View-Dependent Articulated-Body Simulation with Adaptive Forward Dynamics

Daseong Han

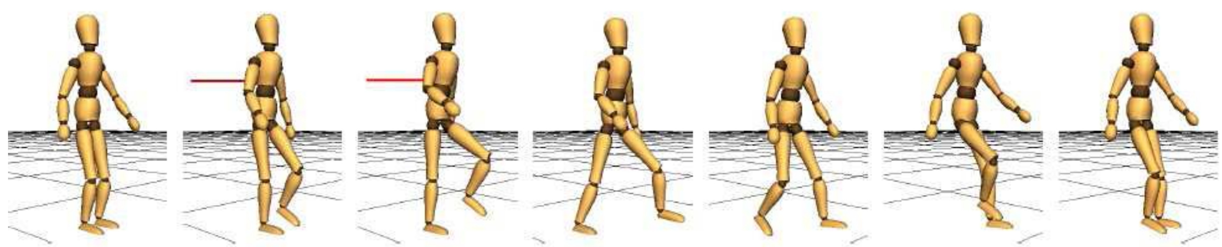
KAIST (Korea Advanced Institute of Science and Technology)

Outline

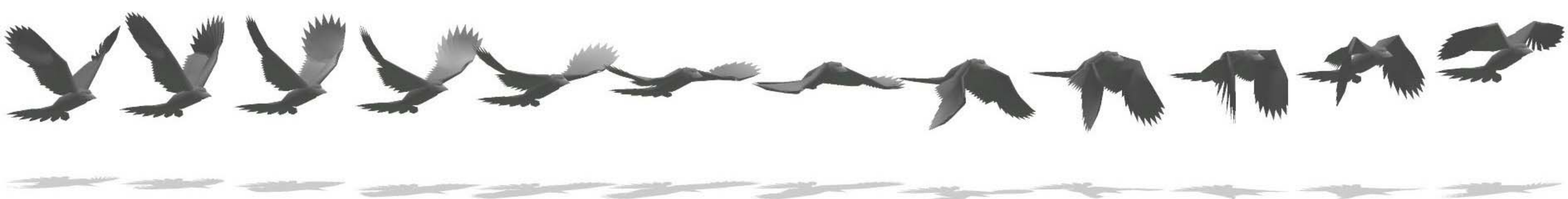
- Introduction
- Related Work
  - Simulation Levels of Detail(SLODs) for Real-time Animation
  - View-Dependent Culling Systems in Virtual Environments
  - Divide and Conquer Algorithm(DCA) For Forward Dynamics
  - Adaptive Forward Dynamics(AFD)
- View-dependent Forward Dynamics
- Conclusion

Introduction

- **Physically-based character animation**
  - **Support physical interaction between a character and its environment**
  - **Generate a variety of physically correct motions**



[SIMBICON: Simple Biped Locomotion Control, KangKang Yin et al., SIGGRAPH 2007]

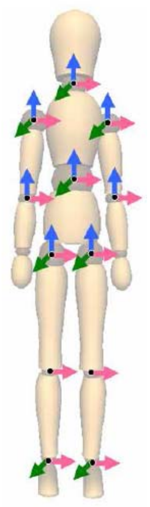


[Realistic Modeling of Bird Flight Animations, Jia-chi Wu, Zoran Popović, SIGGRAPH 2003]

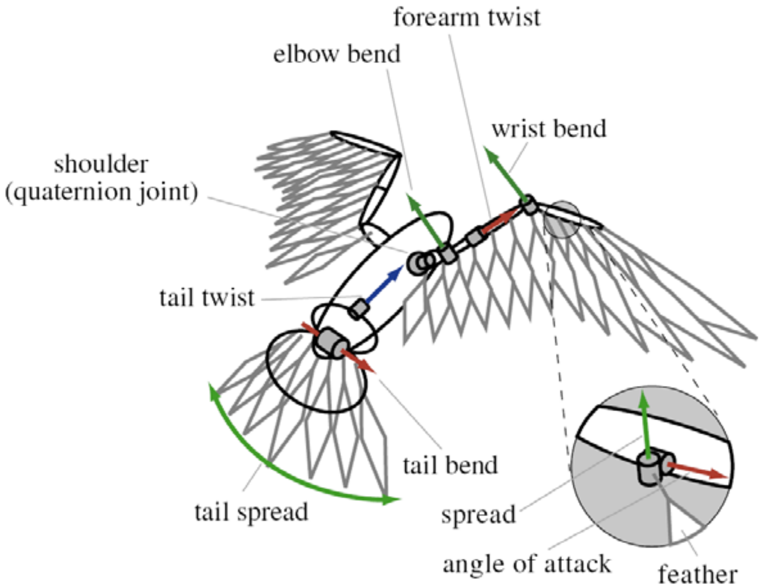
KAIST

Introduction

- **Physically-based character**
  - **Usually modeled as an articulated body**



[KangKang Y., `07]

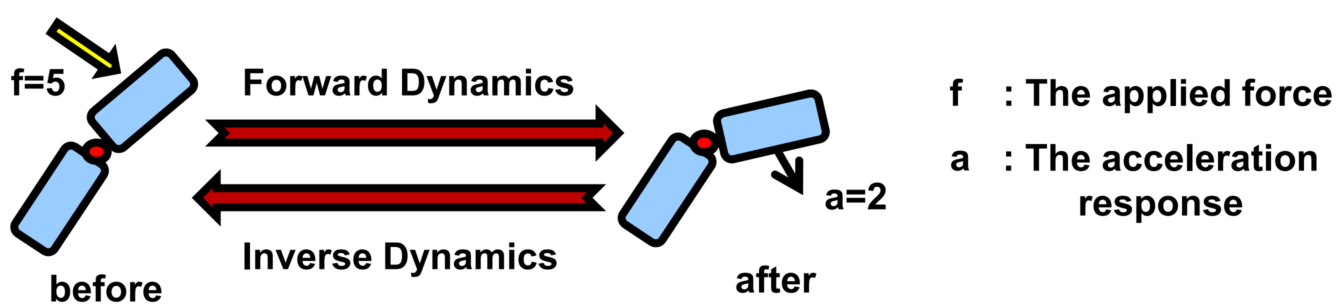


[Jia-chi W., `03]

Articulated body = Rigid bodies + Joints

Introduction

- **Simulation of the articulated-body dynamics**
  - **Forward Dynamics (force → accel.)**
    The calculation of the acceleration response to a given applied force
  - **Inverse Dynamics (accel. → force)**
    The calculation of the force that must be applied in order to produce a given acceleration response



Introduction

- **Optimal forward dynamics algorithms**
  - **Their time complexities are linear in the number of joints, but...**

[Continuum Crowds, Adrien T. at el., SIGGRAPH 2006]

Production constraints

"Dynamics computations should take less than 10-20 seconds per frame to make animators' lives easy"

Sunil Hadap, PDI/DreamWorks

We do need approximation techniques for large-scale dynamics simulations!

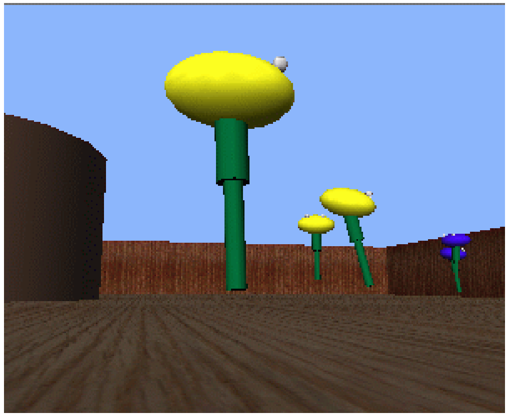
Related Work

- **Optimal Forward Dynamics**
  - **Divide and Conquer Algorithm(DCA) For Forward Dynamics [Roy F., ‘99]**
- **Approximation of Dynamics**
  - **Simulation Levels of Detail(SLODs) for Real-time Animation [Deborah C. and Jessica H., ‘97]**
  - **View-Dependent Culling Systems in Virtual Environments [Stephen C. and David F., ‘97]**
  - **Adaptive Forward Dynamics(AFD) [Stephane R. et al., ‘05]**

KAIST

SLODs for Real-time Animation

- **Simulation of simple hopping robots**
  - **Move around a rectangular court**
  - **Avoid collision with other objects**
- **Three SLODs models**
  - **Dynamic model**
    - **Fully simulated**
  - **Hybrid model**
    - **Kinematic(leg) + Dynamic(position)**
  - **Point-mass model**
    - **Only the position of the body simulated**
    - **No motion of the leg (invisible case)**



Recording of motions needed, Lower scalability to any other models

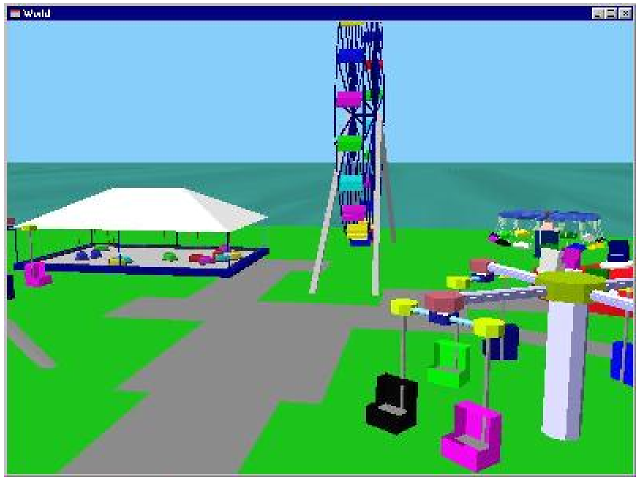
KAIST

View-Dependent Culling Systems

- Simulation of dynamic objects in a virtual amusement park
- Culling dynamics
  - Do not solve dynamic equations of objects that are not visible
- Two main problems
  - Consistency

    In what state should culled objects be when the view turns back?
  - Completeness

    Where and when will culled objects return to the view by themselves?

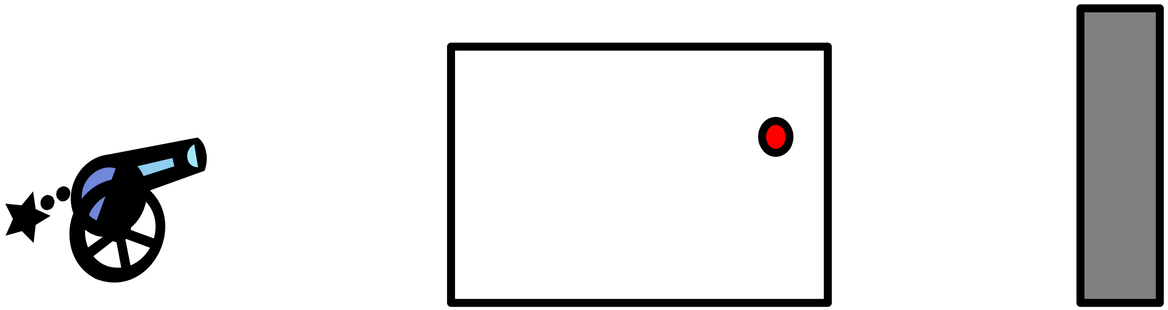


[pages.cs.wisc.edu/~schenney/research/culling]

View-Dependent Culling Systems

- **Consistency**

- **Completeness**



View-Dependent Culling Systems

- **Key observation**
  - **The user has a snapshot of the approximate initial conditions**
- **Approaches to consistency problem**
  - **Predict from the initial conditions the states of the systems to which the view turns back**

    → Apply a different statistical model according to behavioral properties of each dynamic system

Study of the behavioral properties needed,
Completeness problem not yet solved,
Lower scalability to any other models

DCA For Forward Dynamics

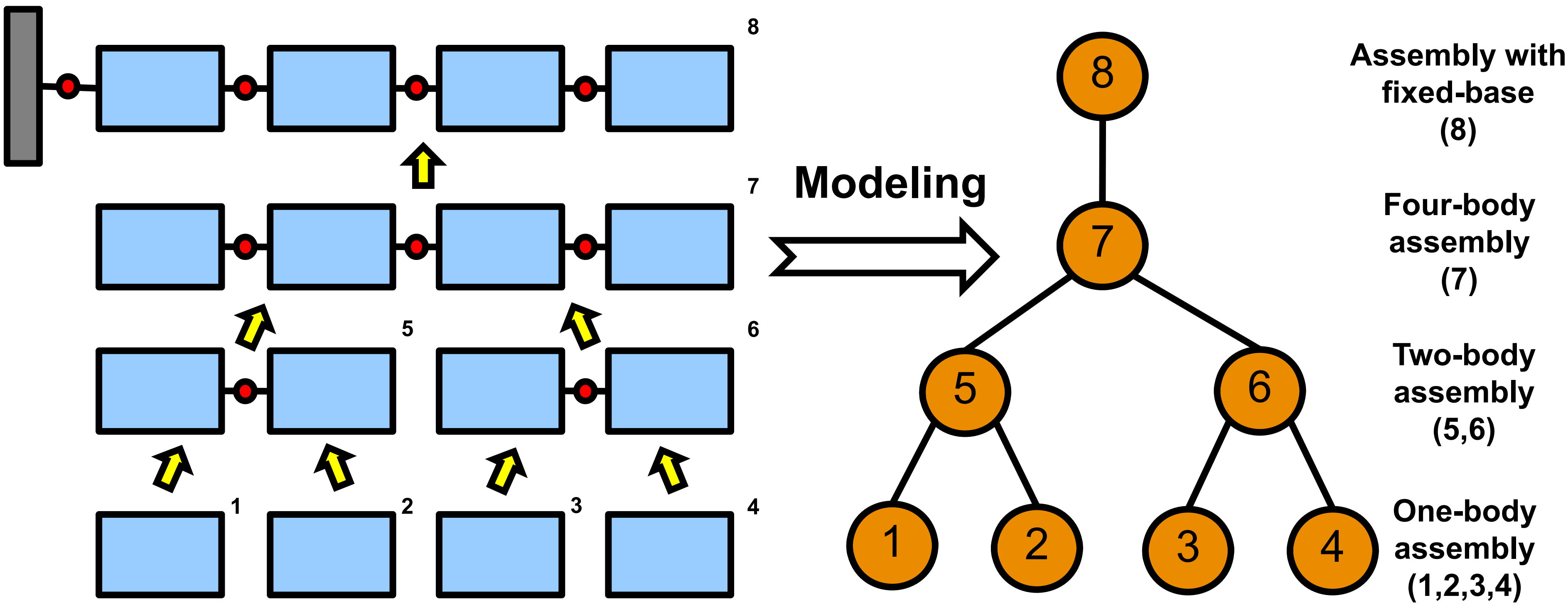
- One of the optimal solutions

- The spatial vector algebra used
  - Spatial vectors: 6D vectors that represent the linear and angular motions together
  - Simple and efficient to express the dynamic equations

