**Dynamics and Real-Time Simulation (DARTS) Laboratory**

# Spatial Operator Algebra (SOA)

## *11. Closed-Chain Dynamics (Constraint Embedding)*

***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.



Jet Propulsion Laboratory
California Institute of Technology

# 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

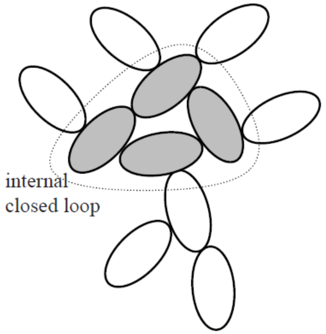
- Looked into the augmented method for closed-chain dynamics (DAE approach)
- Does not have a direct SKO model
- Introduced the notion of operational space inertia matrix (OSIM) and OSCIM
- Discussed the Backward Lyapunov Equation based operator decomposition
- Applied SKO model recursive algorithms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

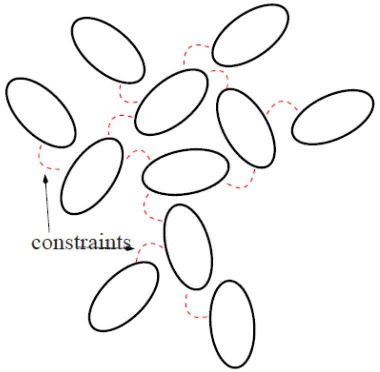
# Closed-Chain Dynamics

# SKO models and loops

- SKO models require an underlying tree structure
- The presence of even a single loop constraint leads to the loss of the tree structure
  - This means that the SKO model analysis and algorithms do not apply!
- The cut-joint method seen earlier provides some – but unsatisfactory – relief
- We will attempt to remedy this situation to allow the use of SKO models with loops

# Closed Chain Modeling Options

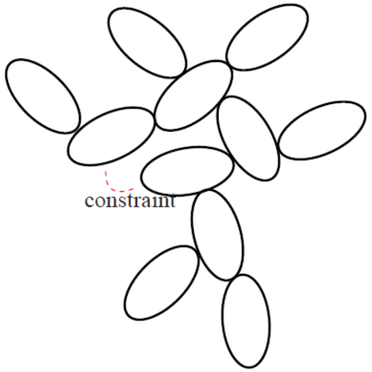



**FA model**

*Non-minimal coords
+ constraints*

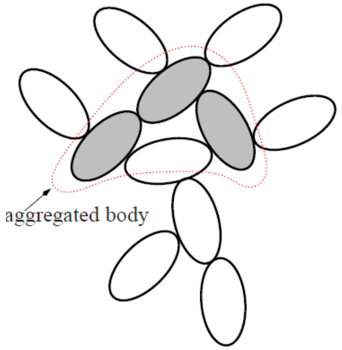
*Simple setup*




**TA model**

*Minimal tree coords
+ constraints*

*Better for large loops*




**CE model**

*Minimal coords*

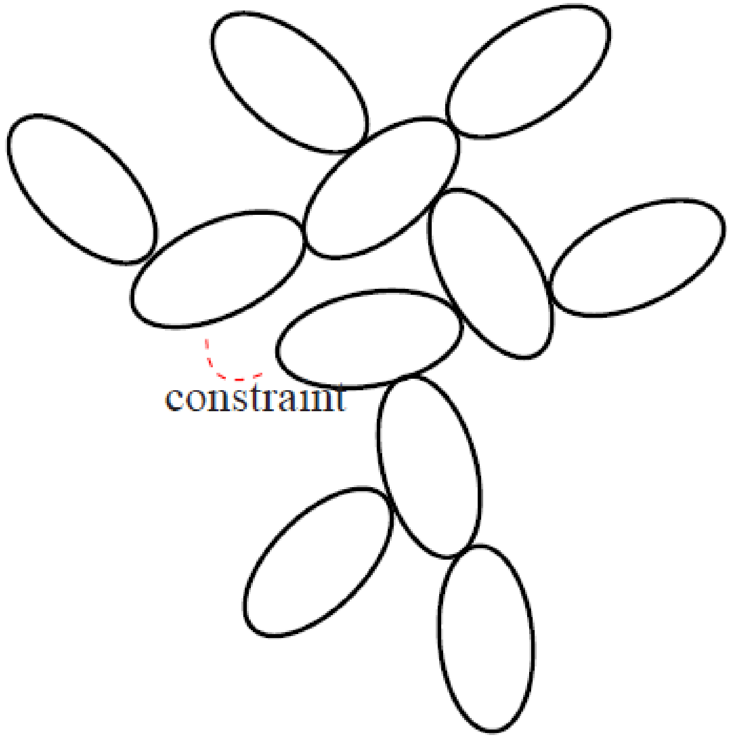
*Optimal for small loops*

# 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



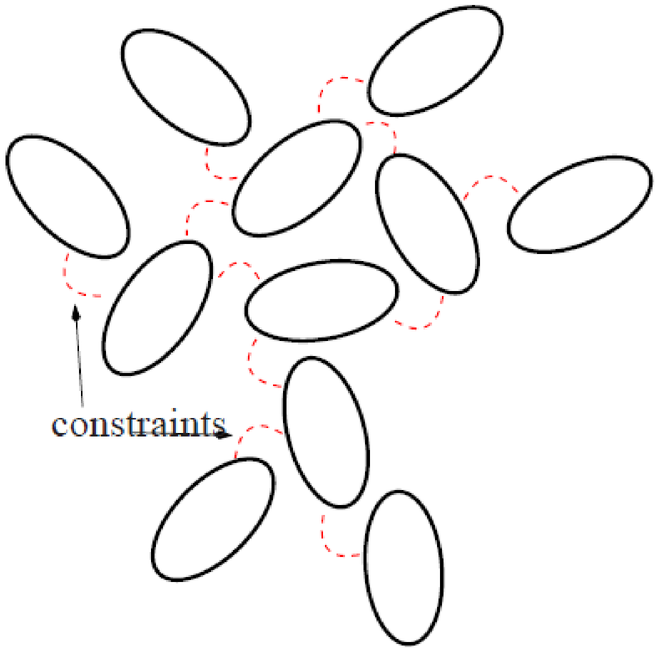
# Augmented method comments

- 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

- However, this still remains a non-minimal coordinates and a DAE approach

# 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

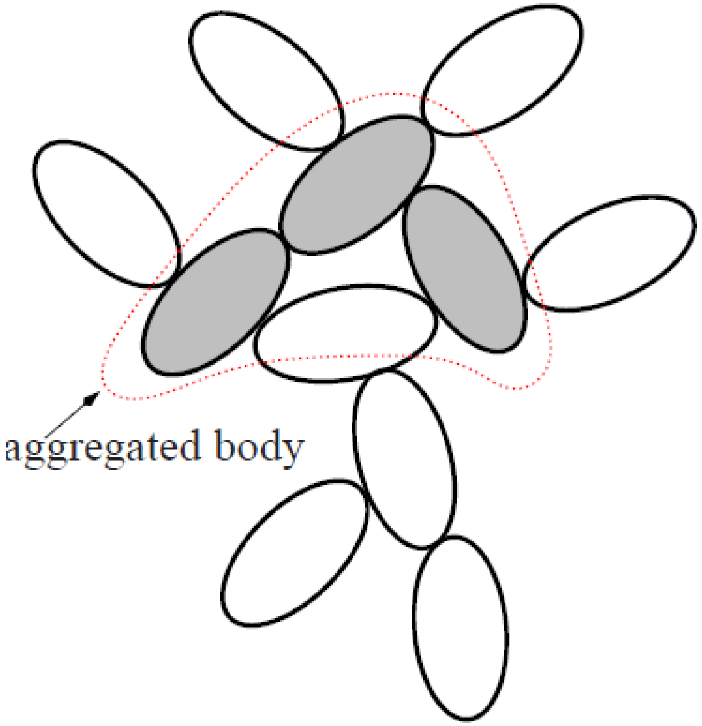


# Projection method comments

- 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

# Constraint embedding solution approach

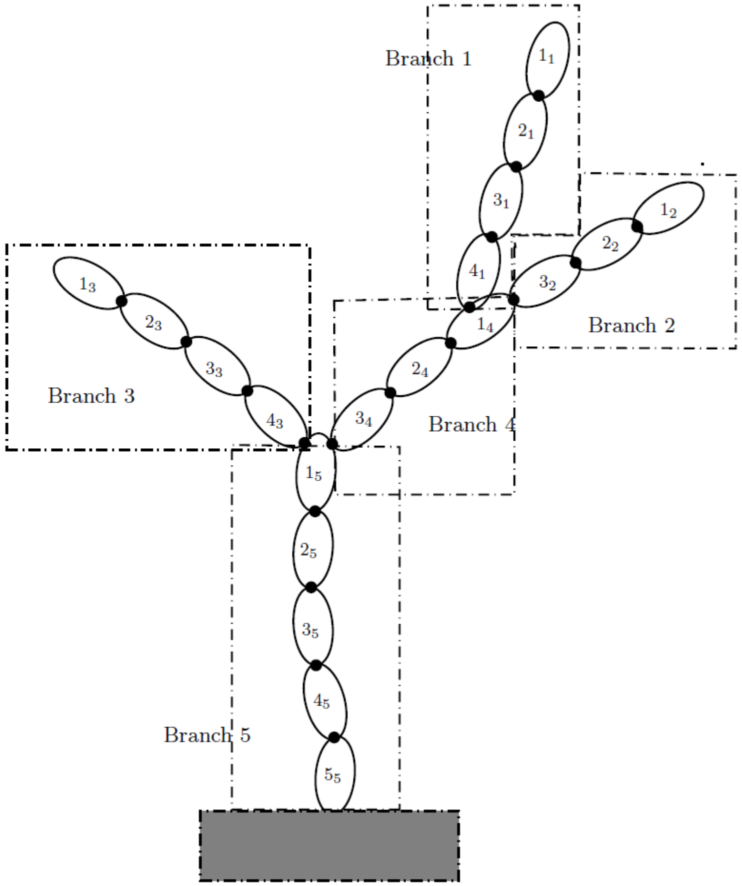
- A minimal coordinate approach that preserves structure
- Uses graph transformation and variable geometry bodies to derive an SKO model and an ODE approach
- All SKO model analysis and recursive algorithms thus apply



# Coarsening Graphs

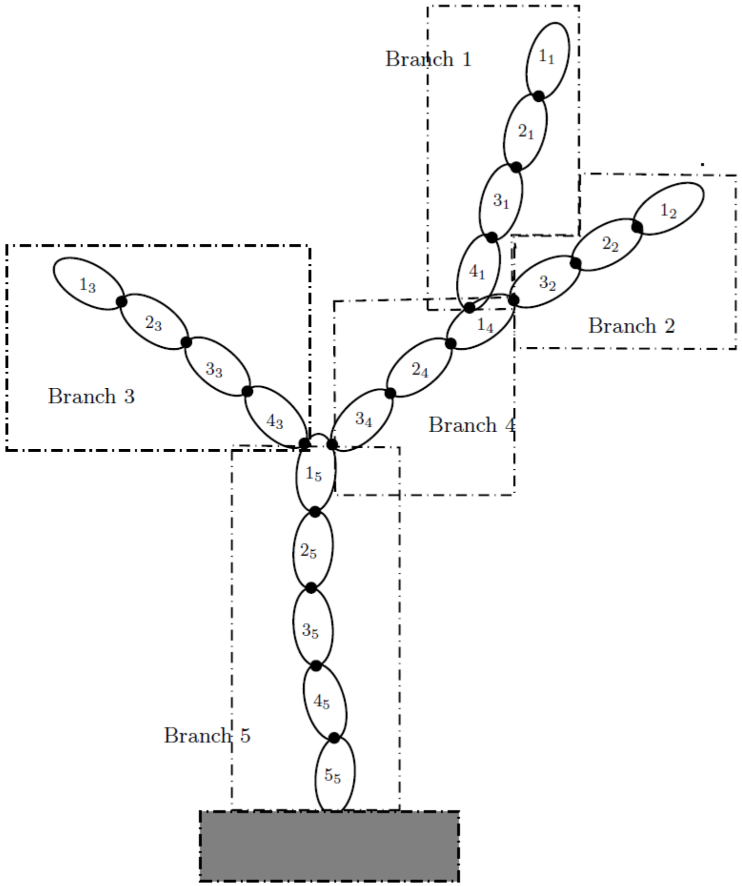
# Coarsened graph representations

- So far our graph representation for bodies has mapped individual bodies to graph nodes
- However, one can also view a multibody system as a collection of connected of sub-trees, i.e. a tree of serial-chains as in the example in the figure
- From this perspective, an alternative graph representation may use nodes for the component serial-chains
- This is a coarser representation of the system using more complex serial chain subgraph nodes



# Coarsening questions

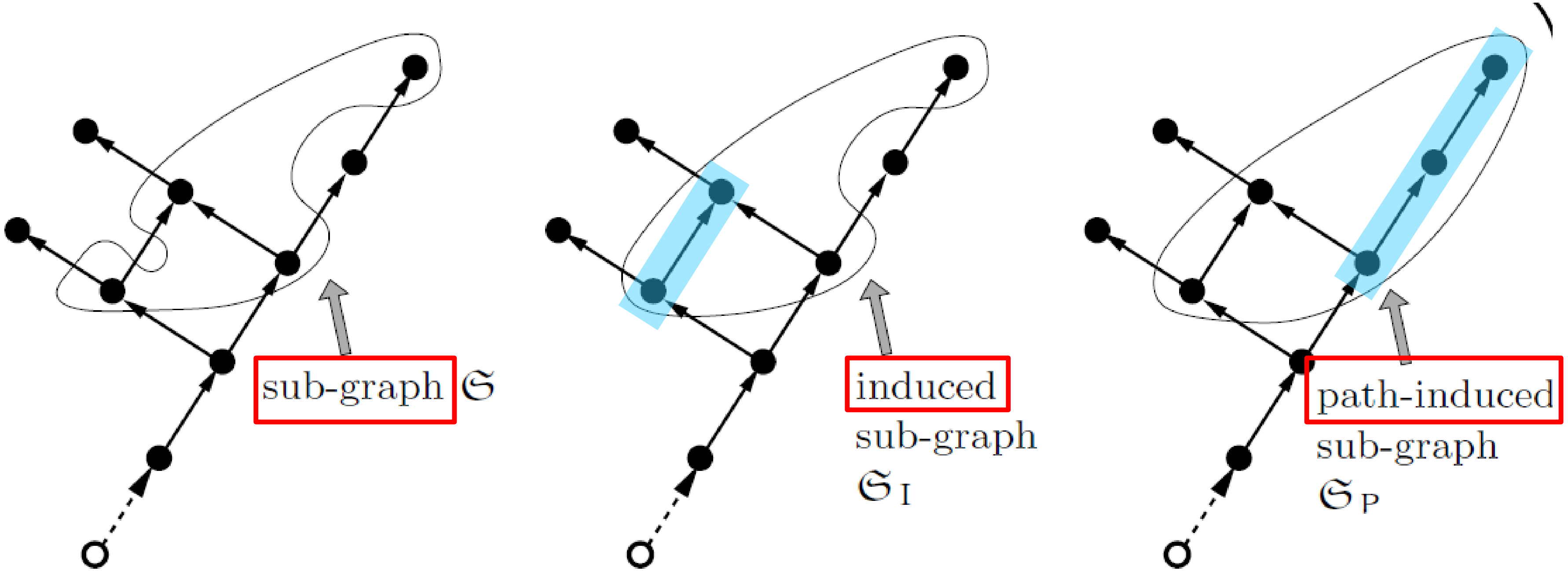
- What kind of coarsening makes sense?
- What are the requirements on the component subgraphs?
- What are the properties of the resulting coarse graph representation?
- What types of coarsening preserve the tree and SKO model structure?



