2013 American Control Conference (ACC)
Washington, DC, USA, June 17-19, 2013

Globally Converging MIMO Optimal Controller for Adaptive Manipulation of Mobile Robots with Redundant Arms

Paul M. Moubarak, *Student Member, IEEE* and Pinhas Ben-Tzvi, *Senior Member, IEEE*

***Abstract*—In this paper, the stability of mobile robots during object manipulation with redundant arms is investigated. A new fast-converging MIMO control algorithm, called the *Circles of INitialization (COIN)*, is introduced to calculate globally optimal postures of redundant manipulators. The algorithm is employed in real-time to prevent mobile robots tip-over during dynamic eccentric manipulation. This is achieved with tip-load measurement by adapting the position of the base relative to the line of action, and recalculating the global joint posture that offsets the risk of toppling. This closed-loop control scheme to maximize the arm's payload and stabilize the robot's posture is applied to serial redundant manipulators with *n*-joints.**

I. Introduction

In MOBILE ROBOTIC manipulation applications, the mobile platform balance is determined by the tilting moment the arm's weight, the inertial dynamics and the object's weight generated around the base pivot, compared to the stabilizing moment the base weight generates around the same pivot. When the tilting moment exceeds the stabilizing moment, the robot tends to tip-over.

In general, the problem of tip-over instability during object manipulation can be seen in two folds. On one hand, the robot needs to calculate a joint posture which enables the end-effector to reach the object, while minimizing the component of the tilting moment that the eccentric weight of the arm generates around the base pivot. On the other hand, the robot should also respond to external load disturbances that result in tip-over instability, such as the prompt increase in the tip load induced by the object on the arm, or the inertial dynamics resulting from the robot's mobility. In this paper, we show that this adaptability can be achieved autonomously, by updating the robot's position relative to the end-effector's trajectory in order to offset the disturbance, and recalculating a new arm posture that maintains end-effector contact with the object.

In the literature, the problem of tip-over instability for mobile robots is considered for applications where the vehicle is following a winding trajectory (e.g., navigating around obstacles), while transporting an object from one place to another [1]–[6]. However, these methods, and others [7], [8], exhibit three limiting aspects:

a) *Offline solution*: where tip-over-free arm trajectories are planned *a priori*, thus discarding the constraint of convergence speed in real-time.
b) *Lack of online adaptability*: where the robot is unable to react to unpredictable disturbances that may occur during a given manipulation task.
c) *No consideration for end-effector manipulation:* where most solutions consider the problem of transporting objects along a pre-defined path, but do not account for tip-over instability during manipulation.

This paper addresses the last shortcoming by integrating the other two as part of the solution. In most robotic applications, tip-over risk is more likely to occur during object manipulation rather than during object transportation over a pre-defined trajectory, due to the low speeds that mobile robots can deliver. The *COIN* algorithm introduced in this paper accomplishes this adaptive manipulation control as a secondary task that supplements the primary objective of inverse kinematics [9]–[13] and path following through redundancy resolution. *COIN* is capable of using *a priori* training data to achieve fast global convergence, and enable a mobile robot to correct its position and arm posture in real-time to offset the risk of tip-over instability during eccentric manipulation tasks.

II. Circles of Initialization (COIN) Algorithm

A. Problem Formulation

For a mobile robot with a serial redundant arm [14]–[15] mounted on a base – such as in the illustration shown in Fig. 1 – the tilting moment that the *n*-links generate around the pivot can be written in a compact recursive form as

$$ {}^{0}\eta(q) = \sum_{i=1}^{n}\left(\begin{bmatrix} {}^{0}\tilde{d}_{Gi}(q)\,{}^{0}R_{i-1}(q) & {}^{0}R_{i-1}(q) \end{bmatrix}_{3\times 6} \begin{bmatrix} {}^{i-1}g_i M_i \in \mathbb{R}^{3\times 1} \\ {}^{i-1}\eta_i^{ext} \in \mathbb{R}^{3\times 1} \end{bmatrix} \right) \quad (1) $$

where ${}^{0}d_{G_i} \in \mathbb{R}^{3\times 1}$ denotes the vector position of the center of mass G_i of link i expressed in frame 0, and ${}^{0}\tilde{d}_{G_i} \in \mathbb{R}^{3\times 3}$ the skew-symmetric matrix of vector d_{G_i}. ${}^{0}R_{i-1}$ defines the rotation matrix from frame *i – 1* to frame 0, M_i the mass of link i, ${}^{i-1}g_i$ the gravitational acceleration of frame i expressed in frame $i-1$, and ${}^{i-1}\eta_i^{ext}$ the external moment vector (inertial forces, *etc.*) applied on link i and expressed in frame *i - 1*.

Manuscript received September 17, 2012. This work is supported by the Robotics and Mechatronics lab at the George Washington University, Washington, DC 20052, USA.

