Iterative estimation of the end-effector apparent gravity force for 3DoF impedance haptic devices

Alessandro Formaglio, Sara Mulatto and Domenico Prattichizzo

***Abstract*— The position-dependent apparent gravity acting on the end-effector of impedance haptic devices may represent a loss of transparency, and generally requires to be actively compensated. To that purpose, a new approach to the problem of gravity compensation was introduced in our previous work [1]. The apparent gravity is preliminarily estimated in a given set of positions inside the workspace, then the acquired data are used to set up a suitable gravity compensation control law. At each position of the aforementioned set, the estimation is performed via an iterative method based on a nonlinear model subject to a feedback-linearizing PD controller. This paper, which builds upon our previous contribution, improves the mathematical formulation of the problem and addresses the analysis of stability and convergence properties of the iterative estimation method. Finally guidelines for performance-based parameter design are discussed. Validation experiments have been performed, and results are in good agreement with theoretical findings.**

I. INTRODUCTION

In the last decade impedance force-feedback devices have been strongly improved thanks to both mechatronic and software developments. As a consequence, improved performance devices at lower prices are available today, as for example the Phantom Omni (Sensable Technologies - www.sensable.com) or the Falcon (Novint - home.novint.com). Thanks to technological development, using haptic devices has become a common practice in a large variety of disciplines, involving even medicine [2], Psychophysics and Neurophysiology [3]. In any virtual reality application, the inertial and gravitational properties characterizing the haptic manipulator may represent a disturbance which can strongly affect the simulation realims. In the scope of this research, we focus on the problem of compensating the apparent gravity acting on the end-effector, due to the gravitational contributions acting on the whole kinematic chain of the haptic device. Several works can be found in the literature where different techniques are employed in order to actively cancel effects of gravity on haptic manipulators [4]–[7]. Although the large variety of approaches, they share some common traits. Generally, gravity compensation is based on the knowledge of device kinematics, or otherwise force sensors are employed in order to achieve device mechanical transparency using admittance control laws.

In robotics, the gravity compensation is rarely approached in the task space, because in general the map from the joint space to the end-effector coordinates is not injective and this causes many problem for control design in the task space. On the other hand, compensating gravity in the task space would dramatically simplify the problem getting an increased robustness and efficiency. However, we noted that most of the commercial haptic interfaces feature 3DoF and a 3-dimensional workspace without any redundancy in the joint space. This led us to adopt the new approach presented first in [1], where the proposed gravity compensation algorithm suites for 3DoF haptic devices and is formulated directly in the task space. This algorithm is split into two phases. The first phase, referred to as *off-line autocalibration*, is devoted to learn the nonlinear injective relationship between the end-effector apparent gravity and its position in the task space. To that end, the device workspace is discretized using a cubic grid, whose vertices build up the set of positions in which the end-effector apparent gravity must be measured. The estimation procedure is composed by two nested loops. In the inner loop, an iterative procedure is implemented to estimate the apparent gravity acting on the end-effector at the current vertex. The outer loop is devoted to iterate the gravity estimation for each vertex of the grid. Note that the off-line autocalibration procedure has been designed to be automatically performed by the device.

Once the gravity compensation terms have been estimated at each vertex of the cubic grid, the second part of the algorithm, referred to as *on-line gravity compensation*, can be applied. According to data acquired off-line, the gravity compensation term at a generical end-effector position X is determined using a piece-wise-linear approximation: first, the grid cube containing X is identified, and then the required compensation term is computed using the trilinear interpolation among the values previously estimated at all vertices of that cube.

In this paper, which builds upon our previous work [1], we report the results of further investigations dealing with the iterative estimation method applied in the off-line autocalibration phase. In particular, we studied stability and convergence properties of that method, and provide guidelines for performance-based parameter design.

The remainder of this article is structured as follows: Section II deals with dynamical modeling and controller design issues. Section III provides the mathematical formulation of the iterative estimation technique. In Section IV stability and convergence properties of the proposed method are discussed. Section V reports experiments and results. Finally, in Section VI concludes this paper.

