Dynamics and Real-Time Simulation (DARTS) Laboratory

Spatial Operator Algebra (SOA)

10. Closed-Chain Dynamics (Cut-Joint Method)

Abhinandan Jain

June 19, 2024

https://dartslab.jpl.nasa.gov/

Jet Propulsion Laboratory
California Institute of Technology

SOA Foundations Track Topics (serial-chain rigid body systems)

1. **Spatial (6D) notation** – spatial velocities, forces, inertias; spatial cross-product, rigid body transformations & properties; parallel axis theorem
2. **Single rigid body dynamics** – equations of motion about arbitrary frame using spatial notation
3. **Serial-chain kinematics** – minimal coordinate formulation, hinges, velocity recursions, Jacobians; first spatial operators; O(N) scatter and gather recursions
4. **Serial-chain dynamics** – equations of motion using spatial operators; Newton–Euler mass matrix factorization; O(N) inverse dynamics; composite rigid body inertia; forward Lyapunov equation; mass matrix decomposition; mass matrix computation; alternative inverse dynamics
5. **Articulated body inertia -** Concept and definition; Riccati equation; alternative force decompositions
6. **Mass matrix factorization and inversion** – spatial operator identities; Innovations factorization of the mass matrix; Inversion of the mass matrix
7. **Recursive forward dynamics** – O(N) recursive forward dynamics algorithm; including gravity and external forces; inter-body forces identity

See https://dartslab.jpl.nasa.gov/References/index.php for publications and references on the SOA methodology.

SOA Generalization Track Topics

8. **Graph theory based structure** – BWA matrices, connection to multibody systems
9. **Tree topology systems** – generalization to tree topology rigid body systems, SKO/SPO operators, gather/scatter algorithms
10. **Closed-chain dynamics (cut-joint)** – holonomic and non-holonomic constraints, cut-joint method, operational space inertia, projected dynamics
11. **Closed-chain dynamics (constraint embedding)** – constraint embedding for graph transformation, minimal coordinate closed-chain dynamics
12. **Flexible body dynamics** – Extension to flexible bodies, modal representations, recursive flexible body dynamics

Recap

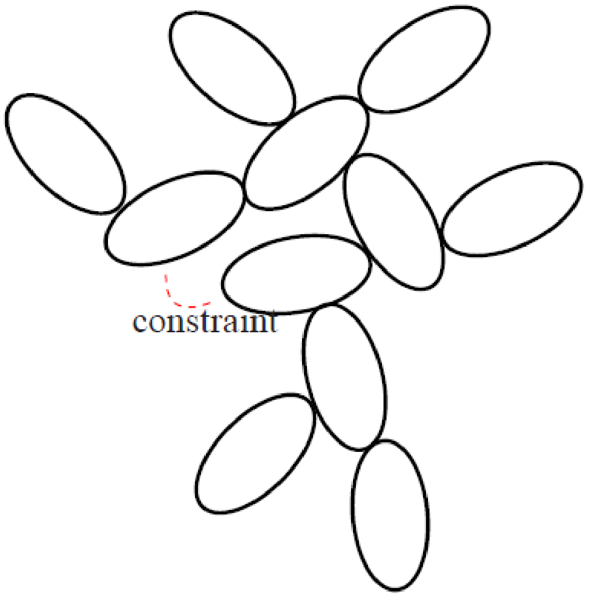
Previous Session Recap

- Built upon BWA concepts to define SKO and SPO operators for multibody systems
- Defined the general class of SKO-models for multibody systems
- Showed the virtually all the analysis and algorithms developed for serial-chain, rigi-body systems carries over to SKO models with only minor generalizations
- This opens the door for applying the operator methods and algorithms to any multibody system with an SKO model
  - As we will see, this is a very broad class of multibody systems

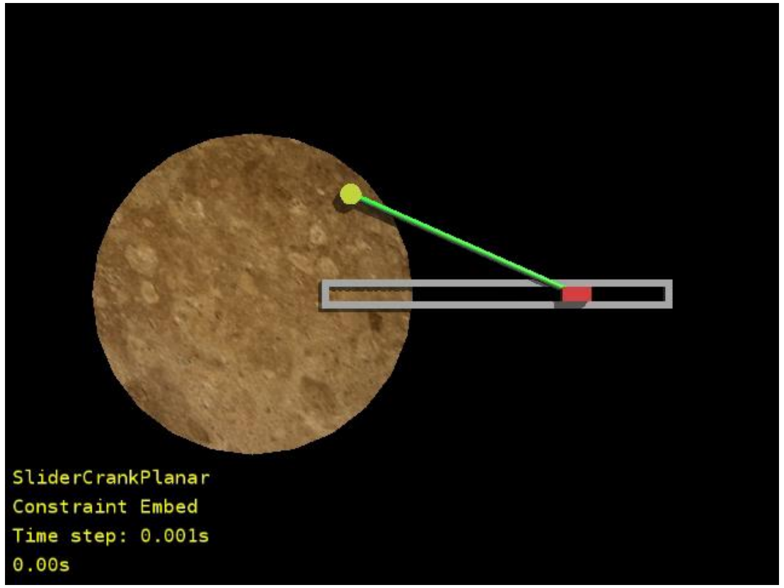
Closed-Chain Dynamics

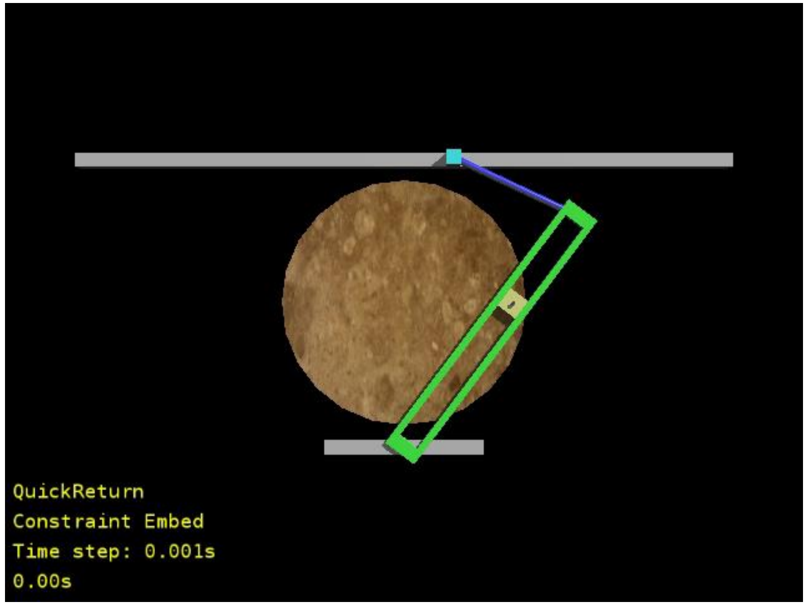
Closed-chain multibody systems

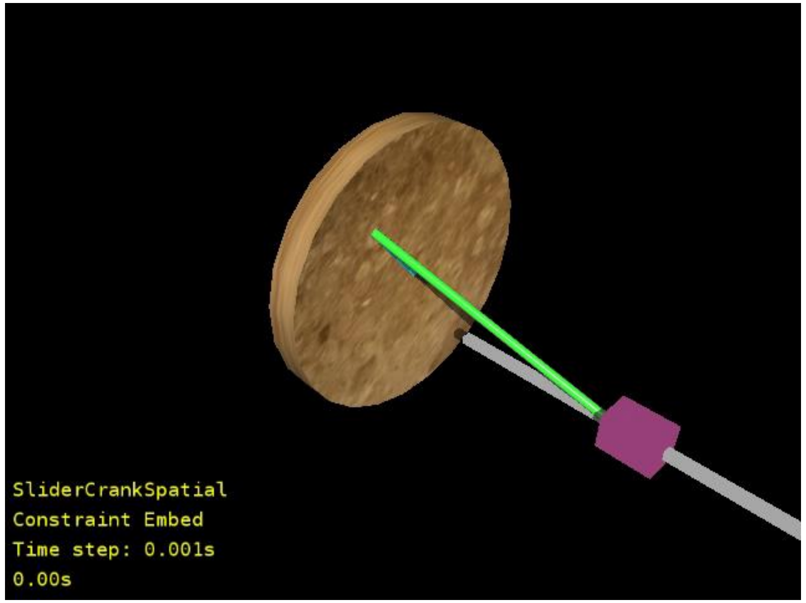
- Our development so far has focused on multibody systems that have SKO models
  - Serial-chain systems
  - Branched tree systems
- The underlying bodies topology has that been of a tree
- Closed-chain system topologies have loops, and hence do not have tree structure or an obvious SKO model

