Dynamics and
Real-Time
Simulation
(DARTS)
Laboratory

Spatial Operator Algebra (SOA)

4. Serial-Chain, Rigid Body Dynamics

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.

Recap

Recap

- Discussed minimal coordinate kinematics model of a rigid body serial-chain
- Introduced stacked notation
- Introduced the $\mathcal{E}_{\phi}$, H and ϕ spatial operators
- Derived recursive kinematics algorithms for poses and body spatial velocities
- Discussed duality between operator expressions and O(N) recursive computations:
  - $y = \phi^* x$ base-to-tip O(N) <u>scatter</u> recursion
  - $y = \phi x$ tip-to-base O(N) <u>gather</u> recursion
- Introduced Jacobian and its operator expression

Serial-Chain Rigid Body Dynamics

Outline

- System mass matrix
  - Newton Euler Factorization
  - Composite body inertias
  - Computing the mass matrix
- Serial chain equations of motion
  - Operator expressions
  - External forces, gravity
- Inverse dynamics
  - O(N) Recursive Newton-Euler
  - Using composite body inertias

System Mass Matrix $\mathcal{M}(\theta)$

System kinetic energy

System kinetic energy is the sum of the body kinetic energies

$$\mathfrak{K}_e \overset{2.5}{=} \frac{1}{2}\sum_{k=1}^{n} \mathcal{V}^*(k)M(k)\mathcal{V}(k)$$

$$= \frac{1}{2}[\mathcal{V}^*(1), \cdots, \mathcal{V}^*(n)] \begin{pmatrix} M(1) & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & M(2) & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & M(n) \end{pmatrix} \begin{pmatrix} \mathcal{V}(1) \\ \mathcal{V}(2) \\ \vdots \\ \mathcal{V}(n) \end{pmatrix}$$

$$\boxed{= \frac{1}{2}\mathcal{V}^*\mathbf{M}\mathcal{V}}$$

where $$\boxed{\mathbf{M} \triangleq \mathrm{diag}\left\{M(k)\right\}_{k=1}^{n} \in \mathcal{R}^{6n\times 6n}}$$

block diagonal spatial inertia operator

System mass matrix $\mathcal{M}(\theta)$

Using $\mathcal{V} = \phi^* H^* \dot{\theta}$

The kinetic energy can be expressed as

$$\mathfrak{K}_e = \frac{1}{2}\mathcal{V}^*\mathbf{M}\mathcal{V} \overset{9,4.3}{=} \frac{1}{2}\dot{\theta}^* H\phi\mathbf{M}\phi^* H^* \dot{\theta} = \frac{1}{2}\dot{\theta}^* \boxed{\mathcal{M}(\theta)}\dot{\theta}$$

mass matrix

$$\boxed{\mathcal{M}(\theta) \overset{\triangle}{=} H\phi\mathbf{M}\phi^* H^* \in \mathcal{R}^{\mathcal{N}\times\mathcal{N}}}$$

Newton-Euler factorization of the mass matrix

$$\boxed{\mathfrak{K}_e = \frac{1}{2}\beta^*\mathcal{M}(\theta)\beta}$$

more general form

Properties of the mass matrix $\mathcal{M}(\theta)$

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

- Square, symmetric and positive definite
- Size is the number of velocity degrees of freedom
- The mass matrix is configuration dependent
- Dense matrix for serial chain systems
  - key reason for its perceived “complexity”
- Maps generalized velocities to system kinetic energy
- Not all of the operators in the Newton-Euler factorization of the mass matrix are square
  - Will encounter other factorizations with square factors
- Elements of $\phi^*\mathrm{H}^*$ are Kane’s partial velocities

Later - The NE operator factorization holds for any tree/branched system.

Computing the mass matrix $\mathcal{M}(\theta)$

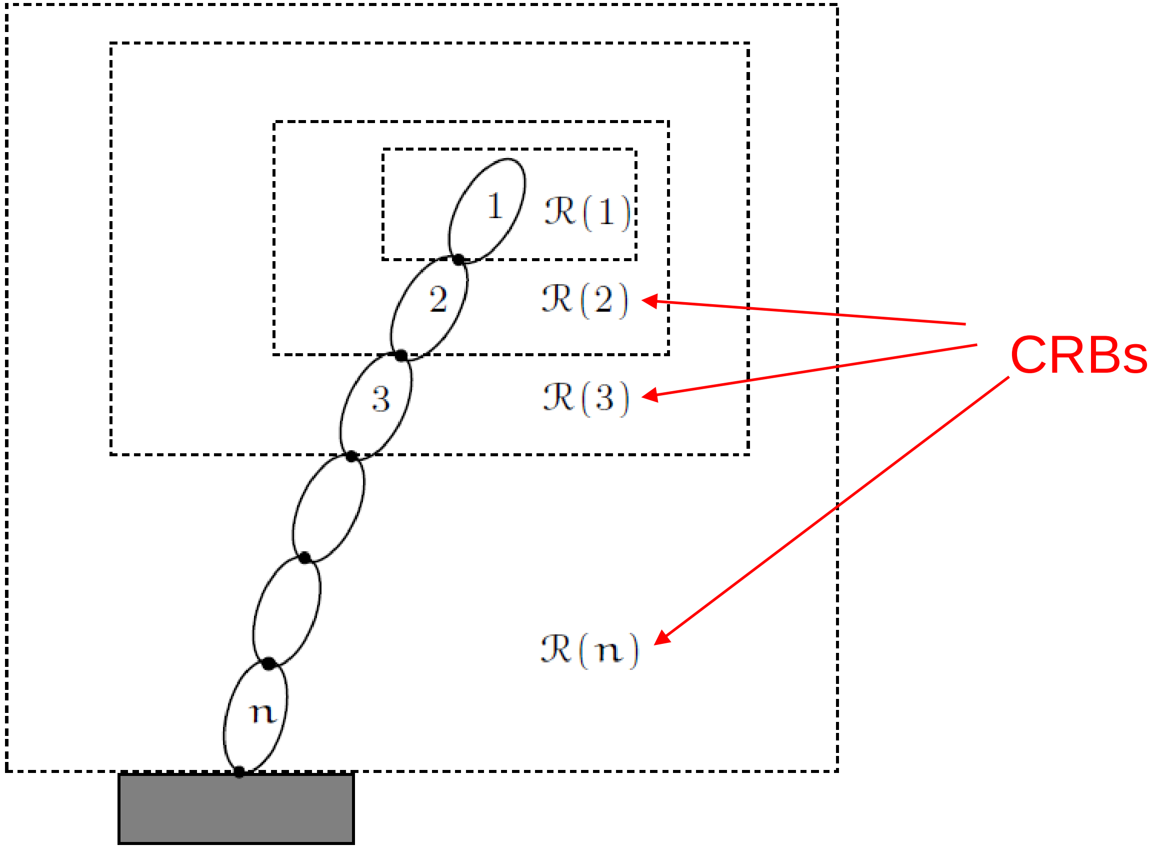
$$\boxed{\mathcal{M}(\theta) \triangleq H\phi\mathbf{M}\phi^* H^* \in \mathcal{R}^{\mathcal{N}\times\mathcal{N}}}$$

- Computing the mass matrix is the major goal of conventional dynamics formulations
- The Newton-Euler factorization can be used to compute the mass matrix
  - Compute each of the component operators, and then take their product via the factored expression
  - Given the size of the operators, this process is of $O(\mathcal{N}^3)$ computational complexity
- Can we do better?
  - Yes, by making use of *composite rigid body inertias*

Composite Rigid Body Inertias

Composite Rigid Body (CRB) Inertias

Composite body inertias (CRB) combine the spatial inertias of connected bodies as if the connecting hinges were frozen



