**Presented as late breaking results contribution**
**21st IFAC World Congress (Virtual)**
**Berlin, Germany, July 12-17, 2020**

# Control of a Manipulator Mounted on an Independent Controlled Moving Platform

**Carlos Eduardo Pedroso de Oliveira** * **Walter Fetter Lages** *
**Renato Ventura Bayan Henriques** *

* *Departamento de Sistemas Elétricos de Automação e Energia, Universidade Federal do Rio Grande do Sul, Porto Alegre, RS 90035-190 Brazil (e-mails: carlospeoliveira@gmail.com, fetter@ece.ufrgs.br, rventura@ece.ufrgs.br)*

**Abstract:** This paper deals with the control of a manipulator robot mounted on a moving platform. It is supposed that the moving platform is independently controlled, therefore sensor and control signals for the platform are not available. As the motion of the platform produces perturbations on the robot motion, it is the purpose of the proposed controllers to compensate for those effects on the end-effector. A 9-axis inertial measurement unit (IMU) mounted on the top of the moving platform is used to recover the orientation, velocity and acceleration of the platform joints, which are required for the implementation of the proposed control laws. The modeling of the IMU to recover the platform motion signals is based on the platform Jacobian and the proposed controllers are based on the computed torque controller. The IMU signal processing and the controllers are implemented using the Robot Operating System and the ros_control framework, making it possible to run the controller in real-time. Simulated results using the Gazebo simulator shows the effectiveness of the proposed controller. The performance improvements with respect to a computed torque controller without compensation for the platform motion were evaluated by using the RMS error of the joints as a metric.

*Keywords:* Mechatronic systems, Motion control systems, Robot manipulators, Modeling, Information and sensor fusion, Vibration control

## 1. INTRODUCTION

Robotic manipulators mounted on non-inertial bases such as ships, offshore platforms or vehicles moving on uneven terrain present interactions between the dynamics of the manipulator and the base (Sadraei and Moghaddam, 2015). The efforts generated on the manipulator due to the base motion may compromise the robot performance.

To control a manipulator mounted on a moving base, the controller should take into account the base motion and compute a control output that compensates for the base induced disturbances on the manipulator end-effector. Assuming that the information about the platform motion is not available, a sensor could be used to estimate the platform state. Dunnigan and Wronka (2011) presents a comparison of modeling and control techniques using an accelerometer to compensate partially for modeled base motion. An inertial measurement unit (IMU) is capable of providing angular rates, accelerations and by using data fusion procedures (Lages and Henriques, 2019), the orientation of a body. Hence, in this paper the IMU data is used by the controller to compensate for the effects of the modeled platform motion on the manipulator end-effector.

The subsequent sections of this paper deal with the control of a robotic manipulator mounted on a non-inertial base. It will be assumed that the motion of the robotic arm do not affect the motion of the platform. The objective is to stabilize the robot while moving its end-effector to a target pose with respect to the platform. It is assumed that the robot links are rigid enough so that the end-effector position can be determined from the joint positions. The controller is implemented in the Robot Operating System (ROS) (Quigley et al., 2009) and results are obtained using the the Gazebo simulator (Koenig and Howard, 2004). The robot used is the 7 degrees of freedom (DoF) Barrett WAM robotic arm (Barrett, 2011) and the IMU is a Bosch BNO055 (Bosch Sensortec, 2016).

## 2. MODEL OF A ROBOT MANIPULATOR MOUNTED ON A NON-INERTIAL BASE

For a manipulator on a moving platform, Wronka (2010) obtained the robot dynamic model using the Lagrange-Euler approach by supposing that the manipulator motion has no influence on the platform, which would be the case when the base is much heavier than the manipulator. This model can be expressed as:

$$\tau = M(q)\ddot{q} + V(q,\dot{q}) + G_p(q,q_p) + F_p(q_p,\dot{q}_p,\ddot{q}_p) + V_p(q_p,\dot{q}_p)\dot{q} \quad (1)$$

