Robotica: page 1 of 13. © Cambridge University Press 2013
doi:10.1017/S0263574713000520

A globally converging algorithm for adaptive manipulation and trajectory following for mobile robots with serial redundant arms

Paul Moubarak and Pinhas Ben-Tzvi*

Robotics and Mechatronics Laboratory, Department of Mechanical and Aerospace Engineering, The George Washington University, Washington, DC 20052, USA

(Accepted May 1, 2013)

SUMMARY
In this paper the tip-over stability of mobile robots during manipulation with redundant arms is investigated in real-time. A new fast-converging algorithm, called the *Circles Of INitialization* (*COIN*), is proposed to calculate globally optimal postures of redundant serial manipulators. The algorithm is capable of trajectory following, redundancy resolution, and tip-over prevention for mobile robots during eccentric manipulation tasks. The proposed algorithm employs *a priori* training data generated from an exhaustive resolution of the arm's redundancy along a single direction in the manipulator's workspace. This data is shown to provide educated initial guess that enables *COIN* to swiftly converge to the global optimum for any other task in the workspace. Simulations demonstrate the capabilities of *COIN*, and further highlight its convergence speed relative to existing global search algorithms.

KEYWORDS: Global optimization; Inverse kinematics; Redundancy resolution; Adaptive manipulation control; Mobile robots, Tip-over instability.

Nomenclature

$x_0 y_0 z_0$	Cartesian frame 0 attached to the base joint of the arm.
$x_i y_i z_i$	Cartesian frame attached to joint i+1.
C_i	Center of frame $x_i y_i z_i$.
n	Number of joints.
q_i	Generalized coordinate of link i measured relative to inertial frame 0.
${}^0\eta(q)$	Tilting moment of all combined forces generated around the arm's pivot (frame 0).
${}^0\eta^s(q)$	Steady-state moment component of ${}^0\eta(q)$, normalized with respect to gravity.
${}^{i-1}\eta_{\mathrm{i}}^{\mathrm{ext}}$	Vector of external moments applied on link i and expressed in frame I–1.
G_i	Center of mass of link i.
l_i	Length of link i.
l_{G_i}	Distance between G_i and the center of joint I–1.
${}^0d_{Gi}$	Vector position of G_i expressed in frame 0 (${}^0d_{Gi} \in \mathbb{R}^{3\times 1}$.)
${}^{i-1}r_{G_i}$	Vector position of G_i relative to frame $i-1$ (${}^{i-1}r_{Gi} \in \mathbb{R}^{3\times 1}$).
${}^0a_{Gi}$	Linear acceleration of G_i relative to frame 0 (${}^0a_{Gi} \in \mathbb{R}^{3\times 1}$).
${}^0\ddot{d}_{C_{i-1}}$	Acceleration of the center of frame $i-1$ relative to frame 0
${}^{i-1}g_i$	Gravity vector acting on link i and expressed in frame I–1.
a_b	Linear acceleration vector of the mobile base ($a_b \in \mathbb{R}^{3\times 1}$).
${}^0R_{i-1}$	Rotation matrix from frame I–1 to frame 0 (${}^0R_{i-1} \in \mathbb{R}^{3\times 3}$).
M_0	Mass of the mobile base.
M_i	Mass of link i.
${}^d\boldsymbol{P}$	Desired task, also defined as the desired vector position of the end-effector relative to frame 0 (${}^d\boldsymbol{P} = [\,P_x\ P_y\ P_z\,]^T$).
${}^d\tilde{\boldsymbol{P}}$	Skew-symmetric matrix of ${}^d\boldsymbol{P}$ (${}^d\tilde{\boldsymbol{P}} \in \mathbb{R}^{3\times 3}$).
${}^{i-1}T_i$	Homogenous transformation matrix mapping frame i attached to joint i+1 into frame I–1 (${}^{i-1}T_i \in \mathbb{R}^{4\times 4}$).
${}^{i-1}I_{G_i}$	Mass moment of inertia matrix of link i about G_i expressed in frame I–1.
λ^T	Vector of Lagrange multipliers.
Σ	Displacement vector of the mobile base relative to the desired position ${}^d\boldsymbol{P}$ ($\Sigma \in \mathbb{R}^{3\times 1}$).
${}^nF_{\mathrm{eff}}$, ${}^n\eta_{\mathrm{eff}}$	Measurements of external forces and external moments in the end-effector's frame n, respectively

1. Introduction

For mobile robots with manipulator arms, the maximum payload capacity of the arm is dictated by the robot's ability to maintain dynamic stability during eccentric manipulation tasks. In fact, the arm's payload capacity is a direct function of its joint posture, as the latter defines the eccentricity of the combined external load relative to the platform's pivot.

From a broad standpoint, the balance of a mobile platform (including industrial hydraulic machines such as excavators[1]) is determined by the tilting moment the arm and the applied external load generate around the pivot axis of the mobile base, compared with the stabilizing moment the weight of the base generates around the same pivot. In the event where

* Corresponding author. E-mail: bentzvi@gwu.edu

the tilting moment exceeds the stabilizing moment, the robot will have the tendency to tip over.

In the literature, the problem of tip-over instability for mobile robots is studied for applications where the arm possesses a low order of kinematic redundancy, and where the vehicle is following a winding trajectory (for example, navigating around obstacles) while transporting an object between two separate locations. For instance, the Zero Moment Point (ZMP)[2] method is proposed as an offline solution for planning a robot's motion along a given path.[3–5] The ZMP is geometrically defined as the point on the ground where the moments of all external forces applied onto the robot are in static equilibrium. Thus, the objective of this solution is to plan *a priori* motion and arm posture such that the ZMP remains inclusive of the contact polygon created between the robot and the ground.

A second perspective monitors the ground reaction forces exerted on the robot's wheels.[6] The Zero Tire Upward Force (ZETUF) method,[7] for instance, states that tip-over instability occurs if the sum of the ground reaction forces acting on two adjacent tires of a four-wheel vehicle reaches zero. Thus, by measuring or estimating these reaction forces, it becomes possible to detect the regions of instability along a given path. Alternatively, tip-over instability can be analyzed as an explicit path planning problem, where the path of a robot is generated for maximum load carrying capacity of the manipulator arm.[8] Such solution is derived as an explicit optimal control problem using Pontryagin's principle.[9]

By analyzing the state-of-the-art contributions to the problem of tip-over instability, the following three limiting properties are found to be dominant:

1. *Offline solutions*: Where the computational cost is not a critical aspect of the numerical solution since the planning problem is confined to an *a priori* trajectory for the mobile robot and its arm.
2. *Lack of online adaptability*: Where the predictive approach for planning the robot's path on a tip-over free trajectory does not account for the possibility of posture adaptation in reaction to any unforeseen disturbances along the path.
3. *No consideration for end-effector manipulation*: Where most solutions consider the problem of tip-over instability associated with object transportation along a pre-defined path, but discard the onset of destabilization occurring during end-effector manipulation from an inadequate posture.

The objective of this paper is to address this third challenge by incorporating the first and second as part of the solution. In fact, in robotic applications, tip-over instability is more likely to occur when a mobile robot is manipulating an object from an inadequate posture, rather than during object transportation, due to the relatively low speeds involved in the motion of mobile robots. Thus, for autonomous applications, a challenge arises with respect to the best position and arm posture that a mobile robot should adopt during an eccentric manipulation task. In this paper, we argue that such problem should be resolved as an on-line reactive adaptive manipulation scheme based on real-time measurements of external loads instead of an offline predictive approach.

For redundant manipulator arms, adaptive manipulation can be accomplished as a secondary task that supplements the primary objective of inverse kinematics and trajectory following through redundancy resolution. However, such optimal problem formulation presents the following two numerical challenges:

1. The mathematical redundancy generates a large number of local solutions where only the smallest of them (the global extremum) guarantees optimal stability. In fact, the difference between a global and a local arm posture when minimizing the tilting moment could mean the difference between the robot maintaining balance during a manipulation task or the robot losing balance and toppling.
2. The practical necessity to minimize the secondary task at the global level makes the iterative convergence speed a primary consideration for on-line applications.

This convergence requirement is made even more compelling given that gravity-based tip-over instability occurs near-instantaneously, thus requiring an algorithm that reacts swiftly to offset the risk of destabilization, and correct the robot's posture to ensure balance. However, existing global solvers are characterized by a slow convergence rate,[10,11] which makes them undesirable for sensor-based applications such as adaptive manipulation. This typically favors local solvers, where redundancy[12] is resolved locally at either the velocity level[13–20] or the position level.[21–24]

In this direction, this paper proposes a new algorithm called *Circles of Initialization* (*COIN*), which is characterized by the following five merits:

1. *Global convergence*: Where the algorithm uses the *a priori* data to preserve the convexity of the search space. This allows the inverse kinematic solution to converge to the global optimum anytime such optimum exists.
2. *Fast convergence*: Which is enabled by initializing the solution near the global optimum by rotating (or translating) the *a priori* data to intersect the location of the arm's task inside its workspace.
3. *On-line dynamic adaptive manipulation*: Where the convergence speed enables the integration of *COIN* in sensor-based applications to correct the position of mobile robots, and adapt their arms' posture to offset the risk of toppling during eccentric manipulation tasks.
4. *Trajectory following*: Where the convergence speed allows a sequential calculation of joint postures that enable the arm to follow a planar or spatial trajectory in tandem with dynamic adaptive manipulation.
5. *Generality*: Where the speed of convergence of *COIN* is shown to extrapolate to serial redundant arms with *n*-joints.

