University of Minnesota–Twin Cities

Human Robotic Control

Realistic Arm Dynamic

RanTian
UMID: 4970588

April 26, 2015

Abstract

Unlike previous model we already built which is torque-driven, we add springs into our model and drive the links by changing the spring coefficients. This model is somewhat similar to the human arm -the springs act like muscles and the state of the muscles correspond to the spring coefficients. In this documentation, we introduce the dynamic model for one link and two links. Based on these two, we can extend it to high-dimension model.

One Link Model

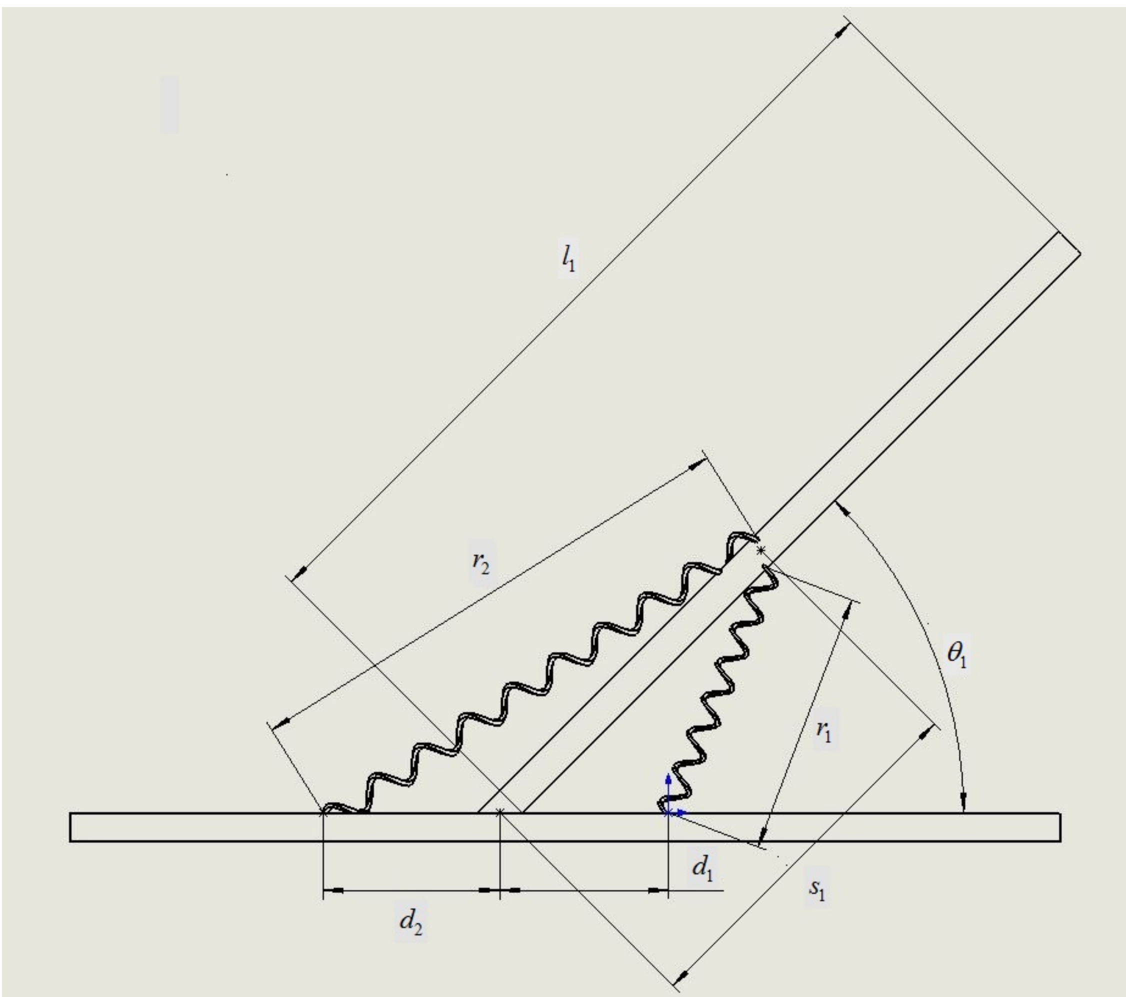

Figure 1: One Link Dynamic Scheme

Figure 1 shows us the model for one link. Here are the assumptions for our model:

- The link is rigid and uniform density, and is fixed on O
- The arm motion is restricted in XY plane, in other word, no movement in Z axis
- The state variable $(\theta, \dot{\theta})$ are available to measure
- The friction in our case is only related to$\dot{\theta}$

Here comes the parameter notations:

- $(\theta, \dot{\theta})$: state variables
- l: link length
- (d_1, d_2): the position where the spring attached in ground
- s: the position where the spring attached in the link
- (k_1, k_2): the coefficients for the two springs
- (r_{1rest}, r_{2rest}): the rest length for the two springs
- (r_1, r_2): the length for the two springs given any θ

The key for this model is to represent (r_1, r_2) with θ , according to the *Cosine Law*:

$$r_1^2 = s^2 + d_1^2 - 2sd_1cos(\theta) \tag{1}$$

$$r_2^2 = s^2 + d_2^2 + 2sd_2cos(\theta) \tag{2}$$

The potential energy for the springs is :

$$P_{spring} = k_1(r_1 - r_{1rest}^2)^2 + k_2(r_2 - r_{2rest})^2 \tag{3}$$

which can be simply noted as $P_{spring}(\theta)$
The kinetic energy and potential energy of gravity are the same as the previous model we have, the input for the dynamic is u: $(k1, k2)$, then final *Lagrange Equation*:

$$M(\theta)\ddot{\theta} + C(\theta, \dot{\theta}) + \phi(\theta, u) = 0 \tag{4}$$

To test the model, refer to the codes in Google drive under Realistic Arm - One Link. Another code is for target reaching using iterative Linear Quadratic Regulator (ILQR).