where $\tau$ is the $n \times 1$ vector of joint torques, $q, \dot{q}, \ddot{q}$ are the $n \times 1$ joint position, velocity and acceleration vectors respectively, $M(q)$ is the $n \times n$ inertia matrix, $V(q,\dot{q})$ is the $n \times 1$ vector of Coriolis and centrifugal forces, $q_p$ is the vector of platform generalized coordinates, $G_p(q,q_p)$ is the gravity vector with platform influence, $F_p(q_p,\dot{q}_p,\ddot{q}_p)$ are the inertial, Coriolis and centrifugal forces vector induced on the manipulator by platform motion and $V_p(q_p,\dot{q}_p)$ are

Presented as late breaking results contribution
21st IFAC World Congress (Virtual)
Berlin, Germany, July 12-17, 2020

the Coriolis and centrifugal forces vector generated by the platform and manipulator motion. Note that $M(q)$ and $V(q, \dot{q})$ depend only on the robot motion an do not depend on the platform motion.

## 3. COMPUTED TORQUE CONTROLLER FOR A ROBOT ON A PLATFORM

The main interest here is to control the joint positions. In principle, a simple position controller with high gains could solve the issue. However, high gains generally imply higher control signal which are not desirable. Furthermore, a manipulator is a highly nonlinear system, which makes it difficult to design proper gains for the whole workspace of the robot. Furthermore, for many years, some manufacturers of industrial robots have introduced manipulators that can be torque controlled (Barrett, 2011; KUKA Roboter, 2011). Thus, control laws based on the computed torque method are a feasible way to improve the performance of those robots.

The model (1) can be used to obtain a version of the computed torque control law for a manipulator mounted on a non-inertial platform given by:

$$\tau = M_n(q)\nu + V_n(q, \dot{q}) + G_{pn}(q, q_p) + F_{pn}(q_p, \dot{q}_p, \ddot{q}_p) + V_{pn}(q_p, \dot{q}_p)\dot{q} \quad (2)$$

where virtual acceleration $\nu$ can be computed as:

$$\nu = \ddot{q}_r + K_p(q_r - q) + K_d(\dot{q}_r - \dot{q}) \quad (3)$$

Thus, producing an error dynamics given by:

$$\ddot{e} + K_d\dot{e} + K_p e = 0 \quad (4)$$

A 9-axis IMU is supposed. This type of device provides acceleration, angular velocity and magnetic field measurements. Some of them include a microprocessor as well and execute a data fusion procedure to estimate the device orientation and its linear acceleration (which differs from the accelerometer measurement due to gravity). Usually an extended Kalman filter (EKF) is used to obtain such estimates (Lages and Henriques, 2019).

In an IMU sensor, acceleration, angular velocity measurements and magnetic field measurements are taken with respect to the inertial frame and represented in the body frame of the sensor (Kok et al., 2017). The estimated orientation is estimated with respect to a frame with axis aligned with the vertical and magnetic north and also represented is this frame. Here, it is assumed the ENU convention as it is the conversion used by ROS (Foote and Purvis, 2010). That is, the X axis points to the East, the Y axis points to the (magnetic) North and the Z axis points upwards.

The platform has two coincident rotational joints, that is $q_p = [q_{p1}\ q_{p2}]^T$, which produce roll and pitch motion. Figure 1 show the frames used in the development that follows. The angle around $Z_p$ is $q_{p1}$ and the angle around $Z_{p1}$ is $q_{p2}$.

As $q_p$ should be obtained from the IMU orientation estimation, it is necessary to consider the orientation of the fixed base of the platform with respect to the inertial frame ${}^0R_p$, the orientation of the top of the platform with respect to its base ${}^pR_t(q_p)$, the orientation of the IMU with respect

