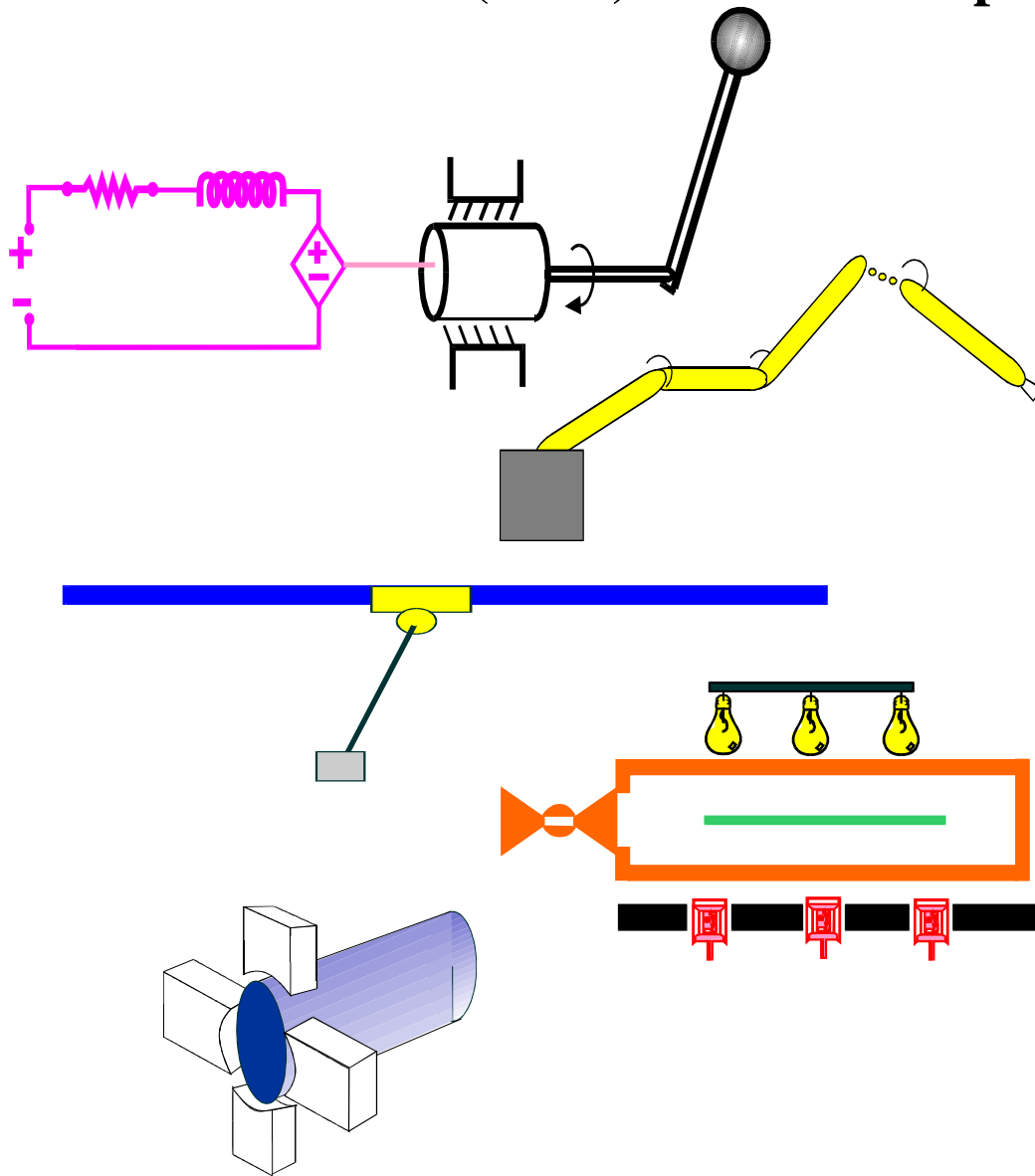


**Clemson University**  
**College of Engineering and Science**  
**Control and Robotics (CRB) Technical Report**



Number: CU/CRB/10/12/05/#1

Title: Whole Arm Grasping Control for Redundant Robot  
Manipulators

Authors: D. Braganza, M. L. McIntyre, D. M. Dawson and I.  
Walker

# Whole Arm Grasping Control for Redundant Robot Manipulators

D. Braganza, M. L. McIntyre, D. M. Dawson and I. Walker

**Abstract**—An approach to whole arm grasping of objects using redundant robot manipulators is presented. A kinematic control development is presented which facilitates the encoding of both the end-effector position, as well as body self-motion positioning information as a desired trajectory signal for the manipulator joints. A joint space controller which provides asymptotic tracking of the encoded desired trajectory in the presence of system uncertainties is then presented. Experimental results for a planar, three link configuration of the Barrett whole arm manipulator are provided to illustrate the validity of the approach.

## I. INTRODUCTION

One of the advantages of redundant robot manipulators is their ability to perform whole arm grasping of objects. Whole arm manipulation [21] is the term used to describe the ability of the manipulator to grasp an object with its entire body (or arm), as compared to fingertip grasping performed by traditional robotic grippers and hands. Whole arm grasping<sup>1</sup> can be performed by allowing the robot manipulator to make contact with the object in a snake or tentacle like manner, using portions of the manipulator itself to wrap around the object and grasp it. The equivalent whole hand and whole finger grasping techniques have been studied in [14] and [20], respectively. Whole arm grasping is also known by the equivalent expressions “power grasping” ([15] and [22]) or “enveloping grasping” [25]. Recently in [19], the authors presented experimental results which demonstrated whole hand grasping with a 12 degree-of-freedom (DOF) robotic hand. However, there has been very little experimental work reported on whole arm grasping with redundant robot manipulators. Specifically, one of the few results in the literature is given in [7] where whole arm grasping with a 30 DOF robotic arm is demonstrated. Shape control is another technique which is being studied for the control of whole arm grasping. In [16], the author proposed an impedance control based approach to control the shape of the whole arm of a redundant manipulator. Whole arm grasping has a number of useful properties as noted by [1], [15], [21], and others. The authors of [15] point out that distribution of contact points enables increased load capacity. The ability to use the entire body of the manipulator for grasping also allows objects of various dimensions to be grasped [21]. These capabilities can

be used in many applications, including, search and rescue, underwater and space exploration.