# Partitioning Graphs

# Regular, induced & path-induced subgraphs

A <u>subgraph</u> is a collection of nodes and edges belonging to the parent graph




Arbitrary collection of nodes and edges

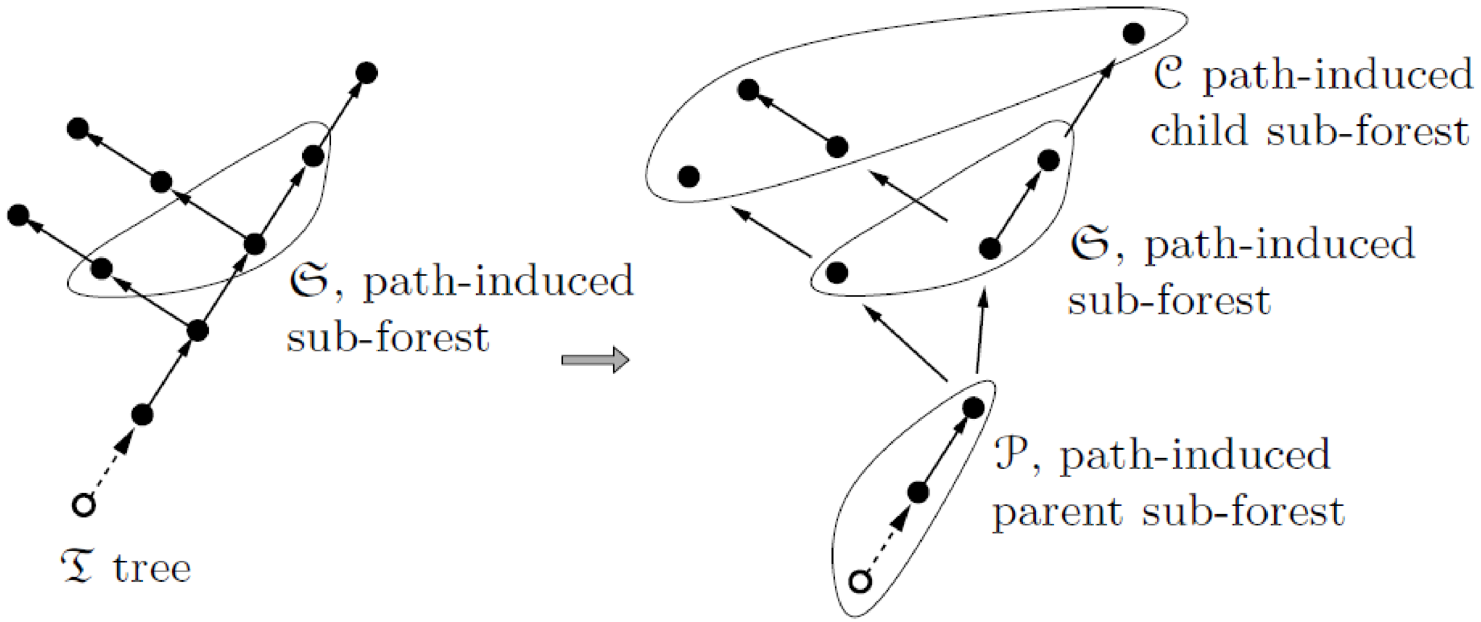
Also includes all edges between nodes in the subgraph

Also includes all edges & nodes on paths between nodes in the subgraph

# Partitioning a tree graph

A <u>path-induced</u> subgraph has the property that it partitions a tree into a disjoint set of

- One or more <u>children</u> path-induced trees
- <u>Itself</u>
- Single <u>parent</u> path-induced tree



# Impact on SKO models

- SKO models require an underlying tree structure
- Path-induced subgraphs partition a tree and a collection of sub-trees
- What is the impact of such partitioning on an SKO model?

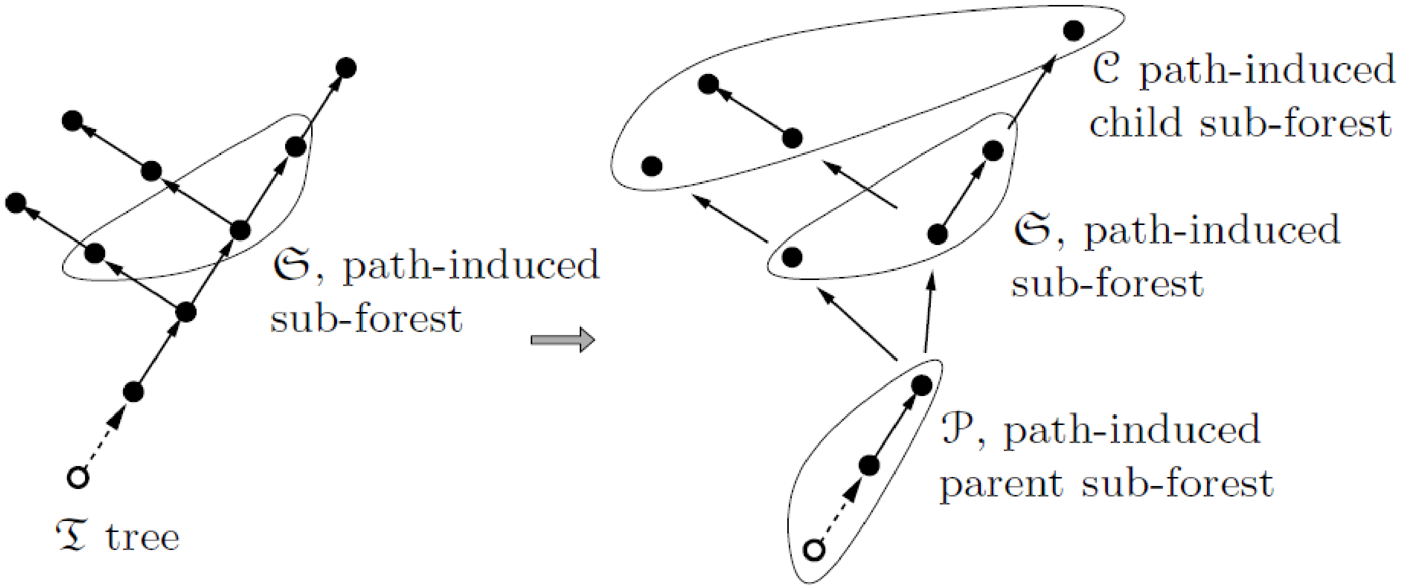
# Partitioning SKO models

# Partitioning SKO models

- Path-induced sub-graphs of SKO-models are SKO models in their own right

$$(\mathbb{H}_{\mathfrak{C}}, \mathbb{A}_{\mathfrak{C}}, \mathbf{M}_{\mathfrak{C}}) \quad (\mathbb{H}_{\mathfrak{S}}, \mathbb{A}_{\mathfrak{S}}, \mathbf{M}_{\mathfrak{S}}) \quad (\mathbb{H}_{\mathcal{P}}, \mathbb{A}_{\mathcal{P}}, \mathbf{M}_{\mathcal{P}})$$

- What is the relationship between the system level SKO model and the partitioned SKO models?



# Partitioned SKO operator

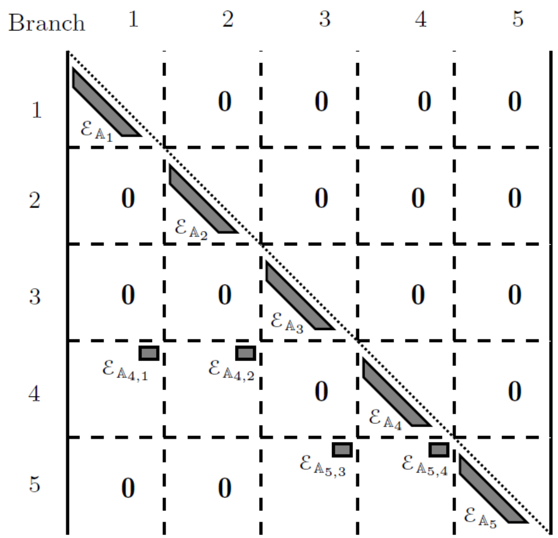
The system level SKO operator has the following partitioned structure:

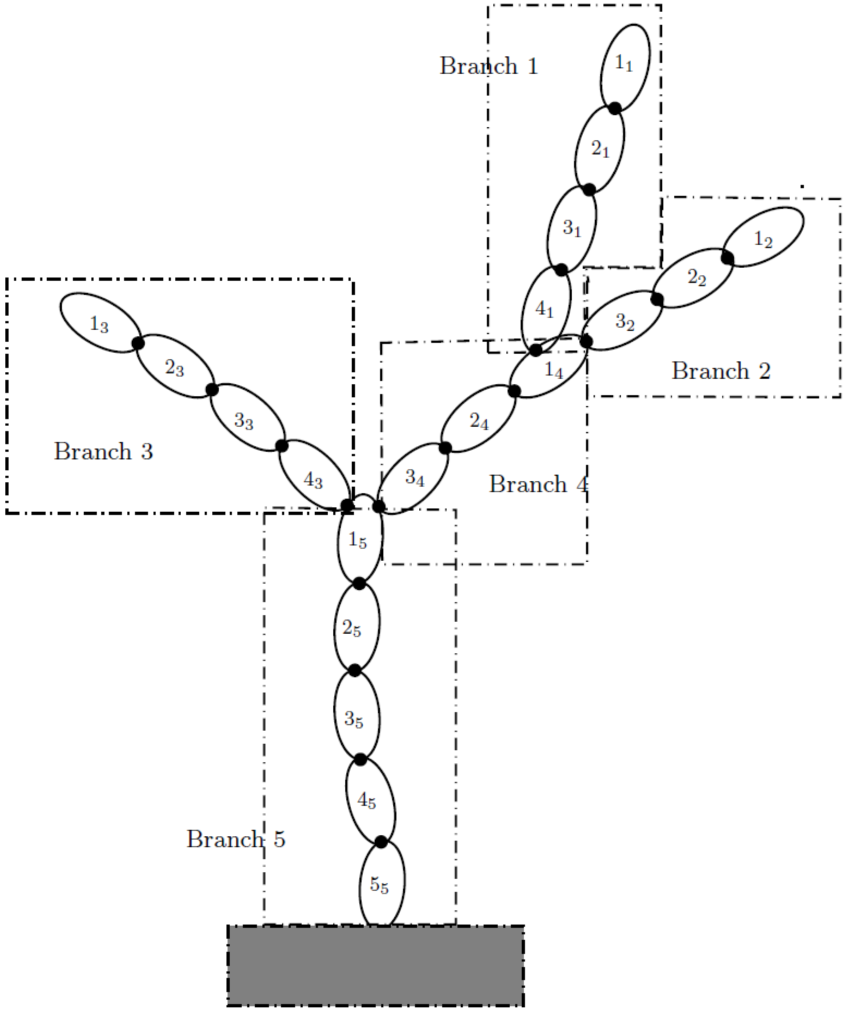
$$\mathcal{E}_{\mathbb{A}} = \begin{pmatrix} \mathcal{E}_{\mathbb{A}_{\mathcal{C}}} & \mathbf{0} & \mathbf{0} \\ \mathcal{B}_{\mathfrak{S}} & \mathcal{E}_{\mathbb{A}_{\mathfrak{S}}} & \mathbf{0} \\ \mathbf{0} & \mathrm{E}_{\mathfrak{S}} & \mathcal{E}_{\mathbb{A}_{\mathcal{P}}} \end{pmatrix}$$

connector to children subgraphs

connector to parent subgraph

# Example of partitioning the SKO operator




$$\mathcal{E}_{\mathbb{A}} = \begin{pmatrix} \mathcal{E}_{\mathbb{A}_1} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathcal{E}_{\mathbb{A}_2} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathcal{E}_{\mathbb{A}_3} & \mathbf{0} & \mathbf{0} \\ \mathcal{E}_{\mathbb{A}_{4,1}} & \mathcal{E}_{\mathbb{A}_{4,2}} & \mathbf{0} & \mathcal{E}_{\mathbb{A}_4} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathcal{E}_{\mathbb{A}_{5,3}} & \mathcal{E}_{\mathbb{A}_{5,4}} & \mathcal{E}_{\mathbb{A}_5} \end{pmatrix}$$

# Partitioned SPO operator

The system level SPO operator can be partitioned using the subgraph SPO operators

$$\mathbb{A}_{\mathcal{C}} \triangleq (\mathbf{I} - \mathcal{E}_{\mathbb{A}_{\mathcal{C}}})^{-1}, \qquad \mathbb{A}_{\mathfrak{S}} \triangleq (\mathbf{I} - \mathcal{E}_{\mathbb{A}_{\mathfrak{S}}})^{-1}, \qquad \mathbb{A}_{\mathcal{P}} \triangleq (\mathbf{I} - \mathcal{E}_{\mathbb{A}_{\mathcal{P}}})^{-1}$$

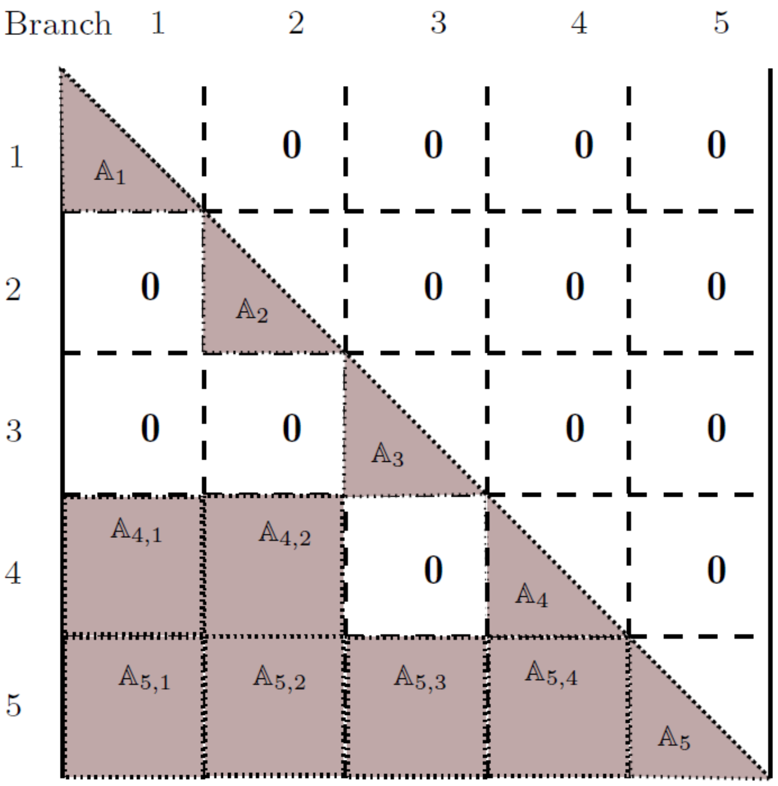
$$\mathbb{A} = \begin{pmatrix} \mathbb{A}_{\mathcal{C}} & \mathbf{0} & \mathbf{0} \\ \mathbb{A}_{\mathfrak{S}} \mathcal{B}_{\mathfrak{S}} \mathbb{A}_{\mathcal{C}} & \mathbb{A}_{\mathfrak{S}} & \mathbf{0} \\ \mathbb{A}_{\mathcal{P}} (\mathrm{E}_{\mathfrak{S}} \mathbb{A}_{\mathfrak{S}} \mathcal{B}_{\mathfrak{S}}) \mathbb{A}_{\mathcal{C}} & \mathbb{A}_{\mathcal{P}} \mathrm{E}_{\mathfrak{S}} \mathbb{A}_{\mathfrak{S}} & \mathbb{A}_{\mathcal{P}} \end{pmatrix}$$

