A Linear-Time VI for multibody sys — 2019 / 2016.

• 模拟多体系统：VI将多体系统运动方程重构为离散E-L方程，并将forward integration转换为寻根问题。该VI保持系统基本特性，更加精确稳定（动量、能量）。当前多数算法计算复杂度为$O(n^3)$。本文的贡献主要是推导了一种拟牛顿算法解决DEL的寻根问题，并可以很好地拓展到人形机器人。创新点在于DEL的计算估计可以转化为discrete inverse dynamic problem。与此同时，approximation of inverse Jacobian 转化为 continuous forward dynamic problem。受递归牛顿欧拉算法 RNEA 和 Articulated Body Algorithm ABA 启发，推导各个体 DEL eqs. Inverse and Forward Dynamic problem 计算复杂度为$O(n)$。

引言.

我们解决复杂多体系统精确而高效的模拟问题（通常称为正动力学问题）。现有的方法使用拉格朗日形式（广义坐标下动能减势能），并通过最小作用原理推导E-L二阶微分方程。任意时刻系统状态由给定初值的微分方程积分得来。但在长时间保持系统守恒量仍是关键挑战。特别是离散时间仿真，即使使用较先进算法求解ODE，由于误差积累最终产生一些数值耗散（见freebody和钟摆的旋转例子）。为了解决这问题，MEW[1]引入离散Lagrangian用于短时间间隔近似其积分，并通过离散变分得到DEL eqs。他们表明基于DEL的VI是辛的，能量守恒[1,2]。不幸的是，尽管VI稳定，但仍存在计算复杂性。VI将运动方程的积分转化为root-finding problem，在以下方面引入复杂性：(1) the evaluation of DEL (2) computation of their gradient (Jacobian) (3)。J+ 尽管存在估计DEL方程有效算法，但他们不使用广义坐标而是将每一个刚体视为freebody并使用约束力to enforce Joints [3,4,5]。当multi-body sys 和 joint constraints增加，将变得非常复杂。最近Johnson和Murphey[6]提出了一种scalable VI。在广义坐标下表示DEL eqs。By representing the multibody sys as a tree structure in generalized coordinates, They showed that DEL + Gradient + Hessian of the Lagrangian can be calculated recursively # 算法复杂度为$O(n^2)$ ← evaluate DEL. $O(n^3)$ ← compute Jacobian（计算复杂度较高）。本文作者的贡献[3]：利用用于多体系统的线性VI（$O(n)$计算复杂度）。并受RNEA和ABA启发，we formulate the DEL eq individually for each body。通过刚体连接关系树，使用线性时间中的离散逆动力学算法估计DEL，the same recursive representation is applied to update the root using an impulse-based forward dynamics algorithm。? 结合上述方法，we提出$O(n)$拟牛顿法专用于寻DEL根，从而得到线性VI。

discrete inverse dynamic

evaluate · DEL 构建

Lie group formulation

root updating

continuous forward dynamic

?

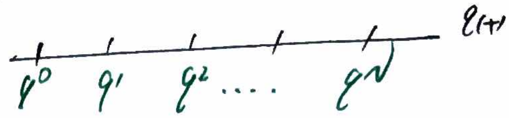
我们将算法与最新的广义坐标VI对比[6]。结果表明，10 dof 快15倍，100 dof 快32倍（相同寻根方法）。10 dof 快5.3倍，100 dof 快53倍（新的拟牛顿法寻根），因为利用了impulse-based forward dynamic algorithm。

背景

工作建立在离散力学和VI。我们需要描述离散力学的标准语言 → Lie group representation.

2.1 VI in GC.

Def (Lagrangian). $L(q,\dot{q})\in R$, generalized coordinates $q\in R^n$. 对于 continuous-time sys. 最小作用原理表明系统遵循最小作用积分 $\int_{t_0}^{t_1} L(q(t),\dot{q}(t))dt$ 轨迹. 然而, 当我们在计算机中模拟时, 力学系统采用离散的时间步. 大概思想是离散力学遵循轨迹, 该离散轨迹上定义近似的作用积分并将其最小化.



定义离散拉格朗日近似 $L(q(t),\dot{q}(t))\Delta t$ 上的积分. $L_d(q^k,q^{k+1})\approx\int_{k\Delta t}^{(k+1)\Delta t} L(q(t),\dot{q}(t))dt$. (1)

定义作用和 (action sum) $\sum_{k=0}^{N-1} L_d(q^k,q^{k+1})$

Minimizing the action sum $r.t$ $\{q^k\}$, 我们得到 E-L eq:

$$D_2L_d(q^{k-1},q^k)+D_1L_d(q^k,q^{k+1})=0. \quad (2)$$

$D_i: R\to R^n$ differential operator w.r.t the i-th parameter.