Traditional robotic grasping control can be broadly classified into two main categories [18]. The first category, known as the geometrical planning based approach, requires the object model and the constraint forces to be known *a priori* (e.g. [1] and [23]). Here, the grasping contact points are pre-planned and the desired constraint force for each contact point are assumed to be known. The grasping control system then moves the hand/arm along a pre-determined trajectory and force feedback (force sensors on the arm/hand) is used to control the interaction forces. The second category for robot grasping control is the sensory approach, where the object model is unknown and the grasping controller relies on tactile force-feedback data. In this sensory based approach, it is often assumed that the arm has a sensory “skin” for force measurements [2]. The arm/hand must either start off close to the object to be grasped, or with all contact points touching the object. Then, the grasp controller positions and re-positions the arm to minimize an error function in an attempt to optimize the grasp configuration [19].

The techniques described above require either that the geometry of the object and the constraint forces be known *a priori* [23], or that the contact forces be measurable using some type of force sensor [1], [2], and [18]. When extending the traditional approaches (i.e., fingertip grasping) to whole-arm grasping, the previously mentioned requirements might not be easily met due to the increased number of contact points and the large number of grasping configurations possible [19]. Motivated by the need to have a whole arm grasping controller which does not require the constraint forces to be known *a priori* while also eliminating the requirement for contact force sensing (due to the inherent inaccuracy/noise in measurement), a grasping controller for redundant robot manipulators is designed which requires only the object geometry to be known *a priori*. In addition, the proposed controller does not require the exact dynamic model for the robot manipulator or the contact forces. This paradigm makes the whole arm grasping technique easily extendable to various manipulator systems.

Roughly speaking, the whole arm grasping objective is achieved by integrating the path planner and the controller such that two tasks, robot end-effector positioning and robot body self-motion positioning, are accomplished simultaneously. The end-effector positioning controller forces the end-effector to follow a path around the object which in turn, forces the robot’s body to wrap itself around the object to be grasped. The body self-motion positioning controller “repels” the body of the manipulator away from the object while the

This work is supported in part by two DOC Grants, an ARO Automotive Center Grant, a DOE Contract, a Honda Corporation Grant, and a DARPA Contract.

The authors are with the Department of Electrical and Computer Engineering, Clemson University, Clemson, SC 29634-0915. dbragan@clemson.edu

<sup>1</sup>For an overview of robotic grasping and manipulation, the reader is referred to [3] and the references therein.

end-effector moves around the object. This control-induced repulsion-like property facilitates object avoidance as well as removes the “slack” from the robot body as the robot begins to move into the grasping position. When all possible slack is removed, the manipulators body makes contact with the object, hence, completing the whole arm grasp of the object.

To facilitate the explanation of the proposed whole arm grasping control design, we first develop a Lyapunov-based kinematic control. The kinematic control input is then passed through a desired trajectory filter which produces a desired, joint level trajectory. The smooth control strategy developed in [26] is then utilized for the joint space controller since it provides asymptotic tracking of the desired trajectory in the presence of dynamic uncertainties. Experimental results for a planar application of the whole arm grasping technique using the Barrett whole arm manipulator are provided to illustrate the performance of the controller.

## II. KINEMATIC AND DYNAMIC MODELS

In this section the kinematic and dynamic models for an  $n$ -joint ( $n \geq 6$ ), revolute, direct drive robot manipulator are presented. The subsequent development is based on these models.

### A. Kinematic Model

The Denavit-Hartenberg based forward kinematic model for an  $n$ -segment redundant manipulator can be developed as follows

$$x_n = f_n(q) \quad (1)$$

where  $x_n(t) \in \mathbb{R}^p$  represents the robot end-effector’s task-space vector,  $q(t) \in \mathbb{R}^n$  denotes the joint position, and  $f_n(q) \in \mathbb{R}^p$  denotes the forward kinematics of the manipulator. The velocity kinematics for the manipulator can be developed as follows

$$\dot{x}_n = J_n(q)\dot{q}(t) \quad (2)$$

where  $\dot{x}_n(t) \in \mathbb{R}^p$  represents the task-space velocity,  $\dot{q}(t) \in \mathbb{R}^n$  denotes the joint velocity, and  $J_n(q) \triangleq \frac{\partial f_n(q)}{\partial q} \in \mathbb{R}^{p \times n}$  denotes the manipulator Jacobian.

### B. Dynamic Model

The dynamic model for an  $n$ -joint ( $n \geq 6$ ), revolute, direct drive robot manipulator is described by the following expression [8]

$$M(q)\ddot{q} + N(q, \dot{q}) + F_e(q, \dot{q}) = \tau \quad (3)$$

where  $M(q) \in \mathbb{R}^{n \times n}$  represents the inertia effects,  $N(q, \dot{q}) \in \mathbb{R}^n$  represents the remaining dynamic terms, such as the centripetal-Coriolis effects, gravitational effects, and frictional effects,  $F_e(q, \dot{q}) \in \mathbb{R}^n$  represents the contact forces placed on the robot manipulator by the environment,  $\tau(t) \in \mathbb{R}^n$  represents the input torque vector. The subsequent development is based on the assumptions that  $q(t)$  and  $\dot{q}(t)$  are measurable,  $M(q)$ ,  $N(q, \dot{q})$ , and  $F_e(q, \dot{q})$  are unknown, second order differentiable, functions of  $q(t)$  and  $\dot{q}(t)$ , and the following property holds [8],

**Property 1:** The inertia matrix  $M(q)$  is symmetric and positive-definite, and satisfies the following inequalities

$$m_1 \|\xi\|^2 \leq \xi^T M(q)\xi \leq m_2 \|\xi\|^2 \quad \forall \xi \in \mathbb{R}^n \quad (4)$$

where  $m_1, m_2 \in \mathbb{R}$  are positive constants, and  $\|\cdot\|$  denotes the standard Euclidean norm.

*Remark 1:* Since this development is only concerned with revolute robot manipulators, the kinematic and dynamic terms denoted by  $M(q)$ ,  $N(q, \dot{q})$ , and  $J(q)$ , are assumed to be bounded for all possible  $q(t)$  (i.e., these kinematic and dynamic terms only depend on  $q(t)$  as arguments of trigonometric functions).

## III. GRASPING WITH KINEMATIC CONTROL

To facilitate the kinematic control development, the pseudo-inverse of  $J_n(q)$  denoted by  $J_n^+(q) \in \mathbb{R}^{n \times p}$ , is defined as follows

$$J_n^+ \triangleq J_n^T (J_n J_n^T)^{-1} \quad (5)$$

where  $J_n^+(q)$  satisfies the following equality

$$J_n J_n^+ = I_p \quad (6)$$

where  $I_p \in \mathbb{R}^{p \times p}$  is the standard identity matrix. As shown in [17], the pseudo-inverse defined by (5) satisfies the Moore-Penrose conditions given below

$$\begin{aligned} J_n J_n^+ J_n &= J_n & J_n^+ J_n J_n^+ &= J_n^+ \\ (J_n^+ J_n)^T &= J_n^+ J_n & (J_n J_n^+)^T &= J_n J_n^+ \end{aligned} \quad (7)$$

In addition to the above properties, the matrix  $(I_n - J_n^+ J_n)$  satisfies the following useful properties

$$\begin{aligned} (I_n - J_n^+ J_n)(I_n - J_n^+ J_n) &= I_n - J_n^+ J_n \\ (I_n - J_n^+ J_n)^T &= (I_n - J_n^+ J_n) \\ J_n (I_n - J_n^+ J_n) &= 0 \\ (I_n - J_n^+ J_n) J_n^+ &= 0 \end{aligned} \quad (8)$$

where  $I_n \in \mathbb{R}^{n \times n}$  is the standard identity matrix.