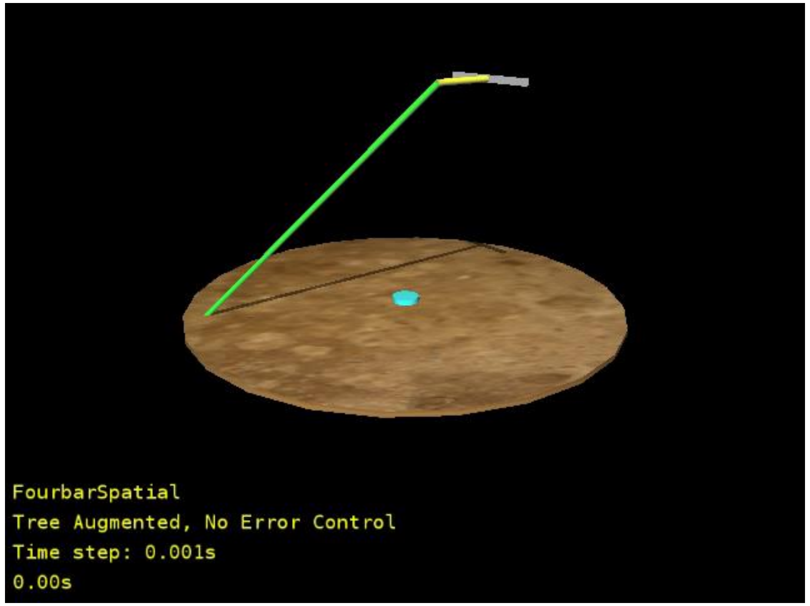


Example Closed-Chain Dynamics Mechanisms

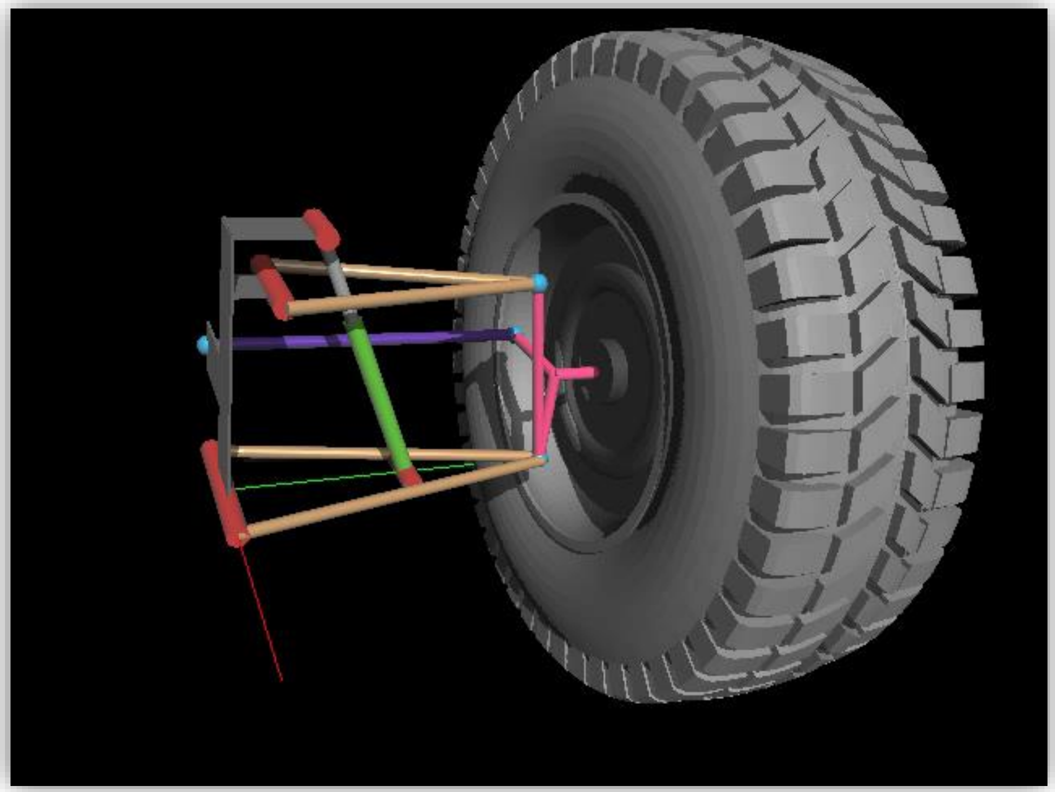






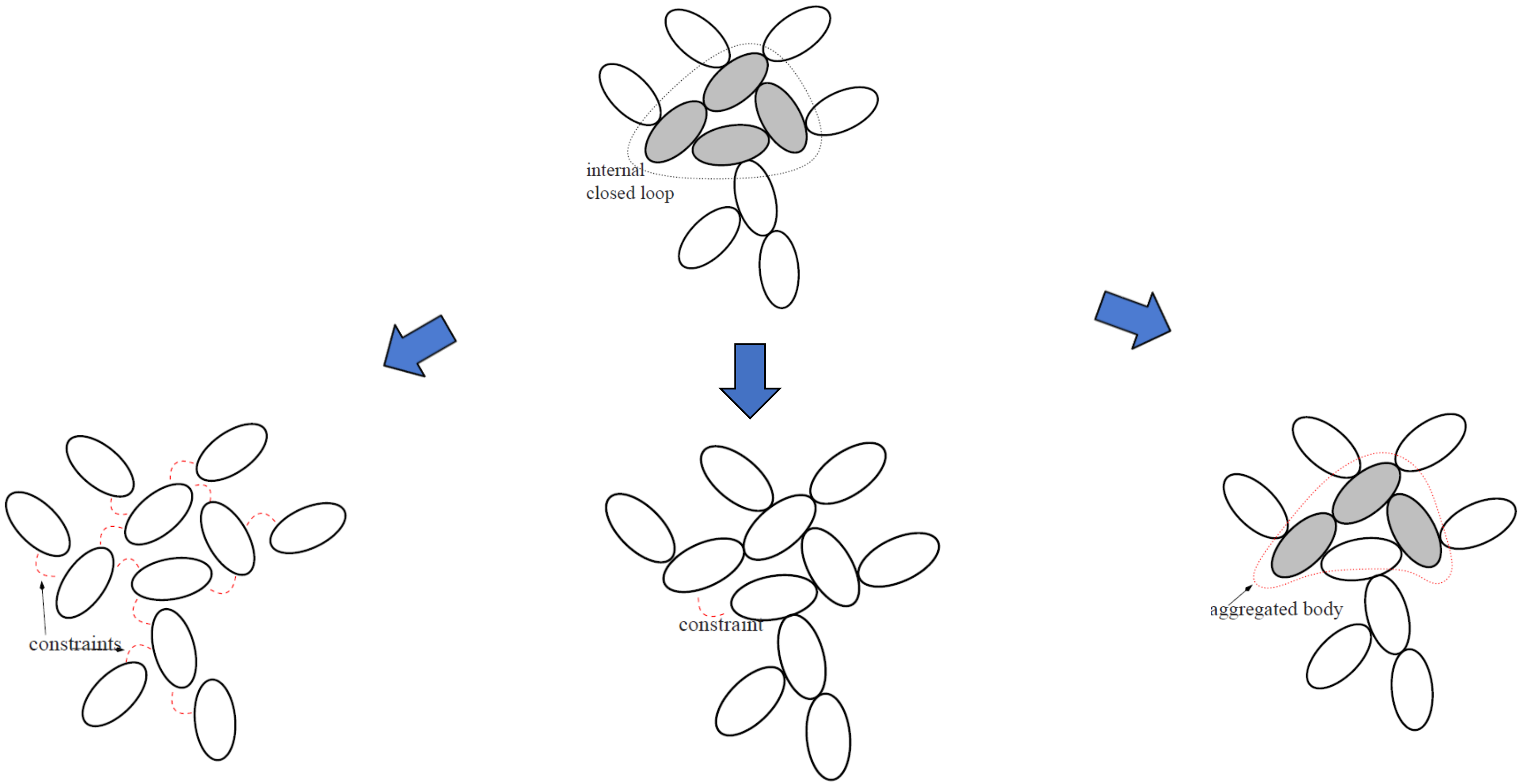


Suspension system example



Multiple closed-loops

Closed Chain Modeling Options



FA model

Non-minimal coords + constraints

Simple setup

TA model

Minimal tree coords + constraints

Better for large loops

CE model

Minimal coords

Optimal for small loops

Constraints

- **Bilateral constraints:** Defined by an equality relationship
  - eg. mechanism loops
  - always active when present
  - Hinges are actual a dual way of describing a bilateral constraint

- **Unilateral constraints:** Defined by inequality constraint
  - eg. contact/collision dynamics
  - can be active/inactive based on the current state

Holonomic Bilateral Constraints

Smooth constraint on the coordinates

$$\mathfrak{d}(\theta, t) = \mathbf{0}$$

Reduces dofs from $\mathcal{N}$ to $(\mathcal{N} - n_c)$ dimension.

