The SpaceDyn: a MATLAB toolbox for space and mobile robots

火箭搬砖博士
Dr. Rocket & Bricks

摘要—The SpaceDyn is an open-source MATLAB toolbox designed for the kinematic and dynamic analysis and simulation of articulated multi-body systems with a moving base. It is particularly suited for modeling space and mobile robots, including free-flying satellites with manipulators, wheeled robots, and legged systems operating in both gravitational and microgravity environments. The toolbox supports open-chain systems with tree-structured topologies and offers flexible coordinate system definitions without relying on the Denavit-Hartenberg convention. It utilizes direction cosine matrices for singularity-free attitude representation and provides comprehensive tools for forward and inverse dynamics computations. SpaceDyn enables users to simulate complex interactions such as ground contact, joint torques, and external forces, making it a valuable tool for research in robotics, aerospace engineering, and planetary exploration. Practical applications include simulations of the ETS-VII space robot, flexible-base manipulators, and asteroid landing missions. Although it has limitations in handling closed-chain systems and dynamically changing contact points, its modular structure allows for user extensions to address specific constraints.

SpaceDyn 是一个开源的 MATLAB 工具箱，用于对带有移动基座的铰接式多体系统进行运动学与动力学分析及仿真。该工具箱特别适用于建模空间和移动机器人，包括带有机械臂的自由飞行卫星、轮式机器人和在重力或微重力环境下运动的步行机器人系统。它支持具有树状拓扑结构的开链系统，并提供了灵活的坐标系定义方法，而不依赖于 Denavit-Hartenberg 约定。该工具箱采用方向余弦矩阵进行无奇异的姿态表示，并提供了用于正向和逆向动力学计算的全面工具。SpaceDyn 使用户能够模拟复杂的交互作用，如地面接触、关节扭矩和外部力，使其成为机器人技术、航空航天工程和行星探测研究中非常有价值的工具。实际应用包括对 ETS-VII 空间机器人、柔性基座机械臂和小行星着陆任务等的仿真。尽管它在处理闭链系统和动态变化的接触点方面存在局限，但其模块化结构允许用户通过扩展来应对特定约束。

I. Introduction

THE Spacedyn is a MATLAB Toolbox for the kinematic and dynanhic analysis and simulation of articulated multi-body systems with a moving base. Examples of such systems are a satellite with mechanical appendages, a free-flying space robot, a wheeled mobile robot, and a walking robot, all of which makes motions in the environment with or without gravity.

Spacedyn 是一个 MATLAB 工具箱，用于对具有移动基座的铰接式多体系统进行运动学和动力学分析与仿真。此类系统的示例包括带有机械附件的卫星、自由飞行的太空机器人、轮式移动机器人和步行机器人，所有这些系统都能在有重力或无重力的环境中运动。

This toolbox can handle open chain systems with topological tree configuration. A parallel manipulator, for example, then cannot be supported directly. A walking robot contacting on the ground with more than two legs or limbs at a time sseems to form a closed chain including the ground, however, we can handle such asystem with a proper model of ground contact at each contact point. Parallel manipulators can be treated with virtual cut of a kinematic chain and a corresponding virtual force model.

该工具箱可处理具有拓扑树结构的开链系统。例如，并联机器人无法直接得到支持。一次有两条以上腿或肢体接触地面的步行机器人似乎会形成一个包含地面的闭链，然而，我们可以通过在每个接触点采用适当的地面接触模型来处理这样的系统。并联机器人可以通过对运动链进行虚拟切割和相应的虚拟力模型来处理。

Some academic papers regarding this toolbox is published by Kazuya Yoshidahis and his co-author(s) . For the technical points of this software, please consult these papers as well as the following chapters of this document.

关于这个工具箱的一些学术论文已由 Kaztuya Yoshidahis 及其合著者发表。关于该软件的技术要点，请参考这些论文以及本文档的以下章节。

We hope that you could find this toolbox useful.

我们希望您能发现这个工具箱有用。

A. *Release Note* 版本记录

This is "The Spacedyn: a MATLAB toolbox for space and mobile robots" version 1.0, release 1.0, as of October 7, 1999.

这是《The Spacedyn：用于空间和移动机器人的 MATLAB 工具箱》1.0 版，1999 年 10 月 7 日发布的 1.0 版本。

This is a freeware to follow the same copyright policy of other softwares in the category termed "freeware."

这是一款免费软件，遵循与“免费软件”类别中其他软件相同的版权政策。

We develop and distribute this software from a purly academic motivation. You can download and use it for your academic purpose only. Any of commercial use is not allowed. You can also modify and redistribute your version at your own responsibility, but in such a case it is an etiquette to refer this toolbox as the original work.

我们纯粹出于学术动机开发和分发此软件。您只能出于学术目的下载和使用它。任何商业用途均不被允许。您也可以自行负责修改并重新分发您的版本，但在这种情况下，将此工具箱视为原始作品进行引用是一种礼节。

We do not take any responsibility on any damage around you due to this software, but we would appreciate it if you could report us any bugs or mistakes, or fallacy. Any feedback is helpful for us to improve our knowledge, and update this toolbox.

对于因本软件导致您周围发生的任何损害，我们不承担任何责任。但如果您能向我们报告任何错误、失误或谬论，我们将不胜感激。任何反馈都有助于我们增进知识并更新此工具箱。

When you make an academic work by using this toolbox and publish the results in papers and technical reports, we would most appreciate it if you could report it and cite this toolbox in a proper manner.

当您使用本工具箱完成学术工作，并在论文和技术报告中发表成果时，如果您能进行汇报并以适当方式引用本工具箱，我们将不胜感激。

B. *Technical Memorandums* 技术备忘录

We developed this toolbox motivated and inspired by **Robotics toolbox** developed by Peter I. Coke, which is available from http://www.mathworks.com/ftp/miscv4.shtml. We took one m-file (cross.m) and use it as the original is, but our toolbox as a whole, does not have compatibility with the Peter Coke's toolbox unfortunately.

我们开发这个工具箱的动机和灵感来自于 Peter I. Coke 开发的机器人工具箱，该工具箱可从 http://www.mathworks.com/ftp/miscv4.shtml 获取。我们采用了一个 m 文件（cross.m）并保持其原始状态使用，但遗憾的是，我们的工具箱整体上与彼得 · 科克的工具箱不兼容。

This toolbox is for the use with MATLAB 5.0 or higher. We use three dimensional array which is not supported in version 4 or lower.

此工具箱适用于 MATLAB 5.0 或更高版本。我们使用的三维数组在 4.0 版本或更低版本中不受支持。

For mathematical symbols, we give the name of variables with more than two letters. For example, vector $\boldsymbol{r}$ and $\boldsymbol{c}$ are coded by RR and cc, respectively. This is to avoid the confusion with control variables such as i, j, k, or l, m, n, etc, which are frequently used as iteration or array counters. But there is an exception: the symbol q is used for the joint variable vector $\boldsymbol{q}$.

对于数学符号，我们用两个以上字母来命名变量。例如，向量 $\boldsymbol{r}$ 和 $\boldsymbol{c}$ 分别编码为 RR 和 cc。这是为了避免与诸如 i, j, k 或 l, m, n 等控制变量混淆，这些变量经常用作迭代计数器或数组计数器。但有一个例外：符号 q 用于关节变量向量 $\boldsymbol{q}$ 。

Since MATLAB doesn't allow 0 for array index, we use R0 and c0 instead of RR[0] and cc[0], for example.

由于 MATLAB 不允许数组索引为 0，例如，我们使用 R0 和 c0 来代替 RR[0] 和 cc[0]。

We use both the input variables and the global variables to pass the values to m-file functions. The input variables inside the braces are the variables changing time to time, such as joint angles, positions, orientations, and so on. The global variables are the ones holding constant once the model is given, such as topological description matrices, kinematic and dynamic parameters.