- Divide and Conquer Algorithm (DCA)
  - Computations are recursively defined
  - Efficient parallel computing in multi-processor systems possible

DCA For Forward Dynamics

- Rigid body system model
  - Binary assembly tree
    - Combines two subassembly trees recursively

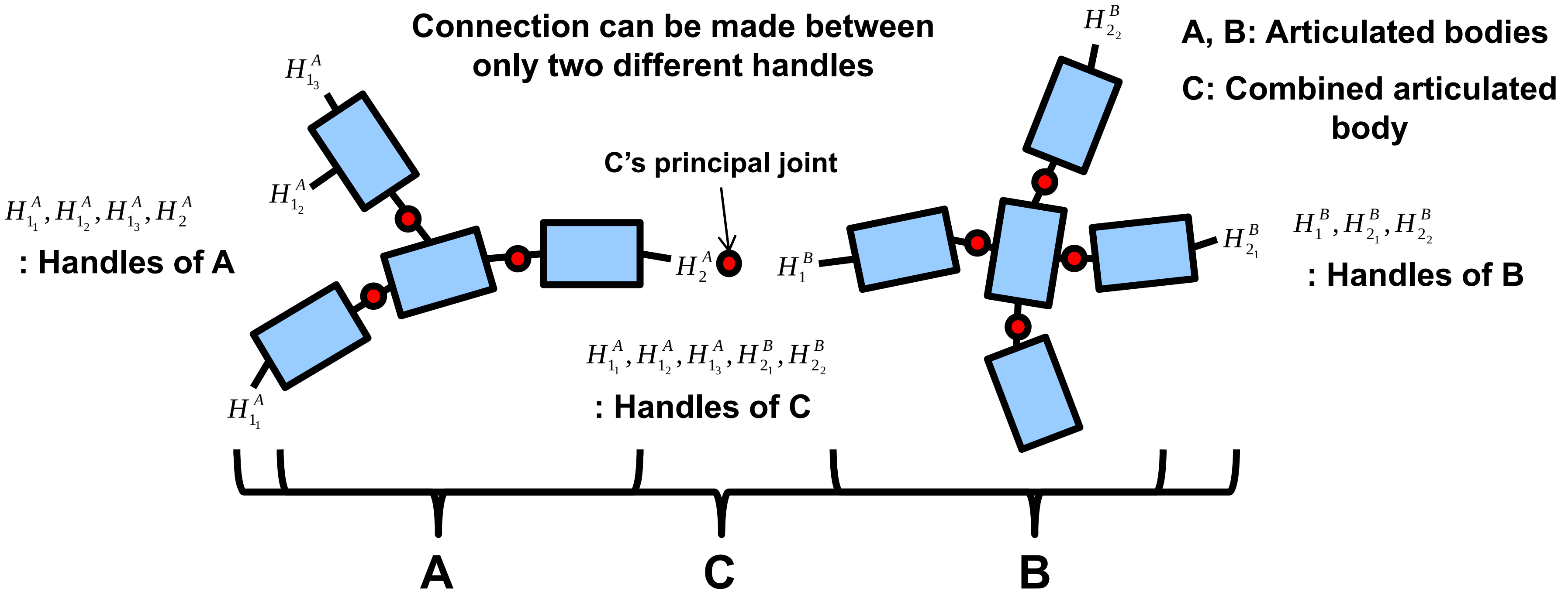


Rigid body system

Assembly Tree

DCA For Forward Dynamics

- **Handles**
  - **Specified locations within a rigid body system at which external forces may be applied**



DCA For Forward Dynamics

- **Articulated-body equation**

$$\begin{bmatrix} \mathbf{a}_1 \\ \mathbf{a}_2 \\ \vdots \\ \mathbf{a}_m \end{bmatrix} = \begin{bmatrix} \mathbf{\Phi}_1 & \mathbf{\Phi}_{12} & \cdots & \mathbf{\Phi}_{1m} \\ \mathbf{\Phi}_{21} & \mathbf{\Phi}_2 & \cdots & \mathbf{\Phi}_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{\Phi}_{m1} & \mathbf{\Phi}_{m2} & \cdots & \mathbf{\Phi}_m \end{bmatrix} \begin{bmatrix} \mathbf{f}_1 \\ \mathbf{f}_2 \\ \vdots \\ \mathbf{f}_m \end{bmatrix} + \begin{bmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \\ \vdots \\ \mathbf{b}_m \end{bmatrix}$$

Body accelerations — a_i : **the acceleration for handle *i***

Inverse inertia tensors — Φ_i : **the inverse inertia tensor for handle *i***

Φ_{ij} : **the inverse inertia tensor mapping a force applied to handle *j* to an acceleration at handle *i***

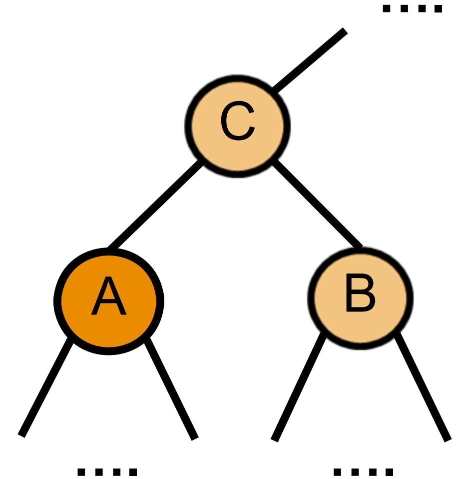
Applied external forces — f_i : **the force applied to handle *i***

Bias accelerations — b_i : **the bias acceleration for handle *i***

DCA For Forward Dynamics

- **Articulated-body equation**
  - **Body A with m+1 handles**

$$(H_{1_1}^A, H_{1_2}^A, \ldots, H_{1_m}^A, H_2^A)$$



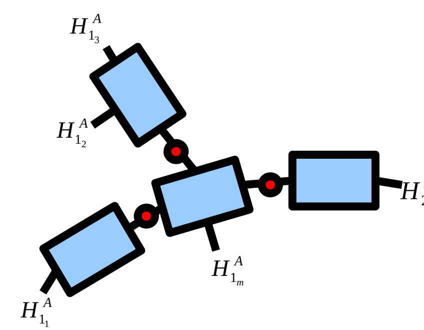
a_1^A Φ_1^A Φ_{12}^A f_1^A b_1^A

$$\begin{bmatrix} a_{1_1}^A \\ a_{1_2}^A \\ \vdots \\ a_{1_m}^A \\ a_2^A \end{bmatrix} = \begin{bmatrix} \Phi_{1_1}^A & \Phi_{1_1 1_2}^A & \cdots & \Phi_{1_1 1m}^A & \Phi_{1_1 2}^A \\ \Phi_{1_2 1_1}^A & \Phi_{1_2}^A & \cdots & \Phi_{1_2 1m}^A & \Phi_{1_2 2}^A \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \Phi_{1_m 1_1}^A & \Phi_{1_m 1_2}^A & \cdots & \Phi_{1_m}^A & \Phi_{1_m 2}^A \\ \Phi_{21_1}^A & \Phi_{21_2}^A & \cdots & \Phi_{21_m}^A & \Phi_2^A \end{bmatrix} \begin{bmatrix} f_{1_1}^A \\ f_{1_2}^A \\ \vdots \\ f_{1_m}^A \\ f_2^A \end{bmatrix} + \begin{bmatrix} b_{1_1}^A \\ b_{1_2}^A \\ \vdots \\ b_{1_m}^A \\ b_2^A \end{bmatrix}$$

Φ_{21}^A

Accelerations of body A — **Inverse inertias** — **Applied forces** — **Bias accelerations**

$$\Rightarrow \begin{bmatrix} a_1^A \\ a_2^A \end{bmatrix} = \begin{bmatrix} \Phi_1^A & \Phi_{12}^A \\ \Phi_{21}^A & \Phi_2^A \end{bmatrix} \begin{bmatrix} f_1^A \\ f_2^A \end{bmatrix} + \begin{bmatrix} b_1^A \\ b_2^A \end{bmatrix}$$



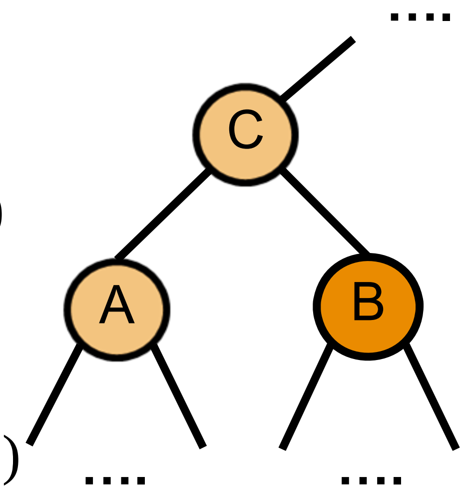
DCA For Forward Dynamics

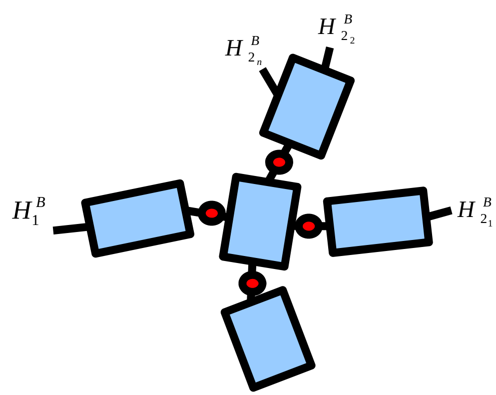
- **Articulated-body equation**
  - **Body A with m+1 handles**

$$\begin{bmatrix} a_1^A \\ a_2^A \end{bmatrix} = \begin{bmatrix} \Phi_1^A & \Phi_{12}^A \\ \Phi_{21}^A & \Phi_2^A \end{bmatrix} \begin{bmatrix} f_1^A \\ f_2^A \end{bmatrix} + \begin{bmatrix} b_1^A \\ b_2^A \end{bmatrix} \qquad (H_{1_1}^A, H_{1_2}^A, \ldots, H_{1_m}^A, H_2^A)$$

  - **Body B with n+1 handles**

$$\begin{bmatrix} a_1^B \\ a_2^B \end{bmatrix} = \begin{bmatrix} \Phi_1^B & \Phi_{12}^B \\ \Phi_{21}^B & \Phi_2^B \end{bmatrix} \begin{bmatrix} f_1^B \\ f_2^B \end{bmatrix} + \begin{bmatrix} b_1^B \\ b_2^B \end{bmatrix} \qquad (H_1^B, H_{2_1}^B, H_{2_2}^B, \ldots, H_{2_n}^B)$$





DCA For Forward Dynamics

- **Articulated-body equation**
  - **Body A with m+1 handles**

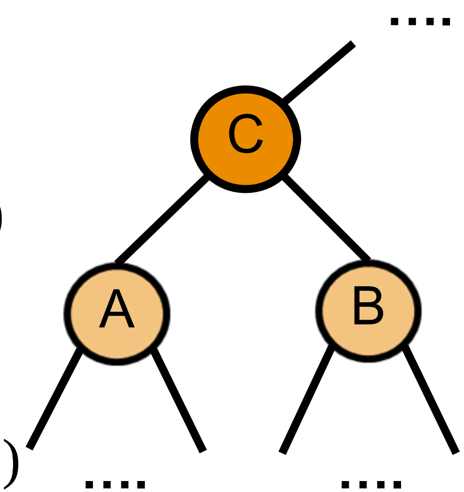
$$\begin{bmatrix} a_1^A \\ a_2^A \end{bmatrix} = \begin{bmatrix} \Phi_1^A & \Phi_{12}^A \\ \Phi_{21}^A & \Phi_2^A \end{bmatrix} \begin{bmatrix} f_1^A \\ f_2^A \end{bmatrix} + \begin{bmatrix} b_1^A \\ b_2^A \end{bmatrix} \qquad (H_{1_1}^A, H_{1_2}^A, \ldots, H_{1_m}^A, H_2^A)$$

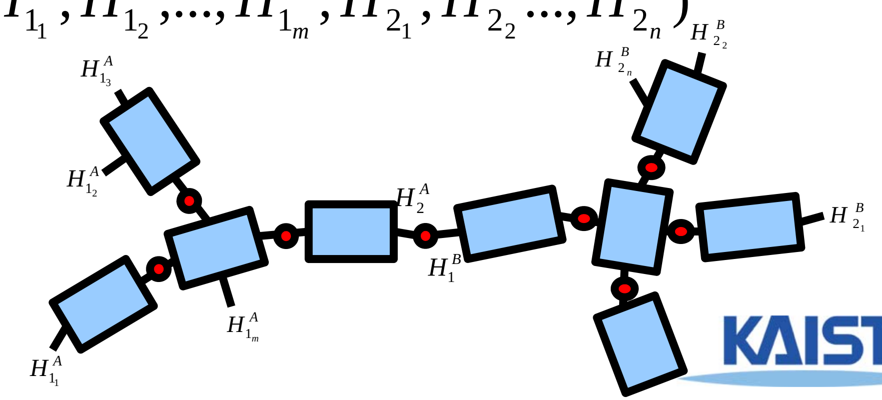
  - **Body B with n+1 handles**

$$\begin{bmatrix} a_1^B \\ a_2^B \end{bmatrix} = \begin{bmatrix} \Phi_1^B & \Phi_{12}^B \\ \Phi_{21}^B & \Phi_2^B \end{bmatrix} \begin{bmatrix} f_1^B \\ f_2^B \end{bmatrix} + \begin{bmatrix} b_1^B \\ b_2^B \end{bmatrix} \qquad (H_1^B, H_{2_1}^B, H_{2_2}^B, \ldots, H_{2_n}^B)$$

  - **Body C with m+n handles**

$$\begin{bmatrix} a_1^C \\ a_2^C \end{bmatrix} = \begin{bmatrix} \Phi_1^C & \Phi_{12}^C \\ \Phi_{21}^C & \Phi_2^C \end{bmatrix} \begin{bmatrix} f_1^A \\ f_2^B \end{bmatrix} + \begin{bmatrix} b_1^C \\ b_2^C \end{bmatrix} \qquad (H_{1_1}^A, H_{1_2}^A, \ldots, H_{1_m}^A, H_{2_1}^B, H_{2_2}^B \ldots, H_{2_n}^B)$$





