Dynamics and Real-Time Simulation (DARTS) Laboratory

Spatial Operator Algebra (SOA)

9. Multibody Graph Generalizations

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

Previous Session Recap

Previous Session Recap

- Used graph theory concepts to define the notion of BWA matrices for trees and graphs
- Showed that tree BWA matrices have a well defined 1-resolvent matrix
- Showed that the spatial operator for serial chains is a tree BWA ma$\mathcal{E}_\phi$
- Developed equations of motion for a tree system using tree BWA operators
- The spatial operator expressions remain unchanged from serial to tree systems

Comments on tree equations of motion

$$\begin{aligned}\mathcal{T} &= \mathrm{H}\,\phi\,\left[\mathbf{M}\,\phi^{*}\,\left(\mathrm{H}^{*}\,\ddot{\theta} + \mathfrak{a}\right) + \mathfrak{b}\right] \\ &= \mathcal{M}(\theta)\ddot{\boldsymbol{\theta}} + \mathcal{C}(\theta, \dot{\boldsymbol{\theta}})\end{aligned}$$

- The equations of motion are identical in form at the operator level to the serial-chain equations of motion!
  - These hold for arbitrary system size and branching
- The differences are
  - At the component level we now have to work with multiple children bodies
  - The operator structure is different – however they are both tree BWA matrices

Comments

- We have been able to generalize the notion of the $\mathcal{E}_\phi$ and ϕ operators & **structure** from serial chains to trees
    - The crucial step was to look at the general property of BWA matrices associated with graphs and to recognize that $\mathcal{E}_\phi$ is a tree BWA matrix
- From the tree BWA property alone we could
    - Establish the existence of the 1-resolvent
    - Establish sparsity property based on topological structure
    - Establish disjointedness property of the 1-resolvent and its transpose
    - Establish the chain rule property for the elements of the 1-resolvent
- We did not require canonical indexing, and triangularity assumptions at all!

Comments (contd)

- We have earlier used a canonical serial chain to develop the SOA operator analysis and algorithm
  - The simple structure of canonical serial-chains allowed us to build up the techniques as well as our intuition
- But as we are starting to see, neither the serial-chain, nor the canonical nature are really that important
  - It is the BWA part that matters
- The specific block entries of the BWA matrix did not matter either
  - In fact they do not even have to be *square* or *rigid body transformation matrices!*

SKO model for Multibody Systems

Structure: Insight and Implications

increasing analytical structure

Absolute coordinates	Non-minimal, explicit constraints	DAE rabbit hole, sparsity structure
Lagrangian approach	$\mathcal{M}(\theta)$	Opaque, non-singular matrix
Kane's approach	$\mathcal{M}(\theta) = C^* \ \mathbf{M} \ C$	C non-square partial velocity matrix
NE factorization	$\mathcal{M}(\theta) = H\phi\mathbf{M}\boxed{\phi^* H^*}$	ϕ is a square matrix
Connectivity structure	$\phi \triangleq (\mathbf{I} - \mathcal{E}_\phi)^{-1}$	ϕ is a 1-resolvent
Recursive structure	$\mathcal{E}_\phi$	$\mathcal{E}_\phi$ **is a BWA matrix**

Opens the gates to SOA analysis

Generalization path forward

- For a tree rigid multibody system we have developed
  - the operator expressions for the equations of motion
  - the Newton-Euler factorization of the mass matrix
  - These expressions are identical in form to those for serial-chain rigid multibody systems
- *Spoiler alert* – the rest of the operator analysis including the factorization and inversion of the mass matrix for serial-chain rigid body systems carries over to tree multibody systems as well
- How general is such analysis?

Spatial Kernel Operator definition

- We refer to the $\mathcal{E}_{\mathbb{A}}$ BWA matrix in the context of tree multibody systems as a ***Spatial Kernel Operator (SKO)***
- The corresponding 1-resolvent matrix $\mathbb{A}$ is referred to as the ***Spatial Propagation Operator (SPO)***
- The SPO operator is entirely determined by it SKO operator – and hence the ‘kernel’ terminology

Multibody SKO model definition

- Multibody model with tree topology structure
- Has SKO and SPO operators
- Block-diagonal, full-rank H operator
- Block-diagonal, positive-definite M operator

$$\boxed{\begin{aligned} \mathcal{V} &= \mathrm{A}^*\mathrm{H}^*\dot{\theta} \\ \alpha &= \mathrm{A}^*(\mathrm{H}^*\ddot{\theta} + \mathfrak{a}) \\ \mathfrak{f} &= \mathrm{A}(\mathbf{M}\alpha + \mathfrak{b}) \\ \mathcal{T} &= \mathrm{H}\mathfrak{f} \end{aligned}}$$

$$\boxed{\mathcal{T} = \mathcal{M}(\theta)\ddot{\theta} + \mathcal{C}(\theta, \dot{\theta})}$$

$$\mathcal{M}(\theta) \triangleq \mathrm{H}\mathrm{A}\mathbf{M}\mathrm{A}^*\mathrm{H}^* \in \mathcal{R}^{\mathcal{N}\times\mathcal{N}}$$

$$\mathcal{C}(\theta, \dot{\theta}) \triangleq \mathrm{H}\mathrm{A}(\mathbf{M}\mathrm{A}^*\mathfrak{a} + \mathfrak{b}) \in \mathcal{R}^{\mathcal{N}}$$

Equations of motion

If these conditions are satisfied, then the model is said to satisfy the **SKO model conditions**, and be referred to as an **SKO model** with the associated $(\mathrm{H}, \mathrm{A}, \mathrm{M})$ operator triplet.

Comments on SKO model

No assumptions have been made about the specific nature of the $\mathcal{E}_{\mathbb{A}}$ and $\mathbb{A}$ operators

Potential generalizations

- Elements are not rigid body $\phi(k+1,k)$
- Elements may be non-square
- Elements may be of different sizes
- Elements may be singular
- Even the tree-topology requirement can be relaxed

Compatible SKO model operators

- The operators have to be derived from the same underlying multibody system – and hence be “compatible” in order to form an SKO model.
- A multibody system’s SKO model is not unique, and depends on the choice of base body, body reference frames etc.

Properties of SKO models

- We will now derive the properties of an SKO model
- These properties will be generalizations of ones we encountered for serial-chain rigid body systems
- In future, any system satisfying the SKO model conditions will automatically inherit these properties

SPO Scatter recursions for SKO Models

Recall: Base-to-tips structure-based O(N) scatter recursion for serial-chain ϕ

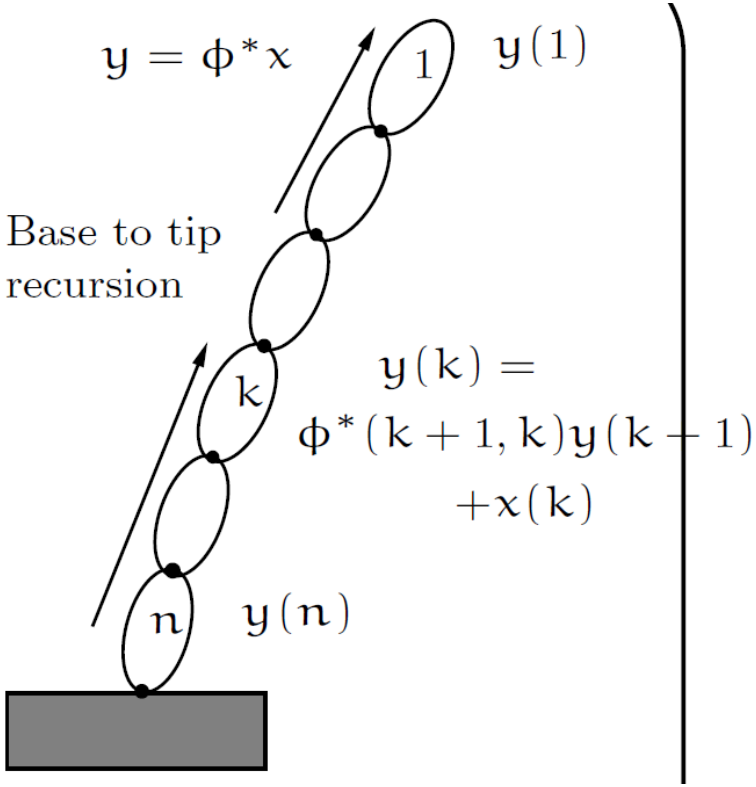
operator transpose/vector product

$$\boxed{y = \phi^* x}$$

- *Applies to any x*
- *Does not require explicit computation of ϕ at all*
- *Only depends on elements of $\mathcal{E}_\phi$*

```
                y(n+1) = 0
for k    n···1
    $y(k) = \phi^*(k+1, k) y(k+1) + x(k)$
end loop
```

O(N) structure-based, base-to-tip scatter recursion



Algorithm flow

Example – link velocity computation

Generalized base-to-tips structure-based scatter recursion

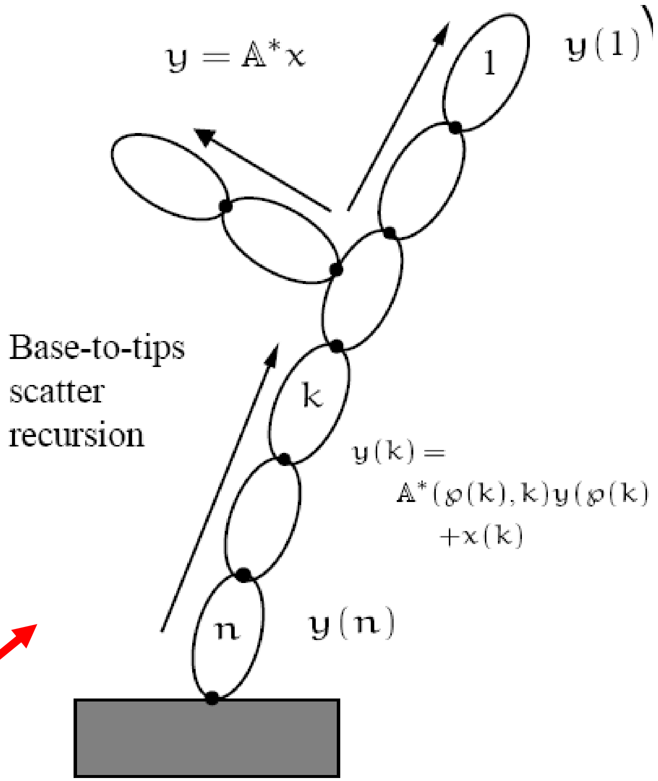
operator transpose/vector product

$$\boxed{y = \mathbb{A}^* x}$$

- *Applies to any x*
- *Does not require explicit computation of $\mathbb{A}$ at all*
- *Only depends on elements of $\mathcal{E}_A$*

```
for all nodes $k$ (base-to-tips scatter)
    $y(k) = \mathbb{A}^*(\wp(k),k)\,y(\wp(k)) + x(k)$
end loop
```

O(N) structure-based scatter algorithm



$y = \mathbb{A}^* x$
1
$y(1)$
Base-to-tips scatter recursion
k
$y(k) = \mathbb{A}^*(\wp(k),k)y(\wp(k)) + x(k)$
n
$y(n)$

Algorithm flow

Derivation of scatter recursion (using tensor notation)