我们使用输入变量和全局变量将值传递给 m 文件函数。花括号内的输入变量是随时间变化的变量，如关节角度、位置、方向等。全局变量是在给定模型后保持不变的变量，如拓扑描述矩阵、运动学和动力学参数。

We assume the system composed of $n+1$ bodies and connected by n joints. Let the body 0 be a reference body. Multiple branches can attach on any single body, as far as the system keeps a topological tree configuration. There must be a single joint between two bodies. We call a terminal point or the point of interest such as manipulator hand as endpoint. Each body, except body 0, can have one endpoint at maximum. In this document, the terms body and link are the same.

我们假设该系统由 $n+1$ 个刚体通过 n 个关节连接而成。设刚体 0 为参考刚体。只要系统保持拓扑树结构，多个分支可以连接到任何单个刚体上。两个刚体之间必须有一个关节。我们将终端点或诸如机械臂末端这样的感兴趣点称为端点。除刚体 0 外，每个刚体最多可有一个端点。在本文档中，“刚体”和“连杆”这两个术语含义相同。

The toolbox allows force/torque input on (1) the centroid of the reference body, (2) each endpoint, and (3) each joint. The toolbox computes the position, velocity and acceleration of (1) the centroid of the reference body, (2) the centroid of each body, (3) each endpoint, and (4) each joint.

该工具箱允许在以下位置输入力/扭矩：(1) 参考刚体的质心，(2) 每个端点，以及 (3) 每个关节。该工具箱会计算以下位置的位置、速度和加速度：(1) 参考刚体的质心，(2) 每个刚体的质心，(3) 每个端点，以及 (4) 每个关节。

Computation of input force/torque are open to user programming. You can arbitrary decide each joint as either active or passive one. If you give always zero torque, such as $\tau_i = 0$, the corresponding joint behaves as a free joint. Or if you give such a torque as:

输入力/扭矩的计算可供用户编程实现。你可以随意将每个关节设定为主动关节或被动关节。如果你始终给出零扭矩，例如 $\tau_i = 0$，那么相应的关节就会像自由关节一样运动。或者，如果你给出这样一个扭矩：

$$\tau_i = -Kq_i - D\dot{q}_i$$

the joint behaves as a passive visco-elastic joint. You can treat even a flexible link, by modeling it as a discrete successive chain of rigid links connected by elastic joints.Of course, you can give any arbitrary control torque determined by your own control law, on all or arbitrary selected joints.

该关节表现为被动粘弹性关节。通过将柔性连杆建模为一系列由弹性关节连接的离散刚性连杆，你甚至可以处理柔性连杆。当然，你可以根据自己的控制律，在所有或任意选定的关节上施加任意控制转矩。

We know that the Denavit-Hartenberg notation is commonly used in the field of manipulator kinematics with the advantage of unique allocation of coordinate systems with minimum parameters. But we know that the DH sometimes locates the coordinate origin away from the location of an actual joint. From the dynamics point of view, the angular velocity and the inertia tensor should be defined around the corresponding joint axis or body centroid. We then do not use the DH notation but introduce a rule to define the coordinate systems with more flexibility. Our rule locates the origin of the coordinate system on each joint and orients the primary axes so that the inertia tensor should be simpler, but admits 3 position and 3 orientation parameters among two successive coordinate systems.

我们知道，Denavit-Hartenberg（DH）符号法在机械臂运动学领域中被广泛使用，其优点是能够以最少的参数唯一地分配坐标系。但是我们也知道，DH 有时会将坐标原点设置在实际关节位置之外。从动力学的角度来看，角速度和惯性张量应该围绕相应的关节轴或物体质心来定义。因此，我们不使用 DH 符号法，而是引入一种规则来更灵活地定义坐标系。我们的规则将坐标系的原点定位在每个关节上，并对主轴进行定向，以便使

惯性张量更简单，但在两个连续的坐标系之间允许有 3 个位置参数和 3 个方向参数。

For the representation of attitude or orientation, we use 3 by 3 direction cosine matrices, coded with a symbol A. For example, A0 is the direction cosines to represent the attitude of the body 0. For the other bodies, a matrix AA is used. The advantage of direction cosine is (1) singularity free, (2) we can easily derive Roll-Pitch-Yaw angles, Euler angles, or quartanions, and (3) it is easy to find the mathematical relationship with angular velocity.

对于姿态或方向的表示，我们使用 3×3 的方向余弦矩阵，用符号 A 表示。例如，A0 是表示物体 0 姿态的方向余弦。对于其他物体，使用矩阵 AA。方向余弦的优点是：(1) 无奇异点，(2) 我们可以很容易地推导出滚转角、俯仰角、偏航角、欧拉角或四元数，(3) 很容易找到与角速度的数学关系。

On the other hand, we frequently need Roll-Pitch-Yaw representation also. For RPY angles, we use the symbol $\mathbf{Q}$. For example, in order to express the twisting angles between two coordinate systems, we consider α (roll) around x axis, β (pitch) around y axis, then γ (yaw) around z axis. The set of these angles are coded by $\mathbf{Q}_i$.

另一方面，我们也经常需要使用滚转-俯仰-偏航（Roll-Pitch-Yaw）表示法。对于滚转、俯仰、偏航角，我们使用符号 $\mathbf{Q}$。例如，为了表示两个坐标系之间的扭转角度，我们考虑绕 x 轴的 α（滚转）角、绕 y 轴的 β（俯仰）角，然后是绕 z 轴的 γ（偏航）角。这些角度的集合用 $\mathbf{Q}_i$ 来表示。

Weak points: The SpaceDyn is not good at dealing with kinematic constraints other than joint axes. It is also weak at dealing with the problems in which a contact point is dynamically changing. For those problems, a good user programming is required to model the constraint forces.

缺点：SpaceDyn 在处理除关节轴之外的运动学约束方面表现不佳。在处理接触点动态变化的问题时也较为薄弱。对于这些问题，需要用户进行良好的编程来对约束力进行建模。

II. Variables Used in this Toolbox
本工具箱中使用的变量

We define the variables as listed in the following pages. In the lists, the mathematical definitions and the expression in programming codes are compared. In the programming codes, i, j, k are array counters, where i, j are 1 to an arbitrary number (usually upto n but depends on each variable, please consult chapter 3 for details), but k is used for 1:3 only. The variables are shown in Tables I, II, and III.

III. Notes on the Modeling of Multibody Dynamics
多体动力学建模笔记

A. *Mathematical Graph Representation* 数学图形表示

In order to mathematically describe the interconnection of the bodies, we adopt a method from mathematical graph theory. We simplify it with additional rules on the assignment of link and joint indices, so that we can easily and uniquely construct two types of matrices (vectors); a connection index $\boldsymbol{B}$ and incidence matrices $\boldsymbol{S}, \boldsymbol{S}_0$, and $\boldsymbol{S}_e$.

为了从数学上描述物体之间的相互连接，我们采用了数学图论中的一种方法。我们通过对连杆和关节索引的分配制定额外规则来简化它，以便我们能够轻松且唯一地构建两种类型的矩阵（向量）：连接索引 $\boldsymbol{B}$ 以及关联矩阵 $\boldsymbol{S}, \boldsymbol{S}_0$, and $\boldsymbol{S}_e$。

The procedure from indices assignment to matrix construction is summarized as follows:

从索引分配到矩阵构建的过程总结如下：

1. Assign the indices of links and joints in the following manner.

按以下方式分配连杆和关节的索引。

(1) Reference body is denoted by link 0.

参考体用连杆 0 表示。

(2) The index of a link i in the physical connection between link 0 and link (j) must be $0 < i < j$.