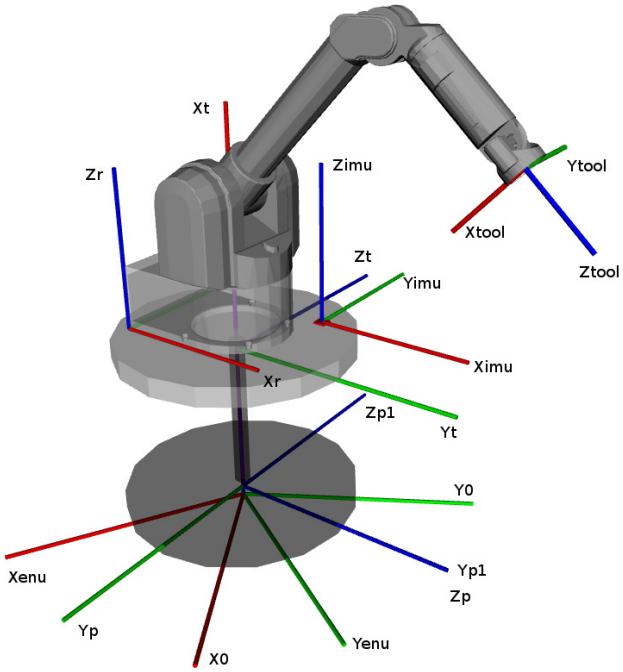


Fig. 1. Platform frame assignment.

to the top of the platform ${}^tR_{imu}$, the orientation of the IMU with respect to the ENU frame ${}^{enu}R_{imu}$ and the orientation of the ENU frame with respect to the inertial frame ${}^0R_{enu}$. Therefore:

$${}^pR_t(q_p) = {}^{enu}R_p^T\ {}^{enu}R_{imu}\ {}^tR_{imu}^T \quad (5)$$

Note that ${}^{enu}R_p$ and ${}^tR_{imu}$ are constants and that ${}^{enu}R_{imu}$ is the orientation estimation obtained from the IMU. Once ${}^pR_t(q_p)$ is computed from (5), the joint angles of the platform can be obtained from:

$$q_p = \begin{bmatrix} \text{atan2}(r_{13}, -r_{23}) \\ \text{atan2}(r_{31}, r_{32}) \end{bmatrix} \quad (6)$$

where $r_{ij}$ is the element of the i-th line and j-th column of ${}^pR_t$.

The joint velocities for the platform can be obtained from the angular velocities measured by the IMU, ${}^{imu}\omega_{imu}$. However, the measurements of the IMU are represented in its own frame (Kok et al., 2017) and can be transformed to the platform frame by:

$${}^p\omega_t = {}^pR_t(q_p)\ {}^tR_{imu}\ {}^{imu}\omega_{imu}$$

where ${}^p\omega_t$ is the angular velocity of the platform top represented in the platform base frame.

On the other hand, the angular velocity of the top of the platform can be computed from the velocities of the platform joints by:

$${}^p\omega_t = J_\omega(q_p)\dot{q}_p$$

where $J_\omega(q_p)$ is the Jacobian relating the platform joint velocities with the angular velocity of the platform top.

Therefore, by using the pseudo-inverse of $J_\omega(q_p)$, the platform joint velocities can be computed from the angular velocity measured by the IMU as:

Presented as late breaking results contribution
21st IFAC World Congress (Virtual)
Berlin, Germany, July 12-17, 2020

$$\dot{q}_p = J_\omega^\dagger(q_p)\ {}^pR_t(q_p)\ {}^tR_{imu}\ {}^{imu}\omega_{imu}$$
$$= \begin{bmatrix} \cos q_{p2} & 0 & \sin q_{p2} \\ 0 & 1 & 0 \end{bmatrix}\ {}^{imu}\omega_{imu} \quad (7)$$

Similarly, the platform joint accelerations can be computed from the acceleration measured by the IMU, ${}^{imu}a_{imu}$. It is important to note that the acceleration measured by the IMU includes the gravity, which should be removed to compute the linear acceleration of the top of the platform:

$${}^pa_{imu} = {}^pR_t(q_p)\ {}^tR_{imu}\left({}^{imu}a_{imu} - {}^0R_{imu}^T g\right) \quad (8)$$

where ${}^pa_{imu}$ is the acceleration of the IMU mounting point on the top of the platform represented in the platform base frame and $g = [0\ 0\ 9.81]^T$ is the gravity vector.

