Mimic Joints

PhysXTeam

September 25, 2024

Preliminaries

Mimic joints are designed to maintain a linear relationship between the positions of two degrees of freedom of an articulation instance. The two degrees of freedom, henceforth labelled A and B, may be in any sub-tree of the articulation.

With $q_A(t)$ and $q_B(t)$ denoting the positions of the two degrees of freedom and $\dot{q}_A(t)$ and $\dot{q}_B(t)$ denoting the speeds of the two degrees of freedom we may introduce the position $s(t)$ and velocity $v(t)$ of a mimic joint:

$$s(t) = \{q_A(t), q_B(t)\} \tag{1}$$

$$v(t) = \{\dot{q}_A(t), \dot{q}_B(t)\} \tag{2}$$

The linear constraint $C(t)$ coupling the two positions $q_A(t)$ and $q_B(t)$ has the form:

$$C(t) = q_A(t) + G \cdot q_B(t) + \gamma = 0 \tag{3}$$

with G playing the role of a gearing ratio and γ playing the role of a constant offset between the two positions.

Constrained Dynamics

The constraint described in Equation (3) will be maintained provided that $\dot{C}(t) = \frac{dC(t)}{dt} = 0$. Differentiating Equation (3) with respect to time re-expresses the mimic joint in terms of the speeds of the degrees of freedom of the mimic joint:

$$\dot{q}_A(t) + G \cdot \dot{q}_B(t) = 0 \tag{4}$$

In practice, it is not possible to ensure that the constraint is resolved exactly. Indeed, it is typical for errors to accumulate due to time discretization and rounding error. There is also the possibility that the joints of the articulation will be initially configured in a way that does not satisfy the mimic joint constraint. To avoid drift propagation and accumulation it is necessary to amend Equation (4) so that $v(t)$ accounts for the velocity required to counteract at least some of the error:

$$\dot{q}_A(t) + G \cdot \dot{q}_B(t) + \frac{erp \cdot C(t)}{\Delta t} = 0 \tag{5}$$

where erp is a constant Baumgarte multiplier to correct a fraction of the error that might be present at time t and Δt is the timestep of the simulation. Increasing erp will more aggressively resolve the accumulated error but comes with the risk of adding energy to the system because it necessarily overshoots the ideal velocity that would occur with zero error.

The goal now is to compute impulses that may be applied to the degrees of freedom A and B such that Equation (5) is satisfied.

Test Impulses

The Featherstone formulation allows link impulses to be propagated inwards from link to root and then the subsequent changes in link spatial velocity to be propagated outwards from root to link. We extend this idea to be able to also propagate impulses applied to individual degrees of freedom, which will be analogous to joint actuation.

Consider a unit test impulse applied to degrees of freedom A and B. A test impulse applied to A will change $\dot{q}_A(t)$ but may also change $\dot{q}_B(t)$. A test impulse applied to B will likewise have an impact on $\dot{q}_B(t)$ and may also have an impact $\dot{q}_A(t)$.

[Note: The effect of a test impulse is described in[1] and is already a feature of PhysX articulations. The difference here is that we have some extra book-keeping to do to compute changes to mimic joint speed.]

We define r_{ij} to be the effect of a test unit impulse applied to degree of freedom j on the speed of degree of freedom i. Continuing with this notation, we may compute r_{AA}, r_{AB}, r_{BB}, r_{BA}. It is worth noting that r_{AB} and r_{BA} will be 0 if the shortest path from A to B crosses a fixed root link.

Mimic Joint Impulse Computation

Equation (5) may be recast in a familiar form:

$$J \cdot v(t) + b = 0 \tag{6}$$

with the Jacobian J as follows:

$$J = \frac{\partial C(t)}{\partial s(t)} = \{1, G\} \tag{7}$$

and the bias velocity b having the form:

$$b = \frac{erp \cdot C(t)}{\Delta t} \tag{8}$$

We may also express $v(t)$ as the result of a constraint force $F_C = \{F_A, F_B\}^T$ (stacked forces on degrees of freedom A and B, respectively) applied at $t - dt$ to the velocity state $v(t - dt)$:

$$v(t) = v(t - dt) + dt \cdot M^{-1} \cdot F_C \tag{9}$$

where M^{-1}

$$M^{-1} = \begin{bmatrix} r_{AA} & r_{AB} \\ r_{BA} & r_{BB} \end{bmatrix} \tag{10}$$

The goal is to compute a constraint force F_C that performs no work. A constraint force that performs no work will have the following form:

$$F_C = J^T \cdot f \tag{11}$$

[Proof: The work of the constraint force is ${F_C}^T \cdot \{v(t) - v(t - dt)\} \cdot dt$. Assume the nominal case of no prior constraint violation ($C(t - dt) = 0$), then substitute Equations (6) and (11) into the expression for the work.]