可以解析计算 D_iL_d. 注意边界位型 q^0 和 q^N 不变 [固定] #

离散力学代替了对 E-L 进行数值积分以模拟轨迹的方法.

解决 root-finding problem.

具体说来 given q^{k-1} 和 q^k, q^{k+1} 由以下函数给定. $f(q^{k+1})=D_2L_d(q^{k-1},q^k)+D_1L_d(q^k,q^{k+1})=0$. (2).

VI vs. Euler & R-K 更具好的能量守恒 因离散版本的 Noether's 定理. VI 的一种几何解释是 DEL eq 起着约束的作用, 迫使离散系统在约束流形上 $f(q^{k+1})=0$ 上. 从这个意义上看, 寻根过程可看作是反馈控制器, 用于下一个时间步找到物理上正确的位型. DEL eq 所构造控制律指示给定的位型距离流形多远. 传统积分方法没有这种机制, 仅考虑当前状态的变化率. 从而导致误差积累.

(2) 式可以用 Newton's method 解决.

A1.

1. 初值 q_0
2. do
3. Evaluate $f(q^{k+1})$ ← $O(n^2)$
4. if $\|f(q^{k+1})\|<\epsilon$. return q^{k+1}
5. update $q^{k+1}\leftarrow q^{k+1}-[J_f(q^{k+1})]^{-1}f(q^{k+1})$ ← $O(n^3)$
6. while num < max.

为了避免计算它, 可以采用各种拟牛顿法来近似 J^{-1}. 在 § 3.2, 引入 linear-time 来近似 $J^{-1}f$.

2.2 VI in SE(3).

在下节我们介绍 linear-time root-finding 算法 (利用为每个 body 重构 DEL eq).

我们从 single rigid body 开始.

刚体位型 $T=\begin{bmatrix}R&p\\0&1\end{bmatrix}\in SE(3)$. $R\in SO(3)$. $p\in R^3$.

The spatial velocity $\mathcal{V}=(\omega,v)\in se(3)$ twist, $\mathcal{V}=\begin{pmatrix}\omega\\v\end{pmatrix}$, $[\mathcal{V}]=\begin{bmatrix}\hat{\omega}&v\\0&0\end{bmatrix}$. $\omega\in so(3)$ angular velocity, $v\in R^3$ linear velocity

Lagrangian of Rigid Body. $L(T,\mathcal{V})=\frac{1}{2}\mathcal{V}^TG\mathcal{V}-P(T)$. $P:SE(3)\to R$, G is spatial inertial matrix #

$G=\begin{bmatrix}I&0\\0&mI\end{bmatrix}$, I ← inertia matrix (com)

类似式(1)，DL表示为 $L_d(T^k, T^{k+1}) \approx \int_{k\Delta t}^{(k+1)\Delta t} L(T, V)\,dt$. 利用梯形公式近似. $L_d(T^k,T^{k+1}) = \frac{\Delta t}{2}L(T^k, V^k) + \frac{\Delta t}{2}L(T^{k+1}, V^k)$ (9).

V^k定义为 $V^k = \frac{1}{\Delta t}\log(\Delta T^k)$，且 $\Delta T^k = (T^k)^{-1}T^{k+1}$ (10)，displacement of rb during t_k and t_{k+1}.

Inverse of exponential map.

为了求解计算 δV^k!

对log map的导数定义为 $\left(\frac{\partial}{\partial T}\log T\right)[W] = \mathrm{dlog}_V([W]\exp(-V))$ (11)，$\mathrm{dlog}_V(W) = \sum_{j=0}^{\infty}\frac{B_j}{j!}ad_V^j(W)$ (12).

[1,14].

$V=\log T$

$W\in se(3)$, arbitrary twist.

Bernoulli numbers [15]

又 $ad_V(W) = [V][W]-[W][V]$，dlog_V矩阵形式可表示为 $[\mathrm{dlog}_V] = \sum_{j=0}^{\infty} \frac{B_j}{j!}[ad_V]^j$，$[ad_V] = \begin{bmatrix}\hat\omega & 0\\ \hat v & \hat\omega\end{bmatrix}$

使用(10),(11)，δV^k可表示为 $\delta V^k = \frac{1}{\Delta t}\mathrm{dlog}_{\Delta t V^k}\left(-{T^k}^{-1}\delta T^k + Ad_{\exp(\Delta t[V^k])}({T^{k+1}}^{-1}T^{k+1})\right)$ (14).