A.Formaglio, S.Mulatto and D.Prattichizzo are with Department of Information Engineering. University of Siena, via Roma 56, 53100 Siena, Italy. {formaglio,mulatto,prattichizzo}@dii.unisi.it

II. MODELING AND CONTROL

The haptic end-effector is modeled as a point-like position-dependent mass $M(X)$, subject to gravity $G(X)$ and to the virtual environment force $F(X,\dot{X})$ to be rendered by actuators, yielding the following non-linear dynamics:

$$M(X)\ddot{X} + G(X) = F(X,\dot{X}) \tag{1}$$

where $X \in \mathbb{R}^3$ is the end-effector position, $M(X) \in \mathbb{R}^{3\times3}$ and $F(X,\dot{X}), G(X) \in \mathbb{R}^3$. We assume an isotropic model of the apparent mass $M(X)$, defined as a function of the apparent gravity, hence:

$$M(X) = \frac{\|G(X)\|}{g} I$$

where $g = 9.81\frac{m}{s^2}$ and I is a $3x3$ identity matrix. Referring to the top of the Fig. 1, let $F_H(X,\dot{X}) = F(X,\dot{X}) - G(X)$ be the resultant force really perceived by the user during haptic interaction. Until effects of gravity are not compensated, $F_H(X,\dot{X})$ may be quite different from $F(X,\dot{X})$, causing a possible loss of transparency. To that purpose, the control force $F(X,\dot{X})$ should integrate a compensation term ideally equal to $G(X)$ in addition to the virtual environment force, in order to cancel or at least strongly reduce the effects of the gravity, (see the bottom of the Fig. 1).

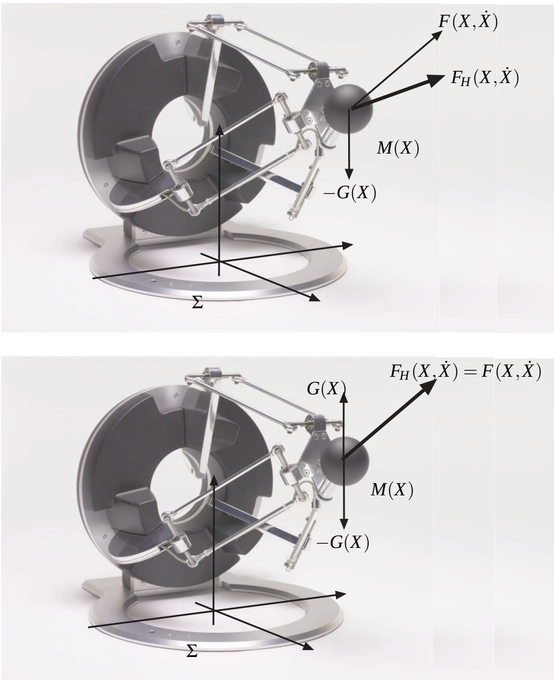

Fig. 1. Top: dynamical contributions normally acting on the haptic device end-effector. Bottom: the effects of apparent can be ideally cancelled applying a compensation term .

A proportional-derivative (PD) controller with a feedforward term is applied to hold the end-effector at the position X_D where the apparent gravity must be estimated. Assuming that the apparent gravity is exactly known, the force rendered by the actuators is:

$$F(X,\dot{X}) = K_P(X_D - X) - K_D\dot{X} + G(X) \tag{2}$$

where K_P and K_D are the proportional and derivative gain matrices.

Combining the (1) with the control (2), it yields the nonlinear dynamical system

$$M(X)\ddot{X} = K_P(X_D - X) - K_D\dot{X} \tag{3}$$

Ideally, the gravitational term $G(X)$ is cancelled by the control, but the non-linearity in the inertial term $M(X)$ holds.

The PD parameters K_P and K_D can be chosen in order to linearize the closed-loop dynamics. In the literature, this approach is referred to as *feedback linearization* [8], and allows to come up with a transformation of the open-loop system yielding a closed-loop linear system. The linearizing PD parameters are:

$$K_P(X) = \beta M(X) \quad \text{and} \quad K_D(X) = \alpha M(X) \tag{4}$$

where $\alpha, \beta \in \mathbb{R}$, and $\alpha, \beta > 0$. Substituting the linearizing parameters (4) in (3), we have:

$$M(X)\ddot{X} = \beta M(X)(X_D - X) - \alpha M(X)\dot{X}$$

and hence:

$$\ddot{X} = \beta(X_D - X) - \alpha\dot{X} \tag{5}$$

which is a linear and asymptotically stable system.

