RiSE Compliance Notes

Jonathan K. Karpick

22 August 2004

RiSE platform analysis is conducted with a combination of four methods:

References:
Salisbury & Mason, *Robot Hands and the Mechanics of Manipulation*
Cutkosky & Kao, *Computing and Controlling the Compliance of a Robotic Hand*
John J. Craig, *Introduction to Robotics*

1 Stance Configuration

Stance configuration is determined via MATLAB scripts which define the robot configuration and location with respect to a ground plane. The ground plane can be oriented anywhere with respect to a set of world coordinates.

Given this initial setup, MATLAB plots the set of possible foot locations (Figure 1). The user then places each foot via a graphical user interface (Figure 2).

The MATLAB script then calculates the joint angles required for the selected foot placement and re-draws the robot as shown in Figure 3.

The script then calculates necessary transforms and jacobians.

Run MATLAB script: `REAL_RiSE_Comp_Script.m`, which uses:

```
getTransforms.m
drawGroundPlane.m
robotVertices.m
```

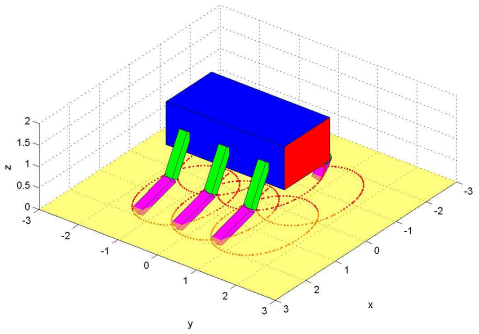

Figure 1: Foot placement possibilities based on robot CG location and robot orientation

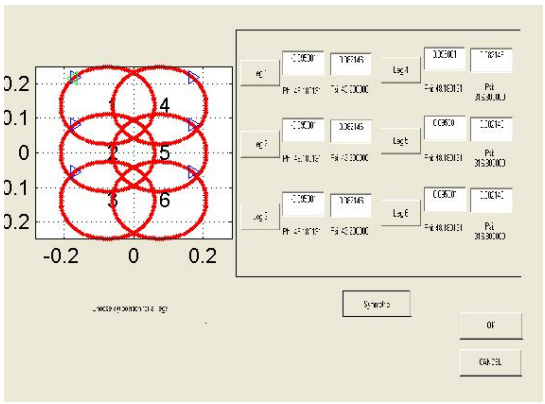

Figure 2: User selects placement of each foot

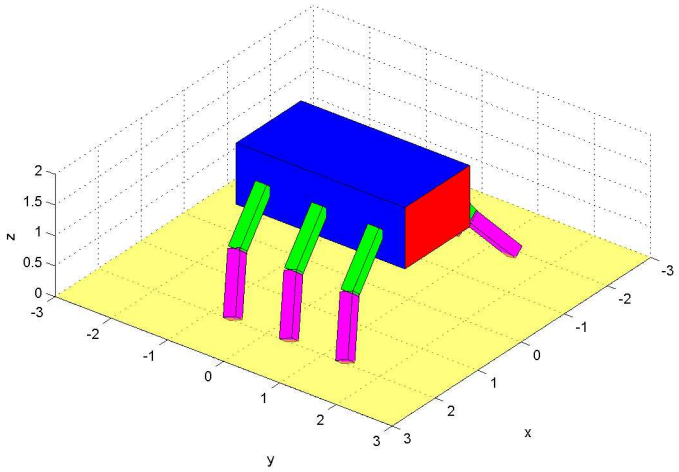

Figure 3: Resultant configuration after user selects placement of each foot (symmetric in this case)

drawRobot.m
footWorkspace.m
drawFootWorkspace.m
foot_placement_gui.m
findJacobians.m

2 Loop Equations

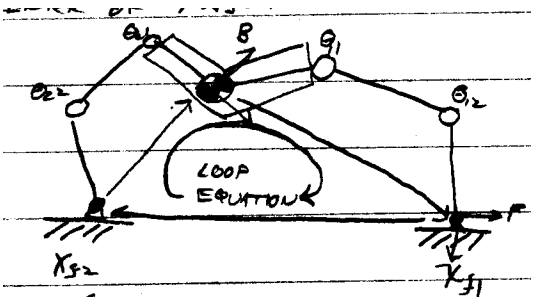

Figure 4: Loop Equation Diagram

Figure 4 shows a two-legged representation of this problem (don't assume that it's planar...). All the equations (Three derivative equations per loop) are assembled into single matrix equation. The null space of this matrix represents joint movements that can be conducted without violating the loop equations. Taking these joint velocities and multiplying by the Jacobian yields the resulting velocity of the robot's CG.