Differentiating:

$$\dot{\mathfrak{d}}(\theta, t) = G_c(\theta, t)\dot{\theta} - \mathfrak{U}(t) = \mathbf{0}$$

$$G_c(\theta, t) \triangleq \nabla_\theta \mathfrak{d}(\theta, t) \in \mathcal{R}^{n_c \times \mathcal{N}}$$

$$\mathfrak{U}(t) \triangleq -\frac{\partial \mathfrak{d}(\theta, t)}{\partial t} \in \mathcal{R}^{n_c}$$

Non-holonomic Bilateral Constraints

The constraints are expressed directly at the velocity level:

$$\dot{\mathfrak{d}}(\theta, t) = G_c(\theta, t)\dot{\theta} - \mathfrak{U}(t) = \mathbf{0}$$

Closed-chain Equations of Motion

Using Lagrange multipliers, DAE form of the equations of motion is:

$$\mathcal{M}(\theta)\ddot{\theta} + \mathcal{C}(\theta, \dot{\theta}) - G_c^*(\theta, t)\lambda = \mathcal{T}$$
$$G_c(\theta, t)\dot{\theta} = \mathfrak{U}(t)$$

The dimension of the Lagrange multipliers and the row dimension of G_c increases with increase in number of cut-joints.

The Lagrange multipliers are the inter-body constraint forces for the cut-joints.

Rearranged matrix form

Using Lagrange multipliers, DAE form

$$\begin{aligned} \mathcal{M}(\theta)\ddot{\theta} + \mathcal{C}(\theta,\dot{\theta}) - G_c^*(\theta,t)\lambda &= \mathcal{T} \\ G_c(\theta,t)\dot{\theta} &= \mathfrak{U}(t) \end{aligned}$$

have

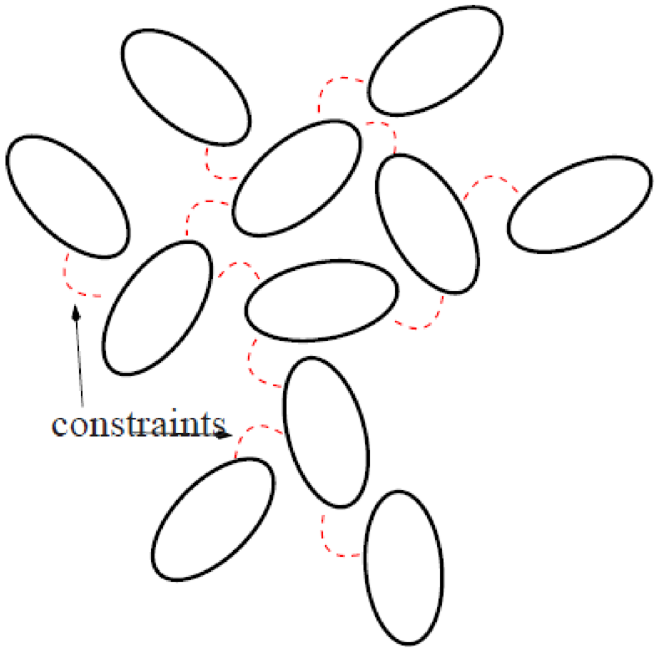
$$\begin{pmatrix} \mathcal{M} & G_c^* \\ G_c & \mathbf{0} \end{pmatrix} \begin{bmatrix} \ddot{\theta} \\ -\lambda \end{bmatrix} = \begin{bmatrix} \mathcal{T} - \mathcal{C} \\ \acute{\mathfrak{U}} \end{bmatrix}$$

$$\acute{\mathfrak{U}} \triangleq \dot{\mathfrak{U}}(t) - \dot{G}_c\dot{\theta} \in \mathcal{R}^{n_c} \qquad \ddot{\mathfrak{d}}(\theta,t) \overset{11.3,11.6}{=} G_c\ddot{\theta} - \acute{\mathfrak{U}}$$

Solution Approaches

1. Projection solution method

- Switch to minimal coordinates form
- Pick $(\mathcal{N} - n_c)$ of the coordinates as independent variables
- Numerically project the equations of motion down to these independent variables
- Solve these equations of motion and lift up the solution to get all coordinate accels
- Expensive process, and has issues with picking indep coords



Projection method details

The main idea is to numerically project the dynamics down to a minimal set of coordinates and hence eliminate the explicit constraints

$$G_c X_c = \mathbf{0} \qquad \dot{\theta} = \dot{\theta}_p + X_c \boxed{\dot{\theta}_r}$$

reduced minimal coordinates

$$\mathcal{T} - \mathcal{C} = \mathcal{M}\ddot{\theta} - G_c^* \lambda \overset{11.34}{=} \mathcal{M}(\underline{\ddot{\theta}}_p + X_c \ddot{\theta}_r) - G_c^* \lambda$$

$$\boxed{\mathcal{M}_r \triangleq X_c^* \mathcal{M} X_c \in \mathcal{R}^{\mathcal{N}-n_c \times \mathcal{N}-n_c}}$$

projected mass matrix

$$\mathcal{M}_r \ddot{\theta}_r = X_c^* (\mathcal{T} - \mathcal{C} - \mathcal{M}\underline{\ddot{\theta}}_p)$$

projected equations of motion

Projection method comments

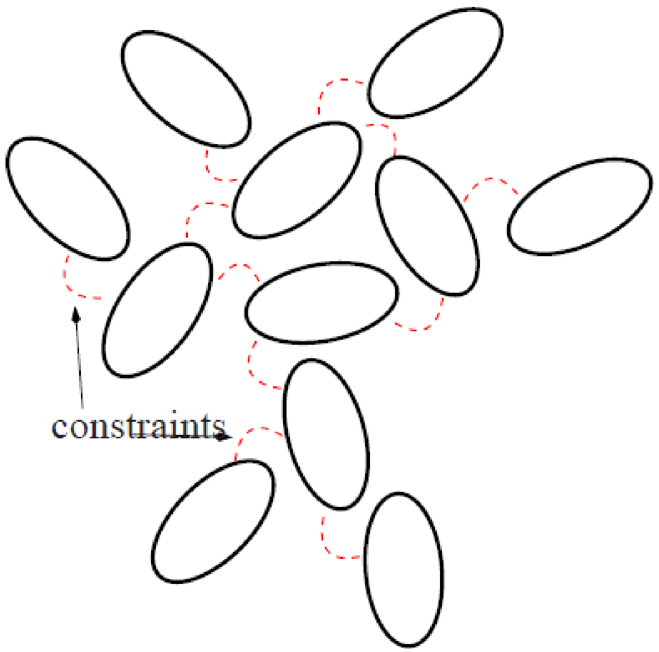
- The projected methods is a minimal coordinates, and hence ODE approach
- However, the mass matrix is obtained by a numerical projection approach – which destroys all structure, and we are left with an expensive to compute mass matrix, with opaque structure
- The lack of structure means that SKO models are not applicable and the recursive techniques cannot be used

2. Direct solution method

- Set up and solve this equation numerically

$$\begin{pmatrix} \mathcal{M} & G_c^* \\ G_c & \mathbf{0} \end{pmatrix} \begin{bmatrix} \ddot{\theta} \\ -\lambda \end{bmatrix} = \begin{bmatrix} \mathcal{T} - \mathcal{C} \\ \acute{\mathfrak{U}} \end{bmatrix}$$

- Usually used by absolute coordinate approaches which use maximal cuts so have individual bodies
  - The "tree" system mass matrix is constant & sparse and consists of independent bodies