Have $\boxed{y = \mathbb{A}^* x}$

$$y(k) = \sum_{j=1}^{n} \mathbb{A}^*(j,k)x(j) \overset{8.18}{=} \sum_{\forall j \succeq k} \mathbb{A}^*(j,k)x(j)$$

$$\overset{9.10,8.20}{=} \mathbb{A}^*(\wp(k),k) \sum_{\forall i \succeq \wp(k)} \mathbb{A}^*(i,\wp(k))x(i) + x(k)$$

$$\overset{9.10}{=} \mathbb{A}^*(\wp(k),k)y(\wp(k)) + x(k)$$

SPO Gather recursions for SKO Models

Recall: Tips-to-base structure-based O(N) gather recursion for serial chain

ϕ

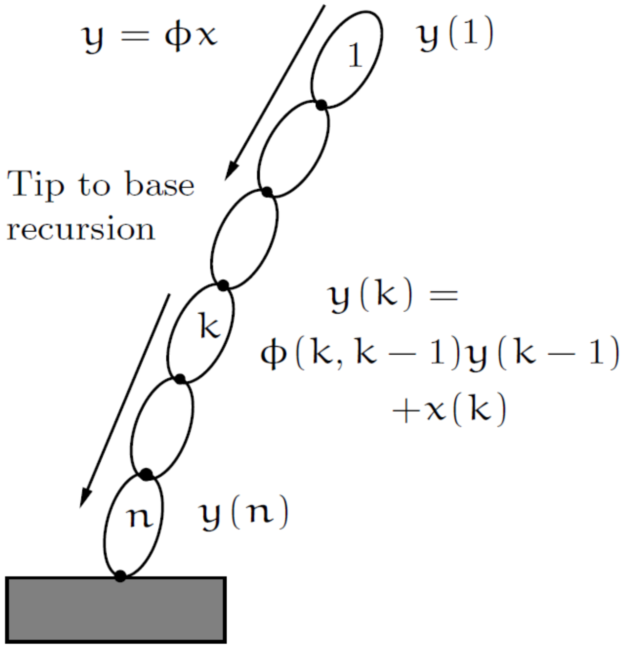
operator/vector product

$$\boxed{y = \phi x}$$

- *Applies to any x*
- *Does not require explicit computation of ϕ at all*
- *Only depends on elements of $\mathcal{E}_\phi$*

```
        $y(0) = 0$
for $k$   $1 \cdots n$
    $y(k) = \phi(k, k-1) y(k-1) + x(k)$
end loop
```

O(N) structure-based tip-to-base gather recursion



Algorithm flow

Example – torque for end-effector force

Generalized tips-to-base structure-based gather recursion

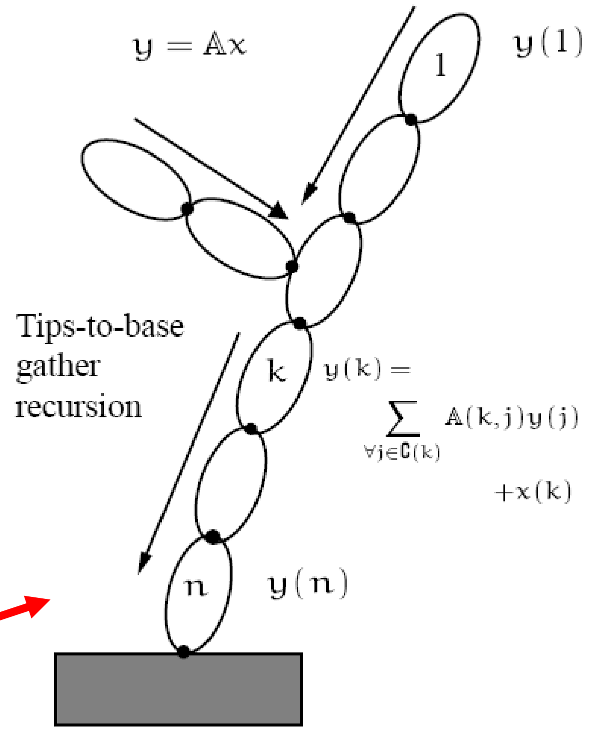
operator/vector product

$$\boxed{y = \mathbb{A}x}$$

- *Applies to any x*
- *Does not require explicit computation of $\mathbb{A}$ at all*
- *Only depends on elements of $\mathcal{E}_A$*

$$\left\{\begin{array}{l} \textbf{for } \textit{all nodes } \mathbf{k} \quad \textit{(tips-to-base gather)} \\ \quad y(k) = \boxed{\sum_{\forall i \in \mathcal{C}(k)} \mathbb{A}(k,i)y(i)} + x(k) \\ \textbf{end loop} \end{array}\right.$$

O(N) structure-based gather algorithm



$y = \mathbb{A}x$
1
$y(1)$
Tips-to-base gather recursion
k
$y(k) = \sum_{\forall j \in \mathcal{C}(k)} \mathbb{A}(k,j)y(j) + x(k)$
n
$y(n)$

Algorithm flow

Derivation of gather recursion (using tensor notation)

Have $\boxed{\mathrm{y} = \mathbb{A}\mathrm{x}}$

Thus

$$\begin{aligned}
\mathrm{y}(\mathrm{k}) &= \mathbb{A}(\mathrm{k},\mathrm{j})\ \mathrm{x}(\mathrm{j}) \\
&= \delta_{\mathrm{k},\wp^{\mathrm{l}}(\mathrm{j})}\ \mathbb{A}(\mathrm{k},\wp^{\mathrm{l}-1}(\mathrm{j}))\cdots\mathbb{A}(\wp(\mathrm{j}),\mathrm{j})\ \mathrm{x}(\mathrm{j}) + \mathrm{x}(\mathrm{k}) \\
&= \delta_{\mathrm{k},\wp(\mathrm{m})}\ \mathbb{A}(\mathrm{k},\mathrm{m})\ \delta_{\mathrm{m},\wp^{\mathrm{l}-1}(\mathrm{j})} \\
&\qquad * \left[\mathbb{A}(\mathrm{m},\wp^{\mathrm{l}-2}(\mathrm{j}))\cdots\mathbb{A}(\wp(\mathrm{j}),\mathrm{j})\ \mathrm{x}(\mathrm{j}) + \mathrm{x}(\mathrm{m})\right] + \mathrm{x}(\mathrm{k}) \\
&= \delta_{\mathrm{k},\wp(\mathrm{m})}\ \mathbb{A}(\mathrm{k},\mathrm{m})\ \mathrm{x}(\mathrm{m}) + \mathrm{x}(\mathrm{k})
\end{aligned}$$

Related scatter/gather recursions

Similar operator expressions to recursions mapping

$$\boxed{y = \tilde{\mathbb{A}} x} \quad \Longrightarrow \quad y(k) = \sum_{\forall j \in \mathbf{C}(k)} \mathbb{A}(k, j) \left[y(j) + x(j) \right]$$

$$\boxed{y = \tilde{\mathbb{A}}^* x} \quad \Longrightarrow \quad y(k) = \mathbb{A}^*(\wp(k), k) \left[y(\wp(k)) + x(\wp(k)) \right]$$

Newton-Euler Inverse Dynamics for SKO Models

Inverse dynamics

- Need to compute RHS of

$$\mathcal{T} = \mathcal{M}(\theta)\ddot{\theta} + \mathcal{C}(\theta, \dot{\theta})$$

- First focus on the $\mathcal{M}(\theta)\ddot{\theta}$ mass matrix term
- One option is to compute the $\mathcal{M}(\theta)$ mass matrix and then the $\mathcal{M}(\theta)\ddot{\theta}$ product

$$\mathcal{M}\,\ddot{\theta} = \underbrace{H\,\phi\,\mathbf{M}\,\phi^*\,H^*}_{\mathcal{M}}\,\ddot{\theta}$$

  - This would be at the minimum a $O(\mathcal{N}^2)$ cost process for computing the $\mathcal{M}(\theta)$ matrix using the optimal CRB algorithm seen earlier
- Can we do better?

Recall: Exploiting Newton-Euler factorization for computing $\mathcal{M}(\theta)\ddot{\theta}$ rigid-body serial-chains

$\mathcal{M}(\theta)\ddot{\theta}$ can be computed using a sequence of O(N) operator/vector products

$$\mathcal{M}\,\ddot{\theta} = \underbrace{H\,\underbrace{\phi\,\underbrace{\mathbf{M}\,\underbrace{\phi^*\,\underbrace{H^*\,\ddot{\theta}}_{y_1}}_{y_2=\phi^* y_1}}_{y_3=\mathbf{M}y_2}}_{y_4=\phi y_3}}_{y_5=Hy_4}$$

diagonal *matrix times vector*

recursive ***scatter*** *alg.*

diagonal *matrix times vector*

recursive ***gather*** *alg.*

diagonal *matrix times vector*

This is another example of being able to directly map operator expressions into low-cost recursive algorithms

Newton-Euler O(N) Recursive Inverse Dynamics for Serial-Chains

Overall O(N) Newton-Euler recursive inverse dynamics

$$
\begin{aligned}
\mathcal{T} &= H\,\phi\,\left[\mathbf{M}\,\phi^*\,\left(H^*\,\ddot{\theta} + \mathfrak{a}\right) + \mathfrak{b}\right] \\
&= \mathcal{M}(\theta)\ddot{\theta} + \mathcal{C}(\theta, \dot{\theta})
\end{aligned}
$$

Originally developed by Luh, Walker & Paul

```
$\mathcal{V}(n+1) = \mathbf{0}, \quad \alpha(n+1) = \mathbf{0}$
for $k$  $n \cdots 1$
    $\mathcal{V}(k) = \phi^*(k+1,k)\mathcal{V}(k+1) + H^*(k)\dot{\theta}(k)$
    $\alpha(k) = \phi^*(k+1,k)\alpha(k+1) + H^*(k)\ddot{\theta}(k) + \mathfrak{a}(k)$
end loop
```

Base-to-tip O(N) recursive ***scatter*** *sweep*

```
$\mathfrak{f}(0) = \mathbf{0}$
for $k$  $1 \cdots n$
    $\mathfrak{f}(k) = \phi(k,k-1)\mathfrak{f}(k-1) + M(k)\alpha(k) + \mathfrak{b}(k)$
    $\mathcal{T}(k) = H(k)\mathfrak{f}(k)$
end loop
```

Tip-to-base O(N) recursive ***gather*** *sweep*

SKO model equations of motion

Equations of motion

$$\boxed{\begin{aligned} \mathcal{V} &= A^* H^* \dot{\theta} \\ \alpha &= A^*(H^* \ddot{\theta} + \mathfrak{a}) \\ \mathfrak{f} &= A(\mathbf{M}\alpha + \mathfrak{b}) \\ \mathcal{T} &= H\mathfrak{f} \end{aligned}}$$

$$\boxed{\mathcal{T} = \mathcal{M}(\theta)\ddot{\theta} + \mathcal{C}(\theta, \dot{\theta})}$$

$$\mathcal{M}(\theta) \triangleq HA\mathbf{M}A^*H^* \in \mathcal{R}^{\mathcal{N}\times\mathcal{N}}$$

$$\mathcal{C}(\theta, \dot{\theta}) \triangleq HA(\mathbf{M}A^*\mathfrak{a} + \mathfrak{b}) \in \mathcal{R}^{\mathcal{N}}$$

mass matrix
Newton-Euler
Factorization

Coriolis vector

Generalized O(N) NE inverse dynamics

Valid for any SKO model:

```
for all nodes k   (base-to-tips scatter)
    $\mathcal{V}(k) = \mathbb{A}^*(\wp(k), k)\mathcal{V}(\wp(k)) + H^*(k)\dot{\theta}(k)$
    $\alpha(k) = \mathbb{A}^*(\wp(k), k)\alpha(\wp(k)) + H^*(k)\ddot{\theta}(k) + \mathfrak{a}(k)$
end loop
```

Base-to-tip O(N) recursive ***scatter*** *sweep*

```
for all nodes k   (tips-to-base gather)
    $\mathfrak{f}(k) = \boxed{\sum_{\forall j \in \mathbf{C}(k)} \mathbb{A}(k, j)\mathfrak{f}(j)} + M(k)\alpha(k) + \mathfrak{b}(k)$
    $\mathcal{T}(k) = H(k)\mathfrak{f}(k)$
end loop
```

Tip-to-base O(N) recursive ***gather*** *sweep*

Forward Lyapunov Equation for SKO models

Recall: Forward Lyapunov Equation for CRBs (rigid-body serial-chain)

