

Controller Formalization

Mahdi Khoramshahi Nitish Thatte

August 6, 2013

Inverse Dynamics Controller

We wish to control a quadrupedal robot with the following per leg linear relationship between leg rotation speed and generated force

$$F_i = A_i + B_i^+ \omega_i^+ + B_i^- \omega_i^- \quad (1)$$

Where $i \in \{1, 2, 3, 4\}$ corresponds to one of the four legs and ω_i^+ and ω_i^- are the leg velocities for the upwards and downwards parts of the trajectories respectively. Our goal is to control the height, pitch, and roll of the robot. We can write the Newton-Euler equations for these degrees of freedom:

$$m\ddot{y} = \sum_{i=1}^4 (A_i + B_i^+ \omega_i^+ + B_i^- \omega_i^-) - mg \quad (2)$$

$$I_x \ddot{\theta} = l_x \left[\sum_{i=\{1,2\}} (A_i + B_i^+ \omega_i^+ + B_i^- \omega_i^-) - \sum_{i=\{3,4\}} (A_i + B_i^+ \omega_i^+ + B_i^- \omega_i^-) \right] \quad (3)$$

$$I_z \ddot{\phi} = l_z \left[\sum_{i=\{1,4\}} (A_i + B_i^+ \omega_i^+ + B_i^- \omega_i^-) - \sum_{i=\{2,3\}} (A_i + B_i^+ \omega_i^+ + B_i^- \omega_i^-) \right] \quad (4)$$

Where m is the mass of the robot, I_x and I_z are the moments of inertia about the x and z -axes respectively, and l_x and l_z are the distances to the center of mass in the x and z directions respectively. In matrix form these equations are

$$\begin{bmatrix} m & 0 & 0 \\ 0 & I_x & 0 \\ 0 & 0 & I_z \end{bmatrix} \begin{bmatrix} \ddot{y} \\ \ddot{\theta} \\ \ddot{\phi} \end{bmatrix} = \begin{bmatrix} \sum A_i - mg \\ \bar{L}_x^T \bar{A} \\ \bar{L}_z^T \bar{A} \end{bmatrix} + \begin{bmatrix} \bar{B}_+^T & \bar{B}_-^T \\ \bar{B}_+^T \hat{L}_x & \bar{B}_-^T \hat{L}_x \\ \bar{B}_+^T \hat{L}_z & \bar{B}_-^T \hat{L}_z \end{bmatrix} \begin{bmatrix} \bar{\omega}^+ \\ \bar{\omega}^- \end{bmatrix} \quad (5)$$

Where

$$\bar{L}_x = l_x \begin{bmatrix} 1 \\ 1 \\ -1 \\ -1 \end{bmatrix} \quad (6)$$

$$\hat{L}_x = l_x \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix} \quad (7)$$

$$\bar{L}_z = l_z \begin{bmatrix} 1 \\ -1 \\ -1 \\ 1 \end{bmatrix} \quad (8)$$

$$\hat{L}_z = l_z \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (9)$$

We can write equation 5 compactly as

$$M\ddot{\bar{x}} = A + B\bar{\omega} \quad (10)$$

Where $\bar{x} = [y, \theta, \phi]^T$ is the state of the robot and $\bar{\omega} = [\bar{\omega}^+, \bar{\omega}^-]^T$.

In order to achieve a desired state x_0 we employ virtual model control using a virtual spring. Therefore, our desired dynamics are

$$\ddot{\bar{x}} = K_p(\bar{x}_0 - \bar{x}) \quad (11)$$

We wish to find a $\bar{\omega}$ that will achieve the above dynamics so we will now solve for a suitable $\bar{\omega}$. Plugging equation 11 into equation 10 and rearranging yields

$$B\bar{\omega} = MK_p(\bar{x}_0 - \bar{x}) - A \quad (12)$$

We cannot simply invert B in order to solve for $\bar{\omega}$ as it is not a square matrix. Rather, in our case the size of B is 3×8 . Therefore, there are potentially an infinite set of $\bar{\omega}$ s that will satisfy our desired dynamics. In order to find one suitable $\bar{\omega}$, we can formulate a constrained optimization that will provide the solution closest to a specified $\bar{\omega}_0$ that adheres to our virtual model:

$$\bar{\omega} = \arg \min \frac{1}{2}(\bar{\omega} - \bar{\omega}_0)^T(\bar{\omega} - \bar{\omega}_0) \text{ such that } B\bar{\omega} = MK_p(\bar{x}_0 - \bar{x}) - A \quad (13)$$

We can solve the above optimization by incorporating the constraint into the cost function using the Lagrange Multipliers Method.

$$\bar{\omega} = \arg \min \frac{1}{2}(\bar{\omega} - \bar{\omega}_0)^T(\bar{\omega} - \bar{\omega}_0) + \bar{\lambda}^T(MK_p(\bar{x}_0 - \bar{x}) - A - B\bar{\omega}) \quad (14)$$

The solution to the above minimization is

$$\bar{\omega} = B^\dagger[MK_p(\bar{x}_0 - \bar{x}) - A] + (I - B^\dagger B)\bar{\omega}_0 \quad (15)$$

Where $B^\dagger = B^T(BB^T)^{-1}$ is the Moore-Penrose pseudoinverse of B and $I - B^\dagger B$ is the nullspace projector of B . Multiplication of the nullspace projector by $\bar{\omega}_0$ will return as much of $\bar{\omega}_0$ as possible while adhering to our desired spring-like dynamics.

