



Stability-based planning and trajectory tracking of a mobile manipulator over uneven terrains

Jae-Yun Jun, Vincent Padois, Faïz Ben Amar

► To cite this version:

Jae-Yun Jun, Vincent Padois, Faïz Ben Amar. Stability-based planning and trajectory tracking of a mobile manipulator over uneven terrains. IEEE International Workshop on Advanced Robotics and its Social Impacts, 2015, Lyon, France. hal-01160230

HAL Id: hal-01160230

<https://hal.science/hal-01160230>

Submitted on 9 Jun 2015

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Stability-based planning and trajectory tracking of a mobile manipulator over uneven terrains

Jae-Yun Jun, Vincent Padois and Faïz Benamar

Abstract—The problem of improving the stability of a mobile manipulator over a sloped terrain is addressed in the present work. Such an improvement is achieved by finding the location of the manipulator’s center of mass that maximizes the overall quasi-static stability, defined here as the force-angle stability, using a stochastic optimization approach known as the Covariance Matrix Adaptation. The tracking of both trajectories for the robot base and for the manipulator is achieved by using an inverse-kinematics controller in simulation.

I. INTRODUCTION

A mobile manipulator is suited for transporting objects from a place to another while avoiding obstacles, as well as for achieving other tasks such as opening doors, tracking objects with a camera mounted on the manipulator, and more. When traversing uneven terrains, such a robot becomes more sensitive to the stability condition, while it also needs to satisfy other constraints such as joint limits and motor speed-torque limits.

Most of existing works on mobile manipulators focus on planning paths (trajectories) and on controlling their motion on level surfaces, while satisfying the aforementioned constraints and guaranteeing the *stability degree* [1], *manipulability* [2] or *task compatibility* [3]. In addition, they assume that the path (trajectory) for the end-effector of the manipulator is given [4], [5].

The present work introduces a method to generate a trajectory for the arm’s¹ center of mass for improving the robot’s overall stability and to track this trajectory using the inverse velocity kinematics of the system, when a trajectory is given for the robot base. When the manipulator’s mass is non-negligible with respect to the robot-base’s mass, then the degrees of freedom that the manipulator brings may be used to increase the robot’s overall stability. The search for the trajectory (for the arm’s center of mass) that optimizes the robot’s overall stability is implemented using a stochastic optimization algorithm called the *Covariance Matrix Adaptation (CMA)* algorithm [6]. The solution-search space is non-convex, and this algorithm can reliably find solutions for this type of problems. As far as the stability definition, the force-angle stability measure (proposed in [7]) is used.

The manipulator employed in the present work has five degrees of freedom. But, because changes of the gripper’s



Fig. 1. A Cameleon EOD, a tracked mobile robot from Eca Robotics [13]

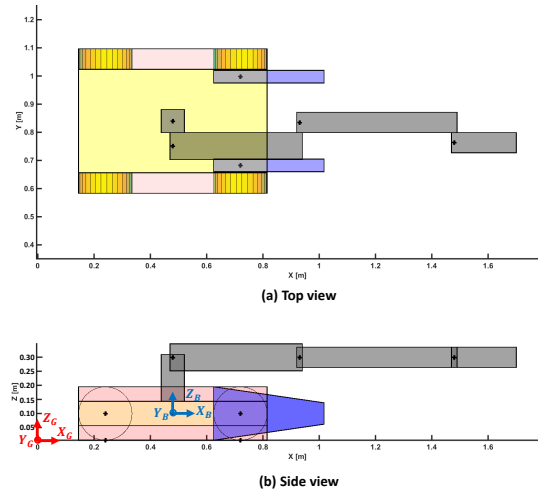


Fig. 2. Tracked mobile robot model: (a) top view, and (b) side view.

configuration do not significantly influence on changes of the location of the arm’s center of mass, the associated degree of freedom is not considered in the present work. Hence, the manipulator is assumed to have four degrees of freedom, and, therefore, it is a redundant system with respect to the control task. The redundancy problem is typically resolved either by kinematic control [8], optimization-based control [9]–[11] or nonlinear optimization methods [12]. In this work, we resolve the redundancy problem by finding joint angular velocities that minimize the error between the desired task-space velocities and the actual ones. In this manner, the mobile manipulator can track both the trajectory for the robot base and the other for the arm’s center of mass.