The linear velocity of the IMU mounting point can be computed from the platform joint velocities as:

$${}^pv_{imu} = J_v(q_p)\dot{q}_p \quad (9)$$

where $J_v(q_p)$ is the Jacobian relating the platform joint velocities with the linear velocity of the IMU mounting point. Hence (Fu et al., 1987):

$$J_{v1}(q_p) = {}^p\hat{Z}_p \times \left({}^pR_1\left({}^1P_t + {}^1R_t\,{}^tP_{imu}\right)\right)$$
$$J_{v2}(q_p) = {}^pR_1\left({}^1\hat{Z}_1 \times \left({}^1P_t + {}^1R_t\,{}^tP_{imu}\right)\right)$$

where the origin of the frame at the top of the platform relative to the frame at the platform base is given by ${}^pP_t$ and the origin of the IMU frame with respect to the frame at the top of the platform is given by ${}^tP_{imu}$.

By differentiating (9) with respect to time, the linear acceleration of the IMU mounting point is:

$${}^pa_{imu} = \frac{d}{dt}\left({}^pv_{imu}\right) = \dot{J}_v(q_p,\dot{q}_p)\dot{q}_p + J_v(q_p)\ddot{q}_p \quad (10)$$

with (Fu et al., 1987):

$$\dot{J}_{v1}(q_p) = \left({}^p\hat{Z}_p \times \left({}^p\hat{Z}_p \times \left({}^pR_1\left({}^1P_t + {}^1R_t\,{}^tP_{imu}\right)\right)\right)\right)\dot{q}_{p1}$$
$$\dot{J}_{v2}(q_p) = {}^pR_1\left({}^1\hat{Z}_1 \times \left({}^1\hat{Z}_1 \times \left({}^1P_t + {}^1R_t\,{}^tP_{imu}\right)\right)\right)\dot{q}_{p2}$$

Thus, from (8) and (10) it is possible to compute the platform joint acceleration from the IMU accelerometer measurements as:

$$\ddot{q}_p = J_v^\dagger(q_p)\left({}^pR_t(q_p)\ {}^tR_{imu}\left({}^{imu}a_{imu} - {}^0R_{imu}^T g\right)\right. \quad (11)$$
$$\left. - \dot{J}_v(q_p,\dot{q}_p)\dot{q}_p\right) \quad (12)$$

Hence, by using the sensors on the joints of the manipulator and the platform joint positions, velocities and accelerations obtained from (6), (7) and (12), it is possible to compute the control laws defined by (2) and (3).

The controllers proposed in section 3 were implemented in ROS. Since ROS nodes do not have real-time capabilities, the `ros_control` framework (Chitta et al., 2017) provides the capability to implement real-time controllers.

## 4. SIMULATION RESULTS

To evaluate the performance of the controller under the effects of the base motion the system was tested on the simulation environment Gazebo. A step on the reference of joint position was applied to each joint. The following controllers are implemented:

**CTCNC:** Classical computed torque controller without compensation for the base motion described in (Fu et al., 1987).

**CTCPD:** Computed torque controller with compensation for the effects of the base motion on the manipulator end-effector.

For the base motion, a sinusoidal signal was applied to each of the platform joints as in Wronka and Dunnigan (2011). That is:

$$q_p(t) = \begin{bmatrix} a_{roll}\sin\left(\frac{2\pi}{T_{roll}}t + \phi_{roll}\right) \\ a_{pitch}\sin\left(\frac{2\pi}{T_{pitch}}t + \phi_{pitch}\right) \end{bmatrix} \quad (13)$$

with $a_{roll} = a_{pitch} = 0.219\,\text{rad}$, $T_{roll} = 4.5\,\text{s}$, $T_{pitch} = 2.25\,\text{s}$, $\phi_{roll} = \pi/2\,\text{rad}$ and $\phi_{pitch} = 0$.