Two Link Model

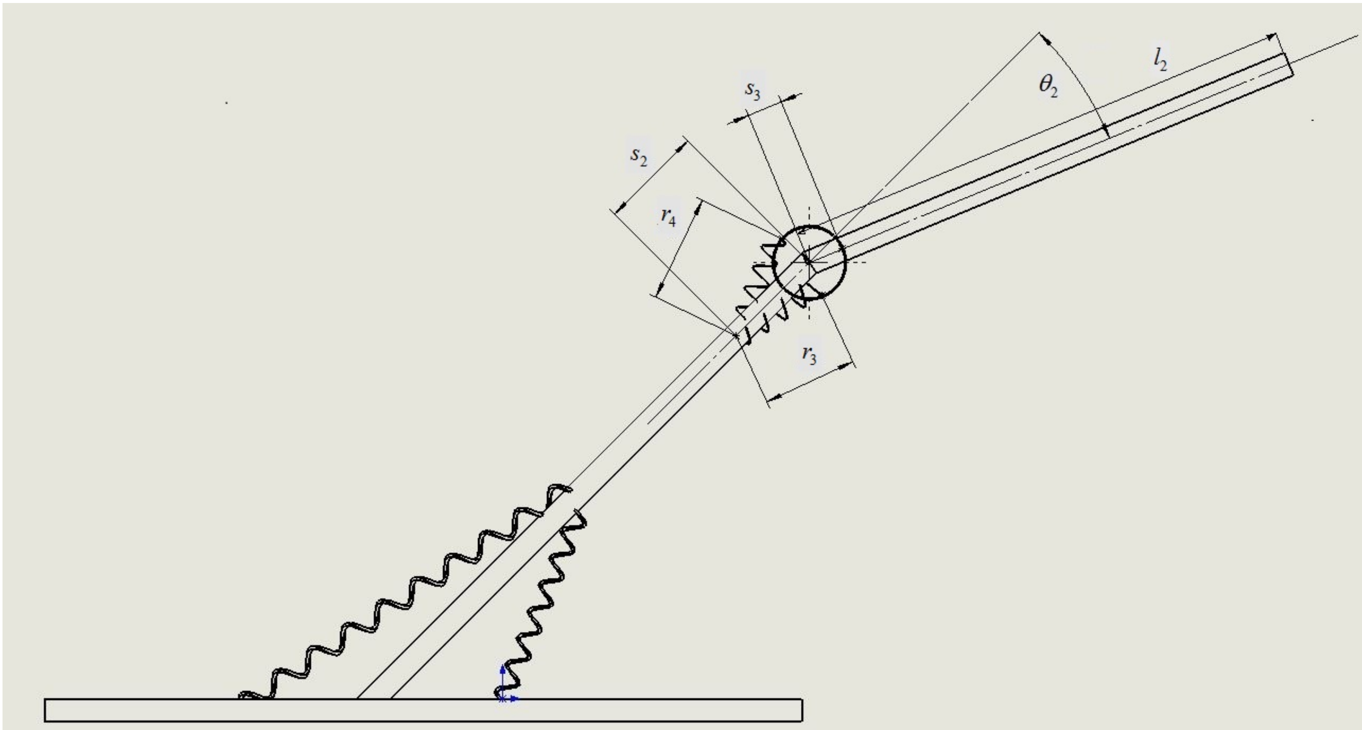

Figure 2: Two Link Dynamic Scheme

Base on the one link model, we add two more springs on the joint of two link, like Figure 2 shows. The assumptions are the same as above, the parameter notations are:

- $(\theta_1, \theta_2, \dot{\theta}_1, \dot{\theta}_2)$: state variables
- (l_1, l_2): link lengths for two links
- (d_1, d_2): the position where the spring attached in ground
- s_1: the position where the first two springs attached in first link
- s_2: the position where the second two springs attached in first link
- s_3: the position where the second two springs attached in second link
- (k_1, k_2, k_3, k_4): the coefficients for the four springs
- $(r_{1rest}, r_{2rest}, , r_{3rest}, r_{4rest})$: the rest length for the four springs
- (r_1, r_2, r_3, r_4): the length for the four springs given any (θ_1, θ_2)

The representations for (r_1, r_2) are the same as above

$$r_1^2 = {s_1}^2 + d_1^2 - 2sd_1cos(\theta_1) \tag{5}$$

$$r_2^2 = {s_1}^2 + d_2^2 + 2sd_2cos(\theta_1) \tag{6}$$

Similar principles apply to (r_3, r_4) except that now the angle is:

$$cos(\pi/2 - \theta_2) = sin(\theta_2) \tag{7}$$

which yield to give us:

$$r_3^2 = {s_2}^2 + s_3^2 - 2s_2s_3sin(\theta_2) \tag{8}$$

$$r_4^2 = {s_2}^2 + s_3^2 + 2s_2s_3sin(\theta_2) \tag{9}$$

The potential energy for the springs is :

$$P_{spring} = k_1(r_1 - r_{1rest}^2)^2 + k_2(r_2 - r_{2rest})^2 + k_3(r_3 - r_{3rest}^2)^2 + k_4(r_4 - r_{4rest})^2 \tag{10}$$

which can be simply noted as $P_{spring}(\theta_1, \theta_2)$
The *Lagrange Equation* is the same as before:

$$M(\theta)\ddot{\theta} + C(\theta, \dot{\theta}) + \phi(\theta, u) = 0 \tag{11}$$

but now the dynamic dimension is 4, which means θ contains (θ_1, θ_2),$\dot{\theta}$ contains $(\dot{\theta_1}, \dot{\theta_2})$
The test codes is on the Google Drive under Realistic Arm - Two Link

Analysis

Base on the simulation result in the Matlab, here comes several results:

- There are two ways to realize the target-reaching goal, linearization at the initial state and inverse Kinetic, the former method works well when the target is not far from the initial state, the reason for this is that the degree of linearity in this model is high.
- We use these two methods to get the first trajectory in the target-reaching goal and applied Iterative Linear Quadratic Regulator(ILQR), both two methods wok well.
- The ILQR method gives us different final cost when we choose different first trajectories.
- During ILQR process, we can always get the case for negative spring coefficients, which means the springs are compressed. But in reality, human muscles can be only stretched, thus we need a constrained LQR way to redo these processes.
- Just like the way we did when we construct the 2-link model from 1-link, we are able to extend our model to higher dimension.

