured according to the application the specialist wishes to perform, connected and executed.

This way, medical personnel do not need to have prior training to control the robot. All they do is select the rehabilitation exercise, set a number of parameters based on the patient (e.g., stiffness and movement exercise) and start the rehabilitation. The controller is the combination of all the parts corresponding to the modules required for accomplishing the selection made, as shown in Fig. 15.

Another important aspect of the modular implementation of the control system is the possibility of configuring and reconfiguring Orocos components at runtime. This is very useful, for example, in

https://doi.org/10.1017/S0263574715000120 Published online by Cambridge University Press

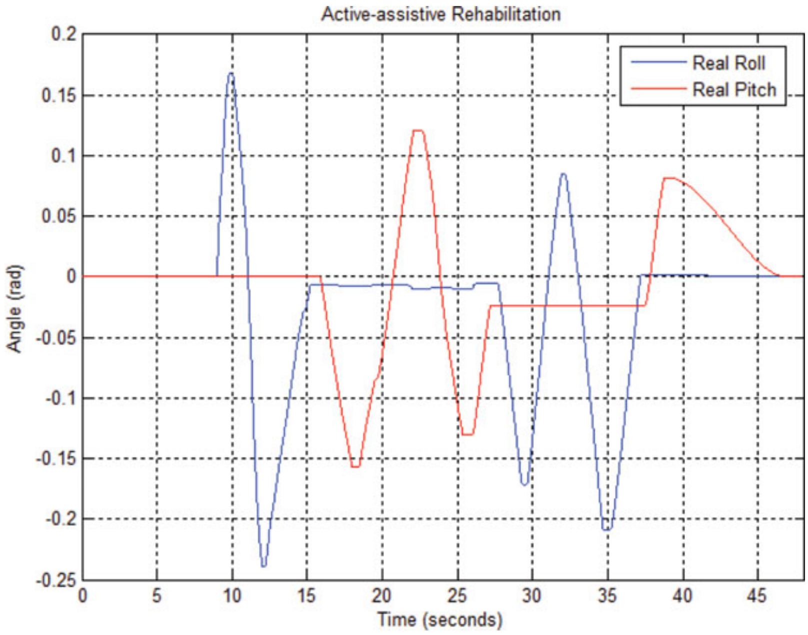

Fig. 13. Active-assistive rehabilitation.

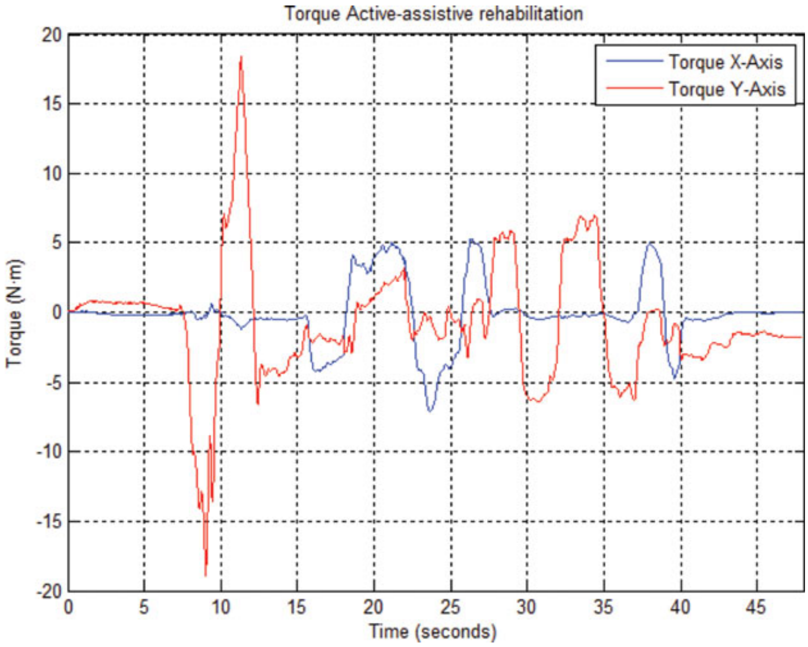

Fig. 14. Forces measured in Active-Assistive rehabilitation.

active-assistive exercises in which, depending on the severity of the injury, the robot must assist the patient more or less.

In experimental tests that have been performed, depending on the subject, the value of the stiffness constant in Eq. (19) should be changed in the rehabilitation module. Thus, this value is easily modified in the configuration phase, so a recompilation of the module is not required.