The presentation of this work is structured as follows. In Section II, a method to generate a trajectory for the arm’s center of mass for improving the robot’s overall stability and to track this trajectory using an inverse-kinematics controller

Jae-Yun Jun, Vincent Padois and Faïz Benamar are with:
1. Sorbonne Universités, UPMC Univ Paris 06, UMR 7222,
ISIR, F-75005, Paris, France

2. CNRS, UMR 7222, ISIR, F-75005, Paris, France
e-mail: jaeyunjk@gmail.com, padois@isir.upmc.fr, amar@isir.upmc.fr

¹The terms ‘manipulator’ and ‘arm’ are used interchangeably throughout the presentation of this work.

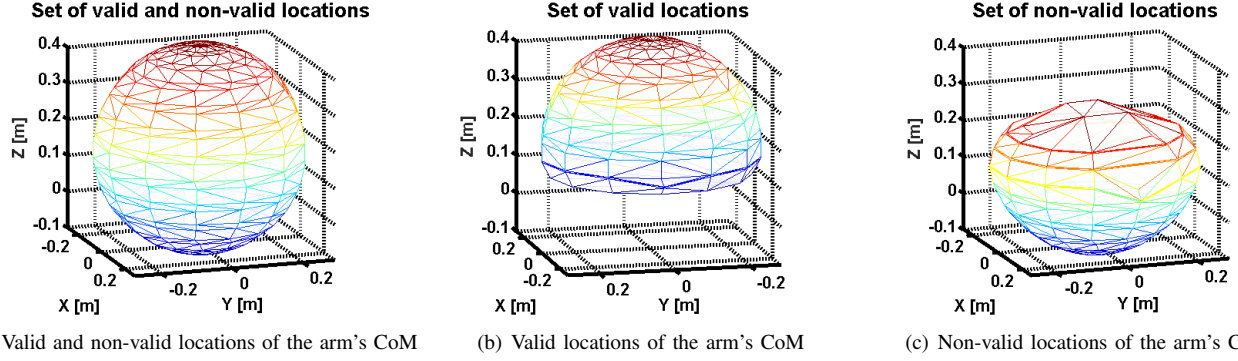


Fig. 3. With the purpose to define the workspace of the arm's center of mass (CoM), a large number of locations of the arm's center of mass are generated by considering all possible combinations of discretized arm joint angle values. These locations are then labeled as valid or non-valid ones based on the self-collision checking results.

is described. In Section III, the results of implementing this method for a mobile manipulator traversing a slope are presented. Finally, a summary of this work and some guidance for future work are given in Section IV.

II. METHODS

A. Characterization of the workspace of the arm's center of mass

The mobile manipulator considered in the present work is modeled after the robot shown in Fig. 1, and this model is shown in Fig. 2. The robot base consists of a tracked mobile robot with two tracks and two flippers. The two tracks are actuated with two DC motors, and the flippers are coupled and driven by one DC motor. The flippers are only considered in the present study as collidable bodies, and they are not actuated. On the other hand, the manipulator has five degrees of freedom with four links and one gripper. Because changes of the gripper configuration do not significantly influence on changes of the location of the arm's center of mass, the gripper is not considered in this study. Hence, the manipulator is assumed to have four degrees of freedom.

To represent the robot configuration, a coordinate frame is attached to the center of the base, representing the body frame. The arm links are represented using the Denavit Hartenberg's parameters [14], and a coordinate system is attached to each joint, with the purpose to represent the center of mass of each link.

The arm's center of mass can be expressed (in global frame ($\mathcal{G}$)) as a function of each arm link's center of mass as

$${}^{\mathcal{G}}\bar{\xi}_A = \frac{{}^{\mathcal{G}}\mathbf{T}_B \sum_{i=0}^3 m_i {}^{\mathcal{B}}\mathbf{T}_{\mathcal{R}_i} \bar{\xi}_i}{\sum_{i=0}^3 m_i}, \quad (1)$$

where ${}^{\mathcal{G}}\bar{\xi}_A$ is the arm's center of mass in the global frame (see Fig. 2(b)), ${}^{\mathcal{G}}\mathbf{T}_B$ is the homogeneous transformation from the body frame ($\mathcal{B}$) to the global frame ($\mathcal{G}$), ${}^{\mathcal{B}}\mathbf{T}_{\mathcal{R}_i}$ is the homogeneous transformation from the i -th manipulator link frame ($\mathcal{R}_i$) to the body frame, $\bar{\xi}_i$ is the center of mass of the i -th manipulator link expressed in the i -th manipulator

link frame ($\mathcal{R}_i$), and m_i is the mass of the i -th manipulator link.