在连杆 0 与连杆 (j) 之间的物理连接中，连杆 (i) 的索引必须满足 $0 < i < j$。

(3) One link can have multiple connections with other links. As a result of the above statement, a link $i (i > 0)$ has one lower connection (which index

表 I
LIST OF VARIABLES 1

Key	Symbol	Interpretation
BB(i)	$\boldsymbol{B}_i$	connection index
		连接索引
SS(i,j)	$\boldsymbol{S}_{ij}$	incidence matrix of directed graph
		有向图的关联矩阵
S0(i)	$\boldsymbol{S}_{0i}$	incidence matrix of directed graph for body 0
		物体 0 的有向图关联矩阵
SE(i)	$\boldsymbol{S}_{ei}$	incidence matrix of directed graph for end-links
		末端连杆的有向图关联矩阵
J_type(i)	R or P	joint type: rotational or prismatic
		R 或 P 关节类型：旋转或平移
cc(k,i,j)	$\boldsymbol{c}_{ij}$	link vector: from centroid of link i to joint j
		连杆向量：从连杆 i 的质心到关节 j
ll(k,i,j)	$\boldsymbol{\ell}_{ij}$	link vector: from joint i to joint j
		连杆向量：从关节 i 到关节 j
ce(k,i)	$\boldsymbol{c}_{ie}$	link vector: from centroid of link i to end-point
		连杆向量：从连杆 i 的质心到末端点
le(k,i)	$\boldsymbol{\ell}_{ie}$	link vector: from joint i to end-point
		连杆向量：从关节 i 到末端点
c0(k,i)	$\boldsymbol{c}_{0i}$	link vector: from centroid of link 0 to joint i
		连杆向量：从连杆 0 的质心到关节 i

is smaller than i) and zero or one or multiple upper connection(s) (which index is greater than i).

一个连杆可以与其他连杆有多个连接。根据上述说明，连杆 $i(i>0)$ 有一个下游连接（其索引小于 i）以及零个、一个或多个上游连接（其索引大于 i）。

(4) There is one single joint interconnecting two links.

有一个单独的关节连接两个连杆。

(5) Indices of joints begin from 1, and joint i is physically attached on link i.

关节的索引从 1 开始，关节 i 在物理上连接到连杆 i。

(6) Then the joint interconnecting link 0 and link 1 is joint 1, and the joint interconnecting link 1 and link 5 is joint 5, for example.

例如，连接连杆 0 和连杆 1 的关节是关节 1，连接连杆 1 和连杆 5 的关节是关节 5。

2. Connection index vector $\boldsymbol{B}$ is used to find the lower connection of a link, which exists uniquely for a link i $(i>0)$.

连接索引向量 $\boldsymbol{B}$ 用于查找链路的下连接，对于链路 i（$i>0$），该下连接是唯一存在的。

(1) The element of $\boldsymbol{B}_i$ is the index of the lower connection of link i.

$\boldsymbol{B}_i$ 的元素是连杆 i 下连接的索引。

3. Incidence matrix $\boldsymbol{S}$ is used to find the upper connection of a link, which may not exist or exist one or more.

关联矩阵 $\boldsymbol{S}$ 用于查找链路的上层连接，该连接可能不存在，也可能存在一个或多个。

Each element of $\boldsymbol{S}_{ij}$ $(i,j=1,\ldots,n)$ is defined by:

$\boldsymbol{S}_{ij}$ $(i,j=1,\ldots,n)$ 的每个元素定义如下：

$$\boldsymbol{S}_{ij}=\begin{cases}+1 & (\text{ if } \quad i=\boldsymbol{B}_j)\\ -1 & (\text{ if } \quad i=j)\\ 0 & (\text{ otherwise })\end{cases}$$

4. Define a matrix $\boldsymbol{S}_{0j}$ $(j=1,\ldots,n)$ as:

定义一个矩阵 $\boldsymbol{S}_{0j}$ $(j=1,\ldots,n)$ 如：

$$\boldsymbol{S}_{0j}=\begin{cases}+1 & (\text{ if } 0=\boldsymbol{B}_j)\\ 0 & (\text{ otherwise })\end{cases}$$

This represents a flag to indicate if link i has a connection with the reference link 0.

这表示一个标志，用于指明连杆 i 是否与连杆 0 有连接。

5. Also define a matrix $\boldsymbol{S}_{ej}(j=1,\ldots,n)$ as:

同样定义一个矩阵 $\boldsymbol{S}_{ej}(j=1,\ldots,n)$ 如：

$$\boldsymbol{S}_{ej} = \begin{cases} +1 & (\text{ if link } j \text{ is a terminal link }) \\ 0 & (\text{ otherwise }) \end{cases}$$

This represents a flag to indicate if link i is a terminal endlink.

这表示一个标志，代表连杆 i 是否为末端连杆。

Figure 1 depicts an example of a system with multiple branches numbered by the rule presented here.

图 1描绘了一个具有多个分支的系统示例，这些分支按照此处给出的规则进行编号。

B. *Coordinate System* 坐标系

Let the inertial reference coordinate frame be denoted by $\{\Sigma_I\}$. (The expression Σ_I is use to represent the basis of a coordinate frame, a set of unit vectors: vectrix）, which is stationary or lineary moving with constant velocity in the inertial space. It is not physically precise but we sometime consider the orbital fixed frame as the inertial frame in the sense of practice.

设惯性参考坐标系记为 $\{\Sigma_I\}$[1]。（表达式 Σ_I 用于表示坐标系的基，即一组单位向量：vectrix），该坐标

表 II

List of variables 2

Key	Symbol	Interpretation
vv(k,i), vd(k,i)	$\boldsymbol{v}_i, \dot{\boldsymbol{v}}_i$	linear velocity and acceleration of the centroid of link i
		连杆 i 质心的线速度和加速度
v0(k), vd0(k)	$\boldsymbol{v}_0, \dot{\boldsymbol{v}}_0$	linear velocity and acceleration of the centroid of link 0
		连杆 0 质心的线速度和加速度
ww(k,i), wd(k,i)	$\boldsymbol{\omega}_i, \dot{\boldsymbol{\omega}}_i$	angular velocity and acceleration of link i around its centroid
		连杆 i 绕其质心的角速度和角加速度
w0(k), wd0(k)	$\boldsymbol{\omega}_0, \dot{\boldsymbol{\omega}}_0$	angular velocity and acceleration of link 0 around its centroid
		连杆 0 绕其质心的角速度和角加速度
q(i)	q_i	joint angle (or prismatic displacement)
		关节角度（或移动位移）
qd(i), qdd(i)	$\dot{q}_i, \ddot{q}_i$	velocity and acceleration of joint
		关节的速度和加速度
Pe(k,h), Qe(k,h)	$\boldsymbol{p}_e, [\boldsymbol{a}_e \boldsymbol{s}_e \boldsymbol{n}_e]$	position and orientation of end-point
		末端点的位置和方向
FF(k,i), TT(k,i)	$\boldsymbol{F}_i, \boldsymbol{N}_i$	(inertial) force and torque applying on the centroid of link i
		作用在连杆 i 质心的（惯性）力和力矩
Fj(k,i), Tj(k,i)	$\boldsymbol{f}_i, \boldsymbol{n}_i$	force and torque applying on joint i
		作用在关节 i 上的力和力矩
Fe(k,i), Te(k,i)	$\boldsymbol{f}_{ei}, \boldsymbol{n}_{ei}$	(external) force and torque applying on end-point i
		作用在末端点 i 上的（外部）力和力矩
F0(k), T0(k)	$\boldsymbol{F}_0, \boldsymbol{N}_0$	force and torque applying on the centroid of link 0
		作用在连杆 0 质心的力和力矩
tau(i)	$\boldsymbol{\tau}_{mi}$	joint torque
		关节扭矩
mass	w	total mass
		总质量
m0, m(i)	m_i	mass of link 0 and link i
		连杆 0 和连杆 i 的质量
inertia0, inertia(k,(i*3-2):(i*3))	$\boldsymbol{I}_0, \boldsymbol{I}_i$	inertia tensor of link 0 and link i
		连杆 0 和连杆 i 的惯性张量
inertia(k,n*3)	$[\boldsymbol{I}_1, \boldsymbol{I}_2, \ldots, \boldsymbol{I}_n] \in R^{3n\times 3}$	a collection of inertia tensors $\boldsymbol{I}_i$
		惯性张量集合
HH	$\boldsymbol{H}$	(augmented) system inertia matrices
		（增广）系统惯性矩阵
Force	$[\boldsymbol{F}_0^T, \boldsymbol{N}_0^T, \boldsymbol{\tau}^T]^T$	(augmented) system generalized force
		（增广）系统广义力
Force0	$\boldsymbol{d}$	velocity dependent non-linear force
		速度相关的非线性力