KAIST

DCA For Forward Dynamics

- **Articulated-body equation**
  - **Body A with m+1 handles**

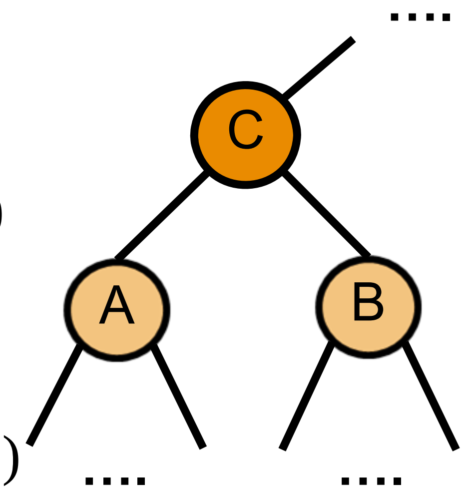
$$\begin{bmatrix} a_1^A \\ a_2^A \end{bmatrix} = \begin{bmatrix} \Phi_1^A & \Phi_{12}^A \\ \Phi_{21}^A & \Phi_2^A \end{bmatrix} \begin{bmatrix} f_1^A \\ f_2^A \end{bmatrix} + \begin{bmatrix} b_1^A \\ b_2^A \end{bmatrix} \qquad (H_{1_1}^A, H_{1_2}^A, \ldots, H_{1_m}^A, H_2^A)$$

  - **Body B with n+1 handles**

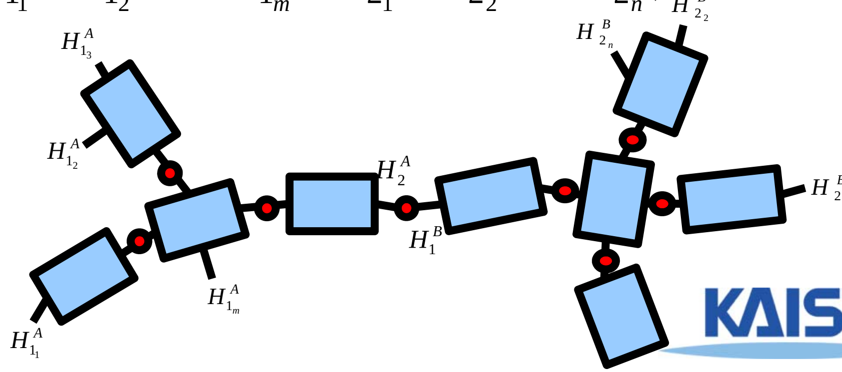
$$\begin{bmatrix} a_1^B \\ a_2^B \end{bmatrix} = \begin{bmatrix} \Phi_1^B & \Phi_{12}^B \\ \Phi_{21}^B & \Phi_2^B \end{bmatrix} \begin{bmatrix} f_1^B \\ f_2^B \end{bmatrix} + \begin{bmatrix} b_1^B \\ b_2^B \end{bmatrix} \qquad (H_1^B, H_{2_1}^B, H_{2_2}^B, \ldots, H_{2_n}^B)$$

  - **Body C with m+n handles**

$$\begin{bmatrix} a_1^C \\ a_2^C \end{bmatrix} = \begin{bmatrix} \Phi_1^C & \Phi_{12}^C \\ \Phi_{21}^C & \Phi_2^C \end{bmatrix} \begin{bmatrix} f_1^A \\ f_2^B \end{bmatrix} + \begin{bmatrix} b_1^C \\ b_2^C \end{bmatrix} \qquad (H_{1_1}^A, H_{1_2}^A, \ldots, H_{1_m}^A, H_{2_1}^B, H_{2_2}^B \ldots, H_{2_n}^B)$$



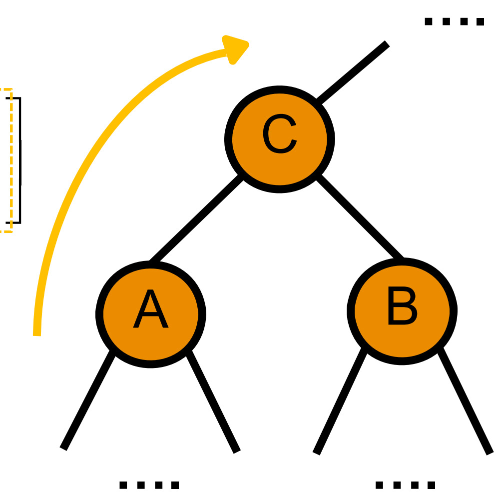
The coefficients of C can be expressed in terms of those of A and B
➔ Divide and Conquer Algorithm (DCA) is possible!!



DCA For Forward Dynamics

- **Two main passes**
  - **Main pass (↑)**

$$\begin{bmatrix} a_1^C \\ a_2^C \end{bmatrix} = \begin{bmatrix} \Phi_1^C & \Phi_{12}^C \\ \Phi_{21}^C & \Phi_2^C \end{bmatrix} \begin{bmatrix} f_1^A \\ f_2^B \end{bmatrix} + \begin{bmatrix} b_1^C \\ b_2^C \end{bmatrix}$$



Inverse inertias

$$\Phi_1^C = \Phi_1^A - \Phi_{12}^A W \Phi_{21}^A$$
$$\Phi_2^C = \Phi_2^B - \Phi_{21}^B W \Phi_{12}^B$$
$$\Phi_{21}^C = \Phi_{21}^B W \Phi_{21}^A$$
$$\Phi_{12}^C = (\Phi_{21}^C)^T$$

Bias accelerations

$$b_1^C = b_1^A - \Phi_{12}^A \gamma$$
$$b_2^C = b_2^A + \Phi_{21}^A \gamma$$

, where

$$V = (\Phi_2^A + \Phi_1^B)^{-1}$$
$$W = V - VS(S^T VS)^{-1} S^T V$$
$$\beta = b_2^A - b_1^A + \dot{S}\dot{q}_0$$
$$\gamma = W\beta + VS(S^T VS)^{-1} Q$$

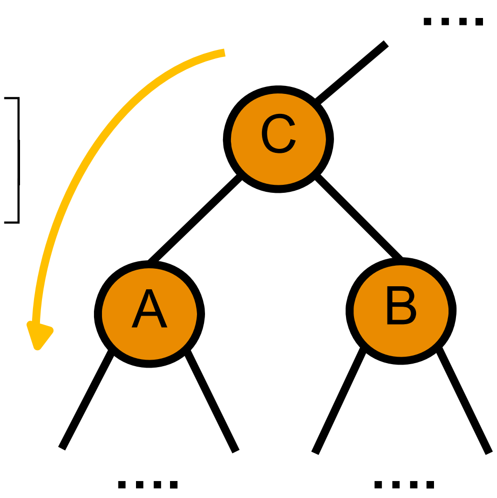
S :Joint motion space
Q : Forces applied by joint actuators
$\dot{q}_0$: Velocity of the principal joint

  - **Back-substitution pass (↓)**

DCA For Forward Dynamics

- **Two main passes**
  - **Main pass (↑)**

$$\begin{bmatrix} a_1^C \\ a_2^C \end{bmatrix} = \begin{bmatrix} \Phi_1^C & \Phi_{12}^C \\ \Phi_{21}^C & \Phi_2^C \end{bmatrix} \begin{bmatrix} f_1^A \\ f_2^B \end{bmatrix} + \begin{bmatrix} b_1^C \\ b_2^C \end{bmatrix}$$



Inverse inertias

$$\Phi_1^C = \Phi_1^A - \Phi_{12}^A W \Phi_{21}^A$$
$$\Phi_2^C = \Phi_2^B - \Phi_{21}^B W \Phi_{12}^B$$
$$\Phi_{21}^C = \Phi_{21}^B W \Phi_{21}^A$$
$$\Phi_{12}^C = (\Phi_{21}^C)^T$$

bias accelerations

$$b_1^C = b_1^A - \Phi_{12}^A \gamma$$
$$b_2^C = b_2^A + \Phi_{21}^A \gamma$$

  - **Back-substitution pass (↓)**

Joint acceleration

$$\ddot{q}_0 = (S^T V S)^{-1}(Q - S^T V(\Phi_{21}^A f_1^A - \Phi_{12}^B f_2^B + \beta))$$

Kinematic constraint forces

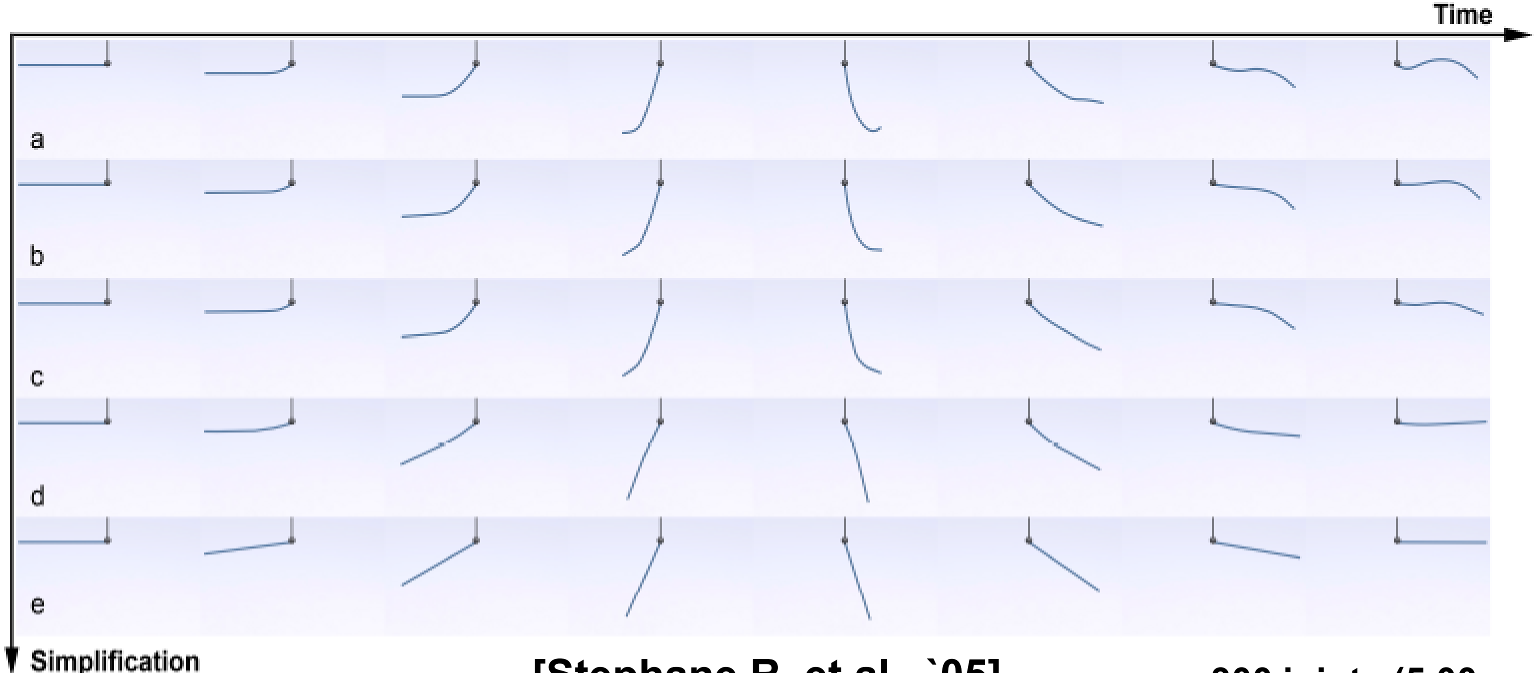
$$f_1^B = W\Phi_{21}^A f_1^A - W\Phi_{12}^B f_2^B + \gamma$$
$$f_2^A = -f_1^B$$

$$\begin{bmatrix} a_1^A \\ a_2^A \end{bmatrix} = \begin{bmatrix} \Phi_1^A & \Phi_{12}^A \\ \Phi_{21}^A & \Phi_2^A \end{bmatrix} \begin{bmatrix} f_1^A \\ \boxed{f_2^A} \end{bmatrix} + \begin{bmatrix} b_1^A \\ b_2^A \end{bmatrix}$$

$$\begin{bmatrix} a_1^B \\ a_2^B \end{bmatrix} = \begin{bmatrix} \Phi_1^B & \Phi_{12}^B \\ \Phi_{21}^B & \Phi_2^B \end{bmatrix} \begin{bmatrix} \boxed{f_1^B} \\ f_2^B \end{bmatrix} + \begin{bmatrix} b_1^B \\ b_2^B \end{bmatrix}$$

KAIST

Adaptive Forward Dynamics



[Stephane R. et al., `05]

a: 300 joints (5.00 ms)

b: 100 joints (1.70 ms)

c: 50 joints (0.70 ms)

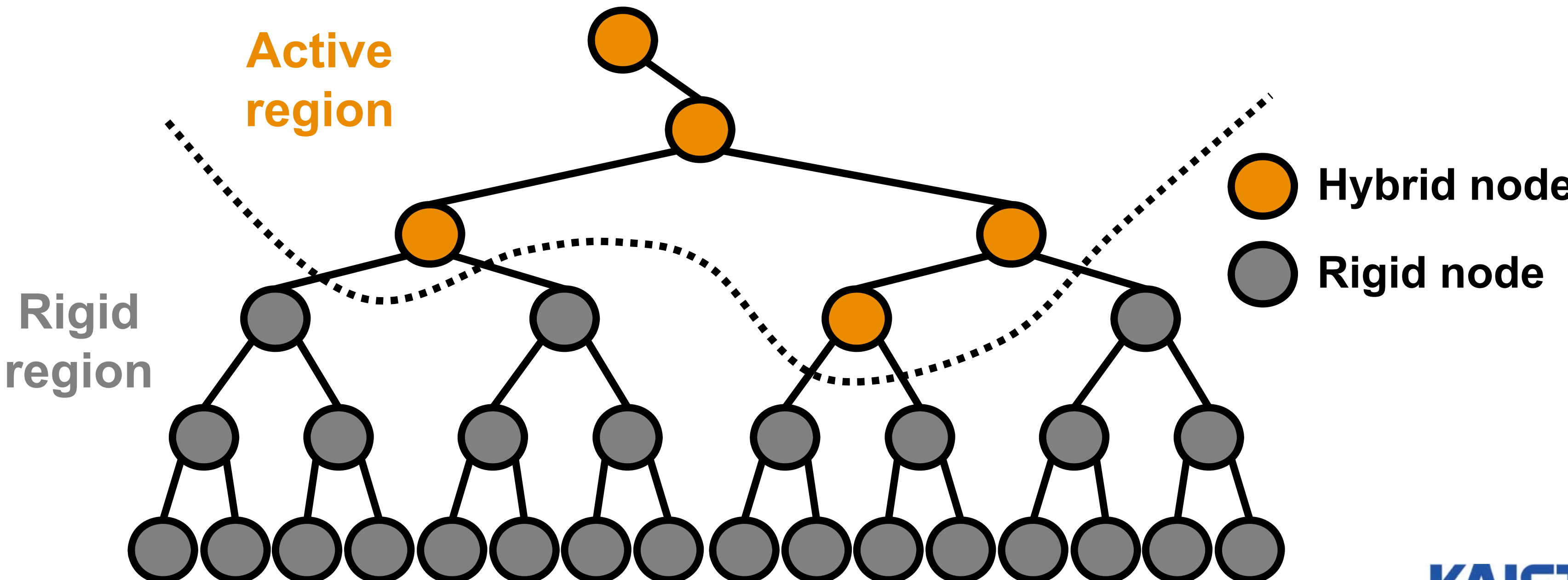
d: 20 joints (0.25 ms)

e: 1 joints (0.02 ms)

We don't need to simulate all of the joints of an articulated body to keep its plausible dynamics!!

