4-9-2015

Design and MIMO control of A Hyper-Redundant Robotic Arm

Xingsheng Xu
University of Dayton, stander@udayton.edu

Follow this and additional works at: http://ecommons.udayton.edu/stander_posters

Part of the Arts and Humanities Commons, Business Commons, Education Commons, Engineering Commons, Life Sciences Commons, Medicine and Health Sciences Commons, Physical Sciences and Mathematics Commons, and the Social and Behavioral Sciences Commons

Recommended Citation

This Book is brought to you for free and open access by the Stander Symposium at eCommons. It has been accepted for inclusion in Stander Symposium Posters by an authorized administrator of eCommons. For more information, please contact frice1@udayton.edu, mschlangen1@udayton.edu.
Design and MIMO Control of A Hyper-Redundant Robotic Arm

Xingsheng Xu
Advisor: Raúl Ordóñez
Department of Electrical and Computer Engineering, University of Dayton

Key Words
- Hyper-redundant robots (HRR) are a kind of robots that have a very large degree of kinematic redundancy.
- Forward kinematics is an arithmetic operation, which determines the position and orientation of the end-effector given the values of each joint parameter of the robot using the kinematic equations.
- Multi-input multi-output (MIMO) control approach inputs torque of each joint to control joint dynamic variables such as position, orientation, velocity and acceleration in a hyper-redundant robotic system.

Motivation
- An application robotic platform has been constructed based on both the kinematic and dynamic model of a hyper-redundant manipulator.
- Also, we apply the MIMO input-output feedback linearization control approach to this hyper-redundant robotic system to control the end-effector in Cartesian space. This control approach can highly improve the robotic performance while executing motion control or tracking a path in a constrained and complex environment.

Kinematic Model
- This model is established at its home position. The arm is composed of a set of revolute joints which are arranged orthogonal to each other while making the shoulder and wrist joint spherical.

Dynamic Model
- Manipulator Jacobian Matrices: An expressions to connect the relationship between angular velocity \( \omega_i \) linear velocity \( \dot{v}_i \) of the end-effector and joint velocity \( \dot{q}_i \) as
  \[
  \dot{v}_i = J_i \dot{q}_i, \quad \dot{r}_i = J_i \dot{q}_i, 
  \]
  where \( J_i \) and \( J_i \) are \( 1 \times n \) matrices.
- Euler-Lagrange Equation: An application leads to a robotic system of \( n \) coupled, second order nonlinear ordinary differential equations of the form
  \[
  \begin{align*}
  \frac{\partial L}{\partial q} & = \tau_i, \\
  \frac{\partial L}{\partial \dot{q}} & = \dot{\tau}_i, \\
  \end{align*}
  \]
  where \( \tau_i \) is input torque of each motor and the the Lagrangian \( L \) is given by
  \[
  L = K - P,
  \]
  where \( K \) is the kinetic energy and \( P \) is the potential energy.

Algorithm of MIMO Control
- Joint Space MIMO Control: We can rewrite the system dynamics from the Euler-Lagrange equation into Input-Output form as
  \[
  \dot{q}_i = J_i(q_i, \dot{q}_i) + g(q_i, \dot{q}_i) \cdot \tau_i,
  \]
  and outputs
  \[
  y_i = q_i, \quad i = 1, \ldots, n.
  \]
  Then, we pick our MIMO controllers as
  \[
  \dot{r}_i = g(q_i, \dot{q}_i) + v_i, \quad i = 1, \ldots, n.
  \]
  The output tracking errors will be
  \[
  e_i = r_i - y_i, \quad i = 1, \ldots, n,
  \]
  where \( e_i \) is joint variable reference. We set \( r_1 = -k_e e_1 - k_v v_1 \) and get outputs \( y_1 = -k_e e_1 - k_v v_1 \).
- Cartesian Space MIMO Control: The Jacobian matrix can be a smooth and invertible mapping between the joint variables \( q \in Q \) and the workspace variables \( \Gamma \in \mathbb{R}^n \) as
  \[
  \Gamma = J q.
  \]
  We take the derivative of both sides of the equation, then substitute \( \dot{q} \) created by the joint space to get
  \[
  \Gamma = F(q, \dot{q}) + G(q, \dot{q}) \cdot \dot{r},
  \]
  where \( F(q, \dot{q}) = J (J(q, \dot{q}) + J(q, \dot{q})) \) and \( G(q, \dot{q}) = J (J(q, \dot{q}) + J(q, \dot{q})) \). After that, we can derive our MIMO workspace controllers as
  \[
  \dot{r}_i = G(q_i, \dot{q}_i) + v_i, \quad i = 1, \ldots, n.
  \]
  The output tracking errors will be
  \[
  e_i = \Gamma_i - r_i,
  \]
  where \( e_i \) is end-effector variable reference. We set \( r_i = -k_E e_i - k_v v_i \) and get outputs \( y_i = -k_E e_i - k_v v_i \).

Performance without Dynamic Control
- Tracking A Rectangle on X-Y Plane:
- Tracking An Inclined Circle:
- Tracking-trajectory of a rectangle on X-Y plane.
- Tracking-trajectory of an inclined circle.

Performance Applying MIMO Control
- Tracking A Circle Reference Path on X-Y Plane:
- Tracking-trajectory of a rectangle on X-Y plane.
- Tracking-trajectory of an inclined circle.

Extra Constrains of Redundant System
- Mathematical Requirement:
The \( 6 \times n \) matrix \( G(q, \dot{q}) \) should be a square matrix in order to derive the Cartesian space MIMO controllers. Therefore, we need to find \( n - 6 \) more constrain equations for a redundant system (ie. \( n > 6 \)).
- Additional Constrains:
The constrain equations can be chosen differently according to the objective of the redundant robotic arm. One can pick geometric limitations of specific joints to avoid obstacles or motion optimizations of the whole arm to save energy, etc.