*Remark 2:* During the control development, the assumption is made that the minimum singular value of the manipulator Jacobian, denoted by  $\sigma_m$  is greater than a known, small positive constant  $\delta > 0$ , such that  $\max\{\|J_n^+(q)\|\}$  is known a priori and all kinematic singularities are always avoided.

Typically in the robotics literature, when a kinematic control is designed,  $\dot{q}(t)$  is taken to be the control input. A joint space controller must then be used to ensure that the actual robot joint angles track this reference trajectory. Following this paradigm, the kinematic controller is first designed as follows

$$\dot{q}(t) \triangleq J_n^+ U_e + (I_n - J_n^+ J_n) U_m \quad (9)$$

where  $U_e(t) \in \mathbb{R}^p$  is the *end-effector positioning* controller, and  $U_m(t) \in \mathbb{R}^n$  is the *robot body self-motion* controller. In the subsequent sub-sections, the design of the robot *end-effector positioning* controller  $U_e(t)$  and the *robot body self-motion* controller  $U_m(t)$  will be discussed in detail.

### A. End-Effector Positioning

The objective of the *end-effector positioning* controller is to force the end-effector to track a desired trajectory that encompasses the surface of the object to be grasped. For this type of problem, instead of a time based trajectory, a velocity field control (VFC) is utilized because it more effectively penalizes the end-effector for leaving the contour ([6], [9], and [10]). The VFC will also not exhibit the *radial reduction* phenomenon which is common with traditional control methods ([6] and [9]). For example, when the object to be grasped is circular, the velocity field generates a desired trajectory that forces the end-effector to spiral inwards, toward and around the surface of the object.

*Remark 3:* A velocity field specifies a desired velocity  $\dot{x}_d(t)$  at each displacement position  $x_n(t)$  on the task space of the system [9]. In [9] and [10], the authors provide specific information about the construction of velocity fields. See [6] and [12] for details of circular velocity fields. The velocity field for a specific planar application is presented subsequently in the Appendix.

The *end-effector positioning* controller  $U_e(t) \in \mathbb{R}^p$  is designed as follows

$$U_e \triangleq \vartheta(x_n) + K_e e + k_n \left\| \frac{\partial V(x_d)}{\partial x_d} \right\|^2 \rho^2(x_n, x_d) e \quad (10)$$

where  $\vartheta(x_n) \in \mathbb{R}^p$  is a task-space velocity field,  $K_e \in \mathbb{R}^{p \times p}$  is a positive definite diagonal gain matrix,  $k_n \in \mathbb{R}^+$  is a scalar gain parameter,  $e(t) \in \mathbb{R}^p$  is the error between the desired and actual task space position and is defined as follows

$$e \triangleq x_d - x_n, \quad (11)$$

where  $x_d(t) \in \mathbb{R}^p$  is the desired task-space position, and  $x_n(t)$  was introduced in (1). In (10),  $V(x_d) \in \mathbb{R}$  is a first order differentiable, nonnegative function and  $\rho(\cdot) \in \mathbb{R}$  is a known positive function that is assumed to be bounded provided  $x_n(t)$  and  $x_d(t)$  are bounded. The function  $V(x_d)$  is defined for a given application subsequently in Section I. For details on how to construct  $\rho(x_n, x_d)$  for a specific application, the reader is referred to [12].

For the whole arm grasping control objective, the desired task space velocity trajectory is defined as

$$\dot{x}_d(t) \triangleq \vartheta(x_n) \quad (12)$$

where  $\vartheta(x_n)$  is the velocity field generated by the task-space position  $x_n(t)$ . The velocity tracking error signal can be derived by taking the first derivative of (11) and using (12), we have

$$\dot{e} = \vartheta(x_n) - \dot{x}_n. \quad (13)$$

After utilizing (2), the expression in (13) can be written as follows

$$\dot{e} = \vartheta(x_n) - J_n \dot{q}. \quad (14)$$

After utilizing (9), the expression in (14) can be written as follows

$$\dot{e} = \vartheta(x_n) - U_e \quad (15)$$

where (6) and (8) has been used. After substituting for  $U_e(t)$ , as defined in (10), the expression in (15) can be written as follows

$$\dot{e} = -K_e e - k_n \left\| \frac{\partial V(x_d)}{\partial x_d} \right\|^2 \rho^2(x_n, x_d) e. \quad (16)$$

*Theorem 1:* The control law described by (10) guarantees that  $e(t)$ ,  $\dot{e}(t)$  and  $U_e(t) \in \mathcal{L}_\infty$  and  $\|e(t)\| \rightarrow 0$  as  $t \rightarrow \infty$ .

*Proof:* See [12], which contains a similar result. ■

The result of Theorem 1 proves that  $\|e(t)\| \rightarrow 0$  as  $t \rightarrow \infty$  and that  $U_e(t) \in \mathcal{L}_\infty$ . Thus, the control law defined in (10) guarantees that the manipulators end-effector follows the desired contour while also ensuring that all signals remain bounded. If the controller defined in (10) is used alone (i.e.  $\dot{q}(t) = J_n^+ U_e$ ), the joint space desired trajectory that is tracked may take a path such that the end-effector and body of the manipulator may make contact with the object while the end-effector tries to follow the contour of the object to be grasped. Since this is an undesirable effect, the *robot body self-motion positioning* controller is designed in such a manner that provides object avoidance as the body of the manipulator moves around the object to be grasped.

