



UNIVERSITÀ
DEGLI STUDI
FIRENZE

FLORE

Repository istituzionale dell'Università degli Studi di Firenze

Design of dynamic simulators for testing of inertial sensors

Questa è la Versione finale referata (Post print/Accepted manuscript) della seguente pubblicazione:

Original Citation:

Design of dynamic simulators for testing of inertial sensors / B. Allotta; F. Giardi; P. Toni; G. Vettori; A. Ridolfi. - ELETTRONICO. - (2011), pp. 1-10. (Intervento presentato al convegno AIMETA tenutosi a Bologna nel 12-15 settembre 2011).

Availability:

This version is available at: 2158/599002 since:

Publisher:

AIMETA 2011

Terms of use:

Open Access

La pubblicazione è resa disponibile sotto le norme e i termini della licenza di deposito, secondo quanto stabilito dalla Policy per l'accesso aperto dell'Università degli Studi di Firenze (<https://www.sba.unifi.it/upload/policy-oa-2016-1.pdf>)

Publisher copyright claim:

(Article begins on next page)

Design of Dynamic Simulators for Testing of Inertial Sensors

B. Allotta¹, F. Giardi¹, A. Ridolfi¹, P. Toni¹, G. Vettori¹

¹*MDM Lab - Laboratorio di Modellazione Dinamica e Meccatronica*

Dipartimento di Energetica, Università di Firenze, Italia

E-mail: gvettori@mapp1.de.unifi.it

Keywords: Odometry, Dynamic Simulator, Testing, Inertial Sensors, Washout Filters.

SUMMARY. MDM Lab, the Laboratory of Mechatronics and Dynamic Modelling of the University of Florence, is developing innovative pose and motion estimation algorithms, based on the integration of odometers and inertial sensors, to be used in modern Automatic Train Protection and Control (ATP/ATC) systems. The testing of INS algorithms on board a real vehicle is expensive but, from the knowledge of the dynamic behaviour of the train, obtained via simulation or on board acquisition, it is possible to deploy a testing simulator capable of replicating in a realistic fashion the dynamic effects of vehicle motion on inertial sensors with a reduction of costs. This paper describes the implementation of algorithms which use an industrial robot as a dynamic simulator for testing of inertial sensors. This algorithm based on Washout Filters has been simulated in the Matlab-Simulink environment with a kinematic model of a 6-DOF serial robot.

1 Introduction

MDM Lab, the Laboratory of Mechatronics and Dynamic Modelling of the University of Florence, is developing an innovative pose estimation algorithm which integrates odometers and an Inertial Measurement Unit (IMU) - a triaxial accelerometer and a triaxial gyroscope - to be used in modern Automatic Train Protection and Control (ATP) systems. The estimation of speed and travelled distance of vehicles is critical for the safety and efficiency of the overall railway system. The objective of the cited stream of research work is to increase the accuracy of the pose and motion estimation, especially in critical adhesion conditions, through the implementation of INS (Inertial Navigation System). The testing of INS algorithms on board a real vehicle is expensive but, from the knowledge of the dynamics of the train, obtained via simulation or on board acquisition, it is possible to set up a testing simulator capable of reproducing realistically the dynamic effects of the motion on inertial sensors with a decrease of costs. This paper describes the implementation of algorithms which use an industrial robot as a dynamic simulator for testing of inertial sensors. Dynamic simulators are very powerful devices which permit to reproduce and to analyze experimentally the behaviour of a system under certain conditions imposed by the user. Very well known examples are flight simulators and drive simulators: a pilot can make experience of his interaction with the vehicle in realistic working conditions. A dynamic simulator is composed essentially by two elements: a Dynamic Model (DM) to be simulated in real-time and a device which replicates as widely as possible the effects of the movement of the vehicle on the pilot. Software components with the aim to make realistic the simulation are called Washout Filters (WF) [1] and hardware devices aimed to reproduce the motion are usually parallel robots (Stewart Platforms) but in the study at hand, even a serial manipulator can be used. In order to reproduce dynamic effects of the motion of a railway vehicle on inertial sensors, kinematic data, either synthetic (i.e. produced via dynamic simulation) or recorded in real test runs are required. A complete three-dimensional multibody model of a railway vehicle developed in Matlab-Simulink by the researchers of the MDM Lab, [2, 3] can produce a large amount of synthetic data with different working conditions and arbitrary tracks (low adhesion, curves, line gradient, etc). Data of acceleration and angular rate are used as input to Classical