CRB recursion

$$\mathcal{R}(k) = \phi(k, k-1)\mathcal{R}(k-1)\phi^*(k, k-1) + M(k)$$

Define CRB spatial operator

$$\mathcal{R} \triangleq \text{diag}\left\{\mathcal{R}(k)\right\}_{k=1}^{n} \in \mathscr{R}^{6n\times 6n}$$

Can re-express as CRB “forward Lyapunov equation” using spatial operators

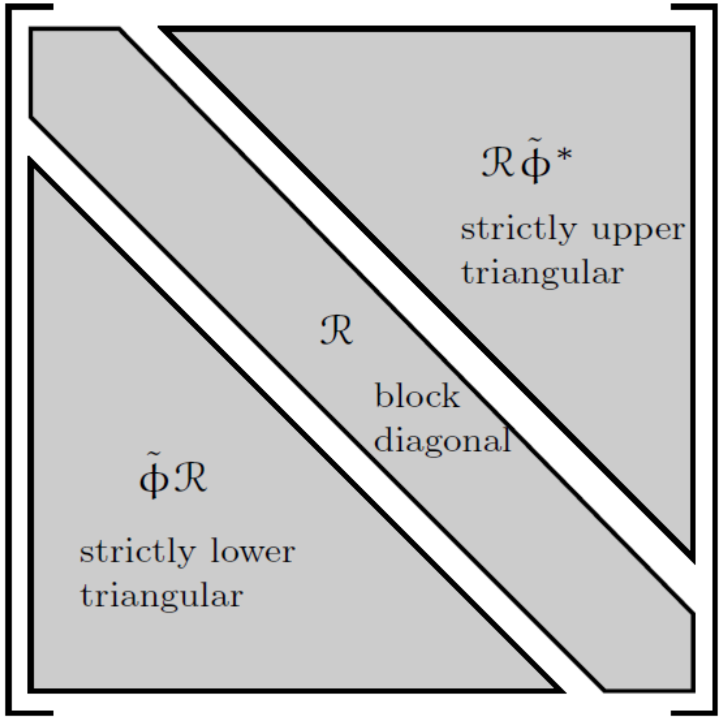
$$\boxed{\mathbf{M} = \mathcal{R} - \mathcal{E}_\phi \mathcal{R} \mathcal{E}_\phi^*}$$

Recall: Decomposition structure of $\phi\mathbf{M}\phi^*$

Previously: $$\phi\mathbf{M}\phi^* = \underset{\text{diagonal}}{\mathcal{R}} + \underset{\text{lower triangular}}{\tilde{\phi}\mathcal{R}} + \underset{\text{upper triangular}}{\mathcal{R}\tilde{\phi}^*}$$

The decomposition consists of 3 disjoint terms – a diagonal, and strictly upper/lower triangular parts

$$\phi\mathbf{M}\phi^* =$$


$\mathcal{R}\tilde{\phi}^*$
strictly upper triangular
$\mathcal{R}$
block diagonal
$\tilde{\phi}\mathcal{R}$
strictly lower triangular

Generalized Forward Lyapunov decomposition

With A & B being SPO operator, and X block diagonal and

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

$$\Downarrow$$

diagonal

$$\boxed{X = Y - \mathcal{E}_{\mathbb{A}} Y \mathcal{E}^*_{\mathbb{B}}} \qquad \boxed{Z = Y + \tilde{\mathbb{A}} Y + Y \tilde{\mathbb{B}}^*}$$

Like CRB

disjoint

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

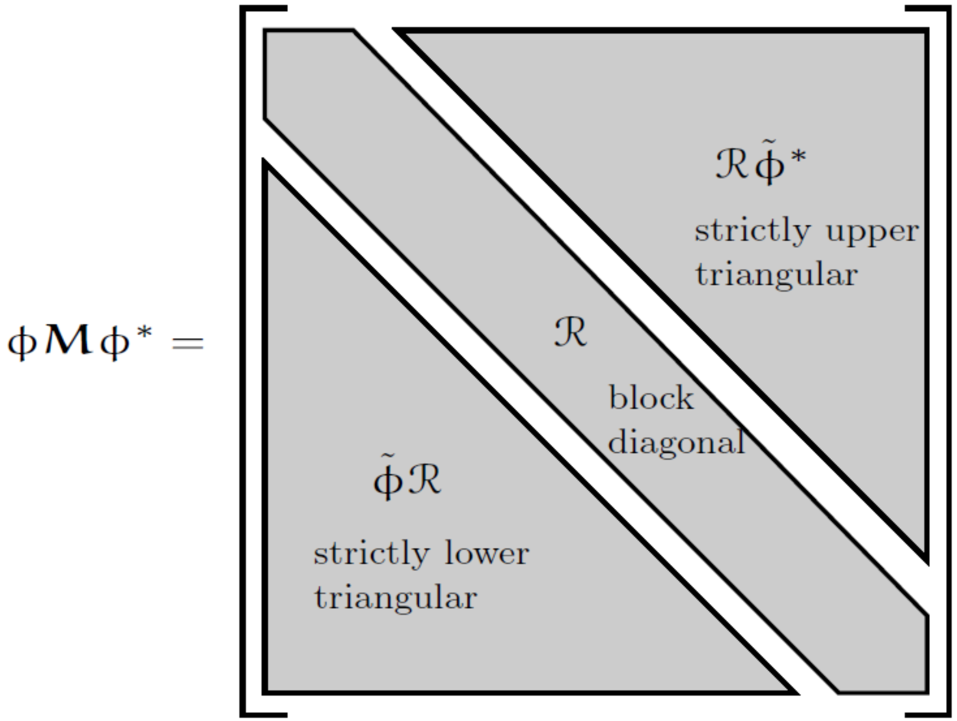
Generalized decomposition

diagonal

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

disjoint

Triangularity only for canonical trees



$\phi \mathbf{M} \phi^* =$
$\mathcal{R}\tilde{\phi}^*$
strictly upper triangular
$\mathcal{R}$
block diagonal
$\tilde{\phi}\mathcal{R}$
strictly lower triangular

While the decomposition terms are always disjoint, the triangular structure holds for only canonical trees

Generalized CRB gather recursion

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

O(N) gather recursion

```
for all nodes k (tips-to-base gather)
    $\mathcal{Y}(k) = \sum_{\forall j \in \mathcal{C}(k)} \mathbb{A}(k,j)\mathcal{Y}(j)\mathbb{B}^*(k,j) + \mathcal{X}(k)$
end loop
```

$$\mathcal{Z}(i,j) = \begin{cases} \mathcal{Y}(i) & \textit{for } i = j \\ \mathbb{A}(i,k)\mathcal{Z}(k,j) & \textit{for } i \succ k \succeq j, \quad k \in \mathcal{C}(i) \\ \mathcal{Z}(i,k)\mathbb{B}^*(j,k) & \textit{for } i \preceq k \prec j, \quad k \in \mathcal{C}(j) \\ \mathbf{0} & \textit{otherwise} \end{cases}$$

Derivation

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

$$\Downarrow$$

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

or

$$\mathrm{Z}(\mathrm{i},\mathrm{i}) = \mathrm{Y}(\mathrm{i})$$

$$\mathrm{Z}(\mathrm{i},\mathrm{j}) = \mathbb{A}(\mathrm{i},\mathrm{j})\ \mathbb{A}(\mathrm{j},\mathrm{p})\ \mathrm{X}(\mathrm{p})\ \mathbb{A}^*(\mathrm{j},\mathrm{p}) = \mathbb{A}(\mathrm{i},\mathrm{j})\ \mathrm{Y}(\mathrm{j}) \qquad \boxed{\mathrm{i} \succ \mathrm{j}}$$

Mass Matrix Computation for SKO Models

Recall: Decomposition of the mass matrix $\mathcal{M}(\theta)$

Can use the CRBs to develop a decomposition of the mass matrix into **disjoint** components

$$\boxed{\mathcal{M}(\theta) \triangleq H\phi\mathbf{M}\phi^*H^*}$$

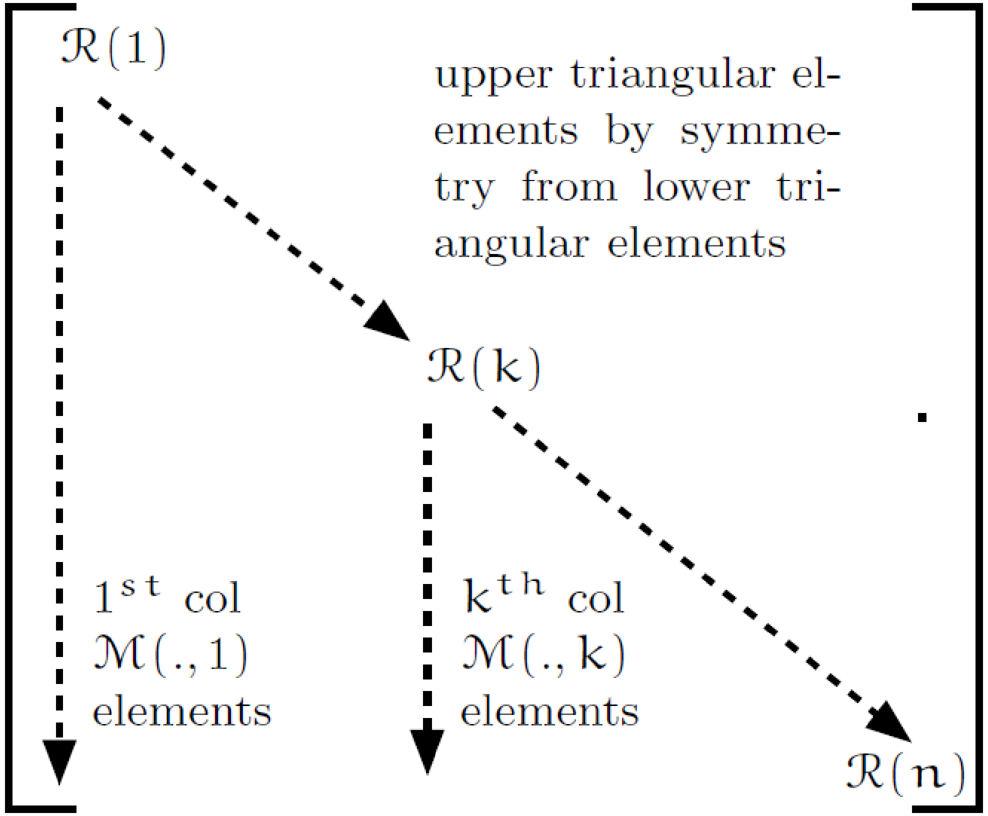
$$\boxed{\phi\mathbf{M}\phi^* = \mathcal{R} + \tilde{\phi}\mathcal{R} + \mathcal{R}\tilde{\phi}^*}$$

$$\boxed{\mathcal{M} = \underset{\text{diagonal}}{H\mathcal{R}H^*} + \underset{\text{strictly lower triangular}}{H\tilde{\phi}\mathcal{R}H^*} + \underset{\text{strictly upper triangular}}{H\mathcal{R}\tilde{\phi}^*H^*}}$$

Recall: Mass matrix computation algorithm structure

Compute diagonal, followed by off-diagonal elements

$$\boxed{\mathcal{M} = H\mathcal{R}H^* + H\tilde{\phi}\mathcal{R}H^* + H\mathcal{R}\tilde{\phi}^*H^*}$$



$\mathcal{R}(1)$

upper triangular elements by symmetry from lower triangular elements

$\mathcal{R}(k)$

1^{st} col $\mathcal{M}(.,1)$ elements

k^{th} col $\mathcal{M}(.,k)$ elements

$\mathcal{R}(n)$

Computation of the mass matrix is rarely needed

This is an early example of being able to directly map operator expressions into low-cost recursive algorithms