5.5. *Teleoperation and display of the parallel robot using ROS*

Finally, using the meta-operating system ROS, as well as integration with Orocos,[44] an application has been developed for displaying and teleoperating the parallel robot. To perform teleoperation, a

https://doi.org/10.1017/S0263574715000120 Published online by Cambridge University Press

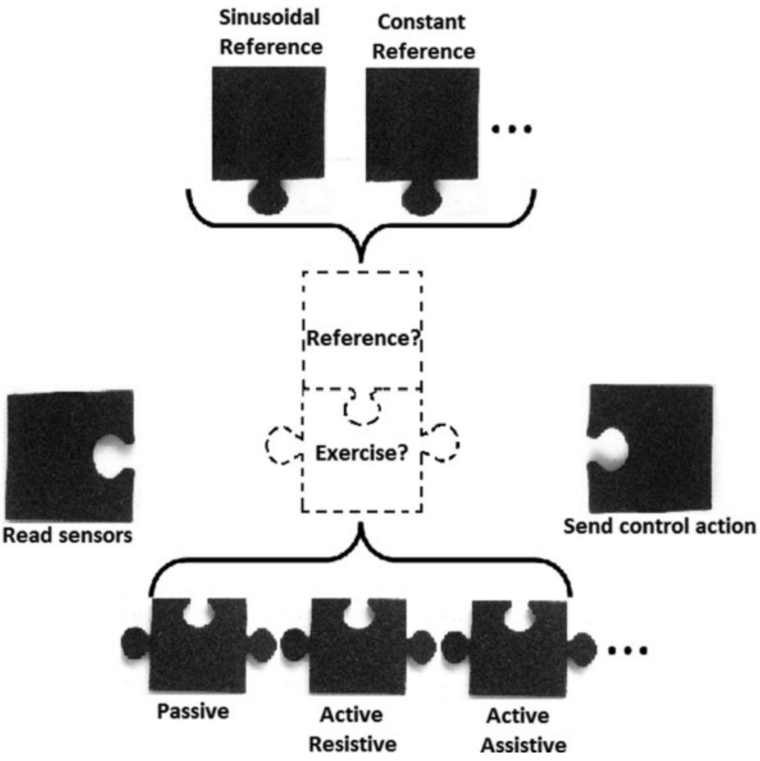

Fig. 15. Modular scheme.

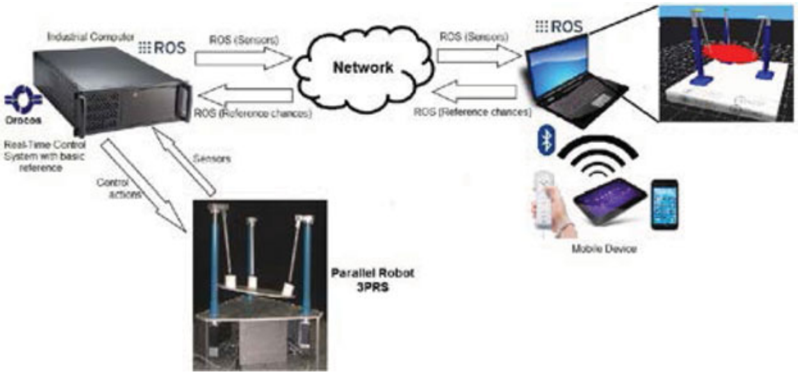

Fig. 16. Scheme application Orocos-ROS integration.

WiiMote has been used. For displaying the robot, a ROS package called Gazebo Simulator has been utilized (see Fig. 17).

As can be seen in Fig. 16, the scheme is divided into two parts: Industrial PC and client PC (laptop).

In the industrial PC, the position control is running in real time (using Orocos) at a frequency of 100 Hz. At each iteration (10 ms), the control action to apply to the three actuators is calculated. In addition to calculating and sending the control action, the Orocos module is responsible for publishing the actual position of the robot at each period on an ROS topic. At the same time, other Orocos module reads the value of the reference changes (with the values of the Wii Remote) and modifies the reference.

Meanwhile, in the client computer, a ROS node implements Bluetooth protocol and communicates with the WiiMote, while another node is responsible for the display depending on the actual position values of the robot.