The controller gains for the PD controller were calculated to obtain a step response without overshoot. For the system (4), with $K_i = 0$, the PD diagonal matrices of gains are chosen with $k_p = \omega_n^2$ and $k_i = 2\xi\omega_n$. With $\xi = 1$ and $\omega_n \approx \frac{4}{\xi T_s}$, where $T_s$ is the settling time for a tolerance of 2%. The gains forboth controllers were $K_p = 25$ and $K_d = 10$.

The controllers were evaluated with a reference step of 0.5 rad applied to each joint of the manipulator. Figure 2 presents the system response for the CTCNC. The reference tracking is severely compromised. As can be seen on Table 1 the total RMS error is 1.2348 rad. The controller does not reject properly the disturbances generated by the base motion.

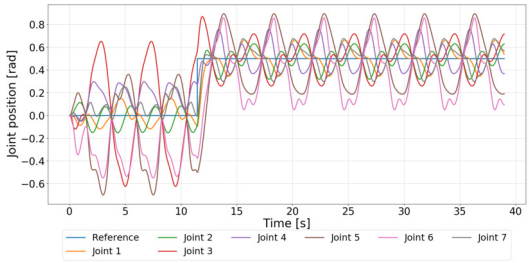


Fig. 2. Step response of the system with CTCNC.

The results for the CTCPD can be visualized in Figure 3. A considerable improvement on performance can be noted. See Table 1, where the RMS error of the manipulator joints is shown..

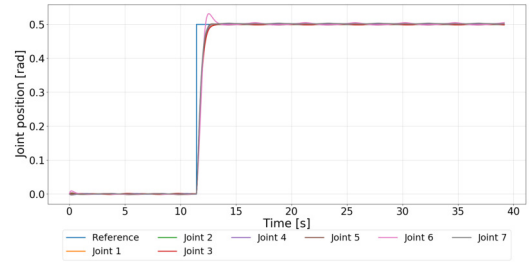


Fig. 3. Step response of the system with CTCPD.

Figures 4 and 5 present the torques for CTCNC and CTCPD, respectively. It can be noted that the controller

Table 1. RMS error for joint position [rad].

| Joint | CTCNC | CTCPD |
|---|---|---|
| 1 | 0.0955 | 0.0435 |
| 2 | 0.0973 | 0.0444 |
| 3 | 0.2246 | 0.0424 |
| 4 | 0.1362 | 0.0439 |
| 5 | 0.2782 | 0.0432 |
| 6 | 0.2880 | 0.0430 |
| 7 | 0.1150 | 0.0447 |
| Total | 1.2348 | 0.3051 |

with compensation for effects of base movement applies smaller torques to the joints..

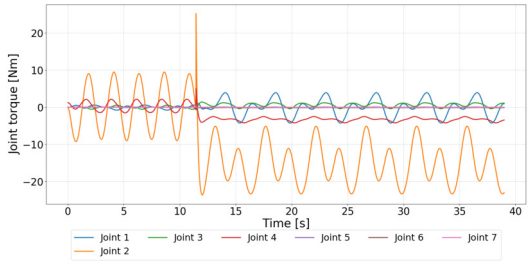


Fig. 4. CTCNC efforts.

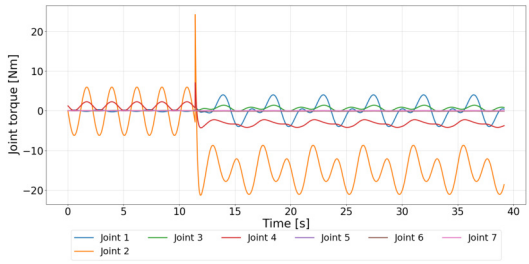


Fig. 5. CTCPD efforts.

The performance of the controllers were evaluated by using and objective metric through the

## 5. CONCLUSION

In this paper, the implementation of a controller for a robotic manipulator on a non-inertial base was presented. A controller without compensation for base movement showed a poor performance and motivates the implementation of a more specific solution to the control problem. The controller with compensation for the effects of base motion on the manipulator end-effector using an IMU sensor presented a lower RMS error, offering a significant improvement in system performance when compared to CTCNC. Future works consider the integration of he controllers implemented in this paper in the real Barrett WAM robot and a Stewart platform in order to consider platform perturbations in 6D.