P. M. Moubarak is a Doctoral student in the Department of Mechanical and Aerospace Engineering at the George Washington University, Washington DC (Phone #: 202-321-7383; e-mail: paul4@gwu.edu)

P. Ben-Tzvi is an Assistant Professor of Engineering and the Director of the Robotics and Mechatronics Lab at the George Washington University, Washington DC (Phone #: 202-994-6149; e-mail: bentzvi@gwu.edu)

978-1-4799-0176-0/$31.00 ©2013 AACC

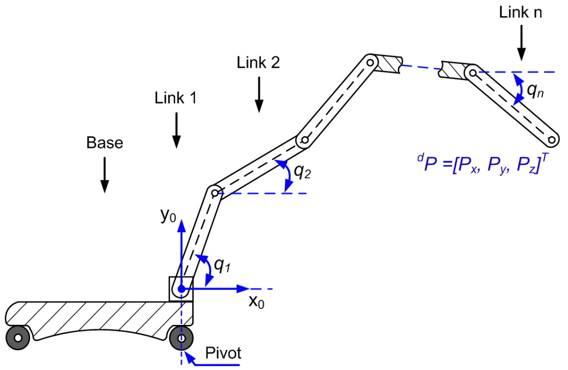

Fig. 1. Schematic representation illustrating a mobile robot with a serial redundant arm (q_i represent generalized coordinates)

Furthermore, the arm's forward kinematics are defined as

$$\begin{bmatrix} {}^d\boldsymbol{P} \in \mathbb{R}^{3\times1} \\ 1 \end{bmatrix} = \left\{ \prod_{i=1}^{n} {}^{i-1}T_i \right\} \begin{bmatrix} 0 \in \mathbb{R}^{3\times1} \\ 1 \end{bmatrix} \qquad (2)$$

where ${}^d\boldsymbol{P} = \begin{bmatrix} P_x & P_y & P_z \end{bmatrix}^T$ is the desired coordinates of the end-effector in the base frame, and ${}^{i-1}T_i \in \mathbb{R}^{4\times4}$ the homogenous transformation matrix that maps the coordinates of frame i into frame $i-1$. This enables the formulation of the optimal inverse kinematics problem as a constrained minimization problem defined as

$$\begin{aligned} &\underset{q}{Min} \quad \eta(q) \\ &subject\ to \quad h(q) = \begin{bmatrix} {}^d\boldsymbol{P} \\ 1 \end{bmatrix} - \left\{ \prod_{i=1}^{n} {}^{i-1}T_i \right\} \begin{bmatrix} 0 \in \mathbb{R}^{3\times1} \\ 1 \end{bmatrix} = 0 \end{aligned} \qquad (3)$$

which can be reiterated as the objective of placing the end-effector at a desired location ${}^dP$, with an arm's posture that maintains the optimal dynamic balance of the robot.

B. *COIN algorithm concept*

The objective of *COIN* is to solve (3) for globality in real-time using the augmented cost function $L(q) = \eta(q) - \lambda^T h(q)$, where λ defines the vector of Lagrange multipliers.

Traditionally, the convergence of a gradient-based optimization solution depends on the location of the initial guess inside the search space, which is often done randomly [16]. With *COIN*, the gradient descent is initialized in the convex space of the solution. This is achieved by extrapolating training data acquired *a priori* about the operation of the redundant arm along one selected direction in the end-effector's workspace. Hereafter, we show that the rotation of this data preserves the convexity of the search space, and enables the solution of (3) to converge to the global optimum in few iterations.

C. *Hypothesis*

To simplify the visualization of *COIN*'s operation, an *n*-link serial arm with a planar workspace is first considered, shown in Fig. 2 (a spatial workspace will be later considered in section III). Furthermore, suppose that an exhaustive global solution exists for (3), such that a *steady-state* (static) global arm posture is generated for a given task located at $x = P_x$, $y = P_y = 0$, $z = P_z = 0$ (blue lines in Fig. 2). Here we chose the *x*-axis as the training direction, although any other axis centered at the origin of frame 0 could be used.

For all other tasks $x = P_x$, $y = P_y \neq 0$ in the arm's workspace whose locus is a circle of radius $R = \sqrt{P_x^2 + P_y^2}$ (red circle), the *COIN* concept states that the global optimal solution of (3) can be initialized with the joint angles of the global posture corresponding to the intersection between this initialization circle and the training axis (blue lines), rotated by an angle $\beta = tg^{-1}(P_y / P_x)$ (black lines in Fig. 2). Such transformation preserves the convexity of the global posture as will be proven subsequently.