The search for the optimal location of the arm's center of mass can be efficient if one defines the workspace of the arm's center of mass in body frame using the brute force method offline. That is, the four arm joint angles are discretized, and the locations of the arm's center of mass corresponding to all possible combinations of these discretized joints angles are recorded. Afterwards, each of these locations is labeled as valid or invalid configuration depending on the result of self-collision checking. In this work, the self-collision is checked using the *Gilbert-Johnson-Keerthi* (GJK) algorithm [15].

Fig. 3 shows the results obtained by implementing this brute force method. Fig. 3(a) shows the set of all the locations of the arm's center of mass (regardless of their corresponding labels), denoted by S_O . On the other hand, Fig. 3(b) and Fig. 3(c) show the sets of valid and non-valid locations and denoted by S_V and S_N , respectively. Both S_O and S_N are convex sets, while S_V is a non-convex set.

In fact, S_V is the workspace of the arm's center of mass. The workspace is non-convex due to the geometric constraints of the manipulator and due to collisions that occur between the manipulator links, as well as between the manipulator and the base platform. But, one can express this workspace as the set subtraction of S_N from S_O . It is preferable to express the workspace in this way because both S_O and S_N are convex sets, and, therefore, the identification procedure of whether a point is within the workspace of the arm's center of mass can be quickly evaluated.

B. Relationship between the tip-over stability and the location of the arm's center of mass

In this work, the optimality criterion employed for searching the desired location of the arm's center of mass is the tip-over stability measure. In particular, the force-angle stability measure [7] is used. The force-angle stability measure can be computed as the product between the magnitude of the net force applied to the overall center of mass and the smallest angle obtained among all the angles formed between the tip-over-axis normals ($\mathbf{l}_j$'s) and the net force vector ($\mathbf{f}_r$), where

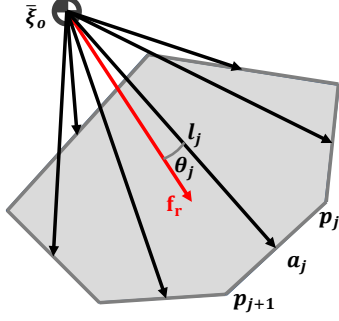


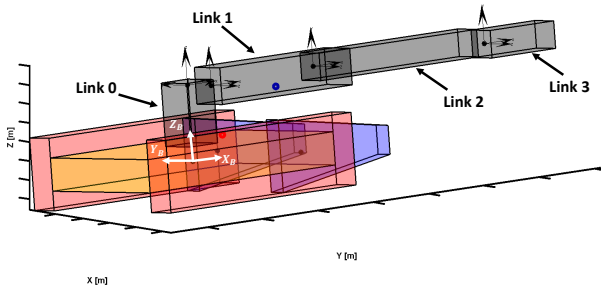
Fig. 4. The robot's tip-over stability can be defined as the force-angle stability measure [7]. The tip-over axes ($\mathbf{a}_j$'s) are the vectors that define the support polygon boundary, and the force-angle stability measure can be computed as the product between the magnitude of the net force and the smallest angle obtained among all the angles formed between the tip-over-axis normals ($\mathbf{l}_j$'s) and the net force vector ($\mathbf{f}_r$). $\bar{\xi}_O$ is the location of the overall center of mass and is defined in (4).

the tip-over axes ($\mathbf{a}_j$'s) are vectors that define the support polygon boundary. Therefore, the stability measure can be defined as

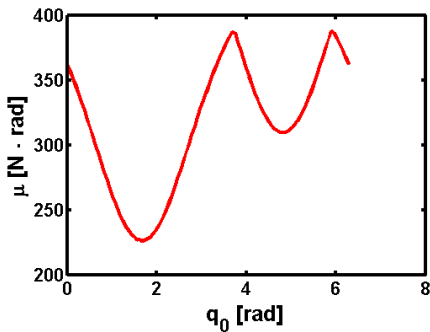
$$\mu = \min_j (\theta_j) \|\mathbf{f}_r\|, \quad j = \{1, \dots, n\}, \quad (2)$$

where n is the total number of tip-over axes that define the support polygon, $\mathbf{f}_r$ is the resulting force vector, $\mathbf{a}_j$ is the j -th tip-over axis, $\mathbf{l}_j$ is the vector normal to the j -th tip-over axis ($\mathbf{a}_j$), and θ_j is the angle formed between $\mathbf{f}_r$ and $\mathbf{l}_j$. See Fig. 4 for clarification.