$Ad_T V = T[V]T^{-1}$

$[Ad_T V] = \begin{bmatrix} R & 0\\ \hat p R & R\end{bmatrix}$

再由(9),(10),(14)推出DEL eq. $D_2L_d(T^{k-1},T^k) + D_1 L_d(T^k,T^{k+1}) = 0$ (16.a)

$D_2L_d(T^{k-1},T^k) = -[Ad_{\exp(\Delta t[V^{k-1}])}]^T[\mathrm{dlog}_{\Delta t V^{k-1}}]^T G V^{k-1} + \frac{\Delta t}{2}T^{k*}P(T^k)$. (16.b)

$D_1L_d(T^k,T^{k+1}) = [\mathrm{dlog}_{\Delta t V^k}]^T G V^k + \frac{\Delta t}{2}T^{k*}P(T^k)$ (16.c).

并且对于受控系统，由Lagrange-d'Alembert原理[1]:

$D_2L_d(T^{k-1},T^k)+D_1L_d(T^k,T^{k+1}) + F^k = 0$. (17). 其中 $F^k\in se^*(3)$为力在时间间隔内的积分.

Linear-Time VI.

我们引入新的线性时间VI(在每个t_k求解式(3))，该VI由两linear-time算法组成，用于evaluating DEL eq和updating the root。我们首先以递归方式推导多体系统的DEL方程，从而得到一个线性时间算法来估计$f(q)$. 接下来，我们介绍一种impulsed-based dynamic algorithm计算下一个位型，我们提出一种新的线性时间拟牛顿寻根方法寻DEL根。

基于冲量的动力学算法

3.1. linear-time evaluation of DEL eq.

如果将$f(q)=0$视作动力学约束，则任何非零的$f(q)$都代表了一种residual impulse残余冲量违反了运动方程。这样，估算$f(q)$可以被看作一个离散的逆动力学问题，给定q^{k-1},q^k,q^{k+1}计算残余冲量。我们推导了一个递归DEL方程，使用逆向-牛顿欧拉算法的类似公式[7,8]. 假设多体系统可以看成一个树结构，其中each body最多具有一个父级和任意数量子级通过关节连接。我们目标是拓展(7)式去囊括树结构的动力学.

我们从刚体位型的递归定义开始。{0}为空间固定的惯性标架，{i}表示为树结构系统中i-th刚体标架

{λ(i)}表示第i个体的父结点刚体标架。系统中刚体位型可表示为

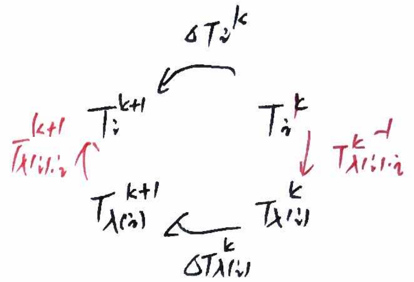
$$T_i^k = T_{\lambda(i)}^k T_{\lambda(i),i}^k \quad (18)$$

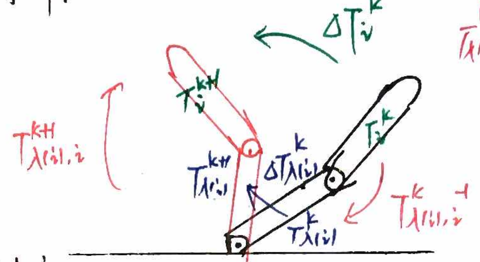
q_i^k ?

其中T_i^k和$T_{\lambda(i)}^k$为惯性标架到{i}、{λ(i)}的变换，$T_{\lambda(i),i}^k$表示从{λ(i)}到{i}的变换(为i-th关节位型q_i^{k+1}的函数).

从18式，刚体位型位移可以写作 $\Delta T_i^k = (T_{\lambda(i),i}^k)^{-1}\Delta T_{\lambda(i)}^k T_{\lambda(i),i}^{k+1}$ (19).

图1.(a)给出了ΔT_i^k和$\Delta T_{\lambda(i)}^k$的位型位移的几何解释.





将(19)代入式(17)，我们计算得到i-th刚体的平均速度

$$V^k = \frac{1}{\Delta t}\log(\Delta T_i^k)$$ 由于log map.