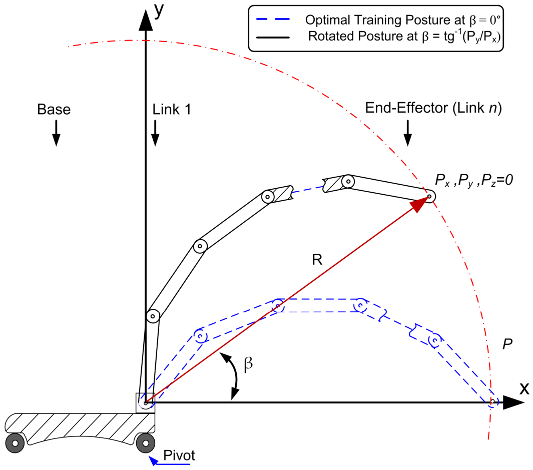
y
Optimal Training Posture at β = 0°
Rotated Posture at β = tg^{-1}(P$_y$/P$_x$)
Base
Link 1
End-Effector (Link *n*)
P_x ,P_y ,P_z=0
R
β
P
x
Pivot

Fig. 2. A redundant arm in global optimal posture for β = 0 *(dashed lines)* and in posture obtained by a rotation $\beta = tg^{-1}(P_y / P_x)$

D. *Proof: Convexity*

In the absence of transients generated by external forces and inertial dynamics (external forces and dynamics will be re-integrated in the algorithm during adaptive manipulation), the tilting moment η in (1), normalized relative to gravity, can be expressed in the plane of the arm as

$$\eta(q) = \sum_{i=1}^{n} M_i l_{Gi} \cos(q_i) + \sum_{i=1}^{n-1} \left(\sum_{k=i+1}^{n} M_k \right) l_i \cos(q_i) \qquad (4)$$

where l_i and l_{G_i} define the length of link i and the location of the center of mass of link i relative to joint $i-1$, respectively. q_i denotes the generalized coordinate of joint i measured relative to the inertial frame.

Equation (4) can be further simplified to

$$\eta(q) = \sum_{i=1}^{n-1} \left\{ M_i l_{Gi} + \left(\sum_{k=i+1}^{n} M_k \right) l_i \right\} \cos(q_i) + M_n l_{Gn} \cos(q_n) \quad (5)$$

where we further define the parameters

$$A_i = M_i l_{Gi} + \left(\sum_{k=i+1}^{n} M_k \right) l_i \qquad A_n = M_n l_{Gn}$$

and rewrite

$$\eta_i = A_i \cos(q_i) \qquad \eta_n = A_n \cos(q_n)$$

to simplify (5) to the following compact expression

$$\eta(q) = \sum_{i=1}^{n-1} \eta_i + \eta_n \tag{6}$$

Using (6) and (2), the solution of (3) can be derived by minimizing the cost function $L(q, \lambda)$, with the constraints appended to the objective using Lagrange multipliers λ

$$L_\beta(q, \lambda) = \eta_\beta(q) - \lambda^T h_\beta(q) \tag{7}$$

In (7), L_β , h_β , η_β respectively represent the *Lagrangian*, the equality constraints, and the tilting moment generated for a task defined by angle β relative to the training direction.

The first objective of this proof is to evaluate the first derivative of L_β and examine its non-violation after a rotation of the initial global posture at β = 0°. The second objective is to investigate the positive definiteness of the Hessian matrix $H_\beta \triangleq \nabla^2 L_\beta$.

To examine the first derivative, we first rewrite the cost function in (6) as $\eta_\beta(q) = \sum_{i=1}^{n-1} \eta_{i,\beta} + \eta_{n,\beta}$ to reflect the contribution of β. In particular, at β=0° the corresponding joint posture is assumed to generate the global minimum for (7). Thus, one can write the global solution at β = 0° as

$$\eta_0^*(q) = \sum_{i=1}^{n-1} \eta_{i,0}^* + \eta_{n,0}^* \tag{8}$$

where the (*) symbol is added to reflect the globality of the posture at β = 0°. The objective is to further express η_β for any angle β in the workspace as a function of the global optimal expression of (8) and its individual terms.

Since we hypothesized that the global configuration for a given $\beta = \tan^{-1}(P_y / P_x)$ can be obtained by a rotation of the global posture corresponding to the end-effector's location at the intersection of the initialization circle of radius $R = \sqrt{P_x^{\ 2} + P_y^{\ 2}}$ with the training direction, we can expand the cost function for any $q_i = \beta + q_i^*$ into

$$\begin{aligned} \eta_\beta(q) = &\left\{ \sum_{i=1}^{n-1} A_i \cos(q_i^*) + A_n \cos(q_n^*) \right\} \cos(\beta) \\ &- \left\{ \sum_{i=1}^{n-1} A_i \sin(q_i^*) + A_n \sin(q_n^*) \right\} \sin(\beta) \end{aligned} \tag{9}$$

In (9), the first bracketed term can be replaced by $\eta_0^*(q)$. In the second bracketed term, *sine* terms can be replaced by

$$A_i \sin(q_i^*) = \sqrt{A_i^2 - \eta_{i,0}^{*\ 2}} \qquad A_n \sin(q_n^*) = \sqrt{A_n^2 - \eta_{n,0}^{*\ 2}} \tag{10}$$

which yields an equation for $\eta_\beta(q)$ as a function of the global expression of the cost function at β = 0° as

$$\begin{aligned} \eta_\beta(q) = &\ \eta_0^* \cos(\beta) \\ &- \left\{ \sum_{i=1}^{n-1} \sqrt{A_i^2 - \eta_{i,0}^{*\ 2}} + \sqrt{A_n^2 - \eta_{n,0}^{*\ 2}} \right\} \sin(\beta) \end{aligned} \tag{11}$$