References

[1] *GeneralizedLQR.pdf* in Google Drive

[2] *SymbolicLagrangianSolverUserGuide.pdf* in Google Drive

[3] *ILQRAlgorithmApplication.pdf* in Google Drive

[4] Matlab code *RealisticArm* in Google Drive

Appendix

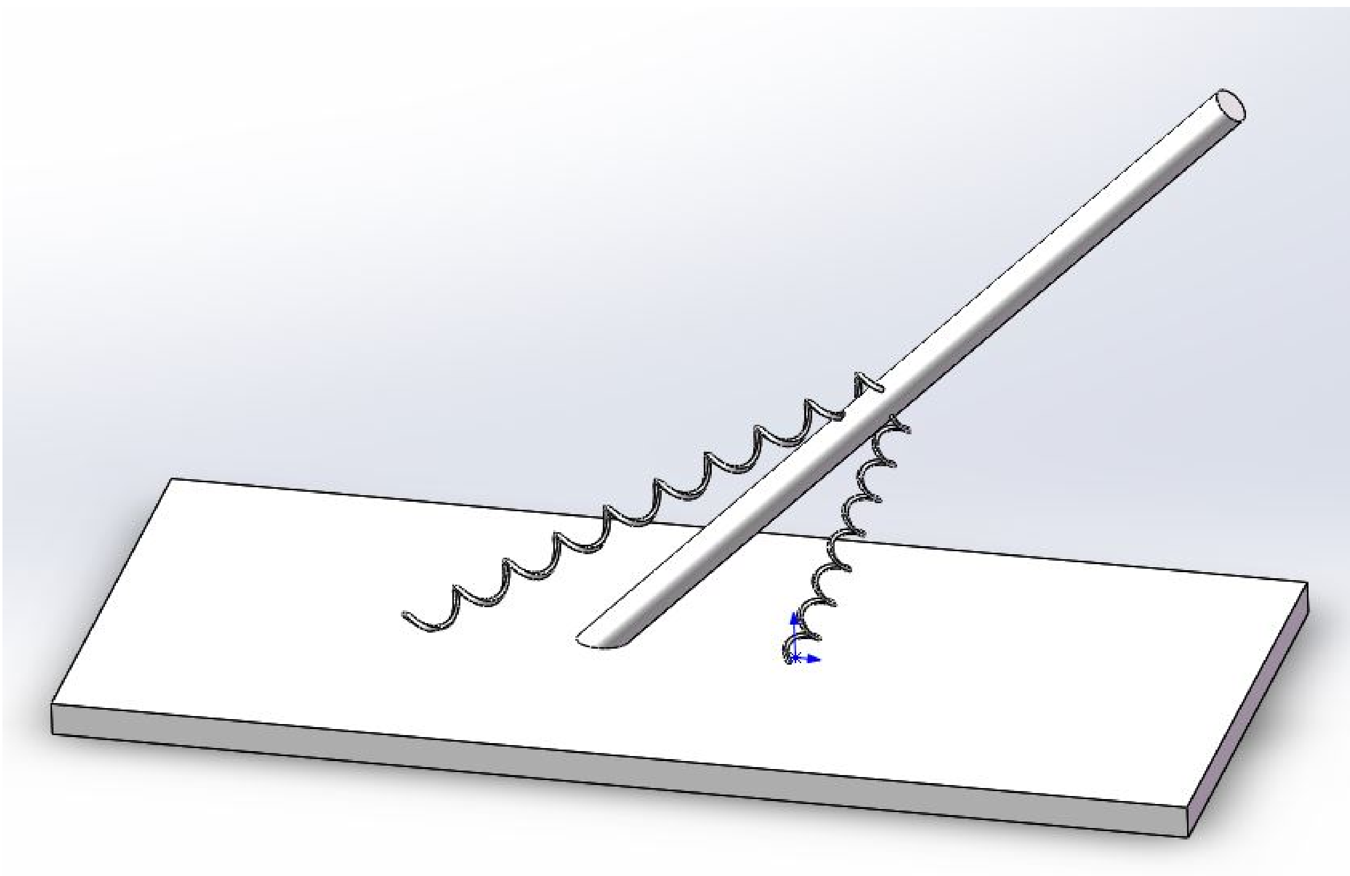

Figure 3: One Link Dynamic 3-D Version

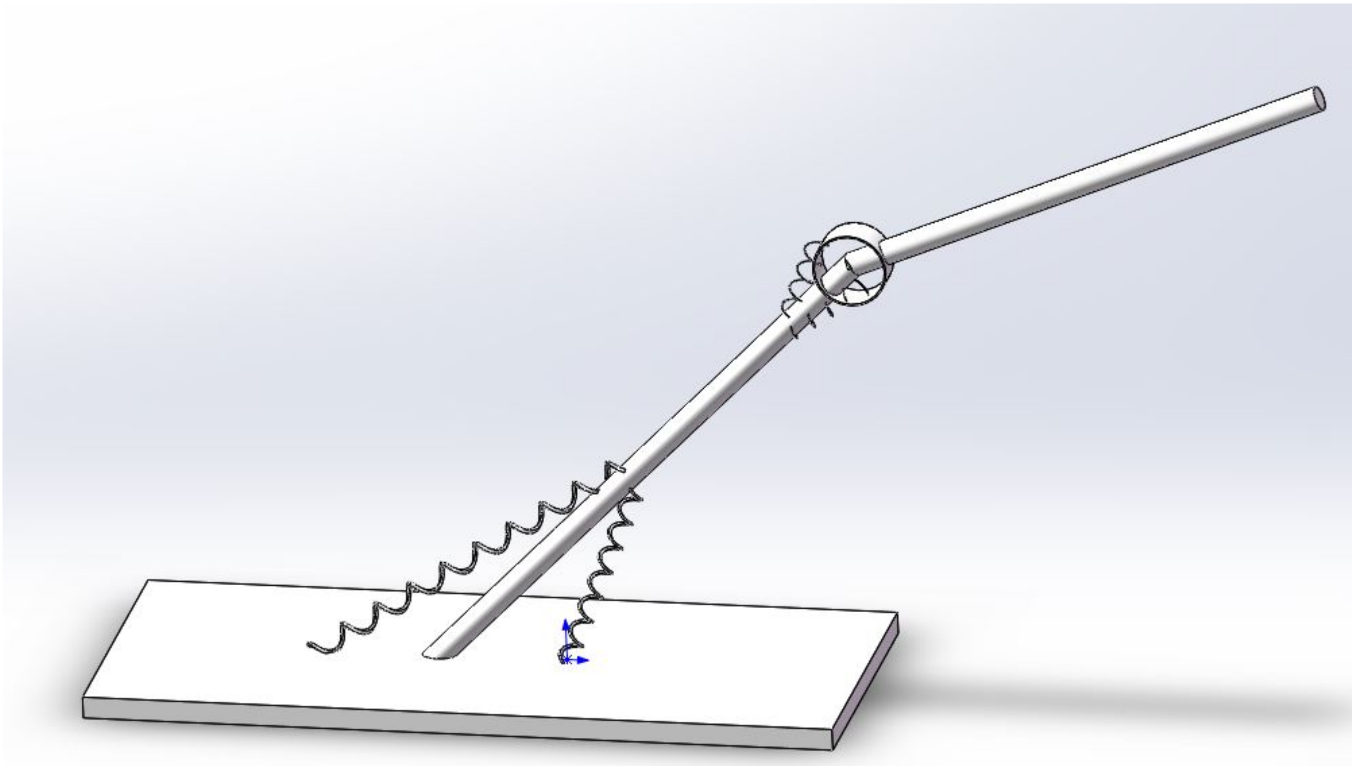

Figure 4: Two Link Dynamic 3-D Version