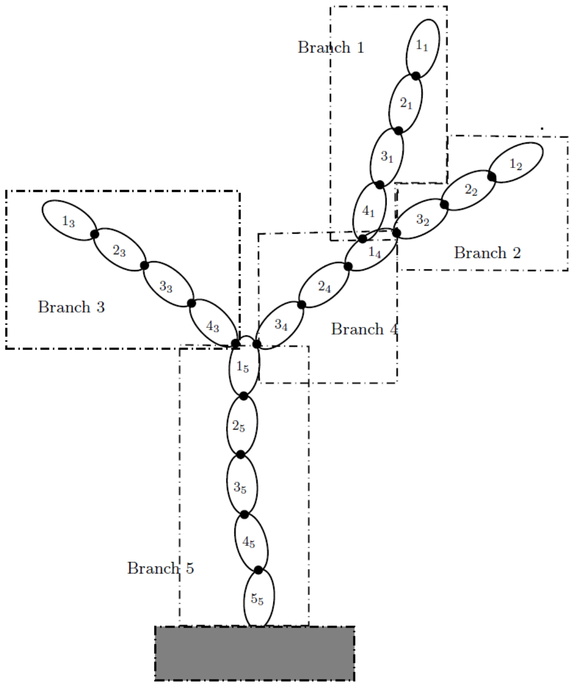
# Proof of partitioning

$$\mathbb{A}^{-1} \overset{8.17}{=\!=} (\mathbf{I} - \mathcal{E}_{\mathbb{A}}) \overset{15.2}{=\!=} \begin{pmatrix} \mathbf{I} - \mathcal{E}_{\mathbb{A}_{\mathfrak{C}}} & \mathbf{0} & \mathbf{0} \\ -\mathcal{B}_{\mathfrak{S}} & \mathbf{I} - \mathcal{E}_{\mathbb{A}_{\mathfrak{S}}} & \mathbf{0} \\ \mathbf{0} & -\mathsf{E}_{\mathfrak{S}} & \mathbf{I} - \mathcal{E}_{\mathbb{A}_{\mathcal{P}}} \end{pmatrix}$$

$$\overset{15.3}{=\!=} \begin{pmatrix} \mathbb{A}_{\mathfrak{C}}^{-1} & \mathbf{0} & \mathbf{0} \\ -\mathcal{B}_{\mathfrak{S}} & \mathbb{A}_{\mathfrak{S}}^{-1} & \mathbf{0} \\ \mathbf{0} & -\mathsf{E}_{\mathfrak{S}} & \mathbb{A}_{\mathcal{P}}^{-1} \end{pmatrix}$$

# SPO operator partitioning example



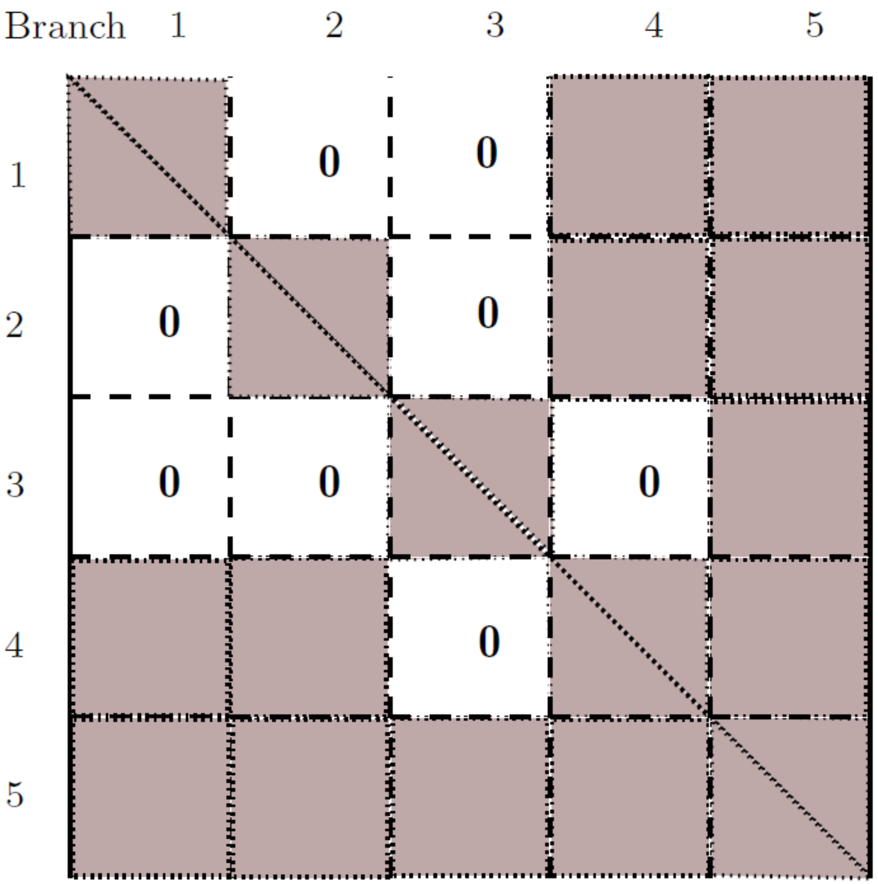

$$\mathbb{A} = \begin{pmatrix} \mathbb{A}_1 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbb{A}_2 & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbb{A}_3 & \mathbf{0} & \mathbf{0} \\ \mathbb{A}_{4,1} & \mathbb{A}_{4,2} & \mathbf{0} & \mathbb{A}_4 & \mathbf{0} \\ \mathbb{A}_{5,1} & \mathbb{A}_{5,2} & \mathbb{A}_{5,3} & \mathbb{A}_{5,4} & \mathbb{A}_5 \end{pmatrix}$$

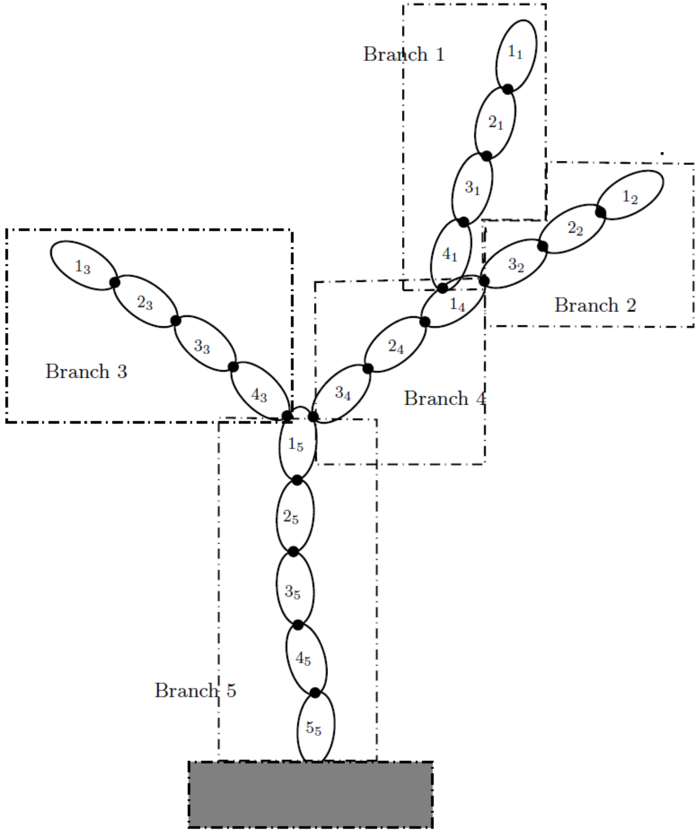
# Partitioned SKO model

Similarly the H and M operators can be partitioned based on the component operators

$$\mathsf{H} = \begin{pmatrix} \mathsf{H}_{\mathfrak{C}} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathsf{H}_{\mathfrak{S}} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathsf{H}_{\mathcal{P}} \end{pmatrix} \qquad \mathbf{M} = \begin{pmatrix} \mathbf{M}_{\mathfrak{C}} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{M}_{\mathfrak{S}} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{M}_{\mathcal{P}} \end{pmatrix}$$

# Partitioned mass matrix



# Aggregating Nodes

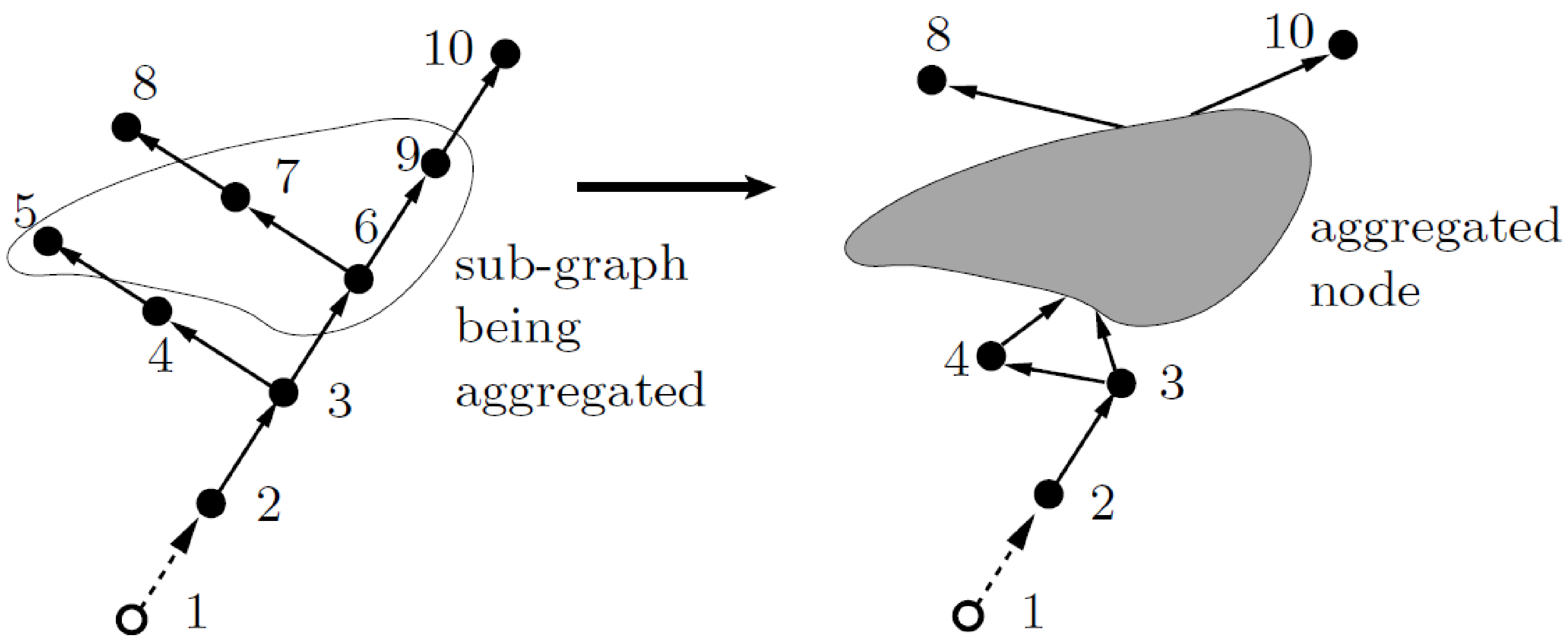
# Tree coarsening

- We have seen earlier that if a subgraph is path-induced then it partitions the graph into sub-trees
- For graph coarsening, we want to represent these sub-trees as nodes in a new graph, i.e. we want to aggregate sub-tree nodes into single compound node
- How do we do this, and what does this do to the tree structure of the new graph?

# Node contractions

Node contraction/aggregation combines nodes into a single node

- All children of the old nodes become children of the new node
- The parents of the old nodes become parents of the new node

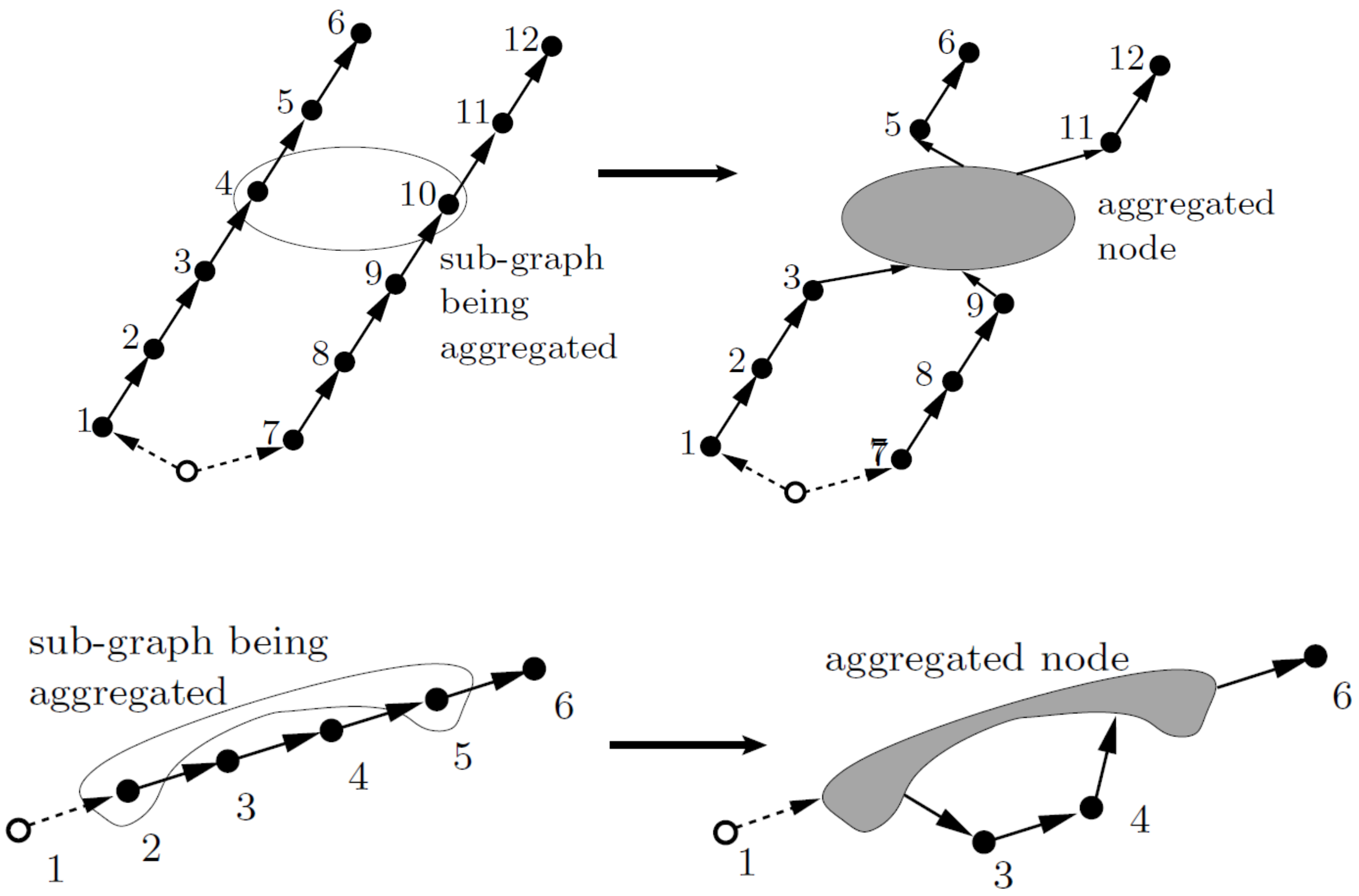



We can <u>lose</u> the tree structure after node contractions.



Jet Propulsion Laboratory
California Institute of Technology

# Node contraction examples




Again, the tree structure has been lost after node aggregation.