In this paper, after a recast of preliminary motivations and investigations reported in Moubarak and Ben-Tzvi (2011),[25] a formulation of the optimality problem is presented along with a mathematical proof of convexity. Simulation results further validate the speed of convergence, and the ability of the algorithm to deliver dynamic adaptive manipulation

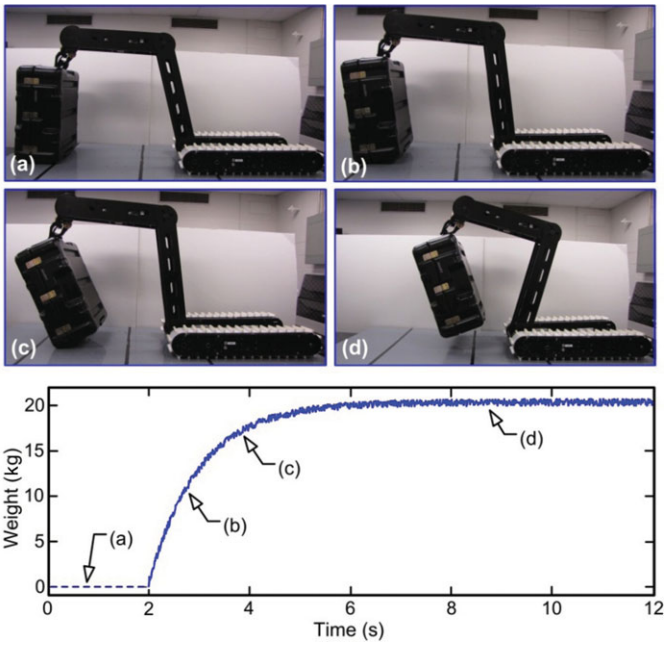

Fig. 1. (Colour online) A progression of sequences executed by a mobile robot[26–28] during object lifting, and corresponding instances of incremental external load (on the arm).

by correcting the robot's posture in a closed-loop control scheme.

2. Early Investigations

The notion of a dynamically adaptive robot posture generated in response to changes in the magnitude of an external load was presented in Moubarak and Ben-Tzvi (2011)[25] for a class of robots shown in Fig. 1.[25–28] A force sensor embedded in the fingers of the end-effector enables the robot to measure a discrete progression of incremental external load during manipulation tasks. This gravity-driven progression is reflected in Fig. 1, where a sequence of manipulation maneuvers is matched with the corresponding change in the external load that the arm experiences during the process.

Starting from a zero-load condition at the onset of the maneuver corresponding to the object being completely on the ground (Fig. 1(a)), the load increases gradually as the contact surface between the object and the ground decreases (Fig. 1(b) and (c)) until load saturation is reached when the object is fully carried by the arm (Fig. 1(d)). In this particular scenario, the kinematic redundancy of the manipulator is of order 1. This allows the resolution of redundancy as an exhaustive accelerated search based on the minimization of the tilting moment that the three-link arm creates around the base pivot (the end-effector denotes the third link).[25]

This search algorithm was further extrapolated for adaptive manipulation in the presence of a tip load. Based on the discretization of external load measurements, the position of the mobile base is adapted relative to the object to minimize the effect of the tilting moment. In such case the minimization is achieved for the combined moment induced by both tip load and arm's eccentric weight. This generates an adaptive movement of the base that entails a corresponding reconfiguration of the arm as reported in Moubarak and Ben-Tzvi (2011).[25]

3. Redundancy Resolution for *n*-Joints: Problem Formulation

A broader extension of the early investigations can be accomplished for a manipulator arm with *n*-links, mounted atop a mobile robot such as in the schematic shown in Fig. 2. The general expression for the arm's tilting moment $\eta(q)$ generated around the base pivot can be written in a recursive vector format as

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

where ${}^0\tilde{d}_{Gi} \in \mathbb{R}^{3\times 3}$ denotes the skew-symmetric matrix of vector ${}^0d_{Gi}$. On the other hand, the general form of the arm's forward kinematics can be written in a matrix format as

$$
\begin{bmatrix} {}^d\boldsymbol{P} \in \mathbb{R}^{3\times 1} \\ 1 \end{bmatrix} = \left\{ \prod_{i=1}^{n} {}^{i-1}T_i \right\} \begin{bmatrix} 0 \in \mathbb{R}^{3\times 1} \\ 1 \end{bmatrix}, \tag{2}
$$

where the end-effector's orientation in the plane of the arm is defined by q_n. This enables the formulation of an optimal inverse kinematics problem as an explicitly constrained minimization problem defined as

$$
\begin{aligned}
&\underset{q}{\text{Min}} \quad {}^0\eta(q) \\
&\text{subject to} \quad {}^0h(q) = \begin{bmatrix} {}^dP \\ 1 \end{bmatrix} - \left\{ \prod_{i=1}^{n} {}^{i-1}T_i \right\} \times \begin{bmatrix} 0 \in \mathbb{R}^{3\times 1} \\ 1 \end{bmatrix} = 0,
\end{aligned} \tag{3}
$$

which can be reiterated as the objective of placing the end-effector at a desired location ${}^dP = [\, P_x \;\; P_y \;\; P_z \,]^T$ with an arm's posture that minimizes the effect of the tilting moment.

The solution of Eq. (3) can be derived using a numerical search under pre-defined convergence criteria. From a mathematical perspective, the convergence rate of local and global optimization methods depends on the initial guess and its location relative to the minima in the search space. While local convergence could often be guaranteed within a reasonable number of iterations, global convergence may or may not occur, and may even be prohibitively slow if the initialization is made randomly.[29] In the following we propose to resolve the redundancy in Eq. (3) by introducing the *COIN* algorithm that initializes the solution in the convex space of the global optimum.

4. *COIN* Algorithm

In order to present a clear discussion on the proposed *COIN* algorithm, the method will be first stated, derived, and proven for a two-dimensional (2D) workspace, and later extrapolated to a 3D space. Readers are encouraged to refer

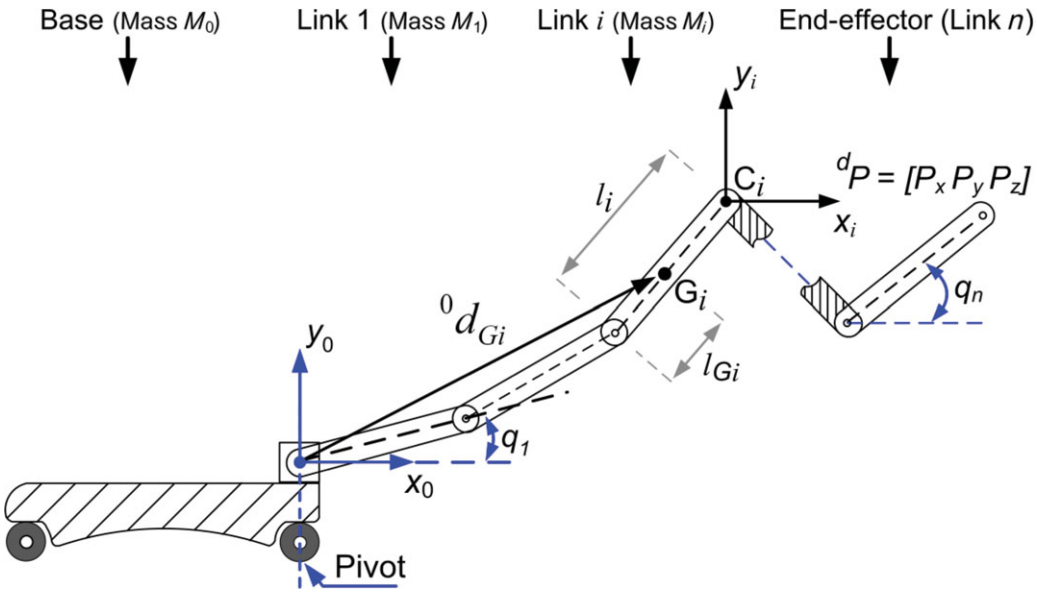

Fig. 2. (Colour online) Kinematic diagram of a mobile robot with a serial redundant arm.

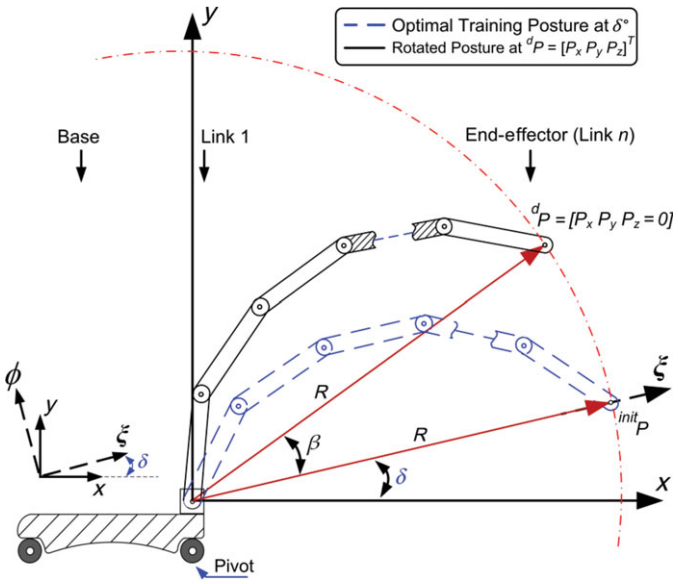

Fig. 3. (Colour online) Schematic showing a redundant arm in an exhaustively generated optimal global configuration at δ° ($\beta = 0^\circ$) (blue dashed lines), and in a configuration obtained by rotation of the optimal posture by $\beta = \tan^{-1}(P_y/P_x) - \delta$ (black solid lines) (R in Fig. 3 depicts R_{2D}).

to the simulation video provided in Moubarak and Ben-Tzvi (2012)[32] for an initial insight into the operation of *COIN*.