Washout Filters [4] readapted for this particular application, because sensors have an higher sensibility than human vestibular apparatus. The consequence is the impossibility to test at the same time accelerometers and gyroscopes. An iterative first order kinematic algorithm (CLIK Closed Loop Inverse Kinematics [5]) with optimization of manipulability index has been designed using position and orientation values provided by WFs as references for the control. This algorithm has been simulated in the Matlab-Simulink environment with a kinematic model of a 6-DOF serial robot. The results show the reproducibility of the spectral components of accelerations and angular rates up to 10 Hz.

2 Washout Filters in Flight and Drive Simulators

The aim of WFs is not to reproduce the movement of the vehicle but that the simulator induces the same dynamic effects the pilot would experience in the real scenario. WF strategy splits the reproduction of high and low acceleration components:

- High frequency accelerations need a limited space so the simulator reproduces directly the same movement of the vehicle (Direct Linear Motion Strategy);
- Low frequency accelerations, which need large movements, can be reproduced tilting the simulator so that gravity vector is oriented with respect to the pilot in order to give the same accelerations to reproduce. This is the Tilting Strategy.

The same for the angular rates:

- High frequency components need small rotations so the simulator reproduces directly them (Direct Angular Motion Strategy);
- Low frequency components, which lead to continuous rotations, are not reproduced otherwise they would alter the perception of linear accelerations.

WF, through high pass and low pass filters, treats separately high and low frequency components of the movement to be reproduced.

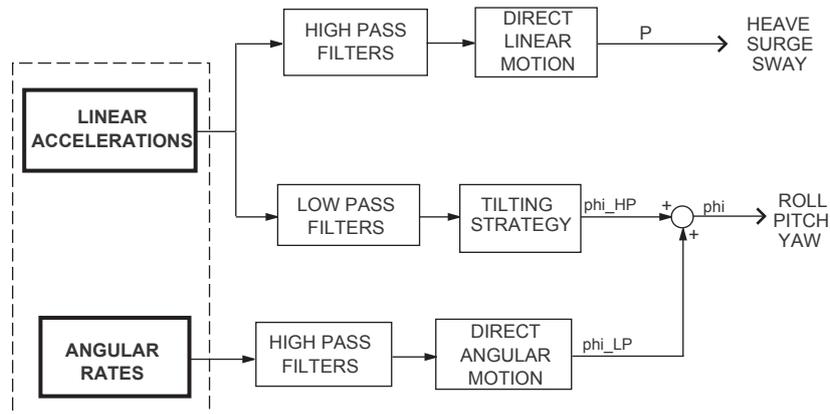


Figure 1: General Schema of Washout Filter

The algorithm used to implement WFs on the computer is called *Classical Washout Algorithm* [4], the first to be studied and, nowadays, the most common for commercial simulators. This is the simplest mathematically and the cheapest computationally. The drawback of this scheme consists in the fact that it uses linear elements, not exploiting fully the capabilities of the simulator. For a detailed description of Washout Algorithm refer to [1].

2.1 Adaptation of Washout Filters to the inertial sensors case

The design of a simulator used for the testing of sensors is substantially different with respect to driving simulators and, for some aspects, it is easier. For first instance there are not real-time needs: DM can calculate accelerations and angular rates and then they can be used as input data for the WF in a second moment.