CRBs gather recursive algorithm

CRB relationship for connected bodies

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

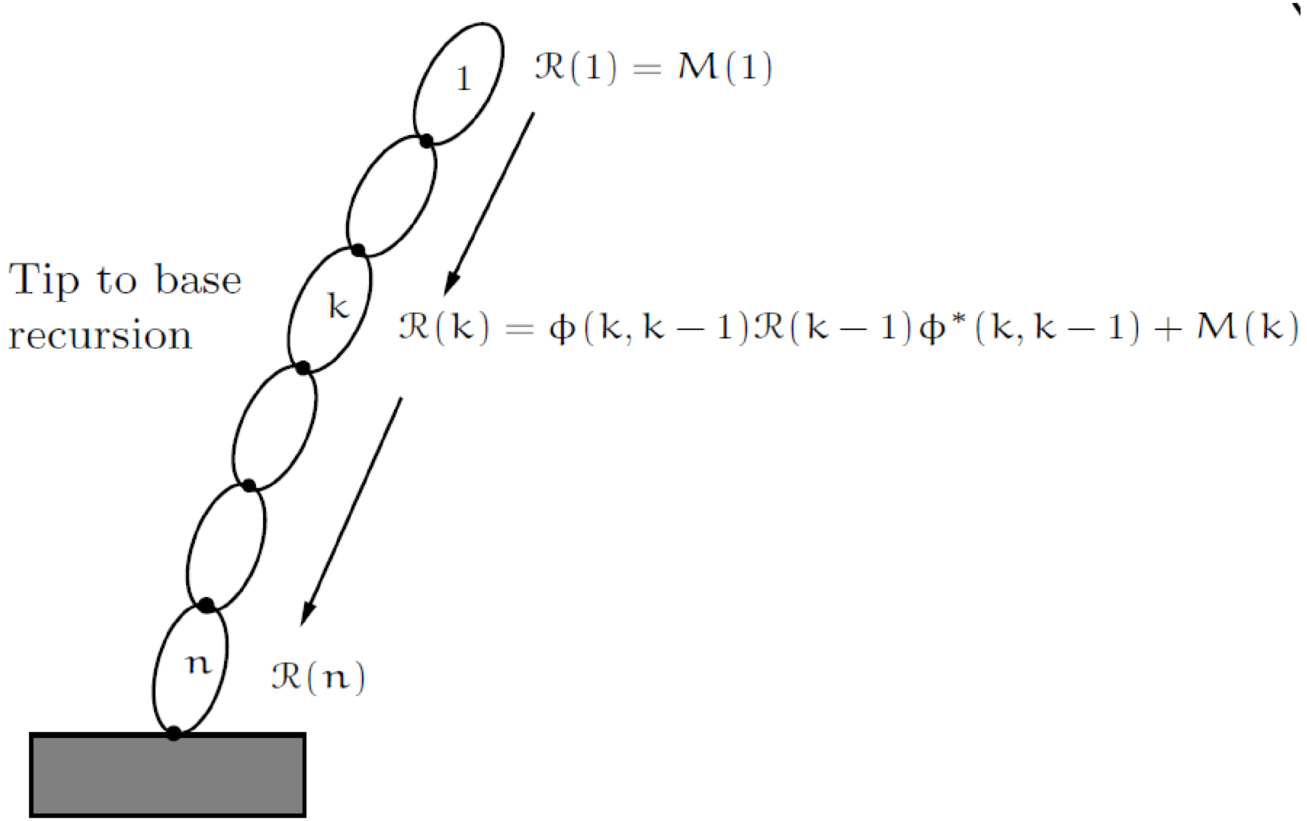
parallel axis transformation of outboard CRB

O(N) recursive, tip-to-base gather algorithm for CRBs

```
$\mathcal{R}(0) = \mathbf{0}$
for $k$   $1 \cdots n$
    $\mathcal{R}(k) = \phi(k, k-1)\mathcal{R}(k-1)\phi^*(k, k-1) + M(k)$
end loop
```

CRBs gather algorithm flow

Structure of the O(N) tip-to-base gather algorithm for the CRBs



Properties of CRB

- CRBs are proper spatial inertias

composite rotational inertia

composite CM location

$$\mathcal{R}(k) = \begin{pmatrix} \mathfrak{J}(k) & \widetilde{\mathfrak{l}}(k, \mathbb{C}_k)\rho(k) \\ -\rho(k)\widetilde{\mathfrak{l}}(k, \mathbb{C}_k) & \rho(k)\mathbf{I} \end{pmatrix}$$

composite mass

- $\mathcal{R}(k)$ is <u>configuration dependent</u>, but depends only on the **outboard** coordinates and does not depend on **inboard** body coordinates

Walker & Orin CRB algorithm

Spatial notation version

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

Equivalent Walker/Orin CRB algorithm at the component level

Great illustration of the compactness of spatial notation expressions

$$\begin{aligned}
\rho(k) &= \rho(k-1) + \mathfrak{m}(k) \\
\mathfrak{l}(k, \mathbb{C}_k)\rho(k) &= \mathfrak{l}(k, \mathbb{C}_{k-1})\rho(k-1) + \mathfrak{m}(k)\mathfrak{p}(k) \\
\mathfrak{J}(k) &= \mathfrak{J}(k-1) + \rho(k-1)[\mathfrak{l}^*(k, \mathbb{C}_{k-1})\mathfrak{l}(k, \mathbb{C}_{k-1})\mathbf{I} \\
&\quad - \mathfrak{l}(k, \mathbb{C}_{k-1})\mathfrak{l}^*(k, \mathbb{C}_{k-1})] \\
&\quad - \rho(k-1)[\mathfrak{l}^*(k-1, \mathbb{C}_{k-1})\mathfrak{l}(k-1, \mathbb{C}_{k-1})\mathbf{I} \\
&\quad - \mathfrak{l}(k-1, \mathbb{C}_{k-1})\mathfrak{l}^*(k-1, \mathbb{C}_{k-1})] + \mathcal{J}(k)
\end{aligned}$$

System spatial inertia and momentum

System center of mass

With pick-off operator

$$\mathrm{E} \triangleq [\mathbf{0}_6, \cdots \mathbf{0}_6, \mathbf{I}_6] \in \mathcal{R}^{6\times 6n}$$

system spatial inertia is the base body's CRB

$$\mathrm{M}_S = \mathcal{R}(n) = \mathrm{E}\mathcal{R}\mathrm{E}^*$$

Its first moment specifies the instantaneous location of the system center of mass.

System spatial momentum

The system spatial momentum is given by

$$\mathfrak{h}_S = E\phi\mathcal{R}H^*\dot{\theta}$$

SHOW!

For floating base systems, the spatial momentum takes the form

$$\mathfrak{h}_S = \underbrace{\boxed{\mathcal{R}(n)\mathcal{V}(n)}}_{\text{spatial momentum if all hinges are locked}} + \sum_{k=1}^{n-1} \underbrace{\phi(n,k)\mathcal{R}(k)H^*(k)\dot{\theta}(k)}_{\text{spatial momentum contribution from internal motion}}$$

SHOW!

System CM spatial velocity

The system CM spatial velocity (inertially referenced to the body frame, i.e. $\mathcal{V}_{\mathbb{C}} = \phi^*(\mathbb{C}_S, n)\mathcal{V}(\mathbb{C}_S)$) is

system CM location

$$\mathcal{V}_{\mathbb{C}} = M_S^{-1}\mathfrak{h}_S = \mathcal{R}^{-1}(n)\sum_{k=1}^{n}\phi(n,k)\mathcal{R}(k)H^*(k)\dot{\theta}(k)$$

For floating base systems, the CM spatial velocity is

$$\mathcal{V}_{\mathbb{C}} = \boxed{\mathcal{V}(n)} + \mathcal{R}^{-1}(n)\sum_{k=1}^{n-1}\phi(n,k)\mathcal{R}(k)H^*(k)\dot{\theta}(k)$$

spatial velocity of the base body

CM spatial velocity contribution from internal motion

Nullifying spatial momentum

- When simulating dynamics of floating-base systems (eg. spacecraft or molecules) conserved quantities such as the spatial momentum can build up numerical drift
- Resetting the spatial momentum is simple

$$\mathcal{V}_{\mathbb{C}} = \mathcal{V}(n) + \mathcal{R}^{-1}(n) \sum_{k=1}^{n-1} \phi(n, k)\mathcal{R}(k)H^*(k)\dot{\theta}(k)$$

Just measure the system body frame referenced CM spatial velocity and subtract it from the base body's spatial velocity to nullify spatial momentum and zero out the CM velocity.

Decomposition of $\phi M \phi^*$

Forward Lyapunov Equation for CRBs

CRB recursion

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

Define CRB spatial operator

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

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

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

Why Lyapunov?

Consider the noisy, discrete, time-domain dynamical system

state propagation matrix

state

$$x(k) = \phi(k, k-1)x(k-1) + w(k)$$

$$\mathcal{T}(k) = H(k)x(k)$$

output

time

white noise with covariance M(k)

The **covariance** of the x(k) state is R(k) which is the solution to the discrete Lyapunov equation

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

This is precisely the CRBs recursion! Hence Lyapunov.