In order to implement the high-level controller for robot-state derived above, we will also need a lower-level of control to guide each leg to along the trajectories specified by $\bar{\omega}$. To do this, we turn to a closed-loop central pattern generator (CPG) network.

CPG Network

Equations for one node in the network.

$$\dot{\phi}_i = \omega_b + \sum_{j \neq i} k_{ij} \sin(\phi_j - \phi_i - \tilde{\varphi}_{ij}) + \zeta_i \quad (16)$$

$$\Gamma_i = A \cos(\phi_i) \quad (17)$$

where ϕ_i is the phase of i th node, k_{ij} and $\tilde{\varphi}_{ij}$ are coupling gain and desired phase lag between i th and j th node in the network. ζ_i is the input force for the i th node, which is determined by the controller discussed in the previous section. Amplitude and frequency, A and ω are equal for all nodes in the system. And finally, Γ_i is the output of node i and will be used as the desired trajectory for the i th motor.

In our network, i can be any of four actuators on the robot, namely *front-right*, *front-left*, *hind-right*, and *hind-left*, as shown in figure below, respectively nodes 1 to 4.

In the CPG network, φ_{ij} determines the gait for robot. In order to ensure moments on the body are balanced we chose the the trot gait. Therefore, the phase lags in matrix form are:

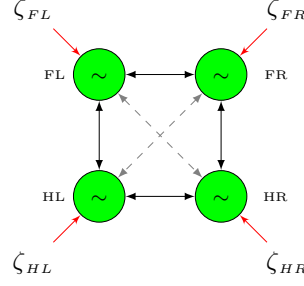


Figure 1: CPG network

$$\tilde{\Phi} = [\tilde{\varphi}_{ij}] = \begin{bmatrix} 0 & \pi & \pi & 0 \\ -\pi & 0 & 0 & \pi \\ -\pi & 0 & 0 & \pi \\ 0 & -\pi & -\pi & 0 \end{bmatrix} \quad (18)$$

This matrix is a skew-symmetric because of the phase lag property:

$$\tilde{\varphi}_{ij} = -\tilde{\varphi}_{ji} \quad (19)$$

As illustrated in Figure 1, where dashed-grey line means coupling gain of zero, we are not using fully connected network. We are using the same coupling gains in the network. Therefore, the matrix representation of the coupling gains is:

$$\kappa = [k_{ij}] = \begin{bmatrix} 0 & k & k & 0 \\ k & 0 & 0 & k \\ k & 0 & 0 & k \\ 0 & k & k & 0 \end{bmatrix} \quad (20)$$

The differential equations for our CPG-Network are as follows:

$$\dot{\phi}_1 = \omega_b + k \sin(\phi_2 - \phi_1 - \tilde{\varphi}_{12}) + k \sin(\phi_3 - \phi_1 - \tilde{\varphi}_{13}) + \zeta_1 \quad (21)$$

$$\dot{\phi}_2 = \omega_b + k \sin(\phi_1 - \phi_2 - \tilde{\varphi}_{21}) + k \sin(\phi_4 - \phi_2 - \tilde{\varphi}_{24}) + \zeta_2 \quad (22)$$

$$\dot{\phi}_3 = \omega_b + k \sin(\phi_1 - \phi_3 - \tilde{\varphi}_{31}) + k \sin(\phi_4 - \phi_3 - \tilde{\varphi}_{34}) + \zeta_3 \quad (23)$$

$$\dot{\phi}_4 = \omega_b + k \sin(\phi_2 - \phi_4 - \tilde{\varphi}_{42}) + k \sin(\phi_3 - \phi_4 - \tilde{\varphi}_{43}) + \zeta_4 \quad (24)$$

Using the following linear transformation, $\varphi_{ij} = \phi_j - \phi_i - \tilde{\varphi}_{ij}$, and knowing that $\dot{\tilde{\varphi}}_{ij} = 0$ and $\varphi_{ij} = -\varphi_{ji}$, we arrive at the differential equations in phase lags coordination:

$$\dot{\phi}_1 = \omega_b + k \sin(\varphi_{12}) + k \sin(\varphi_{13}) + \zeta_1 \quad (25)$$

$$\dot{\phi}_2 = \omega_b + k \sin(\varphi_{21}) + k \sin(\varphi_{24}) + \zeta_2 \quad (26)$$

$$\dot{\phi}_3 = \omega_b + k \sin(\varphi_{31}) + k \sin(\varphi_{34}) + \zeta_3 \quad (27)$$

