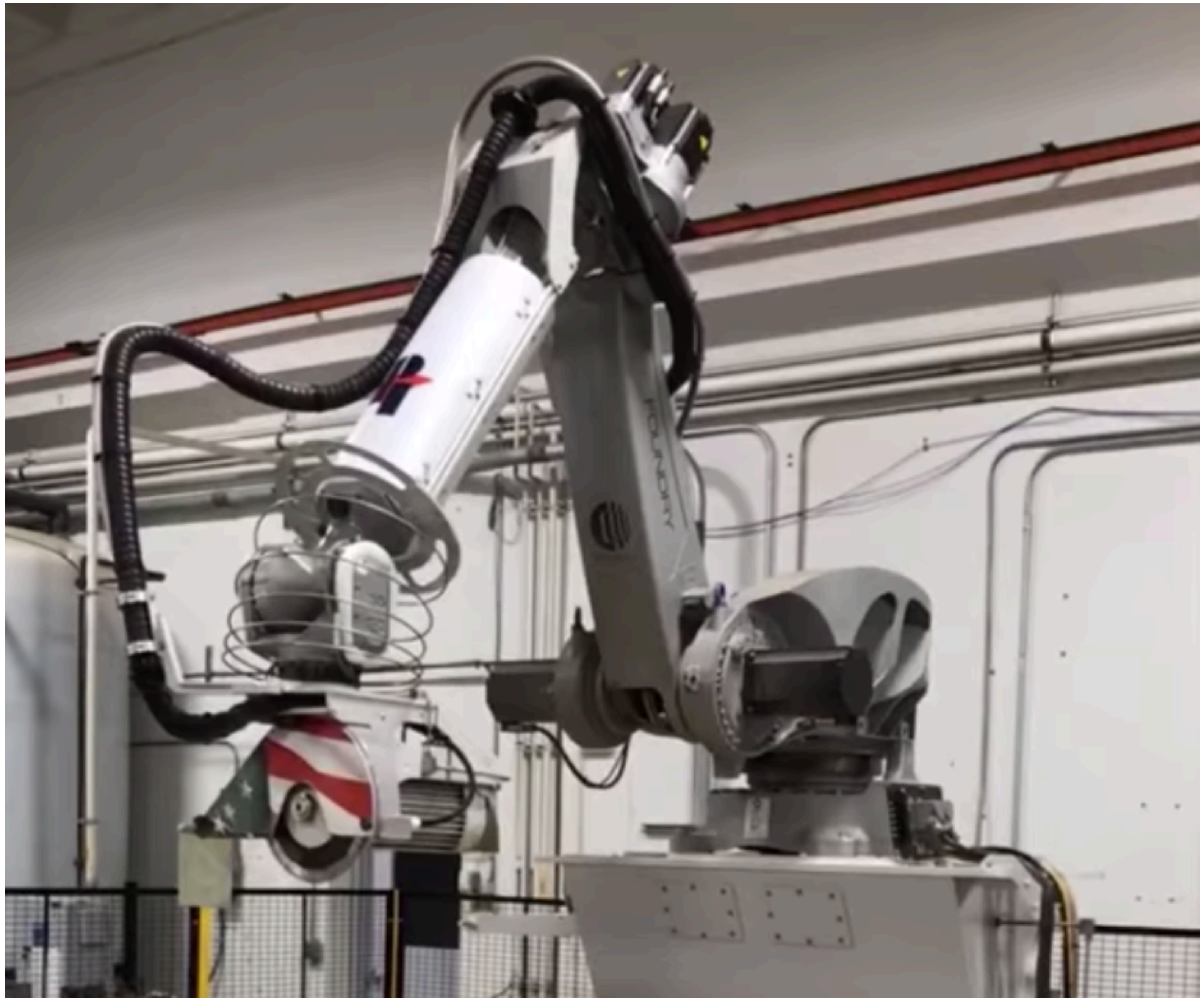
Recursive Newton-Euler

- Modern version of Rigid body dynamics
- Connect links via balance equations
- Express as a factor graph
- Solve

- Closely follows Lynch & Park 2017
- Factor graph story with Mandy Xie

Motivation



https://www.youtube.com/watch?v=1nNQVsvb8TQ

Accelerating a Rigid Body from Rest

- Really just F=ma, but translation and rotation:

$$f_b = m\dot{v}_b$$

$$\tau_b = \mathcal{I}_b\dot{\omega}_b$$

$$\mathcal{F}_b = \mathcal{G}_b\dot{\mathcal{V}}_b$$

$$\mathcal{F}_b = \begin{bmatrix} \tau_b \\ f_b \end{bmatrix}$$

$$\mathcal{V}_b = \begin{bmatrix} \omega_b \\ v_b \end{bmatrix}$$

$$\mathcal{G}_b = \begin{bmatrix} \mathcal{I}_b & \\ & mI \end{bmatrix}$$

Rigid Body in motion

- Adds Coriolis terms, i.e. squared velocity quantities:

$$f_b = m\dot{v}_b - m\hat{\omega}_b^T v_b$$

$$\tau_b = \mathcal{I}_b\dot{\omega}_b - \hat{\omega}_b^T \mathcal{I}_b \omega_b$$

$$\mathcal{F}_b = \mathcal{G}_b \dot{\mathcal{V}}_b - \left[\mathrm{ad}_{\mathcal{V}_b}\right]^T \mathcal{G}_b \mathcal{V}_b$$

$$[\mathrm{ad}_{\mathcal{V}_b}] = \begin{bmatrix} \hat{\omega}_b & \\ \hat{v}_b & \hat{\omega}_b \end{bmatrix}$$

Dynamics in another frame

- use Adjoint (capital A):

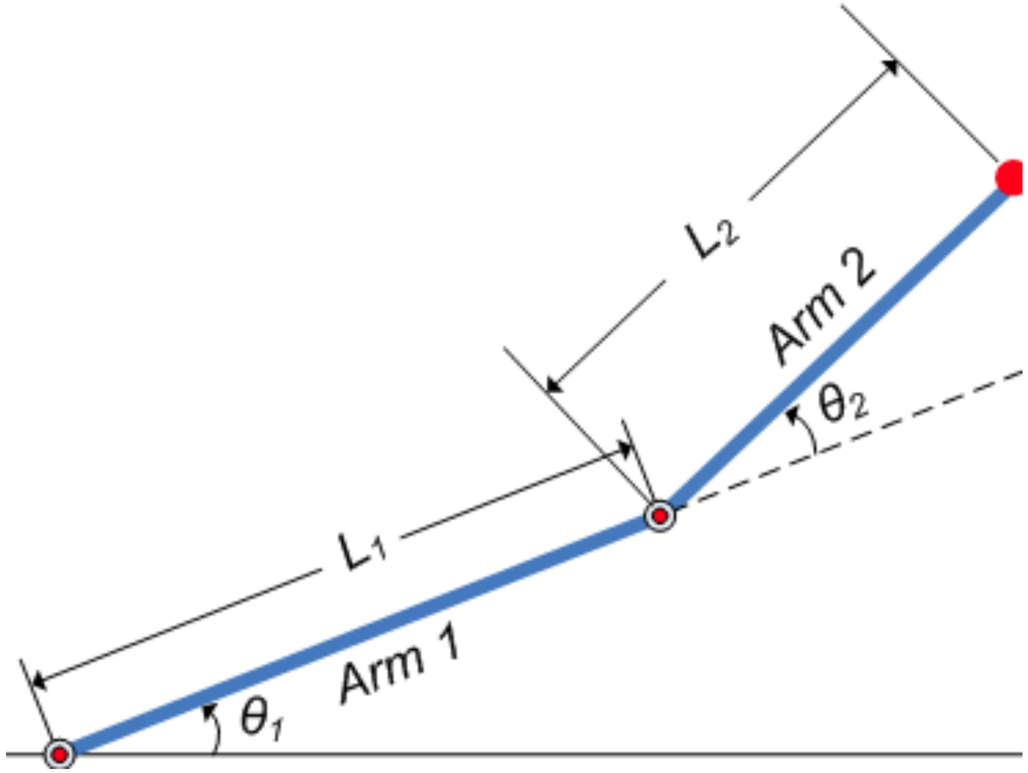
$$\mathcal{G}_a = [\mathrm{Ad}_{T_{ba}}]^{\mathrm{T}} \mathcal{G}_b [\mathrm{Ad}_{T_{ba}}]$$

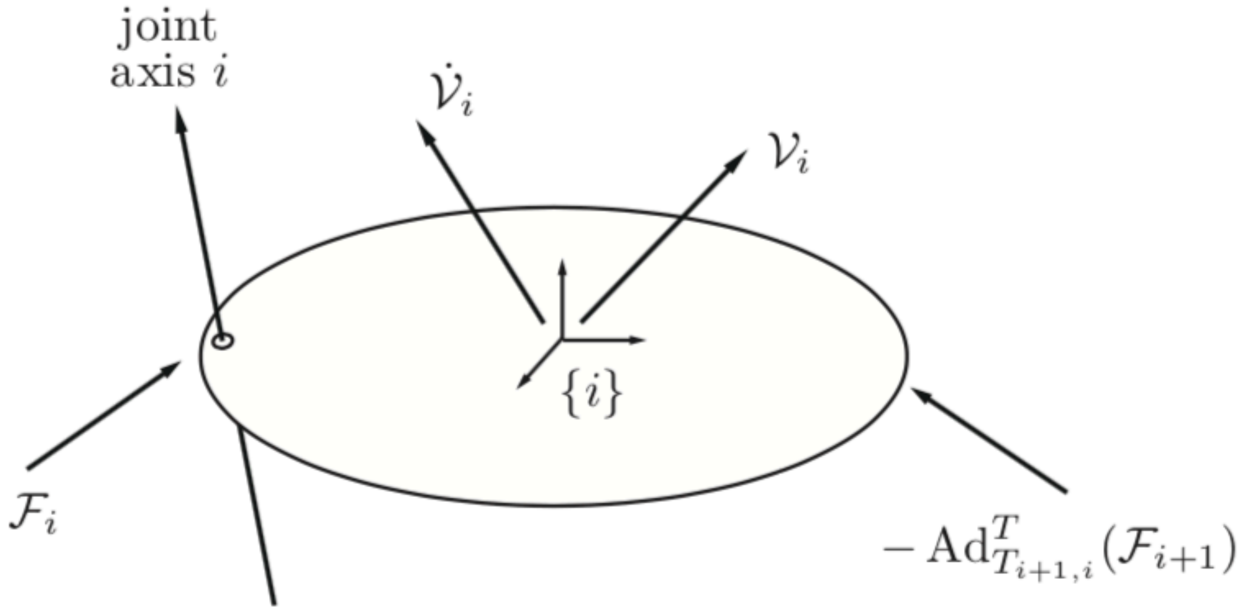
$$\mathcal{F}_a = \mathcal{G}_a \dot{\mathcal{V}}_a - [\mathrm{ad}_{\mathcal{V}_a}]^T \mathcal{G}_a \mathcal{V}_a$$