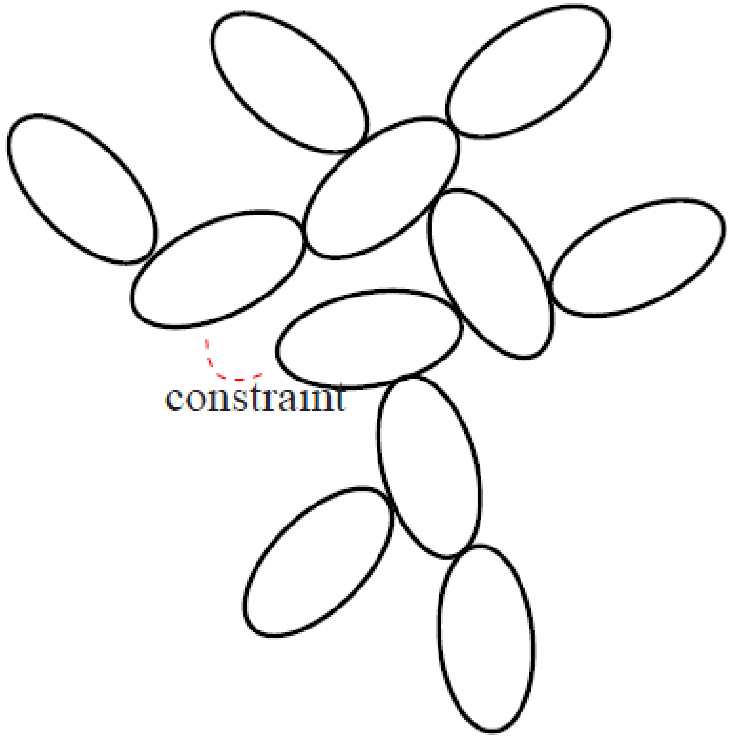


3. Augmented solution method

- Use <u>minimal</u> number of joint cuts so have a spanning tree + cut-joint constraints

$$\begin{pmatrix} \mathcal{M} & G_c^* \\ G_c & \mathbf{0} \end{pmatrix} \begin{bmatrix} \ddot{\theta} \\ -\lambda \end{bmatrix} = \begin{bmatrix} \mathcal{T} - \mathcal{C} \\ \acute{\mathfrak{U}} \end{bmatrix}$$

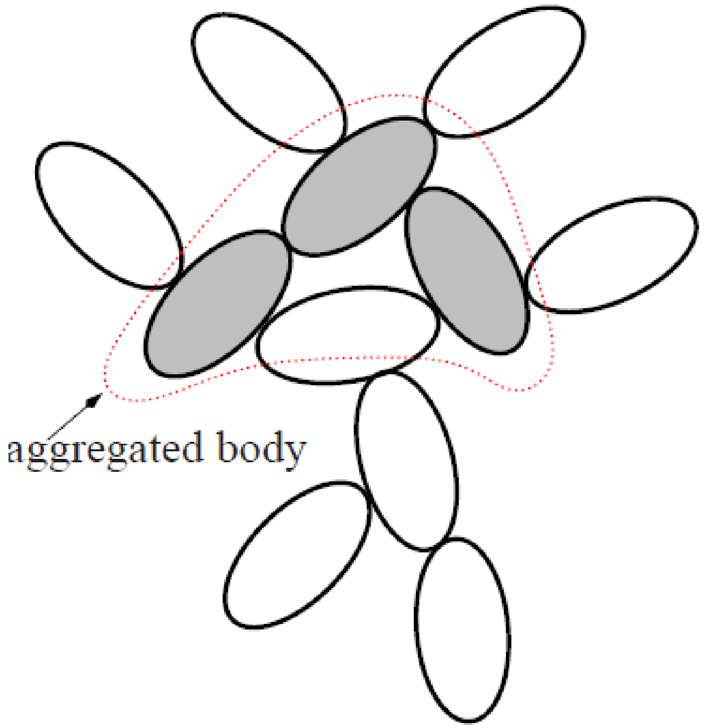
- The tree system is a minimal coordinate multibody system with a configuration dependent mass matrix



Jet Propulsion Laboratory
California Institute of Technology

4. Constraint embedding solution approach

- Structure based minimal coordinate approach
- Uses graph transformation and variable geometry bodies
- Will cover later …



Augmented Solution Method

Augmented solution approach

Have

$$\begin{pmatrix} \mathcal{M} & G_c^* \\ G_c & \mathbf{0} \end{pmatrix} \begin{bmatrix} \ddot{\theta} \\ -\lambda \end{bmatrix} = \begin{bmatrix} \mathcal{T} - \mathcal{C} \\ \acute{\mathfrak{U}} \end{bmatrix}$$

Inverse expression

$$\begin{aligned} \begin{pmatrix} A & B \\ C & D \end{pmatrix}^{-1} &= \begin{pmatrix} \mathbf{I} & -A^{-1}B \\ \mathbf{0} & \mathbf{I} \end{pmatrix} \begin{pmatrix} A^{-1} & \mathbf{0} \\ \mathbf{0} & F_2^{-1} \end{pmatrix} \begin{pmatrix} \mathbf{I} & \mathbf{0} \\ -CA^{-1} & \mathbf{I} \end{pmatrix} \\ &= \begin{pmatrix} A^{-1} + A^{-1}BF_2^{-1}CA^{-1} & -A^{-1}BF_2^{-1} \\ -F_2^{-1}CA^{-1} & F_2^{-1} \end{pmatrix} \end{aligned}$$

$$F_2 = (D - CA^{-1}B)$$

Rearranged solution equations

Spanning tree "free" gen accels

$$\ddot{\theta}_f \triangleq \mathcal{M}^{-1}(\mathcal{T} - \mathcal{C})$$

$$\ddot{\mathfrak{d}}_f \triangleq G_c \ddot{\theta}_f - \acute{\mathfrak{U}}$$

$$\lambda = -\left[G_c \mathcal{M}^{-1} G_c^*\right]^{-1} \ddot{\mathfrak{d}}_f$$

Constraint forces

Constraint violation accels

$$\ddot{\theta}_\delta \triangleq \mathcal{M}^{-1} G_c^* \lambda$$

Correction gen accels

$$\ddot{\theta} = \ddot{\theta}_f + \ddot{\theta}_\delta$$

true gen accels

Simplified dynamics equations

Tree forward dynamics

$$\ddot{\theta}_f \triangleq \mathcal{M}^{-1}(\mathcal{T} - \mathcal{C})$$

$$\ddot{\mathfrak{d}}_f \triangleq G_c \ddot{\theta}_f - \acute{\mathfrak{U}}$$

$$\lambda = -\left[G_c \mathcal{M}^{-1} G_c^*\right]^{-1} \ddot{\mathfrak{d}}_f$$

$$\ddot{\theta}_\delta \triangleq \mathcal{M}^{-1} G_c^* \lambda$$

complex expression

$$\ddot{\theta} = \ddot{\theta}_f + \ddot{\theta}_\delta$$

Augmented algorithm overview

1. Use O(N) method to solve $\ddot{\theta}_f \triangleq \mathcal{M}^{-1}(\mathcal{T} - \mathcal{C})$ "free" generalized accels
2. Compute $G_c \mathcal{M}^{-1} G_c^*$ and use to solve for the Lagrange multipliers in $\lambda = -[G_c \mathcal{M}^{-1} G_c^*]^{-1} \ddot{\mathfrak{d}}_f$
3. Use O(N) method to solve $\ddot{\theta}_\delta \triangleq \mathcal{M}^{-1} G_c^* \lambda$ for correction generalized accels
4. Combine the free and correction generalized accels in $\ddot{\theta} = \ddot{\theta}_f + \ddot{\theta}_\delta$

Non-minimal coordinates, DAE approach. Still can use SKO methods for the spanning tree SKO model

Mass matrix – but singular

$$\ddot{\theta} = \mathcal{Y}_C[\mathcal{T} - \mathcal{C}] + \mathcal{M}^{-1}G_c^*\left[G_c\mathcal{M}^{-1}G_c^*\right]^{-1}\acute{\mathfrak{U}}$$

$$\mathcal{Y}_C \triangleq \mathcal{M}^{-1} - \mathcal{M}^{-1}G_c^*\left[G_c\mathcal{M}^{-1}G_c^*\right]^{-1}G_c\mathcal{M}^{-1} \quad \in \mathcal{R}^{\mathcal{N}\times\mathcal{N}}$$

Mass matrix singularity is a consequence of the non-minimal coordinates, DAE approach.