### B. Body Self-Motion Positioning

The objective of the *body self-motion positioning* controller is to “repel” the end-effector and body of the manipulator away from the object to be grasped, while the end-effector moves around the object. This control-induced repulsion-like property not only facilitates obstacle avoidance but also removes the “slack” from the body of the manipulator as the robot moves into the grasping position. When all possible slack is removed, the manipulator body makes contact with the object, hence completing the whole arm grasp of the object. Following this line of reasoning, the *body self-motion positioning* controller  $U_m(t) \in \mathbb{R}^n$  in (9), is designed as follows

$$U_m \triangleq -k_m [J_s (I_n - J_n^+ J_n)]^T y_a \quad (17)$$

where  $k_m \in \mathbb{R}^+$  is a control gain,  $J_s \in \mathbb{R}^{1 \times n}$  is a subsequently designed Jacobian-like vector,  $I_n \in \mathbb{R}^{n \times n}$  was defined in (8), and  $y_a(t) \in \mathbb{R}$  is an auxiliary scalar signal which is yet to be defined. The signal  $y_a(t)$  encodes the geometric information about the object’s surface and how it relates to the manipulator’s joint positions in an effort to keep the body of the manipulator away from the object. See [24] for details of a general auxiliary signal for self-motion control of a redundant robot manipulator.

For whole arm grasping, a specific auxiliary signal  $y_a(t)$  is designed as follows

$$y_a \triangleq \sum_{i=1}^n h_{ai}(x_i) \quad (18)$$

where  $n$  is the number of joints of the redundant manipulator,  $x_i = [\bar{x}_{i1} \ \bar{x}_{i2} \ \dots \ \bar{x}_{ip}]^T \in \mathbb{R}^p$  is the Euclidean-space coordinate for the  $i^{th}$  joint, and  $h_{ai}(x_i) \in \mathbb{R}$  is the repulsion function for the  $i^{th}$  joint that encodes the geometric information about the surface of the object with respect to the

$i^{th}$  joint's Euclidean position. The repulsion function  $h_{ai}(x_i)$  is defined as follows

$$h_{ai}(x_i) = k_{hi} \exp(-\alpha_i \beta_i^2(x_i)), \quad \forall i = 1, \dots, n \quad (19)$$

where  $k_{hi}$ ,  $\alpha_i \in \mathbb{R}^+$  are constants, and  $\beta_i(x_i) \in \mathbb{R}$  is the joint specific geometric function. The function  $\beta_i(x_i)$  should be designed to be positive when the manipulator is not touching the object as well as that  $\beta_i(x_i) \in \mathcal{L}_\infty$ , if  $x_i(t) \in \mathcal{L}_\infty$ . For example, given a spherical object in three dimensional Euclidean-space,  $\beta_i(x_i)$  could be defined as follows

$$\beta_i(x_i) \triangleq (\bar{x}_{i1} - \bar{x}_{c1})^2 + (\bar{x}_{i2} - \bar{x}_{c2})^2 + (\bar{x}_{i3} - \bar{x}_{c3})^2 - r_o^2$$

where  $\bar{x}_{c1}$ ,  $\bar{x}_{c2}$ ,  $\bar{x}_{c3}$ ,  $r_o \in \mathbb{R}$  are the Euclidean coordinates of the center of the spherical object and its radius, respectively.

To determine the dynamics of  $y_a(t)$ , the time derivative of (18) is taken, and can be written as follows

$$\dot{y}_a = J_s \dot{q} \quad (20)$$

where a Jacobian-type vector  $J_s(t) \in \mathbb{R}^{1 \times n}$  is defined as follows

$$J_s = \frac{\partial y_a(x_1, x_2, \dots, x_n)}{\partial [x_1^T, x_2^T, \dots, x_n^T]} \begin{bmatrix} J_1 \\ \vdots \\ J_n \end{bmatrix} \quad (21)$$

where  $\dot{x}_i = J_i \dot{q}$  and  $J_i \in \mathbb{R}^{p \times n}$  is the Jacobian matrix relating the joint velocities and the Euclidean velocities for the  $i^{th}$  joint. Using (9) and substituting for  $\dot{q}(t)$  in (20), the expression for  $\dot{y}_a(t)$  can be written as follows

$$\dot{y}_a = J_s J_n^+ U_e + J_s (I_n - J_n^+ J_n) U_m. \quad (22)$$

After substituting for  $U_m(t)$  as defined in (17),  $\dot{y}_a(t)$  of (22) can be further expressed as

$$\dot{y}_a = -k_m \|J_s (I_n - J_n^+ J_n)\|^2 y_a + J_s J_n^+ U_e. \quad (23)$$

*Theorem 2:* The control law described by (17) guarantees that  $y_a(t)$  is practically regulated (i.e., ultimately bounded) in the following sense

$$|y_a(t)| \leq \sqrt{|y_a(t_0)|^2 \exp(-2\mu t) + \frac{\omega}{\mu}} \quad (24)$$

provided the following sufficient conditions are true

$$\|J_s (I_n - J_n^+ J_n)\|^2 > \bar{\delta} \quad (25)$$

and

$$k_m > \frac{1}{\bar{\delta} \delta_2} \quad (26)$$

where  $\omega$ ,  $\mu$ ,  $\bar{\delta}$ ,  $\delta_2 \in \mathbb{R}$  are positive constants.

*Proof:* See [24], which has a similar result. ■

*Remark 4:* From (18) and (19), it is clear that  $0 < y_a(t) \leq \sum_{i=1}^n k_{hi}$  and that as  $\beta_i(\cdot)$  increases,  $h_{ai}(t)$  decreases, and hence,  $y_a(t)$  decreases. In addition each  $\beta_i(\cdot)$  is designed such that  $\beta_i(\cdot) > 0$  if the manipulators links are outside the object. From (24), it can be shown that the initial conditions of the manipulator and the bounding constants can be selected such that  $y_a(t) < \sum_{i=1}^n k_{hi}$ , hence, it is clear from (18) and (19) that  $\beta_i(t) > 0 \forall t$ .

The result of Theorem 2 illustrates that the repulsive term  $y_a(t)$  can be bounded by an exponentially decreasing function. This means that when all the manipulators links are in contact with the object the auxiliary repulsion function  $y_a(t)$  will approach a constant value ( $\sum_{i=1}^n k_{hi}$ ), hence  $\beta_i(t) \approx 0$ . Interestingly, as the slack in the robot body is removed, the effect of the control term  $U_m(t)$  is automatically reduced. This is because as the manipulator links make contact with the object, the number of redundant degrees of freedom available to accomplish the task space objective reduces. As a consequence, the self-motion component of the control input becomes almost zero (i.e., the null space projection  $\|(I_n - J_n^+ J_n)\|$  approaches zero), and hence, (25) is no longer satisfied.

#### IV. GRASPING WITH DYNAMIC CONTROL

In the previous section, a kinematic control development was presented which enabled the whole arm grasping objective to be encoded as a desired trajectory signal which can be fed to the subsequently designed joint space tracking controller. This desired trajectory signal encodes information from the two auxiliary controllers, the *end-effector positioning* controller, and the *body self-motion positioning* controller. In the subsequent section, the kinematic control will be utilized to generate a bounded desired joint trajectory such that its higher order derivatives are also bounded.