表 III
LIST OF VARIABLES 3

Key	Symbol	English
JJ_t(k, (i*n-(n-1)):(i*n))	$\boldsymbol{J}_{Ti}$	jacobian matrix for linear velocity of link centroid i
		连杆质心 i 线速度的雅可比矩阵
JJ_r(k, (i*n-(n-1)):(i*n))	$\boldsymbol{J}_{Ri}$	jacobian matrix for angular velocity of link centroid i
		连杆质心 i 角速度的雅可比矩阵
JJ_te(k, (i*n-(n-1)):(i*n))	$\boldsymbol{J}_{Tei}$	jacobian matrix for linear velocity of endpoint i
		末端点 i 线速度的雅可比矩阵
JJ_re(k, (i*n-(n-1)):(i*n))	$\boldsymbol{J}_{Rei}$	jacobian matrix for angular velocity of endpoint i
		末端点 i 角速度的雅可比矩阵
JJ_tg, HH_w, HH_wq, HH_q	$\boldsymbol{J}_{Tg}, \boldsymbol{H}_{\omega}, \boldsymbol{H}_{\omega q}, \boldsymbol{H}_{q}$	submatrices comprising the inertia matrix $boldsymbol H$
		构成惯性矩阵 $boldsymbol H$ 的子矩阵
RR(i)	$\boldsymbol{r}_i$	position vector of the centroid of link i
		连杆 i 质心的位置矢量
R0	$\boldsymbol{r}_0$	position vector of the centroid of endpoint
		末端质心的位置矢量
A0	$\boldsymbol{C}_0^T$ or ${}^I\boldsymbol{A}_0$	direction cosines to represent orientation of link 0
		表示连杆 0 方位的方向余弦
AA(k, (i*3-2):(i*3))	${}^I\boldsymbol{A}_i$	coordinate transformation matrix: $\{\Sigma_i\} \to \{\Sigma_I\}$ (the coordinate transformation matrix is mathematically equivalent to direction cosines)
		坐标变换矩阵：$\{\Sigma_i\} \to \{\Sigma_I\}$ （注意：坐标变换矩阵在数学上等同于方向余弦。）
AA(k, n*3)	$[{}^I\boldsymbol{A}_1, {}^I\boldsymbol{A}_2, ..., {}^I\boldsymbol{A}_n] \in R^{3n\times 3}$	a collection of coordinate transformation matrices
		一组坐标变换矩阵
Qi(k, i)	$(\alpha_i, \beta_i, \gamma_i)$	RPY angles to represent the twists from $\{\Sigma_{B(i)}\}$ to $\{\Sigma_i\}$
		表示从 $\{\Sigma_{B(i)}\}$ 到 $\{\Sigma_i\}$ 的扭转的滚转、俯仰、偏航角
Gravity	$\boldsymbol{g}$	vector for gravitational acceleration
		重力加速度矢量
time	t	time
		时间

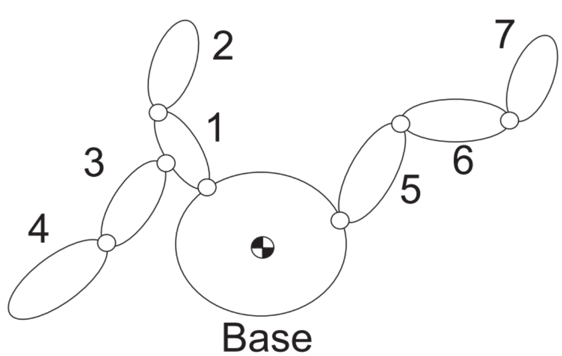


图 1. Sample System

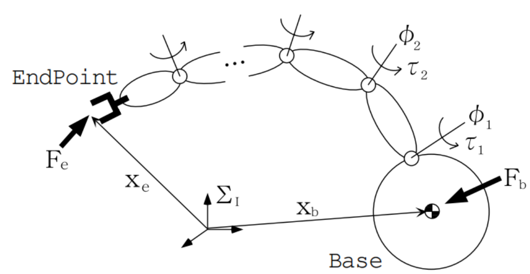


图 2. Multibody System

系在惯性空间中静止或以恒定速度做线性运动。这在物理上并不精确，但在实际应用中，我们有时会将轨道固定坐标系视为惯性系。

We also define moving coordinate frames fixed on each link. We do NOT take the Denavit-Hartenberg convention but introduce a simpler and flexible rule to define the link coordinates. Our rule is as follows:

我们还定义了固定在每个连杆上的移动坐标系。我们不采用 Denavit-Harttenberg 惯例，而是引入一种更简单灵活的规则来定义连杆坐标系。我们的规则如下：

1. If the joint i is revolution, then

如果关节 i 是转动关节，那么

(1) locate the origin of the coordinate system $\{\Sigma_I\}$ on joint i and fixed it to the link i,

将坐标系 $\{\Sigma_I\}$ 的原点定位在关节 i 上，并将其固定在连杆 i 上，

(2) set its z-axis to coincide with the joint rotation axis,

将其 z 轴设置为与关节运动轴重合，

(3) orient its x-axis toward joint $i+1$ or the direction in which the inertia tensor is expressed easier.

将其 x 轴定向为指向关节 $i+1$ 或使惯性张量表达更简便的方向。

2. If the joint i is prismatic, then

如果关节 i 是移动关节，那么

(1) locate the origin of the coordinate system Σ_i on the place when joint i has zero displacement and fixed it to the link $i-1$,

将坐标系 Σ_i 的原点定位在关节 i 位移为零时的位置，并将其固定在连杆 $i-1$ 上，

(2) set its z-axis to coincide with the joint displacement axis, with the positive direction,

将其 z 轴设置为与关节位移轴重合，并确定正方向，

(3) orient its x-axis toward the direction in which the inertia tensor is expressed easier.

将其 x 轴定向为使惯性张量表达更简便的方向。

We may also need a coordinate system located on the link centroid. In such a case, we define the link centroid coordinate i parallel to the coordinate located on joint i.

我们可能还需要一个位于“连杆质心”上的坐标系。在这种情况下，我们定义连杆质心坐标系 i 与位于关节 i 上的坐标系平行。

C. *Direction Cosine and Coordinate Transformation Matrix* 方向余弦与坐标变换矩阵

The direction cosine matrices $\boldsymbol{C}_i$ are commonly used to represent the attitude or orientation of body i in the inertial frame in the field of aerospace engineering. On the other hand, the coordinate transformation matrices with the notation of ${}^I\boldsymbol{A}_i$ are commonly used in the field of robotics. These two are eventually the same:

在航空航天工程领域，方向余弦矩阵 $\boldsymbol{C}_i$ 通常用于表示物体 i 在惯性系中的姿态或方向。另一方面，符号为 ${}^I\boldsymbol{A}_i$ 的坐标变换矩阵常用于机器人领域。这两者最终是相同的：

$$\boldsymbol{C}_i^T = {}^I\boldsymbol{A}_i$$

Since we define the link coordinate system as above, we generally need three axis rotations to coincide from $\{\Sigma_{i-1}\}$ to $\{\Sigma_i\}$. Let $C_1(\alpha_i), C_2(\beta_i), C_3(\gamma_i)$ be coordinate transformation (direction cosine) around each principle axis and $C_3(q_i)$ represents the coordinate transformation by angle q_i around joint i, then we obtain the following relationship (see Figure 3.3):

由于我们如上定义了连杆坐标系，通常需要绕三个轴旋转才能使 $\{\Sigma_{i-1}\}$ 与 $\{\Sigma_i\}$ 重合。设 $C_1(\alpha_i)$、$C_2(\beta_i)$、$C_3(\gamma_i)$ 为绕每个主轴的坐标变换（方向余弦），$C_3(g_i)$ 表示绕关节 i 旋转角度 q_i 的坐标变换，那么我们得到以下关系（见图 3.3）：

$$\begin{aligned}\{\Sigma_i\} &= {}^i\boldsymbol{C}_{i-1}\{\Sigma_{i-1}\} \\ &= [C_3(q_i)\,C_3(\gamma_i)\,C_2(\beta_i)\,C_1(\alpha_i)]^T\{\Sigma_{i-1}\}\end{aligned} \tag{1}$$

where $C_3(\gamma_i)$ and $C_3(q_i)$ seem duplicated, but γ_i corresponds to an offset angle and should be separated from a net rotation angle q_i.