Generalized mass matrix decompositon

Using the forward Lyapunov decomposition

$$\boxed{\mathcal{M} = \mathrm{H}\mathbb{A}\mathbf{M}\mathbb{A}^*\mathrm{H}^*}$$

$$\boxed{\mathcal{M} = \mathrm{H}\mathcal{R}\mathrm{H}^* + \mathrm{H}\tilde{\mathbb{A}}\mathcal{R}\mathrm{H}^* + \mathrm{H}\mathcal{R}\tilde{\mathbb{A}}^*\mathrm{H}^*}$$

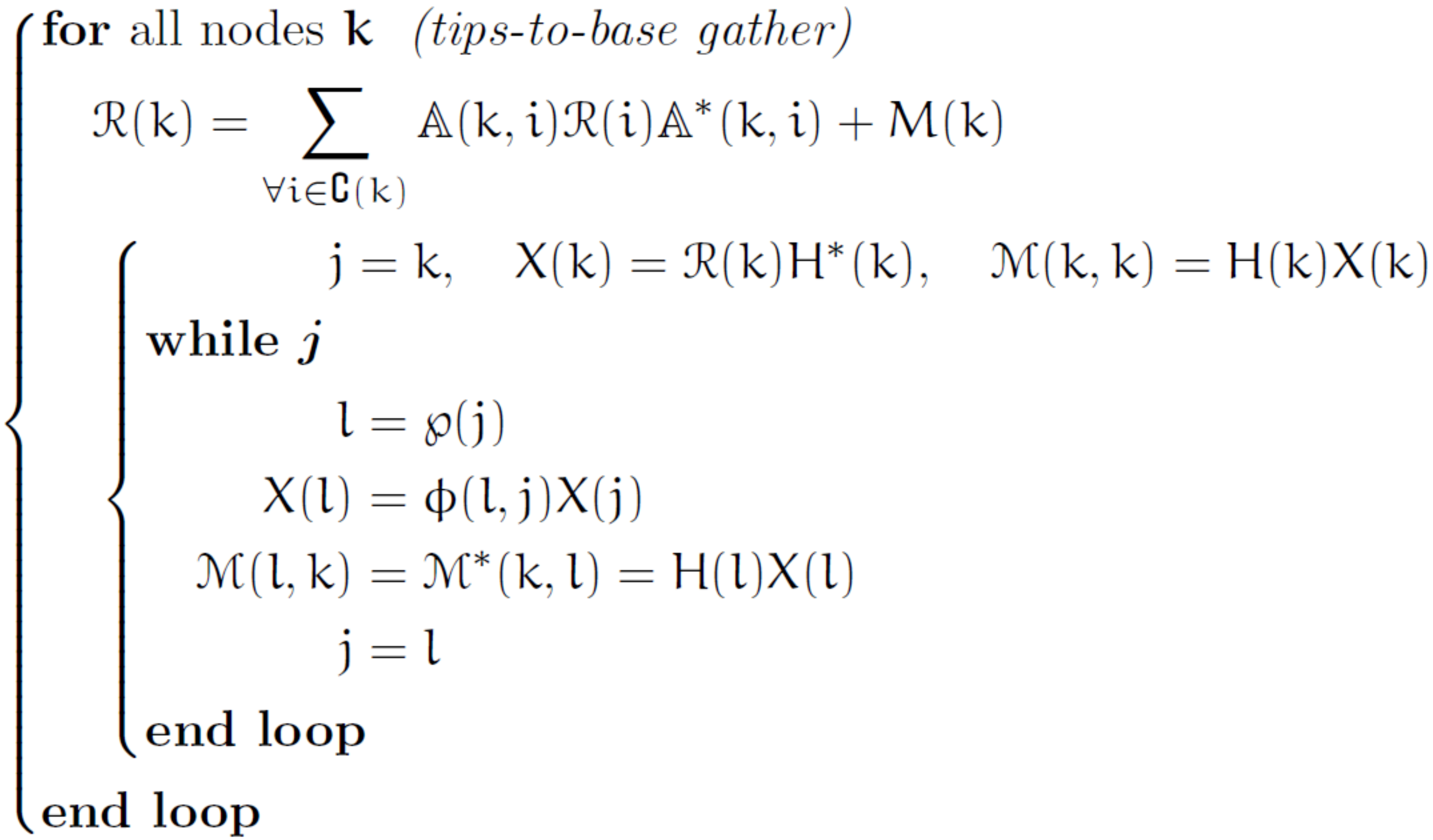
$$\boxed{\mathbf{M} = \mathcal{R} - \mathcal{E}_{\mathbb{A}}\mathcal{R}\mathcal{E}_{\mathbb{A}}^*}$$

Generalized O(N) CRB gather recursion

$$\boxed{\begin{cases} \textbf{for} \text{ all nodes } \mathbf{k} \quad \textit{(tips-to-base gather)} \\ \quad \mathcal{R}(\mathrm{k}) = \sum\limits_{\forall \mathrm{i} \in \mathbf{C}(\mathrm{k})} \mathbb{A}(\mathrm{k},\mathrm{i})\mathcal{R}(\mathrm{i})\mathbb{A}^*(\mathrm{k},\mathrm{i}) + \mathrm{M}(\mathrm{k}) \\ \textbf{end loop} \end{cases}}$$

Generalized mass matrix computation

Compute diagonal CRB elements in a gather recursion followed by the off-diagonal terms



```
for all nodes k   (tips-to-base gather)
    $\mathcal{R}(k) = \sum_{\forall i \in \mathfrak{C}(k)} \mathbb{A}(k,i)\mathcal{R}(i)\mathbb{A}^*(k,i) + M(k)$
    $j = k, \quad X(k) = \mathcal{R}(k)H^*(k), \quad \mathcal{M}(k,k) = H(k)X(k)$
    while j
        $l = \wp(j)$
        $X(l) = \phi(l,j)X(j)$
        $\mathcal{M}(l,k) = \mathcal{M}^*(k,l) = H(l)X(l)$
        $j = l$
    end loop
end loop
```

Generalized mass matrix sparsity

The mass matrix's sparsity mirrors that of the SPO matrix

$$\boxed{\mathcal{M} = \mathrm{H}\mathcal{R}\mathrm{H}^* + \mathrm{H}\tilde{\mathbb{A}}\mathcal{R}\mathrm{H}^* + \mathrm{H}\mathcal{R}\tilde{\mathbb{A}}^*\mathrm{H}^*}$$

Unrelated body pair terms are zero

$$\mathcal{M} = \begin{pmatrix} \mathrm{X} & \cdot & \mathrm{X} & \cdot & \cdot & \mathrm{X} & \mathrm{X} \\ \cdot & \mathrm{X} & \cdot & \mathrm{X} & \cdot & \mathrm{X} & \mathrm{X} \\ \mathrm{X} & \cdot & \mathrm{X} & \cdot & \cdot & \mathrm{X} & \mathrm{X} \\ \cdot & \mathrm{X} & \cdot & \mathrm{X} & \cdot & \mathrm{X} & \mathrm{X} \\ \cdot & \cdot & \cdot & \cdot & \mathrm{X} & \mathrm{X} & \mathrm{X} \\ \mathrm{X} & \mathrm{X} & \mathrm{X} & \mathrm{X} & \mathrm{X} & \mathrm{X} & \mathrm{X} \\ \mathrm{X} & \mathrm{X} & \mathrm{X} & \mathrm{X} & \mathrm{X} & \mathrm{X} & \mathrm{X} \end{pmatrix}$$

Recall: Trace of the serial-chain mass matrix

$$\boxed{\mathcal{M} = H\mathcal{R}H^* + H\tilde{\phi}\mathcal{R}H^* + H\mathcal{R}\tilde{\phi}^*H^*}$$

zero trace

General expression

$$\text{Trace}\{\mathcal{M}(\theta)\} = \sum_{i=1}^{n} \text{Trace}\{H(k)\mathcal{R}(k)H^*(k)\}$$

For 1 dof hinges

$$\text{Trace}\{H(k)\mathcal{R}(k)H^*(k)\} = H(k)\mathcal{R}(k)H^*(k)$$

Trace of the general mass matrix

$$\boxed{\mathcal{M} = \mathrm{H}\mathcal{R}\mathrm{H}^* + \mathrm{H}\tilde{\mathbb{A}}\mathcal{R}\mathrm{H}^* + \mathrm{H}\mathcal{R}\tilde{\mathbb{A}}^*\mathrm{H}^*}$$

zero trace

General expression

$$\mathrm{Trace}\{\mathcal{M}(\theta)\} = \sum_{i=1}^{n} \mathrm{Trace}\{\mathrm{H}(k)\mathcal{R}(k)\mathrm{H}^*(k)\}$$

Generalized Backward Lyapunov Equation for SKO Models

Generalized Forward Lyapunov decomposition

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

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

$$\Downarrow$$

$$\boxed{X = Y - \mathcal{E}_{\mathbb{A}} Y \mathcal{E}^*_{\mathbb{B}}} \qquad \boxed{Z = Y + \tilde{\mathbb{A}} Y + Y \tilde{\mathbb{B}}^*}$$

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

- This product is the dual to the $\boxed{\mathcal{Z} \triangleq \mathbb{A} \mathcal{X} \mathbb{B}^*}$ product used for understanding the mass matrix structure
- Why is this dual product important?
  - It shows up in products of the form $\mathrm{G_c} \mathcal{M}^{-1} \mathrm{G_c^*}$ 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

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

$$\Downarrow$$

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

block diagonal

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

disjoint decomposition

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

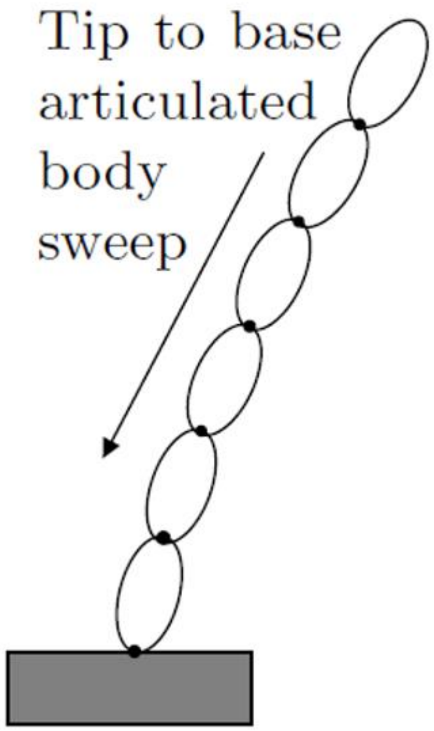
More in later sessions …

ATBI Riccati Equation for SKO Models

Recall: Articulated body inertias algorithm for serial-chain

O(N) tip-to-base gather algorithm for ATBI quantities for the Riccati equation

```
$\mathcal{P}^+(0) = \mathbf{0}, \quad \overline{\tau}(0) = \mathbf{0}$
for $k$   $1 \cdots n$
    $\psi(k, k-1) = \phi(k, k-1)\overline{\tau}(k-1)$
    $\mathcal{P}(k) = \phi(k, k-1)\mathcal{P}^+(k-1)\phi^*(k, k-1) + M(k)$
    $\mathcal{D}(k) = H(k)\mathcal{P}(k)H^*(k)$
    $\mathcal{G}(k) = \mathcal{P}(k)H^*(k)\mathcal{D}^{-1}(k)$
    $\mathcal{K}(k+1, k) = \phi(k+1, k)\mathcal{G}(k)$
    $\overline{\tau}(k) = \mathbf{I} - \mathcal{G}(k)H(k)$
    $\mathcal{P}^+(k) = \overline{\tau}(k)\mathcal{P}(k)$
end loop
```