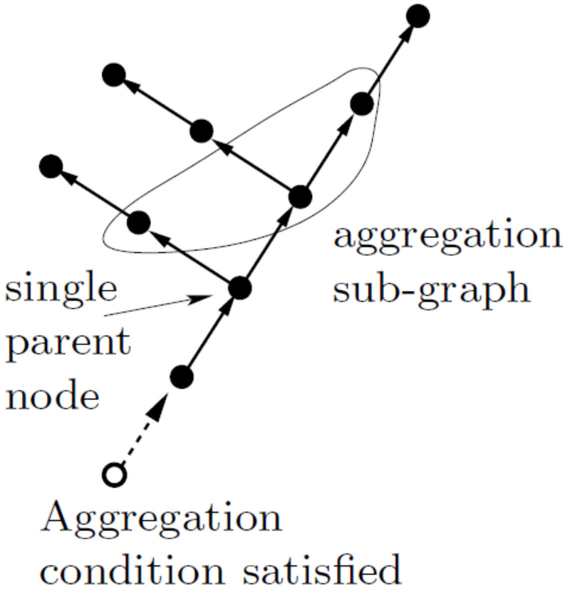
# Preserving tree structure

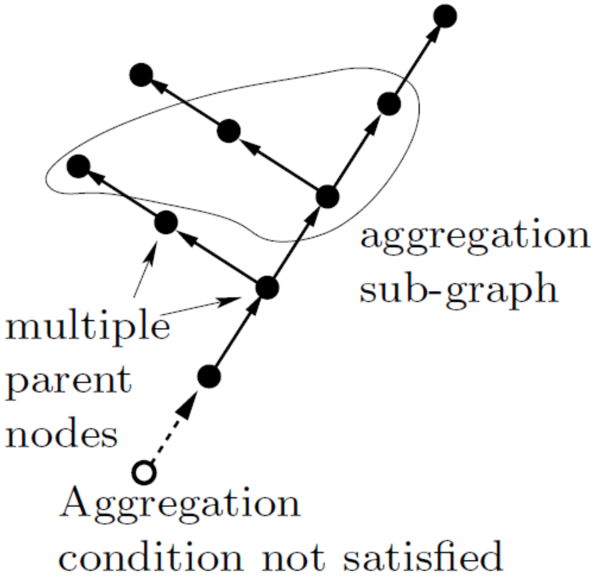
- So even if we take care to work with path-induced subgraphs, node aggregation can lead to a loss in the tree structure of the resulting graph

- How can we coarsen the graph using node aggregation without losing the tree structure?

# Aggregation condition

- When is the tree structure preserved after subgraph nodes aggregation?
- ***Aggregation condition***
  - *the subgraph is an induced subgraph, i.e. it contains all the node edges*
  - *and the subgraph has a single parent node*
- The single parent node requirement forces a path-induced property



# Preserving tree structure

- The new graph with the aggregated nodes is a tree if and only if the subgraph satisfies the aggregation condition

- For a subgraph's aggregation subgraph is defined as the smallest subgraph that contains and satisfies the aggregation condition

| subgraph | induced subgraph | path induced subgraph | aggregation subgraph |
| --- | --- | --- | --- |

$$\mathfrak{S} \subseteq \mathfrak{S}_{\mathrm{I}} \subseteq \mathfrak{S}_{\mathrm{P}} \subseteq \mathfrak{S}_{\mathrm{A}}$$

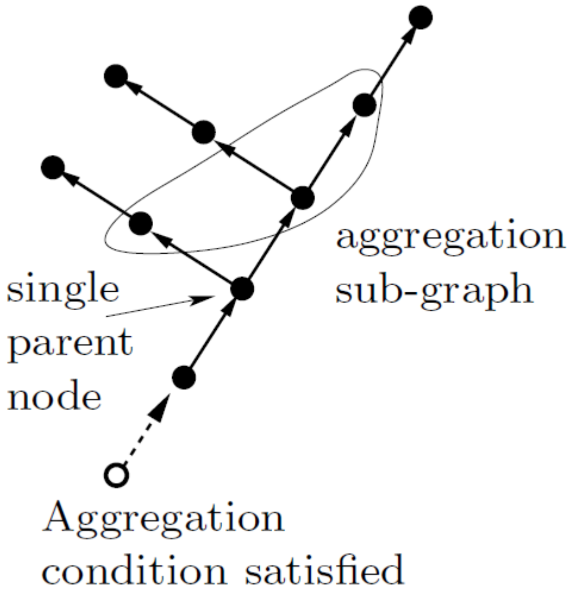
*contains all edges* — *contains all paths* — *has single parent node*

# SKO model for an aggregated tree

# Partitioned tree SKO/SPO operators

Recall earlier partitioning of the SKO and SPO operators:

$$\mathcal{E}_{\mathbb{A}} = \begin{pmatrix} \mathcal{E}_{\mathbb{A}_{\mathcal{C}}} & \mathbf{0} & \mathbf{0} \\ \mathcal{B}_{\mathfrak{S}} & \mathcal{E}_{\mathbb{A}_{\mathfrak{S}}} & \mathbf{0} \\ \mathbf{0} & \mathrm{E}_{\mathfrak{S}} & \mathcal{E}_{\mathbb{A}_{\mathcal{P}}} \end{pmatrix}$$



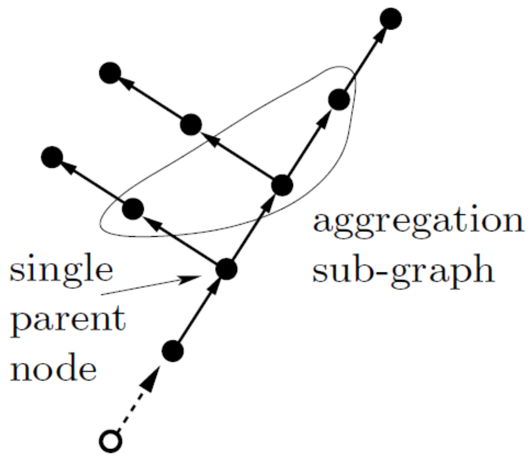

$$\mathbb{A} = \begin{pmatrix} \mathbb{A}_{\mathcal{C}} & \mathbf{0} & \mathbf{0} \\ \mathbb{A}_{\mathfrak{S}}\mathcal{B}_{\mathfrak{S}}\mathbb{A}_{\mathcal{C}} & \mathbb{A}_{\mathfrak{S}} & \mathbf{0} \\ \mathbb{A}_{\mathcal{P}}(\mathrm{E}_{\mathfrak{S}}\mathbb{A}_{\mathfrak{S}}\mathcal{B}_{\mathfrak{S}})\mathbb{A}_{\mathcal{C}} & \mathbb{A}_{\mathcal{P}}\mathrm{E}_{\mathfrak{S}}\mathbb{A}_{\mathfrak{S}} & \mathbb{A}_{\mathcal{P}} \end{pmatrix}$$

# Aggregation for multibody systems

- Node contraction of bodies in multibody context is also referred to aggregation to create compound/aggregated bodies.

- A subgraph of bodies is aggregated in the new graph representation

- The new compound body node can be viewed as representing a variable geometry body
  - Thus a graph node is not restricted to single bodies



# Determing the new SKO operator

- To determine the SKO operator elements we need to identify the recursive relationships between the node velocities in the new tree
- The velocity for an aggregated node is defined as the stacked vector of velocities for the component bodies aggregated in the node
  - Note that the size of this velocity vector is no longer 6, and depends on the number of aggregated bodies

# Component Level Recursive Relationships

# Aggregated body velocity relationship

## General velocity relationship

$$\mathcal{V}(k) = \mathbb{A}^*(\wp(k), k)\mathcal{V}(\wp(k)) + H^*(k)\dot{\theta}(k)$$

## For the aggregated body

$$\mathcal{V}_{\mathfrak{S}} = \mathcal{E}^*_{\mathbb{A}_{\mathfrak{S}}}\mathcal{V}_{\mathfrak{S}} + E^*_{\mathfrak{S}}\mathcal{V}(p) + H^*_{\mathfrak{S}}\dot{\theta}_{\mathfrak{S}} \qquad \text{(implicit)}$$

$$E_{\mathfrak{S}} \triangleq [\mathbf{0}, \ \mathbf{0}, \ \cdots \ \mathbb{A}(p, i)]$$

$$\mathcal{V}_{\mathfrak{S}} \overset{15.3}{=} \boxed{\mathbb{A}^*_{\mathfrak{S}} E^*_{\mathfrak{S}}}\mathcal{V}(p) + \boxed{\mathbb{A}^*_{\mathfrak{S}} H^*_{\mathfrak{S}}}\dot{\theta}_{\mathfrak{S}} \overset{15.20}{=} \mathbb{A}^*_{\mathfrak{S}} E^*_{\mathfrak{S}}\mathcal{V}(p) + \underline{H}^*_{\mathfrak{S}}\dot{\theta}_{\mathfrak{S}} \qquad \text{(explicit)}$$

aggregated node element in the SKO operator

$$\boxed{\mathbb{A}(p, \mathfrak{S}) = E_{\mathfrak{S}}\mathbb{A}_{\mathfrak{S}}}$$

$$\boxed{\underline{H}_{\mathfrak{S}} \triangleq H_{\mathfrak{S}}\mathbb{A}_{\mathfrak{S}}}$$

*aggregated node joint map matrix*

# Aggregated node level velocity and force relationships

$$\mathcal{V}_{\mathfrak{S}} \overset{15.27,15.28}{=} \mathbb{A}^*(\mathrm{p},\mathfrak{S})\mathcal{V}(\mathrm{p}) + \mathrm{H}^*_{\mathfrak{S}}\dot{\theta}_{\mathfrak{S}}$$

$$\mathcal{V}(\mathrm{c}) \overset{15.29,15.30}{=} \mathbb{A}^*(\mathfrak{S},\mathrm{c})\mathcal{V}_{\mathfrak{S}} + \mathrm{H}^*(\mathrm{c})\dot{\theta}(\mathrm{c})$$

$$\mathfrak{f}'_{\mathfrak{S}} \overset{15.21}{=} \sum_{\forall \mathrm{c} \in \mathbf{C}(\mathfrak{S})} \mathbb{A}(\mathfrak{S},\mathrm{c})\mathfrak{f}(\mathrm{c}) + \mathrm{M}_{\mathfrak{S}}\alpha_{\mathfrak{S}} + \mathfrak{b}_{\mathfrak{S}}$$

$$\mathcal{T}_{\mathfrak{S}} \overset{15.21}{=} \mathrm{H}_{\mathfrak{S}}\mathbb{A}_{\mathfrak{S}}\mathfrak{f}'_{\mathfrak{S}} \overset{15.20}{=} \underline{\mathrm{H}}_{\mathfrak{S}}\mathfrak{f}'_{\mathfrak{S}}$$

# Comments on aggregated SKO model

- The size of the recursive relationship depends on the number of aggregated bodies
- This size defines the row size for the aggregated node in the SKO operator
- This is an example of the case where the size is not 6, and can in fact vary depending on the number of bodies being aggregated

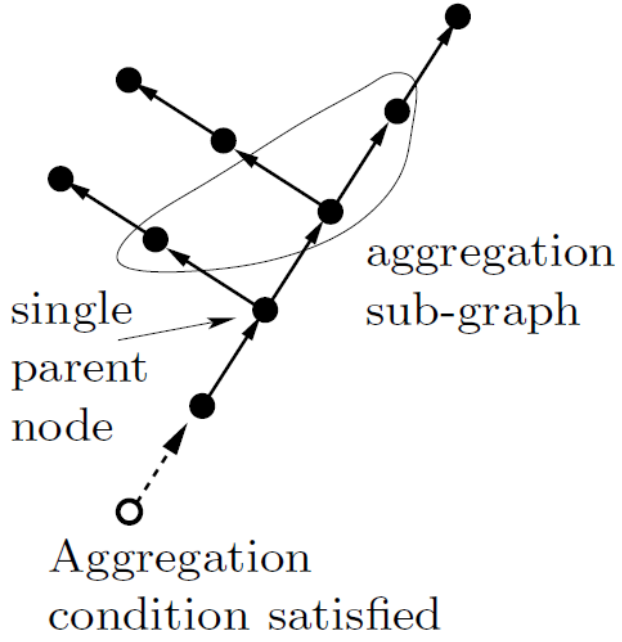
# Aggregated graph – SKO operator

## SKO operator structure for the new aggregated graph

$$\mathcal{E}_{\mathbb{A}} = \begin{pmatrix} \mathcal{E}_{\mathbb{A}_{\mathfrak{C}}} & \mathbf{0} & \mathbf{0} \\ \mathcal{B}_{\mathfrak{S}} & \mathcal{E}_{\mathbb{A}_{\mathfrak{S}}} & \mathbf{0} \\ \mathbf{0} & \mathrm{E}_{\mathfrak{S}} & \mathcal{E}_{\mathbb{A}_{\mathfrak{P}}} \end{pmatrix}$$

$$\mathcal{E}_{\mathbb{A}_{\mathfrak{a}}} \triangleq \begin{pmatrix} \mathcal{E}_{\mathbb{A}_{\mathfrak{C}}} & \mathbf{0} & \mathbf{0} \\ \mathcal{B}_{\mathfrak{S}} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \boxed{\mathrm{E}_{\mathfrak{S}} \mathbb{A}_{\mathfrak{S}}} & \mathcal{E}_{\mathbb{A}_{\mathfrak{P}}} \end{pmatrix}$$

$$\mathbb{A}(\mathfrak{p}, \mathfrak{S}) \equiv \mathrm{E}_{\mathfrak{S}} \mathbb{A}_{\mathfrak{S}}$$



# Aggregated graph – SPO operator

SPO operator structure for the new aggregated graph:

$$\mathbb{A} = \begin{pmatrix} \mathbb{A}_{\mathcal{C}} & \mathbf{0} & \mathbf{0} \\ \mathbb{A}_{\mathfrak{S}} \mathcal{B}_{\mathfrak{S}} \mathbb{A}_{\mathcal{C}} & \mathbb{A}_{\mathfrak{S}} & \mathbf{0} \\ \mathbb{A}_{\mathcal{P}} (\mathrm{E}_{\mathfrak{S}} \mathbb{A}_{\mathfrak{S}} \mathcal{B}_{\mathfrak{S}}) \mathbb{A}_{\mathcal{C}} & \mathbb{A}_{\mathcal{P}} \mathrm{E}_{\mathfrak{S}} \mathbb{A}_{\mathfrak{S}} & \mathbb{A}_{\mathcal{P}} \end{pmatrix}$$

$$\mathbb{A}_{a} \triangleq \mathfrak{J}_{a} \mathbb{A} \stackrel{15.4}{=} \begin{pmatrix} \mathbb{A}_{\mathcal{C}} & \mathbf{0} & \mathbf{0} \\ \boxed{\mathcal{B}_{\mathfrak{S}} \mathbb{A}_{\mathcal{C}}} & \boxed{\mathbf{I}} & \mathbf{0} \\ \mathbb{A}_{\mathcal{P}} (\mathrm{E}_{\mathfrak{S}} \mathbb{A}_{\mathfrak{S}} \mathcal{B}_{\mathfrak{S}}) \mathbb{A}_{\mathcal{C}} & \mathbb{A}_{\mathcal{P}} \mathrm{E}_{\mathfrak{S}} \mathbb{A}_{\mathfrak{S}} & \mathbb{A}_{\mathcal{P}} \end{pmatrix}$$

where $\mathfrak{J}_{a} \triangleq \begin{pmatrix} \mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbb{A}_{\mathfrak{S}}^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} \end{pmatrix}$