Augmented Dynamics with Loop Constraints

Single loop constraint

For loop constraints, the constraint is on the relative motion across bodies

$$\mathcal{Q}^{\mathrm{rel}}_{\mathrm{x}} \mathcal{V}_{\mathrm{x}} = \mathbf{0}$$

or

$$\mathcal{Q}^{\mathrm{rel}} \left[\mathcal{V}_{\mathrm{x}} - \mathcal{V}_{\mathrm{y}}\right] = \mathbf{0}$$

With

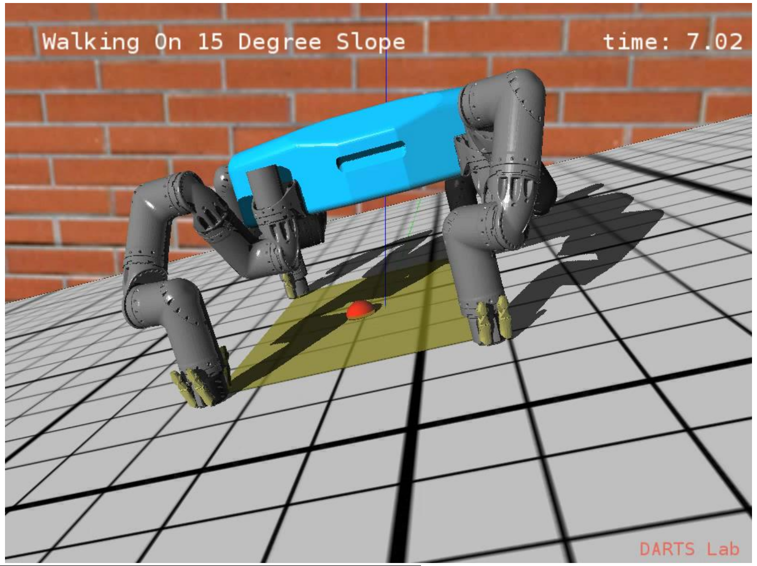
$$\mathcal{V}_{\mathrm{nd}} \triangleq \begin{bmatrix} \mathcal{V}_{\mathrm{x}} \\ \mathcal{V}_{\mathrm{y}} \end{bmatrix}$$

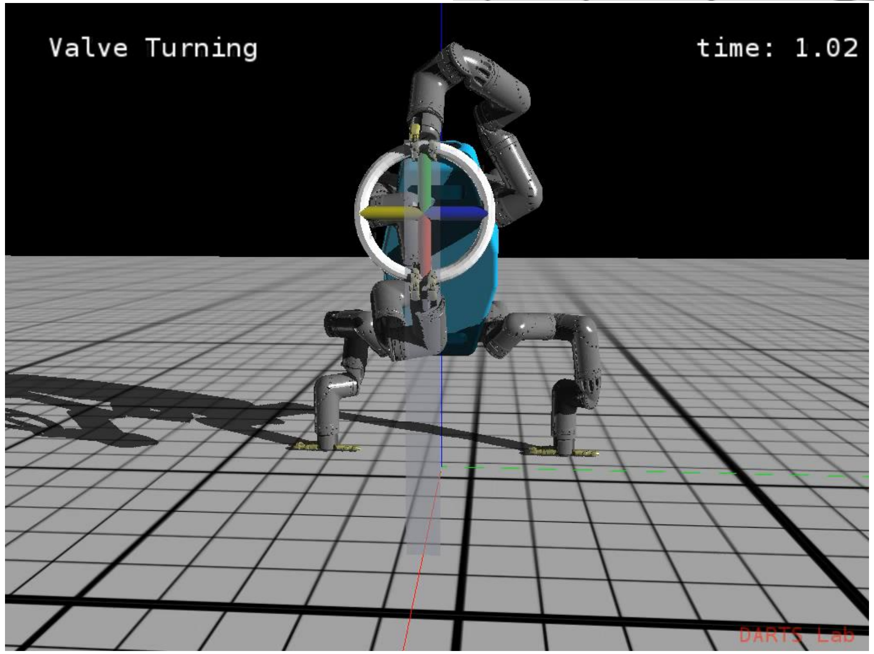
$$\mathcal{Q} \triangleq \left[\mathcal{Q}^{\mathrm{rel}}_{\mathrm{x}}, \ -\mathcal{Q}^{\mathrm{rel}}_{\mathrm{y}}\right]$$

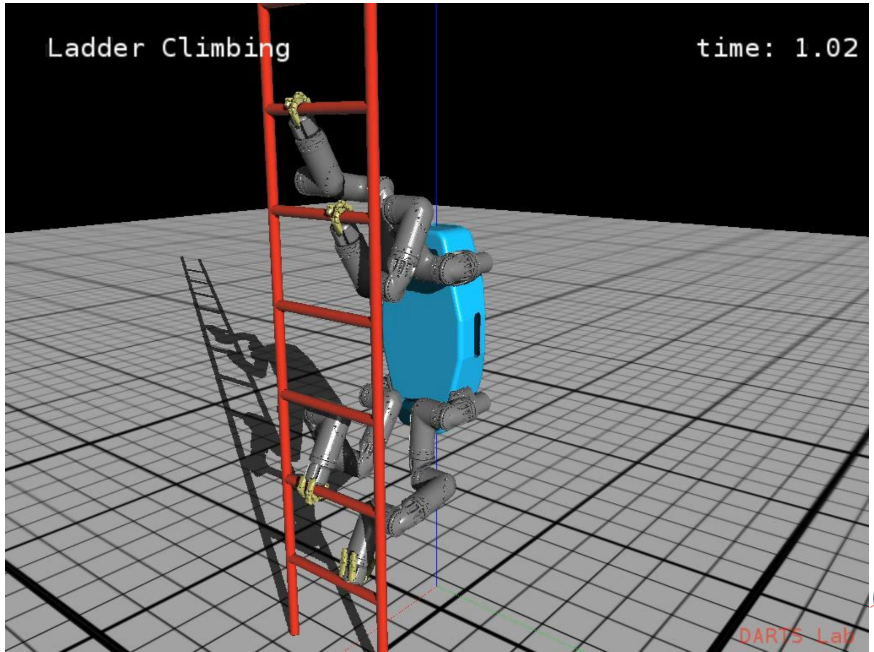
have

$$\mathcal{Q}\mathcal{V}_{\mathrm{nd}} = \mathbf{0}$$

Loop-constraints can change over time







Recall: $\mathcal{B}$ Pick-Off Operator

There are times when we need to narrow attention to the nodes

$$\mathcal{V}_{nd} \triangleq \text{col}\left\{\mathcal{V}_{nd}(k)\right\}_{i=1}^{n} \in \mathcal{R}^{6n_{nd}}$$

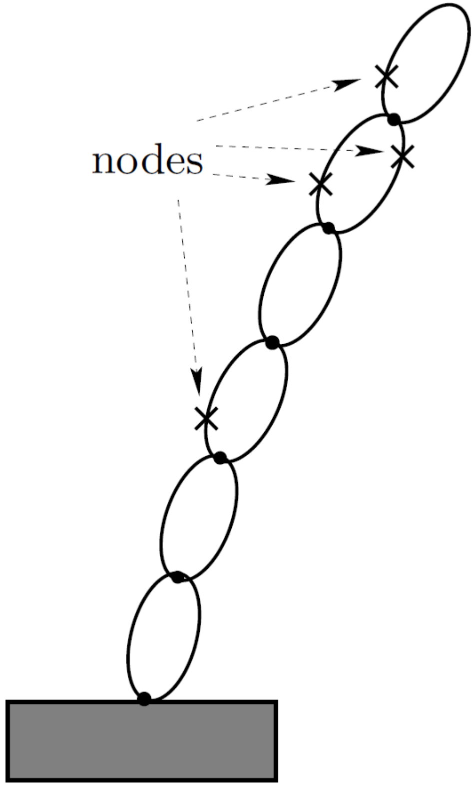
node spatial velocities

$$\mathcal{B} \triangleq \begin{bmatrix} \phi(1, \mathbb{O}_1^0) \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \end{bmatrix}$$

pick-off operator

$$\mathcal{V}(\mathbb{O}_k^i) \overset{1.41}{=} \phi^*(k, \mathbb{O}_k^i)\mathcal{V}(k)$$

single node spatial velocity



$$\mathcal{V}_{nd} = \mathcal{B}^*\mathcal{V}$$

mapping from body to node spatial velocities

Recall: Jacobian Matrix

Combining

$$\mathcal{V}_{nd} \overset{3.47}{=} \mathcal{B}^* \mathcal{V} \quad \text{and} \quad \mathcal{V} = \phi^* H^* \dot{\theta}$$

we have

$$\mathcal{V}_{nd} = \mathcal{J}\,\dot{\theta} \quad \text{where} \quad \boxed{\mathcal{J} \triangleq \mathcal{B}^* \phi^* H^*} \in \mathcal{R}^{6n_{nd} \times \mathcal{N}}$$

Jacobian

operator expression for the Jacobian

The Jacobian relates the generalized velocities to the spatial velocity of one or more nodes of interest

Constraint matrix with loop constraints

$$\boxed{G_c(\theta, t)\dot{\theta} - \mathfrak{U}(t) = \mathbf{0}}$$

$$\mathcal{V}_{nd} = \mathcal{J}\,\dot{\theta} \quad \text{where} \quad \mathcal{J} \triangleq \mathcal{B}^*\phi^*H^*$$

and $$\mathcal{Q}\mathcal{V}_{nd} = \mathbf{0}$$

Hence $$\boxed{G_c = \mathcal{Q}\mathcal{J} = \mathcal{Q}\mathcal{B}^*\phi^*H^*}$$