Adaptive Forward Dynamics

- **Hybrid bodies**
  - **Hybrid node**: its principal joint is active (simulated) but some descendents are rigid
  - **Rigid node**: its principal joint is inactive and all the descendents are rigid / it is a leaf

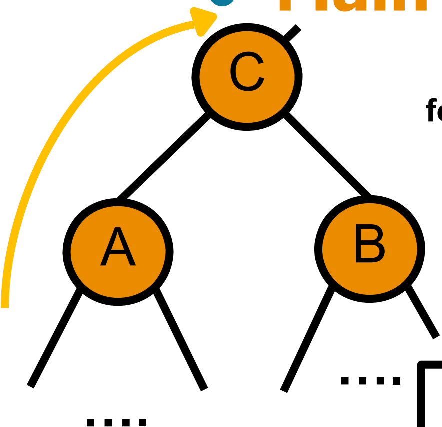


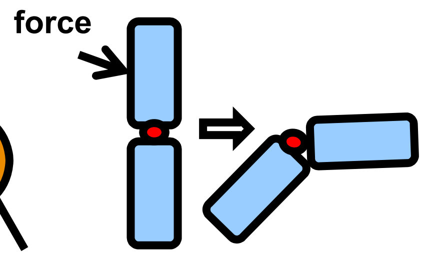
KAIST

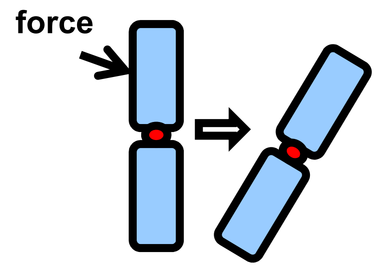
Adaptive Forward Dynamics

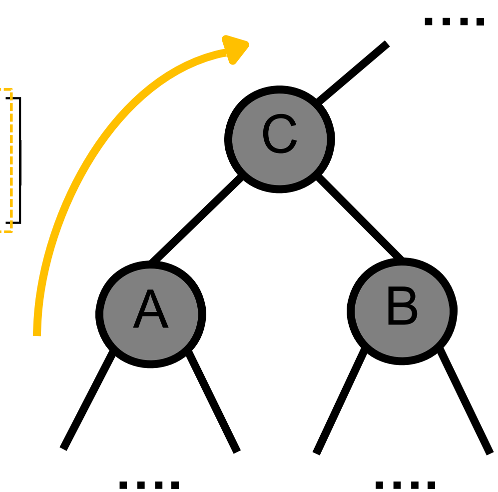
- **Hybrid-body dynamics**
  - **Main pass (↑)**

$$\begin{bmatrix} a_1^C \\ a_2^C \end{bmatrix} = \begin{bmatrix} \Phi_1^C & \Phi_{12}^C \\ \Phi_{21}^C & \Phi_2^C \end{bmatrix} \begin{bmatrix} f_1^A \\ f_2^B \end{bmatrix} + \begin{bmatrix} b_1^C \\ b_2^C \end{bmatrix}$$









Inverse inertias

$$\Phi_1^C = \Phi_1^A - \Phi_{12}^A W \Phi_{21}^A$$
$$\Phi_2^C = \Phi_2^B - \Phi_{21}^B W \Phi_{12}^B$$
$$\Phi_{21}^C = \Phi_{21}^B W \Phi_{21}^A$$
$$\Phi_{12}^C = (\Phi_{21}^C)^T$$

Bias accelerations

$$b_1^C = b_1^A - \Phi_{12}^A \gamma$$
$$b_2^C = b_2^A + \Phi_{21}^A \gamma$$

Rigidification ⇒

Rigidify the principal joint of C

$$\Phi_i^C = \Phi_{ij}^C = \Phi^C$$
$$\Phi^C = \Phi^B (\Phi^A + \Phi^B)^{-1} \Phi^A$$

$$b_i^C = b_{ij}^C = b^C$$
$$b^C = b^A - \Phi^A (\Phi^A + \Phi^B)^{-1} (b^A - b^B)$$

C can be rigidified only if C's principal joint is inactive and both its children A and B are rigid nodes

KAIST

Adaptive Forward Dynamics

- **Adaptive joint selection**
  - **Motion metrics**
    - **Acceleration metric**

$$\mathscr{A}(C) = \begin{bmatrix} \mathbf{f}_1^A \\ \mathbf{f}_2^B \end{bmatrix}^T \begin{bmatrix} \mathbf{\Psi}_1^C & \mathbf{\Psi}_{12}^C \\ \mathbf{\Psi}_{21}^C & \mathbf{\Psi}_2^C \end{bmatrix} \begin{bmatrix} \mathbf{f}_1^A \\ \mathbf{f}_2^B \end{bmatrix} + \begin{bmatrix} \mathbf{f}_1^A \\ \mathbf{f}_2^B \end{bmatrix}^T \begin{bmatrix} \mathbf{p}_1^C \\ \mathbf{p}_2^C \end{bmatrix} + \eta^C,$$

    - **Velocity metric**

$$\mathscr{V}(C) = \sum_{i \in C} \dot{\mathbf{q}}_i^T \mathbf{V}_i \dot{\mathbf{q}}_i$$

$\Psi_i^C, \Psi_{ij}^C, p_i^C, \eta^C$

: Acceleration metric coefficients

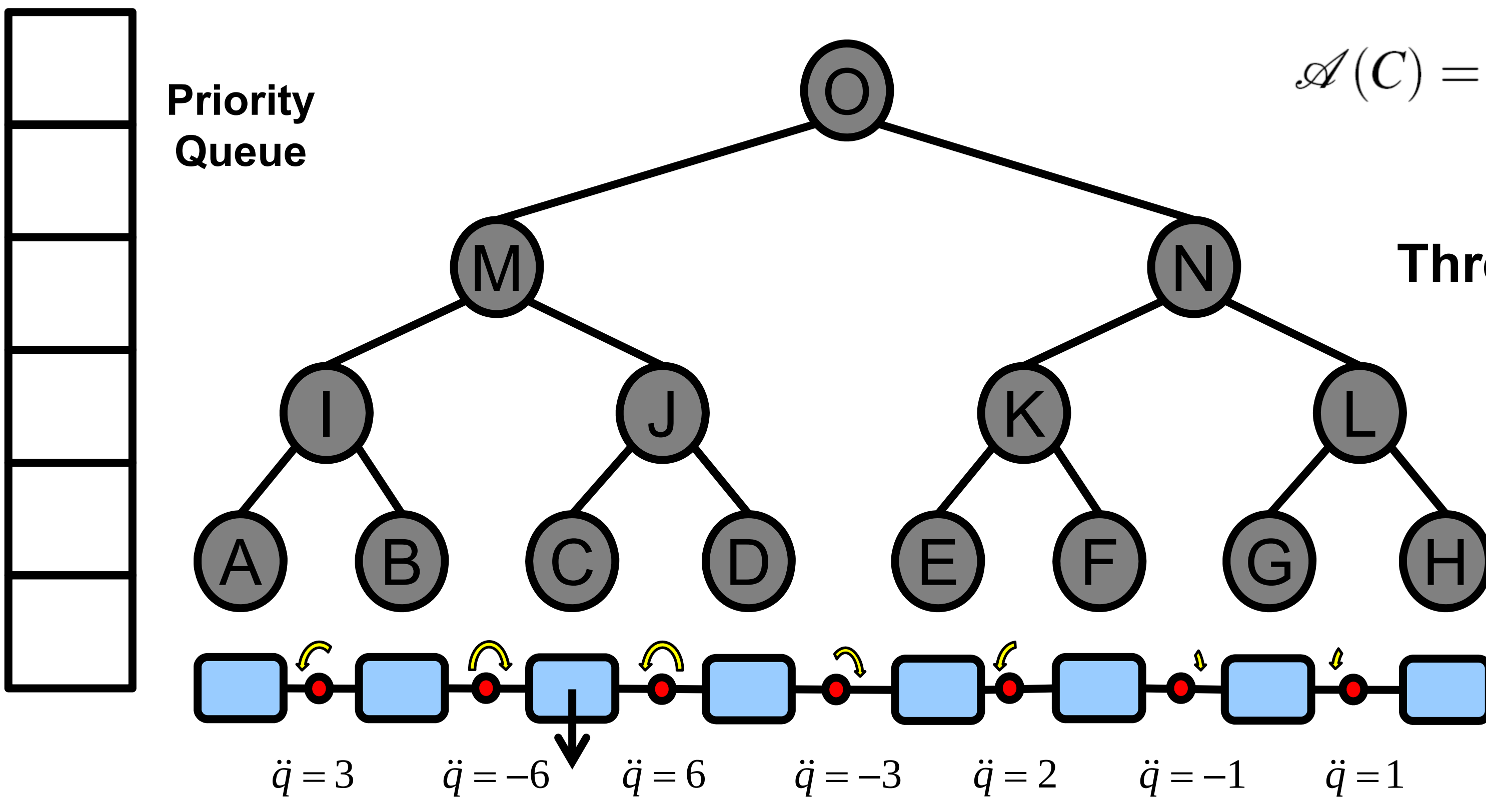
C	**: Articulated body**
$\dot{q}_i$	**: Velocity of joint *I***
$\ddot{q}_i$	**: Acceleration of joint *i***
A_i, V_i	**: Customizable weight matrices of joint *i***

The acceleration metric value of an articulated body can be computed <u>before</u> computing its joint accelerations!!

KAIST

Adaptive Forward Dynamics

- **Adaptive joint selection**
  - **Example: Back-substitution pass (↓)**



$$\mathscr{A}(C) = \sum_{i \in C} \ddot{\mathbf{q}}_i^T \mathbf{A}_i \ddot{\mathbf{q}}_i$$

$$A_i = I$$

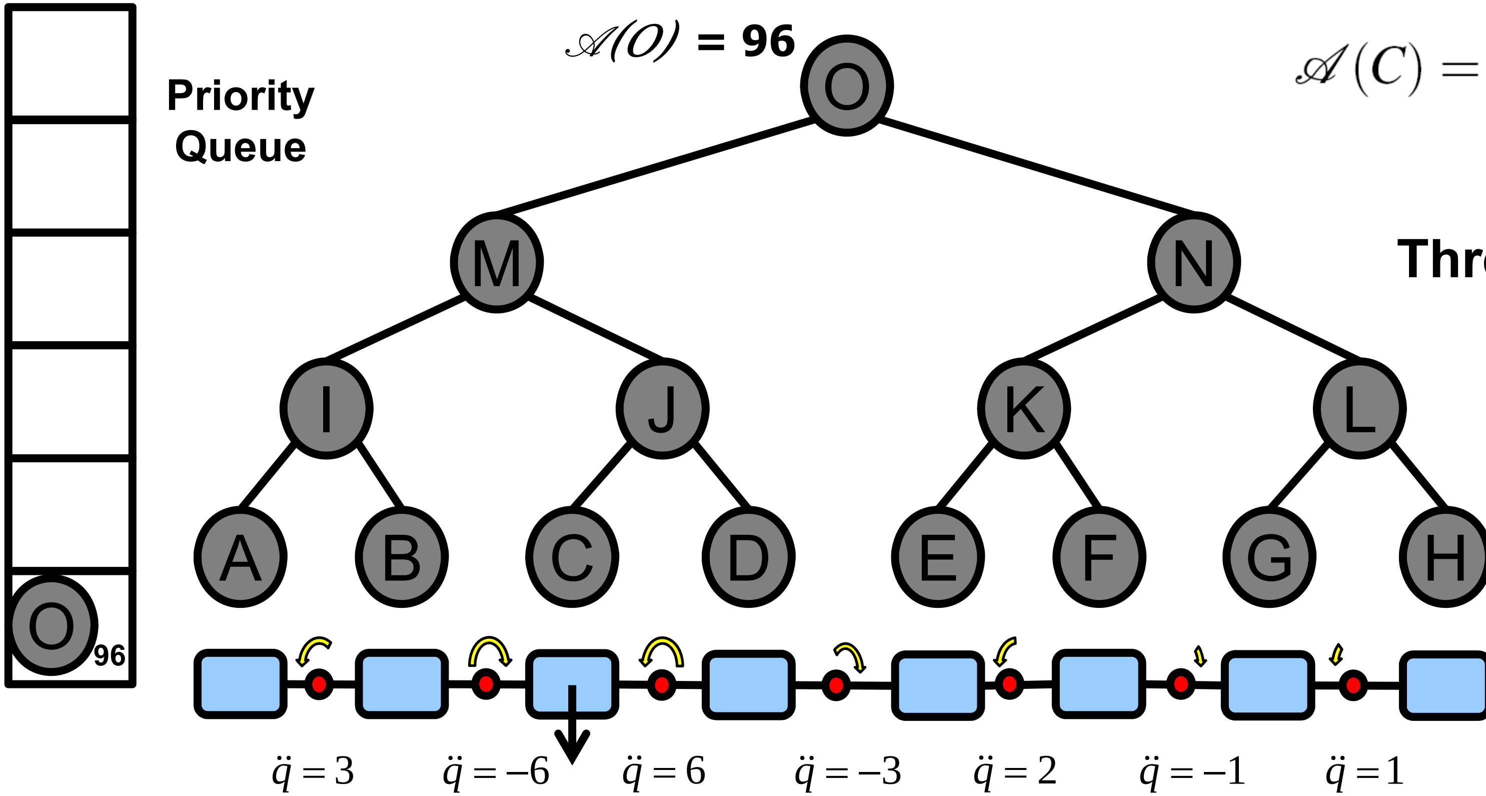
Threshold: 3

The # of simulated joints: 0

KAIST

Adaptive Forward Dynamics

- **Adaptive joint selection**
  - **Example: Back-substitution pass (↓)**



$$\mathscr{A}(C) = \sum_{i \in C} \ddot{\mathbf{q}}_i^T \mathbf{A}_i \ddot{\mathbf{q}}_i$$