##### A. Desired Trajectory Generator

Traditionally for torque based control, the desired trajectory and its higher order derivatives are required for the control implementation. It is assumed that the desired trajectory and its higher order derivatives are always bounded for this problem to be tractable. In this section, a desired trajectory filter which generates bounded desired joint space trajectories for the joint space tracking controller is provided. The structure of the desired trajectory generator is motivated by the choice of the joint space controller [26], which is a continuous, nonlinear integral feedback controller and requires the desired trajectory to be bounded upto its fourth derivative. This controller was selected because of its ability to meet the tracking objective in the presence of system uncertainties (i.e. uncertainty in the robot dynamic model and unmeasurable contact forces).

To ensure that the desired joint space velocity trajectory is bounded, we could use the following expression

$$\dot{q}_d(t) \triangleq \text{sat}(\text{RHS of (9)}) \quad (27)$$

where RHS denotes the right hand side of the equation,  $\text{sat}(\xi) \in \mathbb{R}^n$  is defined as  $\text{sat}(\xi) = [\text{sat}(\xi_1), \text{sat}(\xi_2), \dots, \text{sat}(\xi_n)]^T \forall \xi = [\xi_1, \xi_2, \dots, \xi_n]^T \in \mathbb{R}^n$  where  $\text{sat}(\xi_i) \in \mathbb{R} \forall i = 1, \dots, n$  is the following saturation function

$$\text{sat}(\xi_i) = \begin{cases} -\xi_{min} & \text{if } \xi_i \leq -\xi_{min} \\ \xi_i & \text{if } \xi_i > -\xi_{min} \text{ or } \xi_i < \xi_{max} \\ \xi_{max} & \text{if } \xi_i \geq \xi_{max} \end{cases}$$

where  $\xi_{min}, \xi_{max} \in \mathbb{R}^+$  are constants. If (27) is used to generate the desired trajectory, we cannot prove that  $q_d(t)$  is bounded, so we could use the following filtering operation

$$q_d(s) \triangleq \frac{1}{\left(\frac{s}{\epsilon} + 1\right)} sat(\text{RHS of (9)}) \quad (28)$$

where  $s \in \mathbb{C}$  is the standard Laplace variable, and  $\epsilon \in \mathbb{R}^+$  is an integration constant selected very close to zero. However, in the case of (28), we cannot prove that the higher order derivatives of  $q_d(t)$  will remain bounded. So the desired trajectory  $q_d(t)$  for the manipulator joint angles are generated by the following expression

$$q_d(s) \triangleq \frac{1}{\left(\frac{s}{\epsilon} + 1\right) \left(\frac{s}{\kappa} + 1\right)^3} sat(\text{RHS of (9)}) \quad (29)$$

where  $\kappa \in \mathbb{R}^+$  is an integration constant selected to be very large. From (29), it is clear that  $q_d(t)$ ,  $\dot{q}_d(t)$ ,  $\ddot{q}_d(t)$ ,  $\dddot{q}_d(t)$ , and  $\dots \ddot{q}_d(t) \in \mathcal{L}_\infty$ .

### B. Control Objective

The objective of the closed-loop system is to ensure asymptotic tracking between the manipulator and the desired trajectory in the sense that

$$q(t) \rightarrow q_d(t) \text{ as } t \rightarrow \infty \quad (30)$$

where  $q_d(t) \in \mathbb{R}^n$  is obtained from (29). To quantify the control objective, an error signal  $e_1(t) \in \mathbb{R}^n$  is defined as follows

$$e_1 \triangleq q_d - q. \quad (31)$$

Furthermore, a tracking error signal  $e_2(t) \in \mathbb{R}^n$  is defined as follows

$$e_2 \triangleq \dot{e}_1 + \gamma_1 e_1 \quad (32)$$

where  $\gamma_1 \in \mathbb{R}^+$  is a control gain.

### C. Control Law

Since the robot dynamic model is a nonlinear uncertain multi-input multi-output system, the strategy developed in [26] can be used for the continuous joint space controller. The control objective of (30) can be met with the following controller [26]

$$\begin{aligned} \tau \triangleq & (K_s + I_n) \left[ e_2(t) - e_2(t_0) + \gamma_2 \int_{t_0}^t e_2(\tau) d\tau \right] \\ & + \int_{t_0}^t [\Gamma sgn(e_2(\tau))] d\tau \end{aligned} \quad (33)$$

where  $\tau(t) \in \mathbb{R}^n$  is the control input defined in (3),  $K_s$ ,  $\Gamma \in \mathbb{R}^{n \times n}$  are positive diagonal control gain matrices, and  $sgn(\cdot) \in \mathbb{R}^n$  denotes the vector signum function defined as  $sgn(\xi) = [sgn(\xi_1), sgn(\xi_2), \dots, sgn(\xi_n)]^T \forall \xi = [\xi_1, \xi_2, \dots, \xi_n]^T \in \mathbb{R}^n$ . The controller presented in (33), provides asymptotic convergence of the joint tracking error, i.e.  $\|e_1(t)\| \rightarrow 0$  as  $t \rightarrow \infty$ . For a detailed analysis of the controller the reader is referred to [26].

*Remark 5: The trajectory generator defined in (29) generates a filtered version of (9). This filtered signal is used*

*as a desired trajectory for the joint space controller defined in (33). The joint space controller (33), forces the actual robot joint angles to track the filtered desired trajectory of (29). However, we cannot show that the actual robot joint velocities track the kinematic velocity signal defined in (9). Thus, the results of Theorem 1 and Theorem 2 may not be technically valid, however, the validity of the approach is illustrated through the experimental results presented in the Appendix.*

## V. CONCLUSION

A whole arm grasping controller for redundant robot manipulators was presented. A kinematic control which enables end-effector position tracking information as well as body self-motion positioning control information to be encoded in the desired trajectory was developed. Then, a tracking controller, developed in [26], which forces the robot to track a desired trajectory in the presence of system uncertainties and unmeasurable contact forces was utilized. The controller provides asymptotic tracking which enables the whole arm grasping objective to be completed. Experimental results for a planar, three link configuration of the Barrett WAM are provided to demonstrate the controller performance. Future work will include applying the whole arm grasping technique to a hyper-redundant tentacle manipulator [13] under development at Clemson University.