The main problem we met in the transition is the higher sensibility of the sensors with respect to vestibular apparatus. For example, in order to reproduce an acceleration of 0.3 m/s^2 the platform should tilt with a pitch angle of $\theta = \arcsin(\frac{a}{g}) = 1,8^\circ$, opposite to the direction of motion. Because the smallest angular rate the vestibular apparatus is able to sense is $\dot{\theta}_{max} = 3^\circ/\text{s}$, platform tilts in a very short time: $t_{min} = \frac{\theta}{\dot{\theta}_{max}} = 0.6\text{s}$. So the pilot should sense the acceleration instantly. On the other side, because the gyroscopes have a sensitivity of $0.0125 \div 0.05^\circ/\text{s}/\text{LSB}$, in order that the sensor does not sense the tilting, $t_{min} = 36 \div 144$. It means that, during this time of 3 minutes, the accelerometer can't measure the correct acceleration because it has not reached the steady value. This example shows that it is not possible to test at the same time accelerometers and angular rates, but the procedure needs separate tests.

The main consequence of this separation is that it is possible to reproduce all components of frequency for the gyroscope.

2.2 Design of Washout Filters

As a result of the analysis in the Sect. 2.2 the architecture of the Washout Algorithm used in this work is reported in Fig. 2. We fix three frame systems:

- the absolute frame, A, related to the base of the simulator,
- the relative frame, R, related to the vehicle in motion,
- the frame, S, related to the inertial measurement unit.

Washout Location is the reference with respect to which accelerations and angular rates, both in R and S frame, are measured. So:

$$\vec{a}_{mis}^S = \vec{a}_{mis}^R \quad (1)$$

Both inputs are processed separately: accelerations are processed through the *Tilting Strategy* and the *Direct Linear Motion Strategy*, while angular rates through *Direct Angular Motion Strategy*. From the first two channels the algorithm calculates $\vec{P}_{acc}(t)$ and $\dot{\phi}_{acc}(t)$, that is the desired trajectory for the simulator in the Cartesian space. In order to work in the joint space, we must implement a Closed-Loop Inverse Kinematics (CLIK) [5], so that desired trajectories in the joint space $\vec{q}_{acc}(t)$ are calculated. $\dot{\phi}_{acc}(t)$ allows calculating $R_{S_{acc}}^A$, to be used in the next step of the control algorithm to rotate accelerations from relative (*R-frame*) to absolute frame (*A-frame*). The block *Drift Compensator*, after the high-pass filter (implemented as a first or a second order filter), allows avoiding drift of the position estimate, with a reset to the initial position of the platform, that is:

$$\vec{P} = \int \int a_{HP} dt = \vec{P}(0) \quad (2)$$

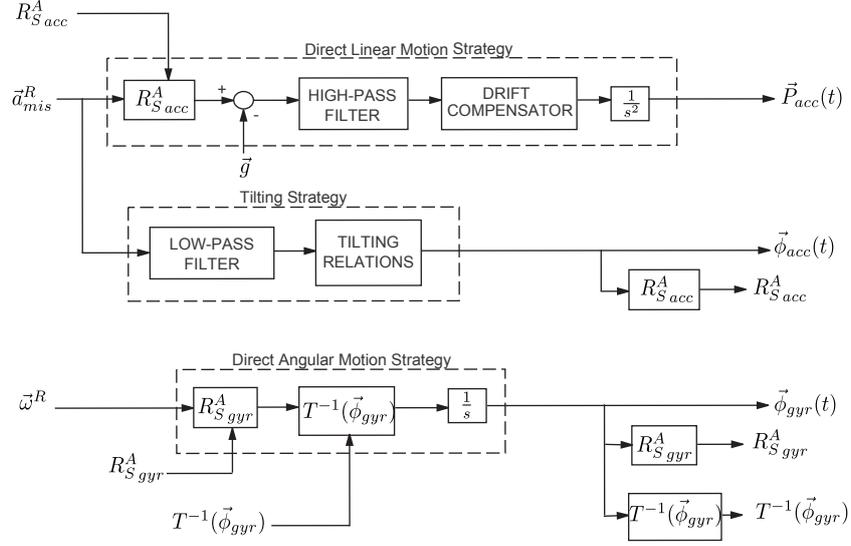


Figure 2: Design of Washout Algorithm Adapted for Inertial Sensors

Angular velocity ω^R is processed in an independent channel. With respect to Classical Washout Filter the high-pass filter does not need, so that the simulator is able to reproduce components for each frequency. *Direct Angular Motion Strategy* allows calculating $\vec{\phi}_{gyr}(t)$ which is used to calculate $R_{S_{gyr}}^A$, the rotation matrix from (*R-frame*) to (*A-frame*) and $T^{-1}(\vec{\phi}_{gyr})$, the matrix which allows calculating the derivatives of the Euler angles from the angular rates. $\vec{\phi}_{gyr}(t)$ is also the input for the inverse kinematic algorithm used to calculate joint coordinates $\vec{q}_{gyr}(t)$ during the testing of the gyroscopes.

3 Kinematic Control of Robot

The outputs of the Washout Filter are the desired trajectories in the Cartesian space which, subsequently, are converted in joint coordinates, because it is implemented a Closed Loop Iterative Kinematic (CLIK) [5]. In particular two different control algorithms are shown: one based on the quaternion and one with the optimization of the index of manipulability.

3.1 Quaternion-based Control

The differential kinematic relation is inverted using the pseudoinverse matrix of the geometric Jacobian $J^* = J^T(JJ^T + k^2I)^{-1}$:

$$\dot{\vec{q}} = J^* \begin{bmatrix} \vec{P}_d + K_p \vec{e}_p \\ \vec{\omega}_d + K_o \vec{e}_o \end{bmatrix} \quad (3)$$

where \vec{e}_p is the error of position, defined as the difference between the reference obtained with WFs and the effective position of the end-effector ($\vec{P}_d - \vec{P}_e$), K_p and K_o are gain matrices, J is the Geometric Jacobian and \vec{e}_o is the error of orientation, based on the quaternion and defined as:

$$\vec{e}_o = \eta_e \vec{e}_d - \eta_d \vec{e}_e - S(\vec{e}_d) \vec{e}_e \quad (4)$$

where $\vec{Q}_d = [\eta_d, \vec{\epsilon}_d]^T$ is the desired orientation of the end-effector and $\vec{Q}_e = [\eta_e, \vec{\epsilon}_e]^T$ is the effective one, obtained through direct kinematic. The evolution of the position error is described by the first order differential equation:

$$\dot{\vec{e}}_p + K_p \vec{e}_p = 0 \quad (5)$$

which ensures the convergence when the damping of J^* is null and the gain K_p is positive. The error of the orientation satisfies the non linear equation:

$$\vec{\omega}_d - \vec{\omega}_e + K_o \vec{e}_o = 0 \quad (6)$$

Thanks to the Lyapunov direct method it is possible to demonstrate that the system is asintotically stable in the origin: in fact, defining V as:

$$V = (\eta_d - \eta_e)^2 + (\vec{\epsilon}_d - \vec{\epsilon}_e)^T (\vec{\epsilon}_d - \vec{\epsilon}_e) \quad (7)$$

differentiating with respect to time:

$$\dot{V} = 2(\eta_d - \eta_e)(\dot{\eta}_d - \dot{\eta}_e) + 2(\vec{\epsilon}_d - \vec{\epsilon}_e)^T (\dot{\vec{\epsilon}}_d - \dot{\vec{\epsilon}}_e) \quad (8)$$

and substituing propagation formulas of the quaternion and Eq. 6, we obtain:

$$\dot{V} = -\vec{e}_o^T K_o \vec{e}_o \quad (9)$$

which is negative if the eigenvalues of K_o are all positive.