$$A_i = I$$

Threshold: 3

The # of simulated joints: 0

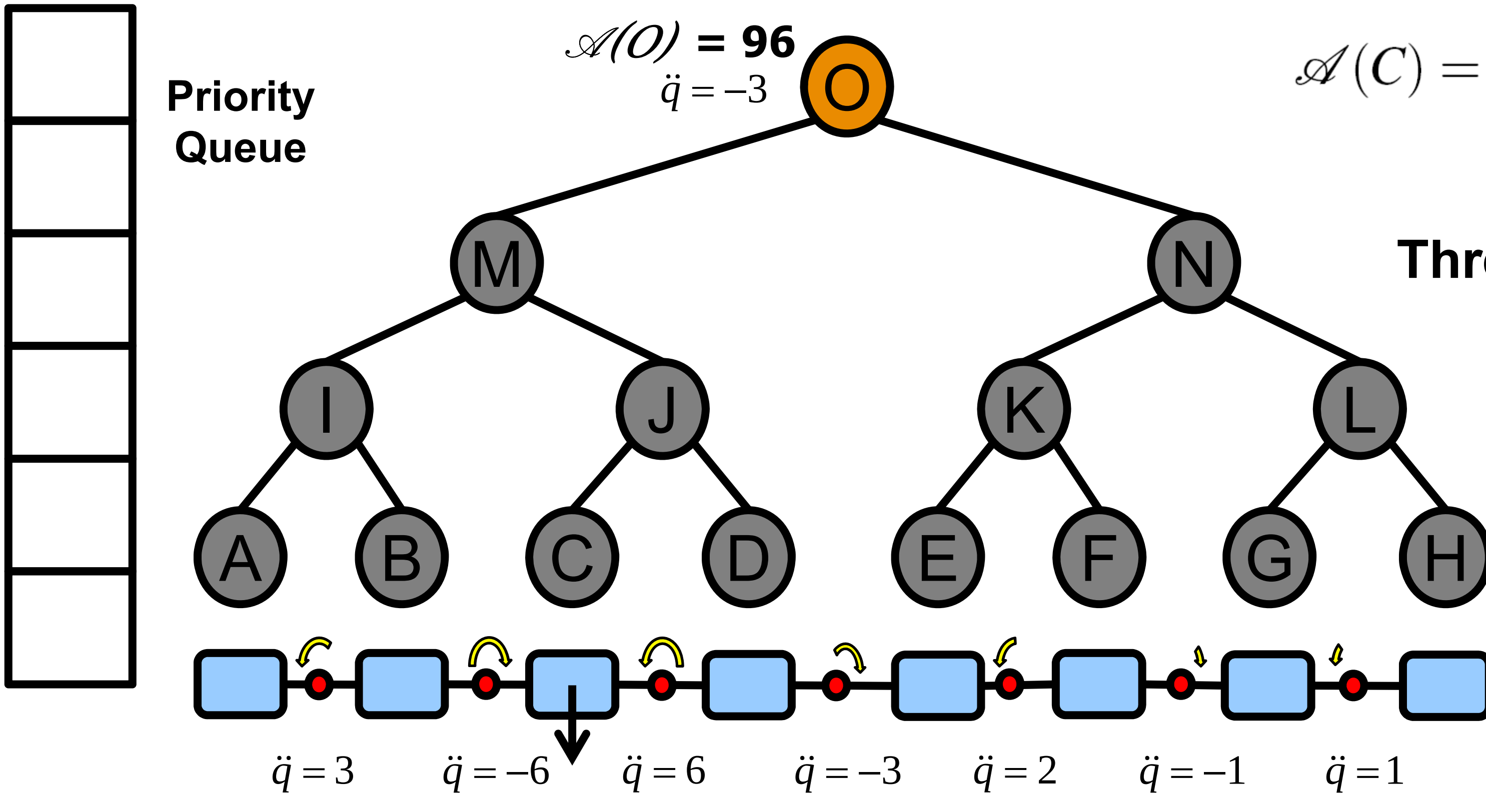
Adaptive Forward Dynamics

- **Adaptive joint selection**
  - **Example: Back-substitution pass (↓)**

$$\mathscr{A}(C) = \sum_{i \in C} \ddot{\mathbf{q}}_i^T \mathbf{A}_i \ddot{\mathbf{q}}_i$$

$$A_i = I$$

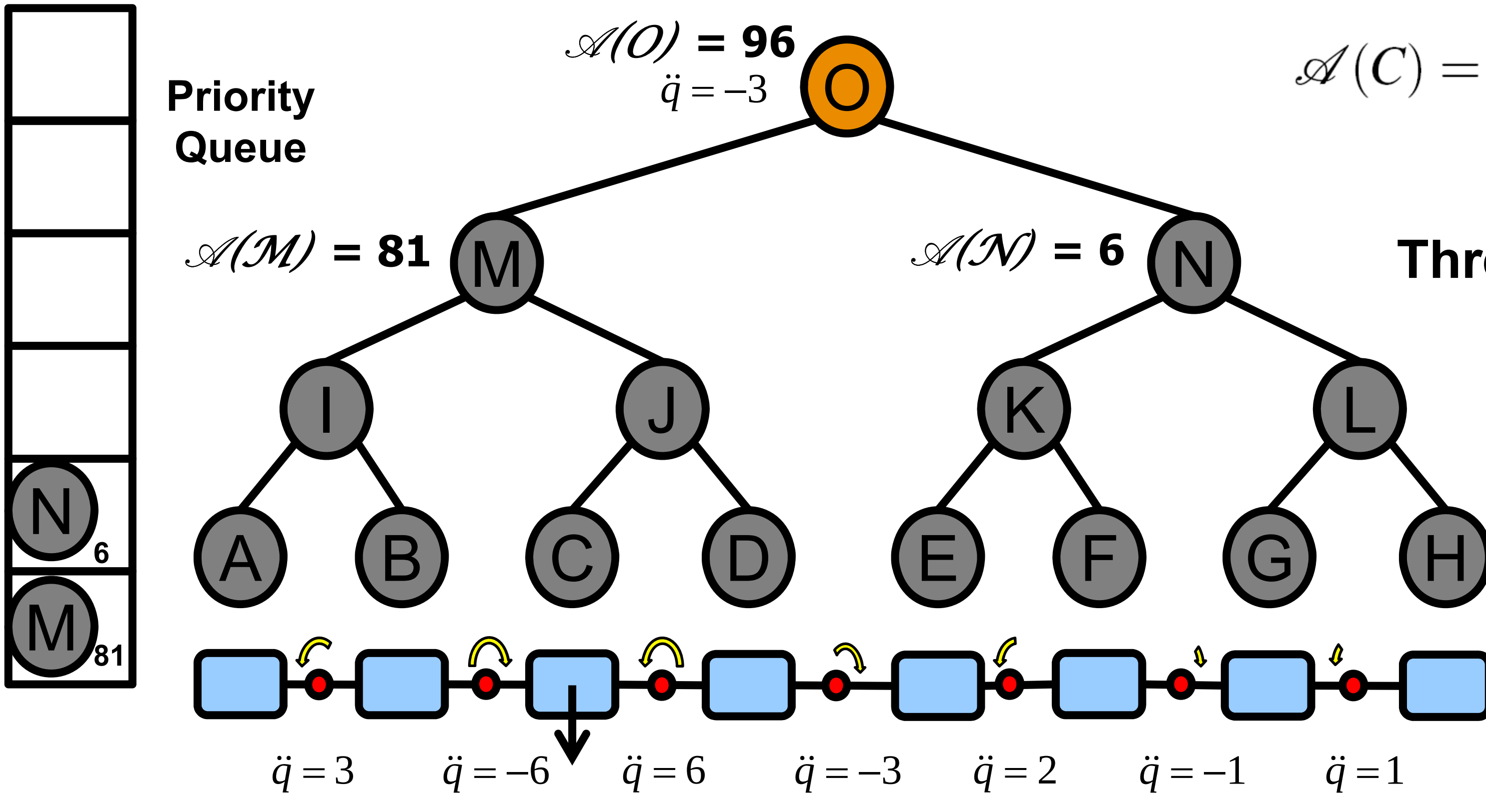
Threshold: 3



The # of simulated joints: 1

Adaptive Forward Dynamics

- **Adaptive joint selection**
  - **Example: Back-substitution pass (↓)**



$$\mathscr{A}(C) = \sum_{i \in C} \ddot{\mathbf{q}}_i^T \mathbf{A}_i \ddot{\mathbf{q}}_i$$

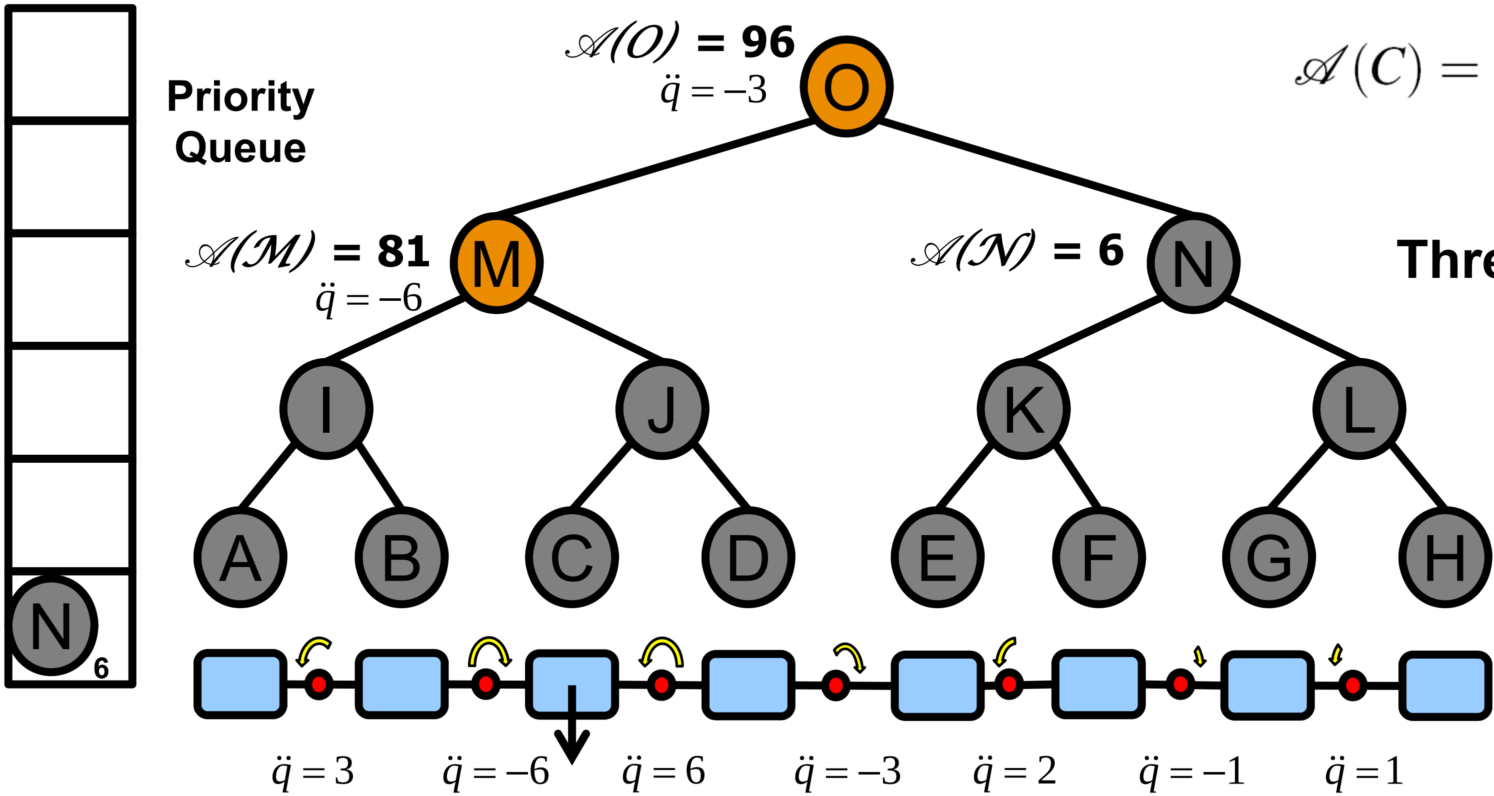
$$A_i = I$$

Threshold: 3

The # of simulated joints: 1

Adaptive Forward Dynamics

- **Adaptive joint selection**
  - **Example: Back-substitution pass (↓)**



$$\mathscr{A}(C) = \sum_{i \in C} \ddot{\mathbf{q}}_i^T \mathbf{A}_i \ddot{\mathbf{q}}_i$$

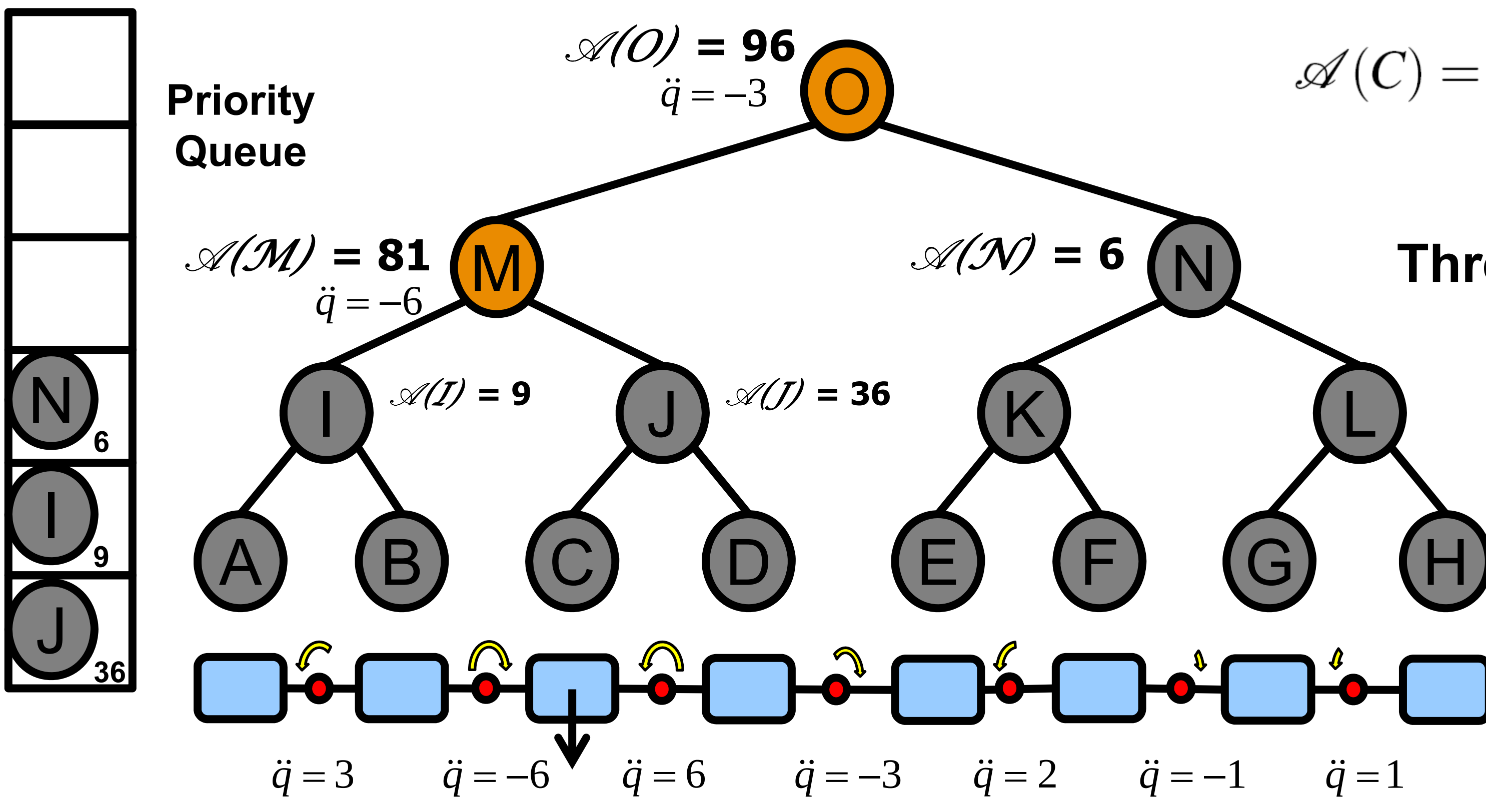
$$A_i = I$$

Threshold: 3

The # of simulated joints: 2

Adaptive Forward Dynamics

- **Adaptive joint selection**
  - **Example: Back-substitution pass (↓)**



$$\mathscr{A}(C) = \sum_{i \in C} \ddot{\mathbf{q}}_i^T \mathbf{A}_i \ddot{\mathbf{q}}_i$$