The MATLAB script `REAL_RiSE_Comp_Script.m` computes the loop equations and gets the null space.

3 Compliance Analysis

Compliance analysis computes the compliance of each leg given the robot geometry, joint compliance, and ankle compliance for each leg.

Compliance/Stiffness Procedure Thus Far:

1. Get description of each foot position in the ground coordinates (from robot cg, robot configuration, and joint angles).
2. Get transformation matrix from each lower leg to the ground frame (4x4) (by going from world to robot CG to foot)
3. Determine vector from foot to robot cg in ground coordinates
4. Form velocity transformation matrices, ${}^{G}_{B}\mathbf{T}_{v}$ and ${}^{G}_{F}\mathbf{T}_{v}$, where G is ground frame, B is body frame, and F is foot (lower leg) frame.
5. Convert **C** (which is either purely symbolic or some function of joint and structural stiffnesses) into ground coordinates via equation (20)
6. Apply $\mathbf{H}\,\mathbf{H}^{\mathrm{T}}$ transform to remove compliances which cannot be applied by the assumed contact, reducing size of effective compliance matrix.

7. Invert to get stiffness matrix in ground coordinates.
8. Apply $\mathbf{H}^{\mathrm{T}}\,\mathbf{H}$ transform to re-constitute the stiffness matrix to 6x6.
9. Convert each leg's stiffness matrix to body coordinates.
10. Sum them all up to get the overall stiffness matrix
11. Examine.

Definitions for velocity transform and the cross-product matrix:

$$ {}^{B}_{A}\mathbf{T}_{v} = \begin{bmatrix} {}^{B}_{A}\mathbf{R} & {}^{B}\mathbf{P}_{B\to A}\times {}^{B}_{A}\mathbf{R} \\ \mathbf{0} & {}^{B}_{A}\mathbf{R} \end{bmatrix} \tag{1} $$

$$ \mathbf{P}\times = \begin{bmatrix} 0 & -P_z & P_y \\ P_z & 0 & -P_x \\ -P_y & P_x & 0 \end{bmatrix} \tag{2} $$

Conversion of joint compliance matrix from joint space to body frame:

$$ {}^{B}\delta\mathbf{x} = -\mathbf{J}_{\theta}\,\delta\theta \tag{3} $$

$$ \tau = -\mathbf{J}_{\theta}^{\mathrm{T}}\,{}^{B}\mathbf{F} \tag{4} $$

$$ \delta\theta = \mathbf{C}_{\theta}\,\tau \tag{5} $$

$$ {}^{B}\delta\mathbf{x} = \mathbf{J}_{\theta}\,\mathbf{C}_{\theta}\,\mathbf{J}_{\theta}^{\mathrm{T}}\,{}^{B}\mathbf{F} \tag{6} $$

$$ \Rightarrow {}^{B}\mathbf{C}_{\theta} = \mathbf{J}_{\theta}\,\mathbf{C}_{\theta}\,\mathbf{J}_{\theta}^{\mathrm{T}} \tag{7} $$

Note that equations (3) and (4) make assumptions about constraints (and resulting motion of the body) via the contacts - need to revisit this later.