The algorithm based on the quaternion has good performances only when the robot is far from kinematic singularities. In this case, in fact, the proof of stability is not more valid and the tracking error diverges. Since the version of Classical Washout Filters adapted for the testing of inertial sensors need 5 DOF, the robot has 1 DOF redundant (the yaw angle in the accelerometer case, the origin of the tool frame in the gyroscopic case) which can be used to keep the robot far from kinematic singularities. The control algorithm is modified to maximize locally an objective function, that is the index of manipulability. All the considerations done for the case of accelerometers are valid also for the gyroscopic case.

3.2 Iterative Control with optimization of manipulability index

Taking into consideration the accelerometer case, in order to handle the redundancy of the manipulator, it is possible to define the vector input of the inverse kinematic algorithm as it follows:

$$\begin{aligned} \vec{x}_d(t) &= [\vec{P}_d(t), \phi(t), \theta(t)] \\ \dot{\vec{x}}_d(t) &= [\dot{\vec{P}}_d(t), \dot{\phi}(t), \dot{\theta}(t)] \end{aligned} \quad (10)$$

without any requirements on the yaw angle, in order that the 5x6 analytic jacobian has a kernel, $\mathcal{N}(J_A) \neq \{\vec{0}\}$. In this we can sum an element of the kernel of the jacobian to the ‘‘pseudoinverse’’ solution:

$$\dot{\vec{q}} = J^* (\dot{\vec{x}}_d + K \vec{e}) + P \vec{q}_A \quad (11)$$

where P is a projector in the kernel of the jacobian. The new term, although it has not effects on the attitude of the end-effector, changes the configuration of the manipulator, producing some eigenmotions. The objective is to design \vec{q}_A in order to maximize locally an objective function, that is the index of manipulability:

$$g(\vec{q}) = \sqrt{\det(J(\vec{q})J^T(\vec{q}))} \quad (12)$$

that is:

$$\dot{\vec{q}}_A = k \left(\frac{\partial g(\vec{q})}{\partial \vec{q}} \right)^T \quad (13)$$

The gradient of $g(\vec{q})$ must be calculated numerically. So, fixing a virtual increment of the independent variable, it is possible to calculate the discrete gradient:

$$\left(\frac{\partial g(\vec{q})}{\partial \vec{q}} \right)^T \approx \left[\frac{g \left(\vec{q} + \begin{bmatrix} \Delta q \\ \vdots \\ 0 \end{bmatrix} \right)^T - g(\vec{q})}{\Delta q}, \dots, \frac{g \left(\vec{q} + \begin{bmatrix} \Delta q \\ \vdots \\ 0 \end{bmatrix} \right)^T - g(\vec{q})}{\Delta q} \right]^T \quad (14)$$

Once the $\dot{\vec{q}}$ has been obtained, we need to simulate the kinematics of the robot, in order to estimate the accelerations and the angular rates which could be measured by the sensors mounted on the end-effector of the manipulator. The equations which describe this are:

- For the accelerations:

$$\vec{a}_{mis}^S = R_A^S \left(\frac{d}{dt} (J_p \dot{\vec{q}}) - \vec{g}^A \right) \quad (15)$$

- For the angular rates:

$$\vec{\omega}_{mis}^S = R_A^S (J_p \dot{\vec{q}}) \quad (16)$$

4 Testing

The development and calibration of the innovative odometry algorithms involves the availability of coherent kinematical inputs (wheel angular speed, acceleration and angular rates) and the simulation of a wide range of working conditions, whose realization by means of experimental test runs is difficult and expensive. On the other hand, for this type of application, the use of commercial multibody softwares is quite difficult, since the simulation of degraded adhesion conditions involves the co-simulation of on-board mechatronic devices, such as WSP, traction system, etc.. In order to overcome all these problems a complete three-dimensional multibody model of a railway vehicle has been developed using Matlab-Simulink [2, 3], which is able to reproduce different working conditions, with arbitrary tracks, including ones which may stress the behavior of the sensors, in order to highlight their critical aspects.

The multibody model of the railway vehicle can be used as the Dynamic Simulator which provides the inputs to the the Washout Filter Algorithm.