Recall: ATBI spatial operators

Now define spatial operators using the ATBI quantities

$$
\begin{aligned}
\mathcal{P} &\triangleq \operatorname{diag}\left\{\mathcal{P}(k)\right\}_{k=1}^{n} && \in \mathcal{R}^{6n\times 6n} \\
\mathcal{D} &\triangleq \operatorname{diag}\left\{\mathcal{D}(k)\right\}_{k=1}^{n} = H\mathcal{P}H^{*} && \in \mathcal{R}^{\mathcal{N}\times\mathcal{N}} \\
\mathcal{G} &\triangleq \operatorname{diag}\left\{\mathcal{G}(k)\right\}_{k=1}^{n} = \mathcal{P}H^{*}\mathcal{D}^{-1} && \in \mathcal{R}^{6n\times\mathcal{N}} \\
\mathcal{K} &\triangleq \mathcal{E}_{\phi}\mathcal{G} && \in \mathcal{R}^{6n\times\mathcal{N}} \\
\tau &\triangleq \operatorname{diag}\left\{\tau(k)\right\}_{k=1}^{n} = \mathcal{G}H && \in \mathcal{R}^{6n\times 6n} \\
\overline{\tau} &\triangleq \operatorname{diag}\left\{\overline{\tau}(k)\right\}_{k=1}^{n} = \mathbf{I}-\tau && \in \mathcal{R}^{6n\times 6n} \\
\mathcal{P}^{+} &\triangleq \operatorname{diag}\left\{\mathcal{P}^{+}(k)\right\}_{k=1}^{n} = \overline{\tau}\mathcal{P}\overline{\tau}^{*} = \overline{\tau}\mathcal{P} = \mathcal{P}\overline{\tau}^{*} && \in \mathcal{R}^{6n\times 6n} \\
\mathcal{E}_{\psi} &\triangleq \mathcal{E}_{\phi}\overline{\tau} && \in \mathcal{R}^{6n\times 6n}
\end{aligned}
$$

Recall: Structure of $\mathcal{E}_{\psi}$

$\mathcal{E}_{\psi}$ has the same structure as $\mathcal{E}_{\phi}$

$$\boxed{\mathcal{E}_{\psi} \triangleq \mathcal{E}_{\phi}\overline{\tau}} \qquad \boxed{\psi(k+1,k) \triangleq \phi(k+1,k)\overline{\tau}(k)}$$

block diagonal

$$\mathcal{E}_{\psi} = \begin{pmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \psi(2,1) & \mathbf{0} & \dots & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \psi(3,2) & \dots & \mathbf{0} & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \dots & \psi(n,n-1) & \mathbf{0} \end{pmatrix}$$

$\mathcal{E}_{\psi}$ has same structure as $\mathcal{E}_{\phi}$ and is **nilpotent**

Recall: ψ is the 1-resolvent of $\mathcal{E}_\psi$

Analogous to

$$\boxed{\phi \overset{\triangle}{=} (\mathbf{I} - \mathcal{E}_\phi)^{-1} = \mathbf{I} + \mathcal{E}_\phi + \mathcal{E}_\phi^2 + \cdots + \mathcal{E}_\phi^{n-1}}$$

$$\phi(i,j) = \phi(i,i-1) \cdots \phi(j+1,j)$$

define

$$\boxed{\begin{aligned}\psi &\overset{\triangle}{=} (\mathbf{I} - \mathcal{E}_\psi)^{-1} = \mathbf{I} + \mathcal{E}_\psi + \mathcal{E}_\psi^2 + \cdots + \mathcal{E}_\psi^{n-1} \\ &= \begin{pmatrix} \mathbf{I} & \mathbf{0} & \dots & \mathbf{0} \\ \psi(2,1) & \mathbf{I} & \dots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \psi(n,1) & \psi(n,2) & \dots & \mathbf{I} \end{pmatrix}\end{aligned}}$$

$$\psi(i,j) \overset{\triangle}{=} \psi(i,i-1) \cdots \psi(j+1,j) \quad \text{for} \quad i > j$$

Recall: Operator level Riccati equation for serial–chain ATBI

Similar to Lyapunov equation

$$\boxed{\mathbf{M} = \Re - \mathcal{E}_{\phi} \mathcal{R} \mathcal{E}_{\phi}^{*}}$$

The ATBI recursion

$$\mathcal{P}(k+1) \overset{6.32}{=} \psi(k+1,k)\mathcal{P}(k)\psi^{*}(k+1,k) + \mathbf{M}(k+1)$$

can be re-expressed at the operator level as

$$\boxed{\mathbf{M} = \mathcal{P} - \mathcal{E}_{\psi} \mathcal{P} \mathcal{E}_{\psi}^{*} = \mathcal{P} - \mathcal{E}_{\psi} \mathcal{P} \mathcal{E}_{\phi}^{*}}$$

Generalized ATBI Riccati Equation algorithm

Gather algorithm for SKO model ATBI

$$
\left\{
\begin{array}{l}
\textbf{for } \textit{all nodes } \mathbf{k} \quad \textit{(tips-to-base gather)} \\
\qquad \mathcal{P}(k) = \boxed{\sum_{\forall j \in \mathbf{C}(k)} \mathbb{A}(k,j)\mathcal{P}^{+}(j)\mathbb{A}^{*}(k,j)} + \mathbf{M}(k) \\
\qquad \mathcal{D}(k) = H(k)\mathcal{P}(k)H^{*}(k) \\
\qquad \mathcal{G}(k) = \mathcal{P}(k)H^{*}(k)\mathcal{D}^{-1}(k) \\
\qquad \tau(k) = \mathcal{G}(k)H(k) \\
\qquad \overline{\tau}(k) = \mathbf{I} - \tau(k) \\
\qquad \mathcal{P}^{+}(k) = \overline{\tau}(k)\mathcal{P}(k) \\
\textbf{end loop}
\end{array}
\right.
$$

Generalized ATBI operators

$$\boxed{\psi(\wp(k),k) \triangleq \mathbb{A}(\wp(k),k)\bar{\tau}(k).}$$

$$\boxed{\begin{aligned} \mathcal{D} &\triangleq \mathrm{H}\mathcal{P}\mathrm{H}^*, & \mathcal{G} &\triangleq \mathcal{P}\mathrm{H}^*\mathcal{D}^{-1}, & \tau &\triangleq \mathcal{G}\mathrm{H} \\ \bar{\tau} &\triangleq \mathbf{I}-\tau, & \mathcal{P}^+ &\triangleq \bar{\tau}\mathcal{P} = \bar{\tau}\mathcal{P}\bar{\tau}^* \end{aligned}}$$

$$\boxed{\mathcal{E}_\psi \triangleq \mathcal{E}_{\mathbb{A}}\bar{\tau}} \Rightarrow \boxed{\psi \triangleq (\mathbf{I}-\mathcal{E}_\psi)^{-1}}$$

$$\boxed{\bar{\tau}\mathcal{P} = \mathcal{P}\bar{\tau}^* = \bar{\tau}\mathcal{P}\bar{\tau}^* = \mathcal{P}^+}$$

Generalized ATBI Riccati operator equation

$$\boxed{\mathbf{M} = \mathcal{P} - \mathcal{E}_{\psi}\mathcal{P}\mathcal{E}_{\psi}^{*} = \mathcal{P} - \mathcal{E}_{\mathbb{A}}\mathcal{P}\mathcal{E}_{\psi}^{*}}$$

$$\Downarrow$$

$$\mathbf{M} = \mathcal{P} - \mathcal{E}_{\mathbb{A}}\Big[\underbrace{\mathcal{P} - \underbrace{\underbrace{\mathcal{P}\mathrm{H}^{*}(\underbrace{\mathrm{H}\mathcal{P}\mathrm{H}^{*}}_{\mathcal{D}})^{-1}}_{\mathcal{G}}\,\mathrm{H}}_{\tau}\,\mathcal{P}}_{\mathcal{P}+}\Big]\mathcal{E}_{\mathbb{A}}^{*}$$

Operator Identities for SKO Models

Recall: $\mathcal{K}$ spatial operator for serial-chains

$\mathcal{K}$ has the same structure as $\mathcal{E}_\phi$

$$\boxed{\mathcal{K} \triangleq \mathcal{E}_\phi \mathcal{G}} \qquad \boxed{\mathcal{K}(k+1,k) = \phi(k+1,k)\mathcal{G}(k)}$$

block diagonal

$$\mathcal{K} = \begin{pmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathcal{K}(2,1) & \mathbf{0} & \dots & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathcal{K}(3,2) & \dots & \mathbf{0} & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \dots & \mathcal{K}(n,n-1) & \mathbf{0} \end{pmatrix}$$

The only non-zero entries are along the first sub-diagonal

Generalizaton of $\mathcal{K}$ to SKO Models

Lower-triangular only for canonical trees

$$\boxed{\mathcal{K} = \mathcal{E}_{\mathbb{A}}\mathcal{G}}$$

These identities continue to hold:

$$\boxed{\begin{array}{c} \mathrm{H}\mathcal{G} = \mathbf{I}, \qquad \mathrm{H}\tau = \mathrm{H}, \qquad \mathrm{H}\overline{\tau} = \mathbf{0} \\ \mathbf{I} + \mathrm{H}\mathbb{A}\mathcal{K} = \mathrm{H}\mathbb{A}\mathcal{G} \end{array}}$$

Identities generalization

Recall serial-chain decomposition:

$$\boxed{\phi \mathbf{M} \psi^* = \mathcal{P} + \tilde{\phi}\mathcal{P} + \mathcal{P}\tilde{\psi}^*}$$

block-diagonal

strictly lower triangular

strictly upper triangular

SKO model generalization (disjoint decomposition):

$$\boxed{\mathbb{A} \mathbf{M} \psi^* = \mathcal{P} + \tilde{\mathbb{A}}\mathcal{P} + \mathcal{P}\tilde{\psi}^*}$$

block-diagonal

Expression for $\mathrm{H}\psi\mathbf{M}\psi^*\mathrm{H}^*$

Identity continues to hold for SKO models

$$\boxed{\mathrm{H}\psi\mathbf{M}\psi^*\mathrm{H}^* = \mathcal{D}}$$

So does this identity

$$\boxed{\psi\mathbf{M}\psi^* = \mathcal{P} + \tilde{\psi}\mathcal{P} + \mathcal{P}\tilde{\psi}^*}$$

Identities generalization (contd)

Recall serial-chain decomposition:

using ATBIs

$$\boxed{\phi \mathbf{M} \phi^* = \mathcal{P} + \tilde{\phi}\mathcal{P} + \mathcal{P}\tilde{\phi}^* + \phi \mathcal{K}\mathcal{D}\mathcal{K}^* \phi^*}$$

block-diagonal

strictly lower triangular

strictly upper triangular

dense

SKO model generalization (disjoint decomposition):

$$\boxed{\mathbb{A} \mathbf{M} \mathbb{A}^* = \mathcal{P} + \tilde{\mathbb{A}}\mathcal{P} + \mathcal{P}\tilde{\mathbb{A}}^* + \mathbb{A} \mathcal{K}\mathcal{D}\mathcal{K}^* \mathbb{A}^*}$$

Recall: Serial-chain Identities

$$\boxed{\psi^{-1} - \phi^{-1} = \mathcal{K}\mathrm{H}}$$

$$\boxed{\begin{aligned}
\psi^{-1}\phi &= \mathbf{I} + \mathcal{K}\mathrm{H}\phi \\
\phi\psi^{-1} &= \mathbf{I} + \phi\mathcal{K}\mathrm{H} \\
\phi^{-1}\psi &= \mathbf{I} - \mathcal{K}\mathrm{H}\psi \\
\psi\phi^{-1} &= \mathbf{I} - \psi\mathcal{K}\mathrm{H}
\end{aligned}}$$

$$\boxed{\begin{aligned}
[\mathbf{I} - \mathrm{H}\psi\mathcal{K}]\mathrm{H}\phi &= \mathrm{H}\psi \\
\phi\mathcal{K}[\mathbf{I} - \mathrm{H}\psi\mathcal{K}] &= \psi\mathcal{K} \\
[\mathbf{I} + \mathrm{H}\phi\mathcal{K}]\mathrm{H}\psi &= \mathrm{H}\phi \\
\psi\mathcal{K}[\mathbf{I} + \mathrm{H}\phi\mathcal{K}] &= \phi\mathcal{K}
\end{aligned}}$$

These identities are very useful in transforming and simplifying operator expressions. We will see their use in a number of instances ahead.