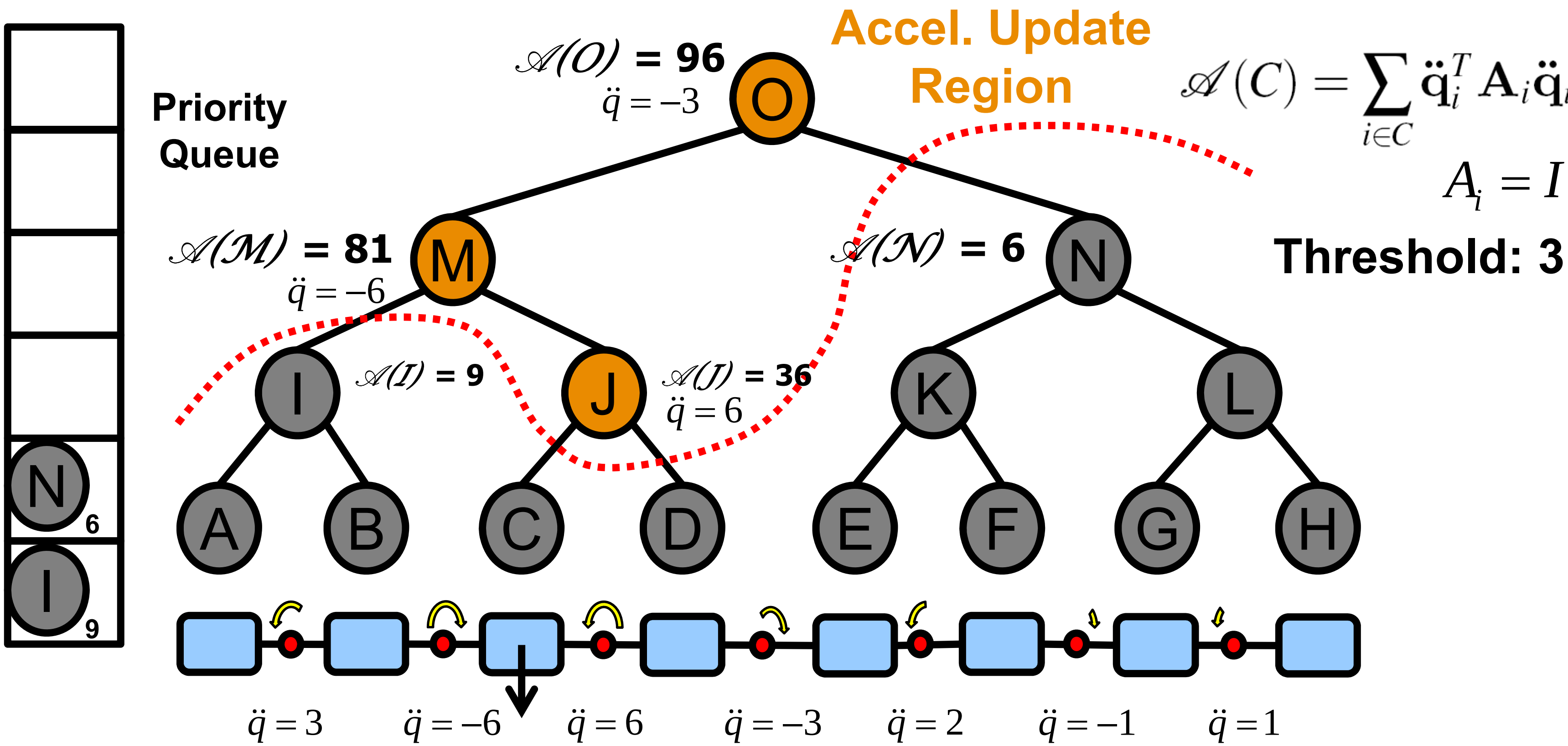
$$A_i = I$$

Threshold: 3

The # of simulated joints: 2

Adaptive Forward Dynamics

- **Adaptive joint selection**
  - **Example: Back-substitution pass (↓)**

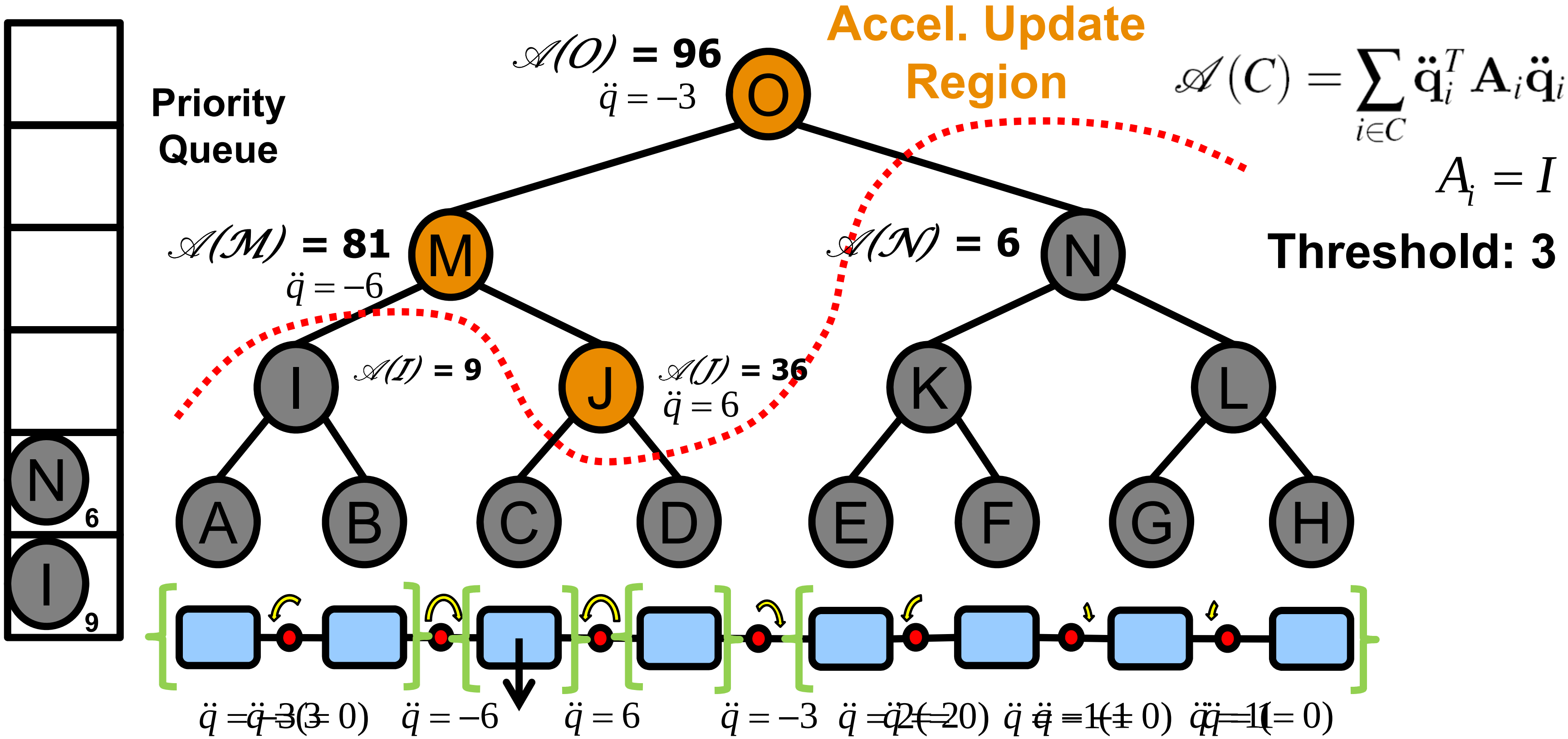


The # of simulated joints: 3

KAIST

Adaptive Forward Dynamics

- **Adaptive joint selection**
  - **Example: Back-substitution pass (↓)**

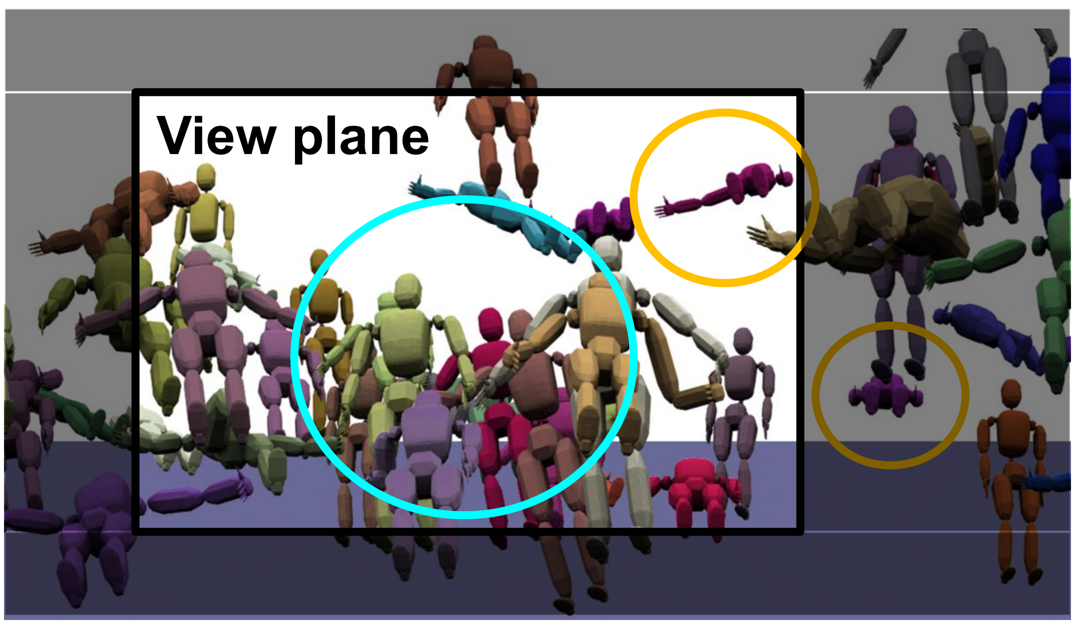


The accelerations of joints within four assemblies are *Implicitly* set to zero.

The # of simulated joints: 3

View-Dependent Forward Dynamics

- How to best utilize the motion metrics to save more computation time?
- How to minimize the error metric values with a given threshold value?



Large-scale virtual world

Much occlusion
Very far objects
Many invisible objects

[Stephane R. et al., `05]

View-Dependent Forward Dynamics

- How to best utilize the motion metrics to save more computation time?

- Occluded bodies
  - Decrease weight coefficients for occluded bodies

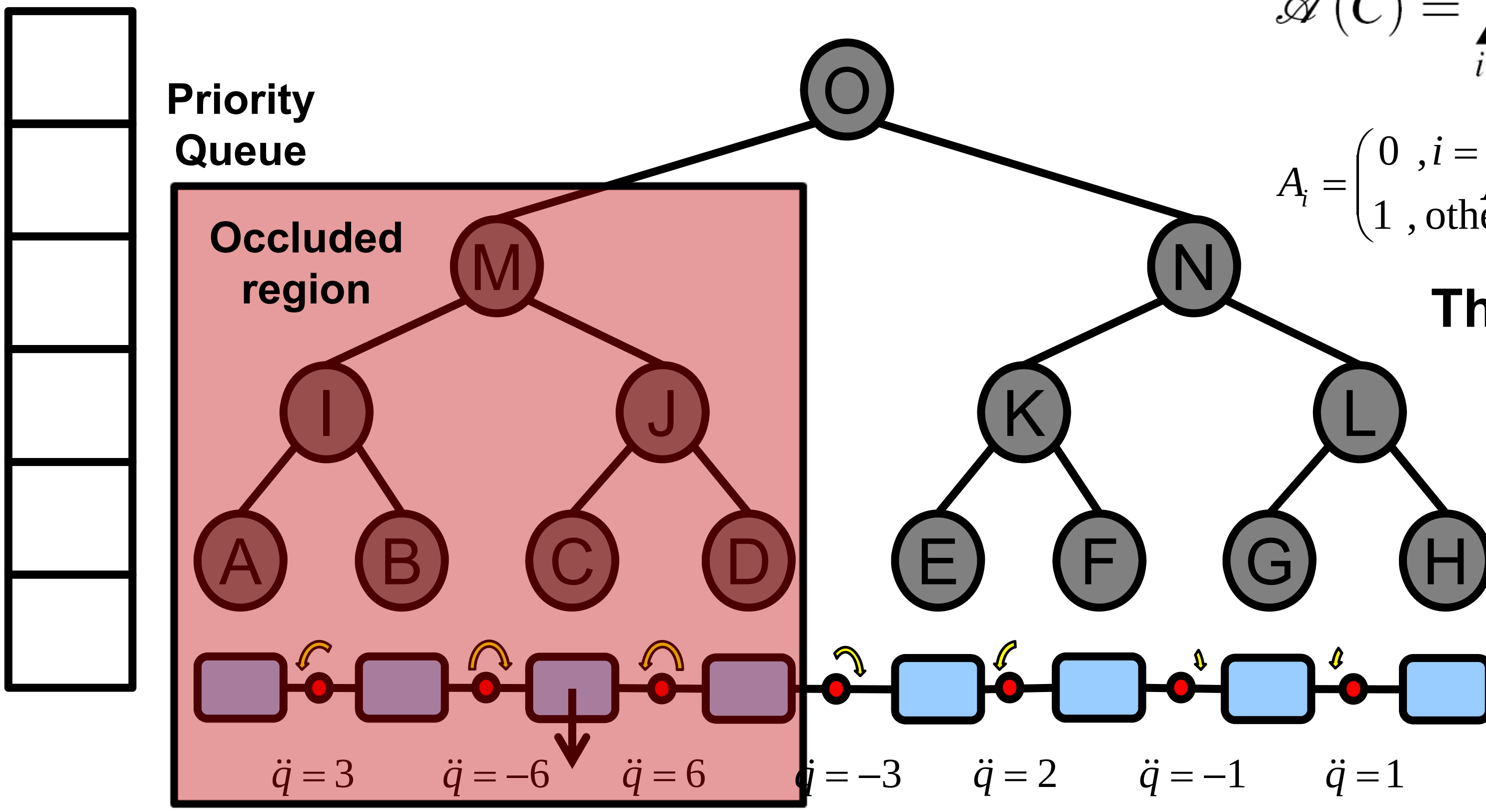
View-Dependent Forward Dynamics

- **View-dependent joint selection**
  - **Example: Back-substitution pass (↓)**

$$\mathscr{A}(C) = \sum_{i \in C} \ddot{\mathbf{q}}_i^T \mathbf{A}_i \ddot{\mathbf{q}}_i$$

$$A_i = \begin{cases} 0 & , i = A, B, C, D, I, J, M \\ 1 & , \text{otherwise} \end{cases} \quad A_i = I$$

