Interaction Control

• Manipulation requires interaction
  – object behavior affects control of force and motion
• Independent control of force and motion is not possible
  – object behavior relates force and motion
    • contact a rigid surface: kinematic constraint
    • move an object: dynamic constraint
• Accurate control of force or motion requires detailed models of
  • manipulator dynamics
  • object dynamics
  – object dynamics are usually known poorly, often not at all
Object Behavior

• Can object forces be treated as external (exogenous) disturbances?
  – the usual assumptions don’t apply:
    • “disturbance” forces depend on manipulator state
    • forces often aren’t small by any reasonable measure
• Can forces due to object behavior be treated as modeling uncertainties?
  – yes (to some extent) but the usual assumptions don’t apply:
    • command and disturbance frequencies overlap
• Example: two people shaking hands
  – how each person moves influences the forces evoked
    • “disturbance” forces are state-dependent
  – each may exert comparable forces and move at comparable speeds
    • command & “disturbance” have comparable magnitude & frequency
Alternative: control port behavior

- Port behavior:
  - system properties and/or behaviors “seen” at an interaction port

- Interaction port:
  - characterized by conjugate variables that define power flow

- Key point:
  port behavior is unaffected by contact and interaction
Impedance & Admittance

- Impedance and admittance characterize interaction
  - a dynamic generalization of resistance and conductance
- Usually introduced for linear systems but generalizes to nonlinear systems
  - state-determined representation:
  - this form may be derived from or depicted as a network model

\[
\begin{align*}
Z(s) &= \frac{e(s)}{i(s)} = \frac{1}{Cs} \\
Z(s) &= \frac{e(s)}{i(s)} = L(s)
\end{align*}
\]

\[
\begin{align*}
\dot{z} &= Z_s(z,V) & \text{State equations} \\
F &= Z_o(z,V) & \text{Output equations} \\
P &= F^tV & \text{Constraint on input & output}
\end{align*}
\]

\[
z \in \mathbb{R}^n, F \in \mathbb{R}^m, V \in \mathbb{R}^m, P \in \mathbb{R}
\]

- nonlinear 1D elastic element (spring)
  \[
  \dot{x} = \nu \\
f = \Phi(x)
\]
Impedance & Admittance (continued)

- Admittance is the causal dual of impedance
  - Admittance: flow out, effort in
  - Impedance: effort out, flow in
- Linear system: admittance is the inverse of impedance
- Nonlinear system:
  - causal dual is well-defined:
  - but may not correspond to any impedance
    - inverse may not exist

\[
Y(s) = Z(s)^{-1}
\]
electrical capacitor

\[
Y(s) = \frac{i(s)}{e(s)} = Cs
\]

\[
\begin{cases}
\dot{y} = Y_s(y, F) \\
V = Y_o(y, F) \\
P = F^tV
\end{cases}
\]
\(y \in \mathbb{R}^n, F \in \mathbb{R}^m, V \in \mathbb{R}^m, P \in \mathbb{R}\)

nonlinear 1D inertial element (mass)

\[
\dot{p} = f \\
v = \Psi(p)
\]
Impedance as dynamic stiffness

- Impedance is also loosely defined as a dynamic generalization of stiffness
  - effort out, displacement in
- Most useful for mechanical systems
  - displacement (or generalized position) plays a key role

\[
\begin{align*}
\dot{z} &= Z_s(z,X) \\
F &= Z_o(z,X) \\
dW &= F^t dX \\
z \in \mathbb{R}^n, F \in \mathbb{R}^m, X \in \mathbb{R}^m, P \in \mathbb{R}
\end{align*}
\]
Interaction control: causal considerations

• What’s the best input/output form for the manipulator?
• The set of objects likely to be manipulated includes
  – inertias
    • minimal model of most movable objects
  – kinematic constraints
    • simplest description of surface contact
• Causal considerations:
  – inertias prefer admittance causality
  – constraints require admittance causality
  – compatible manipulator behavior should be an impedance
• An ideal controller should make the manipulator behave as an impedance
• Hence impedance control
  – Hogan 1979, 1980, 1985, etc.
Robot Impedance Control