Using this model, a real-time display of the parallel robot is performed. It is remarkable that the values read by the node that implements the virtual model come directly from the sensors on the

https://doi.org/10.1017/S0263574715000120 Published online by Cambridge University Press

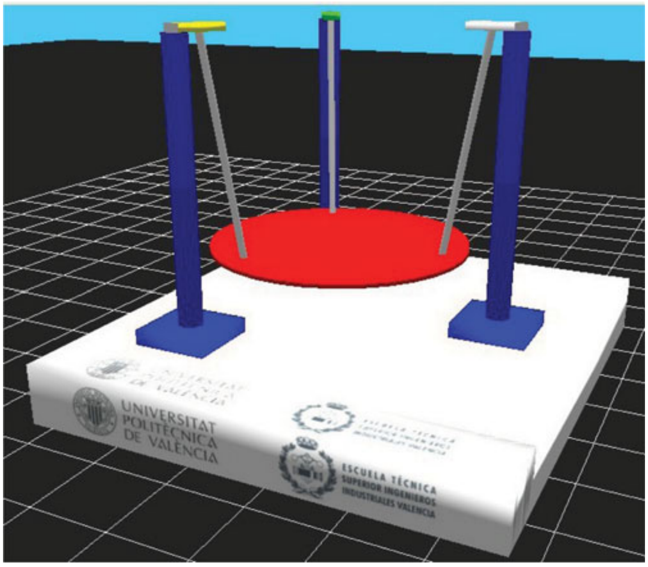

Fig. 17. Gazebo CAD model of the actual parallel robot.

real robot. Thus, if an error occurs in the control or there is an unexpected failure, the exact current position of the actual robot can be seen on this virtual model. Furthermore, since the model only displays three integer data (one for each joint), the bandwidth required for communication is very small.

Different teleoperated rehabilitation exercises have therefore been achieved in a simple and intuitive way.

6. Conclusions

Despite the potential advantages of parallel manipulators for lower limb rehabilitation robots, they have not been used as much as in other fields of robotics. Specifically, different approaches of parallel robots for ankle rehabilitation were described in the introduction.

A low-cost ankle rehabilitation robot has been presented in this paper. The kinematics and dynamics of the rehabilitation robot were developed and validated both in simulation and experimentally, achieving a high degree of accuracy for tracking position and force. Furthermore, due to the kinematic configuration of the robot, it can be adapted to the two relevant axes in ankle rehabilitation.

Different types of ankle rehabilitation exercises have been implemented and tested. Because of the versatility and adaptability of the system, it has been possible to perform several exercises with patients with different physical characteristics without the need to program the robot. These exercises include both passive and active exercises and, in the case of active ones, resistive and assistive exercises. For the latter, force control strategies have been implemented.

The results obtained indicate that these rehabilitation exercises train the specific part of the ankle correctly, following the advice of the physiotherapist.

Taking advantage of a component-based middleware (Orocos), the entire control system has been implemented modularly. So, different parameters such as the stiffness constant or position reference, in addition to the type of exercise, can be changed for each patient with no need for the medical staff to have any knowledge of robotics, meaning that one of the main problems of installing a robot in a clinic has been solved satisfactorily.

Finally, a novelty application related to the display and teleoperation of the robot has been implemented using Orocos-ROS integration. This way, any kind of exercise can be performed and the medical staff is not required to be physically present with each patient. The performance of the robot can be motorized in real time through the CAD model simulation.

https://doi.org/10.1017/S0263574715000120 Published online by Cambridge University Press

Acknowledgements
This work was partially financed by Plan Nacional de I+D, Comisión Interministerial de Ciencia y Tecnología (FEDER-CICYT) under the projects DPI2011-28507-C02-01 and DPI2013-44227-R. This research was also partially funded by the CDCHT-ULA Grant I-1377-13-02-F.