其中 $C_3(\gamma_i)$ 和 $C_3(q_i)$ 看似重复，但 γ_i 对应一个偏置角，应与净旋转角 q_i 区分开来。

Note that the RPY representation of the attitude of link 0 is:

请注意，连杆 0 姿态的 RPY 表示为：

$$\begin{aligned}\{\Sigma_0\} &= C_0\{\Sigma_I\} \\ &= {}^0\boldsymbol{A}_I\{\Sigma_I\} \\ &= C_3(\gamma_0)\,C_2(\beta_0)\,C_1(\alpha_0)\{\Sigma_I\}\end{aligned} \tag{2}$$

where $\alpha_0, \beta_0, \gamma_0$ are Roll, Pitch, Yaw angles respectively.

其中 α_0、β_0、γ_0 分别为滚转角、俯仰角和偏航角。

The direction cosines are redundant way to represent attitude, but its advantage is that the relationship between attitude and angular velocity can be expressed by a simple equation, such that:

方向余弦是表示姿态的一种冗余方式，但其优点是姿态与角速度之间的关系可以用一个简单的方程来表示，即：

$$\dot{C}_0 = -\tilde{\boldsymbol{\omega}}_0 \boldsymbol{C}_0 \tag{3}$$

where $\dot{\boldsymbol{C}}_0$ is a time derivative of $\boldsymbol{C}_0$ and $\tilde{\boldsymbol{\omega}}$ is a skew-symmetric form of the angular velocity $\boldsymbol{\omega}_0$. This relationship is used for the routine of singularity-free integration from angular velocity to attitude.
其中 $\dot{\boldsymbol{C}}_0$ 是 $\boldsymbol{C}_0$ 的时间导数，$\tilde{\boldsymbol{\omega}}$ 是角速度 $\boldsymbol{\omega}_0$ 的反对称形式。这种关系用于从角速度到姿态的无奇点积分程序。

An additional note should be made for numerical operation in practice, the following expression which is known as the Rodorigues formula at infinitesimal rotation provides smaller error than 3:

在实际数值运算中还需额外说明，以下表达式（在无穷小旋转中称为罗德里格斯公式）比式 3的误差更小：

$$^{I}\boldsymbol{A}_0(t+\Delta t) = \left\{\boldsymbol{E} + \sin\theta_0 \tilde{\boldsymbol{\omega}}_0 + (1-\cos\theta_0)\,\tilde{\boldsymbol{\omega}}_0^T \tilde{\boldsymbol{\omega}}_0\right\}^{I} \boldsymbol{A}_0(t) \tag{4}$$

where

$$\theta_0 = |\tilde{\boldsymbol{\omega}}_0 \Delta t| \tag{5}$$

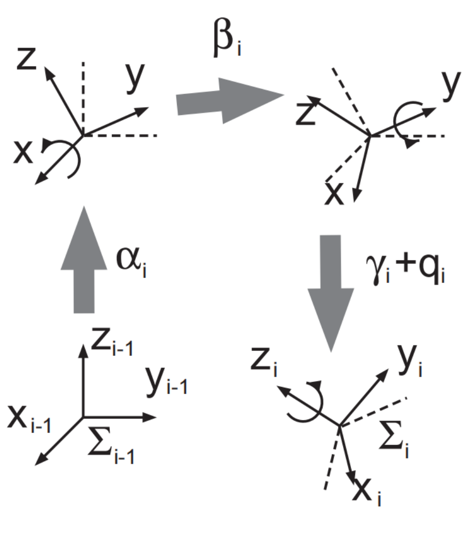


图 3. Coordinate Transformation

D. *Kinematics* 运动学

Link Vectors 连杆矢量

Link vectors for a link i are defined as follows (see Figure 4).

连杆 i 的连杆向量定义如下（见 4）

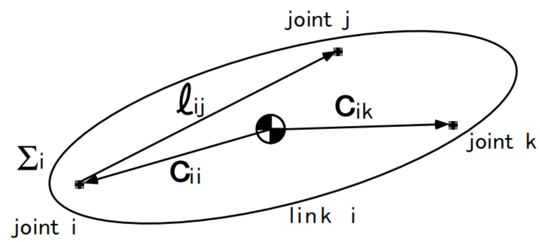


图 4. Position vectors

$\boldsymbol{c}_{ij}$: vector from the centroid of link i to joint j.
$\boldsymbol{c}_{ij}$ ： 从连杆 i 的质心到关节 j 的向量。
$\boldsymbol{\ell}_{ij}$: vector from joint i to joint j.
$\boldsymbol{\ell}_{ij}$ ： 从关节 i 到关节 j 的向量

$$\boldsymbol{\ell}_{ij} = \boldsymbol{c}_{ij} - \boldsymbol{c}_{ii} \tag{6}$$

$\boldsymbol{c}_{ie}$: vector from the centroid of link i to the end-point if link i is an end-link.
$\boldsymbol{c}_{ie}$ ： 若连杆 i 为末端连杆，则为从连杆 i 的质心到端点的向量。
$\boldsymbol{\ell}_{ie}$: vector from joint i to the end-point.
$\boldsymbol{\ell}_{ie}$ ： 从关节 i 到端点的向量。

$$\boldsymbol{\ell}_{ie} = \boldsymbol{c}_{ie} - \boldsymbol{c}_{ii} \tag{7}$$

Revolution Joint 旋转关节

For a successive set of links connected by a revolution joint, velocity $\boldsymbol{v}_i$ and angular velocity $\boldsymbol{\omega}_i$ are calculated recursively (see Figure 3.5). When $\boldsymbol{v}_0$ and $\boldsymbol{\omega}_0$ are given,

对于通过转动关节连接的一系列连续连杆，速度 $\boldsymbol{v}_i$ 和角速度 $\boldsymbol{\omega}_i$ 是递归计算的（见图 3.5）。当给定 $\boldsymbol{v}_0$ 和 $\boldsymbol{\omega}_0$ 时，

$$^{I}\boldsymbol{\omega}_i = {}^{I}\boldsymbol{\omega}_{B_i} + {}^{I}\boldsymbol{A}_i\, {}^{i}\boldsymbol{k}_i\, \dot{\phi}_i. \tag{8}$$

$$^{I}\boldsymbol{v}_i = {}^{I}\boldsymbol{v}_{B_i} + {}^{I}\boldsymbol{\omega}_{B_i} \times {}^{I}\boldsymbol{c}_{B_i i} - {}^{I}\boldsymbol{\omega}_i \times {}^{I}\boldsymbol{c}_{ii}. \tag{9}$$

And accelerations are calculated as the following.