Similarly, the constraints $h(q) \in \mathbb{R}^{1\times 2}$ derived from the inverse kinematics can be written in terms of β and q_i^* as

$$\begin{aligned} h_{1,\beta} &= R\cos(\beta + \delta) - \sum_{i=1}^{n} l_i \cos(q_i^* + \beta) \\ h_{2,\beta} &= R\sin(\beta + \delta) - \sum_{i=1}^{n} l_i \sin(q_i^* + \beta) \end{aligned} \tag{12}$$

with $R = \sqrt{P_x^{\ 2} + P_y^{\ 2}}$. Furthermore, the expansion of the *sine* and *cosine* terms in (12) allows us to rewrite (12) in terms of β and the global equality constraints at β = 0° defined by

$$\begin{aligned} h_{1,0}^* &= R\cos(\delta) - \sum_{i=1}^{n} l_i \cos(q_i^*) = 0 \\ h_{2,0}^* &= R\sin(\delta) - \sum_{i=1}^{n} l_i \sin(q_i^*) = 0 \end{aligned} \tag{13}$$

The subsequent substitutions yield the general expression for (12) in terms of $h_{1,0}^*$, $h_{2,0}^*$ and angle β of the form

$$\begin{aligned} h_{1,\beta} &= h_{1,0}^* \cos(\beta) - h_{2,0}^* \sin(\beta) \\ h_{2,\beta} &= h_{1,0}^* \sin(\beta) + h_{2,0}^* \cos(\beta) \end{aligned} \tag{14}$$

Using (11) and (14), it is now possible to express $L_\beta(q, \lambda)$ in (7) in terms of the global cost function and the equality constraints at $\beta = 0°$ as

$$\begin{aligned} L_\beta = &\left\{ \eta_0^* - \lambda_1 h_{1,0}^* - \lambda_2 h_{2,0}^* \right\} \cos(\beta) \\ &- \left\{ \sum_{i=1}^{n-1} \sqrt{A_i^2 - \eta_{i,0}^{*\ 2}} + \sqrt{A_n^2 - \eta_{n,0}^{*\ 2}} - \lambda_1 h_{2,0}^* + \lambda_2 h_{1,0}^* \right\} \sin(\beta) \end{aligned} \tag{15}$$

Because λ can be assigned any random value, we chose to set $\lambda_1 \triangleq \lambda_1^*$ and $\lambda_2 \triangleq \lambda_2^*$. This simplifies the gradient of L_β in (15) to

$$\nabla L_\beta = -\nabla \left\{ \begin{aligned} &\sum_{i=1}^{n-1} \sqrt{A_i^2 - \eta_{i,0}^{*\ 2}} + \\ &\qquad \sqrt{A_n^2 - \eta_{n,0}^{*\ 2}} - \lambda_1^* h_{2,0}^* + \lambda_2^* h_{1,0}^* \end{aligned} \right\} \sin(\beta) \tag{16}$$

taking into account that $\nabla(\eta_{,0}^* - \lambda_1^* h_{1,0}^* - \lambda_2^* h_{2,0}^*) = 0$ because of optimality at $\beta = 0°$. The remaining expression in (16) is not necessarily zero. This concludes that the rotated global configuration at $\beta = 0°$ does not generate the global optimal posture at the new task defined by $\beta = \tan^{-1}(P_y / P_x)$.

However, the convexity of the posture at $\beta = 0°$ remains unchanged under a rotation β. This means that the extrapolated Hessian matrix of the original global configuration $H(\beta + q^*)$ remains strictly positive definite $\forall \beta$. The expansion of the *sine* and *cosine* terms in $H(\beta + q^*)$ can be rearranged in a compact form in terms of the Hessian matrix of the global posture at $\beta = 0°$ as

$$H(\beta + q^*) = H_0^* \cos(\beta) - H_S \sin(\beta) > 0 \tag{17}$$

where H_0^* represents the Hessian matrix at $\beta = 0°$, which is positive definite because of globality. In (17), matrix H_S is analogous to the gradient of (16), where by definition

$$-H_S \triangleq H_\beta = \nabla \left\{ \frac{\nabla L_\beta}{\sin(\beta)} \right\} \tag{18}$$

with ∇L_β as defined in (16). For $-\delta° \leq \beta \leq 90 - \delta°$ which delimits the domain of the arm's workspace where the risk of toppling is the greatest (the front domain delimited by *x*-*y* axes in Fig. 3), inequality (17) is satisfied $\forall \beta, q^*$ only if matrix H_S is strictly negative definite. This leads to the conclusion that matrix H_β in (18) is strictly positive definite, and remains so for small variations in parameters q^* and β around the rotated posture.

