Kane's Method:
Finding Constraint Equations

Eric Stoneking
Nov 2014

Kane's Equation

Recall Kane's equation in matrix form:

$$\Omega^T (T - I\alpha - \omega \times H) + V^T (F - ma) = 0$$

We've said that the matrices Ω and V project Euler's and Newton's equations into the state space spanned by the generalized speeds.

Let's step back and look at the bigger picture.

Rigid Bodies and Joints

- A system of rigid bodies has a total of $6N_b$ DOF
- Assign 6DOF to the root body
- Examine each joint
  - The outer body motion is dictated by the inner body motion and the joint DOFs
- A joint introduces 0-6 constraints (typically 1-5)
  - Example: a simple hinge introduces 3 translational constraints and 2 rotational constraints, leaving 1 DOF
- So each of the $6N_b$ potential DOFs either becomes a DOF or a constraint: $N_u + N_c = 6N_b$

Doing It the Hard Way

- Typical Newton-Euler formulations (ref Haug, etc) adjoin the constraint equations to the equations of motion and solve them together
- Introduces constraint forces (and torques) as additional unknowns to be solved for
  - These are obtained “for free”, in the sense that you have no choice but to obtain them
- Size of system to be solved is then $6N_b+N_c$
  - Solution time grows like $(6N_b+N_c)^3$
- Due to roundoff, constraints may not be perfectly satisfied
  - This can also degrade the accuracy of the overall solution

Kane's Method is Faster and More Accurate

- Constraints are identically satisfied
  - Numerical issues related to augmented systems are avoided entirely
- Size of system to solve is $6N_b - N_c$
  - Solution time grows like $(6N_b - N_c)^3$
    - Example: $N_b = 2$, $N_c = 5$ (a 1DOF hinge)
      - $6N_b + N_c = 17$, $6N_b - N_c = 7$
      - $7^3/17^3 = 0.07$
    - So Kane's Method is much faster
  - This system is solved 4x per time step for RK4
- Computation of constraint forces/torques is optional, not compulsory
  - 1x per time step (not 4x), and with no matrix inversion

Assigning DOFs and Constraints

- Every system starts out with $6N_b$ potential DOFs
- Give the root body (B_1) 6 DOFs
- Examine each joint in turn
  - Assign each potential DOF to an actual DOF or a constraint
  - For a tree topology, all potential DOFs will be accounted for
    - Non-tree topologies are a topic for another day

An Example Joint Definition

```
0 1                         ! Inner, outer body indices
1   213   GIMBAL            ! RotDOF, Seq, GIMBAL or SPHERICAL
0   123                     ! TrnDOF, Seq
```

- Identify number of rotational, translational DOFs
- Give 3-element sequences in all cases
  - Ex: Rotational axis 2 is free, and axes 1 and 3 are constrained (in that order!)
- Spherical joints are distinguished from 3-DOF gimbal joints by additional keyword

Theory

Imagine the dynamical system with all potential DOFs.
We write Kane's equation:

$$\Omega_P^T[T-I(\Omega\dot{u}+\alpha_r)-\omega\times H]+V_P^T[F-m(V\dot{u}+a_r)]=0$$

where $\Omega_P, \ V_P$ project into all $6N_b$ dimensions of the state space.

Now partition Ω_P, V_P:

$$\Omega_P=\begin{bmatrix}\Omega & \Omega_c\end{bmatrix}, \ V_P=\begin{bmatrix}V & V_c\end{bmatrix}$$

where Ω, V project into the permissible motion subspace, and Ω_c, V_c project into the constrained subspace.

Equations of Motion

$$\Omega^T [T - I(\Omega \dot{u} + \alpha_r) - \omega \times H] + V^T [F - m(V \dot{u} + a_r)] = 0$$

Rearrange to solve for $\dot{u}$:

$$(\Omega^T I \Omega + V^T m V)\dot{u} = \Omega^T (T - I \alpha_r - \omega \times H) + V^T (F - m a_r)$$

- An N_u x N_u system of equations to solve by Gaussian elimination (or similar)
- Propagate using RK4
  - Requires system to be solved four times per timestep
- If constraints are unwanted, this gives the complete solution

Equations of Constraint

$$\Omega_c^T [T - I(\Omega \dot{u} + \alpha_r) - \omega \times H] + V_c^T [F - m(V \dot{u} + a_r)] = 0$$