- Works well for interaction tasks:
  - Automotive assembly
    - (Case Western Reserve University, US)
  - Food packaging
    - (Technical University Delft, NL)
  - Hazardous material handling
    - (Oak Ridge National Labs, US)
  - Automated excavation
    - (University of Sydney, Australia)
  - ... and many more

- Facilitates multi-robot / multi-limb coordination
  - Schneider et al., Stanford

- Enables physical cooperation of robots and humans
  - Kosuge et al., Japan
  - Hogan et al., MIT
OSCAR the robot

Photograph removed due to copyright restrictions.

E.D.Fasse & J.F.Broenink, U. Twente, NL
Network modeling perspective on interaction control

• Port concept
  – control interaction port behavior
  – port behavior is unaffected by contact and interaction

• Causal analysis
  – impedance and admittance characterize interaction
  – object is likely an admittance
  – control manipulator impedance

• Model structure
  – structure is important
  – power sources are commonly modeled as equivalent networks
    • Thévenin equivalent
    • Norton equivalent

• Can equivalent network structure be applied to interaction control?
Equivalent networks

• Initially applied to networks of static linear elements
  • Sources & linear resistors
    – Thévenin equivalent network
  • Thévenin equivalent source—power supply or transfer
  • Thévenin equivalent impedance—interaction
  • Connection—series / common current / 1-junction
    – Norton equivalent network is the causal dual form

• Subsequently applied to networks of dynamic linear elements
  • Sources & (linear) resistors, capacitors, inductors
Nonlinear equivalent networks

• Can equivalent networks be defined for nonlinear systems?
  – Nonlinear impedance and admittance can be defined as above
  – Thévenin & Norton sources can also be defined

• However…
  – In general the junction structure cannot

• In other words:
  – separating the pieces is always possible
  – re-assembling them by superposition is not
Nonlinear equivalent network for interaction control

• One way to preserve the junction structure:
  – specify an equivalent network structure in the (desired) interaction behavior
  – provides key superposition properties

• Specifically:
  – *nodic* desired impedance
    • does not require inertial reference frame
  – “virtual” trajectory
    • “virtual” as it need not be a realizable trajectory

\[
\begin{align*}
V_0 &= V_0 : \{c\} & \text{virtual trajectory} \\
\Delta V &= V_0 - V & \text{network junction structure (0 junction)} \\
\dot{z} &= Z_s(z, \Delta V) : \{c\} & \text{nodic impedance} \\
F &= Z_o(z, \Delta V) : \{c\} & \{c\} \text{ denotes modulation by control inputs}
\end{align*}
\]

Norton equivalent network
Virtual trajectory

- **Nodic impedance:**
  - Defines desired interaction dynamics
  - Nodic because input velocity is defined relative to a “virtual” trajectory
- **Virtual trajectory:**
  - Like a motion controller’s reference or nominal trajectory *but* no assumption that dynamics are fast compared to motion
  - “virtual” because it need not be realizable
    - e.g., need not be confined to manipulator’s workspace
Superposition of “impedance forces”

- Minimal object model is an inertia
  - it responds to the sum of input forces
  - in network terms: it comes with an associated 1-junction

\[
\Delta V_1 = V_{o1} - V \\
\dot{z}_1 = Z_{s1}(z_1, \Delta V_1) \\
F_1 = Z_{o1}(z_1, \Delta V_1)
\]

\[
\Delta V_2 = V_{o2} - V \\
\dot{z}_2 = Z_{s2}(z_2, \Delta V_2) \\
F_2 = Z_{o2}(z_2, \Delta V_2)
\]

\[
\Delta V_3 = V_{o3} - V \\
\dot{z}_3 = Z_{s3}(z_3, \Delta V_3) \\
F_3 = Z_{o3}(z_3, \Delta V_3)
\]