# Aggregated graph – SKO model

## Partitioning

$$\dot{\theta} = \begin{bmatrix} \dot{\theta}_{\mathcal{C}} \\ \dot{\theta}_{\mathfrak{S}} \\ \dot{\theta}_{\mathcal{P}} \end{bmatrix}, \quad \mathcal{V} = \begin{bmatrix} \mathcal{V}_{\mathcal{C}} \\ \mathcal{V}_{\mathfrak{S}} \\ \mathcal{V}_{\mathcal{P}} \end{bmatrix}, \quad \mathfrak{f} = \begin{bmatrix} \mathfrak{f}_{\mathcal{C}} \\ \mathfrak{f}_{\mathfrak{S}} \\ \mathfrak{f}_{\mathcal{P}} \end{bmatrix}, \quad \mathcal{T} = \begin{bmatrix} \mathcal{T}_{\mathcal{C}} \\ \mathcal{T}_{\mathfrak{S}} \\ \mathcal{T}_{\mathcal{P}} \end{bmatrix}$$

$$\mathrm{H}_{\mathfrak{a}} \stackrel{\triangle}{=} \mathrm{H}\mathfrak{J}_{\mathfrak{a}}^{-1} \stackrel{15.18}{=} \begin{pmatrix} \mathrm{H}_{\mathcal{C}} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \underline{\mathrm{H}}_{\mathfrak{S}} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathrm{H}_{\mathcal{P}} \end{pmatrix}$$

$$\underline{\mathrm{H}}_{\mathfrak{S}} \stackrel{\triangle}{=} \mathrm{H}_{\mathfrak{S}} \mathbb{A}_{\mathfrak{S}}$$

*joint map matrix for the aggregated body*

# Aggregated graph – SKO model (contd)

$$\mathcal{V} = \mathbb{A}_{\mathfrak{a}}^{*} \mathrm{H}_{\mathfrak{a}}^{*} \dot{\theta}$$

$$\alpha = \mathbb{A}_{\mathfrak{a}}^{*} (\mathrm{H}_{\mathfrak{a}}^{*} \ddot{\theta} + \underline{\mathfrak{a}}), \quad where \quad \underline{\mathfrak{a}} \triangleq \mathfrak{J}_{\mathfrak{a}}^{-*} \mathfrak{a} = \begin{bmatrix} \mathfrak{a}_{\mathcal{C}} \\ \underline{\mathfrak{a}}_{\mathfrak{S}} \\ \mathfrak{a}_{\mathcal{P}} \end{bmatrix}, \quad \underline{\mathfrak{a}}_{\mathfrak{S}} \triangleq \mathbb{A}_{\mathfrak{S}}^{*} \mathfrak{a}_{\mathfrak{S}}$$

$$\underline{\mathfrak{f}} = \mathbb{A}_{\mathfrak{a}} (\mathbf{M}\alpha + \mathfrak{b}), \quad where \quad \underline{\mathfrak{f}} \triangleq \mathfrak{J}_{\mathfrak{a}} \mathfrak{f} = \begin{bmatrix} \mathfrak{f}_{\mathcal{C}} \\ \mathfrak{f}_{\mathfrak{S}}' \\ \mathfrak{f}_{\mathcal{P}} \end{bmatrix}, \quad \mathfrak{f}_{\mathfrak{S}}' \triangleq \mathbb{A}_{\mathfrak{S}}^{-1} \mathfrak{f}_{\mathfrak{S}}$$

$$\mathcal{T} = \mathrm{H}_{\mathfrak{a}} \underline{\mathfrak{f}}$$

$$\mathcal{T} = \mathcal{M}\ddot{\theta} + \mathcal{C}$$

$$\mathcal{M} = \mathrm{H}_{\mathfrak{a}} \mathbb{A}_{\mathfrak{a}} \mathbf{M} \mathbb{A}_{\mathfrak{a}}^{*} \mathrm{H}_{\mathfrak{a}}^{*}$$

*transformed system mass matrix*

$$\mathcal{C} \triangleq \mathrm{H}_{\mathfrak{a}} \mathbb{A}_{\mathfrak{a}} (\mathbf{M} \mathbb{A}_{\mathfrak{a}}^{*} \underline{\mathfrak{a}} + \mathfrak{b})$$

# Equations of motion invariance

The equations of motion term values remain the same with and without aggregation

$$\mathcal{M} = \mathrm{H}\mathbb{A}\mathbf{M}\mathbb{A}^*\mathrm{H}^* = \mathrm{H}_{\mathfrak{a}}\mathbb{A}_{\mathfrak{a}}\mathbf{M}\mathbb{A}_{\mathfrak{a}}^*\mathrm{H}_{\mathfrak{a}}^*$$

$$\mathcal{C} = \mathrm{H}\mathbb{A}(\mathbf{M}\mathbb{A}^*\mathfrak{a} + \mathfrak{b}) = \mathrm{H}_{\mathfrak{a}}\mathbb{A}_{\mathfrak{a}}(\mathbf{M}\mathbb{A}_{\mathfrak{a}}^*\underline{\mathfrak{a}} + \mathfrak{b})$$

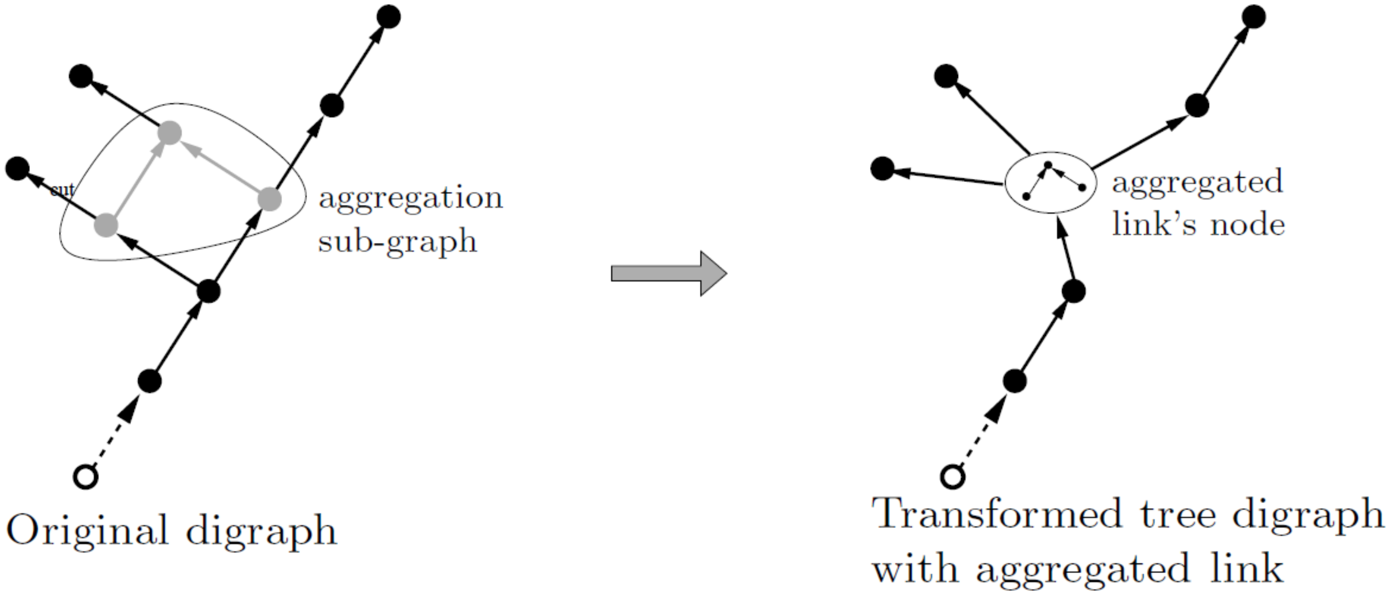
# Constraint Embedding

# Constraint embedding approach

*We have already seen how to partition & transform graphs into new graphs using path-induced subgraphs and node aggregation.*

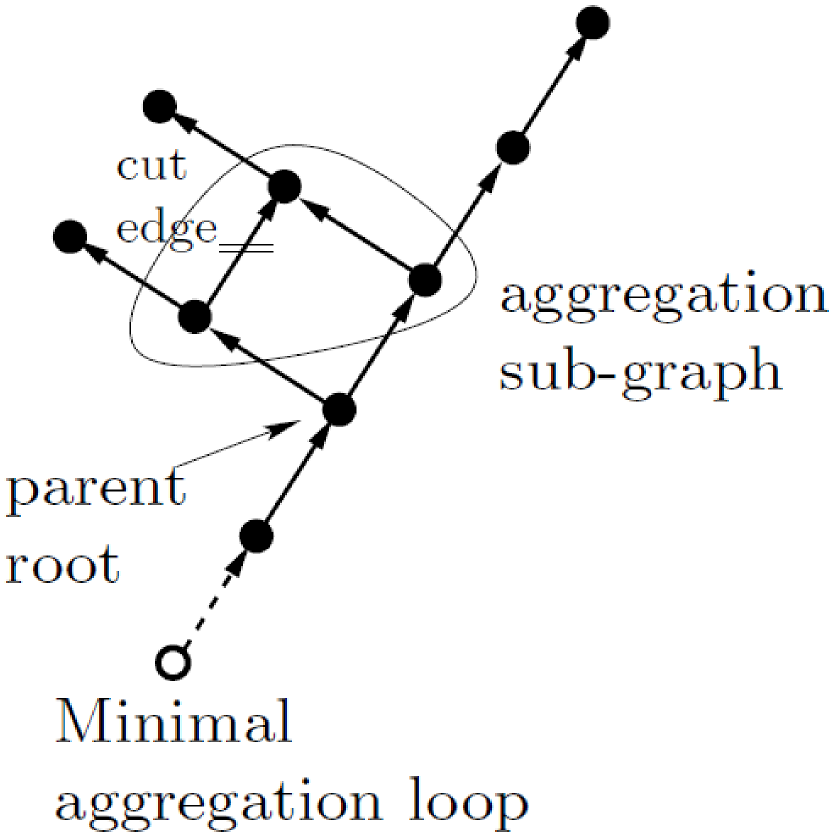
- Cut loops to create spanning tree + constraints (just as for augmented approach)
- For each constraint, identify the constraint nodes/bodies
- Identify a subgraph for the constraint bodies, such that collapsing the subgraph into a new node, leaves us with a tree and an SKO model
- Define the joint map matrix for the aggregated bodies to take into account the loop constraints

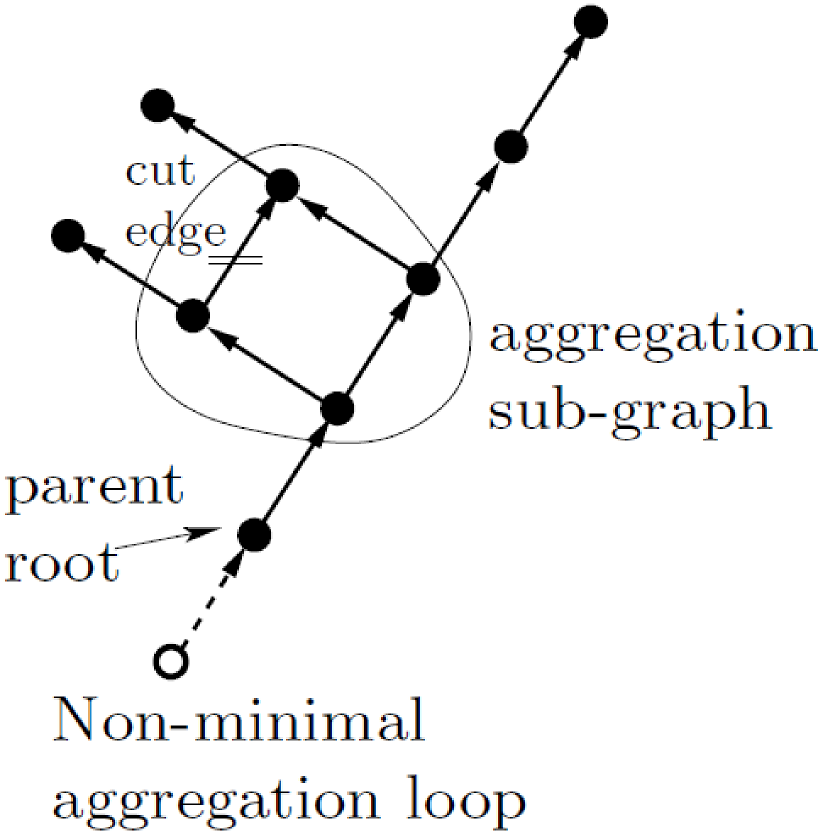
# Approach




- Introduce cut-joints for each closed loop
- For each loop identify the aggregation subgraph for the constraint node/bodies
  - Smallest sub-tree containing the nodes
  - Drop the root node
- Use aggregation to transform the graph
- Absorb the cut-joint into the new compound body hinge

# Minimal aggregation subgraphs



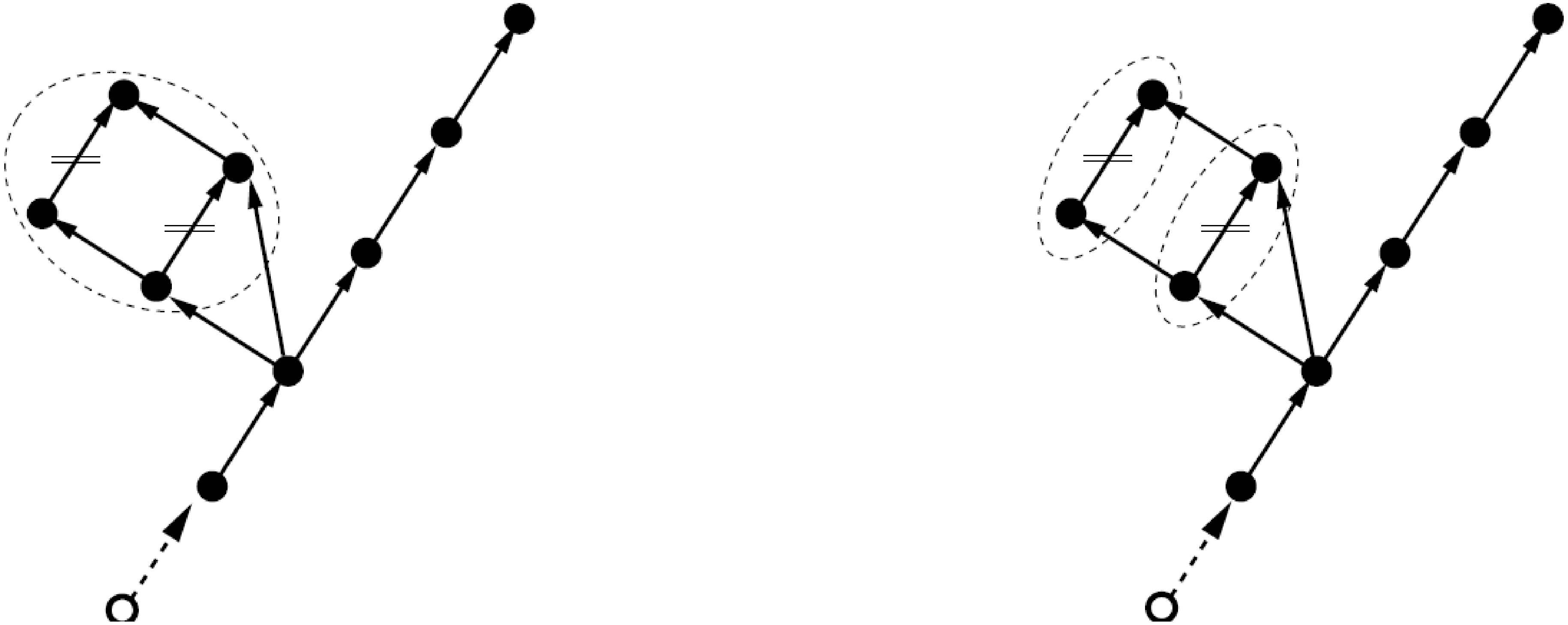



# Compound body and hinge

- The aggregation node contains all the bodies involved in the closed loop
  - Identify a minimal set of coordinates for just the loop
  - Parameterize the loop motion (including shape) based on these coordinates