Substituting Equation (11) into (9), projecting the equation into constraint-space by pre-multiplying with J, and using (5) reveals the following relationship:

$$dt \cdot f = -\frac{b + J \cdot v(t - dt)}{J \cdot M^{-1} \cdot J^T} \tag{12}$$

We seek the impulse $dt \cdot J^T \cdot f$. The impulse I_A applied to degree of freedom A is therefore:

$$I_A = dt \cdot f \tag{13}$$

and the impulse I_B applied to degree of freedom B is

$$I_B = dt \cdot f \cdot G \tag{14}$$

For completeness it is worth expanding $J \cdot M^{-1} \cdot J^T$:

$$J \cdot M^{-1} \cdot J^T = \{r_{AA} + G \cdot (r_{AB} + r_{BA}) + G^2 \cdot r_{BB}\} \tag{15}$$

PGS and TGS Implementation

The constraint impulses described in Equations (13) and (14) guarantee to satisfy the mimic joint in the absence of any other constraints or contacts that impact $v(t)$. In practice, however, it is not sufficient to resolve a constraint just once per simulation step because a typical use case is multiple constraints that compete with each other. The solution is to perform multiple passes over the list of all constraints. This observation leads to a generalisation of the recipe for computing and applying mimic joint constraint forces.

$$dt \cdot f = -\frac{b_n + J \cdot v_n}{J \cdot M^{-1} \cdot J^T} \tag{16}$$

with b_n denoting the bias velocity recorded at the nth solver iteration and v_n denoting the velocity of the mimic joint as recorded at the nth solver iteration. When the PGS solver is engaged both q_A and q_B remain constant throughout all solver iterations of the same simulation step. As a consequence, the bias velocity b_n and $J \cdot M^{-1} \cdot J^T$ will also remain constant. The only variable that may change from iteration to iteration is v_n.

The TGS solver, on the other hand, purposefully updates q_A and q_B at the end of each solver iteration. As a consequence, b_n requires an update during each solver iteration. The impulse responses r_{AA}, r_{AB}, r_{BB}, r_{BA} and the denominator $J \cdot M^{-1} \cdot J^T$ ought to be similarly updated at the start of each solver iteration. To save computation, however, the impulse responses and the denominator are assumed to be constant during the progress of the TGS solver. This approximation means that Equation (16) may be applied without modification to both TGS and PGS.

Extension To N Joints

The techniques outlined in this document may be readily extended to mimic joints that linearly couple multiple degrees of freedom:

$$C(t) = q_0 + G_1 \cdot q_1 + G_2 \cdot q_2 + ...G_{N-1} \cdot q_{N-1} + \gamma \tag{17}$$

One key difference now is that the inverse mass matrix M^{-1} will be a square matrix of rank N instead of a square matrix of rank 2. Similarly, J will take the form $\{1, G_1, G_2.....G_{N-1}\}$ and the impulse to apply to the ith degree of freedom will be $dt \cdot f \cdot G_i$.

Compliance

The preceding description of mimic joints presents them as a type of hard constraint; that is, a mimic joint will push as hard as required to resolve the constraint equation. The addition of erp permits a degree of compliance provided $0 < erp < 1$. In the absence of geometric error C, however, the mimic joint remains as a hard constraint and will push as hard as required to ensure that the joint velocities are commensurate with the gear ratio. This becomes a problem when the mimic joint interacts with other constraints that are also exceedingly stiff such as contact, joint limit and kinematic drive. If nothing yields it is impossible for the solver to find a balance between all the competing system constraints. The outcome is typically an unstable system. This section shall describe a full model of compliance that allows mimic joints to exhibit any degree of softness or stiffness.

A full model of compliance is remarkably simple and arises from the introduction of an extra term cfm (constraint force mixing [2]):

$$dt \cdot f = -\frac{C_n \frac{erp}{dt} + J \cdot v_n}{r + cfm} \tag{18}$$

with C_n denoting the bias at the nth solver iteration and r having the following definition:

$$r = J \cdot M^{-1} \cdot J^T \tag{19}$$

Multiplying numerator and denominator of Equation (18) by $\frac{1}{cfm}$) reveals the following relationship:

$$dt \cdot f = -\frac{C_n \frac{erp}{cfm \cdot dt} + \frac{J \cdot v_n}{cfm}}{1 + \frac{r}{cfm}} \tag{20}$$

This can be viewed as analogous to the equation governing the impulse of a spring simulated with implicit first order integration:

$$dt \cdot f = -\frac{((x - x_T)k_p \cdot dt + v_0 \cdot dt(dt \cdot k_p + k_d)}{1 + r \cdot dt(dt \cdot k_p + k_d)} \tag{21}$$

with x denoting the spring position, x_T denoting the drive target, $(x - x_T)$ representing the bias of the spring, v_0 the speed of the spring, k_p the stiffness of the spring, k_d the damping of the spring and r the reciprocal of the sprung mass.

A quick inspection of the two forms shown in Equations (20) and (21) allows cfm and erp to be recast as spring constants k_p and k_d:

$$cfm = \frac{1}{dt(dt \cdot k_p + k_d)} \tag{22}$$

$$erp = \frac{dt \cdot k_p}{dt \cdot k_p + k_d} \tag{23}$$

A more natural representation of a spring's properties may be found in natural frequency μ [3] and damping ratio ζ [4]:

$$\mu = \sqrt{k_p \cdot r} \tag{24}$$

$$\zeta = \frac{1}{2}\frac{k_d \cdot r}{\mu} \tag{25}$$

Combining the equations above, it is possible to express cfm and erp in terms of r, μ and ζ:

$$k_p = \frac{\mu^2}{r} \tag{26}$$

$$k_d = 2\frac{\mu\zeta}{r} \tag{27}$$

$$cfm = \frac{1}{dt(dt * kp + kd)} \tag{28}$$

$$erp = \frac{dt \cdot kp}{dt \cdot kp + kd} \tag{29}$$

References

[1] Brian Vincent Mirtich, "Impulse-based Dynamic Simulation of Rigid Body Systems"

[2] Russell Smith, "Open Dynamics Engine User Guide"

[3] wikipedia, "Natural Frequency

[4] wikipedia, "Damping Ratio