Therefore, the domain encompassed by the arm's postures around the rotated initial configuration represents a convex set. This means that if an optimum of (7) is found in this convex space via a gradient descent starting from the rotated global posture as initial guess, this optimum will be the *global* minimum, which corresponds to the global arm posture for the desired end-effector position defined by ${}^dP$.

E. Visualization of Training

The concept of *COIN* can be generalized to an infinite family of circles that cover the whole workspace of the manipulator. To visualize this more encompassing aspect of *COIN*, consider a sample scenario of a 4-link serial arm. If the *x*-axis is chosen as a training direction, an exhaustive solution of (3) for a discrete progression of end-effector tasks spanning the length of the training axis (Fig. 3), generates a global joint and Lagrange multipliers history as plotted in Fig. 4. This exhaustive solution is accomplished offline using any existing global search algorithm. As a result, the computational time prior to convergence is not a concern during training. Furthermore, the number of samples for the training process is chosen in a way to provide enough data points to generate a smooth polynomial fit for the history of joint and Lagrange multipliers.

The joint angles history and the Lagrange multipliers in Fig. 4 can be curve fitted with a polynomial function

$$\sigma_j(\xi) = \sum_{i=0}^{10} c_{j(i+1)} x^{10-i} \tag{19}$$

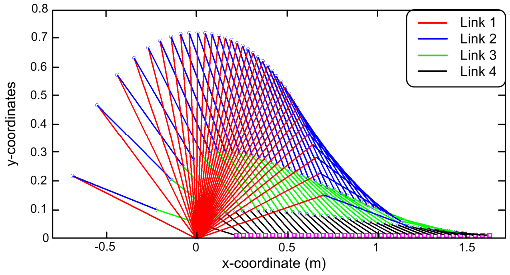

Fig. 3. Training process of 4-link arm with end-effector along $y = 0$

where $c_{j(i+1)}$ are the polynomial coefficients and $\sigma = [q \quad \lambda]^T \in \mathbb{R}^{(n+m)\times 1}$ is the augmented vector of joint angles, with $n = 4$ and $m = 2$ for the 4-link arm.

Once this training data is acquired, *COIN* switches from the Cartesian to the Polar space. This means that any task defined by $x = P_x$, $y = P_y$ in the workspace, can be located on a circle of radius $R = \sqrt{P_x^2 + P_y^2}$, centered at $x_0 y_0$ and intersecting the training axis. At this intersection point, the global posture and the corresponding Lagrange multipliers can be calculated analytically via direct substitution in (19). This posture can then be rotated by an angle $\beta = tg^{-1}(P_y / P_x)$ to create the initial guess for the global solution of any other task in the arm's workspace.

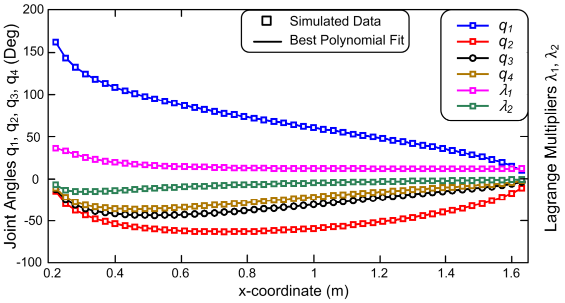

Fig. 4. Global training history of joint angles and Lagrange multipliers, and corresponding polynomial best fit for the 4-link arm of Fig. 3

F. Simulation Results: Planar Case-study

A case-study simulation on 4-link serial arm is considered to visualize the operation of COIN and to compare its convergence speed to existing global search algorithms.

Fig. 4 shows a progression of end-effector tasks located on a circle of radius R = 1.4m. The intersection of this circle with the training axis $y = 0$ generates the initial global posture (blue dashed lines), which is calculated via direct substitution in (19). To calculate the global posture for all other tasks on this circle defined by $x = P_x$, $y=P_y$, the initial posture is rotated by $\beta = tg^{-1}(P_y / P_x)$ to generate the initial guess (black dashed lines) for (3) at every other task.

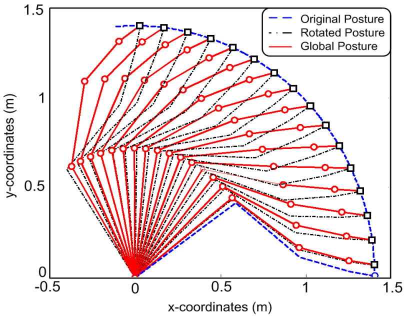

Fig. 5. Initial posture, rotated posture, and optimal posture for a sequence of end-effector tasks located on the same initialization circle radii R = 1.4m

This approach can be identically adopted to calculate the optimal joint posture for other tasks located on any other initialization circle in the arm's workspace. The global convergence of *COIN* for this case-study occurs within 4 – 7 iterations only. This translates into an average computational time of 27ms/task. In comparison, the *MultiStart* global optimization algorithm [17] achieves convergence at an average time of 1.77 sec/task for the 4-joint arm (processor employed: Dual-Core 2.93GHZ, 3.25GB RAM). This means that COIN achieves a computational time reduction by 96.4 – 98.5% as shown in Fig. 6, where the computational time prior to convergence for *COIN* and *Multistart* is plotted as a function of angle β for the 4-joint arm in Fig. 5.