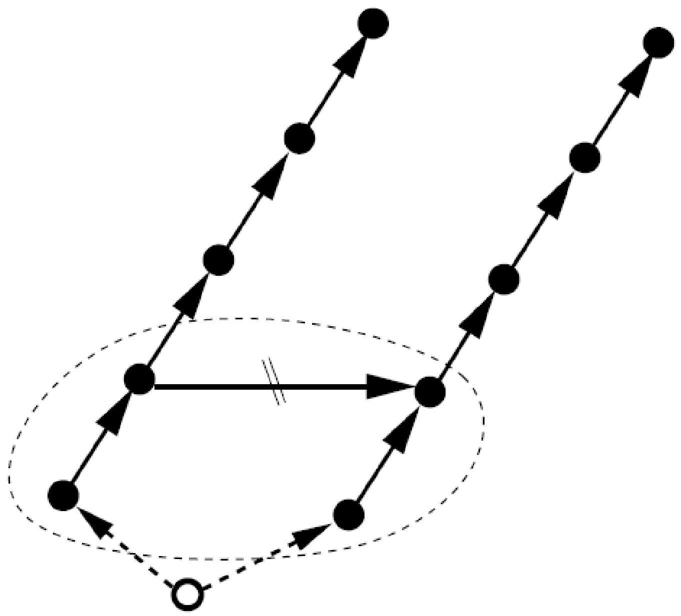
- Define a new compound hinge for this body with these minimal coordinates
  - This hinge replaces the physical hinge in the aggregated SKO model
  - The loop constraint is essentially buried within the compound body & hinge and eliminated from the aggregated tree
  - The compound hinge not only defines the articulation of the compound body, but also the "shape" of these variable geometry bodies

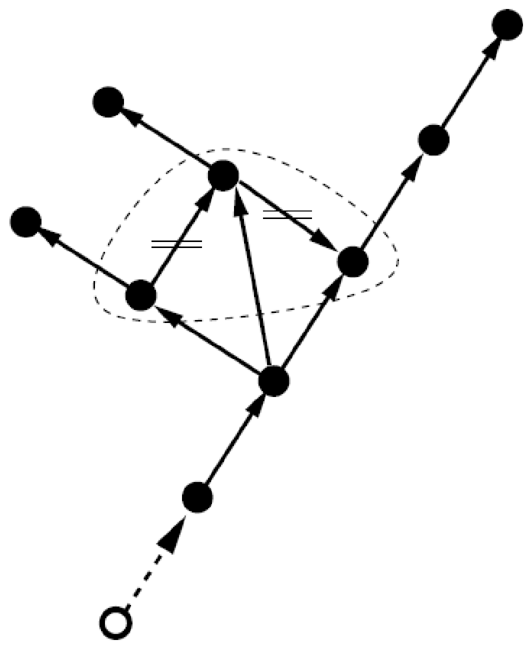
# Constraint embedding examples



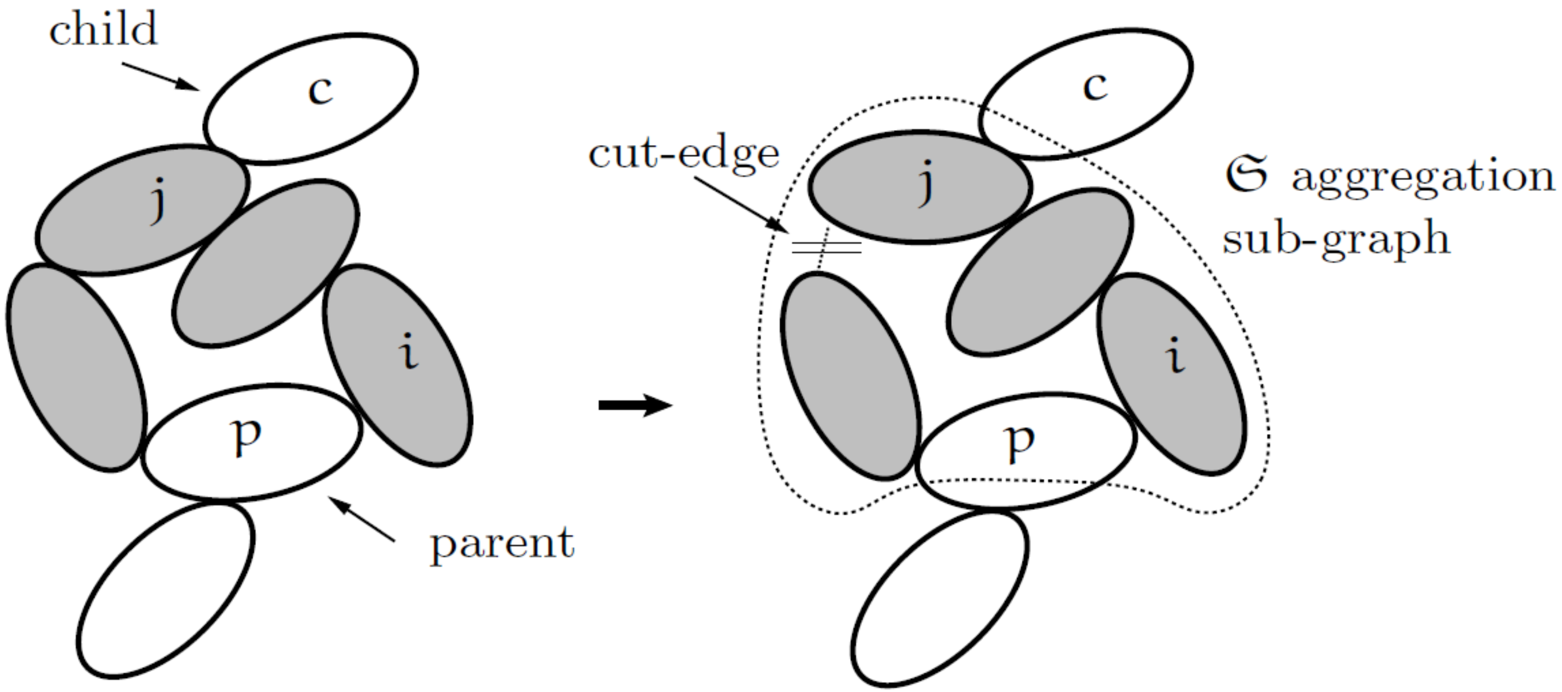
*There are multiple options for defining the aggregation node*

# Constraint embedding examples (contd)





# Compound hinge definition




$$\dot{\theta}_{\mathfrak{S}} = X_{\mathfrak{S}} \dot{\theta}_{R\mathfrak{S}}$$

*The mapping from the independent to the full set of coordinates for the aggregation subgraph*

# Compound hinge joint map matrix

$$\dot{\theta}_{\mathfrak{S}} = X_{\mathfrak{S}} \dot{\theta}_{R\mathfrak{S}}$$

$$\underline{H}_{R\mathfrak{S}} \triangleq X^*_{\mathfrak{S}} \underline{H}_{\mathfrak{S}} \overset{15.20}{=} X^*_{\mathfrak{S}} H_{\mathfrak{S}} \mathbb{A}_{\mathfrak{S}} \quad \text{and} \quad \dot{\theta}_R \triangleq \begin{bmatrix} \dot{\theta}_{\mathcal{C}} \\ \dot{\theta}_{R\mathfrak{S}} \\ \dot{\theta}_{\mathcal{P}} \end{bmatrix}$$

*minimal coordinates velocity mapping*

*The* $\underline{H}_{R\mathfrak{S}} \triangleq X^*_{\mathfrak{S}} \underline{H}_{\mathfrak{S}}$ *aggregated body joint map matrix controls the articulation as well as the internal shape of the compound body.*

# At the operator level …

$$\underline{H}_{R\mathfrak{S}} \triangleq X^*_{\mathfrak{S}} \underline{H}_{\mathfrak{S}} \overset{15.20}{=} X^*_{\mathfrak{S}} H_{\mathfrak{S}} \mathbb{A}_{\mathfrak{S}} \quad \text{and} \quad \dot{\theta}_R \triangleq \begin{bmatrix} \dot{\theta}_{\mathcal{C}} \\ \dot{\theta}_{R\mathfrak{S}} \\ \dot{\theta}_{\mathcal{P}} \end{bmatrix}$$

$$H_{R\mathfrak{a}} \triangleq \begin{pmatrix} \mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & X^*_{\mathfrak{S}} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} \end{pmatrix} H_{\mathfrak{a}} \overset{15.20}{=} \begin{pmatrix} H_{\mathcal{C}} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \underline{H}_{R\mathfrak{S}} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & H_{\mathcal{P}} \end{pmatrix}$$

$$\mathcal{V} = \mathbb{A}^* H^* \dot{\theta} = \mathbb{A}^*_{\mathfrak{a}} H^*_{\mathfrak{a}} \dot{\theta} = \mathbb{A}^*_{\mathfrak{a}} H^*_{R\mathfrak{a}} \dot{\theta}_R$$

# SKO model equations of motion with constraint embedding

$$\mathcal{V} = \mathbb{A}_{\mathfrak{a}}^{*} \mathrm{H}_{\mathrm{R}\mathfrak{a}}^{*} \dot{\theta}_{\mathrm{R}}$$

$$\alpha = \mathbb{A}_{\mathfrak{a}}^{*} (\mathrm{H}_{\mathrm{R}\mathfrak{a}}^{*} \ddot{\theta}_{\mathrm{R}} + \mathfrak{a}'), \quad \textit{where } \mathfrak{a}' \triangleq \begin{bmatrix} \mathfrak{a}_{\mathcal{C}} \\ \mathfrak{a}'_{\mathfrak{S}} \\ \mathfrak{a}_{\mathcal{P}} \end{bmatrix}, \quad \mathfrak{a}'_{\mathfrak{S}} \triangleq \underline{\mathfrak{a}}_{\mathfrak{S}} + \underline{\mathrm{H}}_{\mathfrak{S}}^{*} \dot{\mathrm{X}}_{\mathfrak{S}} \dot{\theta}_{\mathrm{R}\mathfrak{S}}$$

$$\underline{\mathfrak{f}} = \mathbb{A}_{\mathfrak{a}} (\mathbf{M}\alpha + \mathfrak{b})$$

$$\mathcal{T}_{\mathrm{R}} = \mathrm{H}_{\mathrm{R}\mathfrak{a}} \underline{\mathfrak{f}}$$

$$\mathcal{T}_{\mathrm{R}} = \mathcal{M}_{\mathrm{r}} \ddot{\theta}_{\mathrm{R}} + \mathcal{C}_{\mathrm{r}}$$

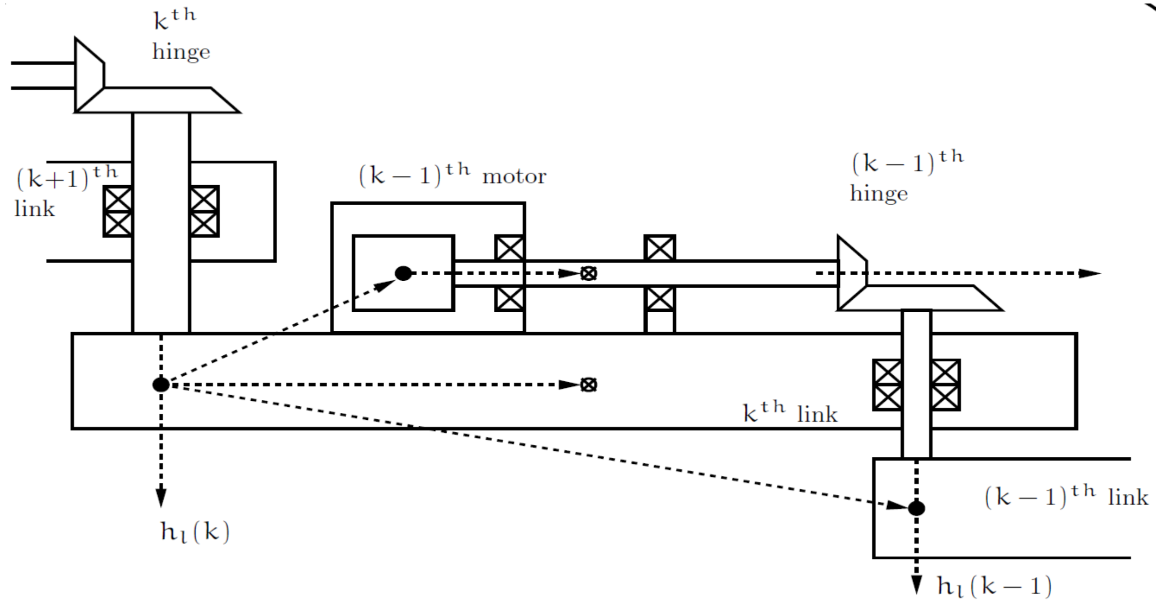
$$\mathcal{M}_{\mathrm{r}} = \mathrm{H}_{\mathrm{R}\mathfrak{a}} \mathbb{A}_{\mathfrak{a}} \mathbf{M} \mathbb{A}_{\mathfrak{a}}^{*} \mathrm{H}_{\mathrm{R}\mathfrak{a}}^{*}$$

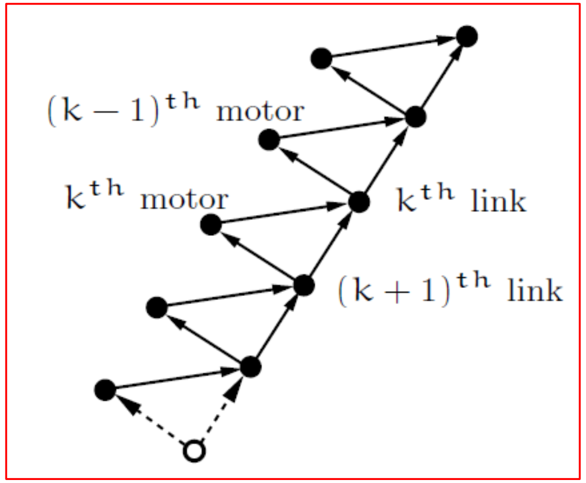
*minimal coordinate mass matrix*

$$\mathcal{C}_{\mathrm{r}} \triangleq \mathrm{H}_{\mathrm{R}\mathfrak{a}} \mathbb{A}_{\mathfrak{a}} (\mathbf{M} \mathbb{A}_{\mathfrak{a}}^{*} \mathfrak{a}' + \mathfrak{b})$$