Threshold: 3



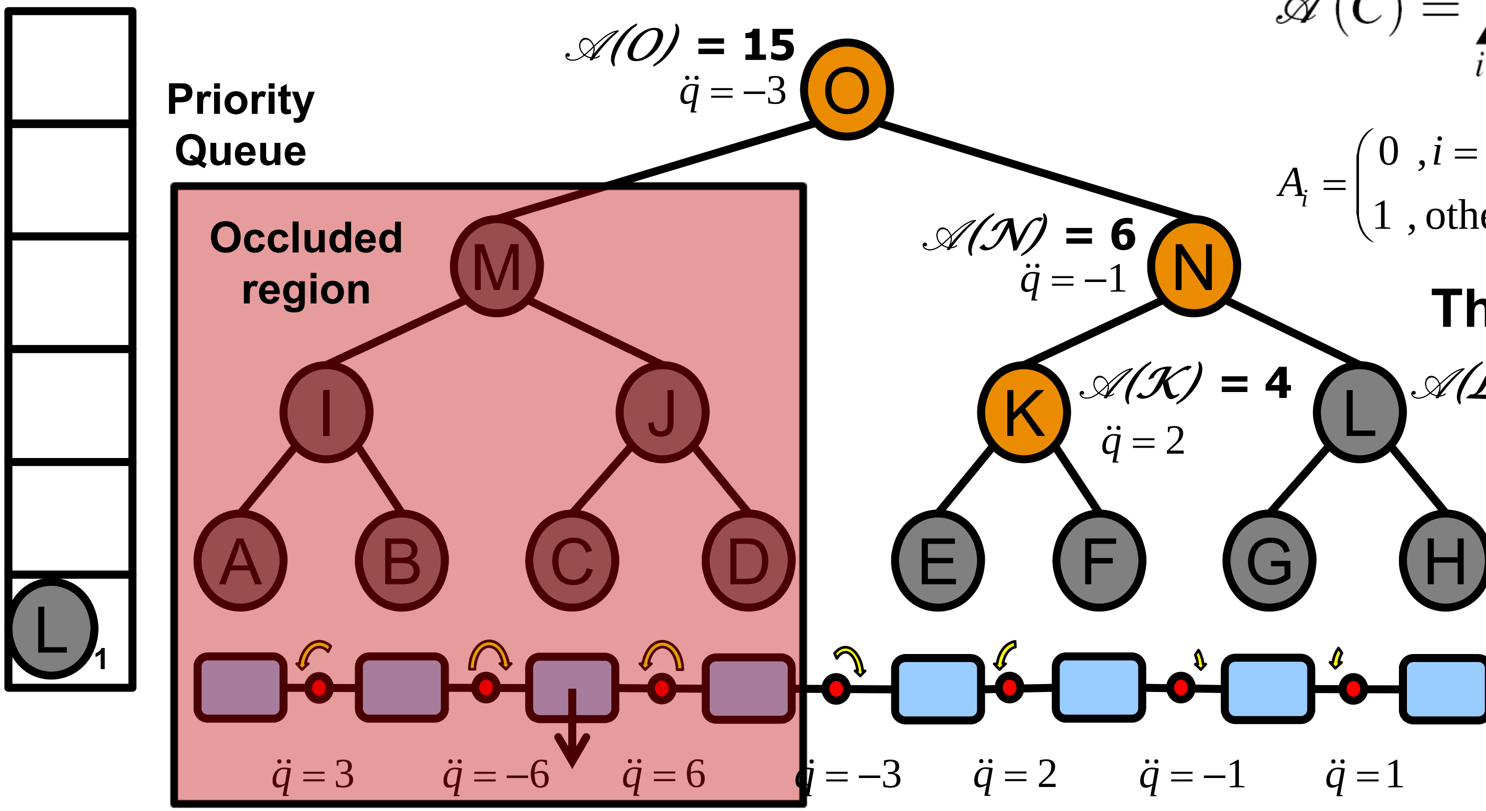
The # of simulated joints: 0

View-Dependent Forward Dynamics

- **View-dependent joint selection**
  - **Example: Back-substitution pass (↓)**

$$\mathscr{A}(C) = \sum_{i \in C} \ddot{\mathbf{q}}_i^T \mathbf{A}_i \ddot{\mathbf{q}}_i$$

$$A_i = \begin{cases} 0 & , i = A, B, C, D, I, J, M \\ 1 & , \text{otherwise} \end{cases}$$



Threshold: 3

The # of simulated joints: 3

KAIST

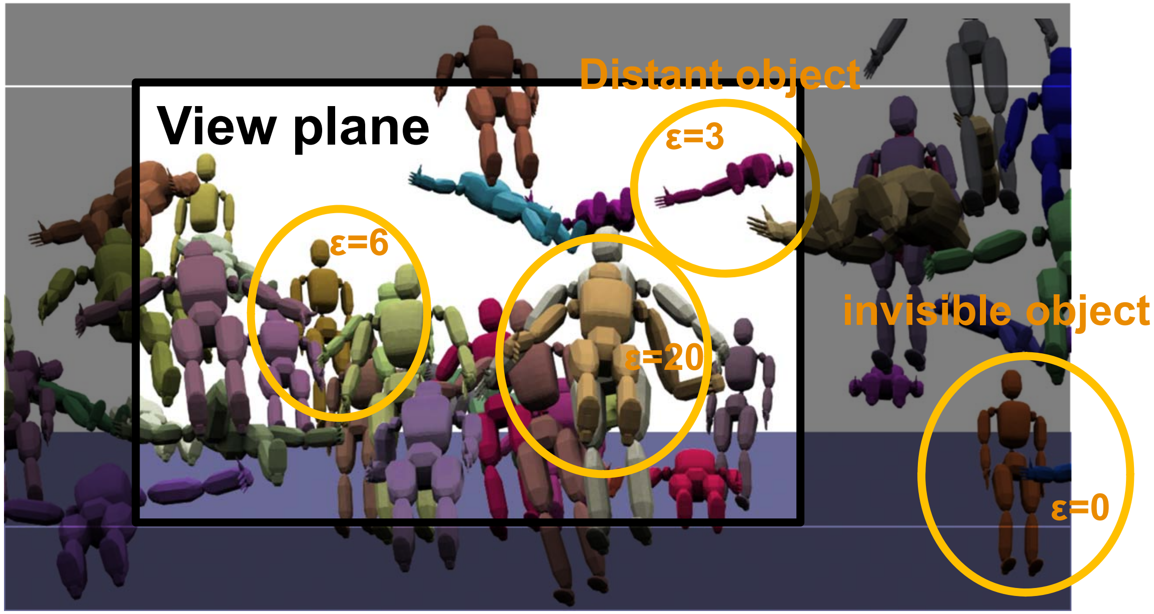
View-Dependent Forward Dynamics

- **How to best utilize the motion metrics to save more computation time?**
- **Occluded bodies**
  - **Decrease weight coefficients for occluded bodies**
- **Totally invisible articulated bodies**
  - **Set the user-specified threshold value to zero**
- **Distant articulated bodies**
  - **Decrease the user-specified threshold value**