## REFERENCES

Barrett (2011). *WAM User Manual*. Barrett Technology, Inc., Cambridge, MA.

Bosch Sensortec (2016). *Data Sheet BNO055: Intelligent 9-Axis Absolute Orientation Sensor*. Bosch Sensortec GmbH, Reutlingen, Germany. Document Number: BST-BNO055-DS000-14, <https://ae-bst.resource.bosch.com/media/_tech/media/datasheets/BST-BNO055-DS000.pdf>.

Chitta, S., Marder-Eppstein, E., Meeussen, W., Pradeep, V., Rodríguez Tsouroukdissian, A., Bohren, J., Coleman, D., Magyar, B., Raiola, G., Lüdtke, M., and Perdomo, E.F. (2017). ros_control: A generic and simple control framework for ROS. *The Journal of Open Source Software*. doi:10.21105/joss.00456. URL http://www.theoj.org/joss-papers/joss.00456/10.21105.joss.00456.pdf.

Dunnigan, M.W. and Wronka, C.M. (2011). Comparison of control techniques for a robotic manipulator with base disturbances. *IET Control Theory Applications*, 5(8), 999–1012. doi:10.1049/iet-cta.2010.0331.

Foote, T. and Purvis, M. (2010). Standard units of measure and coordinate conventions. URL https://www.ros.org/reps/rep-0103.html.

Fu, K.S., Gonzales, R.C., and Lee, C.S.G. (1987). *Robotics Control, Sensing, Vision and Intelligence*. Industrial Engineering Series. McGraw-Hill, New York.

Koenig, N. and Howard, A. (2004). Design and use paradigms for gazebo, an open-source multi-robot simulator. In *Proceedings of the 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2004)*, volume 3, 2149–2154. IEEE Press, Sendai, Japan. doi:10.1109/IROS.2004.1389727.

Kok, M., Hol, J.D., and Schön, T.B. (2017). Using inertial sensors for position and orientation estimation. *Foundations and Trends in Signal Processing*, 11(1-2), 1–153. doi:10.1561/2000000094.

KUKA Roboter (2011). *Lightweight Robot 4+*. KUKA Roboter GmbH, Augsburg, Germany. Version: Spez LBR 4+ V2 en.

Lages, W.F. and Henriques, R.V.B. (2019). Performance evaluation of data fusion for orientation estimation in an intelligent inertial measurement unit. In *Proceedings of the 2019 Latin American Robotics Symposium (LARS), 2019 Brazilian Symposium on Robotics (SBR) and 2019 Workshop on Robotics in Education (WRE)*, 150–155. IEEE, Rio Grande, RS, Brazil. doi:10.1109/LARS-SBR-WRE48964.2019.00034.

Quigley, M., Gerkey, B., Conley, K., Faust, J., Foote, T., Leibs, J., Berger, E., Wheeler, R., and Ng, A. (2009). ROS: an open-source robot operating system. In *Proceedings of the IEEE International Conference on Robotics and Automation, Workshop on Open Source Robotics*. IEEE Press, Kobe, Japan.

Sadraei, E. and Moghaddam, M. (2015). On a moving base robotic manipulator dynamics. *International Journal of Robotics, Theory and Applications*, 4(3), 66–74. URL http://ijr.kntu.ac.ir/article_13764.html.

Wronka, C.M. (2010). *Modelling and Control of a Robotic Manipulator Subject to Base Disturbances*. Ph.D. thesis (electrical, electronic and computer engineering), Heriot-Watt University, Edinburgh.

Wronka, C.M. and Dunnigan, M.W. (2011). Derivation and analysis of a dynamic model of a robotic manipulator on a moving base. *Robotics and Autonomous Systems*, 59(10), 758–769. doi:https://doi.org/10.1016/j.robot.2011.05.010.