与连续速度$V_i = S_i\dot{q}_i$不同(S_i为joint Jacobian[13])，平均速度计算关于q^{k+1}是隐式的。使用dlog是使DEL关于q^{k+1}隐式的关键。对于多体系统中的刚体，impulse term F^k包括了父连杆传来的冲量，以及传给子连杆的冲量F_c^k，以及环境所施加的额外impulses $F_i^{ext,k}$(如图1(b)).

$$F^k = F_i^k - \sum_{c\in O(i)} Ad^*_{T_{i,c}^k} F_c^k + F_i^{ext,k} \quad (20).$$

注意到F_c^k为体坐标系下的量，所以对F_c^k需要坐标系的坐标变换。$[Ad_{T_{i,c}^{k}}^{-1}]^T F_c^k$。将连杆力代入公式(17)并使用16(b)和16(c)，我们得到了i-th body的运动方程.

$$F_i^k = \mu_i^k - [Ad_{\exp(\Delta t[V_i^{k-1}])}]^T \mu_i^{k-1} + \sum_{c\in O(i)}[Ad_{T_{i,c}^{k}}^{-1}]^T F_c^k - F_i^{ext,k} \quad (21a).$$

$$\mu_i^k = [dlog_{\Delta t V_i^k}]^T G_i V_i^k \quad (21b).$$

其中μ_i为body i的离散动量，关节到达q^{k+1}所需的广义impulse由$S_i^T F_i^k$给出，其中$S_i \in R^{6\times n_i}$为i-th的关节Jacobian[13].

若考虑关节摩擦与驱动Q_i^k，残余冲量可计算： $f_i = S_i^T F_i^k - Q_i^k \in R^{n_i}$?

算法2总结了递归过程，我们将其称为离散递归牛顿欧拉算法，它由前向、反向递归构成。前向计算每个body的速度，反向计算关节间的力。假设每个关节自由度为1，则每次计算复杂度为O(n).

A2. DRNEA.

1. for $i = 1 \to n$ do.
2. $\quad T_{\lambda(i),i}^{k+1}$ = function of q_i^{k+1}.
3. $\quad \Delta T_i^k = [T_{\lambda(i),i}^{k}]^{-1} \Delta T_{\lambda(i)}^k T_{\lambda(i),i}^{k+1}$
4. $\quad V^k = \frac{1}{\Delta t} \log(\Delta T_i^k)$.
5. end for
6. for $i = n \to 1$ do.
7. $\quad \mu_i^k = [d\log_{\Delta t} V_i^k]^T G_i V_i^k$
8. $\quad F_i^k = \mu_i^k - [Ad\, \exp_{\Delta t}(V_i^{k-1})]^T \mu_i^{k-1} - F_i^{ext,k} + \sum_{c \in C(i)} [Ad_{T_{i,c}^{k}}^{-1}]^T F_c^k$.
9. $\quad f_i = S_i^T F_i^k - Q_i^k$.
10. end for

3.2. Root updating

除计算 $f(q^{k+1})$，半隐型算法还需计算 Jacobian，这也是每次迭代计算中的瓶颈。在这里，我们提出一种递归冲量方法，可以在线性时间内更新根。我们记当前迭代为 l，下个时间步位型估计为 $q_{(l)}^{k+1}$，对 DEL eq 计算残余冲量 $f(q_{(l)}^{k+1}) = e_{(l)}$。如果 $e_{(l)}$ 为 0 或小于容许值，$q_{(l)}^{k+1}$ 为下一个位型满足 DEL 方程。反之，$e_{(l)}$ 可用于多步聚算法1的 lines 5。如果我们施加负 residual force. $-e_{(l)}/\Delta t$，那么我们将接近 q^{k+1}，Applying such force to the sys can be done by continous forward dynamics in linear-time [8].

利用 $\frac{1}{\Delta t}(q^k - q^{k-1})$ 近似 $\dot q^k$，连续情形运动力学方程用于计算加速度 $\ddot q^k = M^{-1}(q^k)(-C(q^k, \dot q^k)\dot q^k + Q)$ (23).

再使用二阶中心差分 $q^{k+1} = \Delta t^2 \ddot q^k + 2q^k - q^{k-1}$，并结合 $e_{(l)}$ 改善对根的估计

$$q_{(l+1)}^{k+1} = \Delta t^2 M^{-1}(q^k)\left(-C(q^k, \dot q^k)\dot q^k + Q - \sum_{m=0}^{l} \frac{e_{(m)}}{\Delta t}\right) + 2q^k - q^{k-1} \quad (24).$$

化简合并. $q_{(l+1)}^{k+1} = q_{(l)}^{k+1} - \Delta t M^{-1}(q_{(l)}^k) e_{(l)}$. 得到更新规则.