Recall: dynamics equations

$$\ddot{\theta}_f \triangleq \mathcal{M}^{-1}(\mathcal{T} - \mathcal{C})$$

$$\lambda = -\left[G_c \mathcal{M}^{-1} G_c^*\right]^{-1} \ddot{\mathfrak{d}}_f$$

$$\ddot{\theta}_\delta \triangleq \mathcal{M}^{-1} G_c^* \lambda$$

$$\ddot{\theta} = \ddot{\theta}_f + \ddot{\theta}_\delta$$

Complex, so we look for more simplification

$G_c \mathcal{M}^{-1} G_c^*$ operator simplification

Have $$G_c = \mathcal{Q}\mathcal{J} = \mathcal{Q}\mathcal{B}^*\phi^* H^*$$ $$[\mathbf{I} - H\psi\mathcal{K}]H\phi = H\psi$$

Thus,

$$G_c \mathcal{M}^{-1} G_c^* \overset{11.18,9.52}{=} \mathcal{Q}\mathcal{B}^*\phi^* H^*[\mathbf{I} - H\psi\mathcal{K}]^*\mathcal{D}^{-1}[\mathbf{I} - H\psi\mathcal{K}]H\phi\mathcal{B}\mathcal{Q}^*$$

$$\overset{9.46}{=} \mathcal{Q}\mathcal{B}^*\psi^* H^*\mathcal{D}^{-1}H\psi\mathcal{B}\mathcal{Q}^* = \mathcal{Q}\mathcal{B}^*\Omega\mathcal{B}\mathcal{Q}^* \overset{10.12}{=} \mathcal{Q}\underline{\Lambda}\mathcal{Q}^*$$

where

$$\Omega = \psi^* H^* \mathcal{D}^{-1}\psi H$$ *and* $$\underline{\Lambda} = \mathcal{B}^*\Omega\mathcal{B}$$

$\mathcal{Q}\underline{\Lambda}\mathcal{Q}$ *is much much simpler than where we started, but we can do better*

Operator simplification (contd)

$$\mathcal{M}^{-1}\mathrm{G}_{c}^{*} \overset{11.18,9.52}{=} [\mathbf{I} - \mathrm{H}\psi\mathcal{K}]^{*}\mathcal{D}^{-1}[\mathbf{I} - \mathrm{H}\psi\mathcal{K}]\mathrm{H}\phi\mathcal{B}\mathcal{Q}^{*}$$

$$\overset{9.46}{=} [\mathbf{I} - \mathrm{H}\psi\mathcal{K}]^{*}\mathcal{D}^{-1}\mathrm{H}\psi\mathcal{B}\mathcal{Q}^{*}$$

$$[\mathbf{I} - \mathrm{H}\psi\mathcal{K}]\mathrm{H}\phi = \mathrm{H}\psi$$

identity

Simplified equations of motion

$$\ddot{\theta}_f \triangleq [\mathbf{I} - H\psi\mathcal{K}]^* \mathcal{D}^{-1}\{\mathcal{T} - H\psi[\mathcal{K}\mathcal{T} + \mathcal{P}\mathfrak{a} + \mathfrak{b}]\} - \mathcal{K}^*\psi^*\mathfrak{a}$$

$$\lambda = -[\mathcal{Q}\underline{\Lambda}\mathcal{Q}^*]^{-1}\ddot{\mathfrak{d}}$$

$$\ddot{\theta}_\delta \triangleq [\mathbf{I} - H\psi\mathcal{K}]^* \mathcal{D}^{-1} H\psi\mathcal{B}\mathcal{Q}^*\lambda$$

$$\Omega = \psi^* H^* \mathcal{D}^{-1} \psi H \qquad \underline{\Lambda} = \mathcal{B}^*\Omega\mathcal{B}$$

Simpler, but still complex, so we continue look for more simplification

Simplifying $\Omega = \psi^* H^* \mathcal{D}^{-1} \psi H$

Operational space terminology

$$\underline{\Lambda} \triangleq \mathcal{J}\mathcal{M}^{-1}\mathcal{J}^* \quad \in \mathcal{R}^{6n_{nd} \times 6n_{nd}}$$

$$\Lambda \triangleq \underline{\Lambda}^{-1} = (\mathcal{J}\mathcal{M}^{-1}\mathcal{J}^*)^{-1}$$

*aka **Operational Space Inertia Matrix (OSIM)** in robotics*

*aka **Operational Space Compliance Matrix (OSCM)***

*aka **Extended Operational Space Compliance Matrix (EOSCM)***

simpler operator expression

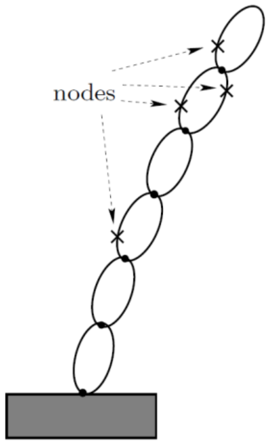
$$\underline{\Lambda} = \mathcal{B}^*\Omega\mathcal{B}$$

where

$$\Omega = \psi^* H^* \mathcal{D}^{-1} \psi H$$

Why Operational Space? Robotics motivation

- The operational space (Khatib) is defined as the task space, that is the world viewed from the end-effector that actually interacts with the world
- Generalization to multiple “end-effectors” for legged robots – where legs meet the ground
- Related to area of “whole-body motion control”
- OSCM always exists, but may be singular. Hence the OSIM may not always exist



$$\Lambda = (\mathcal{J}\mathcal{M}^{-1}\mathcal{J}^*)^{-1}$$

Effective task space mass matrix reflected to the nodes of interest

Focus on Ω EOSCM

The EOSCM is a mapping from external spatial forces at the nodes to the induced spatial accels at the nodes

$$\Omega \triangleq \psi^* \boxed{H^* \mathcal{D}^{-1} H} \psi \in \mathcal{R}^{6n \times 6n}$$

block-diagonal

The EOSCM contains the Backwards Lyapunov form:

$$\boxed{\mathbb{Z} = \mathbb{A}^* \mathbb{X} \mathbb{B}}$$

Generalized Backward Lyapunov Equation for SKO Models

Generalized Forward Lyapunov decomposition

With A & B being SPO operators, and X block diagonal, then

$$Z \triangleq \mathbb{A} X \mathbb{B}^*$$

$$X = Y - \mathcal{E}_{\mathbb{A}} Y \mathcal{E}_{\mathbb{B}}^*$$

$$Z = Y + \tilde{\mathbb{A}} Y + Y \tilde{\mathbb{B}}^*$$

Like CRB

$$Y(k) = \sum_{\forall j \in \mathbf{C}(k)} \mathbb{A}(k,j)\, Y(j)\, \mathbb{B}^*(k,j) + X(k)$$

Look at the $\mathcal{Z} = \mathbb{A}^*\mathcal{X}\mathbb{B}$ product

- This product is the dual to the $\mathcal{Z} \triangleq \mathbb{A}\mathcal{X}\mathbb{B}^*$ product used for understanding the mass matrix structure
- Why is this dual product important?
  - It shows up in products of the $G_c\mathcal{M}^{-1}G_c^*$ form in dynamics analysis
  - One example in cut-joint closed-chain dynamics computations
  - Another example is that of operational space dynamics in robotics

Generalized Backward Lyapunov docomposition

Dual to the forward Lyapunov decomposition

$$Z = \mathbb{A}^* X \mathbb{B}$$

$$X = Y - \mathrm{diagOf}\left\{ \mathcal{E}_{\mathbb{A}}^* Y \mathcal{E}_{\mathbb{B}} \right\}$$

$$Z = Y + \tilde{\mathbb{A}}^* Y + Y \tilde{\mathbb{B}} + R$$

block diagonal

$$R(i,j) = \mathbb{1}_{[i \nprec \nsucc j,\, k = \wp(i,j)]}\ \mathbb{A}^*(k,i)\ Y(k)\ \mathbb{B}(k,j)$$

Diagonal terms

The block-diagonal Y terms can be computed via a O(N) scatter algorithm

$$X = Y - \mathrm{diagOf}\left\{\mathcal{E}_{\mathbb{A}}^{*} Y \mathcal{E}_{\mathbb{B}}\right\}$$

⬇

$$Y(k) = \mathbb{A}^{*}(\wp(k), k) Y(\wp(k)) \mathbb{B}(\wp(k), k) + X(k)$$

$$\begin{cases} \textbf{for } \textit{all nodes } \mathbf{k} \quad \textit{(base-to-tips scatter)} \\ \quad Y(k) = \mathbb{A}^{*}(\wp(k), k) Y(\wp(k)) \mathbb{B}(\wp(k), k) + X(k) \\ \textbf{end loop} \end{cases}$$