- where $[\mathrm{Ad}_T] = \begin{bmatrix} R & \\ \hat{t}R & R \end{bmatrix}$

Newton-Euler Dynamics

- Fix body frame to every link (doesn't have to be COM)
- Define axis A_i in link frame
- Wrench F_i, transmitted through i





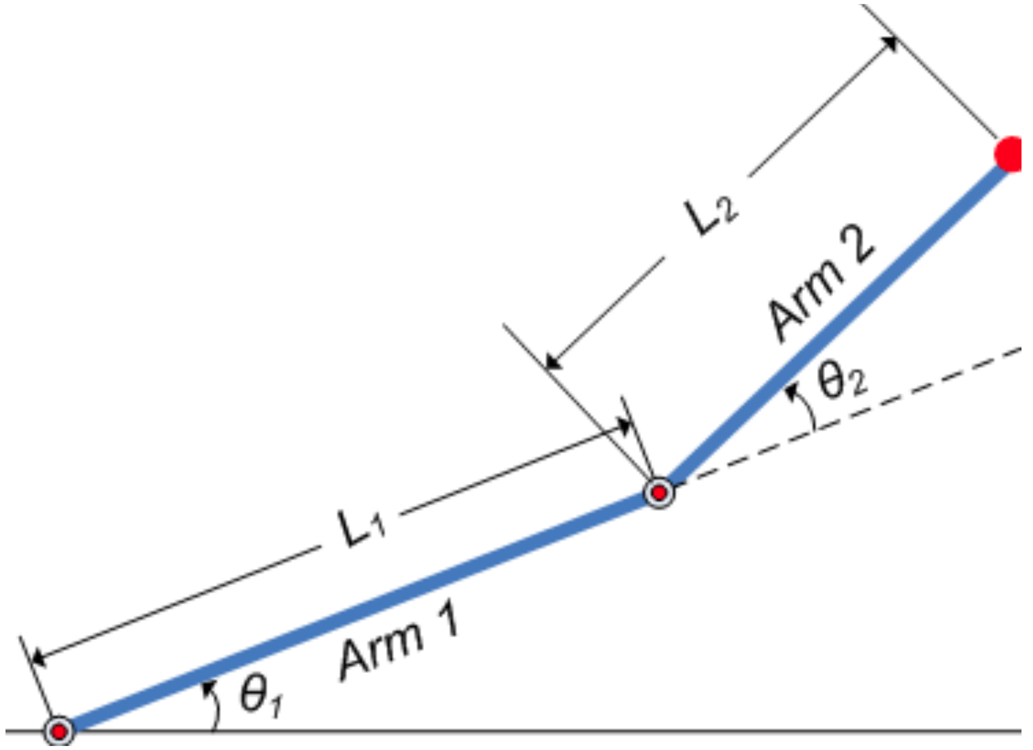
Forward pass: twists

- Twist of link i:

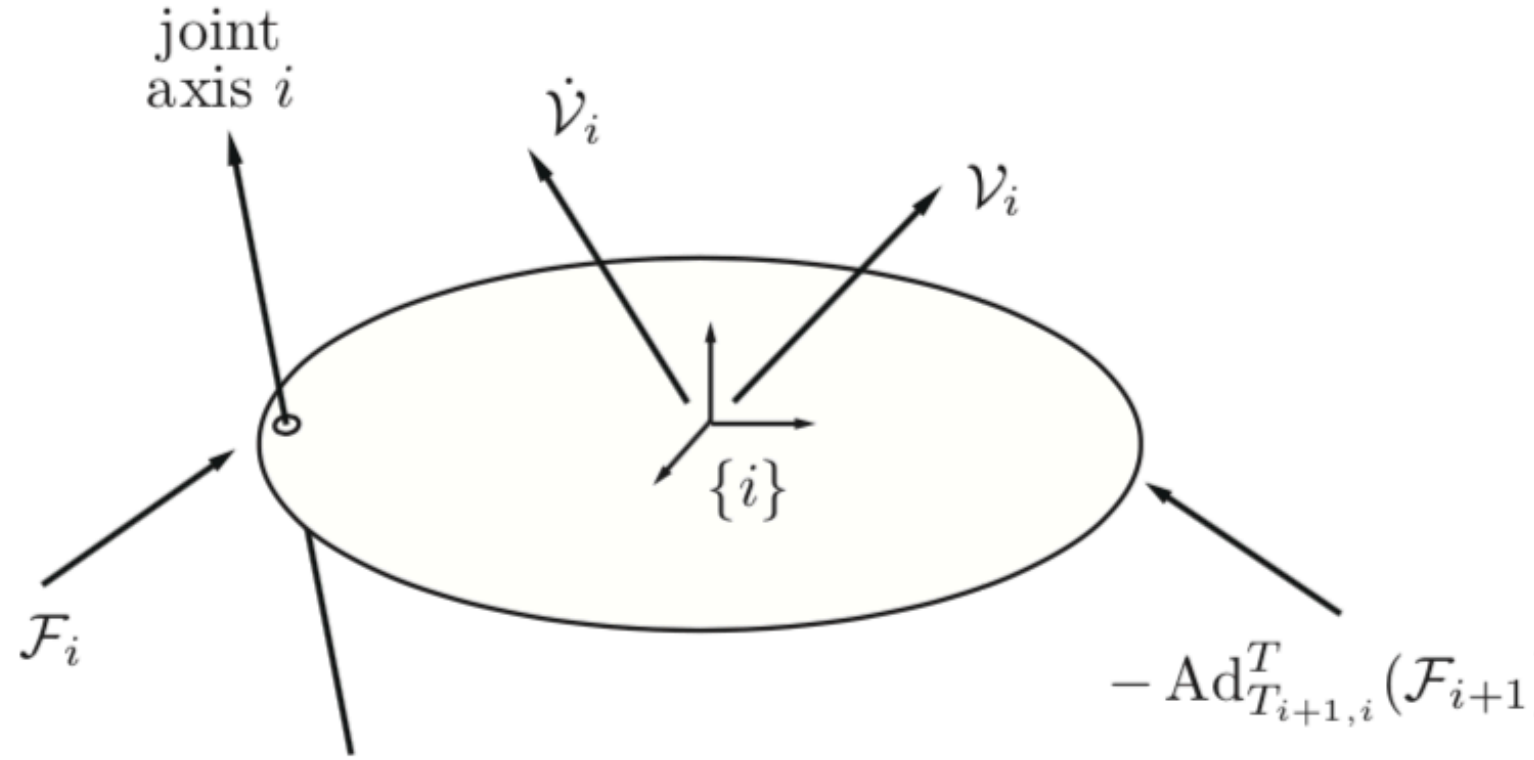
$$\mathcal{V}_i = \mathcal{A}_i\dot{\theta}_i + [\mathrm{Ad}_{T_{i,i-1}}]\mathcal{V}_{i-1}$$

- Take time derivative:

$$\dot{\mathcal{V}}_i = \mathcal{A}_i\ddot{\theta}_i + [\mathrm{Ad}_{T_{i,i-1}}]\dot{\mathcal{V}}_{i-1} + [\mathrm{ad}_{\mathcal{V}_i}]\mathcal{A}_i\dot{\theta}_i$$