## REFERENCES

- [1] F. Asano, Z. Luo, M. Yamakita and S. Hosoe, "Dynamic Modeling and Control for Whole Body Manipulation," *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Las Vegas, NV, 2003, pp. 3162-3167.
- [2] F. Asano, Z. Luo, M. Yamakita, K. Tahara and S. Hosoe, "Bio-Mimetic Control for Whole Arm Cooperative Manipulation," *Proc. IEEE Int. Conf. on Systems Man and Cybernetics*, The Hague, The Netherlands, 2004, pp. 704-709.
- [3] A. Bicchi, "Hands for Dexterous Manipulation and Robust Grasping: A Difficult Road Toward Simplicity," *IEEE Trans. on Robotics and Automation*, vol. 16, no. 6, pp. 652-662, (2000).
- [4] A. Bicchi and V. Kumar, "Robotic Grasping and Contact: A Review," *Proc. IEEE Int. Conf. on Robotics and Automation*, San Francisco, CA, 2000, pp. 348-353.
- [5] D. Braganza, M. McIntyre, D. Dawson and I. Walker, "Whole Arm Grasping Control for Redundant Robot Manipulators", Clemson University CRB Technical Report, CU/CRB/10/12/05#1, <http://www.ces.clemson.edu/ece/crb/publicn/tr.htm>, October, 2005.
- [6] I. Cervantes, R. Kelly, J. Alvarez-Ramirez, and J. Moreno, "A Robust Velocity Field Control," *IEEE Trans. on Control Systems Technology*, vol. 10, no. 6, pp. 888-894, (2002).
- [7] G. Chirikjian and J. W. Burdick, "Design and Experiments with a 30 DOF Robot," *Proc. IEEE Int. Conf. on Robotics and Automation*, Atlanta, GA, 1993, pp. 113-119.
- [8] F. L. Lewis, D. M. Dawson and C. T. Abdallah, *Robot Manipulator Control: Theory and Practice*, Marcel Dekker Publishing Co., New York, NY, 2003.
- [9] J. Li and P. Li, "Passive Velocity Field Control (PVFC) Approach to Robot Force Control and Contour Following," *Proc. of the Japan/USA Symposium on Flexible Automation*, Ann Arbor, Michigan, 2000.

- [10] P. Li and R. Horowitz, "Passive Velocity Field Control of Mechanical Manipulators," *IEEE Trans. on Robotics and Automation*, vol. 15, no. 4, pp. 751-763, (1999).
- [11] M. Loffler, N. Costescu and D. M. Dawson, "QMotor 3.0 and the QMotor Robotic Toolkit - An Advanced PC-Based Real-Time Control Platform," *IEEE Control Systems Mag.*, vol. 22, no. 3, pp. 12-26, (2002).
- [12] M. McIntyre, W. Dixon, D. Dawson and B. Xian, "Adaptive Tracking Control of On-Line Path Planners: Velocity Fields and Navigation Functions," *Proc. of the American Controls Conference*, Portland, Oregon, June 2005, pp. 3168-3173.
- [13] W. McMahan, B. Jones, I. Walker, V. Chitrakaran, A. Seshadri, and D. Dawson, "Robotic Manipulators Inspired by Cephalopod Limbs," *Proc. CDEN Design Conference*, Montreal, Canada, 2004, pp. 1-10.
- [14] C. Melchiorri and G. Vassura, "Implementation of whole-hand manipulation capability in the UB hand system design," *Robotics Society of Japan - Advanced Robotics*, vol. 9, no. 5, pp. 547-560, (1995).
- [15] K. Mirza and D. E. Orin, "Force Distribution for power grasp in the digits system," in CSIM-IFTToMM Symp. Theory and Practice of Robots and Manipulators, 1990.
- [16] H. Mochiyama, "Whole-Arm Impedance of a Serial-Chain Manipulator," *Proc. IEEE Int. Conf. on Robotics and Automation*, Seoul, Korea, 2001, pp. 2223-2228.
- [17] Y. Nakamura, "Advanced Robotics Redundancy and Optimization", Addison-Wesley: Reading, MA, 1991.
- [18] R. Platt, A. H. Fagg and R. A. Grupen, "Nullspace Composition of Control Laws for Grasping," *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Lausanne, Switzerland, 2002, pp. 1717-1723.
- [19] R. Platt, A. H. Fagg and R. A. Grupen, "Extending Fingertip Grasping to Whole Body Grasping," *Proc. IEEE Int. Conf. on Robotics and Automation*, Taipei, Taiwan, 2003, pp. 2677-2682.
- [20] D. Reynaerts, H. Van Brussel, "Whole-finger Manipulation with a Two-fingered Robot Hand," *Robotics Society of Japan - Advanced Robotics*, vol. 9, no. 5, pp. 505-518, (1995).
- [21] K. Salisbury, "Whole Arm Manipulation," *Proc. 4th Int. Symposium Robotics Research*, 1987, pp. 183-189.
- [22] M. J. Sheridan, S. C. Ahalt and D. E. Orin, "Fuzzy Control for Robotic Power Grasp," *Robotics Society of Japan - Advanced Robotics*, vol. 9, no. 5, pp. 535-546, (1995).
- [23] P. Song, M. Yashima and V. Kumar, "Dynamics and Control of Whole Arm Grasps," *Proc. IEEE Int. Conf. on Robotics and Automation*, Seoul, Korea, 2001, pp. 2229-2234.
- [24] E. Tatlicioglu, M. McIntyre, D. Dawson and I. Walker, "Adaptive Nonlinear Tracking Control of Kinematically Redundant Robot Manipulators with Sub-Task Extensions," *Proc. IEEE Conf. on Decision and Control*, Seville, Spain, December 2005, pp. 4373 - 4378.
- [25] J. C. Trinkle, J. M. Abel and R. P. Paul, "An Investigation of Frictionless Enveloping Grasping in the plane," *Int. J. Robot. Res.*, vol. 7, no. 3, pp. 33-51, (1988).
- [26] B. Xian, D. M. Dawson, M. S. de. Queiroz and J. Chen, "A Continuous Asymptotic Tracking Control Strategy for Uncertain Nonlinear Systems," *IEEE Trans. on Automatic Control*, vol. 49, no. 7, pp. 1206 - 1211, (2004).

#### APPENDIX I EXPERIMENTAL RESULTS

The proposed controller was implemented on three links of the Barrett whole arm manipulator (WAM). The WAM is a seven degree of freedom (d.o.f.), highly dexterous and back-drivable robotic manipulator. To simplify the controller implementation, four joints of the robot were locked at fixed

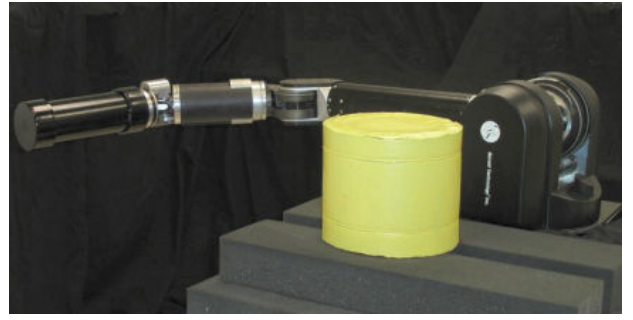


Fig. 1. Experiment setup showing the Barrett Whole Arm Manipulator and object to be grasped.

angles and the remaining links of the manipulator were used as a three d.o.f. planar robot manipulator. Figure 1, shows the experimental setup for a circular object to be grasped.