Rearrange to isolate generalized constraint forces:

$$F_c = \Omega_c^T T + V_c^T F = \Omega_c^T [I(\Omega \dot{u} + \alpha_r) + \omega \times H] + V_c^T [m(V \dot{u} + a_r)]$$

- All terms on RHS are known from solution of equations of motion
- No simultaneous solution needed (no matrix inverse!)
  - Simply matrix multiplication and addition
- Evaluated once per timestep (not four times, as would be if it were solved along with equations of motion)
- Completely optional
  - Find all, some, or none as desired

Conclusion

- Kane's method provides a straightforward way to obtain constraint forces and torques
- Accuracy of equations of motion is not compromised as happens when EOM are augmented with constraint equations
- Computational complexity of Kane's method is much less than for augmented formulation
- Addition of constraint equations is much less burdensome by using Kane's method
  - Constraint equations do not “perturb” equations of motion
  - No matrix inversion
  - Solve once per timestep, not four times (for RK4)
  - Can restrict attention to those constraints of interest

Appendix:
Example from OSIRIS-REX

Example from OSIRIS-REX

- To study the dynamics of the Touch-And-Go (TAG) maneuver, we built a model of O-Rex in 42 with:
  - Main Body: 6 DOF
  - Shoulder and Elbow Joints: All DOF constrained
    - To find constraint forces and torques
  - Pogo Joint: 1DOF Translation
  - Wrist: 2DOF Gimballed
- Following slides show part of output file Tree00.42
  - Documents partition of potential DOFs into DOFs and constraints

Main Body has 6DOF

```
*****************************************************************
Body 00:    RotSeq = 123    TrnSeq = 123
                                  Col in    Col in        Col in
Axis        F/C     u[]  x[]      u00.42    x00.42    Constraint00.42
-----------------------------------------------------------------
Rot1         F       00   00        01        01            --
Rot2         F       01   01        02        02            --
Rot3         F       02   02        03        03            --
(Sph)        -       --   03        --        04            --

Trn1         F       06   07        07        08            --
Trn2         F       07   08        08        09            --
Trn3         F       08   09        09        10            --
```

Shoulder and Elbow are Constrained

```
*****************************************************************
Joint 00:   RotSeq = 213   TrnSeq = 123
                                Col in    Col in       Col in
Axis        F/C     u[]  x[]    u00.42    x00.42   Constraint00.42
-----------------------------------------------------------------
Rot1         C       --   --      --        --          01
Rot2         C       --   --      --        --          02
Rot3         C       --   --      --        --          03

Trn1         C       --   --      --        --          04
Trn2         C       --   --      --        --          05
Trn3         C       --   --      --        --          06
*****************************************************************
Joint 01:   RotSeq = 213   TrnSeq = 123
                                Col in    Col in       Col in
Axis        F/C     u[]  x[]    u00.42    x00.42   Constraint00.42
-----------------------------------------------------------------
Rot1         C       --   --      --        --          07
Rot2         C       --   --      --        --          08
Rot3         C       --   --      --        --          09

Trn1         C       --   --      --        --          10
Trn2         C       --   --      --        --          11
Trn3         C       --   --      --        --          12
```

Pogo and Wrist Joints

```
*****************************************************************
Joint 02:   RotSeq = 213   TrnSeq = 312
                                Col in   Col in       Col in
Axis      F/C    u[]  x[]       u00.42   x00.42   Constraint00.42
-----------------------------------------------------------------
Rot1       C      --   --         --       --           13
Rot2       C      --   --         --       --           14
Rot3       C      --   --         --       --           15

Trn1       F      03   04         04       05           --
Trn2       C      --   --         --       --           16
Trn3       C      --   --         --       --           17
*****************************************************************
Joint 03:   RotSeq = 213   TrnSeq = 123
                                Col in   Col in       Col in
Axis      F/C    u[]  x[]       u00.42   x00.42   Constraint00.42
-----------------------------------------------------------------
Rot1       F      04   05         05       06           --
Rot2       F      05   06         06       07           --
Rot3       C      --   --         --       --           18

Trn1       C      --   --         --       --           19
Trn2       C      --   --         --       --           20
Trn3       C      --   --         --       --           21
```