Operator decomposition of $\phi \mathbf{M} \phi^*$

Claim:

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

Derivation:

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

and thus pre & post multiplying

use identity

$$\boxed{\tilde{\phi} \stackrel{\triangle}{=} \phi - \mathbf{I} = \mathcal{E}_\phi \phi}$$

$$\phi \mathbf{M} \phi^* \overset{4.9}{=} \phi \mathcal{R} \phi^* - \phi \mathcal{E}_\phi \mathcal{R} \mathcal{E}_\phi^* \phi^* \overset{3.41}{=} \phi \mathcal{R} \phi^* - \tilde{\phi} \mathcal{R} \tilde{\phi}^*$$

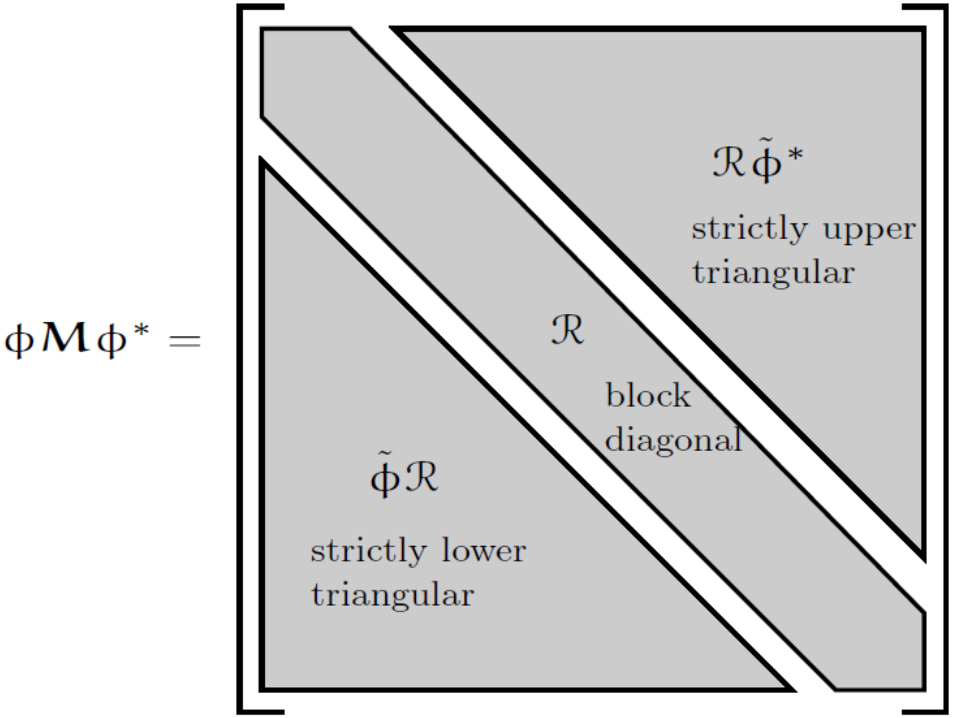
$$\overset{3.40}{=} (\tilde{\phi} + \mathbf{I}) \mathcal{R} (\tilde{\phi} + \mathbf{I}) - \tilde{\phi} \mathcal{R} \tilde{\phi}^* \overset{3.40}{=} \mathcal{R} + \tilde{\phi} \mathcal{R} + \mathcal{R} \tilde{\phi}^*$$

Later – This decomposition holds for any tree/branched system.

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

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

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



Structure of the Mass Matrix using CRBs

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

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

$$\boxed{\mathcal{M}(\theta) \overset{\triangle}{=} H\phi\mathbf{M}\phi^* H^*}$$

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

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

Observations on mass matrix structure

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

Observations:

- *Components are disjointed*
- *The values are full determined by the diagonal CRBs*
- *The sparsity structure of the mass matrix is determined by* $\tilde{\phi}$ *!*
- *Dense for serial chains – but not so for trees.*
- *The operators help reveal the underlying structure – not apparent through other methods*

Later – This mass matrix decomposition holds for any tree/branched system.

Mass matrix as a covariance

Consider the noisy, discrete, time-domain dynamical system

state propagation matrix

state

$$x(k) = \phi(k, k-1)x(k-1) + w(k)$$

$$\mathcal{T}(k) = H(k)x(k)$$

output

time

white noise
with covariance
M(k)

The **covariance** of the x(k) state is R(k) is the solution to the discrete Lyapunov equation

… and $\mathcal{M}$ is the **covariance** of the $\mathcal{T}(k)$ output process!

Elements of the mass matrix $\mathcal{M}(\theta)$

The CRB based decomposition can be used to obtain explicit expressions for the mass matrix elements

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

At the component level

$$\mathcal{M}(i,j) = \begin{cases} H(i)\mathcal{R}(i)H^*(i) & \text{for } i = j \quad \textit{diagonal} \\ H(i)\phi(i,j)\mathcal{R}(j)H^*(j) & \text{for } i > j \quad \textit{lower triangular} \\ \mathcal{M}^*(j,i) & \text{for } i < j \quad \textit{upper triangular} \end{cases}$$

Recall $\boxed{\phi(i,j) = \phi(i,i-1)\ \cdots\ \phi(j+1,j)}$

Recursive computation of the mass matrix $\mathcal{M}(\theta)$

$O(\mathcal{N}^2)$ ecursive, tip-to-base, gather algorithm for the mass matrix based on composite body inertias – *no explicit computation of operators required*

```
$\mathcal{R}(0) = \mathbf{0}$
for $k$   $1 \cdots n$
    $\mathcal{R}(k) = \phi(k, k-1)\mathcal{R}(k-1)\phi^*(k, k-1) + M(k)$
        $X(k) = \mathcal{R}(k)H^*(k), \quad \mathcal{M}(k, k) = H(k)X(k)$
    for $j$   $(k+1) \cdots n$
        $X(j) = \phi(j, j-1)X(j-1)$
        $\boxed{\mathcal{M}(j, k)} = \mathcal{M}^*(k, j) = H(j)X(j)$
    end loop
end loop
```

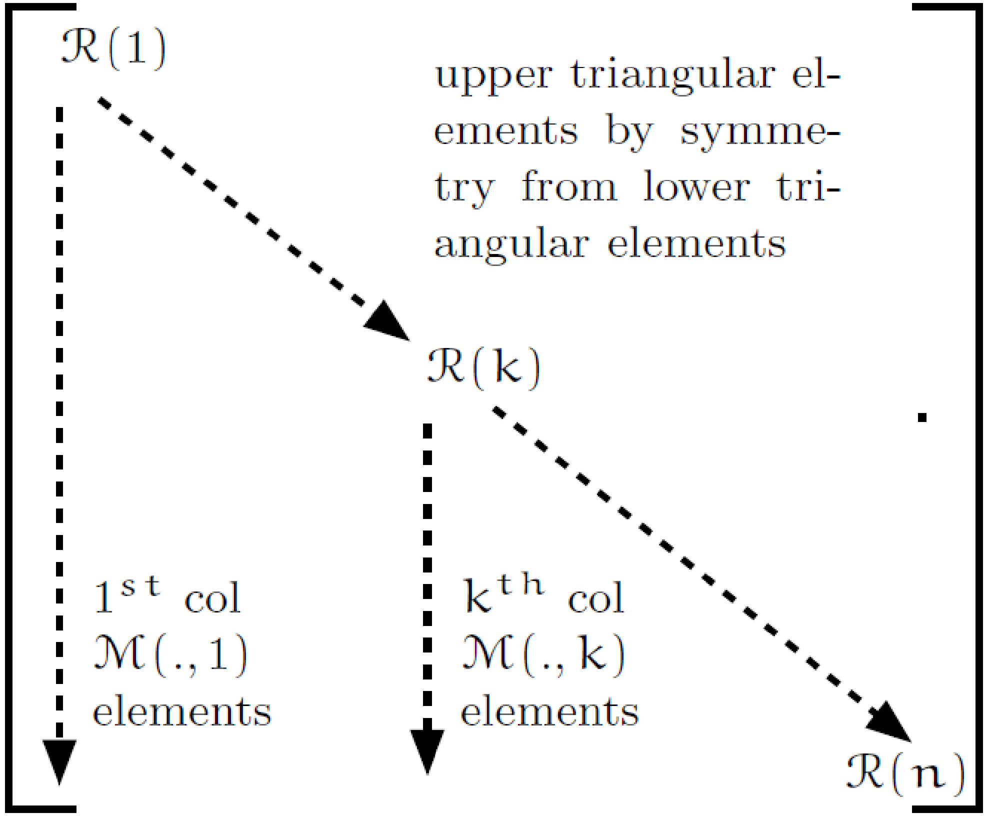
Originally developed by Walker & Orin

Exploiting the CRB based structure has lowered the cost from $O(\mathcal{N}^3)$ to $O(\mathcal{N}^2)$ complexity.