$$\dot{\phi}_4 = \omega_b + k \sin(\varphi_{42}) + k \sin(\varphi_{43}) + \zeta_4 \quad (28)$$

$$\dot{\varphi}_{12} = -2k \sin(\varphi_{12}) - k \sin(\varphi_{13}) + k \sin(\varphi_{24}) + \zeta_2 - \zeta_1 \quad (29)$$

$$\dot{\varphi}_{13} = -2k \sin(\varphi_{13}) - k \sin(\varphi_{12}) + k \sin(\varphi_{34}) + \zeta_3 - \zeta_1 \quad (30)$$

$$\dot{\varphi}_{24} = -2k \sin(\varphi_{24}) + k \sin(\varphi_{12}) - k \sin(\varphi_{34}) + \zeta_4 - \zeta_2 \quad (31)$$

$$\dot{\varphi}_{34} = -2k \sin(\varphi_{34}) + k \sin(\varphi_{13}) - k \sin(\varphi_{24}) + \zeta_4 - \zeta_3 \quad (32)$$

Using a small angle approximation, we can express the above equations as the following linear autonomous system:

$$\dot{\Phi} = \frac{d}{dt} \Phi = \frac{d}{dt} \begin{bmatrix} \varphi_{12} \\ \varphi_{13} \\ \varphi_{24} \\ \varphi_{34} \end{bmatrix} = k \begin{bmatrix} -2 & -1 & 1 & 0 \\ -1 & -2 & 0 & 1 \\ 1 & 0 & -2 & -1 \\ 0 & 1 & -1 & -2 \end{bmatrix} \Phi = \hat{A} \Phi. \quad (33)$$

The eigenvalues of \hat{A} are $-4k$, $-2k$, $-2k$, and 0 . We expect one zero eigenvalue due to the loop in our CPG network, which creates the following linear relationship between phase lags

$$\varphi_{12} - \varphi_{13} + \varphi_{24} - \varphi_{34} = 0 \quad (34)$$

Negative eigenvalues show that our autonomous (unforced) system is locally, asymptotically stable and will converge to our desired phase lags. However, CPG inputs, ζ_i , will still affect phase lag convergence. As we see from equations 29 to 32, phase lags are sensitive to mutual difference of ζ_i . Therefore, we should design the closed-loop system such that we minimize the following costs:

$$\min \zeta_2 - \zeta_1, \zeta_3 - \zeta_1, \zeta_4 - \zeta_2, \zeta_4 - \zeta_3 \quad (35)$$

This minimization is equivalent to the following minimization

$$\min \sigma_\zeta = \sum_{i=1}^4 (\zeta_i - \bar{\zeta})^2, \text{ where } \bar{\zeta} = \frac{1}{4} \sum_{i=1}^4 \zeta_i \quad (36)$$

Therefore, to better achieve convergence to our desired phase lags, it is important to reduce variance of CPG inputs (ζ_i). We take advantage of the dynamical nullspace, $(I - B^\dagger B)\bar{\omega}_0$, from our controller (equation 15) and define this constraint (low variance input) as a task for the controller.

Input Scheduling

In order to control downward and upward motion separately, we split each CPG node input (ζ_i) in two parts as follows:

$$\dot{\phi}_i = \omega_b + \sum_{j \neq i} k_{ij} \sin(\phi_j - \phi_i - \tilde{\varphi}_{ij}) + \zeta_i^+ + \zeta_i^- \quad (37)$$

$$\Gamma_i = A \cos(\phi_i) \quad (38)$$

These inputs have the property $\zeta_i^+ \zeta_i^- = 0$, which means they cannot be active simultaneously. We can distinguish upward and downward motions by checking the sign of $\sin(\phi)$ as shown in Figure 2.

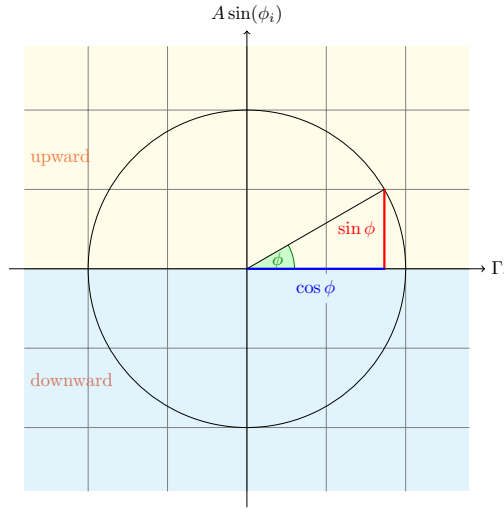


Figure 2: phase

CPG forces (ω^+ , ω^-) are provided by our controller. Thus, in order to build our CPG force (ζ_i), we will use:

$$\zeta_i^+ = \frac{1}{2}[1 + \text{sign}(\sin \phi)]\omega^+ \quad (39)$$

$$\zeta_i^- = \frac{1}{2}[1 - \text{sign}(\sin \phi)]\omega^- \quad (40)$$

$$\zeta_i = \zeta_i^+ + \zeta_i^- \quad (41)$$