On the other hand, one can develop (2) based on the



(a) A robot configuration with some nonzero roll value



(b) Stability measure

Fig. 5. Stability measure as the first joint angle (q_0) is rotated from 0 [deg] to 360 [deg], while the robot base has some nonzero roll value.

trigonometry to relate the stability measure to the overall center of mass as

$$\theta_j = \sigma_j \cos^{-1} \left(\mathbf{f}_r^T \Phi_j^T \Phi_j (\mathbf{p}_{j+1} - \bar{\xi}_O) \right), \quad (3)$$

where $\Phi_j = (\mathbf{I} - \hat{\mathbf{a}}_j \hat{\mathbf{a}}_j^T)$, $\hat{\mathbf{a}}_j = \mathbf{a}_j / \|\mathbf{a}_j\|$, and

$$\sigma_j = \begin{cases} 1, & (\Phi_j \mathbf{f}_r \times \hat{\mathbf{l}}_j)^T \cdot \hat{\mathbf{a}}_j > 0, \\ -1, & \text{otherwise.} \end{cases}$$

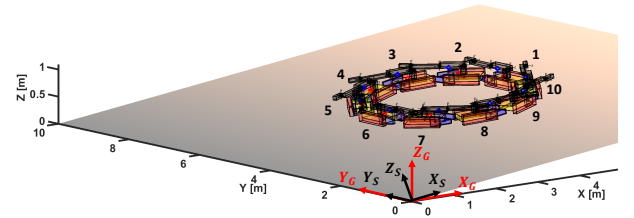
where $\hat{\mathbf{l}}_j = \mathbf{l}_j / \|\mathbf{l}_j\|$.

Finally, $\bar{\xi}_O$ in (3) can be related to $\bar{\xi}_A$ by

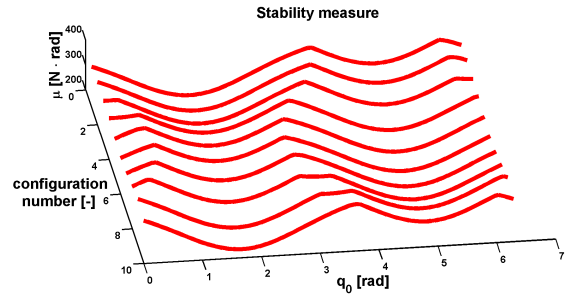
$$\bar{\xi}_O = \frac{m_A \bar{\xi}_A + m_B \bar{\xi}_B}{m_T}, \quad (4)$$

where $\bar{\xi}_O$ is the overall center of mass, $\bar{\xi}_A$ is the arm's center of mass, $\bar{\xi}_B$ is the robot-base's center of mass, $m_T = m_A + m_B$ is the total mass, m_A is the manipulator's mass, and m_B is the robot-base's mass.

To validate the stability measure, as shown in Fig. 5, some nonzero roll value is given to the robot base, and the location of the arm's center of mass is varied by rotating the arm's *Link 0* from 0 [deg] to 360 [deg]. The other arm joint angles are set to be zero. Fig. 5(a) shows the arm configuration with all the arm joint angles being zero. Fig. 5(b) shows two local maxima in stability as function of the first arm joint angle (q_0). These local maxima roughly correspond to having the manipulator directed towards the two corners of the robot side that is lifted. They are not exactly the same because the manipulator is not symmetric due to its links' volumes (geometric constraints).



(a) Ten robot poses on a slope following a circular path



(b) Stability measure values corresponding to the ten configurations

Fig. 6. Ten robot configurations are considered on a slope and following a circular path. For each robot-base configuration, the first joint angle (q_0) is rotated from 0 [deg] to 360 [deg] with a precision of $2\pi/100$ [rad], and the robot's overall stability measure is recorded.

In addition, we considered ten robot-base configurations on a slope following a circular path as shown in Fig. 6(a). For each robot-base configuration, the arm's *Link 0* is rotated 360 [deg] with a precision of $2\pi/100$ [rad], and the corresponding stability measure value is shown in Fig. 6(b). Again, all other joint angles are set to be zero. Hence, the manipulator is stretched all the time while it rotates about the vertical axis of the *Link 0*. These results show again that, for each robot-base configuration, the stability curve presents two local maxima. As explained earlier, one local maximum has larger value than the other one, due to the asymmetry that exists in the distribution of the arm links (see Fig. 5(a)).