The control algorithm was written in "C++" and hosted on an AMD Athlon 1.2 GHz PC running the QNX 6.2.1 real-time operating system. Data logging and on line gain tuning was performed using Qmotor 3.0 control software [11]. Data acquisition and control implementation was performed at a frequency of 1.0 [kHz] using the ServoToGo I/O board. Joint positions were measured using the optical encoders located at the motor shaft of each axis. Joint velocity measurements were obtained using a filtered backwards difference algorithm.

Refer to Figure 2 for explanation of the notions used in this section.  $X_c = [x_c, y_c]^T \in \mathbb{R}^2$  represents the co-ordinates of the center of the object and  $r_0 \in \mathbb{R}$  represents the object radius. We define the task space variable for each of the three links and the mid-point<sup>2</sup> of each of the three links as  $X_i = [x_i, y_i]^T \in \mathbb{R}^2 \forall i = 1, 2, \dots, 6$ . The joint angles for the three links are represented by  $q = [q_1, q_2, q_3]^T \in \mathbb{R}^3$ . The object specific functions defined for each of the three links and the mid-points of the three links are defined as  $\beta_1(\cdot), \beta_2(\cdot), \dots, \beta_6(\cdot) \in \mathbb{R}$ .

The object specific functions for this planar application were defined as follows

$$\beta_i(X_i) \triangleq (x_i - x_c)^2 + (y_i - y_c)^2 - r_0^2 \forall i = 1, \dots, 6. \quad (34)$$

The following task-space velocity field for a planar, circular contour was utilized [6]

$$\dot{X}_d = \vartheta(X_6) = -2K(X_6)f(X_6) \begin{bmatrix} (x_6 - x_c) \\ (y_6 - y_c) \end{bmatrix} + 2c(X_6) \begin{bmatrix} -(y_6 - y_c) \\ (x_6 - x_c) \end{bmatrix} \quad (35)$$

where the functions  $f(X_6)$ ,  $K(X_6)$ , and  $c(X_6) \in \mathbb{R}$  are

<sup>2</sup>The object function for the mid-points of each of the manipulator links was used for this three link application, since it provides more control over the body self-motion positioning control (i.e. the repulsion functions) than just using the joint positions alone.

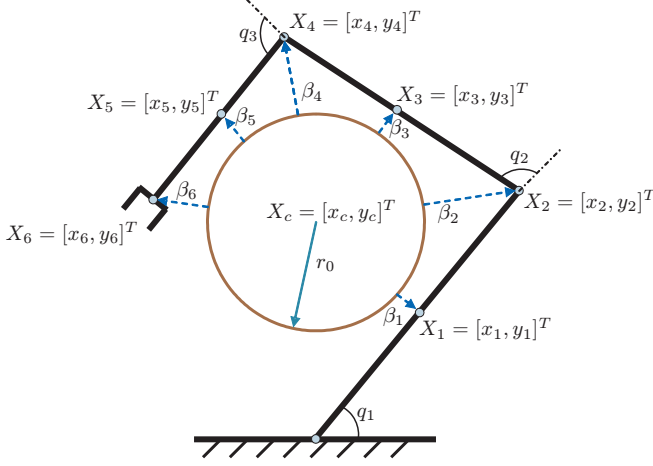


Fig. 2. Planar configuration for the three link robot with a circular object.

defined according to [6] as follows

$$\begin{aligned}
 f(X_6) &= (x_6 - x_c)^2 + (y_6 - y_c)^2 - r_0^2 \\
 K(X_6) &= \frac{k_0^*}{\sqrt{f^2(X_6)} \left\| \frac{\partial f(X_6)}{\partial X_6} \right\| + \epsilon_0} \\
 c(X_6) &= \frac{c_0 \exp(-\mu_0 \sqrt{f^2(X_6)})}{\left\| \frac{\partial f(X_6)}{\partial X_6} \right\|}.
 \end{aligned} \quad (36)$$

In (36), the constant parameters were selected as  $\epsilon_0 = 0.005$  [m<sup>3</sup>],  $\mu_0 = 20$  [m<sup>-1</sup>],  $k_0^* = 0.1$  [ms<sup>-1</sup>], and  $c_0 = 0.1$  [ms<sup>-1</sup>]. The desired position for the end-effector is  $X_d = [x_d, y_d]^T \in \mathbb{R}^2$ . The controller defined in (10) was implemented with  $e = X_d - X_6$ ,  $V(X_d) \triangleq 4 \|X_d\|^2$ , and  $\rho(\cdot) = 1$ .

The initial joint angles were  $q_1(0) = 98$ [deg],  $q_2(0) = 45.8$ [deg],  $q_3(0) = 31$ [deg], which corresponds to a position of  $x_6(0) = 0.368$  [m]  $y_6(0) = -0.883$  [m] for the end-effector in the task space. The position of the object center in the task space was found to be  $x_c = 0.307$  [m],  $y_c = -0.117$  [m], the radius of the circular object was found to be 0.12 [m]. To take into account the width of the manipulator arm, the radius of the object was set to  $r_0 = 0.16$  [m] in the implementation of the repulsion function. The control gains were selected as follows

$$\begin{aligned}
 K_e &= \text{diag}\{800, 800\}, k_n = 1, k_m = 100, \\
 k_h &= \{0.001, 0.01, 0.01, 0.05, 8.5, 8\}, \\
 \alpha_i &= \{3, 3, 3, 3, 5, 5\}, K_s = \text{diag}\{16, 9, 6\}, \\
 \beta &= \text{diag}\{5, 5, 2\}, \gamma_1 = \text{diag}\{1, 1, 1\}, \\
 \gamma_2 &= \text{diag}\{2, 2, 2\}, \epsilon = 0.01, \kappa = 500
 \end{aligned}$$

Since the desired trajectory for the end-effector is a velocity field, it will continuously generate the trajectory. To stop the desired trajectory generation when all the links of the manipulator make contact with the object, the norm of the following vector  $\beta(\cdot) = [\beta_1(\cdot), \beta_2(\cdot), \dots, \beta_6(\cdot)] \in \mathbb{R}^6$  was used. As the links of the manipulator move closer to the object boundary,  $\|\beta(\cdot)\|$  approaches zero, and this gives

an estimate of how close the manipulators links are to the object. For the experiment, we stop the trajectory generation by setting  $\dot{X}_d = 0$  when  $\|\beta(\cdot)\| \leq \eta_0$ , where the constant  $\eta_0 = 0.01$  was determined experimentally.

