Simulating Multibody System With Holonomic Constraints

$\bar{q} \rightarrow n$ generalized coordinates
$\bar{q}_r \rightarrow M$ dependent coordinates
$\Big\}$ $N = n + M$ total coordinates

$$\bar{f}_h(\bar{q}, \bar{q}_r, t) = 0 \rightarrow M \text{ holonomic constraints}$$

These are nonlinear in the coordinates and explicit analytical solutions for $\bar{q}_r$ are generally not possible.

The equations of motion of a holonomic multibody system with holonomic constraints are:

$$\left.\begin{array}{ll}
\text{algebraic eq's} \rightarrow & \bar{f}_h(\bar{q}, \bar{q}_r, t) = 0 \in \mathbb{R}^M \\
\text{diff. eqns} & \left\{\begin{array}{l} \bar{f}_k(\bar{u}, \dot{\bar{q}}, \dot{\bar{q}}_r, \bar{q}, \bar{q}_r, t) = 0 \in \mathbb{R}^n \\ \bar{f}_d(\dot{\bar{u}}, \bar{u}, \ddot{\bar{q}}_r, \dot{\bar{q}}_r, \bar{q}, \bar{q}_r, t) = 0 \in \mathbb{R}^n \end{array}\right.
\end{array}\right\} \quad 2n + M \text{ equations}$$

$\bar{u}, \bar{q} \rightarrow 2n$ $\quad \bar{q}_r, \dot{\bar{q}}_r, \ddot{\bar{q}}_r \Rightarrow 3M$ unknowns

These equations are <u>differential</u> <u>algebraic</u> <u>equations</u> (DAEs).

The "index" of the DAEs can be reduced by introducing the derivatives of $\bar{f}_h$.

$$\dot{\bar{f}}_h(\bar{u}, \bar{q}, \dot{\bar{q}}_r, \bar{q}_r, t) = 0 \in \mathbb{R}^M$$

$$\ddot{\bar{f}}_h(\dot{\bar{u}}, \bar{u}, \bar{q}, \ddot{\bar{q}}_r, \dot{\bar{q}}_r, \bar{q}_r, t) = 0 \in \mathbb{R}^M$$

Recall how we eliminated the dependent generalized speeds in nonholonomic systems (prior lecture). We can do something similar when we have holonomic constraints.

Introduce $\bar{u} \in \mathbb{R}^n$ & $\bar{u}_r \in \mathbb{R}^M$. $\bar{u}$ are generalized speeds and $\bar{u}_r$ are psuedo generalized speeds.

$$\bar{f}_h(\bar{q}, \bar{q}_r, t) = 0 \in \mathbb{R}^M$$

and kin. diff eqns.

$$\bar{f}_k(\bar{u}, \bar{u}_r, \dot{\bar{q}}, \dot{\bar{q}}_r, \bar{q}, \bar{q}_r, t) = 0 \in \mathbb{R}^N$$

$$\dot{\bar{f}}_h(\bar{u}, \bar{u}_r, \bar{q}, \bar{q}_r, t) = 0 \in \mathbb{R}^M$$

$\dot{\bar{f}}_h$ is linear in $\bar{u}_r$, so solve for $\bar{u}_r$

$$\bar{u}_r = -M_{hd}^{-1}\,\bar{g}_{hd}$$

Now the equations of motion can be written in a concise form by substituting for $\bar{u}_r$.

$$\left.\begin{array}{ll} \bar{f}_h(\bar{q}, \bar{q}_r, t) = 0 & \in \mathbb{R}^M \\ \bar{f}_k(\dot{\bar{q}}, \bar{u}, \bar{q}, \bar{q}_r, t) = 0 & \in \mathbb{R}^n \\ \bar{f}_d(\dot{\bar{u}}, \bar{u}, \bar{q}, \bar{q}_r, t) = 0 & \in \mathbb{R}^n \end{array}\right\} \begin{array}{l} M + 2n \text{ eqns} \\ \underset{n}{\bar{u}}, \underset{n}{\bar{q}}, \underset{M}{\bar{q}_r} \end{array}$$

ODEs ($\bar{f}_k$, $\bar{f}_d$)

$\bar{f}_k$ & $\bar{f}_d$ can be integrated as ordinary diff. eqns as long as the initial conditions satisfy the constraints.

$$\bar{f}_h(\bar{q}(t_0), \bar{q}_r(t_0), t_0) = 0$$

The numerical integration error will compound and $\bar{f}_h = 0$ will not hold as $t_0 \rightarrow t_f$.