C. Search for the optimal location of the arm's center of mass

Next, a trajectory for the arm's center of mass is searched to optimize the robot's overall stability, with a trajectory given for the robot base. Because the workspace of the arm's center of mass is non-convex, a stochastic optimization method known as the *Covariance Matrix Adaptation (CMA)* algorithm [6] is used. This algorithm consists of two main procedures: to update the mean of the distribution of successful candidate solutions (samples) towards the maximum value of a fitness function and to update the covariance matrix of this distribution such that the likelihood of these parameters increases with time.

Therefore, this optimization problem can be formulated as

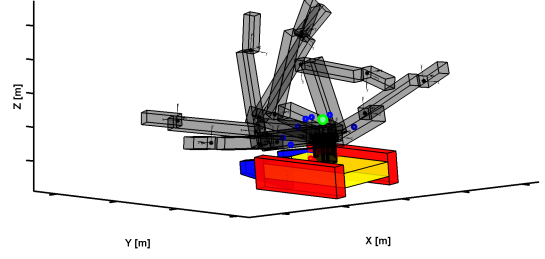
$$\begin{aligned} \max_{\bar{\xi}_A} \quad & \mathcal{F}(\bar{\xi}_A) \\ \text{s.t.} \quad & \bar{\xi}_A \in \mathbf{S}_V, \end{aligned} \quad (5)$$

where $\bar{\xi}_A$ is the arm's center of mass, $\mathbf{S}_V$ is the workspace of the arm's center of mass, and $\mathcal{F}$ is the fitness function and is defined as

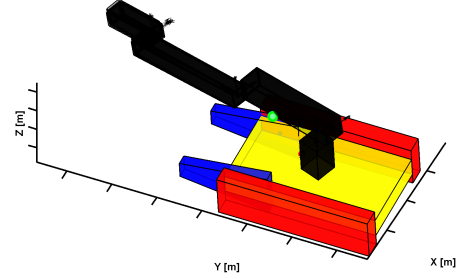
$$\mathcal{F}^{(t)} = \omega_1 \frac{\mu}{\mu_{\max}} - \omega_2 \frac{(\bar{\xi}_A^{(t)} - \bar{\xi}_A^{(t-1)})^2}{d_{\text{workspace}}^2}, \quad (6)$$

where $\omega_1 \geq 0$ and $\omega_2 \geq 0$ are weights satisfying the condition $\omega_1 + \omega_2 = 1$, μ is the stability function introduced in (2), $\bar{\xi}_A^{(t)}$ is the desired location of the arm's center of mass at time instant t , and $\bar{\xi}_A^{(t-1)}$ is the arm's center of mass corresponding to the previous time step. $\mu_{\max}$ is the maximum stability measure value, and $d_{\text{workspace}}$ is a scalar value representing the size of the workspace of the arm's center of mass (see Fig. 3). In this study, we would like to maximize the robot's overall stability, while minimizing the motion that the manipulator must make (from the current configuration) to achieve it. In the optimization procedure, if a randomly sampled configuration does not belong to the workspace of the arm's center of mass, then such a configuration is discarded. As an example, Fig. 7(a) shows an initial set of randomly chosen samples for the arm's center of mass with the robot base having some nonzero roll angle, and Fig. 7(b) shows the CMA convergence result.

Because the CMA algorithm might not be able to search for the global optima but be stuck to a local maximum,



(a) Initial distribution of arm configurations



(b) Converged arm configurations

Fig. 7. Initial and converged distribution of arm configurations after solving (5) with the CMA algorithm

for each robot-base configuration, we considered two sets of initial conditions around the two local minima. Fig. 6(b) indicates that, for all the considered cases, the desired locations of the arm's center of mass correspond to having the joint angle of the *Link 0* around 2.0 [rad] and 5.0 [rad] with the other arm joint angles being nearly zero.

