Efficient Inversion of the KKT Matrix:

n: # Dofs
m: # Constraint Dofs
d: tree depth

$$H = \begin{bmatrix} M & J^T \\ J & 0 \end{bmatrix} \qquad M \in \mathbb{R}^{n \times n} \quad J \in \mathbb{R}^{m \times n}$$

Goal: Solve for x

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = H \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

We will show that it is possible in time

$$O(mn + m^3)$$

Note: Could get it down to $O(n + md + m^3)$ but that method isn't as easy to explain. [my paper w/ Roy]

Block Elimination:

$$\begin{array}{l} -JM^{-1}\otimes \\ \downarrow + \end{array} \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} M & J^T \\ J & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \Rightarrow \begin{bmatrix} y_1 \\ y_2 - JM^{-1}y_1 \end{bmatrix} = \begin{bmatrix} M & J^T \\ 0 & -JM^{-1}J^T \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

So: Looking @ the last row

$$\underbrace{JM^{-1}J^T}_{\Lambda^{-1}} x_2 = \overbrace{JM^{-1}y_1}^{:= a_1} - y_2$$

Compute each column of Λ^{-1} with a call to ABA giving contact accel when $\dot{q}=0$, grav $=0$, force $= e_i$

e_i: i-th unit vector used to calc. i-th col. of Λ^{-1}

$JM^{-1}y_1$: Compute w/ ABA in time $\boxed{O(n)}$

(contact accel when $\dot{q}=0$, grav $=0$, $\tau = y_1$)

$$\boxed{\begin{array}{c}\text{Total time} \\ O(nm)\end{array}}$$

$O(n)$ per ABA call
m columns of Λ^{-1}

Then: Densely solve $\Lambda^{-1}x_2 = a_1 - y_2$ $\boxed{O(m^3) \text{ cost}}$

Block Elim cont.:

$$\begin{bmatrix} y_1 \\ y_2 - JM^{-1}y_1 \end{bmatrix} = \begin{bmatrix} M & J^T \\ 0 & -JM^{-1}J^T \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

Now that we've solved for x_2 we proceed to use the 1st row to solve for x_1:

$$y_1 - J^T x_2 = M x_1$$

Solve for x_1 with ABA $\boxed{O(n) \text{ cost}}$

($\ddot{q}$ when $\tau = y_1$, $f = x_2$, $\dot{q} = 0$, grav$=0$)

Summary: Suppose $\begin{bmatrix} \ddot{q} \\ a_c \end{bmatrix} = ABA(q, \dot{q}, \tau, f)$ [ignore gravity]

① $\begin{bmatrix} * \\ a_1 \end{bmatrix} = ABA(q, 0, y_1, 0)$ $O(n)$

② for $j = 1$ to m

$(\Lambda^{-1})_j = ABA(q, 0, 0, -e_j)$ $O(nm)$

③ $x_2 = (\Lambda^{-1})^{-1} [a_1 - y_2]$ $O(m^3)$

④ $\begin{bmatrix} x_1 \\ * \end{bmatrix} = ABA(q, 0, y_1, x_2)$ $O(n)$

Total:

$O(nm + m^3)$