General expression for the elements

related bodies

diagonal

$$
\mathbb{Z}(i,j) = \begin{cases} Y(i) & \textit{for } i = j \\ \mathbb{A}^*(k,i)\,\mathbb{Z}(k,j) & \textit{for } i \prec k \preceq j, \quad k = \wp(i) \\ \mathbb{Z}(i,k)\,\mathbb{B}(k,j) & \textit{for } i \succeq k \succ j, \quad k = \wp(j) \\ \mathbb{A}^*(k,i)\,Y(k)\,\mathbb{B}(k,j) & \textit{for } i \nsucc j,\ j \nsucc i,\ k = \wp(i,j) \end{cases}
$$

unrelated bodies

OR

$$
\mathbb{Z}(i,j) = \mathbb{A}^*(k,i)\ Y(k)\ \mathbb{B}(k,j)
$$

Derivation

$$\mathrm{Z} = \mathbb{A}^* \mathrm{X} \mathbb{B}$$

⇓

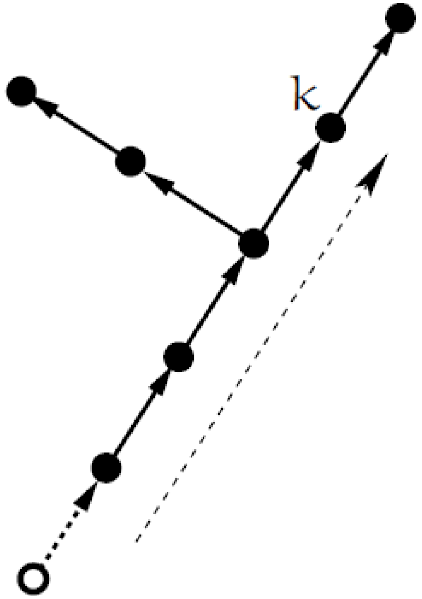
$$\mathrm{Z}(\mathrm{i},\mathrm{j}) = \mathbb{A}^*(\mathrm{p},\mathrm{i})\ \mathrm{X}(\mathrm{p},\mathrm{q})\ \mathbb{B}(\mathrm{q},\mathrm{j}) = \mathbb{A}^*(\mathrm{p},\mathrm{i})\ \mathrm{X}(\mathrm{p})\ \mathbb{B}(\mathrm{p},\mathrm{j})$$
$$= \mathbb{1}_{[\mathrm{i} \preceq \mathrm{p}]}\ \mathbb{1}_{[\mathrm{j} \preceq \mathrm{p}]}\ \mathbb{A}^*(\mathrm{p},\mathrm{i})\ \mathrm{X}(\mathrm{p})\ \mathbb{B}(\mathrm{p},\mathrm{j})$$

⇓

$$\mathrm{Z}(\mathrm{i},\mathrm{j}) = \mathbb{A}^*(\mathrm{k},\mathrm{i})\ \mathrm{Y}(\mathrm{k})\ \mathbb{B}(\mathrm{k},\mathrm{j})$$

Algorithm Structure

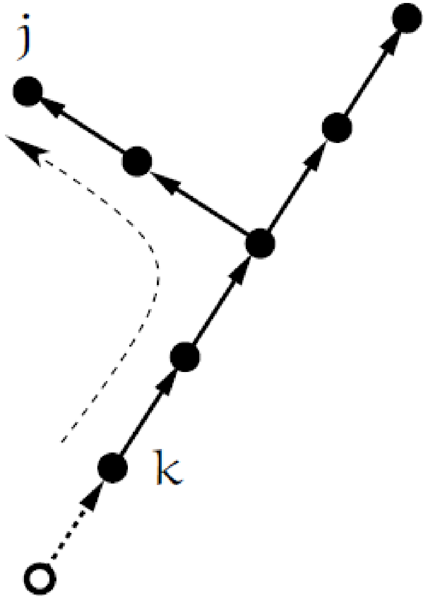
diagonal elements



Case 1: $k = j$

$Z(k,k) = Y(k)$

related elements

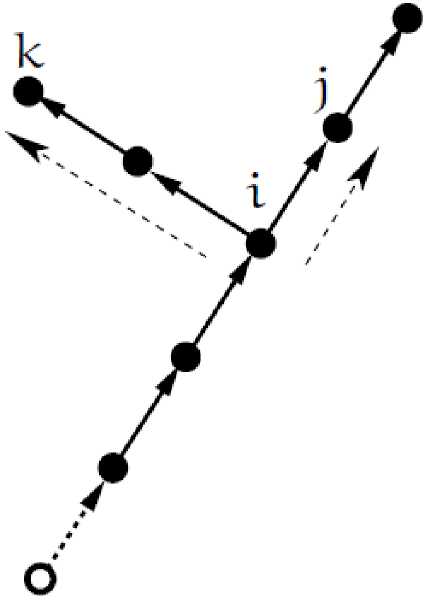


Case 2 & 3: $k \succ j$

$Z(j,k) = \mathbb{A}^*(k,j)Y(k)$

$Z(k,j) = Y(k)\mathbb{B}(k,j)$

unrelated elements



Case 4: $i \succ k, i \succ j$

$Z(k,j) = \mathbb{A}^*(i,k)Y(i)\mathbb{A}(i,j)$

Serial-chain case simplification

$$\mathbb{Z} = \mathbb{A}^* \mathbb{X} \mathbb{B}$$

All bodies are related in a serial-chain, and hence

$$\mathbb{Z} = \mathbb{Y} + \tilde{\mathbb{A}}^* \mathbb{Y} + \mathbb{Y} \tilde{\mathbb{B}} + \mathbb{R}$$

X

all zero

$$\mathbb{X} = \mathbb{Y} - \mathcal{E}_{\mathbb{A}}^* \mathbb{Y} \mathcal{E}_{\mathbb{B}}$$

block-diagonal

Back to Extended Operational Space Compliance Matrix

Applying the Backward Lyapunov decomposition to the EOSCM

$$\Omega \stackrel{\triangle}{=} \psi^* H^* \mathcal{D}^{-1} H \psi$$

$$\Downarrow$$

$$\Omega = \Upsilon + \tilde{\psi}^* \Upsilon + \Upsilon \tilde{\psi} + R$$

$$H^* \mathcal{D}^{-1} H = \Upsilon - \text{diagOf}\left\{ \mathcal{E}_{\psi}^* \Upsilon \mathcal{E}_{\psi} \right\}$$

$$\Upsilon(k) = \psi^*(\wp(k), k)\Upsilon(\wp(k))\psi(\wp(k), k) + H^*(k)\mathcal{D}^{-1}(k)H(k)$$

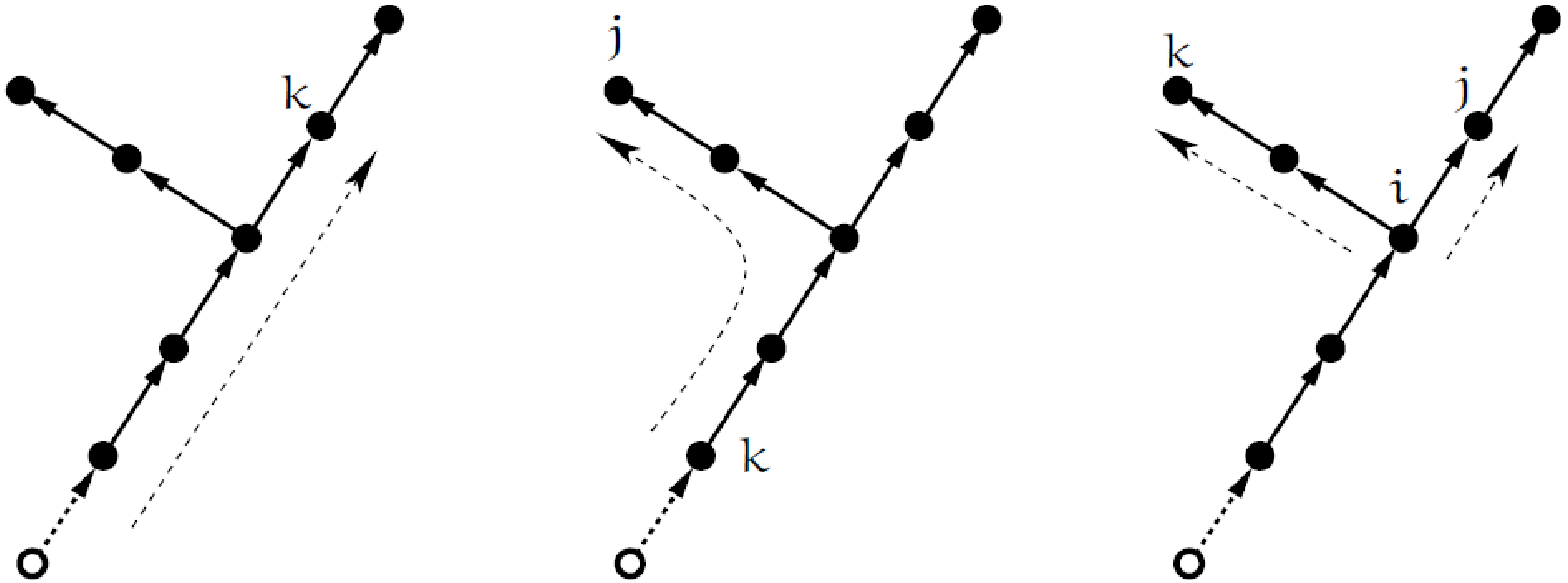
Component values

$$\begin{cases} \textbf{for } \textit{all nodes } \mathbf{k} \quad \textit{(base-to-tips scatter)} \\ \qquad \Upsilon(k) = \psi^*(\wp(k), k)\Upsilon(\wp(k))\psi(\wp(k), k) + H^*(k)\mathcal{D}^{-1}(k)H(k) \\ \textbf{end loop} \end{cases}$$