Conversion of stiffness matrices:

$$ {}^{G}\mathbf{F} = {}^{G}\mathbf{K}\,\delta\mathbf{x}_{g} \tag{8} $$

$$ {}^{B}\mathbf{F} = {}^{B}\mathbf{K}\,\delta\mathbf{x}_{b} \tag{9} $$

$$ \delta\mathbf{x}_{g} = {}^{G}_{B}\mathbf{T}_{v}\,\delta\mathbf{x}_{b} \tag{10} $$

$$ {}^{B}\mathbf{F} = {}^{G}_{B}\mathbf{T}_{v}^{\mathrm{T}}\,{}^{G}\mathbf{F} \tag{11} $$

$$ {}^{G}\mathbf{F} = {}^{G}\mathbf{K}\,{}^{G}_{B}\mathbf{T}_{v}\,\delta\mathbf{x}_{b} \tag{12} $$

$$ {}^{B}\mathbf{F} = {}^{G}_{B}\mathbf{T}_{v}^{\mathrm{T}}\,{}^{G}\mathbf{K}\,{}^{G}_{B}\mathbf{T}_{v}\,\delta\mathbf{x}_{b} \tag{13} $$

$$ \Rightarrow {}^{B}\mathbf{K} = {}^{G}_{B}\mathbf{T}_{v}^{\mathrm{T}}\,{}^{G}\mathbf{K}\,{}^{G}_{B}\mathbf{T}_{v} \tag{14} $$

Conversion of compliance matrices:

$$
\begin{aligned}
{}^{F}\delta\mathbf{x} &= {}^{F}\mathbf{C}\,{}^{F}\mathbf{F} && (15)\\
{}^{G}\delta\mathbf{x} &= {}^{G}_{F}\mathbf{T}_{v}\,{}^{F}\delta\mathbf{x} && (16)\\
{}^{G}\delta\mathbf{x} &= {}^{G}_{F}\mathbf{T}_{v}\,{}^{F}\mathbf{C}\,{}^{F}\mathbf{F} && (17)\\
{}^{F}\mathbf{F} &= {}^{G}_{F}\mathbf{T}_{v}^{\mathrm{T}}\,{}^{G}\mathbf{F} && (18)\\
{}^{G}\delta\mathbf{x} &= {}^{G}_{F}\mathbf{T}_{v}\,{}^{F}\mathbf{C}\,{}^{G}_{F}\mathbf{T}_{v}^{\mathrm{T}}\,{}^{G}\mathbf{F} && (19)\\
\Rightarrow {}^{G}\mathbf{C} &= {}^{G}_{F}\mathbf{T}_{v}\,{}^{F}\mathbf{C}\,{}^{G}_{F}\mathbf{T}_{v}^{\mathrm{T}} && (20)
\end{aligned}
$$

This method was applied to the simplified RiSE platform with the recent Stanford feet and their subsequent modified (zip tied) version. Plots shown in Figure 5.

MATLAB Files:

```
footCompliance.m
findComplianceOnly.m
robotWithEllipsesComplianceOnly.m
robotWithEllipses.m
drawGroundPlane.m
drawRobot.m
surfrotate.m
plotSurf.m
```