*Remark 6:* The value of  $\|\beta(\cdot)\|$  at which we stop the generation of the desired trajectory is specific to a particular grasp configuration. It will change if the object is re-positioned in the task space. However, if we use a highly redundant robot arm which can wrap its entire body around the object, then  $\|\beta(\cdot)\|$  will approach zero when the arm grasps the object, since the entire body of the arm will be in contact with the object.

Figure 3 shows the actual and desired joint angles. Figure 4 and Figure 5 show the joint space tracking error and the joint control torques respectively. Figure 6 show the spatial position of each of the links and their mid-points as defined in Figure 2.

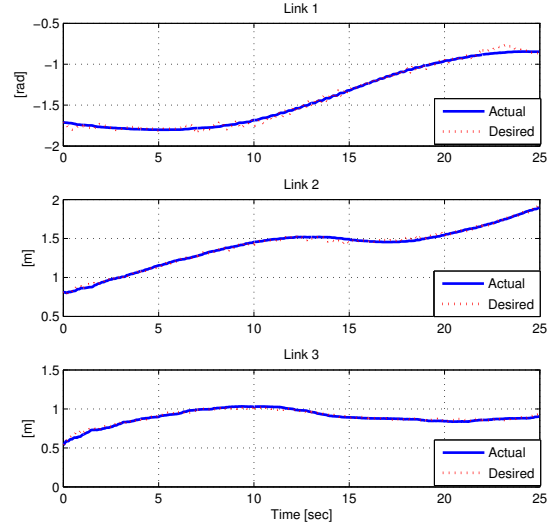


Fig. 3. Desired joint angles  $q_d(t)$  and actual joint angles  $q(t)$ .

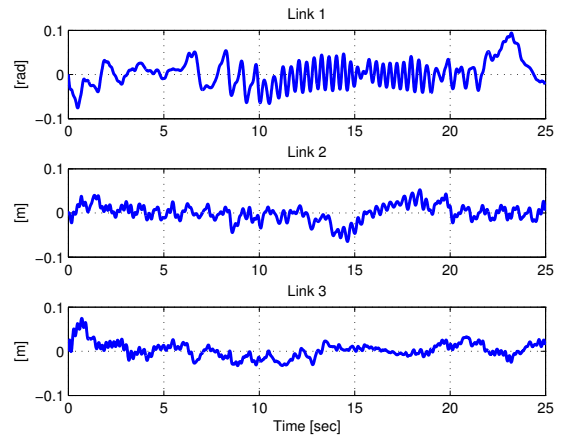


Fig. 4. Joint space position tracking error  $e_1(t)$ .

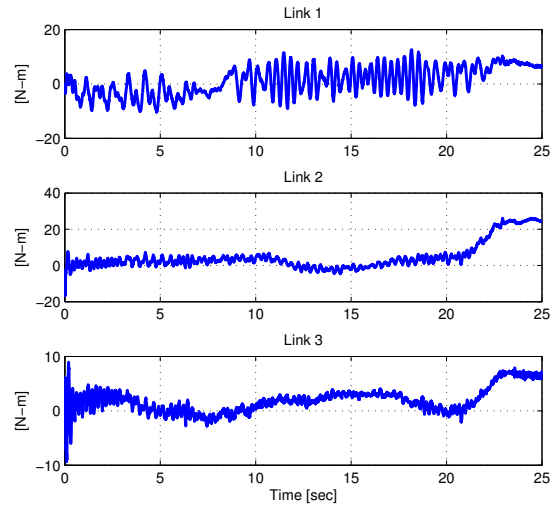


Fig. 5. Joint space control torques  $\tau(t)$ .

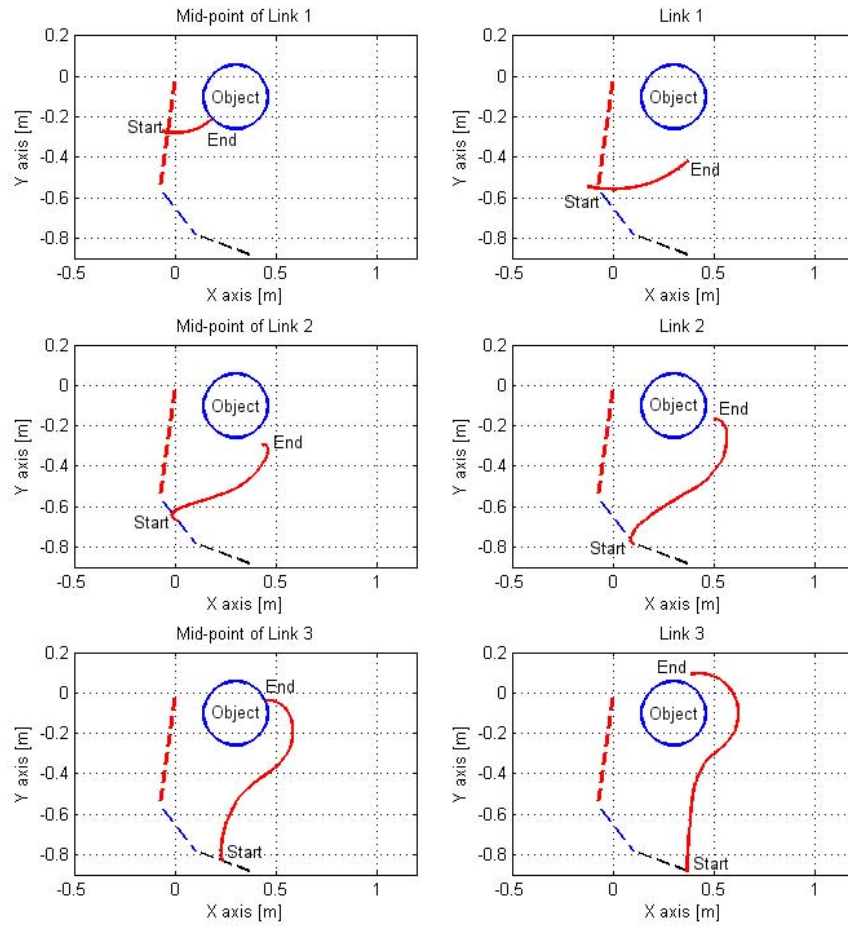


Fig. 6. Spatial position  $X_i \forall i = 1, \dots, 6$  (each link and mid-point of the link).