References

1. Y. D. Patel and P. M. George, "Parallel manipulators applications—a survey," *Mod. Mech. Eng.* **2**, 57–64 (2012).
2. D. Pisla, B. Gherman, C. Vaida, M. Suciu and N. Plitea, "An active hybrid parallel robot for minimally invasivesurgery," *Robot. Comput.-Integr. Manuf.* **29**, 203–221 (2013).
3. I. Díaz, J. Gil and E. Sánchez, "Lower-limb robotic rehabilitation: Literature review and challenges," *J. Robot.* **2011**, Article ID 759764, 1–11 (2011).
4. A. del-Ama, A. Koutsou and J. Moreno, "Review of hybrid exoskeletons to restore gait following spinal cord injury," *J. Rehabil. Res. Dev.* **49**(4), 497–514 (2012).
5. G. Colombo, M. Joerg and R. Schreier, "Treadmill training of paraplegic patients using a robotic orthosis," *J. Rehabil. Res. Dev*. **37**(6), 693–700 (2000).
6. S. Hesse and D. Uhlenbrock, "A mechanized gait trainer for restoration of gait," *J. Rehabil. Res. Dev.* **37**(6), 701–708 (2000).
7. M. Peshkin, D. Brown and J. Santos-Munné, "KineAssist: A Robotic Overground Gait and Balance Training Device," *Proceedings of the 9th IEEE International Conference on Rehabilitation Robotics, (ICORR '05)*, Evanston, Ill, USA (Jul. 2005) pp. 241–246.
8. C. Schmitt, P. Métrailler and A. Al-Khodairy, "The Motion Maker$^{\mathrm{TM}}$: A Rehabilitation System Combining an Orthosis with Closed-Loop Electrical Muscle Stimulation," *Proceedings of the 8th Vienna International Workshop on Functional Electrical Stimulation*, Vienna, Austria (Sep. 2004) pp. 117–120.
9. A. van Delden, C. Peper and G. Kwakkel, "A systematic review of bilateral upper limb training devices for poststroke rehabilitation," *Stroke Res.*, *Treat.* Article 972069 (2012) pp. 1–17.
10. K. Bharadwaj and T. Sugar, "Kinematics of a Robotic Gait Trainer for Stroke Rehabilitation," *Proceedings of the IEEE International Conference on Robotics and Automation, (ICRA '06)*, Orlando, Fla, USA (May 2006) pp. 3492–3497.
11. G. Prange and M. Jannink, "Systematic review of the effect of robot-aided therapy on recovery of the hemiparetic arm after stroke," *J. Rehabil. Res. Dev.* **43**(2), 171–184 (2006).
12. E. Rocon, J. Belda-Lois and A. Ruiz, "Design and validation of a rehabilitation robotic exoskeleton for tremor assessment and suppression," *IEEE Trans. Neural Syst. Rehabil. Eng.* **15**(3), 367–378 (2007).
13. J. Saglia, N. Tsagarakis, J. Dai and D. Caldwell, "Control strategies for patient-assisted training using the ankle rehabilitation robot (ARBOT)," *IEEE/ASME Trans. Mechatronics* **99**, 1–10 (2012).
14. Y. Tsoi, S. Xie and A. Graham, "Design, modeling and control of an ankle rehabilitation robot," *Des. Control Intell. Robot. Syst.*, vol. 177, 377–399 (2012).
15. C. Syrseloudis and I. Emiris, "Design framework for a simple robotic ankle evaluation and rehabilitation device," *30th Annual International IEEE EMBS Conference*, Vancouver (Aug. 2008) pp. 4310–4313.
16. J. Dai, T. Zhao and C. Nester, "Sprained ankle physiotherapy based mechanism synthesis and stiffness analysis of a robotic rehabilitation device," *Auton. Robots* **16**, 207–218 (2004).
17. J. Yoon, J. Ryu and K. Lim, "Reconfigurable ankle rehabilitation robot for various exercises," *J. Robot. Syst.* **11** 15–33 (2006).
18. M. Girone, G. Burdea and M. Bouzit, "The Rutgers Ankle Orthopedic Rehabilitation Interface," *Proceedings of the ASME International Mechanical Enge. Congr.* Dyn. Syst. Control Div., Nashville, TN (Nov. 1999) vol. 67, pp. 305–312.
19. C. Syrseloudis and I. Emiris, "A Parallel Robot for Ankle Rehabilitation-Evaluation and its Design Specifications," *Proceedings of IEEE BioInformatics and BioEngineering* (2008) pp. 1–6.
20. Y. Fan and Y. Yin, "Mechanism Design and Motion Control of a Parallel Ankle Joint for Rehabilitation Robotic Exoskeleton," *Proceedings of IEEE Robotics and Biomimetics*, China (2009) pp. 2527–2532.
21. C. Wang, F. Yuefa, G. Sheng and Z. Chagchum, "Design and kinematic analysis of redundantly actuated parallel mechanisms for ankle rehabilitation," *Robotica* 1–19 (2014).
22. F. Patane and P. Cappa, "A 3-DOF parallel robot with spherical motion for the rehabilitation and evaluation of balance performance," *IEEE Trans. Neural Syst. Rehabil. Eng.* **19**(2), (2011) pp. 157–166.
23. M. Vallés, M. Díaz-Rodríguez, Á. Valera, V. Mata, and Á. Page, "Mechatronic development and dynamic control of a 3-DOF parallel manipulator," *Mech. Based Des. Struct. Mach.* **40**(4), 434–452 (2012).
24. J. de Jalon and E. Bayo, *Kinematic and Dynamic Simulation of Multibody Systems: The Real-Time Challenge* (*Springer-Verlag*, New-York, 1994).
25. Y. Li and Q. Xu, "Kinematic analysis of a 3-PRS parallel manipulator," *Robot. Comput.-Integr. Manuf.* **23**(4), 395–408 (2007).
26. W. Khalil and E. Dombre, *Modeling, Identification and Control of Robots*. (London, Hermes Penton, 2002).
27. M. Díaz-Rodríguez, V. Mata, Á. Valera and Á. Page, "A methodology for dynamic parameters identification of 3-DOF parallel robots in terms of relevant parameters," *Mech. Mach.* **45** 1337–1356 (2010).