Wrenches

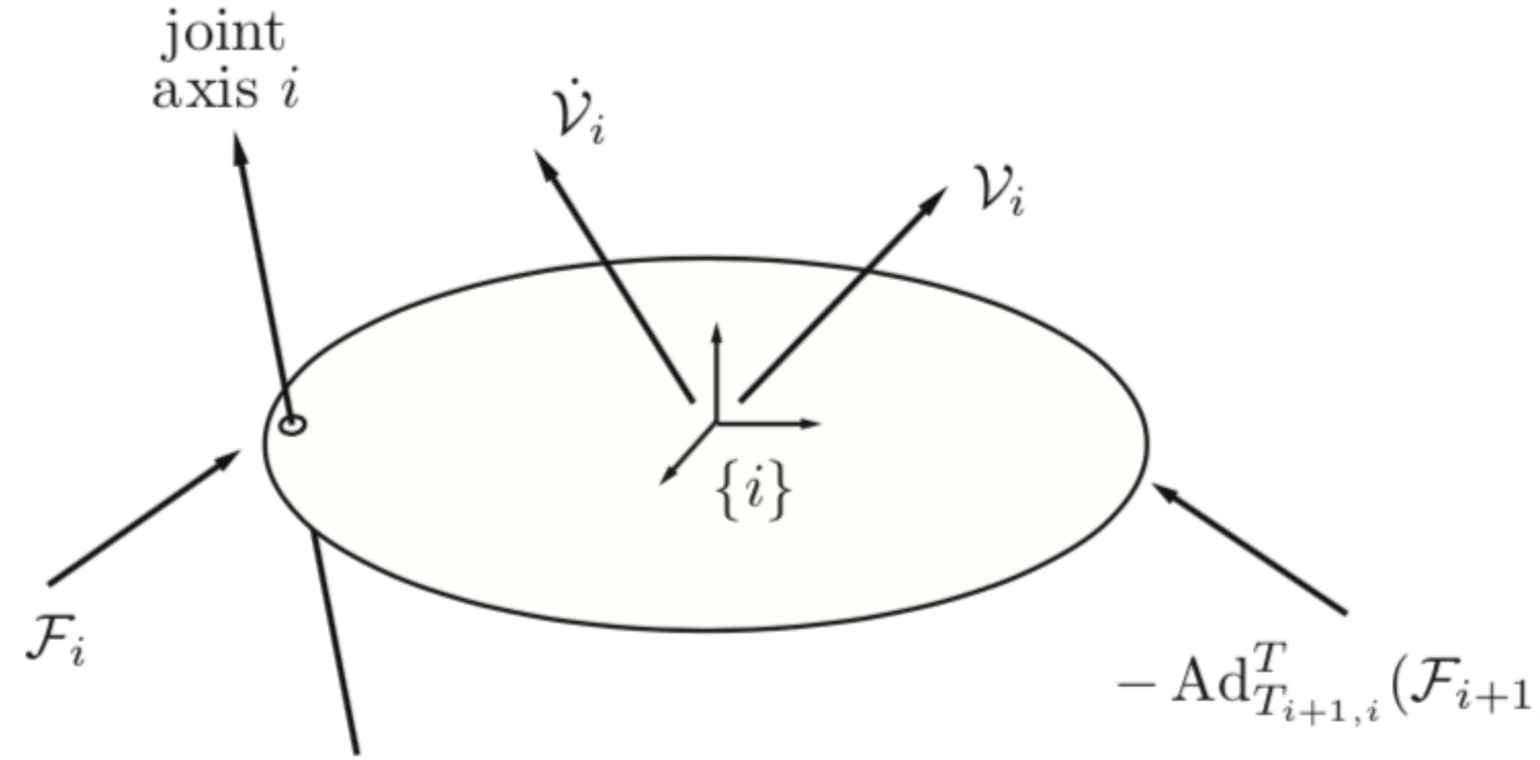


- All twists and accelerations can be computed from link 0 to link n: forward pass!
- Backward pass: calculate wrenches:

$$\mathcal{G}_i\dot{\mathcal{V}}_i - \mathrm{ad}^{\mathrm{T}}_{\mathcal{V}_i}(\mathcal{G}_i\mathcal{V}_i) = \mathcal{F}_i - \mathrm{Ad}^{\mathrm{T}}_{T_{i+1,i}}(\mathcal{F}_{i+1})$$

- Just rigid body dynamics!
- RHS: wrench on link i plus reaction wrench

Torques



- Recall:

$$\mathcal{G}_i\dot{\mathcal{V}}_i - \mathrm{ad}^{\mathrm{T}}_{\mathcal{V}_i}(\mathcal{G}_i\mathcal{V}_i) = \mathcal{F}_i - \mathrm{Ad}^{\mathrm{T}}_{T_{i+1,i}}(\mathcal{F}_{i+1})$$

- 1 DOF of 6 DOF wrench F_i is delivered by torque:

$$\tau_i = \mathcal{F}_i^{\mathrm{T}}\mathcal{A}_i$$

Resulting “classical” RNEA:

- Inverse dynamics: *given joint accelerations, calculate torques*

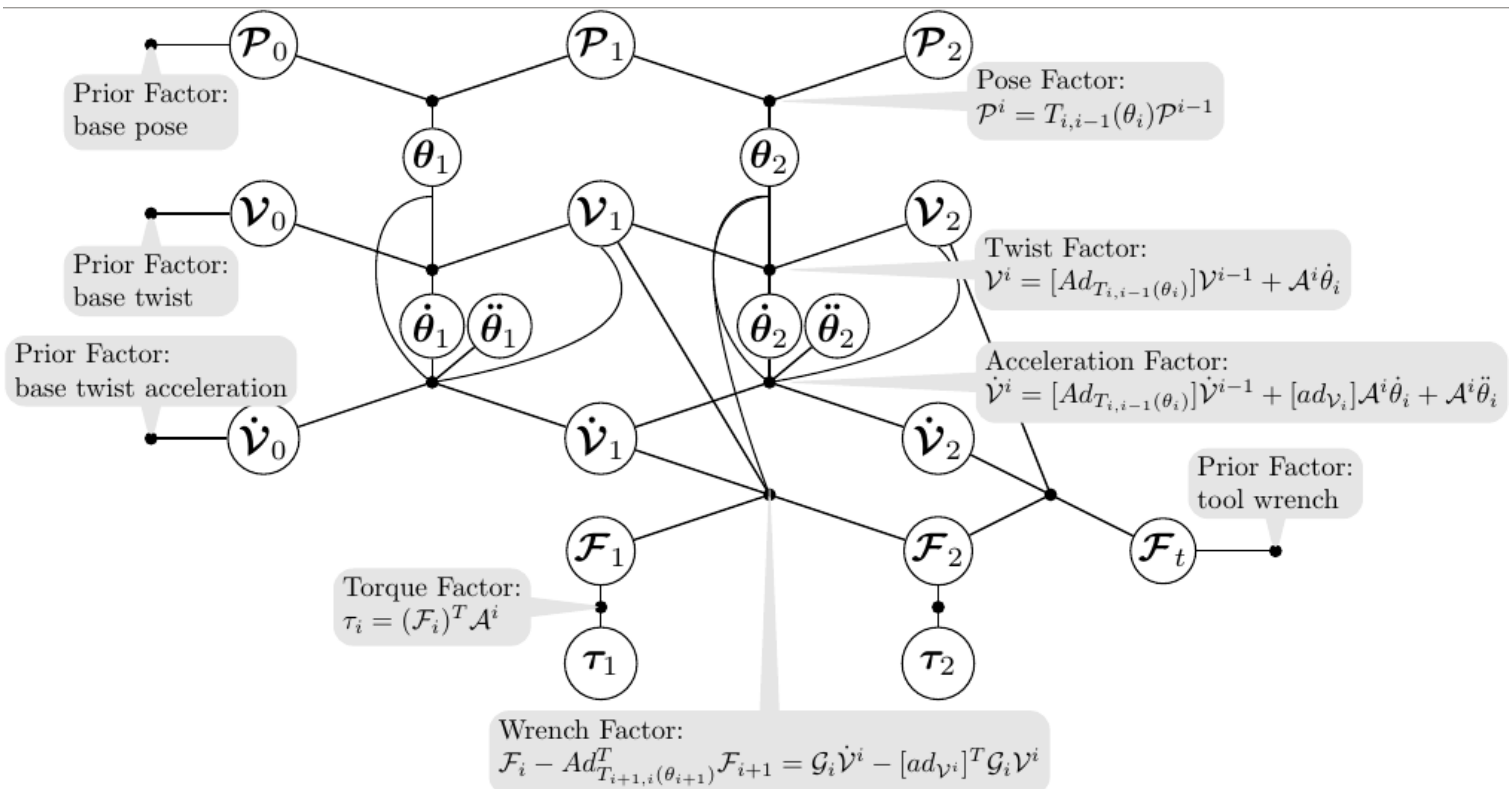
Forward iterations Given $\theta, \dot{\theta}, \ddot{\theta}$, for $i = 1$ to n do

$$
\begin{aligned}
T_{i,i-1} &= e^{-[\mathcal{A}_i]\theta_i} M_{i,i-1}, \\
\mathcal{V}_i &= \mathrm{Ad}_{T_{i,i-1}}(\mathcal{V}_{i-1}) + \mathcal{A}_i \dot{\theta}_i, \\
\dot{\mathcal{V}}_i &= \mathrm{Ad}_{T_{i,i-1}}(\dot{\mathcal{V}}_{i-1}) + \mathrm{ad}_{\mathcal{V}_i}(\mathcal{A}_i)\dot{\theta}_i + \mathcal{A}_i \ddot{\theta}_i.
\end{aligned}
$$