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

$\mathcal{R}(k)$

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

kth 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

SOA based Mass Matrix Computation

Spatial Operator Algebra (SOA) *based mathematical analysis*

Mapping to ***structure based****, recursive algorithms*

Dynamics properties → Transformed Expressions → Low-order structure-based algorithms

$$\mathcal{M}(\theta) \triangleq H\phi\mathbf{M}\phi^*H^* \Rightarrow \mathcal{M} = H\mathcal{R}H^* + H\tilde{\phi}\mathcal{R}H^* + H\mathcal{R}\tilde{\phi}^*H^* \Rightarrow$$

Composite Rigid Body Inertia Algorithm for the Mass Matrix

Trace of the 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

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

Equations of motion

Deriving equations of motion

Now that we have an expression for the kinetic energy using the mass matrix

$$\mathfrak{K}_e = \frac{1}{2}\dot{\theta}^*\mathcal{M}(\theta)\dot{\theta}$$

we can use it as the Lagrangian in the following to derive the equations of motion:

$$\frac{d}{dt}\frac{\partial \mathcal{L}}{\partial \dot{\theta}} - \frac{\partial \mathcal{L}}{\partial \theta} = \mathcal{T}$$

$$\Downarrow$$

equations of motion

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

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

mass matrix

$$\boxed{\mathcal{C}(\theta, \dot{\theta}) \triangleq \dot{\mathcal{M}}(\theta)\dot{\theta} - \frac{1}{2}\frac{\partial\left[\dot{\theta}^*\mathcal{M}(\theta)\dot{\theta}\right]}{\partial\theta}}$$

Coriolis etc. velocity dep. terms

Lagrangian equations of motion

equations of motion

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

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

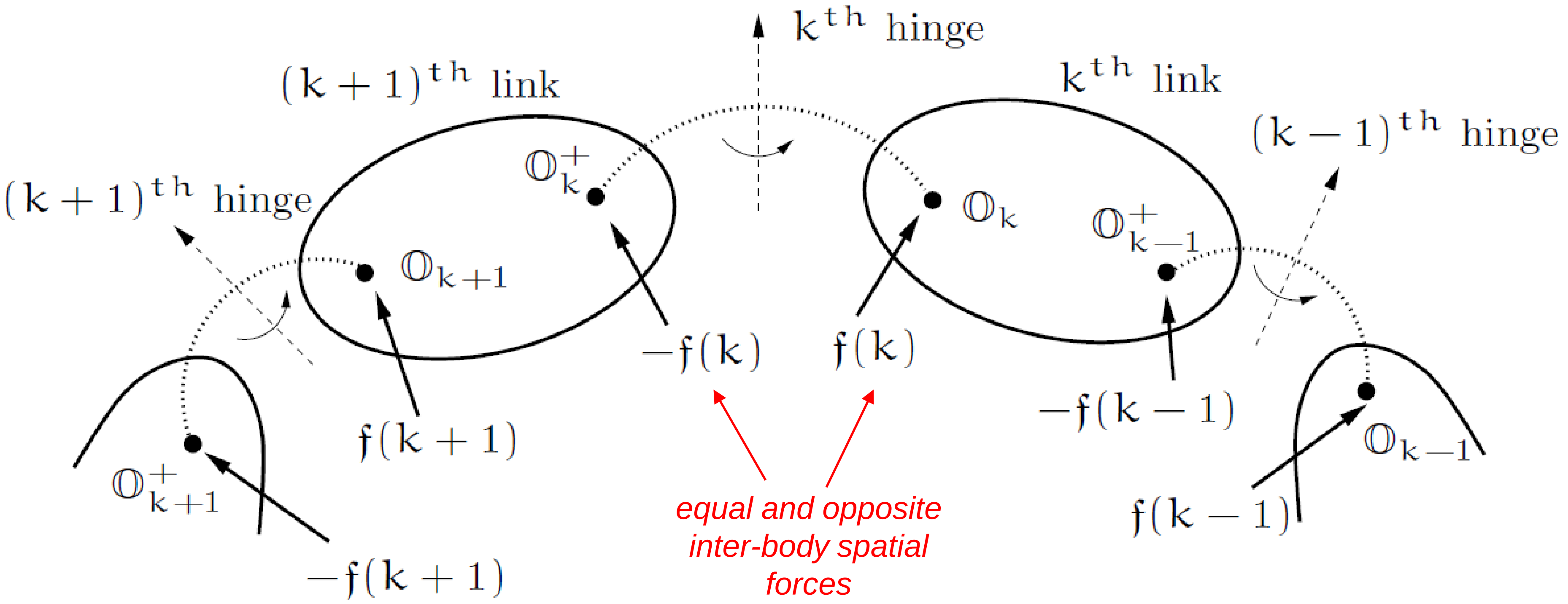
$$\boxed{\mathcal{C}(\theta, \dot{\theta}) \triangleq \dot{\mathcal{M}}(\theta)\dot{\theta} - \frac{1}{2}\frac{\partial\left[\dot{\theta}^*\mathcal{M}(\theta)\dot{\theta}\right]}{\partial\theta}}$$

Options:

- The mass matrix is the critical entity for the eq. of motion
- Most dynamics formulations focus on procedures for deriving the above equations of motion as the <u>ultimate goal</u>
- Derive by hand – not feasible beyond a couple of bodies
- Use automatic differentiation. Does the job but we get a black box and an analysis dead end
- Can do so analytically, but more complex
- We adopt a simpler Newton-Euler approach instead to build up from single body level – and use operators to <u>reveal & exploit structure</u>

Equations of motion for a single link

Force balance for a single link



equal and opposite inter-body spatial forces

$$\boxed{\mathfrak{f}(k) - \phi(k, k-1)\mathfrak{f}(k-1) = \mathcal{M}(k)\alpha(k) + \mathfrak{b}(k)}$$

overall spatial forces from the child and parent bodies

Single link equations of motion

Equations of motion for a single link

$$\mathfrak{f}(k) = \phi(k, k-1)\mathfrak{f}(k-1) + M(k)\alpha(k) + \mathfrak{b}(k)$$

spatial force

where

$$\alpha(k) \triangleq \frac{d_{\mathbb{B}_k}\mathcal{V}(k)}{dt} = \frac{d_k\mathcal{V}(k)}{dt}$$

spatial acceleration

$$\mathfrak{b}(k) \overset{2.28}{=} \overline{\mathcal{V}}(k)M(k)\mathcal{V}(k)$$

gyroscopic force

$$\mathcal{T}(k) = H(k)\mathfrak{f}(k)$$

generalized force

Spatial acceleration recursion

Start with the spatial velocity recursion

$$\mathcal{V}(k) = \phi^*(k+1,k)\mathcal{V}(k+1) + H^*(k)\dot{\theta}(k)$$

Differentiate

$$\alpha(k) = \phi^*(k+1,k)\alpha(k+1) + H^*(k)\ddot{\theta}(k) + \mathfrak{a}(k)$$

Coriolis acceleration

$$\mathfrak{a}(k) \triangleq -\widetilde{\Delta}^{\omega}_{\mathcal{V}}(k)\mathcal{V}(k) + \frac{d_{k+1}\phi^*(k+1,k)}{dt}\mathcal{V}(k+1) + \frac{d_{k+1}H^*(k)}{dt}\dot{\theta}(k)$$

Coriolis accelerations

Coriolis acceleration

$$\mathfrak{a}(k) \triangleq -\widetilde{\Delta}^{\omega}_{\mathcal{V}}(k)\mathcal{V}(k) + \frac{d_{k+1}\phi^*(k+1,k)}{dt}\mathcal{V}(k+1) + \frac{d_{k+1}H^*(k)}{dt}\dot{\theta}(k)$$

Assuming joint map matrix is constant

$$\mathfrak{a}(k) = \widetilde{\mathcal{V}}(k)\Delta_{\mathcal{V}}(k) - \overline{\Delta}_{\mathcal{V}}(k)\Delta_{\mathcal{V}}(k)$$

SHOW!