Identities generalization (contd)

For any SKO model

$$\boxed{\psi^{-1} - \mathbb{A}^{-1} = \mathcal{K}\mathrm{H}}$$

$$\boxed{\begin{aligned}
\psi^{-1}\mathbb{A} &= \mathbf{I} + \mathcal{K}\mathrm{H}\mathbb{A} \\
\mathbb{A}\psi^{-1} &= \mathbf{I} + \mathbb{A}\mathcal{K}\mathrm{H} \\
\mathbb{A}^{-1}\psi &= \mathbf{I} - \mathcal{K}\mathrm{H}\psi \\
\psi\mathbb{A}^{-1} &= \mathbf{I} - \psi\mathcal{K}\mathrm{H}
\end{aligned}}$$

$$\boxed{\begin{aligned}
[\mathbf{I} - \mathrm{H}\psi\mathcal{K}]\mathrm{H}\mathbb{A} &= \mathrm{H}\psi \\
\mathbb{A}\mathcal{K}[\mathbf{I} - \mathrm{H}\psi\mathcal{K}] &= \psi\mathcal{K} \\
[\mathbf{I} + \mathrm{H}\mathbb{A}\mathcal{K}]\mathrm{H}\psi &= \mathrm{H}\mathbb{A} \\
\psi\mathcal{K}[\mathbf{I} + \mathrm{H}\mathbb{A}\mathcal{K}] &= \mathbb{A}\mathcal{K}
\end{aligned}}$$

Mass Matrix Factorization and Inversion for SKO models

Recall: Inverting the mass matrix $\mathcal{M}(\theta)$ for serial chains

$$\boxed{\mathcal{M} = \left[\mathbf{I} + \mathrm{H}\phi\mathcal{K}\right]\mathcal{D}\left[\mathbf{I} + \mathrm{H}\phi\mathcal{K}\right]^*}$$

Lower triangular with identity along block-diagonal

Block-diagonal

Upper triangular with identity along block-diagonal

- All the factors are square in the Innovations Factorization
- So want to look into inverting the mass matrix by inverting its factors
- D is block-diagonal, and easy to invert
- We will thus focus on inverting $\left[\mathbf{I} + \mathrm{H}\phi\mathcal{K}\right]$

Recall: Inverse of $[\mathbf{I} + \mathrm{H}\phi\mathcal{K}]$

Claim:

$$\boxed{[\mathbf{I} + \mathrm{H}\phi\mathcal{K}]^{-1} = [\mathbf{I} - \mathrm{H}\psi\mathcal{K}]}$$

Lower triangular with identity along block-diagonal

Derivation:

Have general identity

$$(\mathbf{I} + \mathrm{A}\mathrm{B})^{-1} = \mathbf{I} - \mathrm{A}(\mathbf{I} + \mathrm{B}\mathrm{A})^{-1}\mathrm{B}$$

$$\begin{aligned}[\mathbf{I} + \mathrm{H}\phi\mathcal{K}]^{-1} &= \mathbf{I} - \mathrm{H}[\mathbf{I} + \phi\mathcal{K}\mathrm{H}]^{-1}\phi\mathcal{K} \overset{7.11}{=} \mathbf{I} - \mathrm{H}(\phi\psi^{-1})^{-1}\phi\mathcal{K} \\ &= \mathbf{I} - \mathrm{H}\psi\mathcal{K}\end{aligned}$$

using $\boxed{\phi\psi^{-1} = \mathbf{I} + \phi\mathcal{K}\mathrm{H}}$

Generalized Innovations Factorization for SKO models

$$\boxed{\mathcal{M} = \mathrm{H}\mathbb{A}\mathbf{M}\mathbb{A}^*\mathrm{H}^*}$$

$$\Downarrow$$

$$\boxed{\mathcal{M} = [\mathbf{I} + \mathrm{H}\mathbb{A}\mathcal{K}]\,\mathcal{D}\,[\mathbf{I} + \mathrm{H}\mathbb{A}\mathcal{K}]^*}$$

identity along block-diagonal

Block-diagonal

identity along block-diagonal

Triangular structure only for canonical trees

Innovations factor inversion for SKO models

$$\boxed{[\mathbf{I} + \mathrm{H}\phi\mathcal{K}]^{-1} = [\mathbf{I} - \mathrm{H}\psi\mathcal{K}]}$$

$$\Downarrow$$

$$\boxed{[\mathbf{I} + \mathrm{H}\mathbb{A}\mathcal{K}]^{-1} = [\mathbf{I} - \mathrm{H}\psi\mathcal{K}]}$$

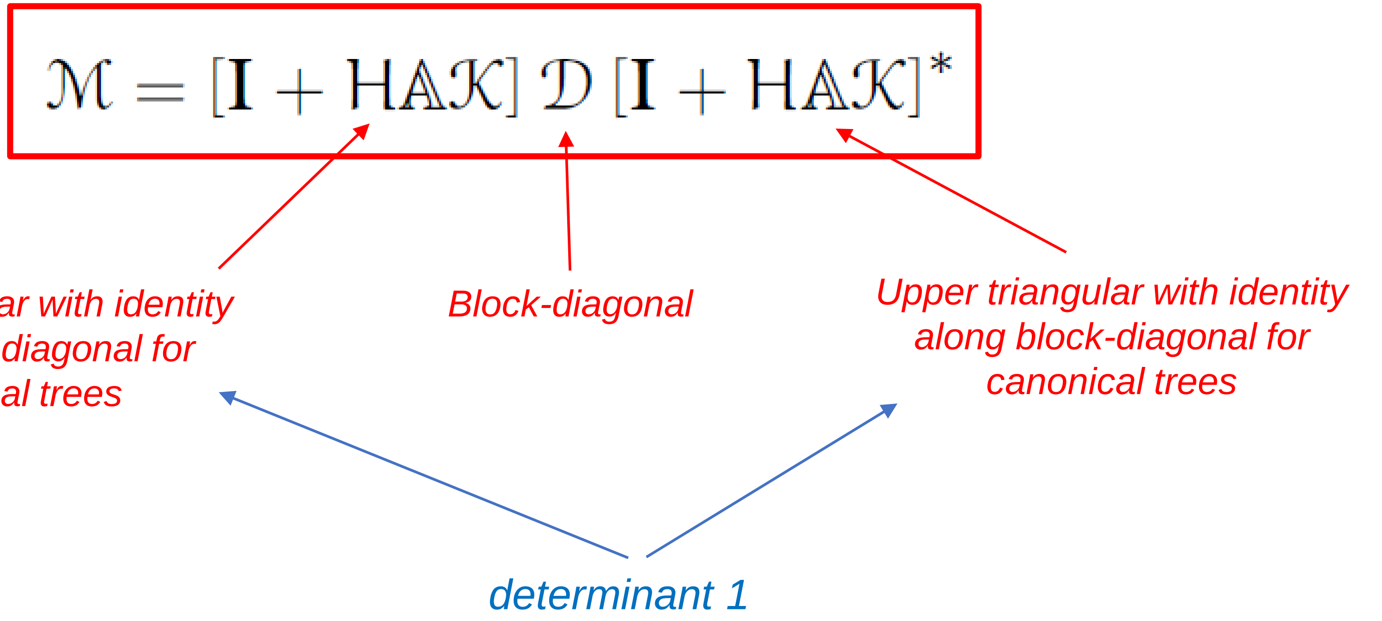
identity along block-diagonal
(lower triangular for canonical trees)

Mass matrix inverse for SKO models

The mass matrix inverse expression continues to hold unchanged for SKO models.

$$\boxed{\mathcal{M}^{-1} = [\mathbf{I} - \mathrm{H}\psi\mathcal{K}]^{*}\mathcal{D}^{-1}[\mathbf{I} - \mathrm{H}\psi\mathcal{K}]}$$

Determinant of the SKO model mass matrix



$$\boxed{\mathcal{M} = [\mathbf{I} + \mathrm{H}\mathbb{A}\mathcal{K}]\, \mathcal{D}\, [\mathbf{I} + \mathrm{H}\mathbb{A}\mathcal{K}]^*}$$

Lower triangular with identity along block-diagonal for canonical trees

Block-diagonal

Upper triangular with identity along block-diagonal for canonical trees

determinant 1

A matrix determinant is the product of the determinant of its square factors

General expression

$$\boxed{\det\{\mathcal{M}\} = \prod_{k=1}^{n} \det\{\mathcal{D}(k)\}}$$

scalar for 1 dof hinge

Recursive Forward Dynamics for SKO Models

Recall: Decomposing the $\ddot{\theta}$ expression for serial-chains

Breaking down the expression:

$$\boxed{\ddot{\theta} = [\mathbf{I} - H\psi\mathcal{K}]^* \underbrace{\mathcal{D}^{-1}\underbrace{[\mathcal{T} - H\,\underbrace{\psi(\mathcal{K}\mathcal{T} + \mathcal{P}\mathfrak{a} + \mathfrak{b})}_{\mathfrak{z}}]}_{\epsilon}}_{\nu} - \mathcal{K}^*\psi^*\mathfrak{a}}$$

$$\mathfrak{z} \triangleq \psi(\mathcal{K}\mathcal{T} + \mathcal{P}\mathfrak{a} + \mathfrak{b})$$

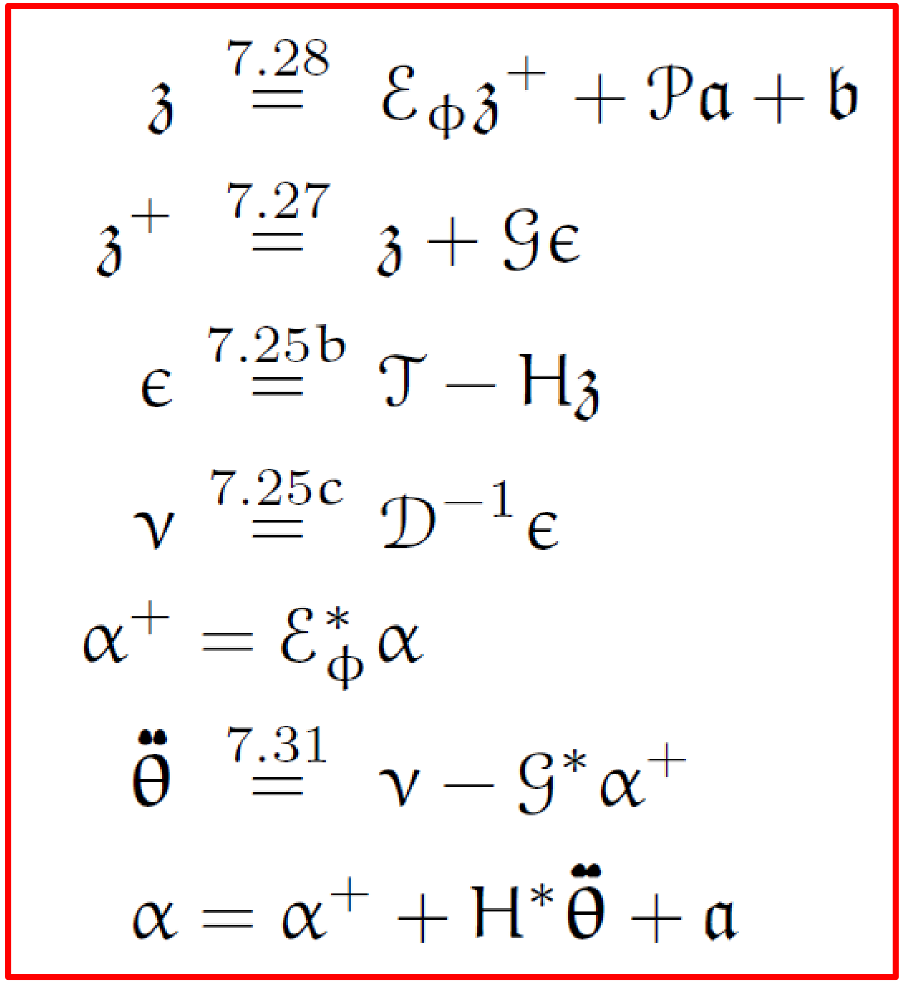
$$\epsilon \triangleq \mathcal{T} - H\mathfrak{z} \overset{7.25a}{=} \mathcal{T} - H\psi(\mathcal{K}\mathcal{T} + \mathcal{P}\mathfrak{a} + \mathfrak{b})$$

$$\nu \triangleq \mathcal{D}^{-1}\epsilon \overset{7.25b}{=} \mathcal{D}^{-1}[\mathcal{T} - H\psi(\mathcal{K}\mathcal{T} + \mathcal{P}\mathfrak{a} + \mathfrak{b})]$$

$$\boxed{\ddot{\theta} \overset{7.21,7.25c}{=} [\mathbf{I} - H\psi\mathcal{K}]^* \nu - \mathcal{K}^*\psi^*\mathfrak{a}}$$