加速度的计算方式如下。

$$
{}^I\dot{\boldsymbol{\omega}}_i = {}^I\dot{\boldsymbol{\omega}}_{B_i} + {}^I\boldsymbol{\omega}_i \times \left({}^I\boldsymbol{A}_i{}^i\boldsymbol{k}_i\dot{\phi}_i\right) + {}^I\boldsymbol{A}_i{}^i\boldsymbol{k}_i\ddot{\phi}_i \tag{10}
$$

$$
\begin{aligned}
{}^I\dot{\boldsymbol{v}}_i =& {}^I\dot{\boldsymbol{v}}_{B_i} + {}^I\dot{\boldsymbol{\omega}}_{B_i} \times {}^I\boldsymbol{c}_{B_i i} + {}^I\boldsymbol{\omega}_{B_i} \times \left({}^I\boldsymbol{\omega}_{B_i i} \times {}^I\boldsymbol{c}_{B_i i}\right) \\
& - {}^I\dot{\boldsymbol{\omega}}_i \times {}^I\boldsymbol{c}_{ii} - {}^I\boldsymbol{\omega}_i \times \left({}^I\boldsymbol{\omega}_i \times {}^I\boldsymbol{c}_{ii}\right)
\end{aligned} \tag{11}
$$

Prismatic Joint 移动关节

If a joint is prismatic, the kinematic relationship becomes as follows, for verocities:

如果关节移动关节，那么对于速度而言，运动学关系如下：

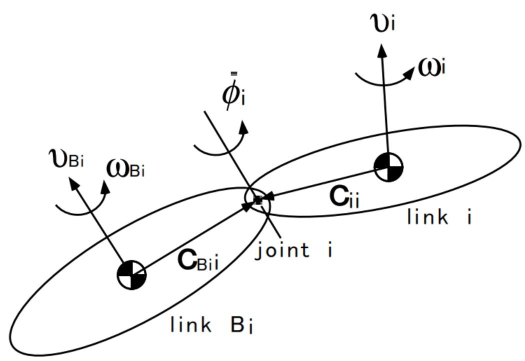


图 5. Kinematics

$$
{}^I\boldsymbol{\omega}_i = {}^I\boldsymbol{\omega}_{B_i} \tag{12}
$$

$$
\begin{aligned}
{}^I\boldsymbol{v}_i =& {}^I\boldsymbol{v}_{B_i} + {}^I\boldsymbol{\omega}_{B_i} \times {}^I\boldsymbol{c}_{B_i i} - {}^I\boldsymbol{\omega}_i \times {}^I\boldsymbol{c}_{ii} \\
& + {}^I\boldsymbol{\omega}_i \times \left({}^I\boldsymbol{A}_i{}^i\boldsymbol{k}_i\phi_i\right) + {}^I\boldsymbol{A}_i{}^i\boldsymbol{k}_i\dot{\phi}_i
\end{aligned} \tag{13}
$$

And for accelerations:

而对于加速度

$$
{}^I\dot{\boldsymbol{\omega}}_i = {}^I\dot{\boldsymbol{\omega}}_{B_i} \tag{14}
$$

$$
\begin{aligned}
{}^I\dot{\boldsymbol{v}}_i =& {}^I\dot{\boldsymbol{v}}_{B_i} + {}^I\dot{\boldsymbol{\omega}}_{B_i} \times \boldsymbol{c}_{B_i i} + {}^I\boldsymbol{\omega}_{B_i} \times \left({}^I\boldsymbol{\omega}_{B_i} \times \boldsymbol{c}_{B_i i}\right) \\
& - {}^I\dot{\boldsymbol{\omega}}_i \times \boldsymbol{c}_{ii} - {}^I\boldsymbol{\omega}_i \times \left({}^I\boldsymbol{\omega}_i \times \boldsymbol{c}_{ii}\right) + {}^I\dot{\boldsymbol{\omega}}_i \times \left({}^I\boldsymbol{A}_i{}^i\boldsymbol{k}_i\phi_i\right) \\
& + {}^I\boldsymbol{\omega}_i \times \left({}^I\boldsymbol{\omega}_i \times \left({}^I\boldsymbol{A}_i{}^i\boldsymbol{k}_i\phi_i\right)\right) + 2{}^I\boldsymbol{\omega}_i \times \left({}^I\boldsymbol{A}_i{}^i\boldsymbol{k}_i\phi_i\right) \\
& + {}^I\boldsymbol{A}_i{}^i\boldsymbol{k}_i\ddot{\phi}_i
\end{aligned} \tag{15}
$$

End-Point Kinematics 末端运动学

The kinematic relationship around the end-points is expressed as follows:

末端点周围的运动学关系表述如下：

$$
\dot{\boldsymbol{x}}_h = \boldsymbol{J}_m\dot{\boldsymbol{\phi}} + \boldsymbol{J}_b\dot{\boldsymbol{x}}_b \tag{16}
$$

$$
\ddot{\boldsymbol{x}}_h = \boldsymbol{J}_m\ddot{\boldsymbol{\phi}} + \dot{\boldsymbol{J}}_m\dot{\boldsymbol{\phi}} + \boldsymbol{J}_b\ddot{\boldsymbol{x}}_b + \dot{\boldsymbol{J}}_b\dot{\boldsymbol{x}}_b \tag{17}
$$

$\boldsymbol{x}_b \in R^6$: position/orientation of the base
$\boldsymbol{x}_h \in R^6$: position/orientation of the end-points
$\boldsymbol{\phi} \in R^n$: joint variables
$\boldsymbol{J}_b \in R^{6\times 6}$: Jacobian matrix for base variables
$\boldsymbol{J}_m \in R^{6\times n}$: Jacobian matrix for joint variables

E. *Equation of Motion* 运动方程

The equation of motion of the system is expressed in the following form:

系统的运动方程以如下形式表示：

$$
\begin{aligned}
\begin{bmatrix} \boldsymbol{H}_b & \boldsymbol{H}_{bm} \\ \boldsymbol{H}_{bm}^T & \boldsymbol{H}_m \end{bmatrix}
\begin{bmatrix} \ddot{\boldsymbol{x}}_b \\ \ddot{\phi} \end{bmatrix} +
\begin{bmatrix} \boldsymbol{c}_b \\ \boldsymbol{c}_m \end{bmatrix} \\
= \begin{bmatrix} \mathcal{F}_b \\ \boldsymbol{\tau} \end{bmatrix} +
\begin{bmatrix} \boldsymbol{J}_b^T \\ \boldsymbol{J}_m^T \end{bmatrix} \mathcal{F}_h
\end{aligned} \tag{18}
$$

where

$$
\boldsymbol{H}_b \in R^{6\times 6} \equiv \begin{bmatrix} w\boldsymbol{E} & w\tilde{\boldsymbol{r}}_{0g}^T \\ w\tilde{\boldsymbol{r}}_{0g} & \boldsymbol{H}_\omega \end{bmatrix} \tag{19}
$$

$$
\boldsymbol{H}_{bm} \in R^{6\times n} \equiv \begin{bmatrix} \boldsymbol{J}_{Tw} \\ \boldsymbol{H}_{\omega\phi} \end{bmatrix} \tag{20}
$$

$$
\boldsymbol{H}_\omega \in R^{3\times 3} \equiv \sum_{i=1}^{n}\left(\boldsymbol{I}_i + m_i\tilde{\boldsymbol{r}}_{0i}^T\tilde{\boldsymbol{r}}_{0i}\right) + \boldsymbol{I}_0 \tag{21}
$$

$$
\boldsymbol{H}_{\omega\phi} \in R^{3\times n} \equiv \sum_{i=1}^{n}\left(\boldsymbol{I}_i\boldsymbol{J}_{Ri} + m_i\tilde{\boldsymbol{r}}_{0i}\boldsymbol{J}_{Ti}\right) \tag{22}
$$

$$
\boldsymbol{H}_m \in R^{n\times n} \equiv \sum_{i=1}^{n}\left(\boldsymbol{J}_{Ri}^T\boldsymbol{I}_i\boldsymbol{J}_{Ri} + m_i\boldsymbol{J}_{Ti}^T\boldsymbol{J}_{Ti}\right) \tag{23}
$$

$$
\boldsymbol{J}_{Tw} \in R^{3\times n} \equiv \sum_{i=1}^{n} m_i\boldsymbol{J}_{Ti}/w \tag{24}
$$

$$\boldsymbol{J}_{Ti} \in R^{3\times n} \equiv [\boldsymbol{k}_1 \times (\boldsymbol{r}_i - \boldsymbol{p}_1), \boldsymbol{k}_2 \times (\boldsymbol{r}_i - \boldsymbol{p}_2), \ldots, \ldots, \boldsymbol{k}_i \times (\boldsymbol{r}_i - \boldsymbol{p}_i), \mathbf{o}, \ldots, \mathbf{o}] \tag{25}$$