D. Trajectory tracking

Once the stability-optimal trajectory for the arm's center of mass is found and having a trajectory given for the robot base, one can control the motion of both the robot base and the arm by solving the following problem

$$\begin{aligned} \min_{\dot{\mathbf{q}}} \quad & \frac{1}{2} \left\| \mathcal{S} \dot{\bar{\xi}} - \mathcal{S} \mathbf{J}(\mathbf{q}) \dot{\mathbf{q}} \right\|_{\omega_{\xi}}^2 + \frac{1}{2} \left\| \dot{\mathbf{q}} \right\|_{\omega_q}^2, \\ \text{s.t.} \quad & \mathbf{q} \leq \mathbf{q}_{\text{limit}}, \\ & \dot{\mathbf{q}} \leq \dot{\mathbf{q}}_{\text{limit}}. \end{aligned} \quad (7)$$

$\mathcal{S} \dot{\bar{\xi}}$ can be designed as

$$\mathcal{S} \dot{\bar{\xi}} = K_p (\mathcal{S} \bar{\xi} - \mathcal{S} \xi), \quad (8)$$

where $\mathcal{S} \bar{\xi} = [\bar{X}_B, \bar{Y}_B, \bar{\gamma}, \bar{X}_A, \bar{Y}_A, \bar{Z}_A]^T$ represents the desired trajectories for the robot base and the manipulator in the slope frame ($\mathcal{S}$) (see Fig. 6(a)). $\mathcal{S} \xi = [X_B, Y_B, \gamma, X_A, Y_A, Z_A]^T$ indicates both the actual robot-base configuration (X-Y coordinates of the robot-base's center of mass and its yaw angle) and the arm's center of mass (X-Y-Z coordinates). $\mathbf{q} = [q_l, q_r, q_0, q_1, q_2, q_3]^T$ corresponds to the two motor joint angles for the robot base and the four joint angles of the manipulator. $\mathbf{J}(\mathbf{q})$ is the 6×6 Jacobian that relates $\dot{\mathbf{q}}$ to $\mathcal{S} \dot{\bar{\xi}}$, $\mathbf{q}_{\text{limit}}$ is 6D joint angle limits, $\dot{\mathbf{q}}_{\text{limit}}$ is 6D joint angular speed

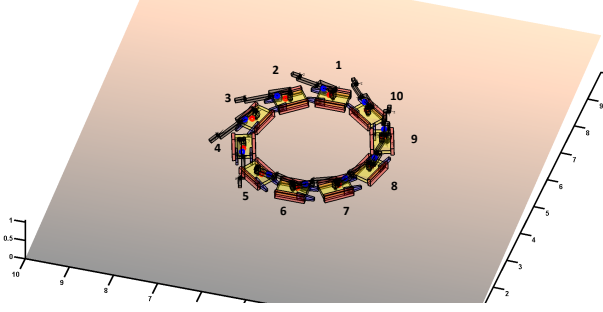


Fig. 8. Converged configurations on a slope.

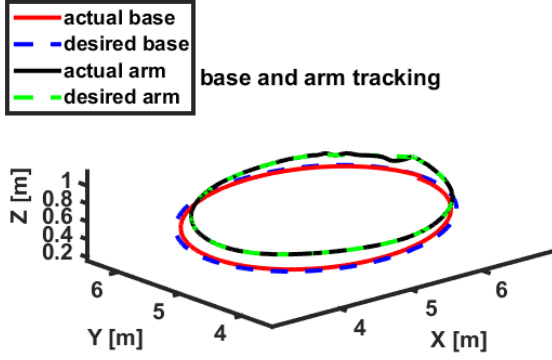


Fig. 9. Desired and actual trajectories for the robot base and the arm on a slope

limits, and $\mathbf{K}_p$ is the 6×6 proportional gain diagonal matrix. Finally, ω_ξ and ω_q represent the weights for the two additive functions employed in the fitness function.

In (7), the trajectories for both the robot base and the arm's center of mass are considered instead of just considering the trajectory for the arm's center of mass. This is because the goal of this study is to improve the robot's overall stability while the robot base moves, by reconfiguring its arm. If only the trajectory for the arm's center of mass was considered, then there would be solutions that require only the arm's motion, having the robot base in place. Clearly, this is not the purpose of the present work.

III. RESULTS

To validate the method proposed in Section II, the problems of trajectory generation for the arm's center of mass (see (5)) and of trajectory tracking (see (7)) are solved for a mobile manipulator traversing a slope and following a circular trajectory for the robot base. In this work, as a proof of principle, the proposed method is implemented in simulation using MATLAB, and the evaluation of the method's efficiency will be a future study subject.