4.1. Definitions

With reference to Fig. 3, a definition of the three main components of *COIN* algorithm are presented as follows:

(a) *Training direction* ξ: By definition, this is a single direction chosen in the workspace of the arm along which an exhaustive offline solution of Eq. (3) is performed to generate the training data for *COIN*. Such direction is at an angle δ with the x-axis ($0^\circ \leq \delta \leq 90^\circ$).

(b) *Training data*: Also referred to as *a priori* data, it is the data resulting from the training process, which is curve-fitted and later extrapolated by *COIN* to generate an educated initial guess for the solution of Eq. (3) for any other task in the arm's workspace.

(c) *Radius of* COIN: It is defined as R relative to the base frame 0, and depicts the radius of the circle along which a planar task $^dP = [\ P_x\ \ P_y\ 0\ ]^T$ is located, or the radius of the sphere along which a spatial task $^dP = [\ P_x\ \ P_y\ \ P_z\ ]^T$ is located. We define $R_{2D} = \sqrt{P_x^2 + P_y^2}$ for a 2D task, and $R_{3D} = \sqrt{P_x^2 + P_y^2 + P_z^2}$ for a 3D task.

4.2. Hypothesis

The physics that governs the *steady-state* global solution of Eq. (3) for a redundant arm in one particular configuration remains the same for any other configuration all across the workspace.

As such, consider a *steady-state* global solution of Eq. (3) given for an end-effector position located along one specific direction ξ known as the training direction in the plane of the arm, and defined by $^{init}P = [\ R \triangleq R_{2D}\ 0\ 0\ ]^T$ (blue dashed lines in Fig. 3) in a right-handed $\xi\phi\psi$ Cartesian system.

The notion of *COIN* states that the numeric cyclic solution for all other end-effector positions defined by vector $^dP = [\ P_x\ \ P_y\ \ P_z \triangleq 0\ ]^T$ in the Cartesian xyz-space, and located on a circle of radius R_{2D} in the equivalent polar space, can be initialized with the joint posture corresponding to the global configuration at ^{init}P, extrapolated by an angle $\beta = \tan^{-1}(P_y/P_x) - \delta$ relative to the $\xi\psi$-plane (black solid lines in Fig. 3, where δ(deg) is the pitch angle between the x-axis and the ξ-axis). This extrapolation rotates the training posture from the initial configuration at ^{init}P to the desired position at ^dP. We hypothesize that this transformation places the arm in a posture that is at the global optimal posture, or *close* to its convex space for the given desired task.

4.3. Proof: Global initialization

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 steady-state tilting moment ${}^0\eta^s$ can be written recursively in the plane of the arm as

$$ {}^0\eta^s(q) = \sum_{i=1}^{n} M_i l_{Gi} \cos(q_i) + \sum_{i=1}^{n-1} \left(\sum_{k=i+1}^{n} M_k \right) l_i \cos(q_i). \tag{4} $$

Equation (4) can be further simplified into a factored-out summation expression in the form of

$$ {}^0\eta^s(q) = \sum_{i=1}^{n-1} \left\{ M_i l_{Gi} + \left(\sum_{k=i+1}^{n} M_k \right) l_i \right\} \cos(q_i) + M_n l_{Gn} \cos(q_n), \tag{5} $$

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

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

to simplify Eq. (5) to the following compact expression

$$ {}^0\eta^s(q) = \sum_{i=1}^{n-1} \eta_i^s + \eta_n^s. \tag{6} $$

Using Eq. (6) and the equality constraints defined in Eq. (2), the numerical solution of the optimization problem in Eq. (3) can be derived by minimizing the corresponding Lagrangian $L(q, \lambda)$, where the constraints are appended to the cost function using Lagrange multipliers λ,

$$ L_{,\beta}(q, \lambda) = {}^0\eta^s_{,\beta}(q) - \lambda^T {}^0h_{,\beta}(q). \tag{7} $$

In Eq. (7), $L_{,\beta}$, $h_{,\beta}$, and $\eta^s_{,\beta}$ represent, respectively, the Lagrangian, the constraints, and the tilting moment generated by the arm for a task defined by angle β relative to the training direction (see Fig. 3).

The primary objective of the proof is to evaluate the first derivative of $L_{,\beta}$ and examine its non-violation after a rotation of the initial global posture at δ ($\delta = 0^\circ$ if $\xi \equiv x$-axis). The second objective is to investigate the positive definiteness of the Hessian matrix, $H_\beta \triangleq \nabla^2 L_{,\beta}$, and the global convexity of the search space encompassed by the vicinity of the rotated posture.

To examine the first derivative test, we first rewrite the cost function in Eq. (6) as ${}^0\eta^s_{,\beta}(q) = \sum_{i=1}^{n-1} \eta^s_{i,\beta} + \eta^s_{n,\beta}$ to reflect the influence of angle β. In particular, for $\beta = 0^\circ$, which corresponds to an end-effector's position located along the training direction ξ, the corresponding joint posture is – by definition – the global minimum of the Lagrangian problem in Eq. (7). Thus, one can write the global solution at δ as

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

where $({}^*)$ symbol is added to reflect the globality of the posture at δ. The objective is to further express ${}^0\eta^s_{,\beta}$ for any angle β in the workspace as a function of global optimal expression of Eq. (8) and its individual terms.

Since we hypothesized that the global arm configuration for a given $\beta = \tan^{-1}(P_y/P_x) - \delta$ 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_{2D} with ξ-direction, we can expand the cost function for any $q_i = \beta + q_i^*$ into

$$ {}^0\eta^s_{,\beta}(q) = \left\{ \sum_{i=1}^{n-1} A_i \cos(q_i^*) + A_n \cos(q_n^*) \right\} \cos(\beta) - \left\{ \sum_{i=1}^{n-1} A_i \sin(q_i^*) + A_n \sin(q_n^*) \right\} \sin(\beta). \tag{9} $$

In Eq. (9), the first bracketed term can be replaced by $\eta^s_{,0}{}^*(q)$. In the second bracketed term, the *sine* terms can be replaced by the following equivalent expressions,

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

which yield an expression for ${}^0\eta^s_{,\beta}(q)$ in terms of the global cost function at δ as

$$ {}^0\eta^s_{,\beta}(q) = \eta^s_{,0}{}^* \cos(\beta) - \left\{ \sum_{i=1}^{n-1} \sqrt{A_i^2 - \left(\eta^s_{i,0}{}^*\right)^2} + \sqrt{A_n^2 - \left(\eta^s_{n,0}{}^*\right)^2} \right\} \sin(\beta). \tag{11} $$

In an identical scheme, the constraints $h_{,\beta}(q) = [\, h_{1,\beta} \; h_{2,\beta} \,]^T$ derived from the inverse kinematics of the end-effector in the plane of the arm can be written in terms of β and the global posture angles q_i^* as follows:

$$ h_{1,\beta} = R_{2D} \cos(\beta + \delta) - \sum_{i=1}^{n} l_i \cos(q_i^* + \beta) $$

$$ h_{2,\beta} = R_{2D} \sin(\beta + \delta) - \sum_{i=1}^{n} l_i \sin(q_i^* + \beta) \tag{12} $$

Furthermore, Eq. (12) can be simplified by expanding the *sine* and *cosine* terms, which allows us to rewrite Eq. (12) as a function of angle β and the global equality constraints at δ

($\beta = 0^\circ$) denoted by

$$h^*_{1,0} = R_{2D}\cos(\delta) - \sum_{i=1}^{n} l_i \cos(q^*_i) = 0,$$
$$h^*_{2,0} = R_{2D}\sin(\delta) - \sum_{i=1}^{n} l_i \sin(q^*_i) = 0. \tag{13}$$

The subsequent substitutions yield the general expression for Eq. (12) in terms of $h^*_{1,0}$, $h^*_{2,0}$, and angle β as follows:

$$h_{1,\beta} = h^*_{1,0}\cos(\beta) - h^*_{2,0}\sin(\beta),$$
$$h_{2,\beta} = h^*_{1,0}\sin(\beta) + h^*_{2,0}\cos(\beta). \tag{14}$$

Using Eqs. (11) and (14), it is now possible to express $L_{,\beta}(q, \lambda)$ in Eq. (7) in terms of the global cost function and the equality constraints at δ as

$$L_{,\beta} = \left(\eta^{s}_{,0}{}^{*} - \lambda_1 h^*_{1,0} - \lambda_2 h^*_{2,0}\right)\cos(\beta) - \left\{\sum_{i=1}^{n-1}\sqrt{A_i^2 - \left(\eta^{s}_{i,0}{}^{*}\right)^2} + \sqrt{A_n^2 - \left(\eta^{s}_{n,0}{}^{*}\right)^2} - \lambda_1 h^*_{2,0} + \lambda_2 h^*_{1,0}\right\}\sin(\beta). \tag{15}$$