For pure rotational
or prismatic hinge:

$$\mathfrak{a}(k) = \widetilde{\mathcal{V}}(k)\Delta_{\mathcal{V}}(k)$$

System level equations of motion

Overall body level equations of motion

Gathering together all the component body-level expressions we have

$$\mathcal{V}(k) = \phi^*(k+1, k)\mathcal{V}(k+1) + H^*(k)\dot{\theta}(k)$$ spatial velocities

$$\alpha(k) = \phi^*(k+1, k)\alpha(k+1) + H^*(k)\ddot{\theta}(k) + \mathfrak{a}(k)$$ spatial accels

$$\mathfrak{f}(k) = \phi(k, k-1)\mathfrak{f}(k-1) + M(k)\alpha(k) + \mathfrak{b}(k)$$ spatial forces

$$\mathcal{T}(k) = H(k)\mathfrak{f}(k)$$ generalized forces

Additional stacked vectors

Define additional system-level stacked vectors for body level quantities

generalized forces

$$\mathfrak{T} \triangleq \operatorname{col}\left\{\mathfrak{T}(k)\right\}_{k=1}^{n} \in \mathcal{R}^{\mathcal{N}}$$

$$\mathfrak{f} \triangleq \operatorname{col}\left\{\mathfrak{f}(k)\right\}_{k=1}^{n} \in \mathcal{R}^{6n}$$

$$\mathfrak{a} \triangleq \operatorname{col}\left\{\mathfrak{a}(k)\right\}_{k=1}^{n} \in \mathcal{R}^{6n}$$

spatial forces

Coriolis accels

spatial accels

$$\alpha \triangleq \operatorname{col}\left\{\alpha(k)\right\}_{k=1}^{n} \in \mathcal{R}^{6n}$$

$$\mathfrak{b} \triangleq \operatorname{col}\left\{\mathfrak{b}(k)\right\}_{k=1}^{n} \in \mathcal{R}^{6n}$$

gyroscopic forces

Operator expressions for equations of motion

$$\begin{aligned}
\mathcal{V}(k) &= \phi^*(k+1,k)\mathcal{V}(k+1) + H^*(k)\dot{\theta}(k)\\
\alpha(k) &= \phi^*(k+1,k)\alpha(k+1) + H^*(k)\ddot{\theta}(k) + \mathfrak{a}(k)\\
\mathfrak{f}(k) &= \phi(k,k-1)\mathfrak{f}(k-1) + M(k)\alpha(k) + \mathfrak{b}(k)\\
\mathcal{T}(k) &= H(k)\mathfrak{f}(k)
\end{aligned}$$

body level expressions

$$\begin{aligned}
\mathcal{V} &= \mathcal{E}_\phi^*\mathcal{V} + H^*\dot{\theta}\\
\alpha &= \mathcal{E}_\phi^*\alpha + H^*\ddot{\theta} + \mathfrak{a}\\
\mathfrak{f} &= \mathcal{E}_\phi\mathfrak{f} + \mathbf{M}\alpha + \mathfrak{b}\\
\mathcal{T} &= H\mathfrak{f}
\end{aligned}$$

equivalent system-level **implicit** operator expressions

Implicit to explicit

Use $\phi \triangleq (\mathbf{I} - \mathcal{E}_{\phi})^{-1}$ identity to convert **implicit** operator expressions into **explicit** ones

$$\begin{aligned}
\mathcal{V} &= \mathcal{E}_{\phi}^{*}\mathcal{V} + H^{*}\dot{\theta} \\
\alpha &= \mathcal{E}_{\phi}^{*}\alpha + H^{*}\ddot{\theta} + \mathfrak{a} \\
\mathfrak{f} &= \mathcal{E}_{\phi}\mathfrak{f} + \mathbf{M}\alpha + \mathfrak{b} \\
\mathcal{T} &= H\mathfrak{f}
\end{aligned}$$

implicit expressions

→

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

explicit expressions

System level equations of motion

Combine the operator expressions to obtain the system level equations of motion

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

$$\Longrightarrow$$

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

familiar **mass matrix** $\mathcal{M}(\theta) = H\phi\mathbf{M}\phi^* H^* \in \mathcal{R}^{\mathcal{N}\times\mathcal{N}}$

Newton-Euler factorization

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

Later – These equations of motion hold for any tree/branched system.

Equivalence to Lagrangian approach

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

The equations of motion derived using the Newton-Euler approach are the same as we would have obtained using alternative approaches such as the Lagrangian approach:

- The mass matrix term equivalence is easy to see
- The Coriolis term takes a lot more work, but can be shown to be equivalent

Including external forces

Including external forces on bodies

Update the force balance equation to include external forces

$$\mathfrak{f}(k) - \phi(k, k-1)\mathfrak{f}(k-1) + \boxed{\sum_{i} \phi(\mathbb{B}_k, \mathbb{O}_k^i)\mathfrak{f}_{ext}^i(k)} = \mathcal{M}(k)\alpha(k) + \mathfrak{b}(k)$$

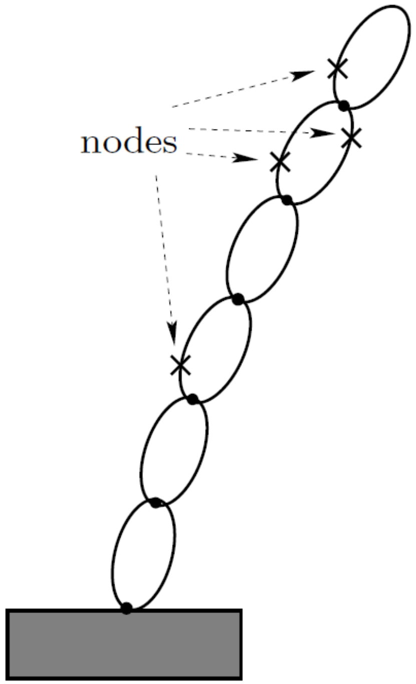
external forces

Using stacked notation

$$\mathfrak{f}_{ext} = \text{col}\left\{\mathfrak{f}_{ext}^i(k)\right\} \quad \in \mathcal{R}^{6n_{nd}}$$

$$\mathfrak{f} = \mathcal{E}_{\phi}\mathfrak{f} - \boxed{\mathcal{B}\mathfrak{f}_{ext}} + \mathbf{M}\alpha + \mathfrak{b}$$

$$\mathfrak{f} = \phi\left(\mathbf{M}\alpha + \mathfrak{b} - \boxed{\mathcal{B}\mathfrak{f}_{ext}}\right)$$



Equations of motion with external forces

$$\mathfrak{f} = \phi\left(\mathbf{M}\alpha + \mathfrak{b} - \boxed{\mathcal{B}\mathfrak{f}_{\mathrm{ext}}}\right)$$

The equations of motion thus take the form

$$\mathcal{T} = \mathcal{M}\ddot{\theta} + \mathcal{C} - \boxed{H\phi\mathcal{B}\mathfrak{f}_{\mathrm{ext}}} \overset{3.53}{=} \mathcal{M}\ddot{\theta} + \mathcal{C} - \mathcal{J}^*\mathfrak{f}_{\mathrm{ext}}$$

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

Jacobian matrix

Can book-keep external forces in Coriolis term

$$\mathcal{C}(\theta, \dot{\theta}) = H\phi[\mathbf{M}\phi^*\mathfrak{a} + \mathfrak{b} - \mathcal{B}\mathfrak{f}_{\mathrm{ext}}]$$