In order to test the efficiency of the WF algorithm has been implemented in Matlab-Simulink a system which includes:

- import of kinematics data from the Dynamic Simulator,
- implementation of the Washout Filters,
- implementation of the CLIK,
- direct kinematics of the manipulator,
- derivation of accelerations and angular rates

The robot modelled is an anthropomorphic manipulator with 6 DOF and a spherical wrist RPR.

5 Results

With respect to the wide set of scenarios simulated with a huge number of working conditions, we report the results for two simulations with the system of control described in the Sect. 4. The tests show the effective validity of the procedure proposed: in fact we can say that, both for the accelerations and for the angular rates, the tracking of reference values has a good behavior. The bandwidth of the filters of the *Direct Linear Motion Strategy* and of *Tilting Strategy* has been chosen partially overlapped: for the longitudinal and lateral accelerations the bandwidth of the high-pass filter is $B_{HP} = [5, +\infty)$, while the one of the low-pass filter is $B_{LP} = [5, 40)$.

First Test Run First simulation is a braking starting from 100 km/h with degraded adhesion. The track is curved and downhill. Fig. 3,4.

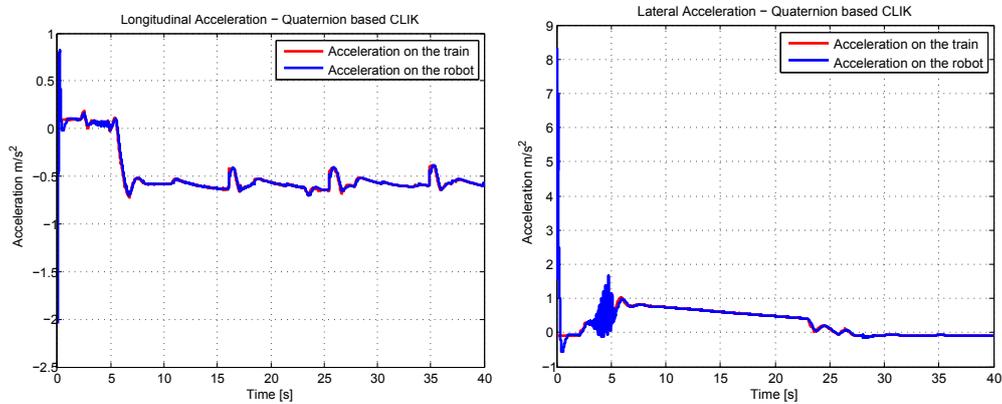


Figure 3: Longitudinal and Lateral Accelerations - First Test Run

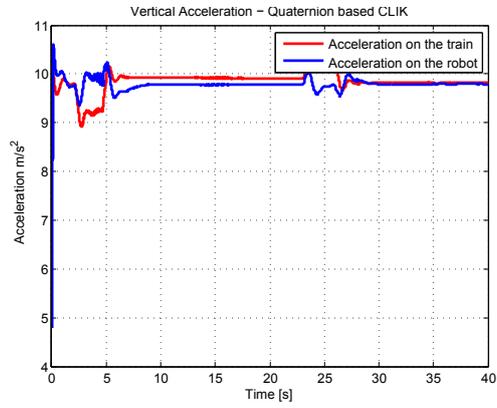


Figure 4: Vertical Acceleration - First Test Run

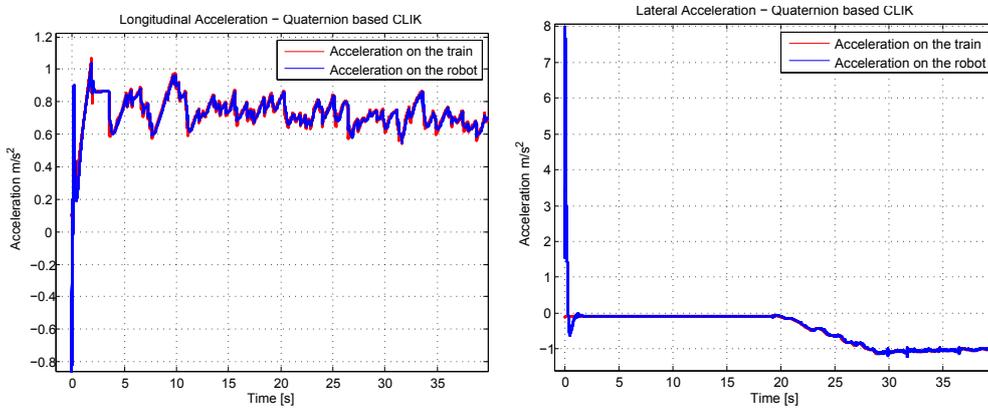


Figure 5: Longitudinal and Lateral Accelerations - Second Test Run

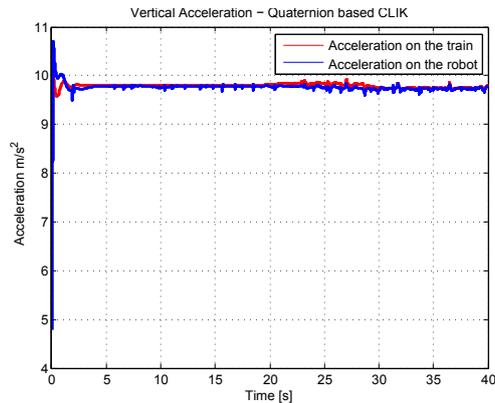


Figure 6: Vertical Acceleration - Second Test Run

Second Test Run The second simulation is a traction, starting from rest to 200 km/h in degraded conditions of adhesion. The track is curved and uphill. Fig. 5,6.