$$\boldsymbol{J}_{Ri} \in R^{3\times n} \equiv [\boldsymbol{k}_1, \boldsymbol{k}_2, \ldots, \boldsymbol{k}_i, \mathbf{o}, \ldots, \mathbf{o}] \tag{26}$$

$$\boldsymbol{r}_{0g} \in R^3 \equiv \boldsymbol{r}_g - \boldsymbol{r}_0 \tag{27}$$

$$\boldsymbol{r}_{0i} \in R^3 \equiv \boldsymbol{r}_i - \boldsymbol{r}_0 \tag{28}$$

m_i : mass of link i of arm k
w : total mass of the system ($w = \sum_{i=1}^{n} m_i$)
$\boldsymbol{r}_i$: position vector of centroid of link i
$\boldsymbol{p}_i$: position vector of joint i
$\boldsymbol{k}_i$: unit vector indicating joint axis direction of link i
$\boldsymbol{r}_0$: position vector of centroid of satellite base body
$\boldsymbol{r}_g$: position vector of a total centroid of the system
$\boldsymbol{c}_b, \boldsymbol{c}_m$: velocity dependent non-linear terms
$\mathcal{F}_b$: external force/moment on the base
$\boldsymbol{\tau}$: joint torque of the arm
$\mathcal{F}_h$: external force/moment on the hand
$\boldsymbol{E}$: 3×3 identity matrix

and a tilde operator stands for a cross product such that $\tilde{\boldsymbol{r}}\boldsymbol{a} \equiv \boldsymbol{r} \times \boldsymbol{a}$. All position and velocity vectors are defined with respect to the inertial reference frame.
并且波浪号运算符表示叉积，使得 $\tilde{\boldsymbol{r}}\boldsymbol{a} \equiv \boldsymbol{r} \times \boldsymbol{a}$。所有位置向量和速度向量都是相对于惯性参考系定义的。

F. *Forward Dynamics: Simulation Procedure* 正向动力学：模拟过程

The procedure to compute a forward dynamics solutions are summarized as follows:

计算正向动力学解的过程总结如下：

1. At time t, compute link positions and velocities, recursively from link 0 to n.

在时刻 t，从连杆 0 到连杆 n 递归地计算连杆位置和速度。

2. Compute the inertia matrices using equations (17)-(26).

使用式 (17)-(26) 计算惯性矩阵。

3. Set accelerations $\ddot{\boldsymbol{x}}_b$ and $\ddot{\boldsymbol{\phi}}$ zero, and external forces $\mathcal{F}_b$ and $\mathcal{F}_h$ zero, then compute the inertial forces recursively from link n to 0. The resultant forces on the coordinates $\boldsymbol{x}_b$ and $\boldsymbol{\phi}$ are equal to the non-liner forces $\boldsymbol{c}_b$ and $\boldsymbol{c}_m$, respectively.

将加速度 $\ddot{\boldsymbol{x}}_b$ 和 $\ddot{\boldsymbol{\phi}}$ 设为零，外力 $\mathcal{F}_b$ 和 $\mathcal{F}_h$ 设为零，然后从连杆 n 到 0 递归计算惯性力。坐标 $\boldsymbol{x}_b$ 和 $\boldsymbol{\phi}$ 上的合力分别等于非线性力 $\boldsymbol{c}_b$ 和 $\boldsymbol{c}_m$。

4. Determine joint control forces $\boldsymbol{\tau}$ and thruster forces on the base $\mathcal{F}_b$ from a control law.

根据控制律确定关节控制力 $\boldsymbol{\tau}$ 和基座上的推进器力 $\mathcal{F}_b$。

5. Compute the accelerations by:

通过以下方式计算加速度：

$$\begin{bmatrix} \ddot{\boldsymbol{x}}_b \\ \ddot{\phi} \end{bmatrix} = \begin{bmatrix} \boldsymbol{H}_b & \boldsymbol{H}_{bm} \\ \boldsymbol{H}_{bm}^T & \boldsymbol{H}_m \end{bmatrix}^{-1} \left\{ \begin{bmatrix} \mathcal{F}_b \\ \boldsymbol{\tau} \end{bmatrix} + \begin{bmatrix} \boldsymbol{J}_b^T \\ \boldsymbol{J}_m^T \end{bmatrix} \mathcal{F}_h - \begin{bmatrix} \boldsymbol{c}_b \\ \boldsymbol{c}_m \end{bmatrix} \right\} \tag{29}$$

6. Integrate the above accelerations to yield the velocities and positions at time $t + \Delta t$.

对上述加速度进行积分，以得出在时间 $t + \Delta t$ 时的速度和位置。

7. go to 1. and continue.

转到步骤 1 并继续。

G. *Inverse Dynamics* 逆向动力学

Inverse dynamic computation is useful for a computed torque control. It is also needed for the forward dynamics in numerical computation the velocity dependent non-linear terms as described in the last section.

逆动力学计算对于计算力矩控制很有用。在数值计算中，如在上一节所述，对于速度相关的非线性项的正向动力学计算，它也是必需的。

For the inverse dynamic computation, an order-n, recursive Newton-Euler approach is well-known.

对于逆动力学计算，一种 n 阶递归牛顿-欧拉方法是众所周知的。

Newton and Euler equations for a link i are expressed as:

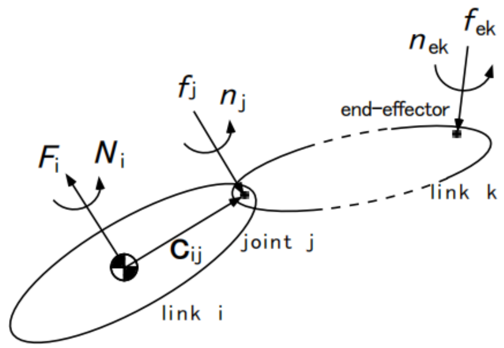


图 6. Dynamic equilibrium

连杆 i 的牛顿方程和欧拉方程表示为：

$$\boldsymbol{F}_i = m_i \dot{\boldsymbol{v}}_i \tag{30}$$

$$\boldsymbol{N}_i = \boldsymbol{I}_i \dot{\boldsymbol{\omega}}_i + \boldsymbol{\omega}_i \times (\boldsymbol{I}_i \boldsymbol{\omega}_i) \tag{31}$$

where $\boldsymbol{F}_i, \boldsymbol{N}_i$ are inertial force and moment exert on the link centroid. Together with the following force and moment exerting on the joint or end-point,
其中 $\boldsymbol{F}_i$、$\boldsymbol{N}_i$ 分别为作用在连杆质心上的惯性力和惯性力矩。再结合作用在关节或端点上的下列力和力矩，
$\boldsymbol{f}_i, \boldsymbol{n}_i$: Force and moment on joint i.
$\boldsymbol{f}_{ei}, \boldsymbol{n}_{ei}$: Force and momnet on end-point (if link i is an end-link)

the dynamic equilibrium expressed in the following form (see Figure 3.6):