其中 $\Delta t M^{-1}(q_{(l)}^k) e_{(l)}$ 可以使用 Featherstone [8] 提出的 recursive impulse-based dynamics. (ABI 算法：articulated body inertia algorithm). 计算，复杂度为 $O(n)$. 特别的，ABI 是前向动力学算法计算式(23). 让我们让 $\dot q \equiv 0$.（消除 coriolis 力）及 $Q \equiv \Delta t e_{(l)}$，ABI 将返回精确的 $\Delta t M^{-1}(q^k) e_{(l)}$. 与算法相对比，$J^{-1}$ 将由 $\Delta t M^{-1}$ 近似. 称该方法为 RIQN（递归 impulse-based Quasi-Newton method).

A3. RIQN.

1. initial guess. q_0^{k+1}
2. do
3. $\quad$ use DRNEA to evaluate $e \leftarrow f(q^{k+1})$ typo?
4. $\quad$ if $\|e\| < \epsilon$ return q^{k+1}
5. $\quad$ use ABI to compute $\Delta t M^{-1}(q^k)e$. $M(q^{k+1})$?
6. $\quad$ update $q^{k+1} \leftarrow q^{k+1} - \Delta t M^{-1}(q^k) e$.
7. while num < max.

3.3. Initial Guess.

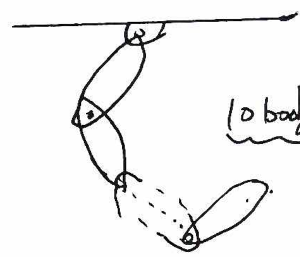
牛顿型算法需要较好初值，对于RZQN的初值，有以下三种方法：① 使用当前位型作为下一位型的初值 $q_{(0)}^{k+1} = q^k$.

② 由显式欧拉积分 $q_{(0)}^{k+1} = q^k + \Delta t \cdot \dot{q}^k$，$\dot{q}^k \approx \frac{1}{\Delta t}(q^k - q^{k-1})$. ③ 由 $\ddot{q}^k = M^{-1}(c + \tau)$ 计算加速度，并利用半隐式Euler积分

$\dot{q}^{k+1} = \dot{q}^k + \Delta t \ddot{q}^k$，再由 $q_{(0)}^{k+1} = q^k + \Delta t \dot{q}^{k+1}$.

数值实验

实现RZQN和DRNEA，并与现有算法进行对比（所有实验time-step 1ms）.



10 body. 重力作用下

4.1 算法在DART [17,18] 实现. DART是多体系统的C++开源库. 所有代码已在github.

4.2 首先我们验证了VI的能量特性：例如图，由连杆关节构成. RZQN vs Semi-implicit method.

易于实现的标准算法.

4.3 性能对比.

影响VI速度因素是 ① DEL eq. $f(q^{k+1})$.

② J^{-1}.

使用RZQN解决 ~~结合~~

对于①，我们将DRNEA与scalable VI对比.

① DRNEA vs SVI. ② Newton + DRNEA

③ Broyden + DRNEA ④ RZQN + DRNEA

对于②，我们比较所提的RZQN与Newton's及Broyden method.

Newton's method需exact Jacobian of DEL，当结合DRNEA，成本更高.

我们也推导了计算DEL eq w.r.t q^{k+1} 导数的递归算法，见附录.

4.4 收敛性.

与Newton's相比，我们考虑了RZQN的收敛速度. 检查误差 $f(q_{(i)}^{k+1}) = e$ 的收敛（在一个时间步的DEL eq迭代中）. 对于为了

Newton + DRNEA > RZQN + DRNEA

定量的收敛（quantitatively visible convergence）. 使用 $q_0^{k+1} = 0$ 代替§3.3中初值. 图4a显示在容许范围内，RZQN收敛慢于Newton's method. 这是符合预期的. 牛顿法具有 quadratic convergence rate，理论上比拟牛顿快. 但在4.3节中，DRNEA + RZQN

二次收敛 > 拟牛顿

表现更好（cpu time）. 图4b显示每时间步的平均迭代次数. Newton's相比于RZQN需更少的迭代 #.

结论

未来方向之一是将linear-time VI应用到约束动力学系统. 如contacts or closed-loop chains. 在动力学中处理这类约束的标准方法是求解DEL方程，约束使用Lagrangian multipliers [1,2]. 为了继承RZQN优势，一种可能的扩展是使用impulse-based forward dynamic类似思路来求解约束力. RZQN实现也可利用变时间步长的VI改善. 可利用变时间步长获得更好性能. 但是，改变时间步长也许会对仿真产生负面影响 [15,21] #.