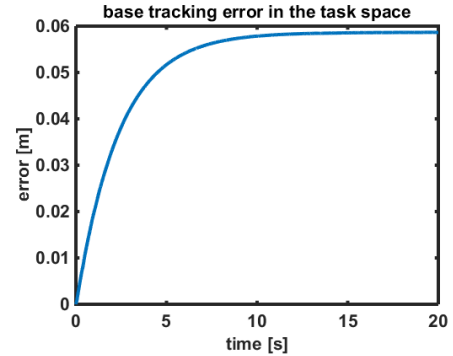
Fig. 8 shows ten converged manipulator configurations as the mobile manipulator is asked to follow a circular path on a slope. For this simulation $\omega_1=0.9$ and $\omega_2=0.1$ are used. For the first four robot configurations, the manipulator is reconfigured to be on the front right side. The ordering of these configurations are also indicated in Fig. 8. Then the manipulator is reconfigured towards the front left side during

the subsequent four robot-base configurations. Finally, the manipulator is again reconfigured to the front right side for the last two configurations. Notice that the CMA algorithm chooses the location of the center of mass corresponding to the stability local maximum closest to the location of the arm's center of mass obtained in the previous time step, unless there is a significant value difference between the two maxima.

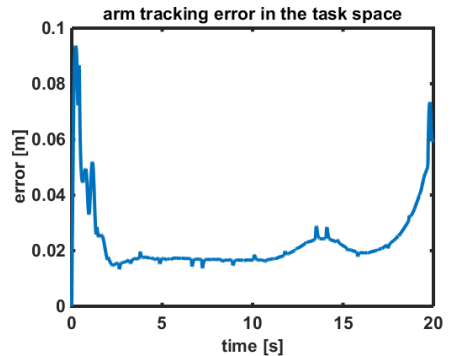
Fig. 9 shows the tracking of both the trajectory for the robot base and the trajectory for the arm's center of mass as the robot traverses on a slope. Fig. 10(a) and Fig. 10(b) show a closer look at the tracking performance using the controller described in the previous section. The results show that the tracking errors for both the base and the arm are bounded.

Fig. 11 shows the joint angular velocities for both the base and the arm. In this study, the arm's joint angular velocities are imposed to be limited at 4π [rad/s]. Fig. 11(a) shows the saturation of the joint angular velocities during the initial stage of the simulation. In general, the first arm joint angular speed is larger than the rest because the desired location of the arm's center of mass mainly shifts parallel to the slope plane with small changes along the direction normal to this plane.

On the other hand, Fig. 11(b) shows that the robot-base's joint angular velocities quickly raise to reach the desired velocities. One can also see that the values of the angular velocity of the right-side joint are always larger than those of the left-side joint. This result is because the robot is

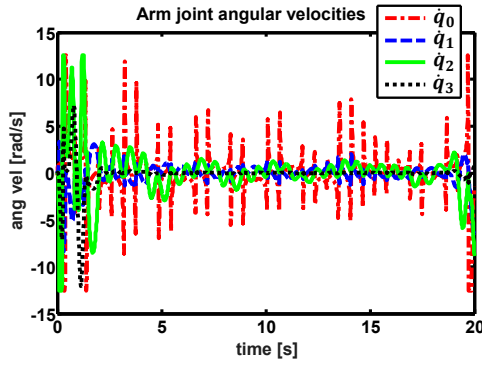


(a) Tracking error for the robot base

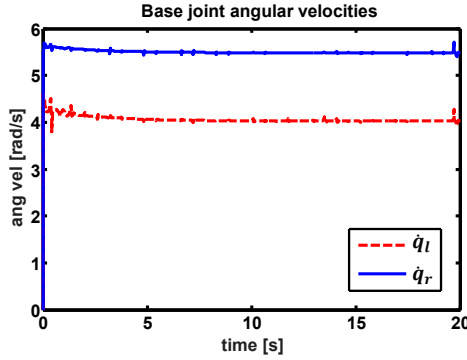


(b) Tracking error for the arm

Fig. 10. Tracking error for both the robot base and the arm



(a) Angular velocities of the arm joints



(b) Angular velocities of the base joints

Fig. 11. Angular velocities for the arm and base joints

asked to track the circular trajectory counterclockwise, and, therefore, the left side of the robot is the inner side as the robot tracks the desired trajectory.

Finally, the results corresponding to a comparison study are illustrated in Fig. 12, with the purpose to show the benefit of reconfiguring the arm for stability. The obtained results indicate that the force-angle stability measure achieved as the robot tracks the trajectories shown in Fig. 9 by reconfiguring its arm are clearly better than when the robot's arm is completely removed.

IV. CONCLUSION AND FUTURE WORK

We propose in this paper a method for motion generation and control of a mobile manipulator intended to navigate on uneven terrains, which can be considered to be as a set of slopes. The arm motion generation is defined by optimizing the tip-over stability of the overall system. The inverse kinematics-based controller ensures the tracking of the trajectories for the robot base and the arm's center of mass. The proposed approach is validated in simulation on a sloped ground, and, in the future, this method will be generalized to other types of uneven terrains.