- This guarantees linear summation of component impedances…
- …even if the component impedances are nonlinear

\[ V = m^{-1}(F_1 + F_2 + F_3) \]
One application: collision avoidance

- Impedance control also enables non-contact (virtual) interaction
  - Impedance component to acquire target:
    - Attractive force field (potential “valley”)
  - Impedance component to prevent unwanted collision:
    - Repulsive force-fields (potential “hills”)
    - One per object (or part thereof)
  - Total impedance is the sum of these components
    - Simultaneously acquires target while preventing collisions
  - Works for moving objects and targets
    - Update their location by feedback to the (nonlinear) controller
  - Computationally simple
    - Initial implementation used 8-bit Z80 processors
    - Andrews & Hogan, 1983

High-speed collision avoidance

• Static protective (repulsive) fields must extend beyond object boundaries
  – may slow the robot unnecessarily
  – may occlude physically feasible paths
  – especially problematical if robot links are protected

• Solution: *time-varying* impedance components
  – protective (repulsive) fields grow as robot speeds up, shrink as it slows down
  – Fields shaped to yield maximum acceleration or deceleration

  • Newman & Hogan, 1987


• See also extensive work by Khatib et al., Stanford
Impedance Control Implementation

- Controlling robot impedance is an ideal
  - like most control system goals it may be difficult to attain
- How do you control impedance or admittance?
- One primitive but highly successful approach:
  - Design low-impedance hardware
    - Low-friction mechanism
      - Kinematic chain of rigid links
    - Torque-controlled actuators
      - e.g., permanent-magnet DC motors
      - high-bandwidth current-controlled amplifiers
  - Use feedback to increase output impedance
    - (Nonlinear) position and velocity feedback control
- “Simple” impedance control
Robot Model

- Effort-driven inertia

\[ I(\theta)\ddot{\theta} + C(\theta, \omega) + G(\theta) = \tau_{\text{motor}} + \tau_{\text{interaction}} \]

- Linkage kinematics transform interaction forces to interaction torques

\[ X = L(\theta) \]

\[ \dot{V} = \dot{X} = \left( \frac{\partial L}{\partial \theta} \right) \dot{\theta} = J(\theta)\omega \]

\[ \tau_{\text{interaction}} = J(\theta)^T F_{\text{interaction}} \]

\( \theta \): generalized coordinates, joint angles, configuration variables

\( \omega \): generalized velocities, joint angular velocities

\( \tau \): generalized forces, joint torques

\( I \): configuration-dependent inertia

\( C \): inertial coupling (Coriolis & centrifugal accelerations)

\( G \): potential forces (gravitational torques)
Simple Impedance Control

- Target end-point behavior
  - Norton equivalent network with elastic and viscous impedance, possibly nonlinear
- Express as equivalent (joint-space) configuration-space behavior
  - use kinematic transformations
- This defines a position-and-velocity-feedback controller...
  - A (non-linear) variant of PD (proportional+derivative) control
- ...that will implement the target behavior

\[
\mathbf{F}_{impedance} = K(X_o - X) + B(V_o - V)
\]

\[ X_o: \text{virtual position} \]
\[ V_o: \text{virtual velocity} \]
\[ K: \text{displacement-dependent (elastic) force function} \]
\[ B: \text{velocity-dependent force function} \]

\[
\mathbf{\tau}_{motor} = J(\theta)^{\dagger} \mathbf{F}_{impedance}
\]
\[
\mathbf{\tau}_{motor} = \mathbf{J}(\theta)^{\dagger} (K(X_o - L(\theta)) + B(V_o - J(\theta)\omega))
\]

Dynamics of controller impedance coupled to mechanism inertia with interaction port:

\[
I(\theta)\ddot{\omega} + C(\theta, \omega) + G(\theta) =
J(\theta)^{\dagger} (K(X_o - L(\theta)) + B(V_o - J(\theta)\omega))
+ J(\theta)^{\dagger} \mathbf{F}_{interaction}
\]
Mechanism singularities