https://doi.org/10.1017/S0263574715000120 Published online by Cambridge University Press

28. R. Ortega and M. Spong, “Adaptive motion control of rigid robots: A tutorial,” *Autom.* **25**(6), 877–888 (1989).
29. B. Paden and R. Panja, “Globally asymptotically stable PD+ controller for robot manipulators,” *Int. J. Control* **47**(6), 1697–1712 (1988).
30. J. J. E. Slotine and W. Li, “On the adaptive control of robot manipulators,” *Int. J. Robot. Res.* **6**, 49–59 (1987).
31. N. Sadegh and N. R. Horowitz, “Stability and robustness analysis of a class of adaptative controllers for robotics manipulators,” *Int. J. Robot. Res.* **9**, 74–94 (1990).
32. R. Volpe and P. Khosla, “A theoretical and experimental investigation of explicit force control strategies for manipulators,” *IEEE Trans. Autom. Control* **38**(11), 1634–1650 (1993).
33. K. Aström and R. Murray, *Feedback Systems: An Introduction for Scientists and Engineers*. Princeton University Press (2010).
34. B. Siciliano and L. Villani, *Robot Force Control* (Kluwer Academic Publishers, Norwell, MA, USA, 2000, ISBN: 0792377338).
35. N. Hogan, “Impedance control: An approach to manipulation: Part illapplications,” *J. Dyn. Syst. Meas. Control* **107**(2) 17–24 (1985).
36. L. Sciavicco and B. Siciliano, *Modelling and Control of Robot Manipulators*. (Mc Graw Hill, London, 2000).
37. Z. Lu and A. A. Goldenberg, “Robust impedance control and force regulation: Theory and experiments,” *Int. J. Robot. Res.* **14**(3), 225–254 (1995).
38. H. Bruyninckx, “Open robot control software: The OROCOS project,” Available at : http://www.orocos.org. 2013.
39. D. Alonso, J. Pastor, P. Sánchez, and B. Álvarez, “Generación automática de software para sistemas de tiempo real: Un enfoque basado en componentes, modelos y frameworks,” *Rev. Iberoamericana de Autom. e Inform. Ind.* **9**(2), 170–181 (2012).
40. M. Quigley, K. Conley and B. Gerkey, “ROS: An Open-Source Robot Operating System,” *In ICRA Workshop on Open Source Software*, vol. 3, no. 3.2. (2009).
41. S. Siegler, J. Chen and C. Schneck, “The three-dimensional kinematics and flexibility characteristics of the human ankle and subtalar joints–part I: Kinematics,” *J. Biomech.* **110**(4) 364–373 (1988).
42. M. Dettwyler, A. Stacoff and I. K. Quervain, “Modelling of the ankle joint complex. Reflections with regards to ankle prostheses,” *Foot Ankle Surg.* **10**(3), 109–119 (2004).
43. M. Safran, R. Benedetti and A. Bartolozzi, “Lateral ankle sprains: A comprehensive review: part 1: etiology, pathoanatomy, histopathogenesis, and diagnosis,” *Med. Sci.* **31**(7), 429–437 (1999).
44. R. Smits and H. Bruyninckx, “Composition of complex robot applications via data flow integration,” *IEEE Int. Conf. Robotics and Automation (ICRA)*, Shanghai, China (2011) pp. 5576–5580.

https://doi.org/10.1017/S0263574715000120 Published online by Cambridge University Press