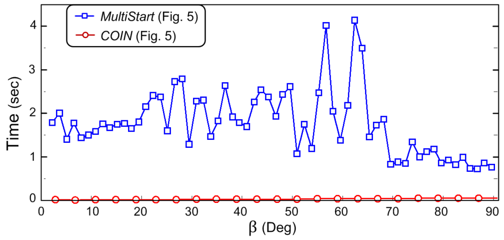

Fig. 6. Comparison of convergence time between *COIN* and *MultiStart* algorithms for the two case studies of Fig. 5 and Fig. 6

Such capabilities of *COIN* can be extrapolated to include spatial trajectories. In such case, an end-effector task defined by ${}^{d}\boldsymbol{P} = \begin{bmatrix} P_x & P_y & P_z \end{bmatrix}^T$ is located on a sphere of radius $R = \sqrt{P_x^2 + P_y^2 + P_z^2}$ rather than a circle of radius $R = \sqrt{P_x^2 + P_y^2}$. As a result, the training posture undergoes two consecutive transformations consisting of a yaw rotation by an angle $\gamma = tg^{-1}(P_z / P_x)$ in the *xz*-plane, followed by a pitch rotation by $\beta = tg^{-1}(P_y / \sqrt{P_x^2 + P_y^2})$ about the z-axis (Fig. 7) to intersect the location of a desired task defined by vector ${}^{d}P$.

These two rotations generate the initial guess for *COIN* which calculates the global joint posture from (3) at ${}^{d}P$. A simulation of this process is shown in Fig. 7, where an 11-joint arm is shown tracking a spatial knot trajectory. Note that the global convergence time in this case is identical to the planar trajectories (~20ms/task) simulated in Fig. 5.

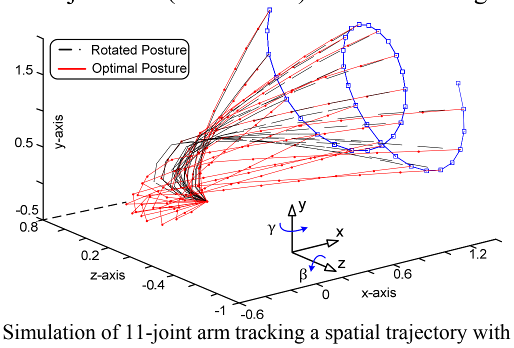

Fig. 7. Simulation of 11-joint arm tracking a spatial trajectory with *COIN*

III. ADAPTIVE MANIPULATION

In the previous discussion, neither the base stabilizing moment, nor the arm dynamics and external load were taken into consideration. This is because the purpose of *COIN* was to resolve redundancy and calculate a *steady-state* static global joint posture that meets the forward kinematic constraints defined for a specific task ${}^{d}P$. However, in real-time adaptive manipulation, the arm inertial dynamics and external tip load create transient forces/moments that disturb the stable posture at ${}^{d}P$. This causes the robot to lose balance, despite the globally optimal static posture.

The primary objective of *COIN* is to compensate for these transients as a primary task, by generating swift adaptive manipulation commands that reposition the base relative to the object. The initiation of these commands is based on an extrapolation of (3) to account for inequality constraints.

A. Dynamic Stability Control

During object manipulation, the load on the arm increases as the object gradually loses contact with the ground. The measurement of end-effector's force and moment components is sampled at a rate k (period *T*), which generates a discrete progression of external forces ${}^{n}F_{eff}(k)$ and moments ${}^{n}\eta_{eff}(k)$ in the end-effector's frame *n*. As these measurements are fed back to an optimal controller running *COIN* (Fig. 8), the base repositioning and the arm reconfiguration are accomplished when stability condition

$$M_0\,{}^{0}l_{c0} \times {}^{0}g + {}^{0}\eta_b - {}^{0}\delta\eta > \\ {}^{0}\eta(q) + \begin{bmatrix} {}^{d}\tilde{P}\left({}^{0}R_n\right) & {}^{0}R_n \end{bmatrix}_{3\times 6} \begin{bmatrix} {}^{n}F_{eff}(k) \in \mathbb{R}^{3\times 1} \\ {}^{n}\eta_{eff}(k) \in \mathbb{R}^{3\times 1} \end{bmatrix} \quad (20)$$

is violated. In (20), M_0 denotes the mass of the mobile base, ${}^{0}\eta_b$ defines the external moments acting on the base, ${}^{0}g$ and ${}^{0}l_{c0} = \begin{bmatrix} x_0 & y_0 & z_0 \end{bmatrix}^T$ denote (respectively) the gravitational acceleration and the position of the base's center of mass, both relative to frame *0*. ${}^{0}\eta(q) \in \mathbb{R}^{3\times 1}$ is the arm's tilting moment, ${}^{d}\tilde{P} \in \mathbb{R}^{3\times 3}$ depicts the skew-symmetric matrix of vector ${}^{d}P$, and ${}^{0}R_n \in \mathbb{R}^{3\times 3}$ the rotation matrix mapping the coordinates of frame *n* into frame *0*. ${}^{0}\delta\eta \in \mathbb{R}^{3\times 1}$ is further added as a safety margin to the stabilizing moment.