A fast-tracking PD controller can be designed using classical control theory [9]. The PD parameters can be chosen according to the desired closed-loop performance, in terms of the raise time T_s and the percentage overshoot $\hat{p}$ characterizing the unit-step time response of the haptic probe position X. Given the user-desired pair (T_s, $\hat{p}$), the corresponding parameters α and β can be computed through analytic relationships, graphically represented in Figures 2 and 3. We summarize the simple graphical procedure to design the controller parameters α and β. First, choose the desired values T_s and $\hat{p}$; then, referring to the curves in Figure 2, select the value of β corresponding to the desired T_s and $\hat{p}$; finally, select the value of α using the curves in Figure 3 according to the desired $\hat{p}$ and the β previosly chosen.

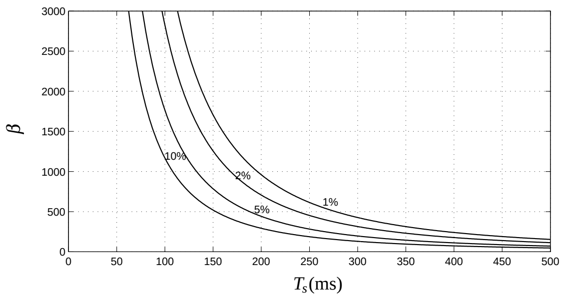

Fig. 2. Design plots for parameter β. The curves represent the relationship between β and the raise time T_s, parametrized for several values of percentage overshoot $\hat{p}$.

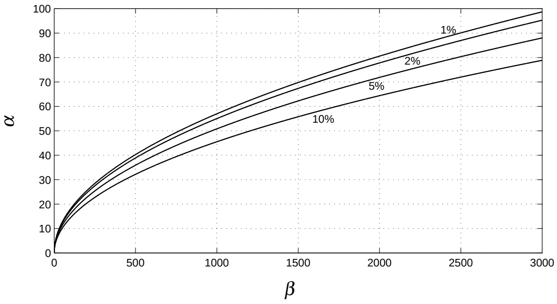

Fig. 3. Design plots for parameter α. The curves represent the relationship between α and β, parametrized for several values of percentage overshoot $\hat{p}$.

III. ITERATIVE ESTIMATION TECHNIQUE

The model discussed in the previous section assumes that the nonlinear terms are known so that the controlled system can match the performance speficications. Actually, these are the terms that the proposed procedure aims to measure and are unknown. The basic idea consists of replacing the exact terms $M(X)$ and $G(X)$ appearing in (2) and (4) with their estimates $\hat{M}(X)$ and $\hat{G}(X)$, and using recursion to update such values in order for the estimation error to converge to zero.

Henceforth, $\hat{G}_j(X_i)$, $\hat{M}_j(X_i)$, $K_P(i,j)$ $K_D(i,j)$ will represent the values computed at the j^{th} iteration performed at the i^{th} vertex X_i.

Note that the PD control parameters will depend on the apparent gravity estimation, hence the following constraint must hold:

$$\hat{G}_j(X_i) \neq [0,0,0]^T \quad \forall i,j. \tag{6}$$

A. Initialization rules

As pointed out in the Introduction, the off-line autocalibration in composed by two nested loops [1]. In order to globally initialize loop variables for $i=0$ and $j=0$, a simple procedure is applied. The user is asked to manually bring the end-effector almost in the center of its workspace. At a time instant decided by the user, a central elastic force field (with user-defined stiffness k_0) is activated. Then the user can release the end-effector while the force field holds it hanged to the position chosen by the user. The steady-state positioning error between the field center and the actual end-effector position is used to initialize the estimate $\hat{G}_0$.

Let $X_0 \in \mathbb{R}^3$ be the position arbitrarily chosen by the user, as well as the center of the force field. Let X be the current end-effector position. The force field is rendered according to the equation:

$$F(X,\dot{X}) = k_0(X_0 - X)$$

The steady state position X_∞ is considered to be reached as soon as n_S consecutive velocity samples satisfy the condition $\|\dot{X}\| \leq \varepsilon_S$, where n_S and ε_S are user-defined parameters.