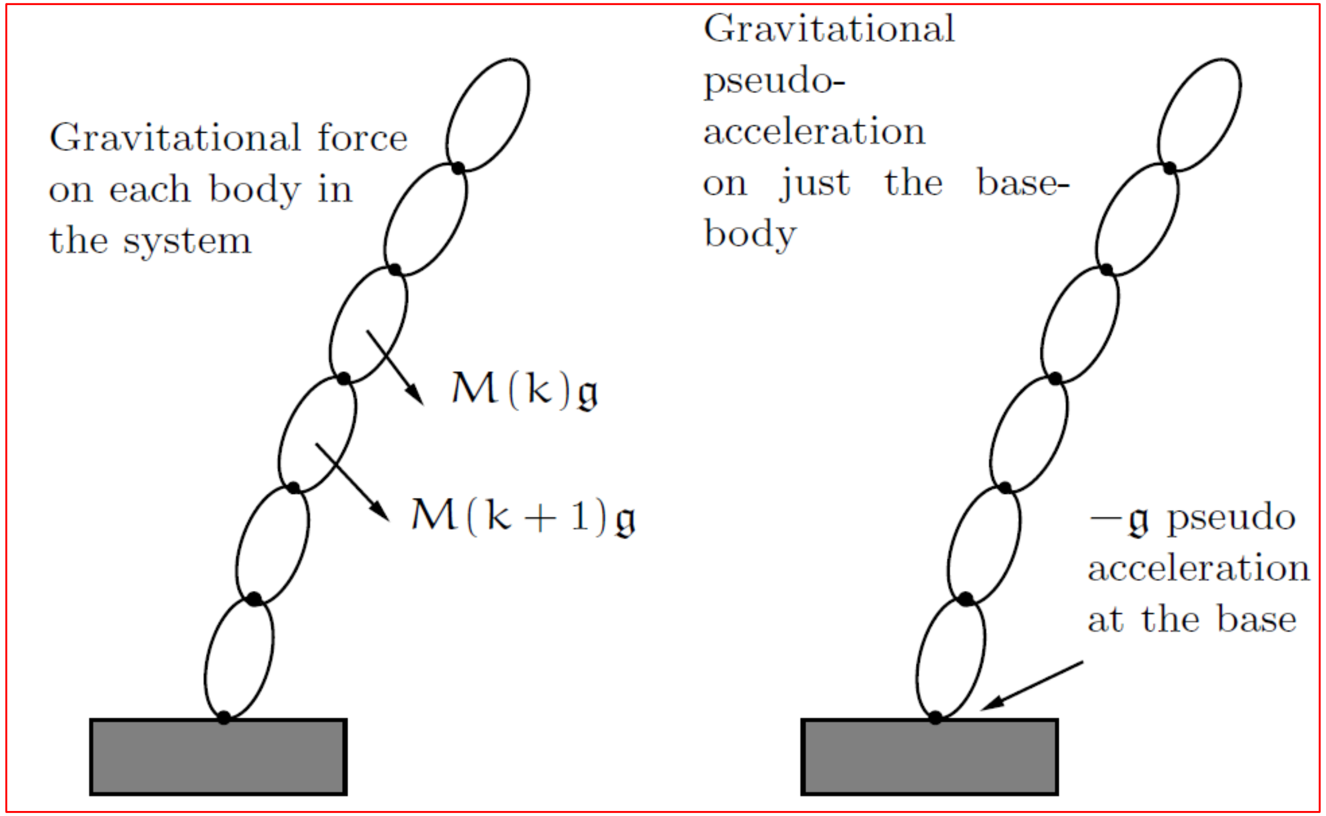
Including external gravity

Handling gravity

Gravity effects can be handled as **external forces** or as a **pseudo acceleration**

gravitational
spatial accel

$$\mathfrak{g} = \begin{bmatrix} 0 \\ \mathfrak{g}_l \end{bmatrix}$$



Using pseudo-accelerations

Including gravity effect in the Coriolis vector

$$\mathfrak{C}(\theta, \dot{\theta}) = H\phi\left[\mathbf{M}\phi^*(\mathfrak{a} + \boxed{E^*\mathfrak{g}}) + \mathfrak{b}\right]$$

using the pick-off stacked vector

$$E \triangleq [\mathbf{0}_6, \cdots \mathbf{0}_6, \mathbf{I}_6] \in \mathcal{R}^{6\times 6n}$$

Forward and Inverse Dynamics

Inverse and Forward dynamics

Inverse dynamics:

- Given the state, and generalized accelerations, use the equations of motion to compute the generalized forces

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

- Important for feedforward control applications

Forward dynamics:

- Given the state, and generalized forces, solve the equations of motion to compute the generalized accelerations

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

- Important for simulation applications

Inverse Dynamics

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?

Exploiting Newton-Euler factorization for computing $\mathcal{M}(\theta)\ddot{\theta}$

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

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

***diagonal** matrix times vector*

*recursive **scatter** alg.*

***diagonal** matrix times vector*

*recursive **gather** alg.*

***diagonal** matrix times vector*

Another example of being able to directly map operator expressions into low-cost recursive algorithms

Comptuting Coriolis term $\mathcal{C}(\theta, \dot{\theta})$

$\mathcal{C}(\theta, \dot{\theta})$ can also be computed using a sequence of scatter and gather O(N) recursions

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

Moreover we can combine the O(N) recursions into a single sequence of scatter and gather recursions.