Recall: Overall decomposed expressions

Putting it all together



$$
\begin{aligned}
\mathfrak{z} &\overset{7.28}{=} \mathcal{E}_{\phi}\mathfrak{z}^{+} + \mathcal{P}\mathfrak{a} + \mathfrak{b} \\
\mathfrak{z}^{+} &\overset{7.27}{=} \mathfrak{z} + \mathcal{G}\epsilon \\
\epsilon &\overset{7.25b}{=} \mathcal{T} - H\mathfrak{z} \\
\nu &\overset{7.25c}{=} \mathcal{D}^{-1}\epsilon \\
\alpha^{+} &= \mathcal{E}_{\phi}^{*}\alpha \\
\ddot{\theta} &\overset{7.31}{=} \nu - \mathcal{G}^{*}\alpha^{+} \\
\alpha &= \alpha^{+} + H^{*}\ddot{\theta} + \mathfrak{a}
\end{aligned}
$$

Recall: O(N) ATBI forward dynamics algorithm for serial-chains

ATBI recursion from before

gather sweep

```
$\mathcal{P}^+(0) = \mathbf{0}, \quad \mathfrak{z}^+(0) = \mathbf{0}, \quad \mathcal{T}(0) = \mathbf{0}, \quad \overline{\tau}(0) = \mathbf{0}$
for $k$   $1 \cdots n$
    $\mathcal{P}(k) = \phi(k, k-1)\mathcal{P}^+(k-1)\phi^*(k, k-1) + M(k)$
    $\mathcal{D}(k) = H(k)\mathcal{P}(k)H^*(k)$
    $\mathcal{G}(k) = \mathcal{P}(k)H^*(k)\mathcal{D}^{-1}(k)$
    $\overline{\tau}(k) = \mathbf{I} - \mathcal{G}(k)H(k)$
    $\mathcal{P}^+(k) = \overline{\tau}(k)\mathcal{P}(k)$
    $\mathfrak{z}(k) = \phi(k, k-1)\mathfrak{z}^+(k-1) + \mathcal{P}(k)\mathfrak{a}(k) + \mathfrak{b}(k)$
    $\epsilon(k) = \mathcal{T}(k) - H(k)\mathfrak{z}(k)$
    $\nu(k) = \mathcal{D}^{-1}(k)\epsilon(k)$
    $\mathfrak{z}^+(k) = \mathfrak{z}(k) + \mathcal{G}(k)\epsilon(k)$
end loop
```

O(N) computational complexity, fastest available algorithm

scatter sweep

```
$\alpha(n+1) = \mathbf{0}$
for $k$   $n \cdots 1$
    $\alpha^+(k) = \phi^*(k+1, k)\alpha(k+1)$
    $\boxed{\ddot{\theta}(k)} = \nu(k) - \mathcal{G}^*(k)\alpha^+(k)$
    $\alpha(k) = \alpha^+(k) + H^*(k)\ddot{\theta}(k) + \mathfrak{a}(k)$
end loop
```

Generalized accels expression for SKO models

Same expression as for serial-chains

$$\ddot{\theta} = \mathcal{M}^{-1}(\mathcal{T} - \mathcal{C}).$$

$$= \mathcal{M}^{-1}(\mathcal{T} - \mathcal{C}) = [\mathbf{I} - \mathrm{H}\psi\mathcal{K}]^{*}\,\mathcal{D}^{-1}\left[\mathcal{T} - \mathrm{H}\psi(\mathcal{K}\mathcal{T} + \mathcal{P}\mathfrak{a} + \mathfrak{b})\right] - \mathcal{K}^{*}\psi^{*}\mathfrak{a}$$

SKO model ATBI quantities

$$\ddot{\theta} = [\mathbf{I} - H\psi\mathcal{K}]^* \underbrace{\mathcal{D}^{-1}[\underbrace{\mathcal{T} - H\,\underbrace{\psi(\mathcal{K}\mathcal{T} + \mathcal{P}\mathfrak{a} + \mathfrak{b})}_{\mathfrak{z}}]}_{\epsilon}}_{\nu} - \mathcal{K}^*\psi^*\mathfrak{a}$$

↓

$$\begin{aligned}
\mathfrak{z} &= \mathcal{E}_{\mathbb{A}}\mathfrak{z}^+ + \mathcal{P}\mathfrak{a} + \mathfrak{b} \\
\mathfrak{z}^+ &= \mathfrak{z} + \mathcal{G}\epsilon \\
\epsilon &= \mathcal{T} - H\mathfrak{z} \\
\nu &= \mathcal{D}^{-1}\epsilon \\
\alpha^+ &= \mathcal{E}_{\mathbb{A}}^*\alpha \\
\ddot{\theta} &= \nu - \mathcal{G}^*\alpha^+ \\
\alpha &= \alpha^+ + H^*\ddot{\theta} + \mathfrak{a}
\end{aligned}$$

SKO model O(N) ATBI Forward Dynamics

```
for all nodes k   (tips-to-base gather)
    $\mathfrak{z}(k) = \sum_{\forall j \in \mathfrak{C}(k)} \mathbb{A}(k, j)\mathfrak{z}^{+}(j) + \mathcal{P}(k)\mathfrak{a}(k) + \mathfrak{b}(k)$
    $\epsilon(k) = \mathcal{T}(k) - H(k)\mathfrak{z}(k)$
    $\nu(k) = \mathcal{D}^{-1}(k)\epsilon(k)$
    $\mathfrak{z}^{+}(k) = \mathfrak{z}(k) + \mathcal{G}(k)\epsilon(k)$
end loop
```

gather sweep

```
for all nodes k   (base-to-tips scatter)
    $\alpha^{+}(k) = \mathbb{A}^{*}(\wp(k), k)\alpha(\wp(k))$
    $\ddot{\theta}(k) = \nu(k) - \mathcal{G}^{*}(k)\alpha^{+}(k)$
    $\alpha(k) = \alpha^{+}(k) + H^{*}(k)\ddot{\theta}(k) + \mathfrak{a}(k)$
end loop
```

scatter sweep

SKO model process for multibody systems

Structure: Insight and Implications

increasing analytical structure

Absolute coordinates	Non-minimal, explicit constraints	DAE rabbit hole, sparsity structure
Lagrangian approach	$\mathcal{M}(\theta)$	Opaque, non-singular matrix
Kane's approach	$\mathcal{M}(\theta) = C^* \ \mathbf{M} \ C$	C non-square partial velocity matrix
NE factorization	$\mathcal{M}(\theta) = H\phi\mathbf{M}\boxed{\phi^* H^*}$	ϕ is a square matrix
Connectivity structure	$\phi \triangleq (\mathbf{I} - \mathcal{E}_\phi)^{-1}$	ϕ is a 1-resolvent
SKO models	$\mathcal{E}_\phi$	$\boxed{\mathcal{E}_\phi \text{ is a BWA matrix}}$

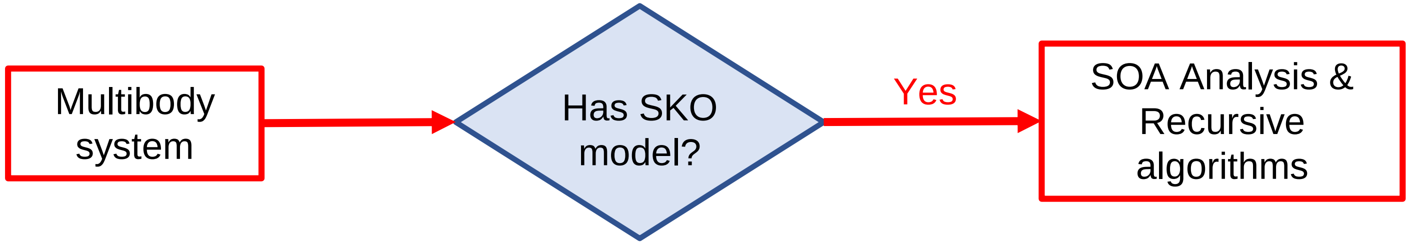
Opens the gates to SOA analysis

Diverse, but similar SOA solutions

What are the common patterns and properties that make the SOA process work across such a broad family of system types?

- The answer lies in the SKO model structure
- Recognizing SKO model structure allows us to avoid daunting, tedious and repetitious formulation, analysis, and algorithm development process over and over for the different systems and their combinations
- Given the common patterns that were clear in the SOA based mass matrix factorization inversion and algorithm development across the different types we have the question:

Diverse, but similar SOA solutions

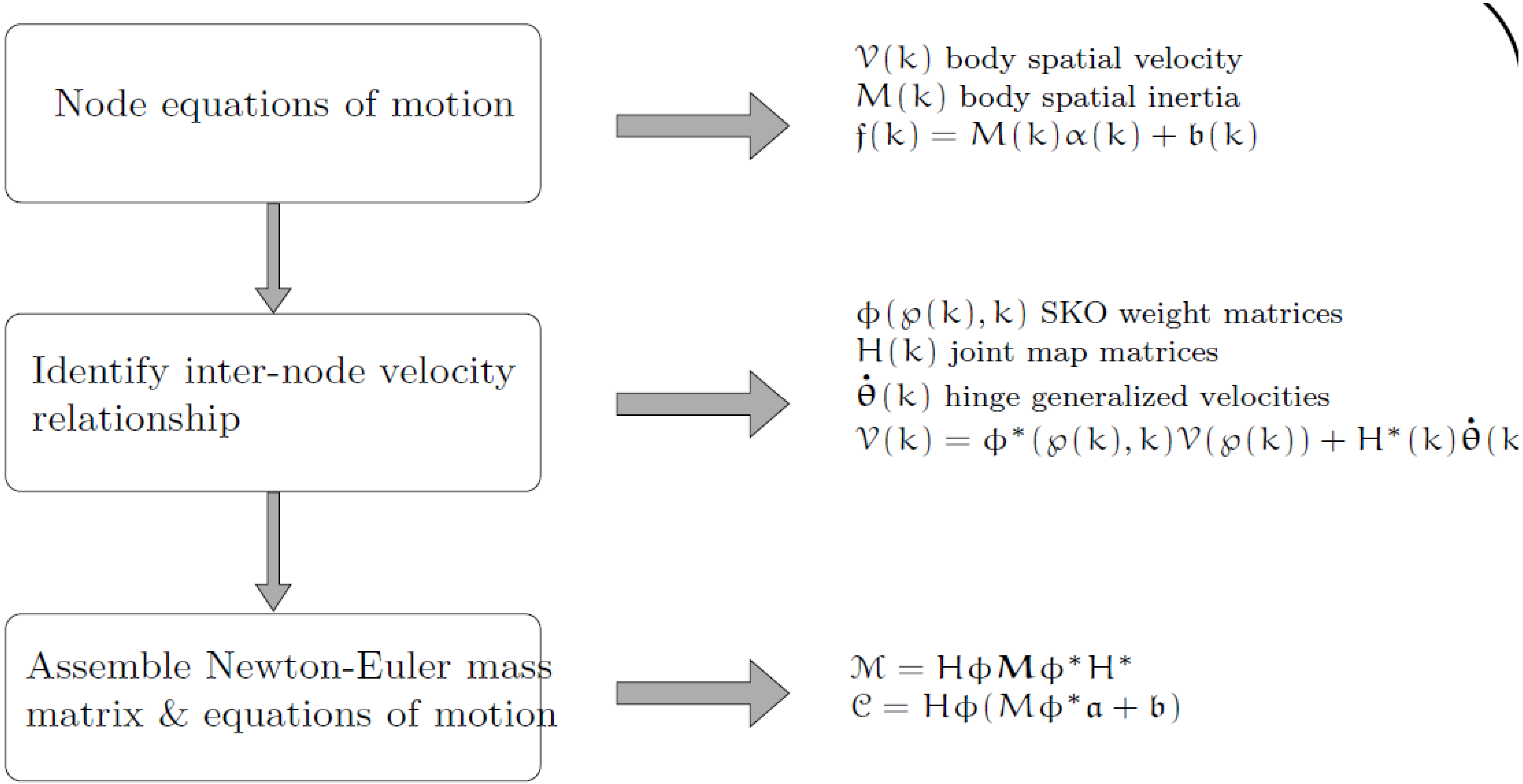


SKO models are available for a broad class of multibody systems

- Rigid/flex bodies
- Regular and flex joints
- Geared joints
- Branching and loops
- Prescribed and non-prescribed motion

SKO model development for a multibody system

First identify a tree-topology structure



Node equations of motion

$\mathcal{V}(k)$ body spatial velocity
$M(k)$ body spatial inertia
$\mathfrak{f}(k) = M(k)\alpha(k) + \mathfrak{b}(k)$

Identify inter-node velocity relationship

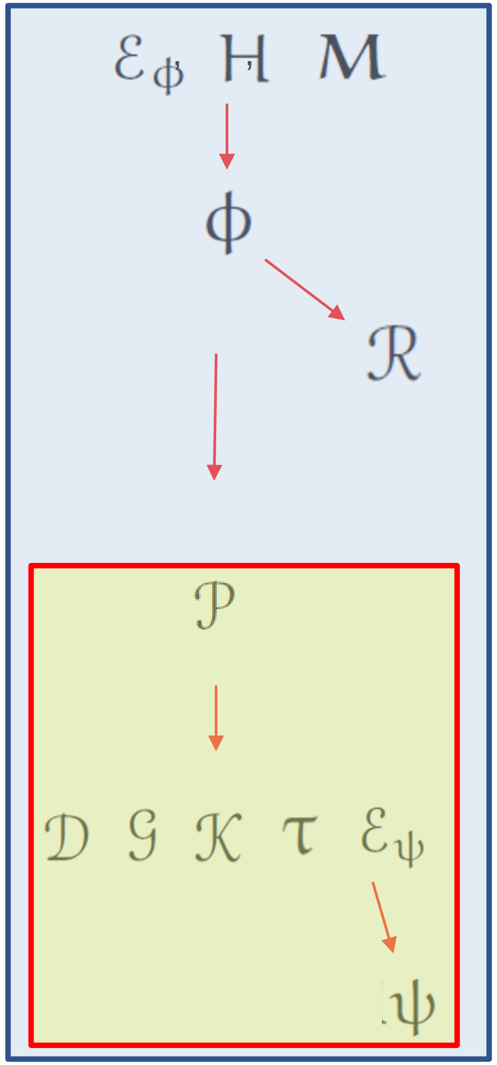
$\phi(\wp(k), k)$ SKO weight matrices
$H(k)$ joint map matrices
$\dot{\theta}(k)$ hinge generalized velocities
$\mathcal{V}(k) = \phi^*(\wp(k), k)\mathcal{V}(\wp(k)) + H^*(k)\dot{\theta}(k$

Assemble Newton-Euler mass matrix & equations of motion

$\mathcal{M} = H\phi\mathbf{M}\phi^*H^*$
$\mathcal{C} = H\phi(M\phi^*\mathfrak{a} + \mathfrak{b})$

General SKO models - analysis

- Velocity expression
- Jacobian
- Mass matrix NE factorization
- Lyapunov equation for CRBs
- Mass matrix decomposition
- Riccati equation for ATBI
- Several operator identities
- Mass matrix Innovations factorization
- Mass matrix determinant
- Mass matrix inverse and factorization


$\mathcal{E}_{\phi}$ $\mathcal{H}$ $\mathcal{M}$
ϕ
$\mathcal{R}$
$\mathcal{P}$
$\mathcal{D}$ $\mathcal{G}$ $\mathcal{K}$ τ $\mathcal{E}_{\psi}$
ψ

spatial operators family

General SKO models - Recursive Computational Algorithms

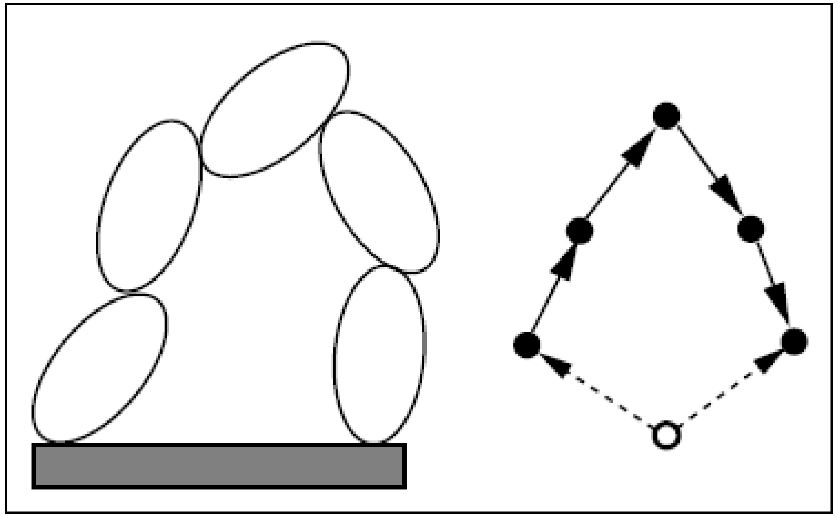
- O(N) Gather and scatter recursions pattern
- O(N) Body velocities scatter recursion
- O(N) CRBs gather recursion
- $O(\mathcal{N}^2)$ mass matrix computation
- O(N) NE scatter/gather inverse dynamics
- $O(\mathcal{N}^2)$ inverse dynamics based mass matrix
- O(N) CRBs based inverse dynamics
- O(N) ATBI gather recursion
- $O(\mathcal{N}^2)$ forward dynamics
- O(N) ATBI forward dynamics
- O(N) hybrid dynamics

While all of this automatically comes for free for SKO models, further optimization of the algorithms is usually possible by exploiting the specific structure of the operator elements in the recursion steps.

When do we not get SKO models ‘naturally’?

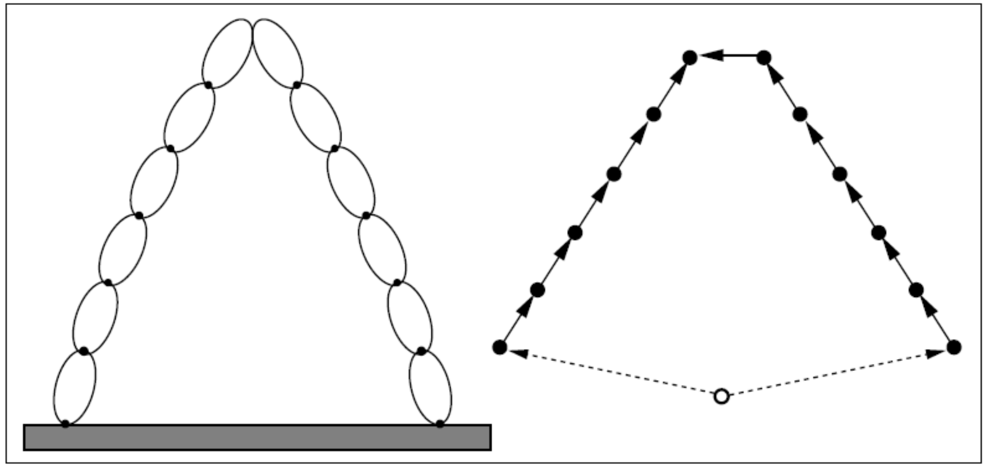
For systems with loops

- The higher powers of the BWA matrix never vanish
- Hence the BWA matrix for a system is not nilpotent
- Hence the 1-resolvent does not exist and so the SKO and SPO operators do not exist



For multiply connected systems

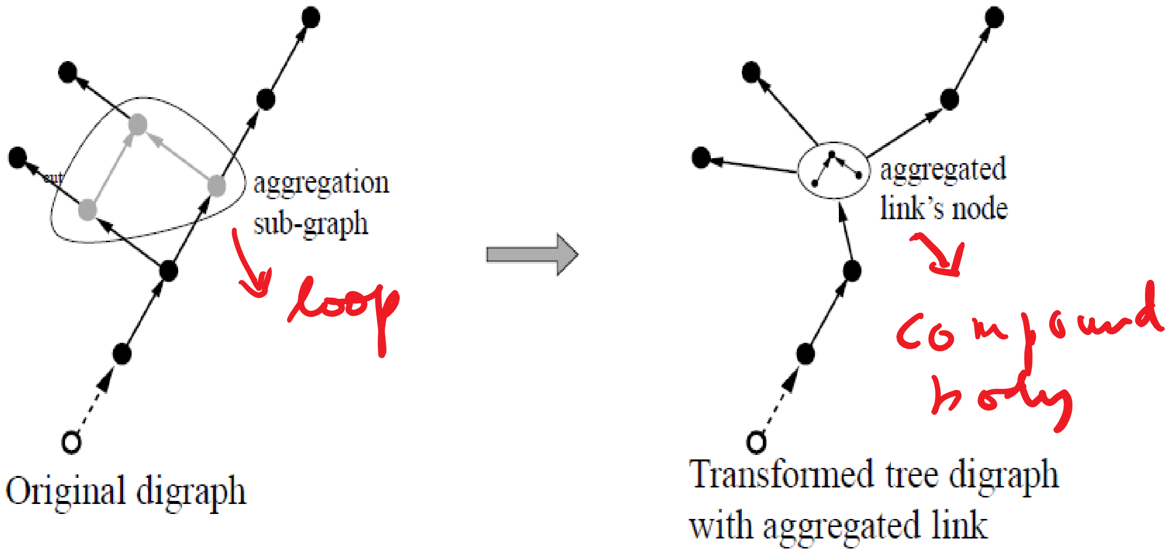
- Such systems do not have cycles, but have multiply connected loops
- The BWA matrix may be nilpotent in this case
- However it is not possible to define coordinates for the multiple paths independently, and H is not well defined
- Hence no SKO model



Constraint Embedding to the rescue

With closed-loop, mass matrix is singular – paradise lost!

Constraint embedding transforms a closed-loop system graph into a tree graph



*The compound body is a **"variable geometry body"!***

- *Minimal coordinate ODE model for a closed-loop system*
- *Tree analytical structure is regained*
- *Well defined non-singular mass matrix*
- *Mass matrix inversion results hold again*
- *Recursive O(N) methods are available again as well*

Paradise regained!

Recap

Summary

- 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

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