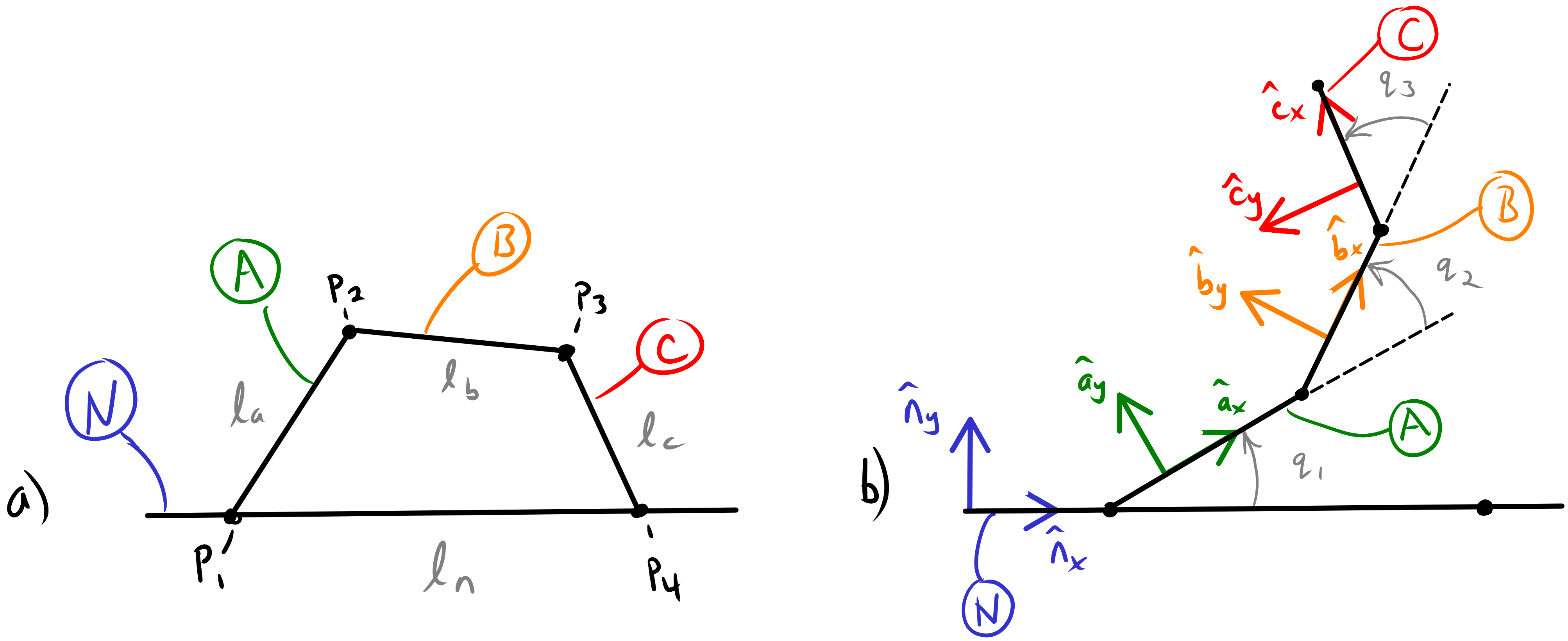
There are several techniques for ensuring that $\mathfrak{f}_h = 0$ over an integration period, most involve correcting the state at each integration step.

For our purposes, we will use a DAE solver (similar to solve_ivp). For use in Python scikits.odes will be used. It has a wrapper to a solver IDA ⇒ implicit differential algebraic solver.

Key things for this solver:

- provide equations of motion in implicit form ⇒
  $\mathfrak{f}(\dot{\bar{u}}, \bar{u}, \bar{q}, \bar{q}_r, t) = 0$
- identify which equations are the constraints
- ensure initial conditions are satisfied: $\dot{\bar{u}}_0, \bar{u}_0, \bar{q}_0$
- select tolerances to minimize the constraint ε residuals

Example: four-bar linkage


a)
N
A
B
C
P_1
P_2
P_3
P_4
l_a
l_b
l_c
l_n
b)
$\hat{n}_y$
$\hat{n}_x$
$\hat{a}_y$
$\hat{a}_x$
$\hat{b}_y$
$\hat{b}_x$
$\hat{c}_y$
$\hat{c}_x$
q_1
q_2
q_3