However, since it can be proved under not very restrictive conditions that the Lagrange multipliers λ^T change continuously with small variations in the constraints and cost function,[30] it is valid to assume that vector λ^T for the rotated posture is close to that for $\beta = 0^\circ$. As a result, one can set $\lambda_1 \triangleq \lambda^*_1$ and $\lambda_2 \triangleq \lambda^*_2$, i.e., the λ^T values corresponding to the global posture along the training direction ξ. This property is subsequently highlighted in the training data in Fig. 5, where vectors λ^*_1 and λ^*_2 do not vary significantly for different end-effector tasks. This simplifies the gradient of $L_{,\beta}$ in Eq. (15) to

$$\nabla L_{,\beta} = -\nabla\left\{\sum_{i=1}^{n-1}\sqrt{A_i^2 - \left(\eta^{s}_{i,0}{}^{*}\right)^2} + \sqrt{A_n^2 - \left(\eta^{s}_{n,0}{}^{*}\right)^2} - \lambda_1 h^*_{2,0} + \lambda_2 h^*_{1,0}\right\}\sin(\beta), \tag{16}$$

where we set $\nabla(\eta^{s}_{,0}{}^{*} - \lambda_1 h^*_{1,0} - \lambda_2 h^*_{2,0}) = 0$ because of optimality at δ° ($\beta = 0^\circ$). The remaining expression of $\nabla L_{,\beta}$ in Eq. (16) is not necessarily equal to zero. This leads to the conclusion that the rotated global configuration at δ° does not generate the global optimal posture at the new task defined by $\beta = \tan^{-1}(P_y/P_x) - \delta$.

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

$$H(\beta + q^*) = H^*_0\cos(\beta) - H_S\sin(\beta) > 0, \tag{17}$$

where H^*_0 represents the Hessian matrix at δ, which is positive definite because of globality. In Eq. (17) we note that matrix H_S is analogous to the gradient of Eq. (16), where by definition

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

with $\nabla L_{,\beta}$ as defined in Eq. (16). For $-\delta^\circ \le \beta \le 90^\circ - \delta^\circ$, which delimits the domain of the arm's workspace where the risk of toppling is the greatest (the front domain bounded by the x–y coordinate 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 Eq. (18) is strictly positive definite, and remains so for small variations in parameters q^* and β.

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

We postulate that this conclusion is valid for an n- assembly of revolute and prismatic joints.

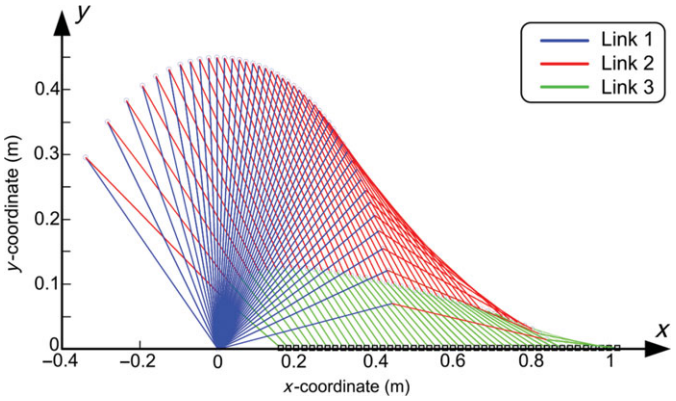

Fig. 4. (Colour online) Visualization of the training process for a planar arm with three joints, where Link 3 denotes the end-effector (training direction $\xi \equiv x$-axis).

5. Simulations and Convergence Speed

5.1. *Inverse kinematics: three-joint revolute arm*

The first simulation of *COIN* refers to the manipulator arm of the robot shown in Fig. 1. A training direction ξ is chosen along the x-axis of the workspace, and is discretized into a sequence of consecutive end-effector positions. For each position, an offline global solution of the augmented minimization problem in Eq. (7) is generated exhaustively as visualized in Fig. 4.

This sequential global solution of Eq. (7) generates a history of joint angles and Lagrange multipliers λ as

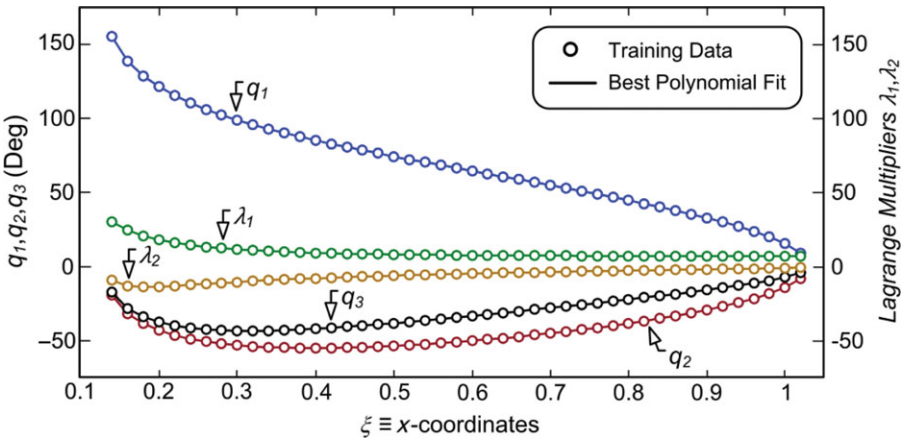

Fig. 5. (Colour online) Joint angles, Lagrange multipliers history, and corresponding best polynomial fit resulting from the training process of the three-joint arm in Fig. 4.

the end-effector covers the reachable span of the training direction. This history, plotted in Fig. 5, is curve-fitted with a polynomial of order m in the generalized spatial coordinate ξ

$$\rho_j(\xi) = \sum_{i=0}^{m} \mu_{j(i+1)} \xi^{m-i}, \tag{19}$$

where $\xi \equiv x$ in the case of Fig. 4. In Eq. (19), $\mu_{j(i+1)} \in \mathbb{R}^{(n+p)\times(m+1)}$ is the matrix of polynomial coefficients, where p denotes the length of vector $h(q)$, with $1 \leq j \leq n+p$ (n is the number of joints, and m is the order of polynomials ρ_j). $\rho = [\, q \;\; \lambda \,]^T \in \mathbb{R}^{(n+p)\times 1}$ represents the augmented vector of joint angles, with $n = 3$ and $p = 2$ for the three-joint arm of Fig. 4.

Once the training data $\rho_j(\xi)$ is generated, the solution of Eq. (7) is performed in the polar space. This means that for the arm shown in Fig. 4, any end-effector position defined by vector ${}^d P = [\, P_x \;\; P_y \;\; 0\,]^T$ is located on an initialization circle of radius R_{2D}. The intersection of this circle with the training direction ξ (the x-axis in this case) generates the global training configuration $\rho^{\text{train}} = \rho(\xi = R_{2D})$. The rotation of this training posture by an angle $\beta = \tan^{-1}(P_y/P_x) - \delta$ ($\delta = 0^\circ$ for the arm of Fig. 4) delivers the initial guess for the global solution of Eq. (7) that lies in the convex space of the desired position ${}^d P$. The visualization of this process is shown in two sample simulations depicted in Figs. 6 and 7.

In Fig. 6, a sequence of end-effector positions is defined on the same initialization circle of radius $R_{2D} = 0.8$ m. The intersection of this circle with the training direction (x-axis) generates the global training posture $\rho^{\text{train}}(0.8)$ (blue dashed lines). The rotation of this posture by an angle $\beta = \tan^{-1}(P_y/P_x)$ corresponding to every sequential task, furnishes the initial guess (black dashed lines) in the global search space. This enables the numeric solution of Eq. (7) to converge to the global optimum (red solid lines) within three to four iterations.

A more general simulation of *COIN* is shown in Fig. 7 for a series of end-effector positions located on different circles of initialization. The circle on which every task is situated intersects the training direction (x-axis) at a point that generates a corresponding training posture from Eq. (19)

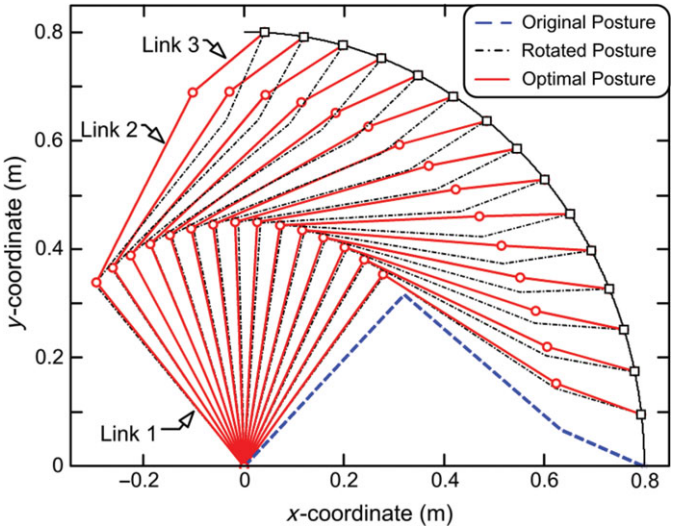

Fig. 6. (Colour online) Initial posture (blue dashed lines), rotated posture (black dashed lines), and optimal posture (red solid lines) for a sequence of end-effector positions located on the same initialization circle of radius $R_{2D} = 0.8$ m.

(blue dashed lines). The rotation of this posture by an angle β creates the initial guess (black dashed lines) that enables the convergence of the global optimal solution in Eq. (7) within three to four iterations.

5.2. Inverse kinematics: 10-joint revolute arm

The concept of *COIN* can be further extrapolated to a robotic manipulator with n-joints such as the 10-joint arm illustrated in Figs. 8(a) and (b).