ACKNOWLEDGEMENT

This work is partially supported by the RAPID-FRAUDO project (Num. 112906242) funded by the DGA (French Defence Agency). Jae-Yun Jun appreciates Alexandre Eudes for the discussions maintained during the early phase of the present work.

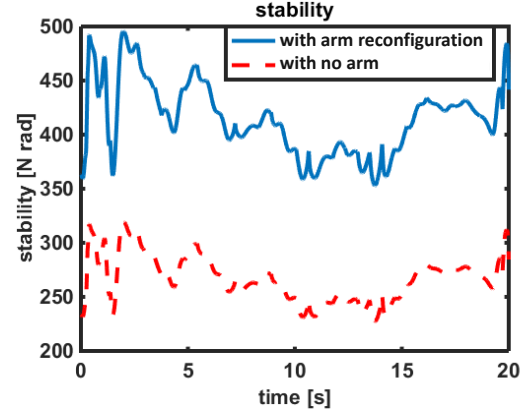


Fig. 12. Stability measure recorded while the mobile manipulator tracks the trajectories

REFERENCES

- [1] S. Sugano, Q. Huang, and I. Kato, "Stability criteria in controlling mobile robotic systems," in *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems*, Tokyo, Japan, 1993.
- [2] T. Yoshikawa, *Foundations of robotics: analysis and control*. Cambridge, Massachusetts: The MIT Press, 1990.
- [3] S. L. Chiu, "Task compatibility of manipulator postures," *The International Journal of Robotics Research*, vol. 7, no. 5, pp. 13–21, 1988.
- [4] A. Mohri, S. Furuno, and M. Yamamoto, "Trajectory planning of mobile manipulator with end-effector's specified path," in *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2001.
- [5] G. Oriolo and C. Mongillo, "Motion planning for mobile manipulators along given end-effector paths," in *Proc. of IEEE International Conference on Robotics and Automation*, 2005.
- [6] N. Hansen and A. Ostermeier, "Adapting arbitrary normal mutation distributions in evolution strategies: The covariance matrix adaptation," in *Proc. of IEEE International Conference on Evolutionary Computation*, 1996.
- [7] E. Papadopoulos and D. Rey, "A new measure of tipover stability margin for mobile manipulators," in *Proc. of IEEE International Conference on Robotics and Automation*, Minneapolis, MN, USA, 1996.
- [8] B. Siciliano, "Kinematic control of redundant robot manipulators: a tutorial," *Journal of intelligent and robotic systems*, vol. 3, pp. 201–212, 1990.
- [9] D. P. Martin, J. Baillieul, and J. M. Hollerbach, "Resolution of kinematic redundancy using optimization techniques," *IEEE Transactions on Robotics and Automation*, vol. 5, no. 4, pp. 529–533, 1989.
- [10] W. Miksch and D. Schroeder, "Performance-functional based controller design for a mobile manipulator," in *Proc. of IEEE International Conference on Robotics and Automation*, 1992.
- [11] C.-C. Wang and V. Kumar, "Velocity control of mobile manipulators," in *Proc. of IEEE International Conference on Robotics and Automation*, 1993.
- [12] F. G. Pin, J.-C. Culioli, and D. B. Reister, "Using minimax approaches to plan optimal task commutation configurations for combined mobile platform-manipulator systems," *IEEE Transactions on Robotics and Automation*, vol. 10, no. 1, pp. 44–54, 1994.
- [13] Cameleon EOD, Eca Robotics: [http://www.eca-robotics.com/en/robotic-vehicle/robotics-terrestrial-unmanned-ground-vehicles-\(ugv\)-cameleon-eod-lightweight-eod-ugv/22.htm](http://www.eca-robotics.com/en/robotic-vehicle/robotics-terrestrial-unmanned-ground-vehicles-(ugv)-cameleon-eod-lightweight-eod-ugv/22.htm).
- [14] R. Hartenberg and J. Denavit, *Kinematic synthesis of linkages*. McGraw-Hill, 1965.
- [15] E. G. Gilbert, D. W. Johnson, and S. Keerthi, "Fast procedure for computing the distance between complex objects in three-dimensional space," *IEEE Journal of Robotics and Automation*, vol. 4, no. 2, pp. 193–203, 1988.