Backward iterations For $i = n$ to 1 do

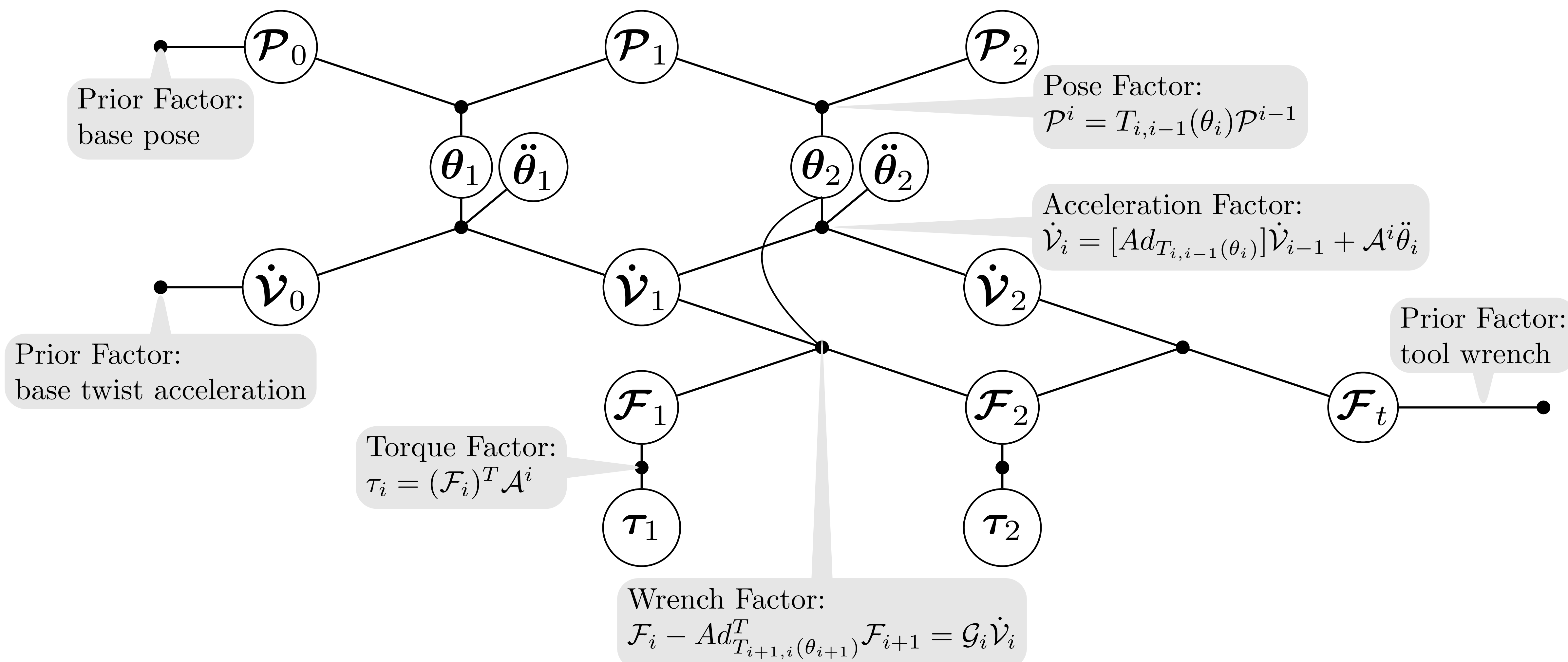
$$
\begin{aligned}
\mathcal{F}_i &= \mathrm{Ad}^{\mathrm{T}}_{T_{i+1,i}}(\mathcal{F}_{i+1}) + \mathcal{G}_i \dot{\mathcal{V}}_i - \mathrm{ad}^{\mathrm{T}}_{\mathcal{V}_i}(\mathcal{G}_i \mathcal{V}_i), \\
\tau_i &= \mathcal{F}_i^{\mathrm{T}} \mathcal{A}_i.
\end{aligned}
$$

RR-link manipulator dynamics factor graph



RR at rest (no Coriolis)

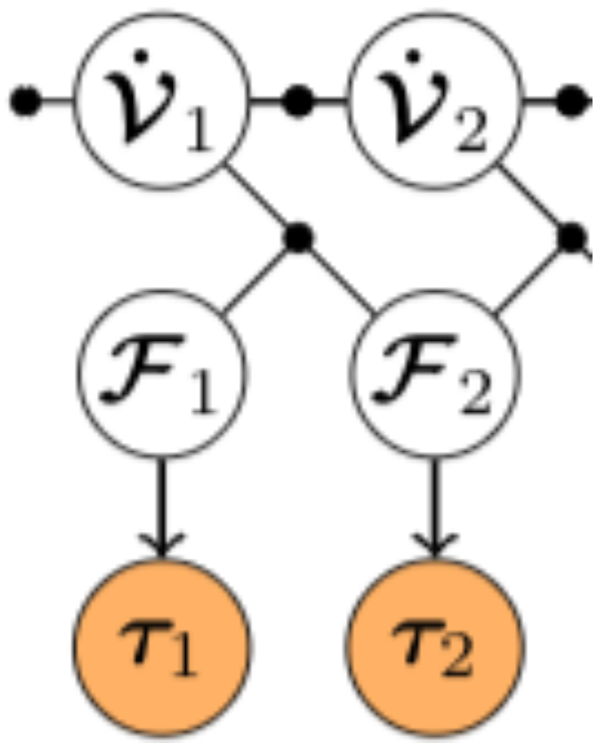
- Makes it a bit easier to see basic structure

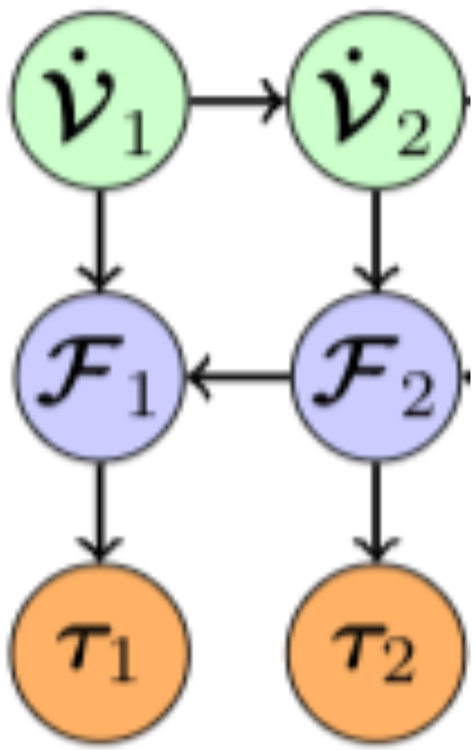


F=ma

Elimination

- First torques, then wrenches, then twists:





1 $\dot{\mathcal{V}}_1 = \mathcal{A}_1\ddot{\theta}_1$;

2 $\dot{\mathcal{V}}_2 = [Ad_{T_{2,1}}]\dot{\mathcal{V}}_1 + \mathcal{A}_2\ddot{\theta}_2$;

3 $\mathcal{F}_2 = Ad^T_{T_{t,2}}\mathcal{F}_t + \mathcal{G}_2\dot{\mathcal{V}}_2$;

4 $\mathcal{F}_1 = Ad^T_{T_{2,1}}\mathcal{F}_2 + \mathcal{G}_1\dot{\mathcal{V}}_1$;

5 $\tau_1 = \mathcal{F}_1^T\mathcal{A}_1$;

6 $\tau_2 = \mathcal{F}_2^T\mathcal{A}_2$;

Back to Mass Matrix

- Do *symbolic* back-substitution:

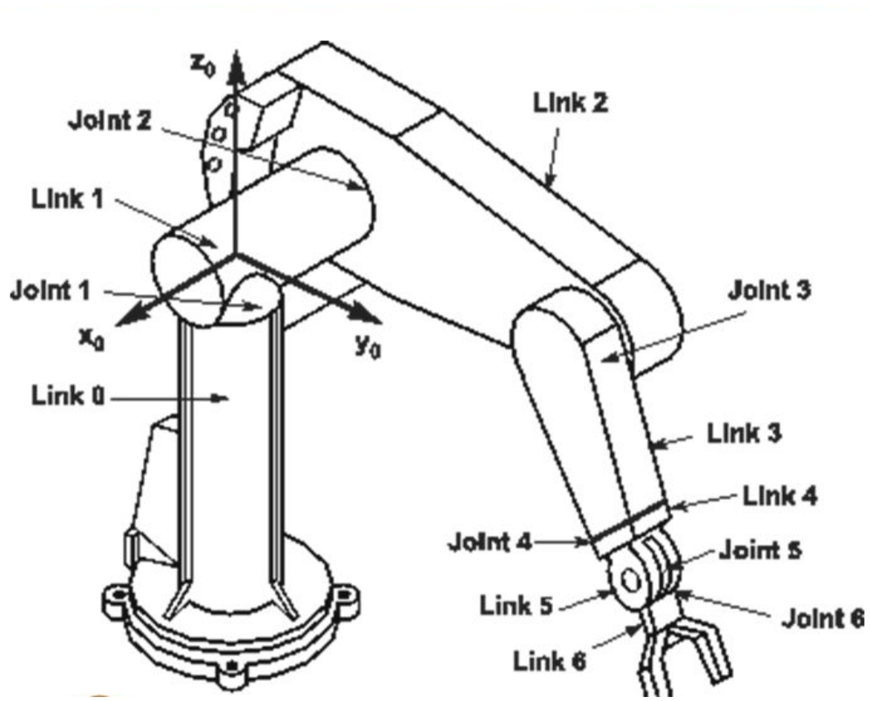
$$
\begin{aligned}
\dot{\mathcal{V}}_1 &= \mathcal{A}_1 \ddot{\theta}_1 \;; \\
\dot{\mathcal{V}}_2 &= [Ad_{T_{2,1}}] \dot{\mathcal{V}}_1 + \mathcal{A}_2 \ddot{\theta}_2 \;; \\
\mathcal{F}_2 &= Ad^T_{T_{t,2}} \mathcal{F}_t + \mathcal{G}_2 \dot{\mathcal{V}}_2 \;; \\
\mathcal{F}_1 &= Ad^T_{T_{2,1}} \mathcal{F}_2 + \mathcal{G}_1 \dot{\mathcal{V}}_1 \;; \\
\tau_1 &= \mathcal{F}_1^T \mathcal{A}_1 \;; \\
\tau_2 &= \mathcal{F}_2^T \mathcal{A}_2 \;;
\end{aligned}
$$

$$
\begin{aligned}
\mathcal{F}_2 &= Ad^T_{T_{t,2}} \mathcal{F}_t + \mathcal{G}_2 [Ad_{T_{2,1}}] \mathcal{A}_1 \ddot{\theta}_1 + \mathcal{G}_2 \mathcal{A}_2 \ddot{\theta}_2 \;; \\
\mathcal{F}_1 &= Ad^T_{T_{2,1}} \mathcal{F}_2 + \mathcal{G}_1 \mathcal{A}_1 \ddot{\theta}_1 \;; \\
\tau_1 &= \mathcal{F}_1^T \mathcal{A}_1 \;; \\
\tau_2 &= \mathcal{F}_2^T \mathcal{A}_2 \;;
\end{aligned}
$$