Newton-Euler O(N) Recursive Inverse Dynamics

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
```

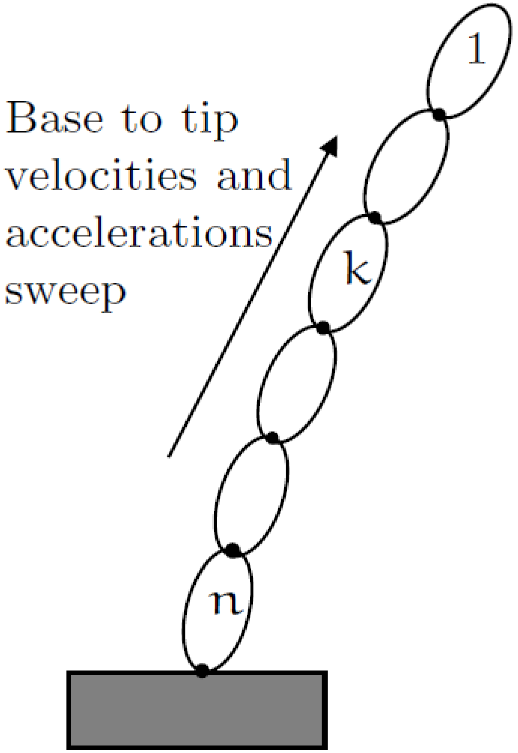
*Base-to-tip O(N) recursive **scatter** sweep*

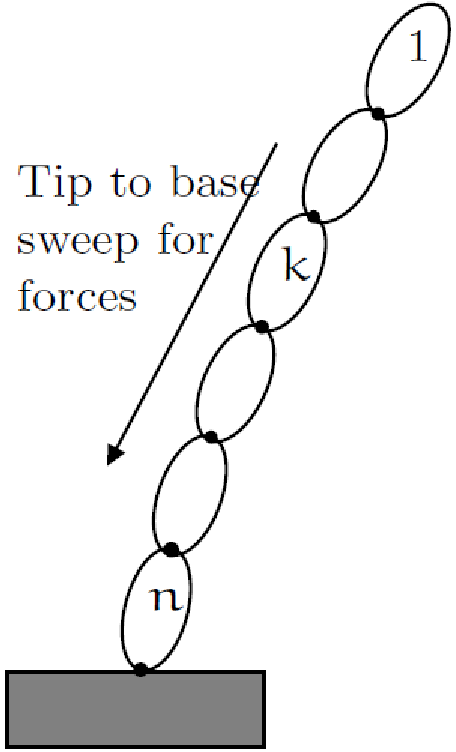
```
$\mathfrak{f}(0) = \mathbf{0}$
for $k$  $1 \cdots n$
    $\mathfrak{f}(k) = \phi(k,k-1)\mathfrak{f}(k-1) + M(k)\alpha(k) + \mathfrak{b}(k)$
    $\mathcal{T}(k) = H(k)\mathfrak{f}(k)$
end loop
```

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

Inverse dynamics algorithm structure

Sequence of scatter and gather O(N) recursive sweeps





Including external forces and gravity in inverse dynamics

Simply update the spatial forces step in the inverse dynamics algorithm to handle external forces and gravity

$$
\begin{aligned}
\mathfrak{f}(k) = \phi(k, k-1)\mathfrak{f}(k-1) + M(k)\big(\alpha(k) + \mathfrak{g}\big) + \mathfrak{b}(k) \\
- \sum_{i} \phi(\mathbb{B}_k, \mathbb{O}^i_k)\mathfrak{f}^i_{ext}(k)
\end{aligned}
$$

SOA based O(N) Inverse Dynamics

Spatial Operator Algebra (SOA) *based mathematical analysis*

Mapping to ***structure based****, recursive algorithms*

Dynamics properties → Transformed Expressions → Low-order structure-based algorithms

$$\mathcal{T} = \mathcal{M}\ddot{\theta} + \mathcal{C} \Rightarrow \mathcal{M} = \mathrm{H}\phi\mathbf{M}\phi^{*}\mathrm{H}^{*} \Rightarrow$$

O(N) scatter + gather recursions

Later – The scatter/gather form of the O(N) NE inverse dynamics algorithm holds for any tree/branched system.

Newton-Euler Inverse Dynamics Algorithm

Mass matrix using inverse dynamics

Using inverse dynamics for the mass matrix

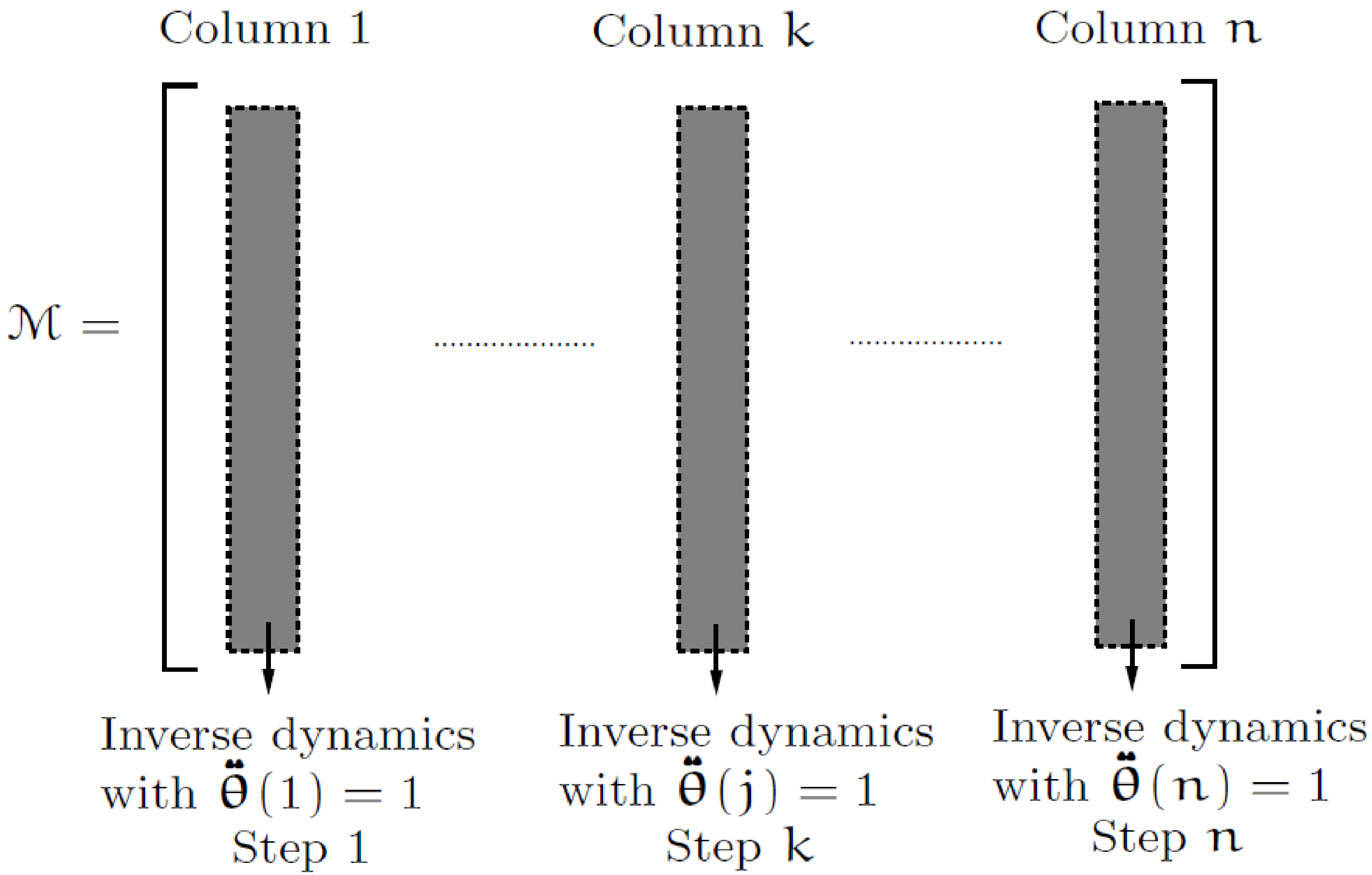
The equations of motion are

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

- The Coriolis vector is zero when velocities, external forces and gravity are zero.
- Inverse dynamics with all-zero generalized accel except kth element being 1 yields the kth column of the mass matrix
- Repeat this procedure for each element of the generalized accels to get the full mass matrix
- The CRB-based algorithm is however faster

Structure of the algorithm

Algorithm consists of a sequence of inverse dynamics computations



Inverse dynamics using CRBs

Inverse dynamics revisited

- Earlier we exploited the Newton-Euler factorization of the mass matrix to develop the O(N) inverse dynamics algorithm

$$\mathcal{M}(\theta) = H\phi\mathbf{M}\phi^*H^* \in \mathcal{R}^{\mathcal{N}\times\mathcal{N}}$$

- We also developed a CRBs based decomposition of the mass matrix

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

- We now use CRBs to develop an alternative inverse dynamics algorithm

Alternative expression for forces

Have

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

and

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

Thus

$$\begin{aligned} \mathfrak{f} &\overset{5.23}{=} \phi[\mathbf{M}\phi^*(H^*\dot{\theta} + \mathfrak{a}) + \mathfrak{b}] \overset{4.10}{=} (\tilde{\phi}\mathcal{R} + \mathcal{R}\phi^*)(H^*\ddot{\theta} + \mathfrak{a}) + \phi\mathfrak{b} \\ &\overset{5.23}{=} \mathcal{R}\alpha + \phi\left[\mathfrak{b} + \mathcal{E}_{\phi}\mathcal{R}(H\ddot{\theta} + \mathfrak{a})\right] \end{aligned}$$

$$\boxed{\mathfrak{f} = \mathcal{R}\alpha + \mathfrak{y}} \qquad \boxed{\mathfrak{y} \triangleq \phi\left[\mathfrak{b} + \mathcal{E}_{\phi}\mathcal{R}(H^*\ddot{\theta} + \mathfrak{a})\right]}$$

CRB-based Inverse dynamics algorithm

$$\mathfrak{f} = \mathcal{R}\alpha + \mathfrak{y} \qquad\qquad \mathfrak{y} \overset{\triangle}{=} \phi\left[\mathfrak{b} + \mathcal{E}_{\phi}\mathcal{R}(H^{*}\ddot{\theta} + \mathfrak{a})\right]$$

- Use CRB gather algorithm to compute the CRB spatial inertias
- Compute the y values via a gather algorithm