Figure 8(a) shows a simulation of close-proximity end-effector tasks located on the same initialization circle of radius $R_{2D} = 0.5$ m, while Fig. 8(b) shows a simulation of distant tasks located on a circle of radius $R_{2D} = 2.4$ m. In both cases, the global solution of Eq. (7) for every task ${}^d P$ is initialized with the posture corresponding to the intersection of the initialization circle with the training direction, rotated by $\beta = \tan^{-1}(P_y/P_x) - \delta$. For the 10-joint arm, the training axis was arbitrarily chosen at $\delta = +45^\circ$.

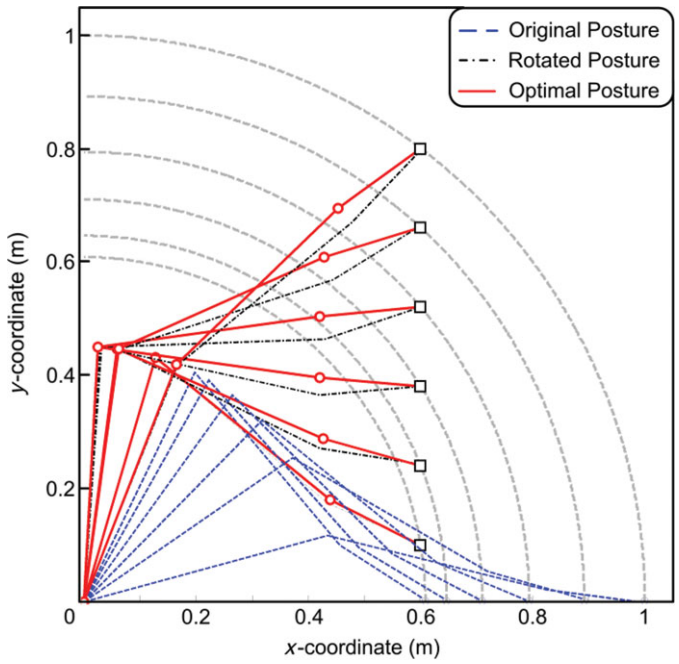

Fig. 7. (Colour online) Initial posture (blue dashed lines), rotated posture (black dashed lines), and optimal posture (red solid lines) for a sequence of end-effector positions located on different initialization circles.

The convergence of the solution to the global arm posture is accomplished in all cases within four to seven iterations. This performance expands to any other task, where the fast convergence rate applies to the entire family of circles covering the whole workspace of the end-effector.

5.3. *Redundancy resolution: comparison of convergence speed*

To highlight the global convergence speed of *COIN*, a comparison with existing global search routines is established, where the optimization problem stated in Eq. (3) is solved using the *MultiStart* algorithm.[31]

Although this comparative analysis should also account for existing redundancy resolution routines such as the *Damped Least Squares*[17] and *Extended Jacobian*,[18,19] the local convergence of these methods makes such comparison mathematically irrelevant. This is further inhibited by the variable convergence rate of these algorithms, where the convergence speed depends directly on the location of the initial guess inside the search space, which is often selected randomly.

As such, the time required prior to convergence for the *COIN* and the *MultiStart* algorithms is plotted in Fig. 9 as a function of angle β $(-\delta \leq \beta \leq 90 - \delta)$ for the sample case study simulations shown in Figs. 6 and 8 (processor employed: Dual-Core 2.93 GHZ, 3.25 GB RAM). For the three-joint problem, the average time required by the *MultiStart* algorithm to converge to the global solution was 1.87 s, compared with 27 ms for the *COIN* algorithm.

For the 10-joint case study, the *MultiStart* algorithm converges to the global solution in an average time of 1.324 s for the close-proximity simulation (Fig. 8(a)), and 2.154 s for distant tasks (Fig. 8(b)). In comparison, *COIN* converges to the same globally optimal posture in an average time of 19 ms for both simulations. This means that *COIN* reduces the on-line computational time by an average of 98–99%, which makes it ideally fast for adaptive manipulation applications as will be discussed in Section 6. We note that the termination criteria for the gradient descent subroutine of *COIN* was set to $\|\nabla L_{,\beta}\| < 0.001$ in all simulations, where $L_{,\beta}$ is as defined in Eq. (7), and $\|\cdot\|$ denotes the Euclidean norm.

5.4. *Trajectory tracking: extrapolation to 3D space*

The previous simulations focused on planar workspaces, where the objective was to visualize the operation of *COIN* and compare its convergence speed to existing global search routines.

In this section we extrapolate the concept of *COIN* to include spatial trajectories. In a 3D space, the desired end-effector position defined by ${}^{d}P = [\, P_x \;\; P_y \;\; P_z \neq 0\,]^T$ is located on a sphere of radius R_{3D} instead of a circle of radius R_{2D}. The ball B resulting from the union of all spheres that

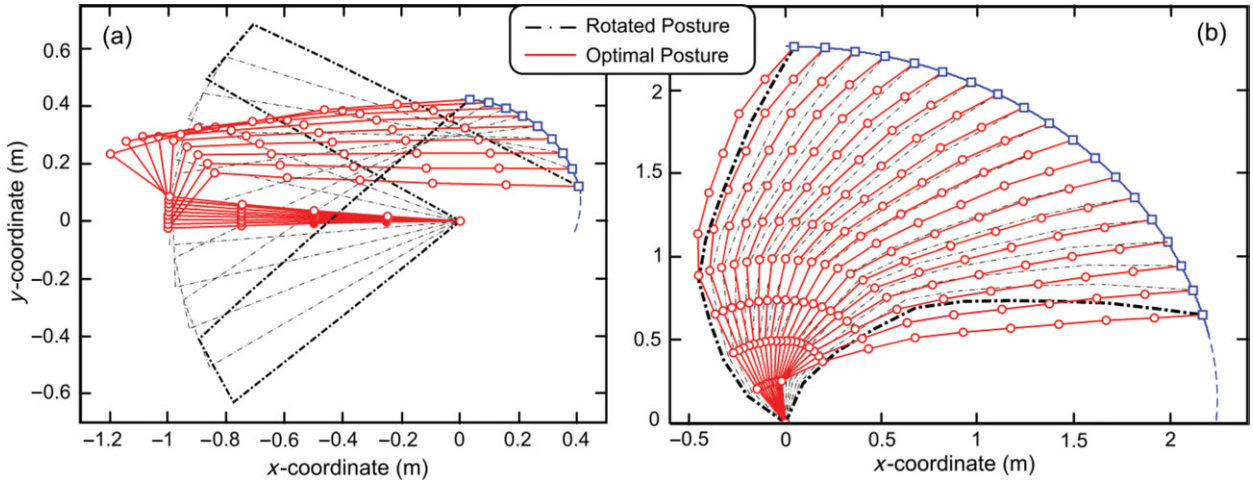

Fig. 8. (Colour online) Rotated posture (black dashed lines) and optimal posture (solid red lines) for a 10-joint arm for a sequence of end-effector tasks. (a) Close-proximity tasks located on an initialization circle of radius $R_{2D} = 0.5$ m. (b) Distant positions located on the circle of radius $R_{2D} = 2.4$ m.

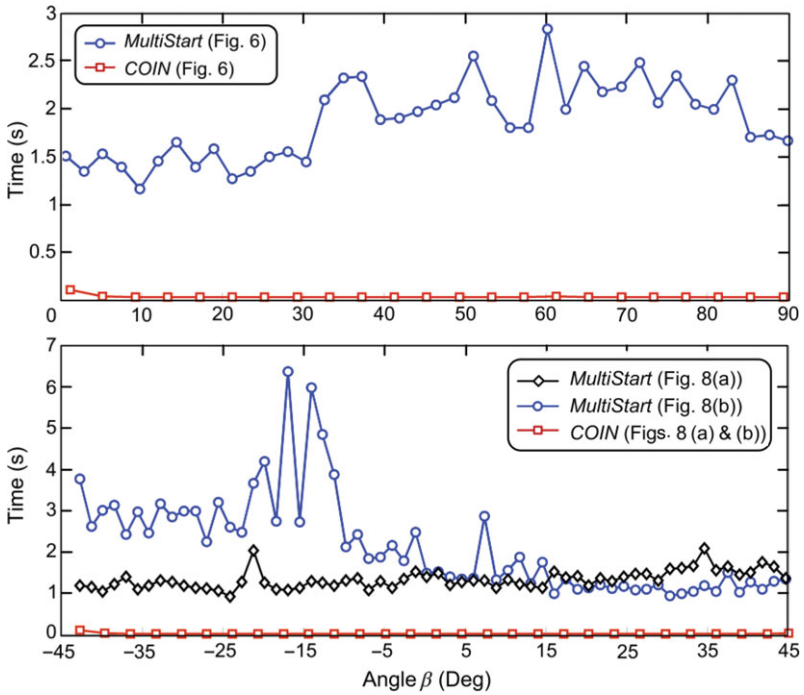

Fig. 9. (Colour online) Comparison of computational time prior to convergence between *COIN* and the *MultiStart*[31] algorithms for redundancy resolution of the arms in Figs. 6, 8 (a), and (b).

cover the spherical-equivalent of the Cartesian workspace is further segregated into an infinite family of circles C_{pq}, where $B = \bigcup_{p=1}^{\infty} (\bigcup_{q=1}^{\infty} C_{pq})$. Each sphere $S_p = \bigcup_{q=1}^{\infty} C_{pq}$ is defined by the global training posture $\rho_p^{\text{train}} = \rho(\xi = R_{3D})$ generated via direct substitution in Eq. (19), and resulting from the intersection of S_p of radius $R_{3D}$with the training direction ξ.