Hence, the global initialization of the gravity compensation term $\hat{G}_0(X_0)$ is computed as:

$$\hat{G}_0(X_0) = k_0\,(X_0 - X_\infty) \tag{7}$$

then we have:

$$\hat{M}_0(X_0) = \frac{\|\hat{G}_0(X_0)\|}{g} I$$

and finally:

$$K_D(0,0) = \alpha \hat{M}_0(X_0) \quad \text{and} \quad K_P(0,0) = \beta \hat{M}_0(X_0) \tag{8}$$

The inner loop terminates as soon as the final estimation $\hat{G}(X_i)$ is achieved at vertex X_i. The outer loop will move the end-effector to the next vertex, and a new initialization for the inner loop at X_{i+1} is obtained by setting:

$$\hat{G}_0(X_{i+1}) = \hat{G}(X_i), \quad \forall i. \tag{9}$$

B. Recursion rules

Recursion for this method builds upon a simple basic principle. At the j^{th} iteration for a generical vertex X_i, the apparent gravity estimation error will reflect in a non-zero end-effector positioning error at the steady state. Hence, the steady-state effort exerted by the proportional part of the control is used to update the gravity estimation. Then this procedure is iterated. In mathematical terms, given the current vertex X_i, the force to render at the end-effector at the j^{th} step is computed as:

$$F_j(X,\dot{X}) = K_P(i,j)(X_i - X) - K_D(i,j)\dot{X} + \hat{G}_j(X_i) \tag{10}$$

The PD parameter $K_P(i,j)$ and $K_D(i,j)$ are computed according to the estimation at $(j-1)^{th}$ step:

$$K_P(i,j) = \beta \hat{M}_{j-1}(X_i) \quad \text{and} \quad K_D(i,j) = \alpha \hat{M}_{j-1}(X_i) \tag{11}$$

where

$$\hat{M}_{j-1}(X_i) = \frac{\|\hat{G}_{j-1}(X_i)\|}{g} I$$

The apparent gravity estimate is updated at steady state of the j^{th} iteration, according to the following recursion rule:

$$\hat{G}_{j+1}(X_i) = K_P(i,j)(X_i - X) + \hat{G}_j(X_i) \tag{12}$$

Again, the steady state is considered to be reached as soon as n_S consecutive velocity samples satisfy the condition $\|\dot{X}\| \leq \varepsilon_S$.

The final estimation $\hat{G}(X_i)$ at the i^{th} vertex is available as soon as the convergence is achieved. The stop condition for the inner loop is matched as soon as n_G consecutive values of estimated gravity satisfy the condition $\|\hat{G}_j(X_i) - \hat{G}_{(j-1)}(X_i)\| \leq \varepsilon_G$, where n_G and ε_G are user-defined parameters, .

Now, in order to move the end-effector towards the vertex $i+1$, the controller (10) is used again in the outer loop, with $\hat{G}_0(X_{i+1})$, computed according to the relationship (9) and the PD parameters computed as:

$$K_D(i+1,0) = \alpha \hat{M}(X_i) \quad \text{and} \quad K_P(i+1,0) = \beta \hat{M}(X_i) \tag{13}$$

IV. STABILITY AND CONVERGENCE

In this section the convergence of the iterative method implemented in the inner loop is analyzed. To that end, we will study stability and convergence in an arbitrary desired position X_D in the workspace, such that the results can be applied for each vertex of the desired grid, without loss of generality.

Recalling that the proposed method originates from the one proposed in [10], also the proof of convergence presented in the following is derived from the work by De Luca and Panzieri.