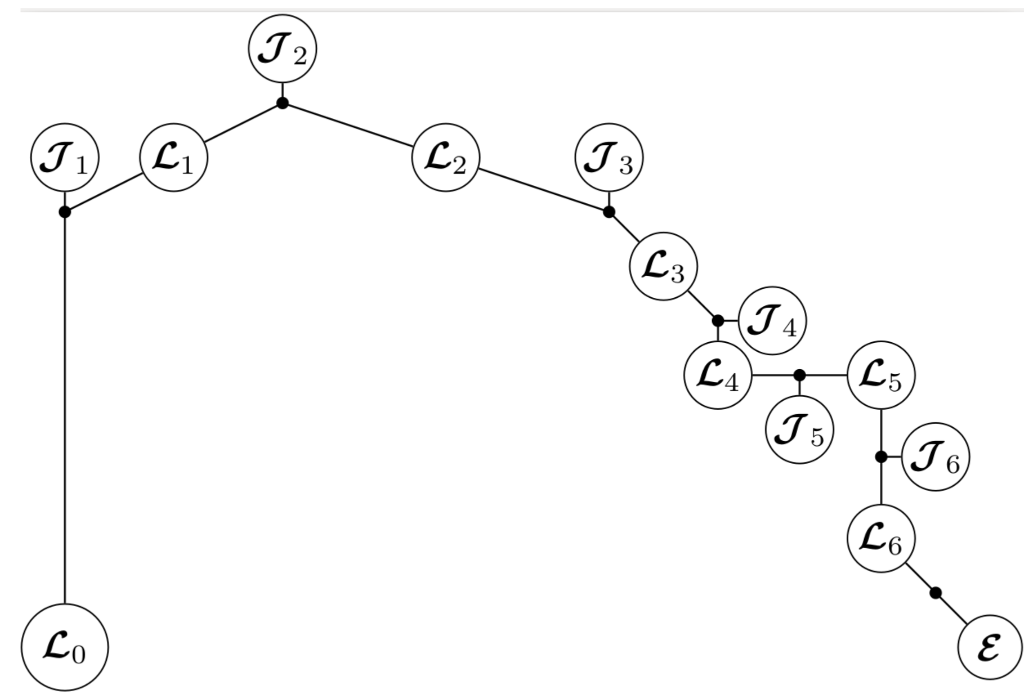
$$
\begin{aligned}
\tau_1 &= \mathcal{A}_1^T Ad^T_{T_{t,1}} \mathcal{F}_t + \\
&\quad (\mathcal{A}_1^T \mathcal{G}_2^1 \mathcal{A}_1 + \mathcal{A}_1^T \mathcal{G}_1 \mathcal{A}_1) \ddot{\theta}_1 + \mathcal{A}_1^T Ad^T_{T_{2,1}} \mathcal{G}_2 \mathcal{A}_2 \ddot{\theta}_2 \;; \\
\tau_2 &= \mathcal{A}_2^T Ad^T_{T_{t,2}} \mathcal{F}_t + \\
&\quad \mathcal{A}_2^T \mathcal{G}_2 [Ad_{T_{2,1}}] \mathcal{A}_1 \ddot{\theta}_1 + \mathcal{A}_2^T \mathcal{G}_2 \mathcal{A}_2 \ddot{\theta}_2 \;;
\end{aligned}
$$

$$
M(q) = \begin{bmatrix} \mathcal{A}_1^T \mathcal{G}_2^1 \mathcal{A}_1 + \mathcal{A}_1^T \mathcal{G}_1 \mathcal{A}_1 & \mathcal{A}_1^T Ad^T_{T_{2,1}} \mathcal{G}_2 \mathcal{A}_2 \\ \mathcal{A}_2^T \mathcal{G}_2 [Ad_{T_{2,1}}] \mathcal{A}_1 & \mathcal{A}_2^T \mathcal{G}_2 \mathcal{A}_2 \end{bmatrix}
$$

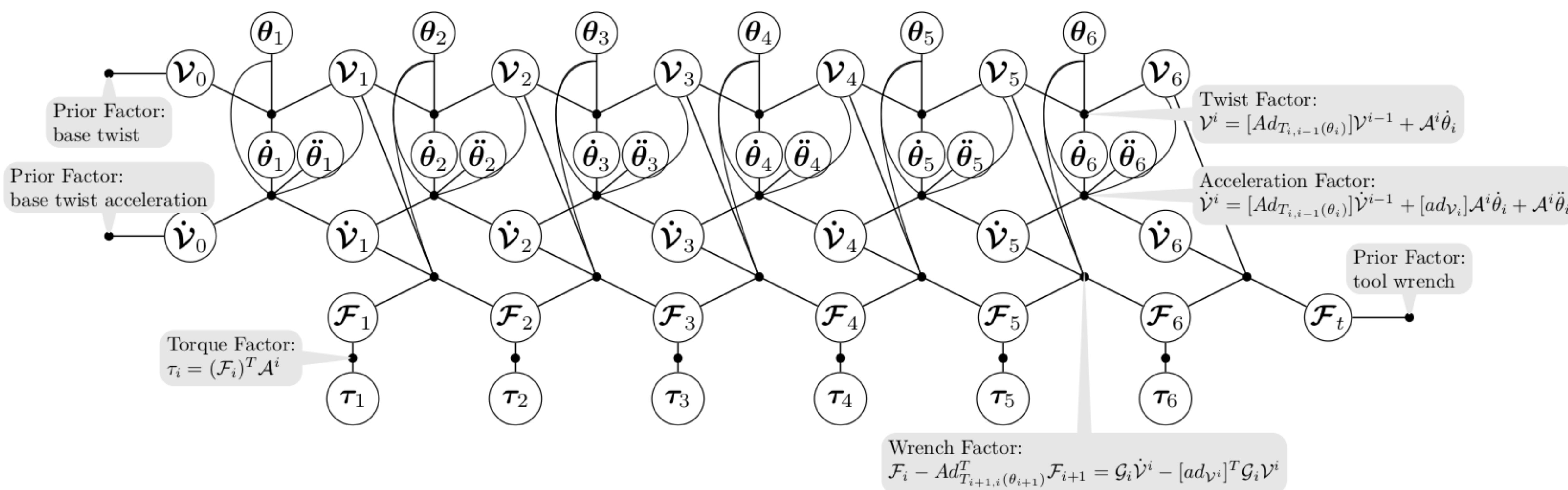
6R-Link Manipulator: Puma560 Robot



Puma560 (1)



