27   Walking Robot Constraints

The "Big Pig" project, which involves a pig-like quadruped robot, wants to understand how to control the device when it operates on a slippery surface. Viewed from above, each foot is at a location described by the vector $\vec{r}_i$, comprising the x (positive forward) and y (positive to the left) coordinates relative to the center of mass; the vertical direction is z. Clearly static stability is possible only if the centroid is contained within one of the triangles created by the planted foot locations. Hence, slow walking can be achieved by moving the centroid into different triangle sets, and moving the feet in turn. Also in every such configuration, the roll and pitch moments have to balance to zero. Things get harder during actual locomotion, because we do not want slippage on any of the feet. This occurs when force in the horizontal plane exceeds a critical fraction of the vertical (normal) force: when $F_{i,x} \geq \mu F_{i,z}$, where μ is the Coulomb friction coefficient. A similar constraint would apply of course in the y-direction.

Here we take on a simplified, three-leg problem that lays out some very practical issues in large and slowly moving walking robots. Consider the statically stable condition
`r = [[1.1, 0.8];[1.2,-0.7];[-1.5,-0.7]]` meters, where each pair gives the x and y coordinates of three planted legs (forward left, forward right, back right). The weight of the Big Pig is 4600 Newtons, and the value of μ is 0.12. Along with the non-slip constraint for each foot in the x-direction, we have a mechanical design constraint that the x force magnitude on any foot cannot exceed 250 Newtons. Also, since a foot cannot "pull" on the ground, each $F_{i,z}$ has to be positive. This is a total of nine inequality constraints.

There are also three equality constraints. First, the sum of the vertical forces has to equal the vehicle weight. Second, the sum of moments due to the vertical forces in the pitch direction (nose-up or nose-down) has to be equal to zero. Third, due to the construction of the robot, moments caused by the actuators that make x-direction forces have to equalize front and back: $F_{fl,x} + F_{fr,x} - F_{br,x} = 0$.

Note that forces in the y-direction do not play any role in our problem.

<u>The Question</u>: What is the maximum forward force ($\sum_{i=1}^{3} F_{i,x}$) that can be exerted on the chassis without slipping? Please give the the three F_z's and the three F_x's that go with your answer.

There are three equalities and nine inequalities, for six unknowns. We write:

$$
\begin{aligned}
F_{1,z} + F_{2,z} + F_{3,z} &= 4600 \text{ (sum of vertical forces supports the weight)} \\
r_{1,x}F_{1,z} + r_{2,x}F_{2,z} + r_{3,x}F_{3,z} &= 0 \text{ (sum of pitch moments due to vertical forces is zero)} \\
F_{1,x} + F_{2,x} - F_{3,x} &= 0 \text{ (actuator moment constraint)} \\
-F_{1,z} &\leq 0 \text{ (vertical forces positive only for each foot)} \\
-F_{2,z} &\leq 0 \\
-F_{3,z} &\leq 0 \\
-\mu F_{1,z} + F_{1,x} &\leq 0 \text{ (no-slip condition for each foot)}
\end{aligned}
$$

$$
\begin{aligned}
-\mu F_{2,z} + F_{2,x} &\leq 0 \\
-\mu F_{3,z} + F_{3,x} &\leq 0 \\
F_{1,x} &\leq 250 \text{ (absolute horizontal load limit for each foot)} \\
F_{2,x} &\leq 250 \\
F_{3,x} &\leq 250.
\end{aligned}
$$

Since the solution lies on a vertex in this linear problem, we have to consider all possible combinations of the three equality constraints, and three of the nine inequalities. We solve a 6x6 $Ax = b$ problem for each one, and the number of combinations is $9!/(9-3)!3! = 84$. Solutions are made infeasible by either of the following problems: 1) the 6x6 linear problem cannot be solved because the A-matrix is singular, or 2) the solution violates one of the other six inequalities. I get six feasible solutions:

```
#1 Feasible Solution:
    s[    0.00 2555.56 2044.44 |   0.00 245.33 245.33 ] -> 490.67 N
#2 Feasible Solution:
    s[    0.00 2555.56 2044.44 |  -4.67 250.00 245.33 ] -> 490.67 N
#3 Feasible Solution:
    s[ 2653.85   -0.00 1946.15 | 233.54   0.00 233.54 ] -> 467.08 N
#4 Feasible Solution:
    s[ 2653.85   -0.00 1946.15 | 250.00 -16.46 233.54 ] -> 467.08 N
#5 Feasible Solution:
    s[ 2083.33  549.38 1967.28 | 250.00 -13.93 236.07 ] -> 472.15 N
#6 Feasible Solution:
    s[  490.38 2083.33 2026.28 |  -6.85 250.00 243.15 ] -> 486.31 N
```

and the best of them is 490.67 N. Note the variety of feasible solutions and how close they are to one another!

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Solve the Big Pig three-leg, no-slip problem.