Note that according to the recursion rules reported in the previous section, a generical iteration j ends only when the corresponding dynamical system has reached the steady state. This in turn means that before analyzing the convergence properties of the iterative method, it is required to guarantee the stability of the controlled system at each iteration. To that end, consider:

$$M(X)\ddot{X} + G(X) = K_{P_j}(X_D - X) - K_{D_j}\dot{X} + \hat{G}_j \tag{14}$$

that is obtained combining equations (1) with (10) and simplifying the notation by defining:

$$K_{D_j} = \frac{\alpha\|\hat{G}_j\|}{g}I \quad \text{and} \quad K_{P_j} = \frac{\beta\|\hat{G}_j\|}{g}I$$

Let the gravity estimation error at j^{th} iteration $E_j(X)$ be defined as:

$$E_j(X) = G(X) - \hat{G}_j$$

In the following, in order to come up with analytical results we will assume that the error depends linearly on position X, in other terms we will approximate it with the first order term of its Taylor series:

$$E_j(X) \approx \tilde{E}_j X$$

where $\tilde{E}_j \in \mathbb{R}^{3x3}$ is the Jacobian matrix. The system (14) can be rewritten as:

$$M(X)\ddot{X} = K_{P_j}(X_D - X) - K_{D_j}\dot{X} - \tilde{E}_j X \tag{15}$$

In order to study the stability of this dynamical system, we must note that the term $\tilde{E}_j X$ will shift the equilibrium point, then the system can eventually converge at the new destination point defined as:

$$X_{D_j} = [K_{P_j} + \tilde{E}_j]^{-1} K_{P_j} X_D$$

Since K_{P_j} is a controller parameter, it is always possible to choose a value β such that $[K_{P_j} + \tilde{E}_j] > 0$ (in the sense of positive definiteness), thus guaranteeing also that the inverse matrix $[K_{P_j} + \tilde{E}_j]^{-1}$ exists. Finally, the system (14) can be rewritten as:

$$M(X)\ddot{X} = \tilde{F}_j \tag{16}$$

where

$$\tilde{F}_j = -K_{P_j}(X - X_{D_j}) - K_{D_j}\dot{X} \tag{17}$$

Now consider the following energy-based Lyapunov function:

$$V(X) = \frac{1}{2}\left[\dot{X}^T M(X)\dot{X} + (X - X_{D_j})^T [K_{P_j} + \tilde{E}_j](X - X_{D_j})\right] \tag{18}$$

Since $M(X) > 0$ and $[K_{P_j} + \tilde{E}_j] > 0$ for a suitable choice of β, we have $V(X) > 0$. From straightforward computations, the derivative of $V(X)$ is:

$$\dot{V}(X) = -\dot{X}^T[\tilde{F}_j + K_{P_j}(X - X_{D_j})] \tag{19}$$

Using the (17), $\dot{V}(X)$ becomes:

$$\dot{V}(X) = -\dot{X}^T K_{D_j}\dot{X} \tag{20}$$

Since $K_{D_j} > 0$, we have $\dot{V}(X) < 0$, hence the system controlled by a feedback-linearizing PD controller is stable and converges at the equilibrium point X_{D_j}, whatever the initital position X from which the tracking begins.

Once the stability has been proven for the generical iteration j, we can discuss now the convergence of the estimation method. According to (12), the $(j+1)^{th}$ gravity estimation is computed as the steady-state effort of control (10) at the end of iteration j, which in turn is ideally equal to the apparent gravity force $G(X_{D_j})$ at position X_{D_j} (see equation (14)). Then, the difference between the estimations achieved at two consecutive iterations is:

$$\begin{aligned}\|\hat{G}_{j+1} - \hat{G}_j\| &= \|F_j(X_{D_j},0) - F_{j-1}(X_{D_j},0)\| \\ &= \|G(X_{D_j}) - G(X_{D_{j-1}})\|\end{aligned}$$

Note that being X the physical end-effector position inside its workspace, we have $X \in \mathbb{D} \subset \mathbb{R}^3$, where $\mathbb{D}$ is a compact set, and $G(X)$ is continuous and differentiable over $\mathbb{D}$ (since it is a physical feature of the system). Hence there exists $c \in \mathbb{R},\ c > 0$, such that:

$$\begin{aligned}\|G(X_{D_j}) - G(X_{D_{j-1}})\| &\leq c\|X_{D_j} - X_{D_{j-1}}\| \\ &\leq c(\|\Delta_j\| + \|\Delta_{j-1}\|)\end{aligned}$$

where $\Delta_j = X_D - X_{D_j}$ is the positioning error at steady state of iteration j. By comparing the inequalities above, it yields that:

$$\|\hat{G}_{j+1} - \hat{G}_j\| \leq c(\|\Delta_j\| + \|\Delta_{j-1}\|)$$

and hence, according to the (12):

$$\frac{\beta\|\hat{G}_j\|}{g}\|\Delta_j\| \leq c(\|\Delta_j\| + \|\Delta_{j-1}\|)$$

that can be finally rewritten as:

$$\|\Delta_j\| \leq \left(\frac{cg}{\beta\|\hat{G}_j\| - cg}\right)\|\Delta_{j-1}\|$$