Puma560 abstract factor graph

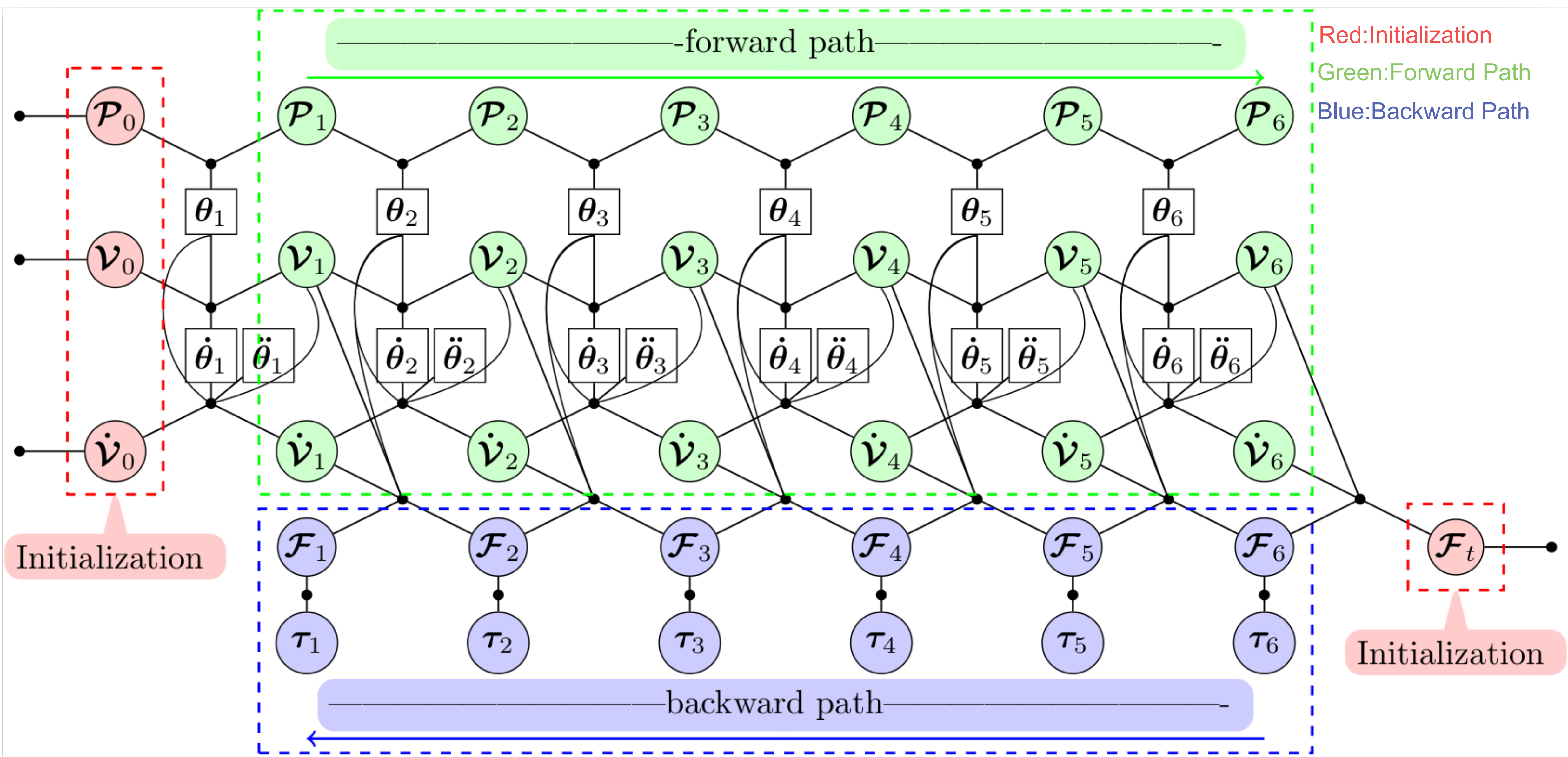


Puma560 dynamics factor graph

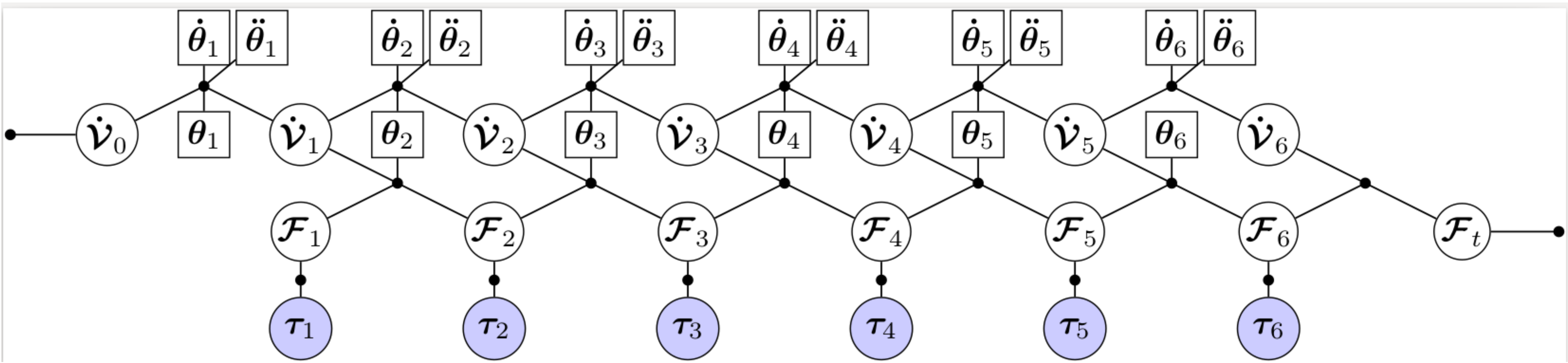
(1) https://www.google.com/search?q=puma+560+robot&tbm=isch&source=iu&ictx=1&fir=Z2n6nwl1bFv52M%253A%252CoM3AtpXY6Fgc-M%252C_&usg=AI4_-kRK9BklbTrnSMt_0kgzqamqISBWfA&sa=X&ved=2ahUKEwias6fegoPgAhUBna0KHZxdAWUQ9QEwAXoECAMQBg#imgdii=Qy9JrzD5VrBtyM:&imgrc=LYoe0vPf-aY4IM:

Puma 560 Inverse Dynamics Factor Graph

- Inverse Dynamics:
- Input: joint positions, velocities and accelerations;
- Output: required joint torques



Elimination Orders on Inverse Dynamics Factor Graph



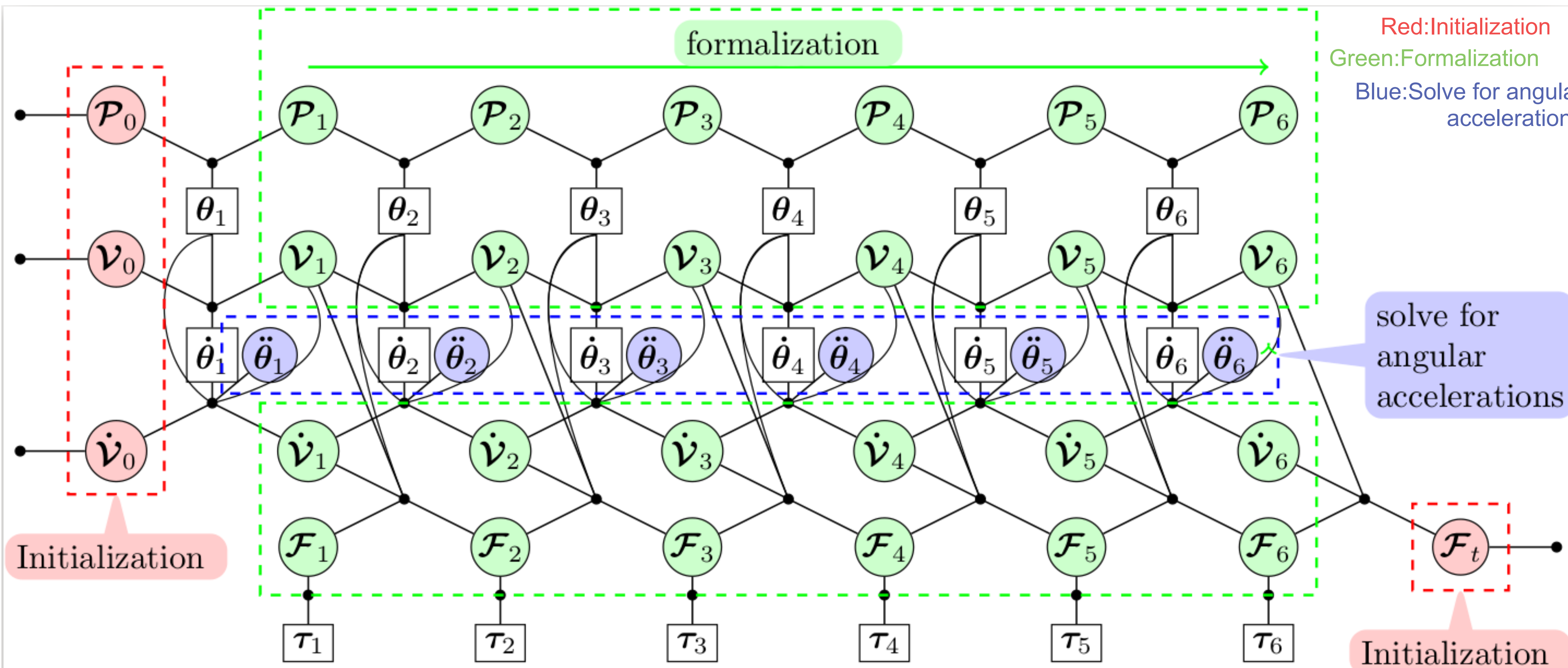
Simplified Inverse Dynamics Factor Graph

Elimination Ordering:
1. Colamd
2. Metis
3. Customized: matches Recursive Newton-Euler Algorithm (RNEA)
4. Customized: matching Lynch & Park’s book (Lynch)

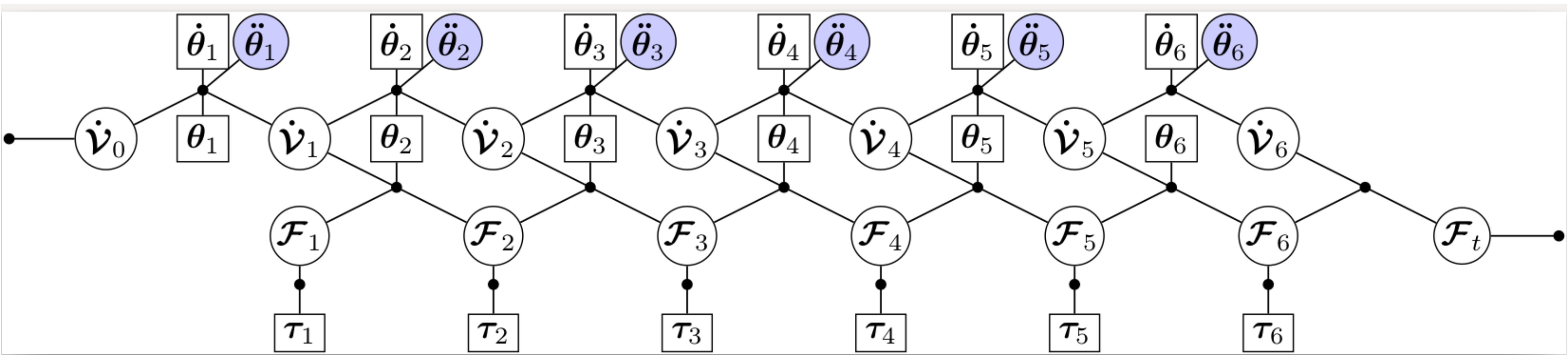
Elimination Method	Elimination Order	Average Time(μs)
COLAMD	$\tau_6, \tau_5, \tau_4, \tau_3, \tau_1, \tau_2, \dot{\mathcal{V}}_0, \mathcal{F}_1, \dot{\mathcal{V}}_1, \mathcal{F}_2, \dot{\mathcal{V}}_2, \mathcal{F}_3, \dot{\mathcal{V}}_3, \mathcal{F}_4, \dot{\mathcal{V}}_4, \mathcal{F}_5, \dot{\mathcal{V}}_5, \mathcal{F}_6, \dot{\mathcal{V}}_6, \mathcal{F}_7$	11.0322
METIS	$\tau_3, \tau_2, \tau_1, \dot{\mathcal{V}}_0, \mathcal{F}_3, \mathcal{F}_1, \dot{\mathcal{V}}_2, \mathcal{F}_2, \dot{\mathcal{V}}_1, \tau_6, \tau_5, \tau_4, \mathcal{F}_7, \dot{\mathcal{V}}_4, \dot{\mathcal{V}}_6, \mathcal{F}_5, \dot{\mathcal{V}}_5, \mathcal{F}_6, \mathcal{F}_4, \dot{\mathcal{V}}_3$	11.7108
CUSTOMIZED (RNEA)	$\tau_6, \tau_5, \tau_4, \tau_3, \tau_2, \tau_1, \dot{\mathcal{V}}_6, \mathcal{F}_1, \dot{\mathcal{V}}_5, \mathcal{F}_2, \dot{\mathcal{V}}_4, \mathcal{F}_3, \dot{\mathcal{V}}_3, \mathcal{F}_4, \dot{\mathcal{V}}_2, \mathcal{F}_5, \dot{\mathcal{V}}_1, \mathcal{F}_6, \dot{\mathcal{V}}_0, \mathcal{F}_7$	20.2105
CUSTOMIZED (Lynch)	$\tau_1, \mathcal{F}_1, \tau_2, \mathcal{F}_2, \tau_3, \mathcal{F}_3, \tau_4, \mathcal{F}_4, \tau_5, \mathcal{F}_5, \tau_6, \mathcal{F}_6, \mathcal{F}_7, \dot{\mathcal{V}}_0, \dot{\mathcal{V}}_1, \dot{\mathcal{V}}_2, \dot{\mathcal{V}}_3, \dot{\mathcal{V}}_4, \dot{\mathcal{V}}_5, \dot{\mathcal{V}}_6$	26.5747