Anytime condition (20) is violated, a new displacement vector $\Delta(k)$ is calculated in terms of the skew symmetric matrix ${}^{0}\tilde{F}_{eff}(k)$ as

$$\Delta(k) = \left[{}^{0}\tilde{F}_{eff}(k) \right]^{-1}_{3\times 3} \begin{bmatrix} {}^{0}\eta(q) + {}^{0}\delta\eta + {}^{0}\eta_{eff}(k) \\ - {}^{0}\eta_b - M_0\,{}^{0}l_{c0} \times {}^{0}g \end{bmatrix}_{3\times 1} \quad (21)$$

and the robot is moved forward by a distance $\Delta(k-1) - \Delta(k)$. However, when this adaptive mobility is initiated, inertial forces resulting from the acceleration of the robot are induced onto the arm, which are seen as disturbances by *COIN*. These forces are included in vector $\eta_i^{ext}(q)$ in (1) and (6), whose components are expanded as

$${}^{0}\eta_i^{ext}(q) = {}^{0}\eta_i^{lin}(q) + {}^{0}\eta_i^{ang}(q) + {}^{0}\eta_i^{per}(q) \quad (22)$$

where
$$\begin{aligned} {}^{0}\eta_i^{lin}(q) &= M_i \left\{ {}^{0}d_{G_i} \times \left({}^{0}a_{G_i} + a_b \right) \right\} \\ {}^{0}\eta_i^{ang}(q) &= {}^{0}R_{i-1} \left({}^{i-1}I_{G_i} \times {}_{i-1}\ddot{q}_i \right) \textit{for a revolute joint} \end{aligned} \quad (23)$$

with ${}^{0}a_{G_i}$ representing the acceleration of the center of mass of link i relative to frame *0*, a_b the acceleration of the base resulting from adaptive mobility, ${}^{i-1}I_{G_i}$ the mass moment of inertia matrix of link i about the center of mass G_i expressed in frame $i-1$, and ${}^{0}\eta_i^{per}(q)$ the vector of any additional peripheral measurable external moments.

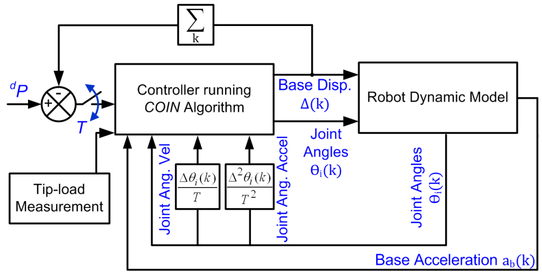

Fig. 8. Block diagram for adaptive mobility with a MIMO controller

Note that $^{0}\eta_{n}^{per}(q)=0$ in (22) since the external moments acting on link *n* are explicitly isolated in (20) and (21). Also note that the components of the linear and angular accelerations of the links are calculated via discrete finite difference of joint displacements (Fig. 8), which are measured using absolute encoders. The component of a_b can also be measured using an inertial measurement unit.

B. Case-study Simulation of Adaptive Manipulation

A simulation of dynamic adaptive manipulation is shown in Fig. 10 for a case-study 3-link arm mounted atop a mobile robot. The simulated task consists of lifting a 35Kg tip load in 3 seconds, starting from an inadequate posture of the arm.

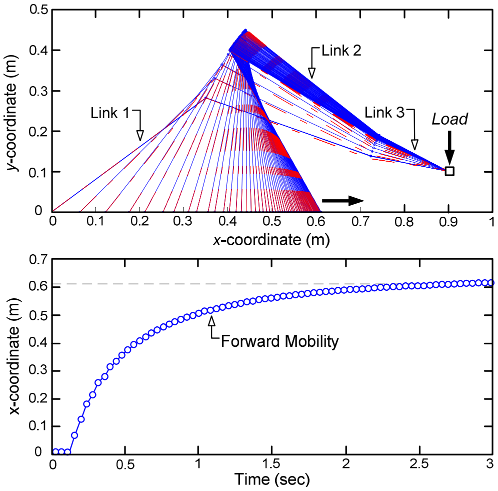

Fig. 9. Simulation of dynamic adaptive manipulation for a 3-joint arm (red lines are training data translated with the arm)

Stability condition (20) is evaluated at every instance *k* of tip-load measurement. For the first few instances, the incremental tip load does not violate (20), and thus the robot's initial position is maintained. However, as the tip load continues to increase while the object loses contact with the ground, condition (20) is violated at multiple subsequent instances. At every violation, the robot moves forward by $\varDelta(k-1)-\varDelta(k)$ to offset the risk of toppling induced by both the tip load, and the inertial dynamics resulting from forward mobility. This autonomous locomotion continues as shown in Fig. 9 until a stable tip-over-free posture for full-load manipulation is found after a total displacement of 0.6m.