Summarizing, choosing β such that $\beta\|\hat{G}_j\| \geq 2cg$, it stems that as $j \to \infty$ the error $\Delta_j \to 0$, thus also $X_{D_j} \to X_D$, $\hat{G}_j \to G(X_D)$, and $\tilde{E}_j \to 0$, for any position the tracking starts from, provided that the estimated gravity is never equal to zero. Hence in addition to stability, a suitable choice of the parameter β can guarantee also the convergence of the method, providing a reliable estimation of the apparent gravity force $G(X_D)$ acting on the end-effector at the desired position X_D.

Actually, the proofs above hold until actuator saturation is avoided, i.e. until the control effort at each time instant does not exceed the maximum nominal force that the device is capable to exert.

V. EXPERIMENTS

Several validation experiments have been performed applying the proposed algorithm to different haptic devices with 3DoF. As an example, in the following we report results achieved with an Omega Haptic Device (www.forcedimension.com), estimating the apparent gravity at the desired position $X_D = [0, -30, 0]^T$.

To design the PD controller we chose performance specifications $T_s = 200$ms and 5% overshoot, yielding the values $\alpha = 40$ and $\beta = 700$ (see Figures 2 and 3). The other user-defined parameters where $k_0 = 0.5\frac{N}{mm}$, $n_S = 20$, $\varepsilon_S = 0.1$(mm/s), $n_G = 5$ and $\varepsilon_G = 0.5$g.

The method required 14 iterations to achieve the estimation $\hat{G}(X_D)$, starting with the end-effector in an initial position randomly chosen as $[-9.2, -68.3, 11.6]^T$.

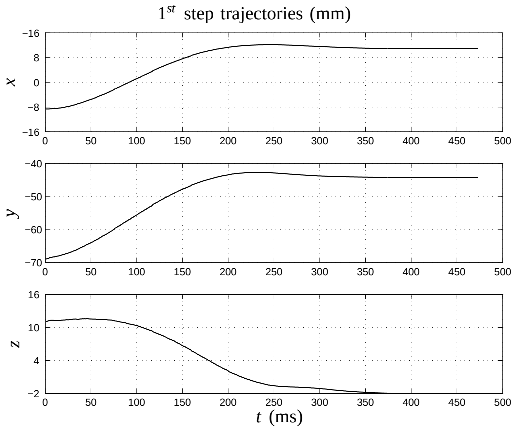

Fig. 4. End-effector trajectory recorded during the 1^{st} iteration step, decomposed in its x, y and z components.

As it was expected, the PD controlled system revealed to be stable at each iteration. In order to show in detail the dynamical behavior of the system during the iterations, as an example we report in Figures 4 through 7 the end-effector trajectories recorded during steps 1, 3, 6 and 9, respectively. For the reader's ease, each trajectory has been decomposed in its x, y and z components. In steps 1, 3 and 6 the raise time and the overshoots are close to those defined during the design phase. On the other hand, in step 9 the positioning error is already very close to zero, hence the system is reaching the convergence and is almost still, except for the slight effects of measurement noise.

Note that the time duration can be different for each trajectory, depending on the time required to match the steady-state condition defined by n_S an ε_S (in our experiments, we decided to lower bound the duration of each step at 100ms). Moreover, as the iterations proceed, the steady-state values of the x, y and z components approach to those of the target destination $X_D = [0, -30, 0]^T$. To discuss in detail the convergence properties of the applied method, the Figures 8 and 9 respectively show the positioning error Δ_j and the estimated mass $\hat{M}_j$ over the 14 iterations. As in our expectations, the error rapidly converges to zero, while the apparent mass reaches the convergence at $\hat{M}(X_D) = 217$g, which is very close to the value 214g we measured in X_D through a force sensor (represented by the dashed line in Figure 9).