Puma 560 Forward Dynamics Factor Graph

- Forward Dynamics:
- Input: joint positions, velocities and torques;
- Output: joint accelerations



Elimination Orders on Forward Dynamics Factor Graph



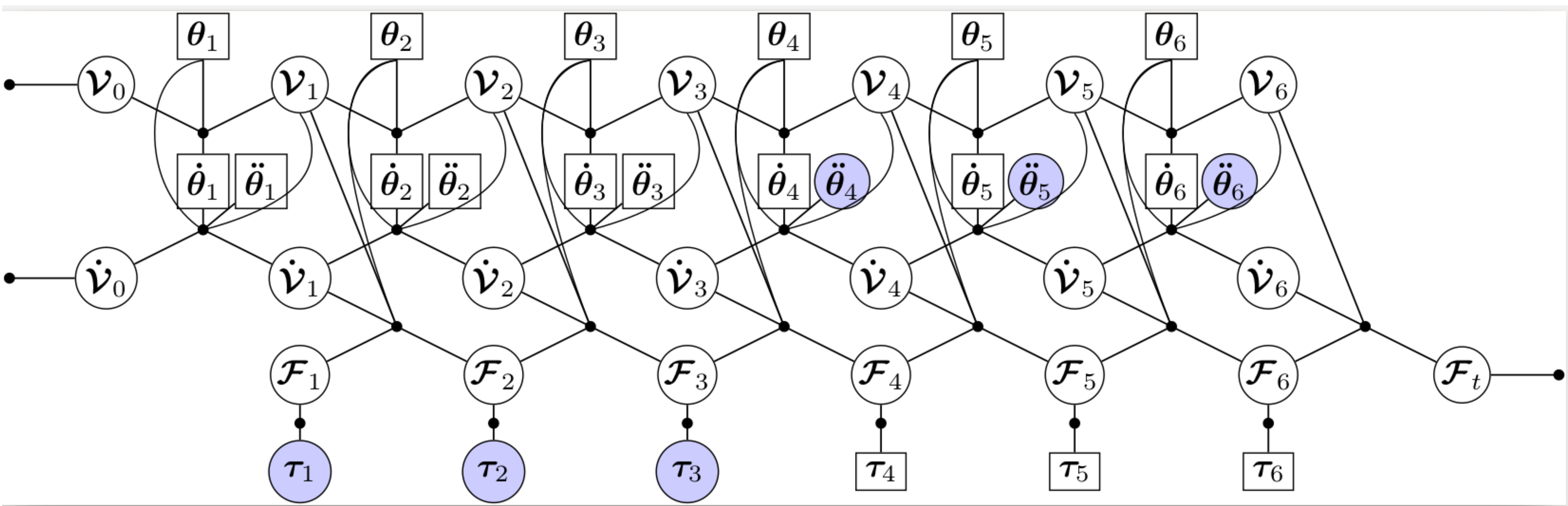
Simplified Forward Dynamics Factor Graph

Elimination Ordering:
1. Colamd
2. Metis
3. Customized: matches Composite-Rigid-Body Algorithm (CRBA)
4. Customized: matches Articulated-Body Algorithm (ABA)

Elimination Method	Elimination Order	Average Time(μs)
COLAMD	$\mathcal{F}_7, a_5, a_4, \tau_3, \tau_1, \tau_2, \dot{\mathcal{V}}_0, \mathcal{F}_1, \dot{\mathcal{V}}_1, \mathcal{F}_2, \dot{\mathcal{V}}_2, \mathcal{F}_3, \dot{\mathcal{V}}_3, \mathcal{F}_4, \dot{\mathcal{V}}_4, \mathcal{F}_5, \mathcal{F}_6, \dot{\mathcal{V}}_5, \dot{\mathcal{V}}_6, a_6$	11.6785
METIS	$a_4, \mathcal{F}_7, a_6, a_5, \mathcal{F}_5, \dot{\mathcal{V}}_6, \dot{\mathcal{V}}_4, \mathcal{F}_6, \dot{\mathcal{V}}_5, \tau_3, \tau_2, \tau_1, \dot{\mathcal{V}}_0, \mathcal{F}_3, \mathcal{F}_1, \dot{\mathcal{V}}_2, \mathcal{F}_2, \dot{\mathcal{V}}_1, \mathcal{F}_4, \dot{\mathcal{V}}_3$	13.0836
CUSTOMIZED 1	$\mathcal{F}_1, \mathcal{F}_2, \mathcal{F}_3, \mathcal{F}_4, \mathcal{F}_5, \mathcal{F}_6, \mathcal{F}_7, \dot{\mathcal{V}}_0, \dot{\mathcal{V}}_1, \dot{\mathcal{V}}_2, \dot{\mathcal{V}}_3, \dot{\mathcal{V}}_4, \dot{\mathcal{V}}_5, \dot{\mathcal{V}}_6, a_1, a_2, a_3, a_4, a_5, a_6$	34.6130
CUSTOMIZED 2	$a_1, a_2, a_3, a_4, a_5, a_6, \mathcal{F}_1, \mathcal{F}_2, \mathcal{F}_3, \mathcal{F}_4, \mathcal{F}_5, \mathcal{F}_6, \mathcal{F}_7, \dot{\mathcal{V}}_0, \dot{\mathcal{V}}_1, \dot{\mathcal{V}}_2, \dot{\mathcal{V}}_3, \dot{\mathcal{V}}_4, \dot{\mathcal{V}}_5, \dot{\mathcal{V}}_6$	24.5449

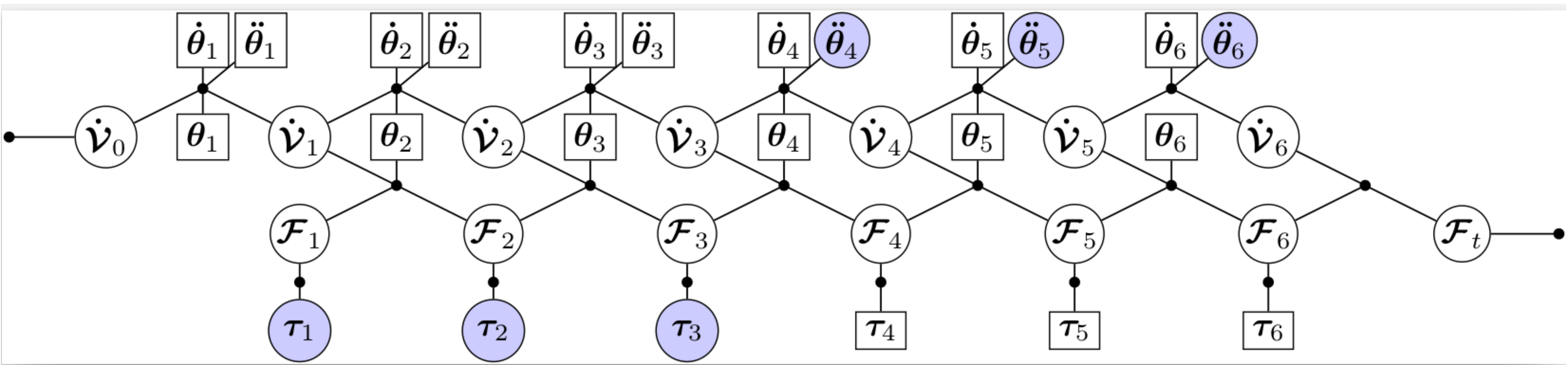
Puma 560 Hybrid Dynamics Factor Graph

- Hybrid Dynamics:
- Input: joint positions, velocities and partial joint accelerations and partial joint torques;
- Output: the rest of joint accelerations, and joint torques



Inputs: accelerations for joint 1, 2, 3, and torques for joint 4, 5, 6
Outputs: torques for joint 1, 2, 3 and accelerations for joint 4 ,5, 6.