The training posture ρ_p^{train} is then yawed by $\gamma = \tan^{-1}(P_z/P_x)$ about the y-axis (Fig. 10) to intersect the individual circle C_{pq} (longitudinal line) on which ${}^dP$ is located. At this intersection, ρ_p^{train} is further pitched by $\beta = \tan^{-1}(P_y/\sqrt{P_x^2 + P_z^2}) - \delta$ about the z-axis to intersect the latitude line of ${}^dP$. These two transformations generate the initial guess for *COIN*, which calculates the global posture from Eq. (7) at ${}^dP$.

A simulation of this process is depicted in Fig. 10, and further visualized in the video file in Moubarak and Ben-Tzvi (2012),[32] where an 11-joint spatial arm is shown tracking a 3D knot trajectory via *COIN*. The initial guess is shown in black lines, which is generated from the intersection of the training direction ξ with the corresponding sphere of radius R_{3D}, followed by a yaw and pitch transformations. These transformations place the end-effector of the training posture at ${}^dP$, and enable *COIN* to calculate the global solution of Eq. (7) (red solid lines) to track a spatial trajectory at a convergence rate (19 ms) similar to the one shown in Fig. 9 for planar trajectories.

6. Adaptive Manipulation

The primary objective of *COIN* is to maintain the dynamic balance of a mobile robot during eccentric manipulation tasks. The choice of the cost function in Eq. (1) fulfills this objective as a secondary task while resolving the arm's redundancy to meet forward kinematic constraints.

In the previous sections, the formulation of this problem and the speed of convergence of *COIN* were presented for planar and spatial applications in the absence of inertial forces and external load disturbances. This is because the objective of the solution was to resolve redundancy to meet the forward kinematic constraints at ${}^dP$ while minimizing the tilting moment generated by the *steady-state* posture of the extended arm.

In sensor-based applications, however, external loads (object weight, inertia, etc.) play an important role in defining the dynamic posture of the arm, and the autonomous position of the mobile platform relative to the end-effector's trajectory to prevent tip-over instability. Indeed, the inertial dynamics of the arm and the external tip load create transient moments that disturb the steady-state gravity-based stable posture of the arm at ${}^dP$, causing the mobile robot to lose balance despite the globally optimal steady-state posture.

The formulation of the optimal problem in Eq. (3) and the ensuing *COIN* algorithm enable the compensation for these transients in a closed-loop scheme. This is achieved by initiating real-time swift control commands that adapt the position of the robot relative to the end-effector's trajectory in order to offset the impact of disturbance. This autonomous repositioning is accompanied by a recalculation of the global arm posture that maintains contact with either the object, or the desired trajectory of the end-effector during a given manipulation task.

This scheme is shown in Fig. 11, where the input-desired end-effector position ${}^dP$ is supplemented by force and joint angle measurements to generate two bundles of control signals. One bundle consists of joint commands, while the other consists of mobility commands that adaptively

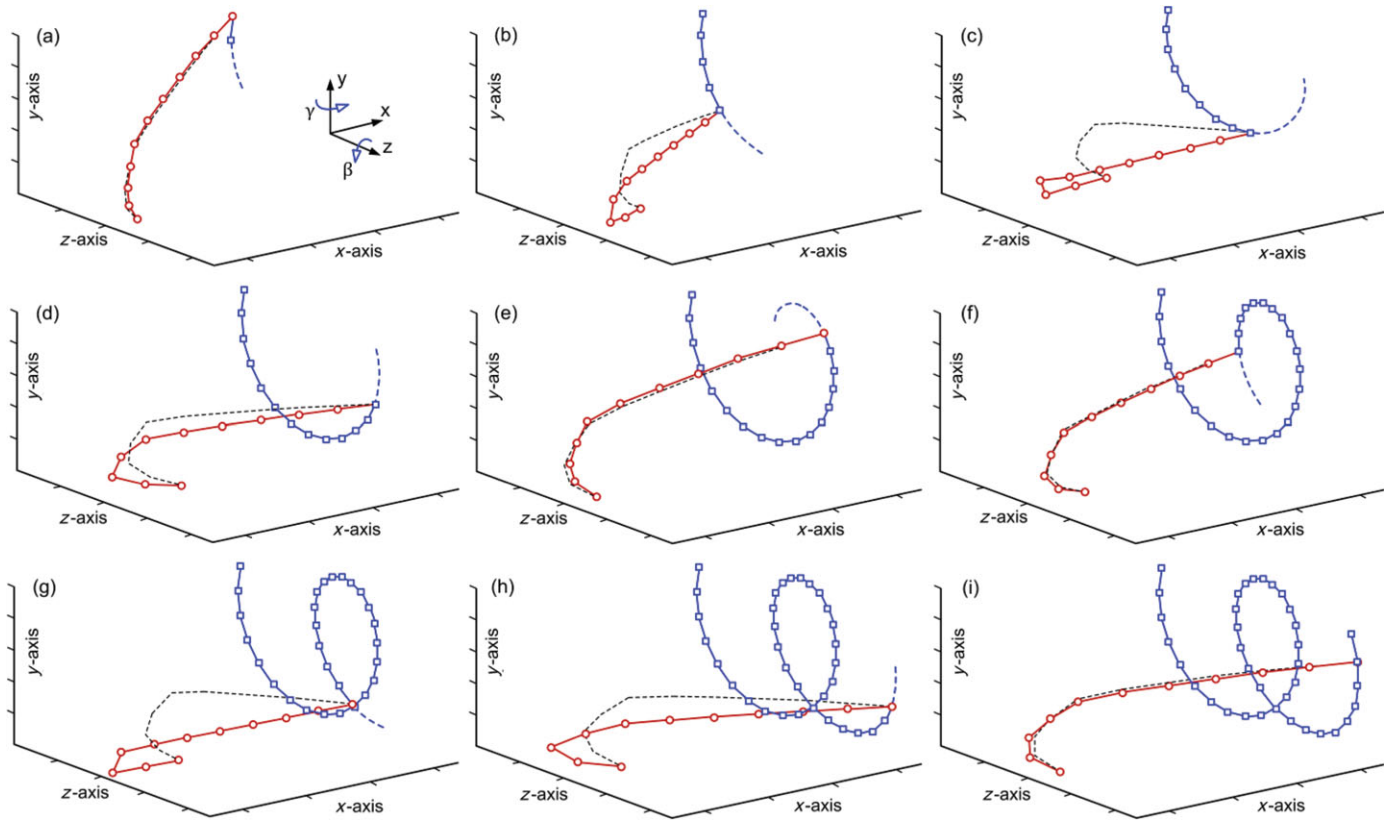

Fig. 10. (Colour online) Simulation of an articulated redundant arm (11 joints) tracking a 3D knot trajectory[32] (black dashed postures represent rotated training data).

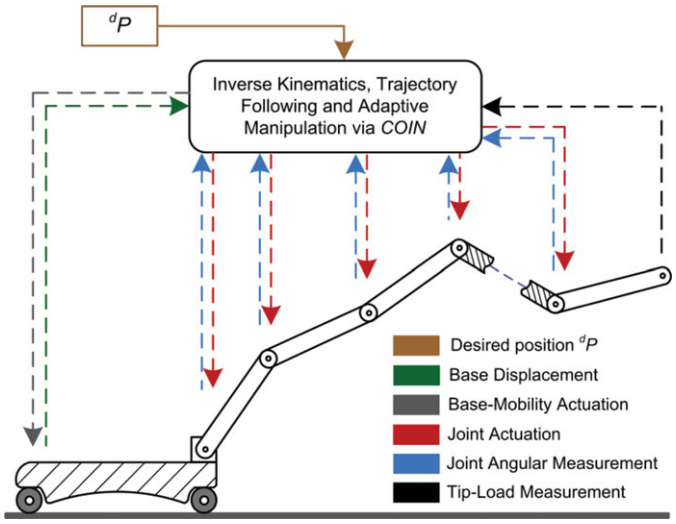

Fig. 11. (Colour online) Schematic illustration showing the integration of *COIN* in a closed-loop control scheme with force and joint angle measurements.

reposition the mobile base relative to the end-effector's position. The initiation of these bundles is based on an extrapolation of the global solution of Eq. (7) to account for external moments, inertial dynamics, and inequality constraints.