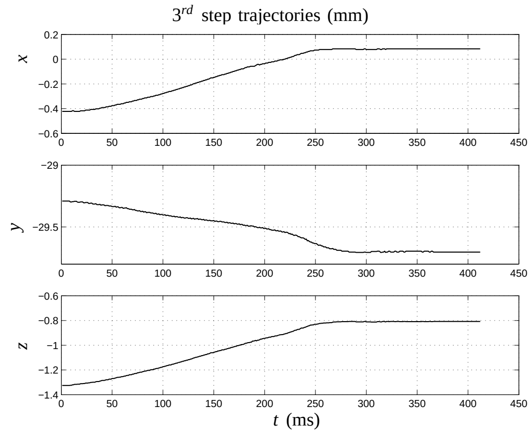

Fig. 5. End-effector trajectory recorded during the 3^{rd} iteration step, decomposed in its x, y and z components.

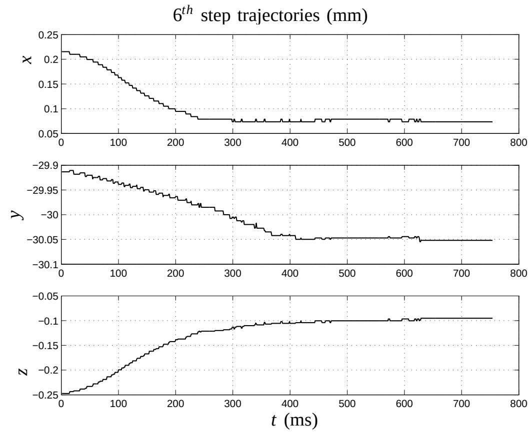

Fig. 6. End-effector trajectory recorded during the 6^{th} iteration step, decomposed in its x, y and z components.

VI. DISCUSSION AND CONCLUDING REMARKS

In this paper, we report results of further investigation on the autocalibrated gravity compensation algorihtm introduced in our previous work [1]. In particular, stability and convergence properties characterizing the iterative method for apparent gravity estimation have been analyzed. We provided analytical proofs that the system is stable during each iteration. Besides, the iterative procedure converges yielding the estimation of the apparent gravity acting in the target position. Validation experiments were performed with

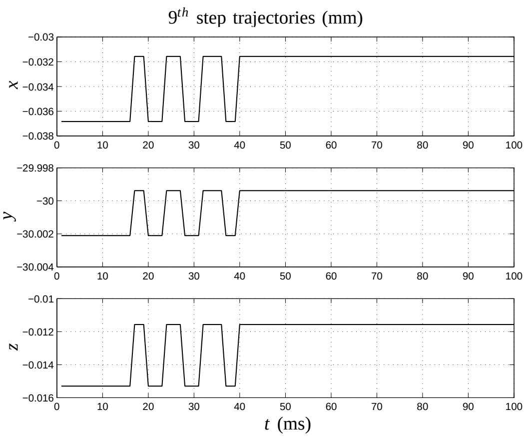

Fig. 7. End-effector trajectory recorded during the 9^{th} iteration step, decomposed in its x, y and z components.

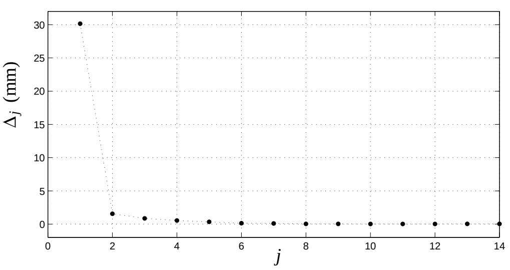

Fig. 8. Steady state postioning error Δ_j over all iterations.