$$\Omega(i, j) = \begin{cases} \Upsilon(i) & \textit{for } i = j \\ \Omega(i, k)\psi(k, j) & \textit{for } i \succeq k \succ j, \quad k = \wp(j) \\ \Omega^*(j, i) & \textit{for } i \prec j \\ \Omega(i, k)\psi(k, j) & \textit{for } i \not\succ j,\ j \not\succ i,\ k = \wp(i, j) \end{cases}$$

Scatter Algorithm Structure

diagonal elements | related elements | unrelated elements



Case 1: $k = j$
$\mathsf{Z}(k,k) = \mathsf{Y}(k)$

Case 2 & 3: $k \succ j$
$\mathsf{Z}(j,k) = \mathbb{A}^*(k,j)\mathsf{Y}(k)$
$\mathsf{Z}(k,j) = \mathsf{Y}(k)\mathbb{B}(k,j)$

Case 4: $i \succ k, i \succ j$
$\mathsf{Z}(k,j) = \mathbb{A}^*(i,k)\mathsf{Y}(i)\mathbb{A}(i,j)$

Another example of low-cost algorithms developed from the SOA methodology. This is the fastest available algorithm to date for the EOSCM.

Serial-chain simplification

For a serial-chain system, the R term is zero. Thus

$$\Omega = \Upsilon + \tilde{\psi}^* \Upsilon + \Upsilon \tilde{\psi}$$

$$H^* \mathcal{D}^{-1} H = \Upsilon - \mathcal{E}_{\psi}^* \Upsilon \mathcal{E}_{\psi}$$

$\underline{\mathrm{G_c}}\mathcal{M}^{-1}\mathrm{G}_c^*$ simplification steps summary

$$\mathrm{G_c}\mathcal{M}^{-1}\mathrm{G}_c^*$$

↓ *Avoids computation and inversion of mass matrix, and expensive products*

$$\mathcal{Q}\mathcal{B}^*\phi^*\mathrm{H}^*[\mathbf{I}-\mathrm{H}\psi\mathcal{K}]^*\mathcal{D}^{-1}[\mathbf{I}-\mathrm{H}\psi\mathcal{K}]\mathrm{H}\phi\mathcal{B}\mathcal{Q}^*$$

↓ *Reduces SPO products from 4 down to 2*

$$\mathcal{Q}\mathcal{B}^*\psi^*\mathrm{H}^*\mathcal{D}^{-1}\mathrm{H}\psi\mathcal{B}\mathcal{Q}^* = \mathcal{Q}\mathcal{B}^*\Omega\mathcal{B}\mathcal{Q}^* \stackrel{10.12}{=} \mathcal{Q}\underline{\Lambda}\mathcal{Q}^*$$

↓ *Reduces costs to O(N^2) using the Υ block-diagonal term*

$$\Omega = \Upsilon + \tilde{\psi}^*\Upsilon + \Upsilon\tilde{\psi} + \mathrm{R}$$

↓ *Reduces costs to O(N) and just the cut-joint terms*

$$\mathcal{Q}\mathcal{B}^*\Omega\mathcal{B}\mathcal{Q}^*$$

Significant reduction in cost by exploiting structure using SOA operators

Back to Augmented Dynamics with Loop Constraints

Recap

$$\underline{\Lambda} \triangleq \mathfrak{J}\mathcal{M}^{-1}\mathfrak{J}^* \quad \in \mathcal{R}^{6n_{nd} \times 6n_{nd}}$$

$$\Lambda \triangleq \underline{\Lambda}^{-1} = (\mathfrak{J}\mathcal{M}^{-1}\mathfrak{J}^*)^{-1}$$

*aka **Operational Space Inertia Matrix (OSIM)** in robotics*

*aka **Operational Space Compliance Matrix (OSCM)***

*aka **Extended Operational Space Compliance Matrix (EOSCM)***

simpler operator expression

$$\underline{\Lambda} = \mathcal{B}^*\Omega\mathcal{B} \qquad \textit{where} \qquad \Omega = \psi^* H^* \mathcal{D}^{-1} \psi H$$

Low-cost recursive algorithms

$$\ddot{\theta}_f \triangleq [\mathbf{I} - H\psi\mathcal{K}]^*\mathcal{D}^{-1}\{\mathcal{T} - H\psi[\mathcal{K}\mathcal{T} + \mathcal{P}\mathfrak{a} + \mathfrak{b}]\} - \mathcal{K}^*\psi^*\mathfrak{a}$$

$$\lambda = -[\mathcal{Q}\underline{\Lambda}\mathcal{Q}^*]^{-1}\ddot{\mathfrak{d}}$$

$$\ddot{\theta}_\delta \triangleq [\mathbf{I} - H\psi\mathcal{K}]^*\mathcal{D}^{-1}H\psi\mathcal{B}\mathcal{Q}^*\lambda$$

EOSCM recursive computation

O(N) ATBI forward dynamics

$$\Omega = \psi^*H^*\mathcal{D}^{-1}\psi H$$

$$\underline{\Lambda} = \mathcal{B}^*\Omega\mathcal{B}$$

Augmented method comments

- Even though the augmented method approach does not lend itself directly to be SKO model, we find that the SKO algorithms can be used to efficiently carry out each of the augmented method steps
- The minimal cuts augmented method is much better than the maximal cuts approach
  - We can take advantage of the fast SKO method
  - Smaller number of constraints to manage error for
- Changing of constraints are easily accommodated by the SKO gather & scatter algorithms
- The augmented approach still remains a non-minimal coordinates and DAE approach
  - Hence some type of error control (eg. Baumgarte, projection, implicit method) is needed when integrating

Recap

Summary

- Looked into the augmented method for closed-chain dynamics (DAE approach)
- Does not have a direct SKO model
- Introduced the notion of operational space inertia matrix (OSIM) and OSCIM
- Discussed the Backward Lyapunov Equation based operator decomposition
- Applied SKO model recursive algorithms for the various steps in the augmented approach

SOA Foundations Track Topics (serial-chain rigid body systems)

1. **Spatial (6D) notation** – spatial velocities, forces, inertias; spatial cross-product, rigid body transformations & properties; parallel axis theorem
2. **Single rigid body dynamics** – equations of motion about arbitrary frame using spatial notation
3. **Serial-chain kinematics** – minimal coordinate formulation, hinges, velocity recursions, Jacobians; first spatial operators; O(N) scatter and gather recursions
4. **Serial-chain dynamics** – equations of motion using spatial operators; Newton–Euler mass matrix factorization; O(N) inverse dynamics; composite rigid body inertia; forward Lyapunov equation; mass matrix decomposition; mass matrix computation; alternative inverse dynamics
5. **Articulated body inertia -** Concept and definition; Riccati equation; alternative force decompositions
6. **Mass matrix factorization and inversion** – spatial operator identities; Innovations factorization of the mass matrix; Inversion of the mass matrix
7. **Recursive forward dynamics** – O(N) recursive forward dynamics algorithm; including gravity and external forces; inter-body forces identity

See https://dartslab.jpl.nasa.gov/References/index.php for publications and references on the SOA methodology.

SOA Generalization Track Topics

8. **Graph theory based structure** – BWA matrices, connection to multibody systems
9. **Tree topology systems** – generalization to tree topology rigid body systems, SKO/SPO operators, gather/scatter algorithms
10. **Closed-chain dynamics (cut-joint)** – holonomic and non-holonomic constraints, cut-joint method, operational space inertia, projected dynamics
11. **Closed-chain dynamics (constraint embedding)** – constraint embedding for graph transformation, minimal coordinate closed-chain dynamics
12. **Flexible body dynamics** – Extension to flexible bodies, modal representations, recursive flexible body dynamics