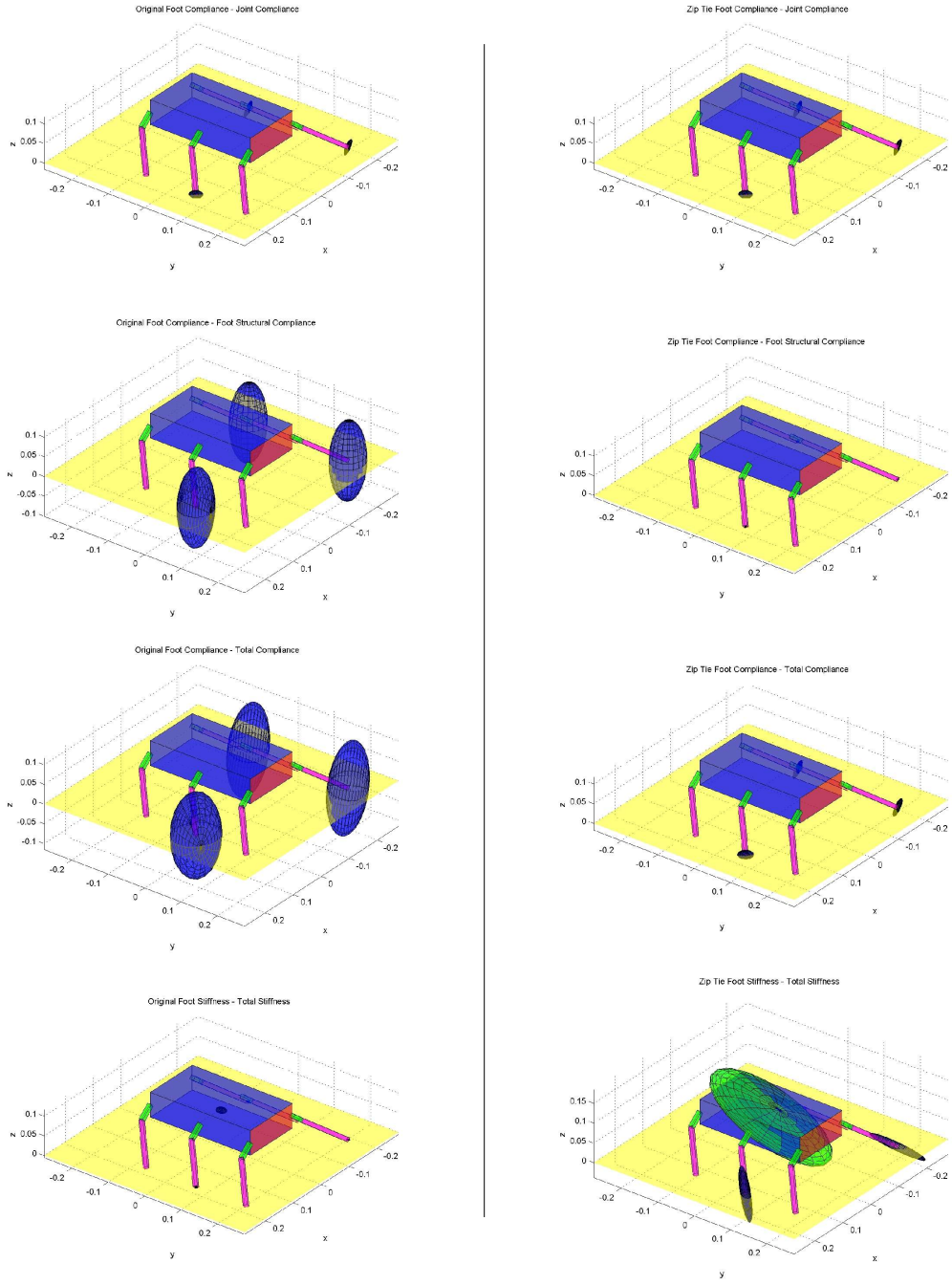

Figure 5: Comparison of Original and Zip Tie Feet Compliance and Stiffness

4 Grip Transform

Ref: Salisbury & Mason

$$\mathbf{F}_{\text{res}} = [f_x, f_y, f_z, m_x, m_y, m_z, \lambda_1, \lambda_2, \cdots, \lambda_{n-6}]^{\mathrm{T}}$$
$$\mathbf{F}_{\text{contact}} = [c_1, c_2, c_3, \cdots, c_n]^{\mathrm{T}}$$
$$\mathbf{F}_{\text{res}} = \mathbf{G}^{-\mathrm{T}}\mathbf{F}_{\text{contact}}$$
$$\mathbf{F}_{\text{contact}} = \mathbf{G}^{\mathrm{T}}\mathbf{F}_{\text{res}}$$

You've taken the **H** matrix for each foot contact point, translated it to a common reference system, and augmented it with internal wrenches to get the above equations.

Also gives you:

$$\mathbf{V}_{\text{res}} = [v_x, v_y, v_z, \omega_x, \omega_y, \omega_z, \gamma_1, \cdots, \gamma_{n-6}]^{\mathrm{T}}$$
$$\mathbf{V}_{\text{contact}} = [d_1, \cdots, d_n]^{\mathrm{T}}$$
$$\mathbf{V}_{\text{contact}} = \mathbf{G}^{-1}\mathbf{V}_{\text{res}}$$
$$\mathbf{V}_{\text{res}} = \mathbf{G}\mathbf{V}_{\text{contact}}$$

MATLAB Scripts used:

`GripTransform.m`