```
$\mathfrak{y}^{+}(0) = \mathbf{0}$
for $k$   $1 \cdots n$
    $\mathfrak{y}(k) = \phi(k, k-1)\mathfrak{y}^{+}(k-1) + \mathfrak{b}(k)$
    $\mathfrak{y}^{+}(k) = \mathfrak{y}(k) + \mathcal{R}(k)\left[H^{*}(k)\ddot{\theta}(k) + \mathfrak{a}(k)\right]$
end loop
```

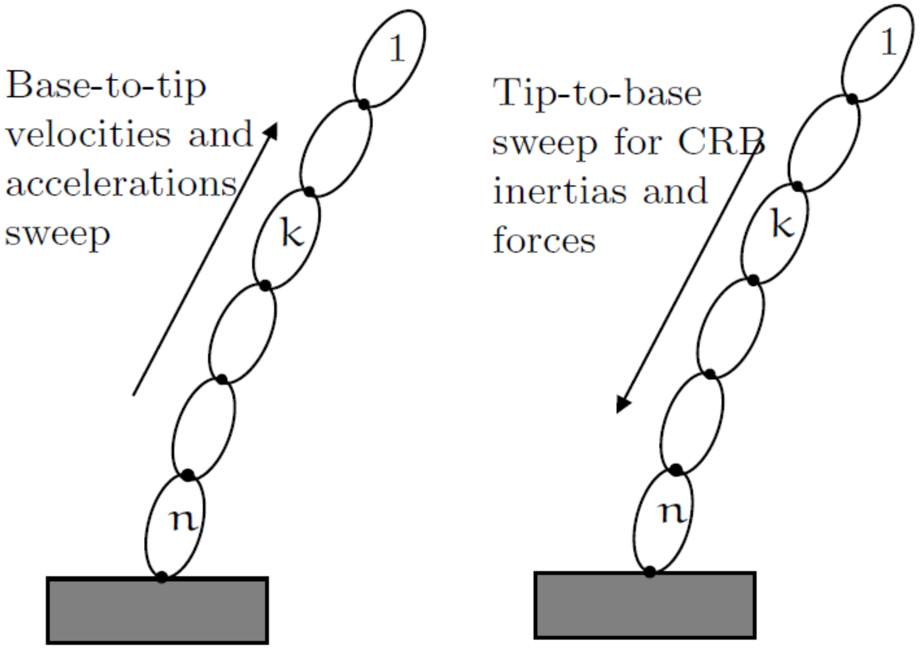
Another example of directly mapping operator expressions into low-cost recursive algorithms

- Compute the generalized forces

$$\mathcal{T}(k) \overset{5.21}{=} H(k)\mathfrak{f}(k) \overset{5.44}{=} H(k)\left[\mathcal{R}(k)\alpha(k) + \mathfrak{y}(k)\right]$$

Structure of the algorithm

- This CRB-based algorithm also has O(N) computational cost



- But is more expensive compared to the O(N) NE inverse dynamics algorithm

Inter-body spatial force decompositions

- Ignore Coriolis terms for the moment
- From the equations of motion we had

$$\mathfrak{f}(k) = \underbrace{M(k)}_{\text{depends on \textbf{kth} body}}\alpha(k) + \phi(k, k-1)\underbrace{\mathfrak{f}(k-1)}_{\text{depends on \textbf{all} bodies}}$$

- Using CRBs we have the alternative expression

$$\mathfrak{f}(k) = \underbrace{\mathcal{R}(k)}_{\text{depends on \textbf{outboard} bodies only}}\alpha(k) + \underbrace{\mathfrak{y}(k)}_{\text{depends on \textbf{outboard} generalized accels}}$$

- The more complex **inertia** term simplifies the **residual** term in the force decompositions
- We will see more such decompositions later

Equations of motion using inertial reference frame

Inertially referenced spatial velocities

- We have used alternative choices for body spatial velocity and acceleration – including the inertially reference versions

$$\mathcal{V}_{\mathbb{I}}(k) = \phi^*(\mathbb{O}_k, \mathbb{I})\mathcal{V}(\mathbb{O}_k) = \begin{bmatrix}\omega(k)\\ \nu_{\mathbb{I}}(k)\end{bmatrix}$$

- We can use these for the equations of motion to derive the following:

$$\begin{aligned}\mathcal{M}(\theta) &\triangleq H_{\mathbb{I}}\phi_{\mathbb{I}}\mathbf{M}_{\mathbb{I}}\phi_{\mathbb{I}}^*H_{\mathbb{I}}^*\\ \mathcal{C}(\theta,\dot{\theta}) &\triangleq H_{\mathbb{I}}\phi_{\mathbb{I}}\frac{d[\mathbf{M}_{\mathbb{I}}\phi_{\mathbb{I}}^*H_{\mathbb{I}}^*]}{dt}\dot{\theta} = H_{\mathbb{I}}\phi_{\mathbb{I}}[\dot{\mathbf{M}}_{\mathbb{I}}\mathcal{V}_{\mathbb{I}} + \mathbf{M}_{\mathbb{I}}\phi_{\mathbb{I}}^*\dot{\mathbf{H}}_{\mathbb{I}}^*\dot{\theta}]\end{aligned}$$

Observations on new equations of motion

- The component operators are different

$$\mathcal{E}_{\mathbb{I}} = \begin{pmatrix} 0 & \dots & \dots & \dots & 0 \\ \mathbf{I} & 0 & \dots & \dots & \vdots \\ 0 & \mathbf{I} & 0 & \dots & \vdots \\ \vdots & & \ddots & \ddots & \vdots \\ 0 & \dots & \dots & \mathbf{I} & 0 \end{pmatrix} \qquad \phi_{\mathbb{I}} = \begin{pmatrix} \mathbf{I} & \dots & \dots & \mathbf{0} \\ \mathbf{I} & \mathbf{I} & & \vdots \\ \vdots & & \ddots & \vdots \\ \mathbf{I} & \dots & \dots & \mathbf{I} \end{pmatrix}$$

- The mass matrix and Coriolis vector however remain unchanged

SHOW!

$$\mathcal{M}(\theta) \triangleq \mathrm{H}_{\mathbb{I}}\phi_{\mathbb{I}}\mathbf{M}_{\mathbb{I}}\phi_{\mathbb{I}}^{*}\mathrm{H}_{\mathbb{I}}^{*}$$

$$\mathcal{C}(\theta, \dot{\theta}) \triangleq \mathrm{H}_{\mathbb{I}}\phi_{\mathbb{I}}\frac{\mathrm{d}[\mathbf{M}_{\mathbb{I}}\phi_{\mathbb{I}}^{*}\mathrm{H}_{\mathbb{I}}^{*}]}{\mathrm{d}t}\dot{\theta} = \mathrm{H}_{\mathbb{I}}\phi_{\mathbb{I}}\left[\dot{\mathbf{M}}_{\mathbb{I}}\mathcal{V}_{\mathbb{I}} + \mathbf{M}_{\mathbb{I}}\phi_{\mathbb{I}}^{*}\dot{\mathrm{H}}_{\mathbb{I}}^{*}\dot{\theta}\right]$$

Observations

Kane’s partial velocities

In the operator expressions

$$\mathcal{V} = \phi^* H^* \dot{\theta} \quad \text{and} \quad \mathcal{M}(\theta) = H\phi \mathbf{M} \phi^* H^*$$

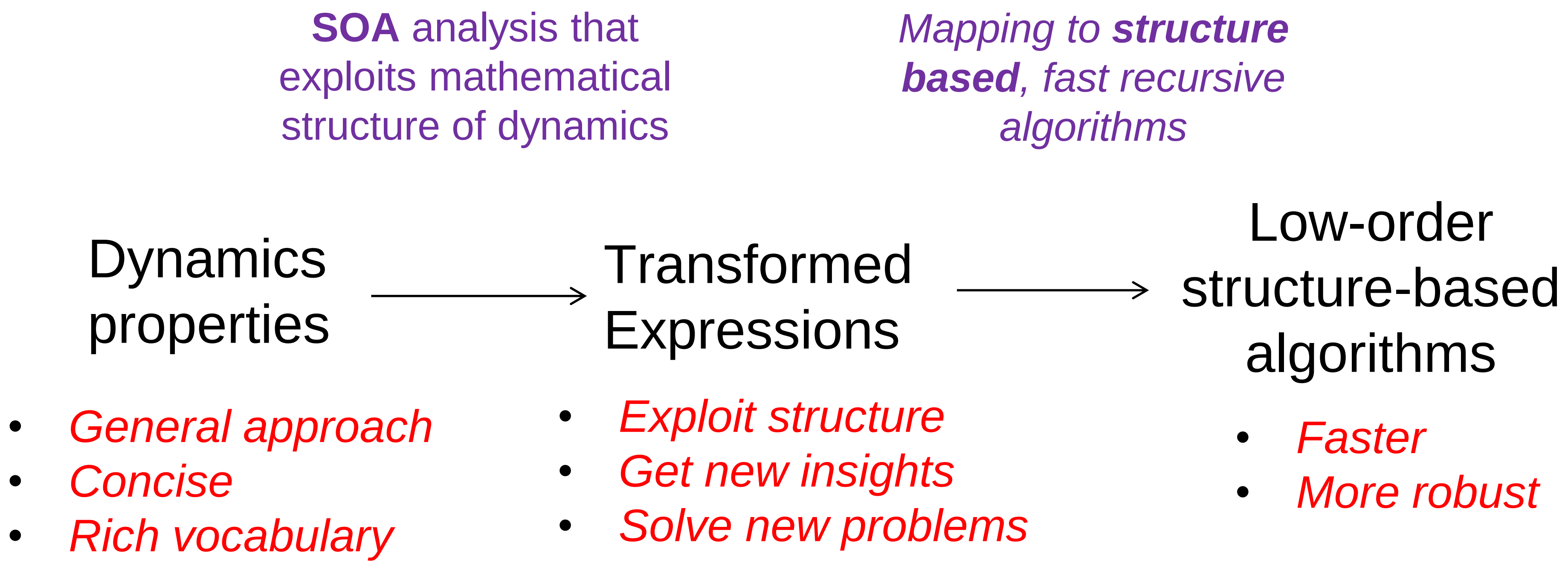
the

$$\phi^* H^*$$

matrix elements are the “partial velocities” from Kane’s method.

- We would neither get the CRB decomposition of the mass matrix or the recursive NE inverse dynamics structure if we evaluated $\phi^* H^*$ into a partial velocities matrix!
- The operator form is key to preserving structure.

Transforming SOA operator expressions into recursive algorithms



SOA analysis that exploits mathematical structure of dynamics

*Mapping to **structure based**, fast recursive algorithms*

Dynamics properties

- *General approach*
- *Concise*
- *Rich vocabulary*

Transformed Expressions

- *Exploit structure*
- *Get new insights*
- *Solve new problems*

Low-order structure-based algorithms

- *Faster*
- *More robust*

“Structure-based”: Because the pattern of the recursive algorithms is entirely driven by the underlying multibody topology.

Summary

- Developed Newton-Euler factorization of the mass matrix
- Introduced CRB inertias for the decomposition of the mass matrix and its $O(\mathcal{N}^2)$ computation
- Developed operator form of system equations of motion
- Developed O(N) Newton-Euler inverse dynamics algorithm
- Explored inverse dynamics based computation of mass matrix, and CRB based inverse dynamics and force decompositions

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