IV. Conclusion and Future Work

This paper presented a new *COIN* algorithm characterized by a fast global convergence rate that can be used for redundancy resolution, trajectory following, and tip-over stability of mobile robots via dynamic adaptive manipulation in real-time. Future work will explore the stability of *COIN* by investigating the possibility of adaptive initialization in the event when the gradient descent diverges. This adaptive initialization will seek to update the offline training data in real-time in order to provide robustness to the controller in event of numerical divergence.

References

[1] Q. Huang, K. Tanie and S. Sugano, "Coordinated Motion Planning for a Mobile Manipulator Considering Stability and Manipulation," *International J. Robotics Research*, vol.19, no. 8, pp. 732 – 742, 2000.

[2] S. Czarnetzki, S. Kerner, and O. Urbann, "Observer-based Dynamic Walking Control for Biped Robots," *J. Robotics and Autonomous Systems*, vol. 57, no. 8, pp. 839 – 845, 2009.

[3] J. Kim, W.K. Chung, Y. Youm and B.H. Lee, " Real-time ZMP Compensation Method using Null Motion for Mobile Manipulators," *IEEE International Conference on Robotics and Automation*, ICRA'05, Washington, DC, May 2005.

[4] M.H. Korayem, V. Azimirad and A. Nikoobin, "Maximum Load-carrying Capacity of Autonomous Mobile Manipulator in an Environment with Obstacle Considering Tip Over Stability," *International Journal of Advanced Manufacturing Technology*, vol. 46, pp. 811 – 829, 2010.

[5] D.A. Rey and E.G. Papadoupoulos,"Online Automatic Tipover Prevention for Mobile Manipulators," *IEEE/RSJ International Conference on Intelligent Robots and Systems*, IROS'97, France, September 1997.

[6] A. Meghdari, D. Naderi and M.R. Alam, "Neural-network-based Observer for Real-time Tipover Estimation," *Journal of Mechatronics*, vol. 15, pp. 989 – 1004, 2005.

[7] M.H. Korayem, H. Ghariblu and A. Basu, "Maximum Allowable Load of Mobile Manipulators for Two Given End Points of End Effector," *International Journal of Advanced Manufacturing Technology*, vol. 24, pp. 743 – 751, 2004.

[8] M.H. Korayem, A. Nikoobin and V. Azimirad, "Maximum Load Carrying Capacity of Mobile Manipulators: Optimal Control Approach," *Robotica*, vol. 27, pp. 147 – 159, 2009.

[9] M. Tarokh, and M. Kim," Inverse Kinematics of 7-DOF Robots and Limbs by Decomposition and Approximation," *IEEE Transactions on Robotics*, vol. 23, no. 3, pp. 595 – 600, 2007.

[10] Z. Keméný, "Redundancy Resolution in Robots Using Parameterization Through Null Space," *IEEE Transactions on Industrial Electronics*, vol. 50, no. 4, pp. 777 – 783, 2003.

[11] L. Sciavicco, and B. Siciliano, "A Solution Algorithm to the Inverse Kinematic Problem for Redundant Manipulators," *IEEE Journal of Robotics and Automation*, vol. 4, no. 4, pp. 403 – 410, 1988.

[12] C. W. Wampler, II, "Manipulator Inverse Kinematic Solutions Based on Vector Formulations and Damped Least-Squares Methods," *IEEE Transactions on Systems, Man, and Cybernetics,vol. SMC-16*, no. 1, pp. 93 – 101, 1986.

[13] K. Tchoń, Repeatable, "Extended Jacobian Inverse Kinematics Algorithm for Mobile Manipulators," *Systems & Control Letters*, vol. 55, no. 2, pp. 87 – 93, 2006.

[14] P. Ben-Tzvi, A.A. Goldenberg, J.W. Zu, "Design and Analysis of a Hybrid Mobile Robot Mechanism with Compounded Locomotion and Manipulation Capability", *Journal of Mechanical Design, Transactions of the ASME*, vol. 130, no. 7, pp. 1–13, July 2008.

[15] P. Ben-Tzvi, "Experimental Validation and Field Performance Metrics of a Hybrid Mobile Robot Mechanism", *Journal of Field Robotics*, vol. 27, no. 3, pp. 250–267, May 2010.

[16] J.A. Carretero and M.A. Nahon, "Solving Minimum Distance Problems With Convex or Concave Bodies Using Combinatorial Global Optimization Algorithms," *IEEE Transactions on Systems, Man, and Cybernetics—Part B,* vol. 35, no. 6, pp. 1144 – 1155, 2005.

[17] U. Zsolt, L. Lasdon, J.C. Plummer, F. Glover, J. Kelly, and R. Martí, "Scatter Search and Local NLP Solvers: A Multistart Framework for Global Optimization," *J. Computing*, vol. 19, no. 3, pp. 328 – 340, 2007.