# Examples

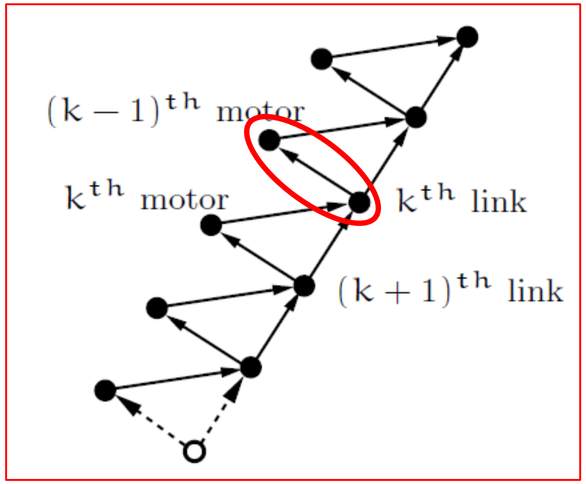
# Geared links




Gearing introduces loops involving 3 bodies at each stage

# Geared aggregation node – 2 bodies

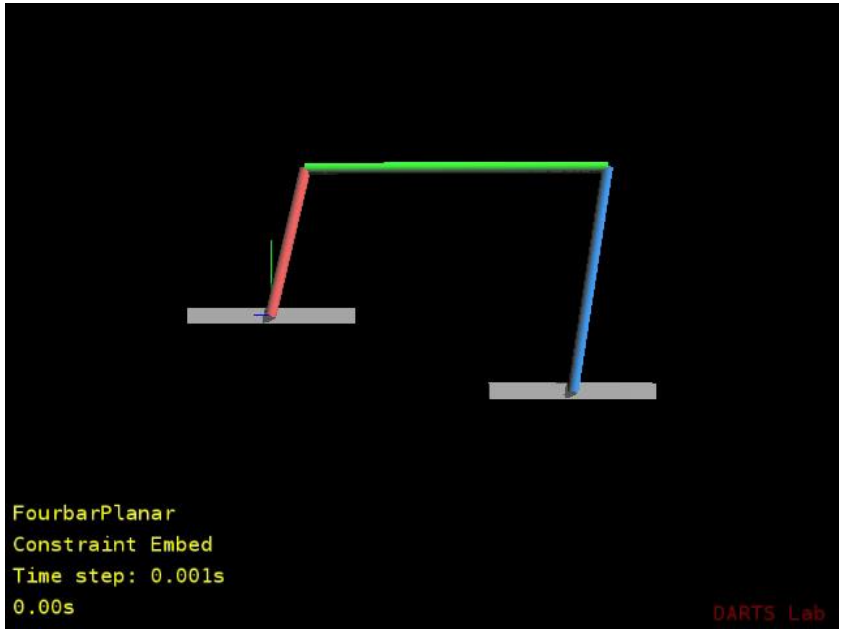


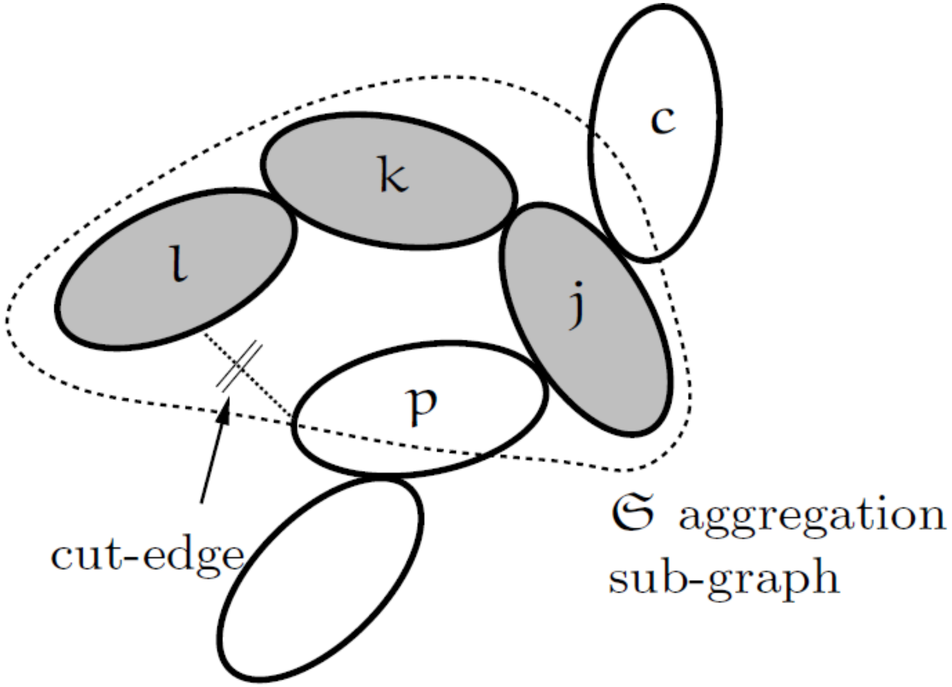

$$\dot{\theta}_{\mathfrak{G}} = \begin{bmatrix} \theta_m(k-1) \\ \theta_l(k-1) \end{bmatrix} \in \mathcal{R}^2, \qquad \dot{\theta}_{R\mathfrak{G}} = \theta_l(k-1) \in \mathcal{R}^1$$

$$X_{\mathfrak{G}} = \begin{bmatrix} \mu_G(k-1) \\ 1 \end{bmatrix} \in \mathcal{R}^{2\times 1}$$

$$\dot{X}_{\mathfrak{G}} = \mathbf{0}, \qquad \mathcal{E}_{\phi_{\mathfrak{G}}} = \mathbf{0} \in \mathcal{R}^{12\times 12}, \qquad \phi_{\mathfrak{G}} = \mathbf{I} \in \mathcal{R}^{12\times 12}$$

# Four-bar planar linkage



# 4-bar aggregation node – 3 bodies

$$\mathcal{E}_{\phi_{\mathfrak{S}}} = \begin{pmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \phi(j,k) & \mathbf{0} \end{pmatrix} \in \mathcal{R}^{18\times 18}$$

$$\phi_{\mathfrak{S}} = \begin{pmatrix} \mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \phi(j,k) & \mathbf{I} \end{pmatrix} \in \mathcal{R}^{18\times 18}, \qquad X_{\mathfrak{S}} = \begin{bmatrix} r_l \\ r_k \\ 1 \end{bmatrix} \in \mathcal{R}^{3\times 1}$$

$$E_{\mathfrak{S}} = \left[\phi(p,l),\ \mathbf{0},\ \phi(p,j)\right] \in \mathcal{R}^{6\times 18}, \quad \mathcal{B}_{\mathfrak{S}} = \begin{bmatrix} \mathbf{0} \\ \phi(j,c) \\ \mathbf{0} \end{bmatrix} \in \mathcal{R}^{18\times 6}$$

# Constraint Embedding SKO model algorithms

# ATBI recursion step for the aggregated body

*The only change to the ATBI recursion steps are around the aggregated body.*

$$\mathcal{P}^+(c) = \overline{\tau}(c)\mathcal{P}(c)$$

$$\mathcal{P}_{\mathfrak{S}} = \sum_{\forall c \in \mathbf{C}(\mathfrak{S})} \mathbb{A}(\mathfrak{S}, c)\mathcal{P}^+(c)\mathbb{A}^*(\mathfrak{S}, c) + M_{\mathfrak{S}}$$

$$\mathcal{D}_{\mathfrak{S}} = \underline{H}_{R\mathfrak{S}}\mathcal{P}_{\mathfrak{S}}\underline{H}^*_{R\mathfrak{S}}$$

$$\mathcal{G}_{\mathfrak{S}} = \mathcal{P}_{\mathfrak{S}}\underline{H}^*_{R\mathfrak{S}}\mathcal{D}^{-1}_{\mathfrak{S}}$$

$$\tau_{\mathfrak{S}} = \mathcal{G}_{\mathfrak{S}}\underline{H}_{R\mathfrak{S}}$$

$$\mathcal{P}^+_{\mathfrak{S}} = \mathcal{P}_{\mathfrak{S}} - \tau_{\mathfrak{S}}\mathcal{P}_{\mathfrak{S}}$$

$$\mathcal{P}(p) = \mathbb{A}(p, \mathfrak{S})\mathcal{P}^+_{\mathfrak{S}}\mathbb{A}^*(p, \mathfrak{S}) + M(p)$$

# Mass matrix inversion still holds

*After constraint embedding, we have minimal coordinates and an SKO model, and consequently the mass matrix factorization and inversion properties continue to hold.*

$$
\begin{aligned}
\mathcal{M}_r &\overset{16.8}{=} \mathrm{H}_{\mathfrak{R}\mathfrak{a}}\mathbb{A}_{\mathfrak{a}}\mathrm{M}\mathbb{A}_{\mathfrak{a}}^*\mathrm{H}_{\mathfrak{R}\mathfrak{a}}^* \\
&= [\mathbf{I} + \mathrm{H}_{\mathfrak{R}\mathfrak{a}}\mathbb{A}_{\mathfrak{a}}\mathcal{K}]\mathcal{D}[\mathbf{I} + \mathrm{H}_{\mathfrak{R}\mathfrak{a}}\mathbb{A}_{\mathfrak{a}}\mathcal{K}]^* \\
[\mathbf{I} + \mathrm{H}_{\mathfrak{R}\mathfrak{a}}\mathbb{A}_{\mathfrak{a}}\mathcal{K}]^{-1} &= \mathbf{I} - \mathrm{H}_{\mathfrak{R}\mathfrak{a}}\psi_{\mathfrak{a}}\mathcal{K} \\
\mathcal{M}_r^{-1} &= [\mathbf{I} - \mathrm{H}_{\mathfrak{R}\mathfrak{a}}\psi_{\mathfrak{a}}\mathcal{K}]^*\mathcal{D}^{-1}[\mathbf{I} - \mathrm{H}_{\mathfrak{R}\mathfrak{a}}\psi_{\mathfrak{a}}\mathcal{K}]
\end{aligned}
$$

# Recursive ATBI forward dynamics

$$\ddot{\theta}_{R} = [\mathbf{I} - H_{Ra}\psi_{a}\mathcal{K}]^{*}\,\mathcal{D}^{-1}\left[\mathcal{T}_{R} - H_{Ra}\psi_{a}(\mathcal{K}\mathcal{T}_{R} + \mathcal{P}\mathfrak{a}' + \mathfrak{b})\right] - \mathcal{K}^{*}\psi_{a}^{*}\mathfrak{a}'$$

- The algorithm is O(N) in the number of nodes
- However, steps involving compound bodies have higher computational cost proportional to the square of the number of aggregated bodies

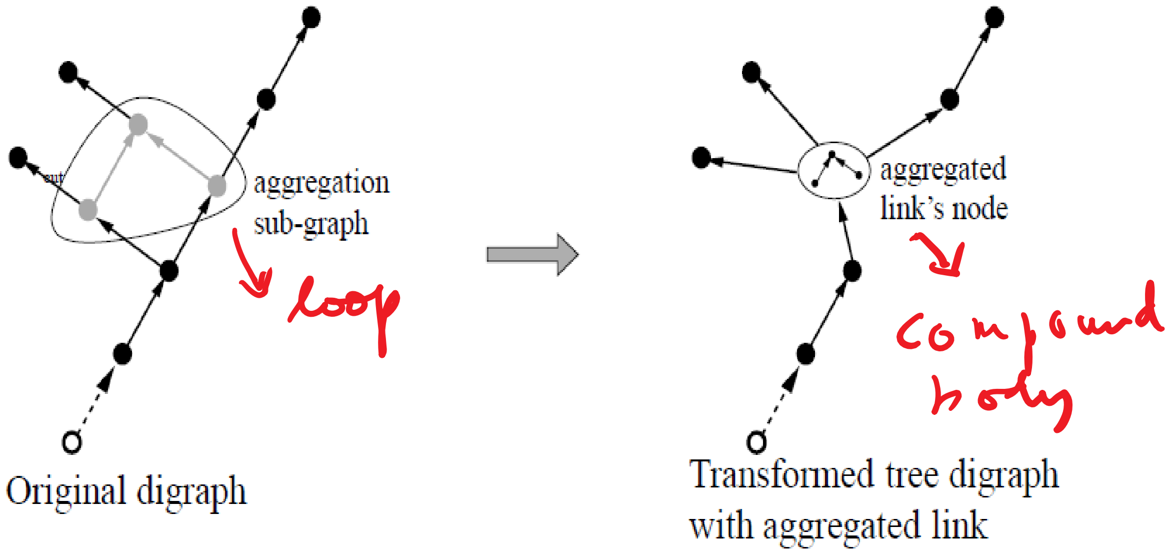
# Comments on CE approach

- The obvious tree structure is not available with loop constraints – so we had to work harder to do a graph transformation to obtain a tree needed for an SKO model
- With CE, the tree system has minimal coordinates, hence an ODE solver can be used
- We may think of CE is achieving the goals of the projection approach, but doing so while preserving structure
  - Hence all the benefits of a SKO model – analysis and algorithms – become available to even closed-loop systems
- Often we have analytical solution for the loop kinematics (eg. 4-bar case) which simplifies the determination of the X matrix and speeds up computations

# Constraint Embedding to the rescue

**_With closed-loop, no tree structure – paradise lost!_**

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

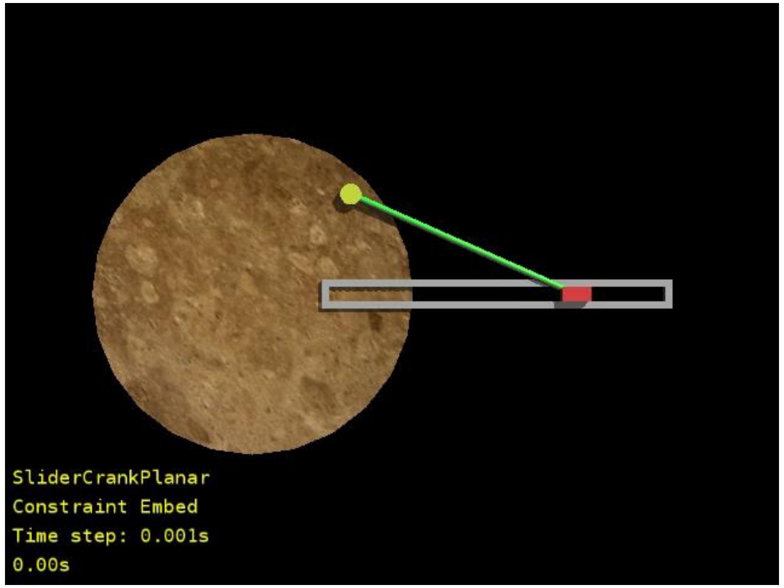


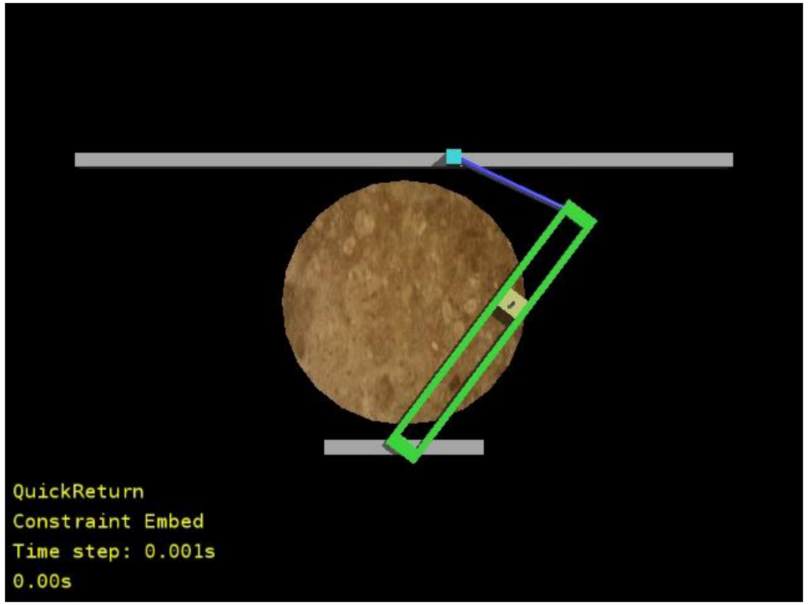

*The compound body is a **"variable geometry body"!***

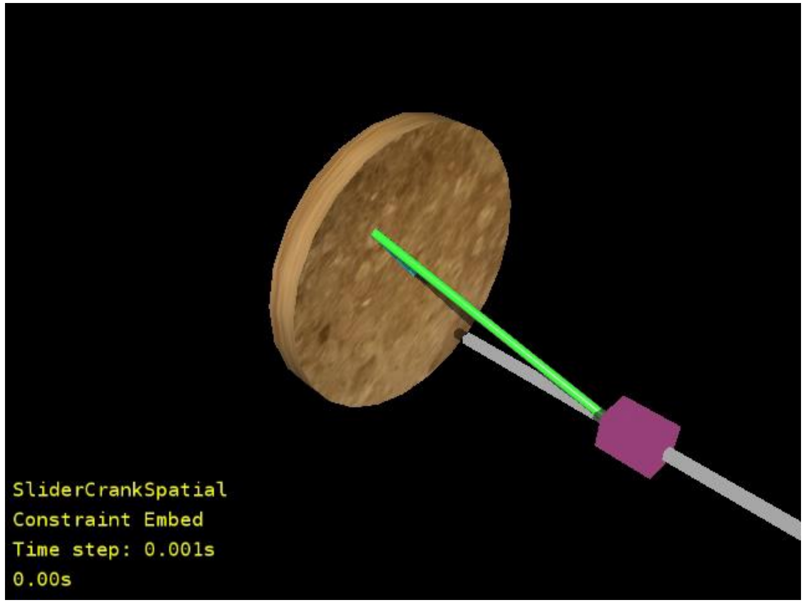
- *Minimal coordinate ODE model for a closed-loop system*
- *Tree analytical structure is regained*
- *Well defined non-singular mass matrix*
- *Mass matrix inversion results hold again*
- *Recursive O(N) methods are available again as well*