% FSH MIT 2.017 March 2008
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

clear all;

% radius vectors for the feet, meters [x y]
r = [[1.1, 0.8];[1.2,-0.7];[-1.5,-0.7]] ;

W = 4600 ; % weight of the chassis
mu = .12 ; % Coulomb friction coefficient
```

```
maxXForce = 250 ; % maximum x-force allowed on any leg

% plot the layout of the feet
figure(1);clf;hold off;
plot(r(:,1),r(:,2),'ro',0,0,'bs','LineWidth',2);
axis('equal');
grid;

% Unknowns are the three Fz's, and the three Fx's;
% let the vector of unknowns be x = [Fz1 Fz2 Fz3 Fx1 Fx2 Fx3].
% We write everything in the form of Ax = b, where the top part
% has the equalities, and the lower part has the inequalities of the
% form Ax <= b

% three equality constraints first - the first two rows of A and b
A(1,1:3) = ones(1,3) ; % sum of Fz's equals the weight
b(1,1) = 4600 ;
A(2,1:3) = r(:,1)' ; % sum of pitch moments due to Fz's equals zero
b(2,1) = 0 ;
A(3,4:6) = [1 1 -1] ; % sum of moments due to feet actuators is zero
b(3,1) = 0 ;

% now nine inequality constraints - rows 4-12 of A and b
A(4:6,1:3) = -eye(3,3) ; % Fz's are positive only
b(4:6,1) = zeros(3,1) ;
A(7:9,1:3) = -mu*eye(3,3) ; % no slippage:  F_xi - mu*Fzi <= 0
A(7:9,4:6) = eye(3,3) ;
b(7:9,1) = zeros(3,1) ;
A(10:12,4:6) = eye(3,3) ; % upper limit on x-load for each foot
b(10:12,1) = maxXForce*ones(3,1) ;

% We have six unknowns - so at each vertex we have to solve six
% equations.  There are twelve equations total, of which we must
% use the first three (equalities).  We have nine inequalities,
% so there will be 9!/(9-3)!3! = 84 combinations of three
% inequalities to try out.  Re the lecture notes,
I = 9;
E = 3;
n = 6;

% Count out some index sets for the equations we will use and not use in
% each linear system we solve.  These are SPECIFIC to the problem as
% written, i.e., only three equations to use, out of nine possible
```

```
% each row indexes the set of equations to use
ct = 0 ;
for i = 4:10,
    for j = i+1:11,
        for k = j+1:12,
            ct = ct + 1 ;
            indUse(ct,:) = [i j k] ;
        end;
    end;
end;

% each row of indUnused is the set of equations to _not to use_
% (also problem SPECIFIC as written, to three used inequalities)
for i = 1:length(indUse),
    ct = 0 ;
    for j = E+1:I+E,
        if indUse(i,1) ~= j & indUse(i,2) ~= j & indUse(i,3) ~= j,
            ct = ct + 1 ;
            indUnused(i,ct) = j ;
        end;
    end;
end;

% make a plot just to make sure we have no mistakes! (the figure should
% be nothing but a straight line)
figure(2);clf;hold off ; hold on;
for i = 1:length(indUnused),
     plot(sort([indUnused(i,:),indUse(i,:)]));
end;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Here is the linear programming engine, in a general form

% now go through the various n-dimensional subproblems
Atemp(1:E,:) = A(1:E,:) ; % first E rows for all cases - equalities
btemp(1:E,1) = b(1:E,1) ;

for i = 1:length(indUse),
    % make up the other rows we need
    Atemp(E+1:n,:) = A(indUse(i,:),:) ;
    btemp(E+1:n,:) = b(indUse(i,:),1) ;

    % solve the system if Atemp can be inverted
    if rank(Atemp) == n,
```

```
        xtemp(:,i) = Atemp \ btemp ; % (see MATLAB help - solves A*x=b)

        % check against all the unused constraints
        for j = 1:I+E-n,
            if A(indUnused(i,j),:)*xtemp(:,i) > b(indUnused(i,j),:),
                xtemp(:,i) = NaN*ones(n,1); % mark bad ones with NaN
                break ;
            end;
        end;
    else, % if we can't invert Atemp, we cannot get a solution
        xtemp(:,i) = NaN*ones(n,1) ;
    end;
end;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% display results and optimum Big Pig x-load
ct = 0 ;
f = fopen('bigPigSolution.dat','w');
fprintf(f,'\n\\begin{verbXXXatim}\n');
for i = 1:length(xtemp),
    if ~isnan(xtemp(1,i)),
        ct = ct + 1 ;
        disp('-----------------------------------------------------------');
        disp(sprintf('#%d Feasible Solution [%g %g %g  |  %g %g %g]',...
            ct,xtemp(:,i)'));
        disp(sprintf('has x-load of %g N.',sum(xtemp(4:6,i)')));
        fprintf(f,'#%d Feasible Solution: \n',ct);
        fprintf(f,...
            '    s[ %7.2f %7.2f %7.2f | %6.2f %6.2f %6.2f ] -> %6.2f N\n',...
            xtemp(:,i)',sum(xtemp(4:6,i)'));
    end;
end;
fprintf(f,'\\end{verXXXbatim}\n');
fclose(f) ;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

2.017J Design of Electromechanical Robotic Systems
Fall 2009