- Impedance control also facilitates interaction with the robot’s own mechanics
  - Compare with motion control:
- Position control maps desired end-point trajectory onto configuration space (joint space)
  - Requires inverse kinematic equations
    - Ill-defined, no general algebraic solution exists
      - one end-point position usually corresponds to many configurations
      - some end-point positions may not be reachable
- Resolved-rate motion control uses inverse Jacobian
  - Locally linear approach, will find a solution if one exists
  - At some configurations Jacobian becomes singular
    - Motion is not possible in one or more directions
- A typical motion controller won’t work at or near these singular configurations

\[
X = L(\theta) \\
\theta_{desired} = L^{-1}(X_{desired}) \\
V = J(\theta)\omega \\
\omega_{desired} = J(\theta)^{-1}V_{desired}
\]
Mechanism junction structure

• Mechanism kinematics relate configuration space \( \{\theta\} \) to workspace \( \{X\} \)
  - In network terms this defines a multiport modulated transformer
  - Hence power conjugate variables are well-defined in opposite directions

\[
\begin{align*}
\eta & \quad \text{p} \\
\tau & \quad \text{F} \\
\dot{\omega} & \quad \text{MTF} \quad \dot{\omega} \\
\theta & \quad \text{X}
\end{align*}
\]

• Generalized coordinates uniquely define mechanism configuration
  - By definition

• Hence the following maps are always well-defined
  - generalized coordinates (configuration space) to end-point coordinates (workspace)
  - generalized velocities to workspace velocity
  - workspace force to generalized force
  - workspace momentum to generalized momentum
Control at mechanism singularities

• Simple impedance control law was derived by transforming desired behavior…
  – Norton equivalent network in workspace coordinates
  …from workspace to configuration (joint) space
• All of the required transformations are guaranteed well-defined at all configurations
  – \( \mathbf{X} \leftarrow \theta \)
  – \( \mathbf{V} \leftarrow \omega \)
  – \( \tau \leftarrow \mathbf{F} \)
  \( \tau_{\text{motor}} = J(\theta)^T (K(\mathbf{X}_o - L(\theta)) + B(\mathbf{V}_o - J(\theta)\omega)) \)
• Hence the simple impedance controller can operate near, at and through mechanism singularities
Generalized coordinates

• Aside:
  – Identification of generalized coordinates requires care
    • Independently variable
    • Uniquely define mechanism configuration
    • Not themselves unique
  – Actuator coordinates are often suitable, but not always
    • Example: Stewart platform
  – Identification of generalized forces also requires care
    • Power conjugates to generalized velocities
    • \( P = \tau'\omega \) or \( dW = \tau'd\theta \)
  – Actuator forces are often suitable, but not always
Inverse kinematics

• Generally a tough computational problem
• Modeling & simulation afford simple, effective solutions
  – Assume a simple impedance controller
  – Apply it to a simulated mechanism with simplified dynamics
  – Guaranteed convergence properties
    – Hogan 1984
    – Slotine & Yoerger 1987

• Same approach works for redundant mechanisms
  – Redundant: more generalized coordinates than workspace coordinates
  – Inverse kinematics is fundamentally “ill-posed”
  – Rate control based on Moore-Penrose pseudo-inverse suffers “drift”
  – Proper analysis of effective stiffness eliminates drift
    – Mussa-Ivaldi & Hogan 1991

Intrinsically variable impedance

• Feedback control of impedance suffers inevitable imperfections
  – “parasitic” sensor & actuator dynamics
  – communication & computation delays
• Alternative: control impedance using intrinsic properties of the actuators and/or mechanism
  – Stiffness
  – Damping
  – Inertia
Intrinsically variable stiffness

- Engineering approaches
  - Moving-core solenoid
  - Separately-excited DC machine
    - Fasse et al. 1994
  - Variable-pressure air cylinder
  - Pneumatic tension actuator
    - McKibben “muscle”
  - …and many more
- Mammalian muscle
  - antagonist co-contraction increases stiffness & damping
  - complex underlying physics
    - see 2.183
  - increased stiffness requires increased force