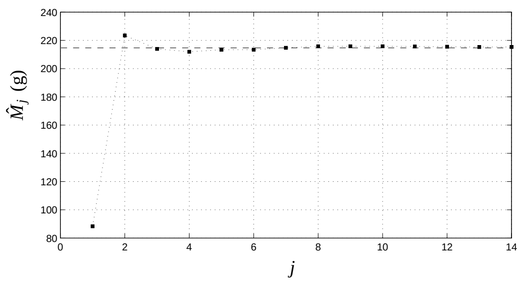

Fig. 9. Apparent mass estimation $\hat{M}_j$ over all iterations. The straight dashed line represents the apparent mass 214g measured in X_D through a force sensor.

several 3DoF devices, and the experimental results were in good agreement with the theoretical findings.

Note that the off-line autocalibration has been developed to be as general as possible, in order to be applied to any 3DoF impedance haptic device. Anyway, a preliminary phase of parameter tuning is required, according to the desired specifications such as the accuracy of the estimation and the resolution of the cubic grid. Achieving a reliable gravity estimate at a given position requires an amount of time depending on the particular dynamical behavior of the used haptic device, but also on system parameters. In order to upper bound the duration of the whole off-line autocalibration, the user can create a low-resolution grid featuring a low number of vertices, choose stop conditions not too strict and design a fast-tracking PD controller. On the other hand, the design choices that allow a fast autocalibration can conflict with each other and with the requirement to achieve an accurate gravity estimation. Recall that the on-line gravity compensation is based on a piece-wise linear approximation of the nonlinear relationship between the apparent gravity and the end-effector position [1]. Hence, it is clear that decreasing the grid resolution would improve performance in terms of time, but will make the interpolation intervals larger, with a possible degradation of the approximation accuracy. Similar remarks hold for the stop conditions: relaxing the conditions to check the steady state reaching and the estimate achievement would improve time performance but will degradate estimation accuracy as well.

Time performance can be improved by selecting low values of T_s and $\hat{p}$, corresponding to high α and β. Note that a high value β would also guarantee the stability and the convergence of the iterative method, but may also lead to actuators saturation. At worse case, saturation may occurr when the tracking error is maximum, i.e. while the end-effector begins to move from the current vertex from the next one, hence it directly depends on the side lenght of each cube of the grid. Given the maximum nominal force that the device is capable to exert, as β increases the cube side lenght should decrease in order to avoid saturation, with the aforementioned consequences in terms of time performance.

In summary, since the effective behavior of the proposed procedure will depend also on the unknown dynamics characterizing the haptic device of interest, it is hard to define precise specifications that hold for any existing device. It is our opinion that the main target should be the estimation accuracy in spite of a longer execution time, anyway a different parameter tuning can be chosen according to the remarks discussed so far.

REFERENCES

[1] A. Formaglio, M. Fei, S. Mulatto, M. de Pascale, and D. Prattichizzo. *Eurohaptics 2008*, volume LNCS 5024 of *Lecture Notes in Computer Science*, chapter Autocalibrated gravity compensation for 3DoF impedance haptic devices. Springer Verlag, 2008.

[2] RM Satava and SB Jones. Current and future applications of virtual reality for medicine. volume 86, pages 484–489, 1998.

[3] D. Prattichizzo and S. Rossi. Special Issue on Robotics and Neuroscience. *Brain Research Bullettin*, 2007. In Press.

[4] D. Checcacci, E. Sotgiu, A. Frisoli, CA Avizzano, and M. Bergamasco. Gravity compensation algorithms for parallel haptic interface. pages 140–145, 2002.

[5] R. Ott, M. Gutierrez, D. Thalmann, and F. Vexo. Improving user comfort in haptic virtual environments trough gravity compensation. 2005.

[6] JT Feddema and JL Novak. Whole arm obstacle avoidance for teleoperated robots. pages 3303–3309, 1994.

[7] M.C. Çavusoglu, D. Feygin, and F. Tendick. A critical study of the mechanical and electrical properties of the PHANToM TM haptic interface and improvements for high performance control. *Presence*, 11(6):1–18, 2002.

[8] H.K. Khalil. *Nonlinear systems*. Prentice Hall Upper Saddle River, NJ, 2002.

[9] S.P. Boyd and C.H. Barratt. *Linear Controller Design: Limits of Performance*. Prentice Hall, 1991.

[10] A. De Luca and S. Panzieri. An Iterative Scheme for Learning Gravity Compensation in Flexible Robot Arms. *Automatica*, 30(6):993–1002, 1994.