View-Dependent Forward Dynamics

- **Example**

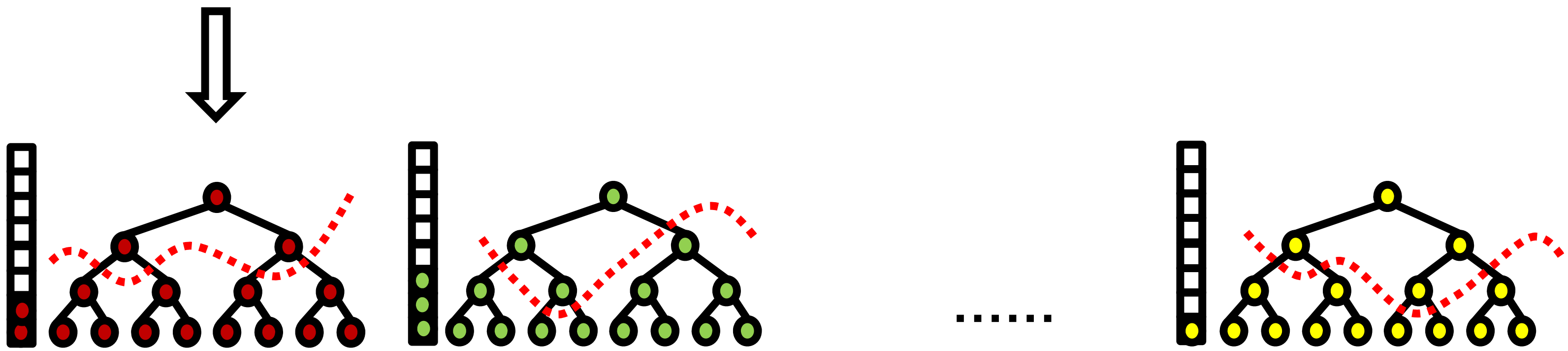
ε: threshold

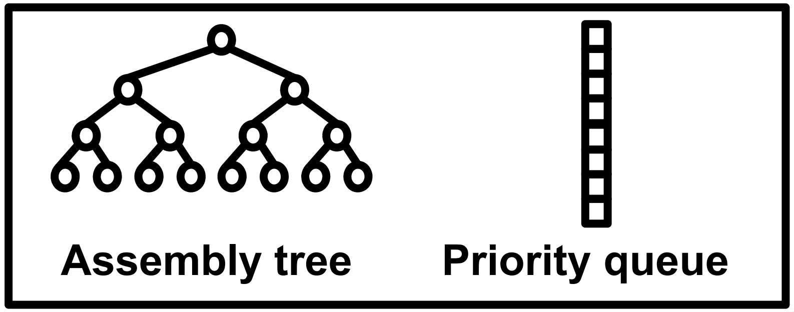


View-Dependent Forward Dynamics

- **How to minimize the error metric values with a given threshold value?**

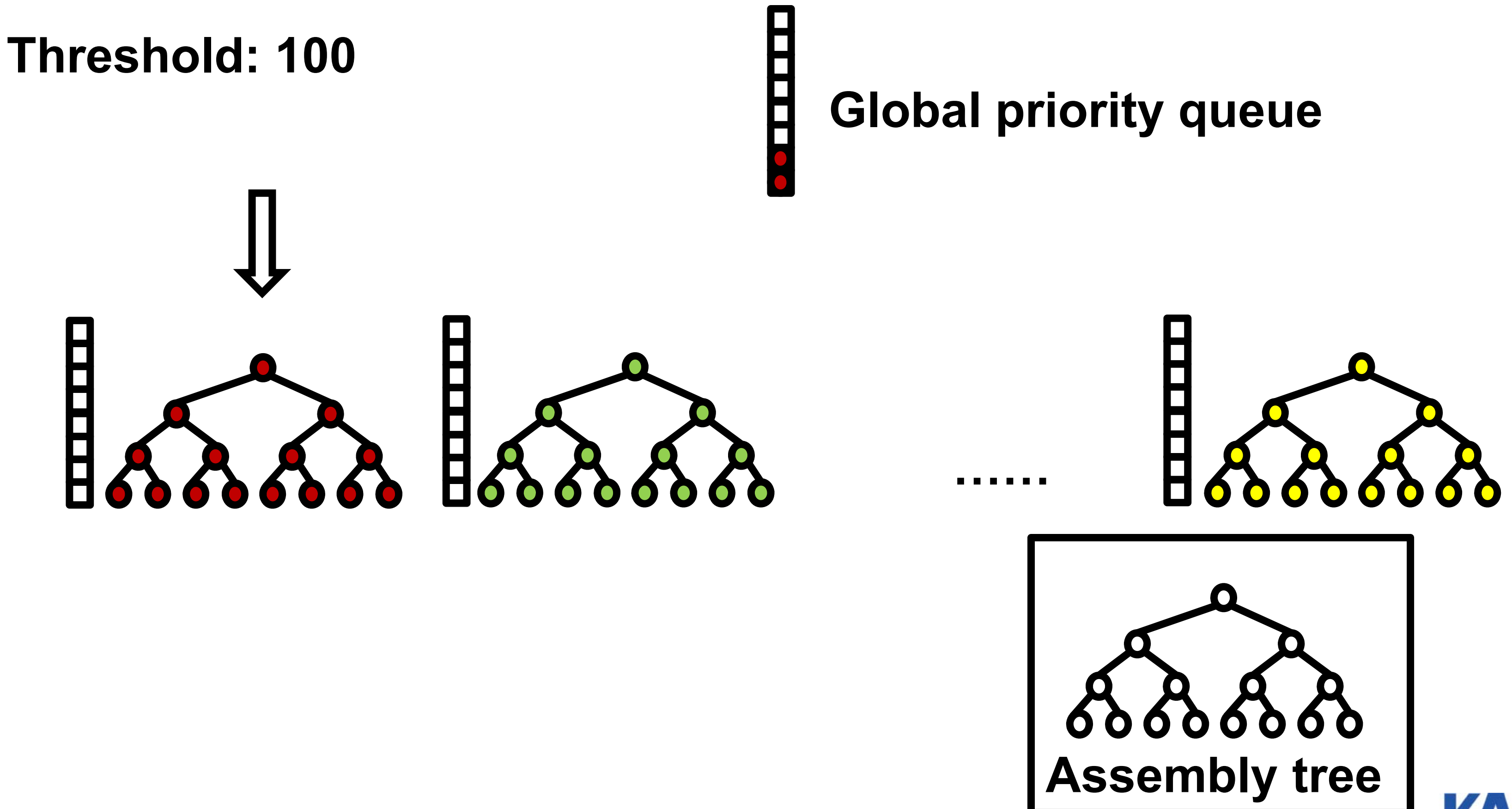
Threshold: 100





View-Dependent Forward Dynamics

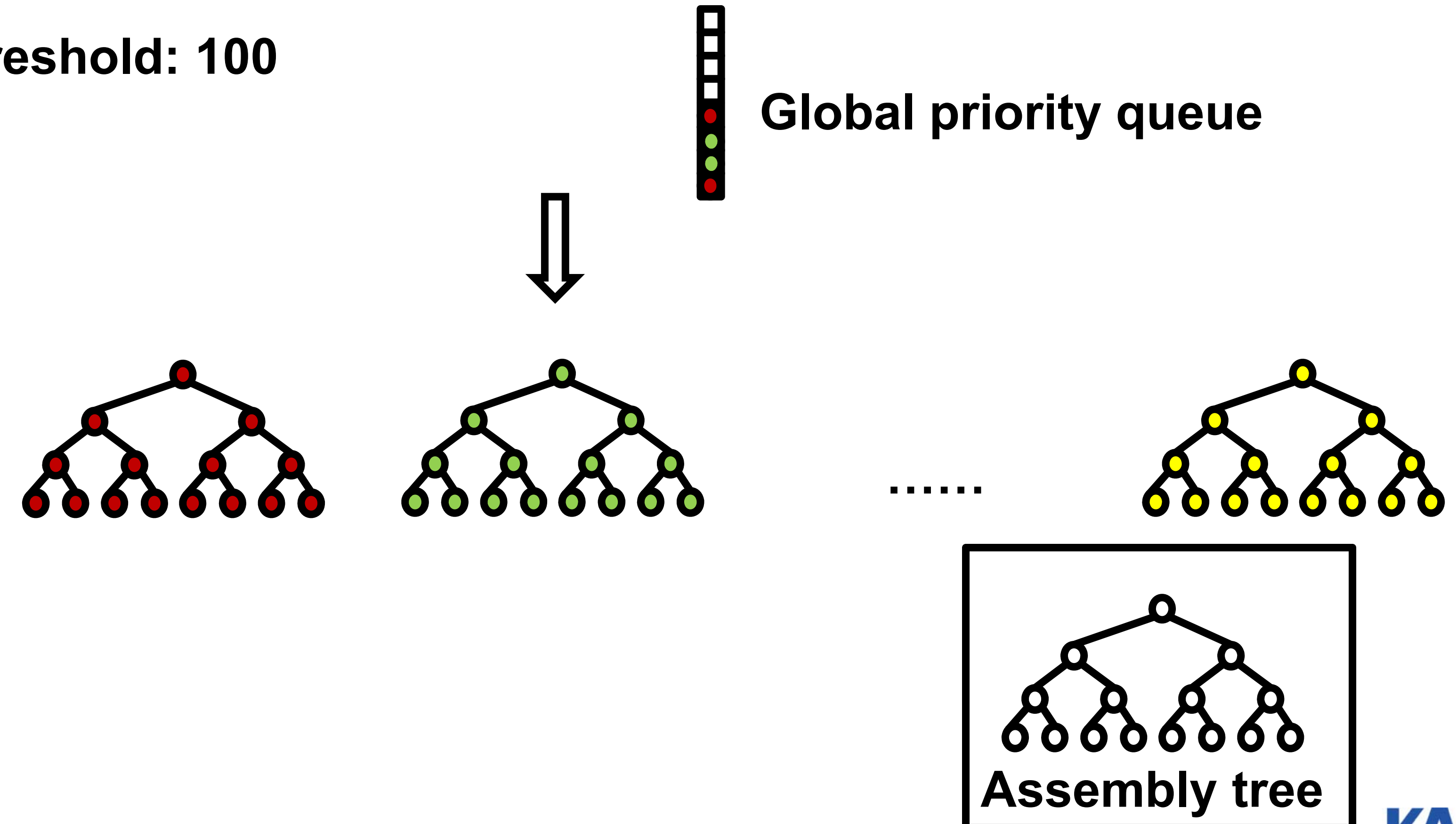
- **How to minimize the error metric values with a given threshold value?**



View-Dependent Forward Dynamics

- **How to minimize the error metric values with a given threshold value?**

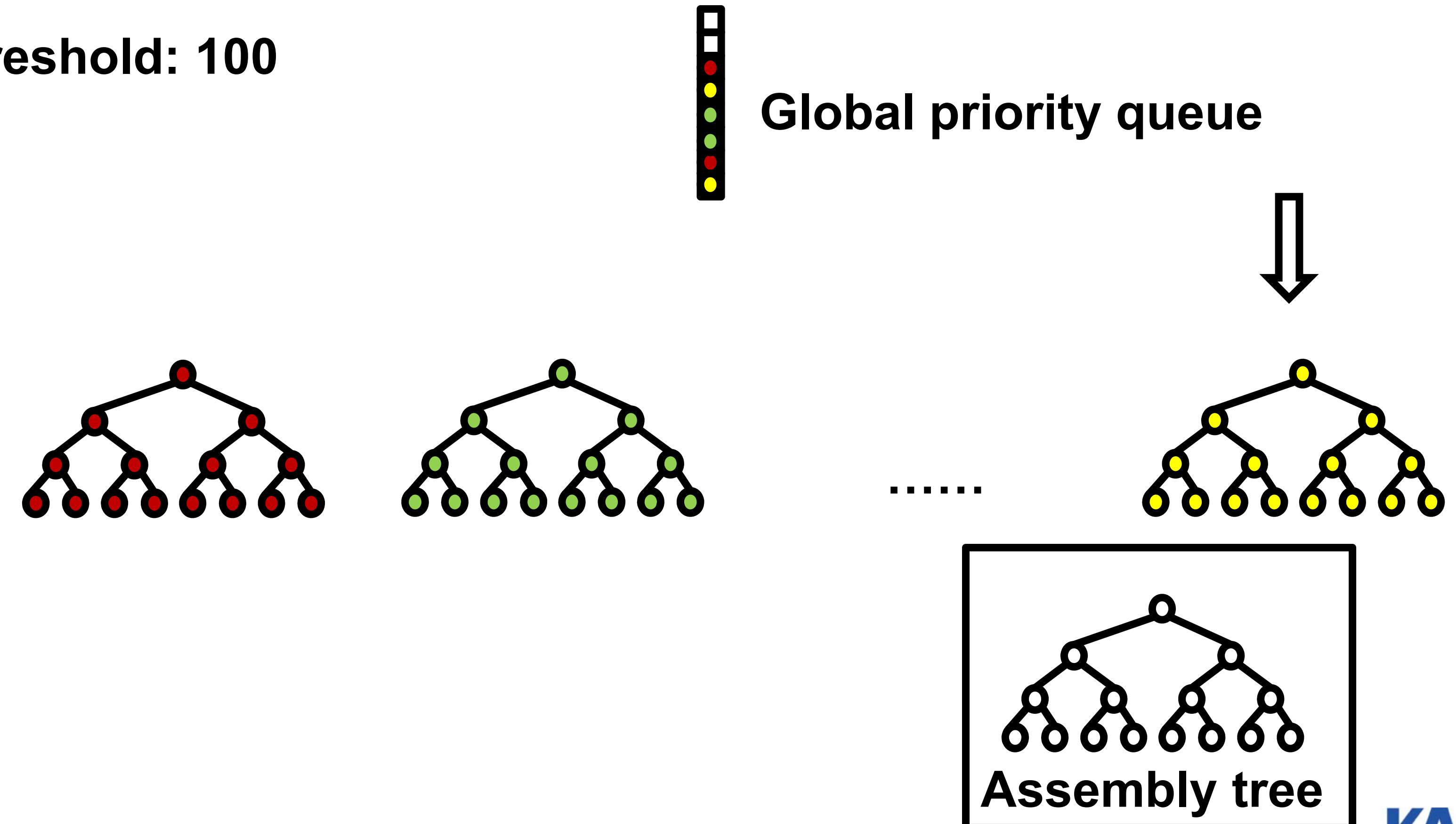
Threshold: 100



View-Dependent Forward Dynamics

- **How to minimize the error metric values with a given threshold value?**

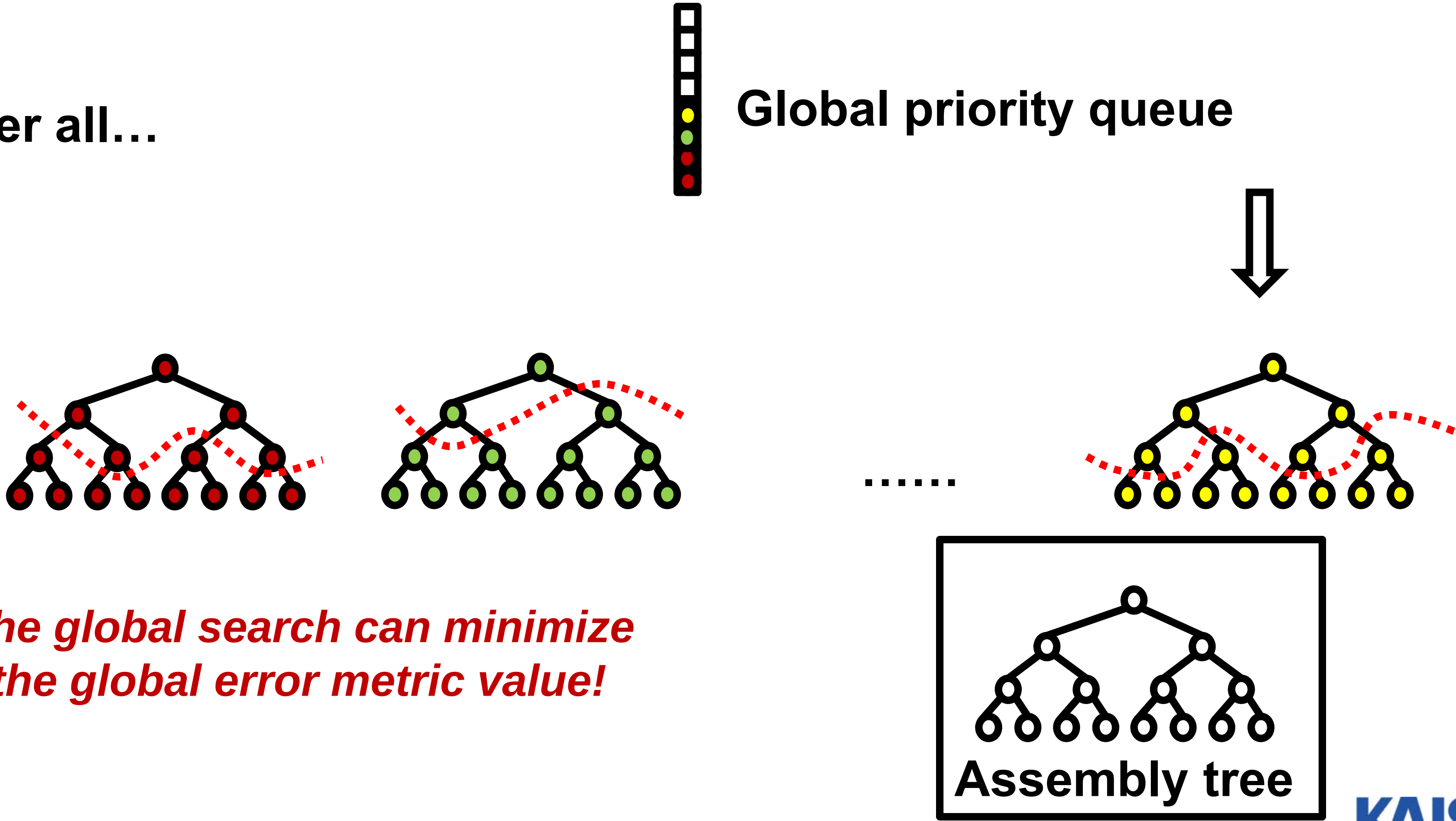
Threshold: 100



View-Dependent Forward Dynamics

- **How to minimize the error metric values with a given threshold value?**

After all…



The global search can minimize the global error metric value!

Conclusion

- SLODs
  - Hybrid kinetic/dynamic model
  - Visibility culling
    - Consistency problem
    - Completeness problem
  - Adaptive forward dynamics
    - Control the number of simulated joints by the customizable motion metrics
- View-dependent forward dynamics
  - Effectively utilize the motion metrics with visibility info

KAIST

DCA For Forward Dynamics

- **Appendix I**
  - **Derivation of the articulated-body equation**

I. Single handle

$$Iv = \text{constant}$$

$$f = \frac{d}{dt}(Iv) = Ia + (v\times^* I - Iv\times)v = Ia + v\times^* Iv$$

$$f = Ia + p \qquad (p = v\times^* Iv)$$

$$a = \Phi f + b \qquad (\Phi = I^{-1}, \mathrm{b} = -\Phi p)$$

II. Multiple handles

$$\begin{bmatrix} \mathbf{a}_1 \\ \mathbf{a}_2 \\ \vdots \\ \mathbf{a}_m \end{bmatrix} = \begin{bmatrix} \mathbf{\Phi}_1 & \mathbf{\Phi}_{12} & \cdots & \mathbf{\Phi}_{1m} \\ \mathbf{\Phi}_{21} & \mathbf{\Phi}_2 & \cdots & \mathbf{\Phi}_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{\Phi}_{m1} & \mathbf{\Phi}_{m2} & \cdots & \mathbf{\Phi}_m \end{bmatrix} \begin{bmatrix} \mathbf{f}_1 \\ \mathbf{f}_2 \\ \vdots \\ \mathbf{f}_m \end{bmatrix} + \begin{bmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \\ \vdots \\ \mathbf{b}_m \end{bmatrix}$$