以下列形式表示的动态平衡（见图 3.6）：

$$\boldsymbol{f}_i = \boldsymbol{F}_i + \sum_{j=i+1}^{n} \boldsymbol{S}_{ij} \boldsymbol{f}_j + \boldsymbol{S}_{ei} \boldsymbol{f}_{ei} \tag{32}$$

$$\begin{aligned} \boldsymbol{n}_i =& \boldsymbol{N}_i + \sum_{j=i+1}^{n} \boldsymbol{S}_{ij} \left(\boldsymbol{\ell}_{ij} \times \boldsymbol{f}_j + \boldsymbol{n}_j \right) \\ &+ \boldsymbol{S}_{ii} \boldsymbol{c}_{ii} \times \boldsymbol{F}_i + \boldsymbol{S}_{ei} \left(\boldsymbol{\ell}_{ie} \times \boldsymbol{f}_{ei} + \boldsymbol{n}_{ei} \right) \end{aligned} \tag{33}$$

for around a revolution joint, and
围绕一个旋转关节，并且

$$\boldsymbol{f}_i = \boldsymbol{F}_i + \sum_{j=i+1}^{n} \boldsymbol{S}_{ij} \boldsymbol{f}_j + \boldsymbol{S}_{ei} \boldsymbol{f}_{ei} \tag{34}$$

$$\begin{aligned} \boldsymbol{n}_i =& \boldsymbol{N}_i + \sum_{j=i+1}^{n} \boldsymbol{S}_{ij} \left(\boldsymbol{\ell}_{ij} \times \boldsymbol{f}_j + \boldsymbol{n}_j \right) \\ &+ \boldsymbol{S}_{ii} \left(\boldsymbol{c}_{ii} - \phi_i \right) \times \boldsymbol{F}_i + \boldsymbol{S}_{ei} \left(\boldsymbol{\ell}_{ie} \times \boldsymbol{f}_{ei} + \boldsymbol{n}_{ei} \right) \end{aligned} \tag{35}$$

for around a prismatic joint.
围绕一个移动关节。

After the computation of whole $\boldsymbol{f}_i$ and $\boldsymbol{n}_i$ for $i = 1$ to n, we can obtain joint torque as:

在计算完从 $i = 1$ 到 n 的所有 $\boldsymbol{f}_i$ 和 $\boldsymbol{n}_i$ 后，我们可以得到关节扭矩如下：

if revolution joint:

$$\boldsymbol{\tau}_i = \boldsymbol{n}_i^{TI} \boldsymbol{k}_i \tag{36}$$

if prismatic joint:

$$\boldsymbol{\tau}_i = \boldsymbol{f}_i^{TI} \boldsymbol{k}_i \tag{37}$$

And the reaction force/moment on the base centroid is obtained as follows:

而底座形心处的反力/反力矩按如下方式获得：

$$\boldsymbol{F}_0 = \sum_{i=1}^{n} \boldsymbol{S}_{0i} \boldsymbol{f}_i \tag{38}$$

$$\boldsymbol{N}_0 = \sum_{i=1}^{n} \boldsymbol{S}_{0i} \left(\boldsymbol{c}_{0i} \times \boldsymbol{f}_i + \boldsymbol{n}_i \right) \tag{39}$$

H. Application Examples

Here, some of applications for dynamic simulation of moving-base sytems are illustrated, which all are relevant to actual space flight missions.

在此，我们举例说明了一些适用于移动基座系统动态仿真的应用，这些应用都与实际的太空飞行任务相关。

Figure 7 (a) depicts a simulation model of ETS-VII, a Japanese free-flying space robot with 2 meter long 6 DOF manipulator arm. The satellite was launched November, 1997. It is currently flying in orbit, as of August 1998, and a number of significant experiments on space robotics are conducting on the satellite. Free-flying system dynamics including manipulator reaction and the vibrations of solar paddles can be analyzed with the SpaceDyn toolbox.

图 7（a）展示了 ETS-VII 的仿真模型，这是一款日本的自由飞行太空机器人，配备有一个 2 米长的 6 自由度机械臂。该卫星于 1997 年 11 月发射。截至 1998 年 8 月，它仍在轨道上运行，并且正在该卫星上开展多项重要的太空机器人实验。借助 SpaceDyn 工具箱，可

以对包括机械臂反作用力和太阳帆板振动在内的自由飞行系统动力学进行分析。

Figure 7 (b) depicts a flexible-base robot. Practical examples of such a system are SRMS-SPDM system, a Canadian made space station manipulator system and JEMRMS, a macro-mini manipulator system for the Japanese Experimental Module of the station. For these systems the internal dynamics, as presented in the following section, will be a key technology in terms of the reaction and vibration management.

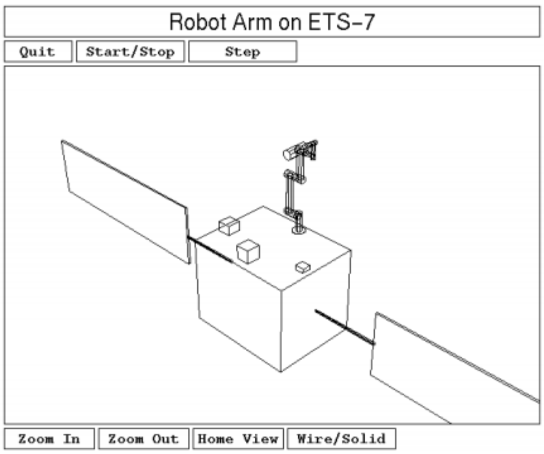
图 7（b）展示了一个柔性基座机器人。此类系统的实际例子包括加拿大制造的空间站机械臂系统 SRMS-SPDM 以及日本实验舱的宏微机械臂系统 JEMRMS。对于这些系统而言，下一节将要介绍的内部动力学，在反作用力和振动管理方面将是一项关键技术。

Both figures 7 (a) and (b) are illustrated using a useful animation tool named "XAnimate."

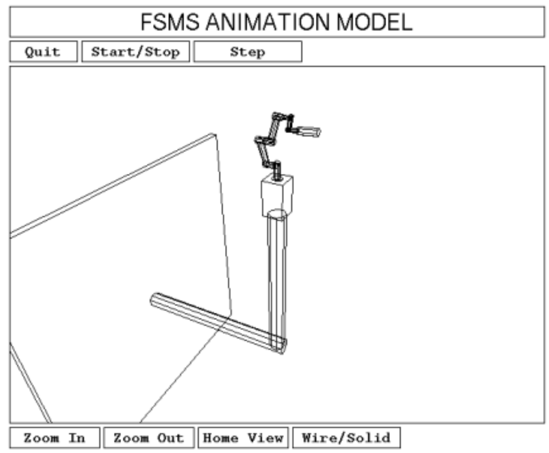
图 7（a）和（b）均是使用一款名为"XAnimate"的实用动画工具绘制的。

Figure 7 (c) is a touch-down simulation of MUSES-C asteroid sample-return spacecraft. For this simulation, impulsive ground contact is a key issue and the contact model discussed in the previous subsection is applied. With the development of the contact model for tire mechanics, the dynamic motion of off-road articulated vehicle can be also simulated, as shown in Figure 7 (d), such an application is found in a mission of a planetary exploration rover.

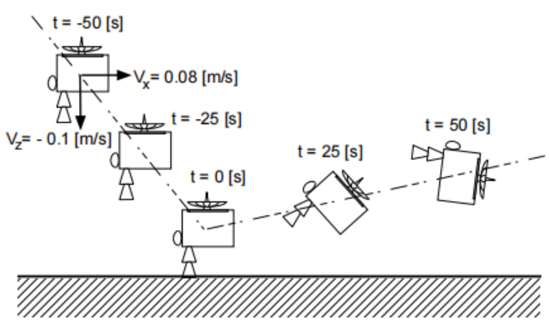
图 7（c）是 MUSES-C 小行星样本返回航天器的着陆仿真。在这项仿真中，地面冲击接触是一个关键问题，并且应用了上一小节中讨论的接触模型。随着轮胎力学接触模型的发展，还能够对越野铰接车辆的动态运动进行仿真，如图 7（d）所示，这种应用可见于行星探测漫游车的任务中。



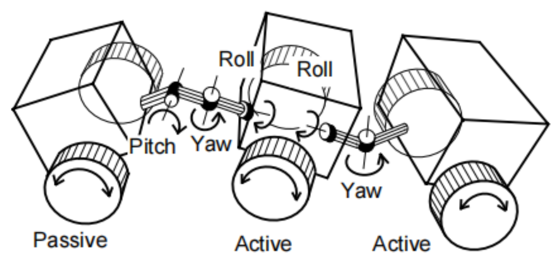
(a) A simulation model for Free-Flying Space Robot



(b) A simulation model for Flexible-Base Robot



(c) Touch-down simulation of MUSES-C Asteroid Sample-Return Spacecraft



(d) An example of an articulated off-road vehicle as a potential design of a planetary rover

图 7. Practical applications of the dynamics simulation of moving-base robots by "SpaceDyn" toolbox