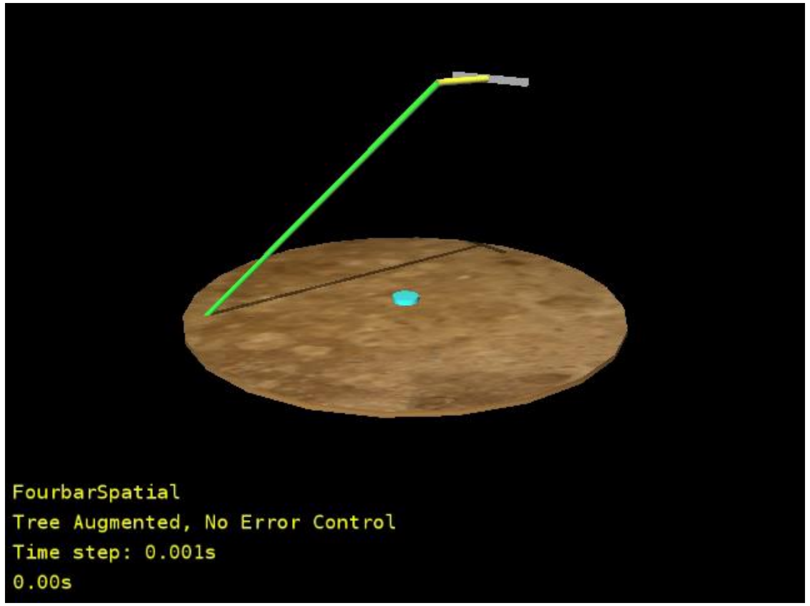
***Paradise regained!***

# Constraint Embedding Examples – Basic Mechanisms

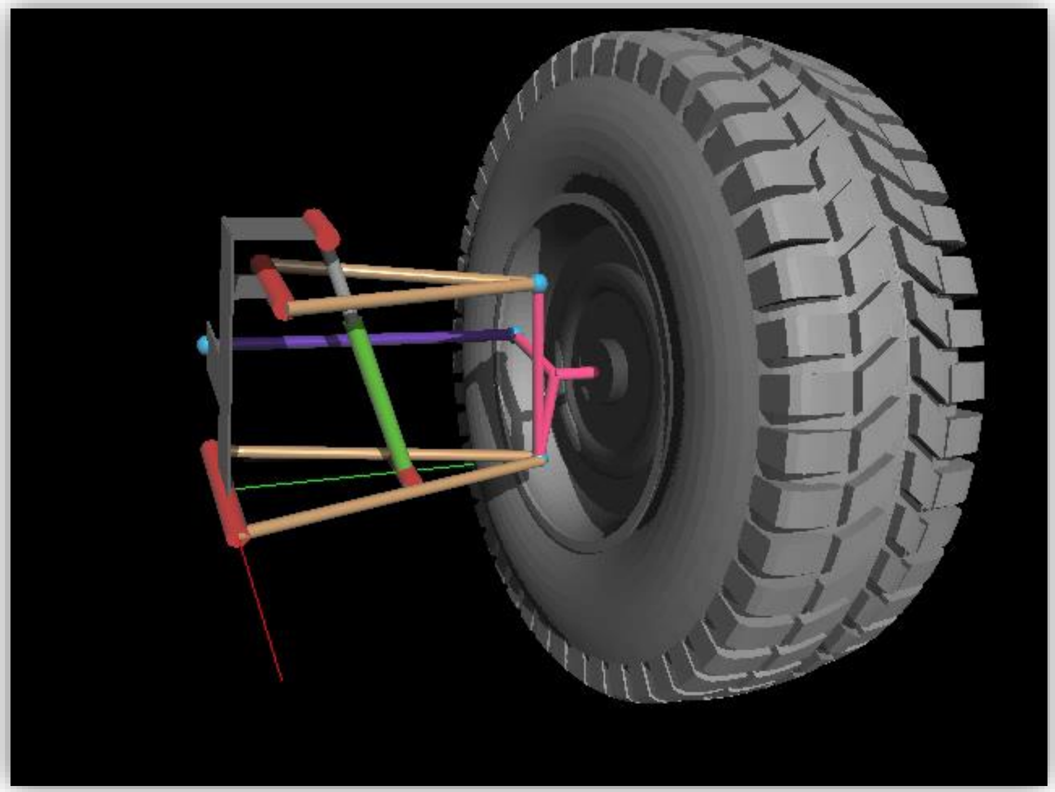


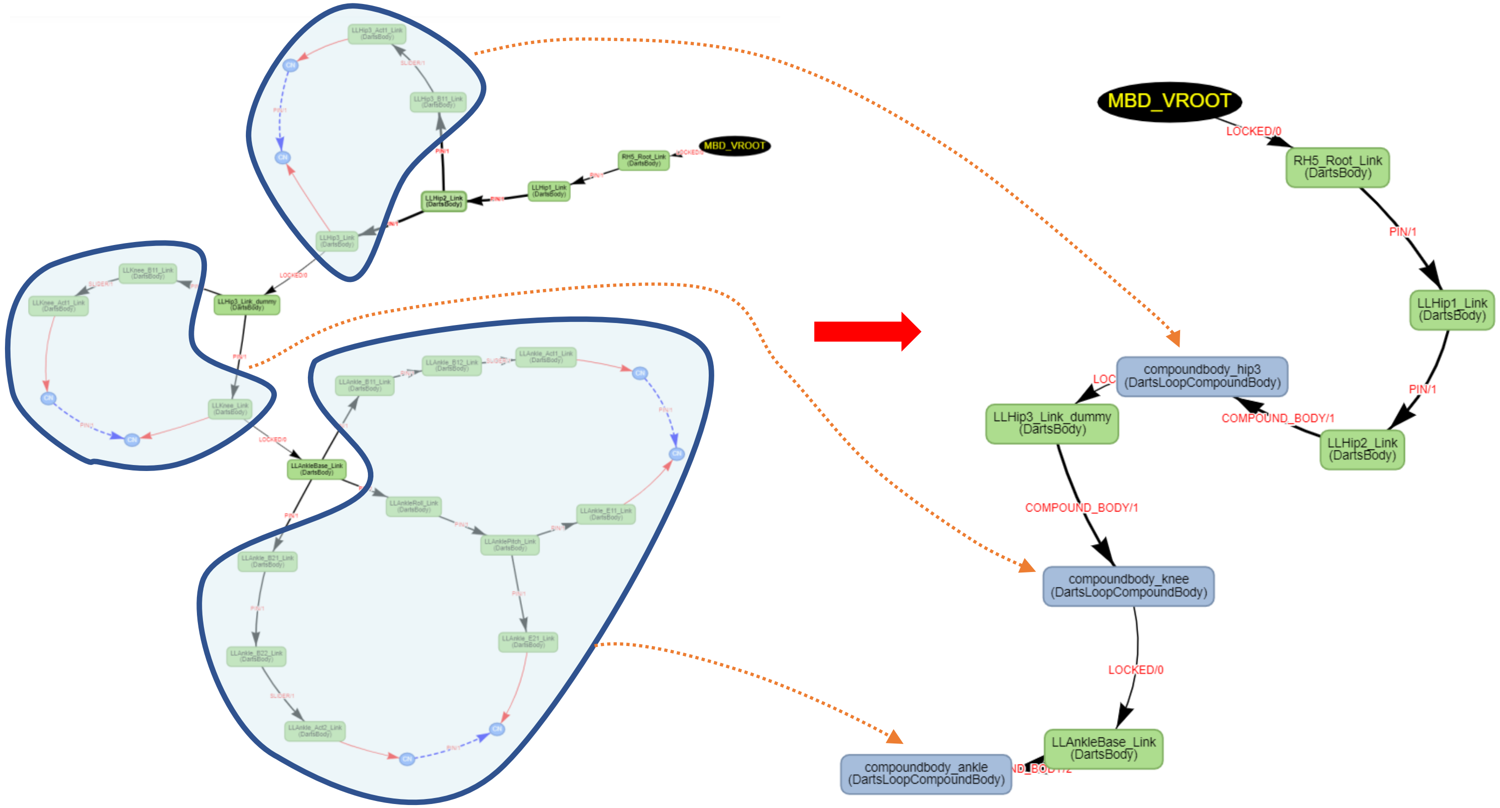



# Constraint Embedding Examples – Double wishbone suspension



*Multiple closed-loops*

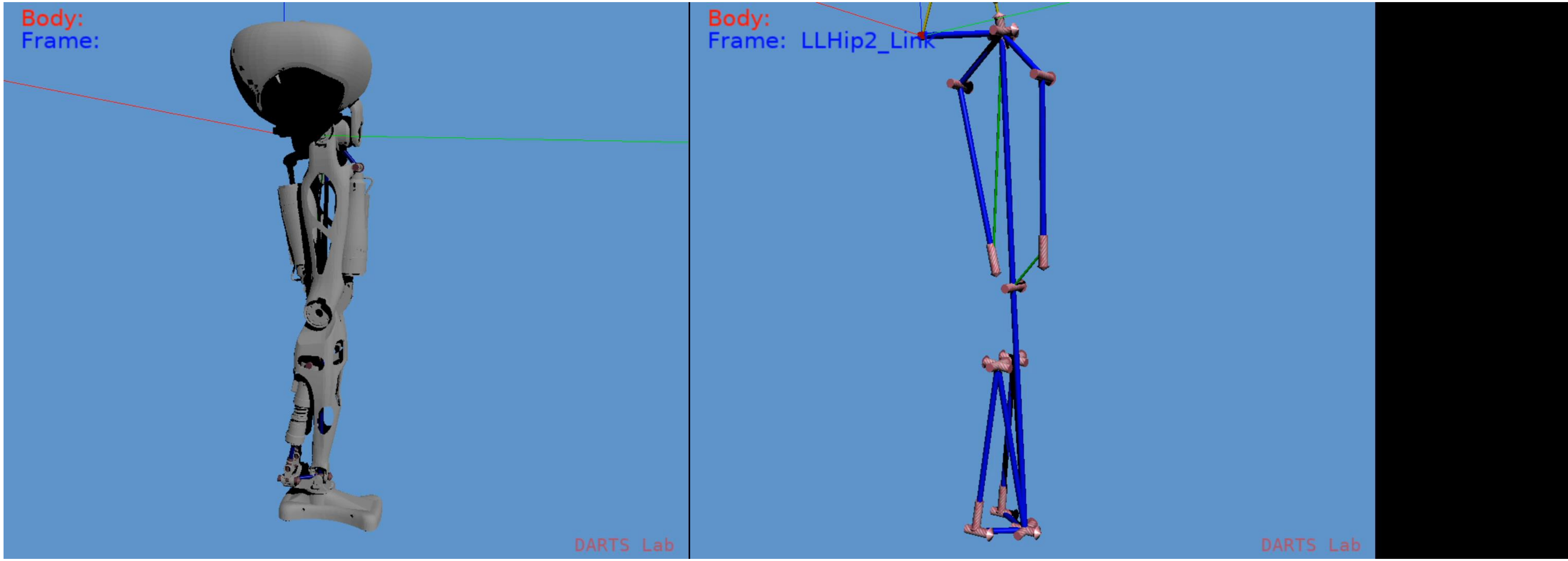
# Humanoid leg example




*Graph before constraint embedding*

*Graph after constraint embedding*

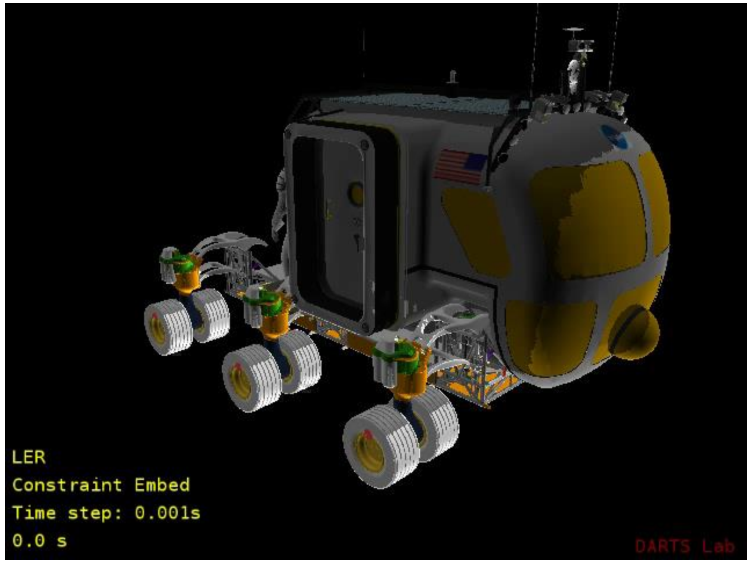
# Humanoid robot leg (multiple loops)




*Serial/parallel mechanism with 20 bodies, 18 unconstrained dofs, and 6 actual dofs*

# Performance comparisons for vehicle

Comparison of the speed and error performance of the CE and full/tree augmented approaches for a vehicle with single-wishbone suspensions dynamics model.




| Method | Tree dofs | Constraints size | Constraint error | Fidelity error | Normalized wall clock time |
|---|---|---|---|---|---|
| CE (analytic) | 24 | 0 | $1.3069e^{-31}$ | 0.00563109 | 1.0000 |
| CE (non-analytic) | 24 | 0 | $3.7367e^{-15}$ | 0.00563108 | 1.7296 |
| TA (no error control) | 36 | 12 | $5.2351e^{-13}$ | 0.00563094 | 1.6007 |
| TA w/ Baumgarte | 36 | 12 | $1.3466e^{-13}$ | 0.00563121 | 1.6362 |
| TA w/ projection | 36 | 12 | $5.2175e^{-13}$ | 0.00999659 | 1.5907 |
| FA (no error control) | 186 | 162 | $2.4183e^{-09}$ | 0.02183112 | 59.4236 |
| FA w/ Baumgarte | 186 | 162 | $5.7032e^{-11}$ | 0.01082618 | 59.4648 |
| FA w/ projection | 186 | 162 | Incomplete | - | - |

# Recap

# Summary

- Developed notions of graph partitioning
- Applied these to partitioning SKO models
- Defined conditions for partitioning to preserve tree structure
- Developed notion of subgraph aggregation
- Derived SKO model for aggregated graph
- Built constraint embedding idea on notion of subgraph aggregation
- Developed SKO model and algorithms for closed-loop systems using constraint embedding

# 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