Elimination Orders on Hybrid Dynamics Factor Graph

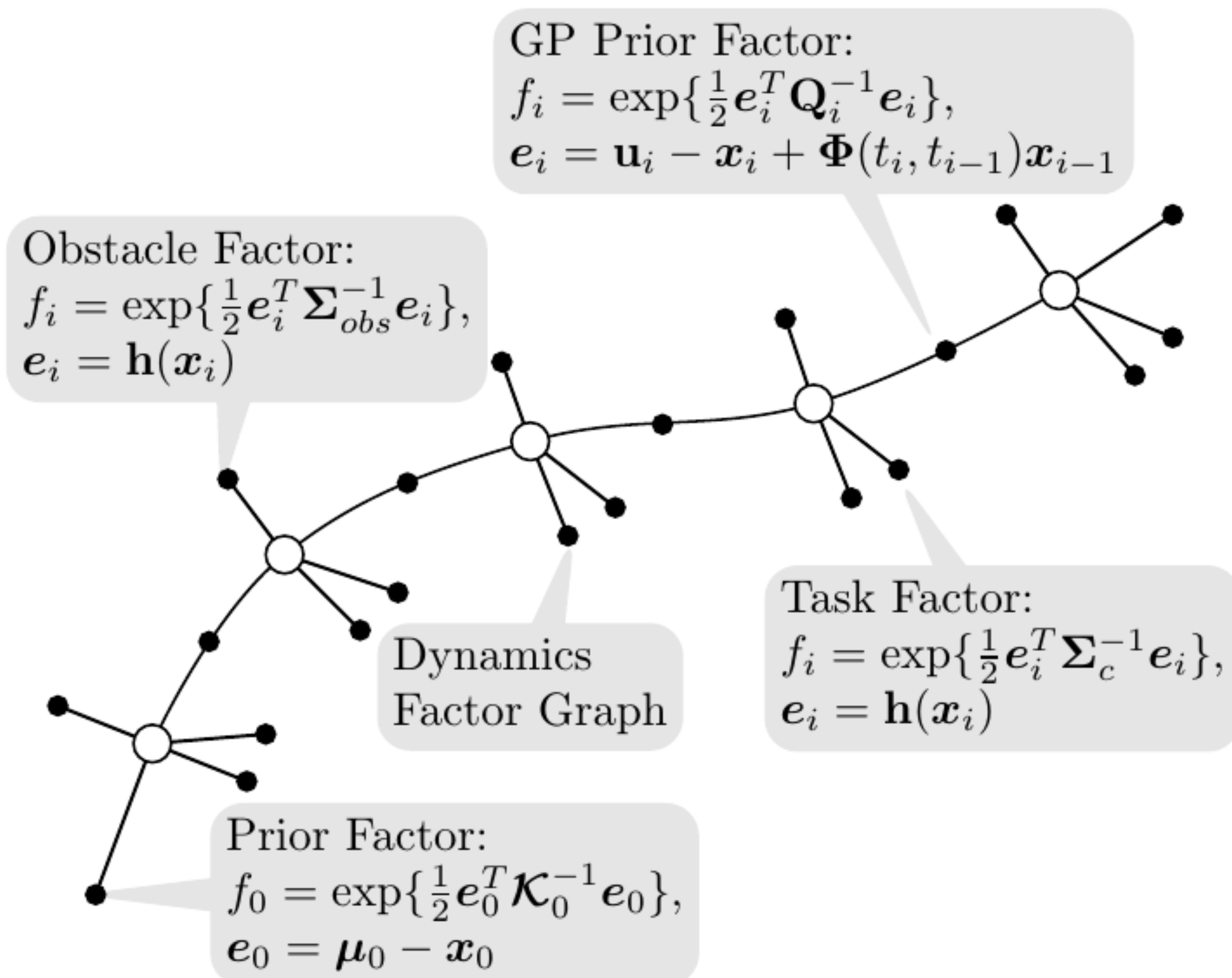


Simplified Hybrid Dynamics Factor Graph

Elimination Ordering:
1. Colamd
2. Metis
3. Customized 1
4. Customized 2

Elimination Method	Elimination Order	Average Time(ms)
COLAMD	$\mathcal{F}_7, a_5, a_4, \tau_3, \tau_1, \tau_2, \dot{\mathcal{V}}_0, \mathcal{F}_1, \dot{\mathcal{V}}_1, \mathcal{F}_2, \dot{\mathcal{V}}_2, \mathcal{F}_3, \dot{\mathcal{V}}_3, \mathcal{F}_4, \dot{\mathcal{V}}_4, \mathcal{F}_5, \mathcal{F}_6, \dot{\mathcal{V}}_5, \dot{\mathcal{V}}_6, a_6$	0.0116785
METIS	$a_4, \mathcal{F}_7, a_6, a_5, \mathcal{F}_5, \dot{\mathcal{V}}_6, \dot{\mathcal{V}}_4, \mathcal{F}_6, \dot{\mathcal{V}}_5, \tau_3, \tau_2, \tau_1, \dot{\mathcal{V}}_0, \mathcal{F}_3, \mathcal{F}_1, \dot{\mathcal{V}}_2, \mathcal{F}_2, \dot{\mathcal{V}}_1, \mathcal{F}_4, \dot{\mathcal{V}}_3$	0.0130836
CUSTOMIZED 1	$\mathcal{F}_1, \mathcal{F}_2, \mathcal{F}_3, \mathcal{F}_4, \mathcal{F}_5, \mathcal{F}_6, \mathcal{F}_7, \dot{\mathcal{V}}_0, \dot{\mathcal{V}}_1, \dot{\mathcal{V}}_2, \dot{\mathcal{V}}_3, \dot{\mathcal{V}}_4, \dot{\mathcal{V}}_5, \dot{\mathcal{V}}_6, a_1, a_2, a_3, a_4, a_5, a_6$	0.0346130
CUSTOMIZED 2	$a_1, a_2, a_3, a_4, a_5, a_6, \mathcal{F}_1, \mathcal{F}_2, \mathcal{F}_3, \mathcal{F}_4, \mathcal{F}_5, \mathcal{F}_6, \mathcal{F}_7, \dot{\mathcal{V}}_0, \dot{\mathcal{V}}_1, \dot{\mathcal{V}}_2, \dot{\mathcal{V}}_3, \dot{\mathcal{V}}_4, \dot{\mathcal{V}}_5, \dot{\mathcal{V}}_6$	0.0245449

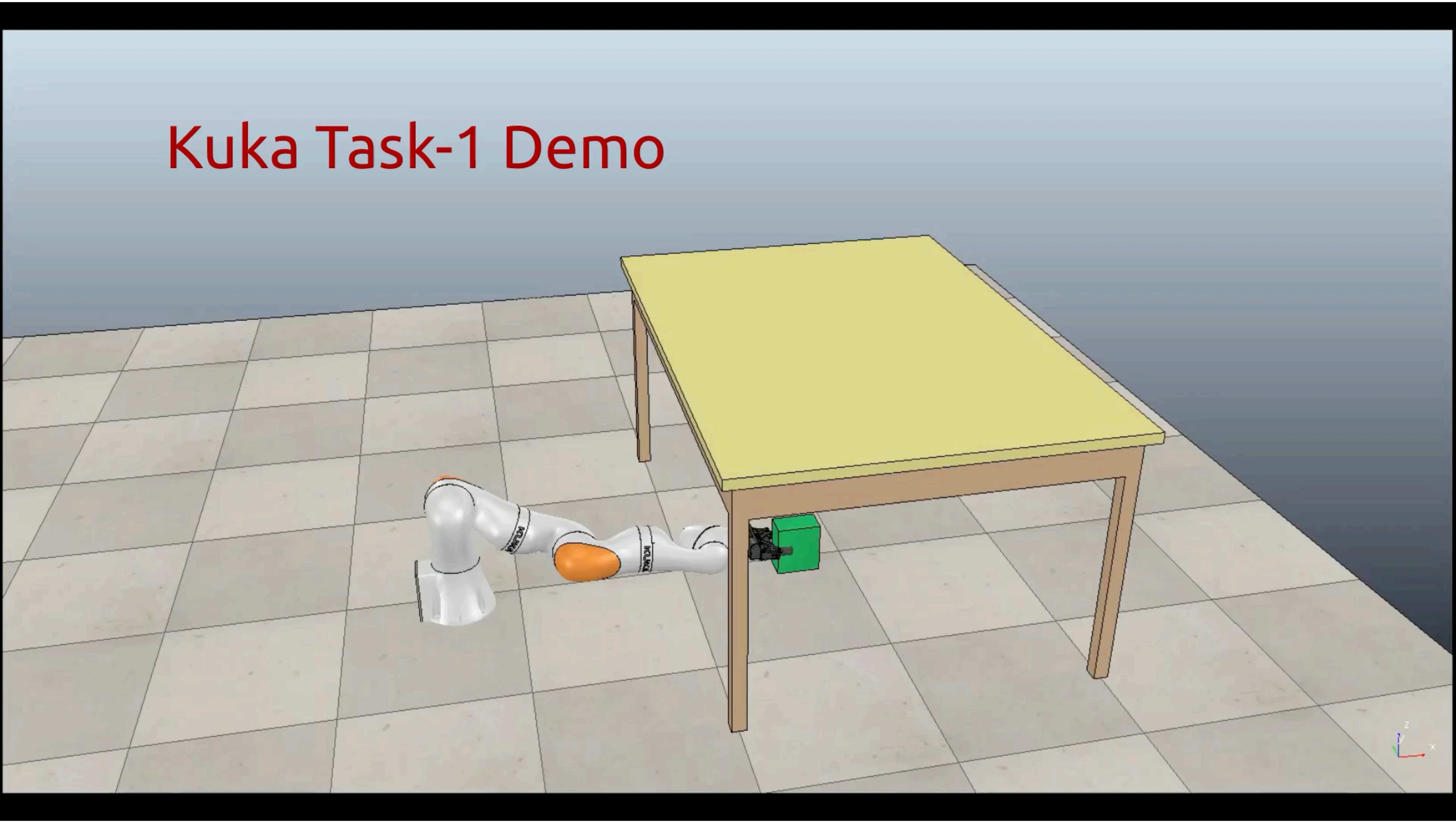
Kinodynamics Motion Planning With Factor Graph



- Formulate the kinodynamic motion planning problem as probabilistic inference.
- Solve the problem by finding the maximum a posteriori (MAP) trajectory given the following constraints:

1) Dynamics constraints to ensure dynamically feasible and stable motion;
2) Obstacle likelihood function to ensure collision-free motion;
3) Task-specified cost to satisfy certain motion requirement;
4) Prior distribution on the space of trajectories to encourage smoothness in motion.

Kinodynamics Motion Planning for KUKA



Acrobot