We note that, for both runs, the reproduction of the vertical acceleration is affected by error: this is predictable since *Tilting Strategy* does not allow reproducing the components of acceleration along this axis. The fact that high frequency components (over 50 Hz) are not exactly reproduced is not a problem, because the inertia of the joints should cut the frequency at lower values (10 Hz). Since the absence of filters in the test for gyroscopes, there are not any errors in the reproduction of angular rate.

6 Conclusions

The wide number of tests done in a simulated environment, realized with a huge number of working conditions and track profiles, point out that the procedure of testing through HIL test rig, that is a control compound of Washou Filters and an inverse iterative kinematic algorithm of the first order with optimization of the index of manipulability, guarantees the possibility to reply the most relevant frequencies of the signals of acceleration and angular rate until 10 Hz. So it could be used to test the innovative odometry algorithm based on inertial measures. Further activities will be carried out in order to implement the testing procedure described in this paper with the anthropomorphic manipulator Comau SMART SiX 6-1.4 with the Comau C4G-Open controller and the inertial sensor MTI-G of xSens.

References

- [1] Grant, P.R., Reid, L.D., Motion Washout Filter Tuning: Rules and Requirements, *Journal of Aircraft*, Vol.34, No.2, pp. 145-151, March-April 1997 (1997).
- [2] Meli, E., Malvezzi, M., Papini, S., Pugi, L., Rinchi, M., Rindi, A., A railway vehicle multibody model for real-time applications, *Vehicle System Dynamics*, Vol. 46, Issue 12, pp. 1083-1105, 2008 (2008).
- [3] Allotta, B., Pugi, L., Malvezzi, M., Vettori, G. and Ridolfi, A., "A multibody model of a railway vehicle for the development of innovative odometry systems," in *Proc. of Eccomas Multibody Dynamics 2011*, Brussels, Belgium, 4-7 July 2011 (2011).

- [4] Nahon, M.A., Reid, L.D., Simulator Motion-Drive Algorithms: A Designer's Perspective, Journal of Aircraft, Vol.13, No.2, pp. 356-362, March-April 1990.
- [5] Siciliano, B. Kinematic Control of Redundant Robot Manipulators: a Tutorial, Journal of Intelligent and Robotic Systems, Volume 3, Number 3, pp. 201-212, 1990.