6.1. Dynamic stability constraints

During manipulation tasks, the measurement of external loads, in particular the end-effector's force and moment components, are sampled at a rate k. This sampling converts the change in the external tip load into a discrete progression of instances k of external forces ${}^{n}F_{\text{eff}}(k)$ and external moments ${}^{n}\eta_{\text{eff}}(k)$ in the end-effector's frame n. With these measurements fed back to *COIN*, the repositioning of the mobile base and the recalculation of the optimal arm posture are initiated anytime the stability constraint

$$\begin{aligned} M_0 {}^0l_{G_0} \times {}^0g + {}^0\eta_b(k) - {}^0\delta\eta > {}^0\eta(q)\big|_k \\ + \left[{}^d\tilde{P}(k) \left({}^0R_n\right) {}^0R_n \right]_{3\times 6} \begin{bmatrix} {}^nF_{\text{eff}}(k) \in \mathbb{R}^{3\times 1} \\ {}^n\eta_{\text{eff}}(k) \in \mathbb{R}^{3\times 1} \end{bmatrix} \end{aligned} \tag{20}$$

is violated. In Eq. (20), ${}^0\eta_b$ defines the external moments acting on the base (such as friction), ${}^0g$ and ${}^0l_{G_0} = [\, x_0 \; y_0 \; z_0 \,]^T$ denote, respectively, the gravitational acceleration and the position of the base's center of mass, both relative to the base frame 0, ${}^0\eta \in \mathbb{R}^{3\times 1}$ is as defined in Eq. (1), and ${}^0R_n \in \mathbb{R}^{3\times 3}$ 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 $\Sigma(k)$ is calculated according to

$$\begin{aligned} \Sigma(k) = \left[{}^0\tilde{F}_{\text{eff}}(k) \right]^{-1}_{3\times 3} \left[{}^0\eta(q)\big|_k + {}^0\delta\eta + {}^0\eta_{\text{eff}}(k) \right. \\ \left. - {}^0\eta_b(k) - M_0 {}^0l_{G_0} \times {}^0g \right]_{3\times 1} \end{aligned} \tag{21}$$

and the robot is moved by a distance $\Delta(k) = \Sigma(k-1) - \Sigma(k)$ until a stable position is reached from which full-load

manipulation can be achieved with no risk of toppling. In Eq. (21), we note that ${}^0\tilde{F}_{\text{eff}}(k)$ refers to the skew-symmetric matrix of ${}^0F_{\text{eff}}(k)$.

However, as this adaptive mobility is initiated at increments $\Delta(k)$, the acceleration of the robot generates inertial forces and moments that are induced onto the arm. These inertial loads are seen by *COIN* as additional transient external moments, which contribute to the destabilization of the steady-state posture of the arm at ${}^dP$. Such transients are implicitly included in vector ${}^0\eta(q)$ in Eqs. (1), (6), and (21), whose components can be expanded for link i as

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

with the moment generated by the linear acceleration component defined as

$$ {}^0\eta_i^{\text{lin}}(q) = M_i \left\{ {}^0d_{G_i} \times \left({}^0a_{G_i} + a_b \right) \right\}, \tag{23} $$

where the acceleration of the center of mass of link i relative to frame 0 can be written as

$$ \begin{aligned} {}^0a_{G_i} = {}^0\ddot{d}_{C_{i-1}} + {}^0R_{i-1} \big\{ & {}_{i-1}\ddot{q}_i \times {}^{i-1}r_{G_i} + {}_{i-1}\dot{q}_i \\ & \times \left({}_{i-1}\dot{q} \times {}^{i-1}r_{G_i} \right) + 2\,{}_{i-1}\dot{q}_i \times {}^{i-1}v_{G_i} + {}^{i-1}a_{G_i} \big\}. \end{aligned} \tag{24} $$

In Eq. (24), ${}^{i-1}v_{G_i}$ and ${}^{i-1}a_{G_i}$, respectively, define the velocity and acceleration of G_i relative to frame $i-1$ (when they exist, for a prismatic joint). Furthermore, the moment generated by the angular acceleration component can be written as

$$ \begin{aligned} &{}^0\eta_i^{\text{ang}}(q) = {}^0R_{i-1} \left({}^{i-1}I_{G_i} \times {}_{i-1}\ddot{q}_i \right) \text{ for a revolute joint,} \\ &{}^0\eta_i^{\text{ang}}(q) = 0 \text{ for a prismatic joint.} \end{aligned} \tag{25} $$

In Eq. (22), the remaining component ${}^0\eta_i^{\text{per}}(q)$ denotes the vector of any additional peripheral measurable external moment (if applicable). Note that ${}^0\eta_n^{\text{per}}(q) = 0$ for $i = n$ in Eq. (22) since the external forces and moments acting on the end-effector are explicitly isolated in Eqs. (20) and (21).

The linear and angular acceleration components of the arm's links are either measured with dedicated sensors, or calculated via discrete finite difference of joint displacements measured using encoders (with appropriate filtering). In such case, since the adaptive dynamic manipulation is initiated reactively with *COIN*, the discretization of the velocity and acceleration of link i at instance k is written as

$$ \begin{aligned} \dot{q}_i(k) &= \frac{q_i(k) - q_i(k-1)}{T}, \\ \ddot{q}_i(k) &= \frac{q_i(k) - 2q_i(k-1) + q_i(k-2)}{T^2}, \end{aligned} \tag{26} $$

where T defines the sampling period. On the other hand, the linear acceleration components of the mobile base (a_b) can be measured in real-time via an on-board inertial measurement unit.

6.2. Simulation of adaptive manipulation

In Eq. (21), vector $\Sigma(k) = [\, x_d(k)\ y_d(k)\ z_d(k) \,]^T$ defines the 3D displacement that the robot should accomplish in order to offset the risk of toppling induced by a tilting moment vector with spatial components. In practical applications, however, mobile robots typically provide unidirectional mobility with *Hilare*-type or car-like steering. This means that the compensation for the effects of the tilting moment can only be accomplished for the moment's component whose axis is orthogonal to the direction of displacement. For mobile robots with no omni-directional mobility, this component is laterally orthogonal to the direction of motion (i.e., about the z-axis, see Figs. 2 and 3).

Compensation for this moment component is accomplished autonomously via forward mobility during object manipulation tasks, as well as during trajectory following in the presence of tip load, such as object placement. A case-study simulation of dynamic adaptive manipulation via *COIN* is shown in Fig. 12. For simplicity of visualization, we consider the case of eccentric object lifting where the robot lifts a stationary load from an inadequate posture. However, such adaptive manipulation control can be equally extrapolated to other applications, such as object placement along a trajectory.

The first simulated task depicts the mobile robot shown earlier in Fig. 1 (three-joint arm), and consists of lifting a 20 kg payload, while the second task consists of lifting a 10 kg payload with a serial arm of four joints. Both tasks are accomplished in 3 s starting from an inadequate posture of the arm.

During this adaptive manipulation process, the stability condition is evaluated at every instance k of external load measurement. Anytime inequality (20) is violated, a new vector $\Sigma(k)$ is calculated. A control signal is then initiated to move the robot forward by $\Delta(k) = \Sigma(k-1) - \Sigma(k)$ along with its training data, and a new global arm posture is generated by relocating ${}^dP$ on a new corresponding initialization circle. If condition (20) is not violated, the robot remains idle while the arm continues the execution of the object-lifting task until the next measurement is acquired and evaluated. This process is shown in Fig. 12, which depicts the forward mobility of the robot as a function of time.

The flat phases of the robot's displacement in Fig. 12 (right column) visualize the stationary intervals of the forward mobility process, while the ascending slope shows the adaptive repositioning of the mobile base relative to the exponentially increasing external load. Such combination of stationary and dynamic mobility phases continues until the robot approaches a stable posture from which full-load manipulation can be executed with no risk of toppling.

This autonomous mobility is further accompanied by an adaptive reconfiguration of the arm's posture in order to maintain contact with the object. The fast global convergence of *COIN* enables such adaptation at an average computational time of $<$20 ms for every joint reconfiguration. This allows the robot to react swiftly to the risk of toppling in order to maintain its balance throughout the progression of an eccentric manipulation task.