Opposing actuators at a joint

- Assume
  - constant moment arms
  - linear force-length relation
  - (grossly) simplified model of antagonist muscles about a joint

\[ f_g = k_g l_g \]
\[ f_n = k_n l_n \]
\[ l_g = l_{go} - r_g q \]
\[ l_n = l_{no} + r_n q \]
\[ t = r_g f_g - r_n f_n = r_g k_g \left(l_{go} - r_g q\right) - r_n k_n \left(l_{no} + r_n q\right) \]

Equivalent behavior:
- Opposing torques subtract
- Opposing impedances add
  - Joint stiffness positive if actuator stiffness positive

\[ q_o = \left(r_g k_g l_{go} - r_n k_n l_{no}\right) \]
\[ K = \left(r_g^2 k_g + r_n^2 k_n\right) \]
Configuration-dependent moment arms

- Connection of linear actuators usually makes moment arm vary with configuration

- Joint stiffness, $K$:
  - Second term always positive
  - First term may be negative

\[
K = -\left( \frac{\partial r_g}{\partial q} f_g + \frac{\partial r_n}{\partial q} f_n \right) + \left( r_g^2 k_g + r_n^2 k_n \right)
\]

More typical: change signs on the transformers

\[
t = -r_g(q)f_g(q) - r_n(q)f_n(q)
\]

\[
\frac{\partial t}{\partial q} = -\frac{\partial r_g}{\partial q} f_g - r_g \frac{\partial f_g}{\partial q} - \frac{\partial r_n}{\partial q} f_n - r_n \frac{\partial f_n}{\partial q}
\]
This is the “tent-pole” effect

- Consequences of configuration-dependent moment arms:
- Opposing “ideal” (zero-impedance) tension actuators
  - agonist moment grows with angle, antagonist moment declines
  - *always* unstable
- Constant-stiffness actuators
  - stable only for limited tension
- Mammalian muscle:
  - stiffness is proportional to tension
    - good approximation of complex behavior
    - can be stable for all tension

- Take-home messages:
- Kinematics matters
  - “Kinematic” stiffness may dominate
- Impedance matters
  - Zero output impedance may be highly undesirable
Intrinsically variable inertia

- Inertia is difficult to modulate via feedback but mechanism inertia is a strong function of configuration
- Use excess degrees of freedom to modulate inertia
  - e.g., compare contact with the fist or the fingertips
- Consider the apparent (translational) inertia at the tip of a 3-link open-chain planar mechanism
  - Use mechanism transformation properties
- Translational inertia is usually characterized by \( \mathbf{p} = \mathbf{Mv} \)
- Generalized (configuration space) inertia is \( \eta = I(\theta)\omega \)
  - Jacobian: \( \mathbf{v} = \mathbf{J}(\theta)\omega \)
    \[ \eta = \mathbf{J}(\theta)^t \mathbf{p} \]
  - Corresponding tip (workspace) inertia:
    \[ \mathbf{p} = \mathbf{J}(\theta)^{-t} \mathbf{I}(\theta)\mathbf{J}(\theta)^{-1} \mathbf{v} \]
    \[ \mathbf{M}_{\text{tip}} = \mathbf{J}(\theta)^{-t} \mathbf{I}(\theta)\mathbf{J}(\theta)^{-1} \]
- Snag: \( \mathbf{J}(\theta) \) is not square—inverse \( \mathbf{J}(\theta)^{-1} \) does not exist
Causal analysis

• Inertia is an admittance
  – prefers integral causality

• Transform inverse configuration-space inertia
  – Corresponding tip (workspace) inertia
  – This transformation is always well-defined

• Does $I(\theta)^{-1}$ always exist?
  – consider how we constructed $I(\theta)$ from individual link inertias
  – $I(\theta)$ must be symmetric positive definite, hence its inverse exists

• Does $M_{\text{tip}}^{-1}$ always exist?
  – yes, but sometimes it loses rank
    • inverse mass goes to zero in some directions—can’t move that way
  – causal argument: input force can always be applied
    • mechanism will “figure out” whether & how to move