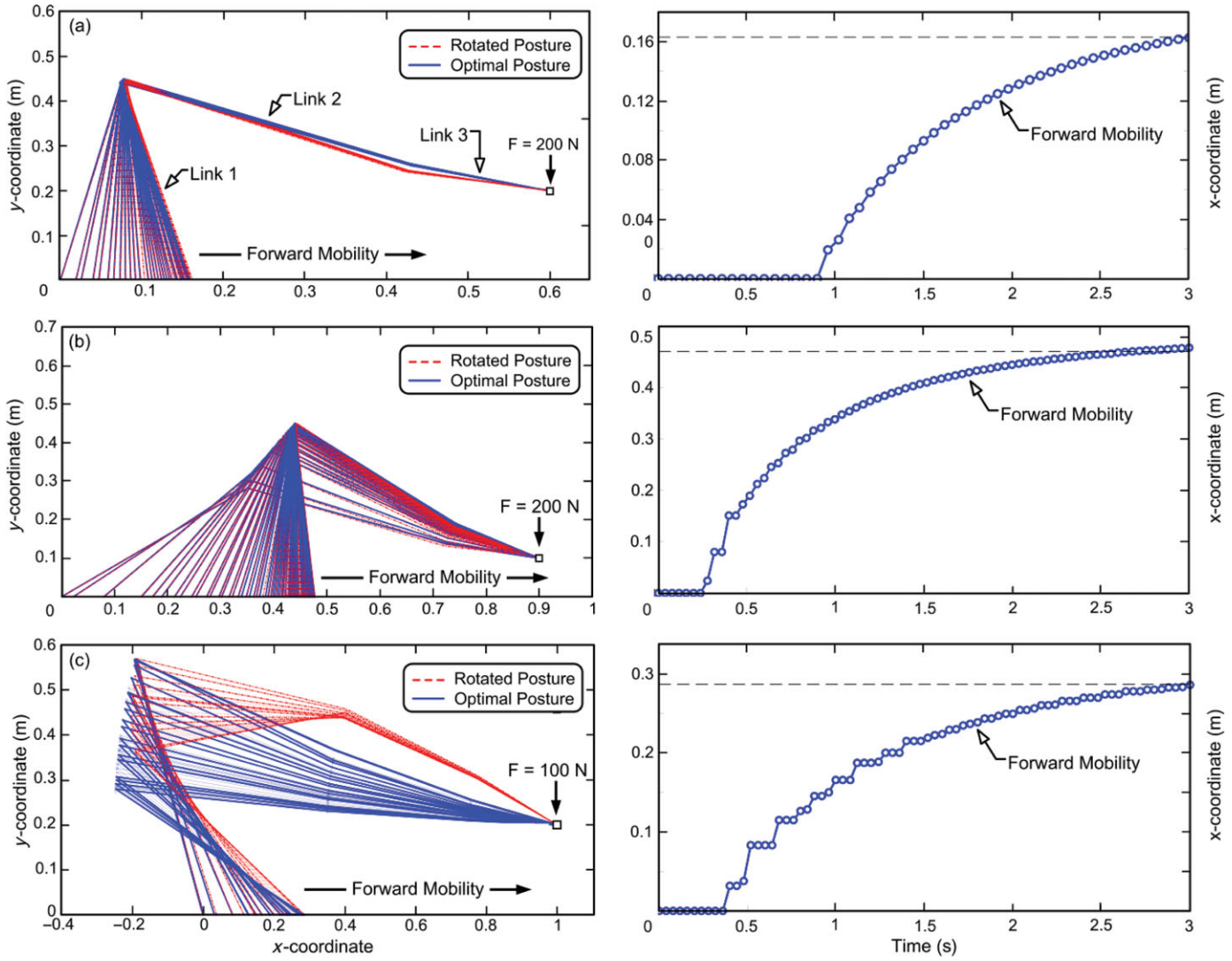

Fig. 12. (Colour online) Simulation of dynamic adaptive manipulation and corresponding forward mobility via *COIN* for (a) three-joint arm in close-proximity, (b) three-joint arm for a distant task, and (c) four-joint arm (red lines are training data translated with the robot).

7. Conclusion

This paper presented a new algorithm called *COIN*, which is characterized by a fast global convergence rate, and can be employed for redundancy resolution, spatial trajectory following, and tip-over stability of mobile robots via dynamic adaptive manipulation in real-time. Simulation results further demonstrated the fast convergence attributes and broader applicability of the proposed algorithm to serial redundant arms with n-joints. These adaptive properties of *COIN* set the basis for our future investigations, where we will further study the impact of the chosen training direction on the convergence and speed of the algorithm, and explore the applicability of the established methods to the adaptive manipulation of modular robots, such as Self-configurable and Transformable Omni-directional Robotic Modules (STORM).[33]

Acknowledgments

This work was supported in part by Defense Advanced Research Projects Agency (DARPA) under grant number HR0011-11-1-0012. We also would like to gratefully acknowledge the support and advice provided by the DARPA Program Manager, Dr. Gill Pratt.

References

1. R. F. Abo-Shanab and N. Sepehr, "On dynamic stability of manipulators mounted on mobile platforms," *Robotica* **19**(4), 439–449 (2001).
2. S. Czarnetzki, S. Kerner and O. Urbann, "Observer-based dynamic walking control for biped robots," *J. Robot. Auton. Syst.* **57**(8), 839–845 (2009).
3. Q. Huang, K. Tanie and S. Sugano, "Coordinated motion planning for a mobile manipulator considering stability and manipulation," *Int. J. Robot. Res.* **19**(8), 732–742 (2000).
4. J. Kim, W. K. Chung, Y. Youm and B. H. Lee, "Real-Time ZMP Compensation Method using Null Motion for Mobile Manipulators," *Proceedings of the IEEE International Conference on Robotics and Automation*, Washington, DC (May 2005).
5. 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," *Int. J. Adv. Manuf. Technol.* **46**, 811–829 (2010).

6. D. A. Rey and E. G. Papadoupoulos, "Online Automatic Tipover Prevention for Mobile Manipulators," *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems* (IROS'97), France (Sep. 1997).
7. A. Meghdari, D. Naderi and M. R. Alam, "Neural-network-based observer for real-time tipover estimation," *J. Mechatronics*, **15**, 989–1004 (2005).
8. M. H. Korayem, H. Ghariblu, M. H. Korayem, H. Ghariblu and A. Basu, "Maximum allowable load on wheeled mobile manipulators imposing redundancy constraints," *J. Robot. Auton. Syst.* **44**(2), 151–159 (2004).
9. M. H. Korayem, A. Nikoobin and V. Azimirad, "Maximum load carrying capacity of mobile manipulators: Optimal control approach," *Robotica* **27**, 147–159 (2009).
10. Z. B. Tang, "Adaptive partitioned random search to global optimization," *IEEE Trans. Autom. Control* **39**(11), 2235–2244 (1994).
11. G. Z. Grudi and P. D. Lawrence, "Iterative inverse kinematics with manipulator configuration control," *IEEE Trans. Robot. Autom.* **9**(4), 476–483 (1993).
12. S. Sasaki, "On numerical techniques for kinematic problems of general serial-link robot manipulators," *Robotica* **12**(4), 309–322 (1994).
13. J. Lenarcic, "Effective secondary task execution of redundant manipulators," *Robotica* **16**(4), 457–462 (1998).
14. Z. Zhang and Y. Zhang, "Variable joint-velocity limits of redundant robot manipulators handled by quadratic programming," *IEEE/ASME Trans. Mechatronics* **18**(2), 674–686 (2013).
15. P. P. Kumar and L. Behera, "Visual servoing of redundant manipulator with jacobian matrix estimation using self-organizing map," *J. Robot. Auton. Syst.* **58**(8), 978–990 (2010).
16. A. A. Goldenberg and R. G. Fenton, "A null-space solution of the inverse kinematics of redundant manipulators based on a decomposition of screws," *J. Mech. Design* **115**(3), 530–539 (1993).
17. F. Caccavale, S. Chiaverini and B. Sicillano, "Second-order kinematic control of robot manipulators with Jacobian damped least-squares inverse: Theory and experiments," *IEEE/ASME Trans. Mechatronics* **2**(3), 188–194 (1997).
18. K. Tchoń, "Repeatable, extended Jacobian inverse kinematics algorithm for mobile manipulators," *Syst. Control Lett.* **55**(2), 87–93 (2006).
19. E. Zergeroglu, D. D. Dawson, I. W. Walker and P. Setlur, "Nonlinear tracking control of kinematically redundant robot manipulators," *IEEE/ASME Trans. Mechatronics* **9**(1), 129–132 (2004).
20. L. Sciavicco and B. Siciliano, "A solution algorithm to the inverse kinematic problem for redundant manipulators," *IEEE Trans. Robot. Autom.* **4**(4), 403–410 (1988).
21. W. J. Chung, W. K. Chung and Y. Youm, "Kinematic control of planar redundant manipulators by extended motion distribution scheme," *Robotica* **10**(3), 255–262 (1992).
22. F. Fahimi, *Autonomous Robots, Modeling, Path Planning and Control* (Springer, New York, 2008) ch. 2.
23. G. S. Chirikjian and J. W. Burdick, "A modal approach to hyper-redundant manipulator kinematics," *IEEE Trans. Robot. Autom.* **10**(3), 343–354 (1994).
24. S. Kumar, P. Premkumar, A. Dutta and L. Behera, "Visual motor control of a 7-DOF redundant manipulator using redundancy preserving learning network," *Robotica* **28**(6), 795–810 (2010).
25. P. Moubarak and P. Ben-Tzvi, "Field Testing and Adaptive Manipulation of a Hybrid Mechanism Mobile Robot," *Proceedings of the IEEE International Symposium on Robotic and Sensor Environments (ROSE 2011)*, Montreal, Canada (Sep. 2011).
26. P. Ben-Tzvi, "Experimental validation and field performance metrics of a hybrid mobile robot mechanism," *J. Field Robot.* **27**(3), 250–267 (2010).
27. P. Ben-Tzvi, A. A. Goldenberg and J. W. Zu, "Design and analysis of a hybrid mobile robot mechanism with compounded locomotion and manipulation capability," *J. Mech. Des.* **130**(7), 1–13 (2008).
28. P. Ben-Tzvi, A. A. Goldenberg and J. W. Zu, "Articulated hybrid mobile robot mechanism with compounded mobility and manipulation and on-board wireless sensor/actuator control interfaces," *Mechatronics J.* **20**(6), 627–639 (2010).
29. J. A. Carretero and M. A. Nahon, "Solving minimum distance problems with convex or concave bodies using combinatorial global optimization algorithms," *IEEE Trans. Syst. Man Cybern.* **35**(6), 1144–1155 (2005).
30. A. E. Bryson and Y. C. Ho, *Applied Optimal Control: Optimization, Estimation, and Control*, rev. ed. (Hemisphere, Washington, DC, 1975) ch. 1.
31. U. Zsolt, L. Lasdon, J. C. Plummer, F. Glover, J. Kelly and R. Martí, "Scatter search and local NLP solvers: A multi-start framework for global optimization," *J. Comput.* **19**(3), 328–340 (2007).
32. P. Moubarak, P. Ben-Tzvi, "Spatial trajectory following with COIN algorithm," available at: http://www.seas.gwu.edu/∼bentzvi/COIN/COIN.html (2012) (accessed April, 2012).
33. P. Moubarak, P. Ben-Tzvi, "STORM Animation," available at: http://www.seas.gwu.edu/∼bentzvi/STORM/STORM_VR_Animation.html (Feb. 2011) (accessed April, 2012).