Graph-Based Dynamics and Network Control of a Single Articulated Robotic System

1st Given Name Surname
dept. name of organization (of Aff.)
name of organization (of Aff.)
City, Country
email address or ORCID

2nd Given Name Surname
dept. name of organization (of Aff.)
name of organization (of Aff.)
City, Country
email address or ORCID

***Abstract*—Short summary of the paper**

I. Introduction

Vector-network method, what is the same and different from that paper? Why overparameterization?

II. Preliminaries

Talk about prereqs (Graph theory, definitions)

- Incidence matrix $D(\mathcal{G})$
- diag extracts the diagonal of a square matrix as a column vector
- $\odot$ denotes the Hadamard (elementwise) product of two matrices with the same dimensions.

III. Network Dynamics

In order to explore this way of looking at robotic systems, we present a very simple class of systems in 2D space: a collection of n point masses fixed together with k massless rods, which act as distance constraints. Masses with multiple rods connected to them act as revolute joints. We define the Euler-Lagrange equation for this system as:

$$M\ddot{Q} + MG + \Gamma(Q,\dot{Q}) = F \tag{1}$$

where:

$$Q = \begin{bmatrix} x_1 & y_1 \\ \vdots & \vdots \\ x_n & y_n \end{bmatrix} = \begin{bmatrix} r_1^T \\ \vdots \\ r_n^T \end{bmatrix} \in \mathbb{R}^{n\times 2},\ G = \begin{bmatrix} 0 & g \\ \vdots & \vdots \\ 0 & g \end{bmatrix} \in \mathbb{R}^{n\times 2}$$

$$M = \begin{bmatrix} m_1 & & \\ & \ddots & \\ & & m_n \end{bmatrix} \in \mathbb{R}^{n\times n}$$

$$F = \begin{bmatrix} f_{1,x} & f_{1,y} \\ \vdots & \vdots \\ f_{n,x} & f_{n,y} \end{bmatrix} \in \mathbb{R}^{n\times 2}$$

and $\Gamma(Q,\dot{Q}) \in \mathbb{R}^{n\times 2}$ is the matrix of constraint forces. The goal of this section is to show how this matrix relates to and can be represented with a graph structure. We start by defining a network of constraints where the n point masses represent nodes and the k distance constraints represent edges. The incidence matrix of this graph can be written as:

$$D(G) = \begin{bmatrix} | & & | \\ d_1 & \dots & d_k \\ | & & | \end{bmatrix} \in \mathbb{R}^{n\times k}$$

where the column d_i corresponds to the ith edge and contains a 1 and -1 at the indices where masses are connected.

Continuing in the vein of graph theory, we can define an edge coordinates matrix given by the transformation:

$$Q_e = D(\mathcal{G})^T Q = \begin{bmatrix} r_{1e}^T \\ \vdots \\ r_{ke}^T \end{bmatrix} \in \mathbb{R}^{k\times 2}$$

$$r_{ie} = Q^T d_i$$

where each row r_{ie}^T of Q_e is the vector displacement between masses with a distance constraint. Therefore $\|r_{ie}\| = \ell_i$ is the constant length of the ith massless rod. We will refer to the rows of Q as the node coordinates and the rows of Q_e as the edge coordinates.

The constraint equations are written as:

$$h_i(Q) = \|Q^T d_i\|^2 - \ell_i^2 = 0$$

$$\frac{dh_i}{dQ} = 2Q^T d_i d_i^T$$

Putting the constraints together, we define the constraint matrix as:

$$A(Q) \triangleq \frac{1}{2}\frac{dh}{dQ} = \begin{bmatrix} Q^T d_1 d_1^T \\ \vdots \\ Q^T d_k d_k^T \end{bmatrix} \in \mathbb{R}^{2k\times n}$$

We also define the vector of constraint forces λ which are the magnitudes of the forces on the k constraint rods:

$$\lambda = \begin{bmatrix} \lambda_1 \\ \vdots \\ \lambda_k \end{bmatrix} \in \mathbb{R}^{k\times 1}$$

Identify applicable funding agency here. If none, delete this.

Now we can begin to solve for Γ:

$$\begin{aligned}
\Gamma &= A(Q)^T(\lambda \otimes I_{2\times 2}) \\
&= \begin{bmatrix} Q^T d_1 d_1^T \\ \vdots \\ Q^T d_k d_k^T \end{bmatrix}^T \begin{bmatrix} \lambda_1 I_{2\times 2} \\ \vdots \\ \lambda_k I_{2\times 2} \end{bmatrix} \\
&= \begin{bmatrix} d_1 d_1^T Q & \dots & d_k d_k^T Q \end{bmatrix} \begin{bmatrix} \lambda_1 I_{2\times 2} \\ \vdots \\ \lambda_k I_{2\times 2} \end{bmatrix} \\
&= \sum_{i=1}^{k} d_i \lambda_i d_i^T Q \\
&= D(\mathcal{G}) \Lambda D(\mathcal{G})^T Q \\
&= L_w(\mathcal{G}) Q
\end{aligned}$$

Talk about why this is interesting

$$M\ddot{Q} + MG + D(\mathcal{G})\Lambda D(\mathcal{G})^T Q = F$$

Now, we must solve for λ.

$$\ddot{Q} = -G + M^{-1}(F - D(\mathcal{G})\Lambda D(\mathcal{G})^T Q)$$

$$\begin{bmatrix} d_1^T \dot{Q} Q^T d_1 \\ \vdots \\ d_k^T \dot{Q} Q^T d_k \end{bmatrix} = 0$$

Take the derivative:

$$\begin{bmatrix} d_1^T \ddot{Q} Q^T d_1 + \|\dot{Q}^T d_1\|^2 \\ \vdots \\ d_k^T \ddot{Q} Q^T d_k + \|\dot{Q}^T d_k\|^2 \end{bmatrix} = 0$$

$$\begin{bmatrix} d_1^T M^{-1}(F - D(\mathcal{G})\Lambda D(\mathcal{G})^T Q) Q^T d_1 + \|\dot{r}_{1e}\|^2 \\ \vdots \\ d_k^T M^{-1}(F - D(\mathcal{G})\Lambda D(\mathcal{G})^T Q) Q^T d_k + \|\dot{r}_{ke}\|^2 \end{bmatrix} = 0$$

Note that $d_i^T G = 0$

$$\begin{bmatrix} d_1^T M^{-1} D(\mathcal{G}) \Lambda Q_e r_{1e} \\ \vdots \\ d_k^T M^{-1} D(\mathcal{G}) \Lambda Q_e r_{ke} \end{bmatrix} = \begin{bmatrix} d_1^T M^{-1} F r_{1e} + \|\dot{r}_{1e}\|^2 \\ \vdots \\ d_k^T M^{-1} F r_{ke} + \|\dot{r}_{ke}\|^2 \end{bmatrix}$$

$$\begin{bmatrix} \mathrm{Tr}\left(Q_e r_{1e} d_1^T M^{-1} D(\mathcal{G})\Lambda\right) \\ \vdots \\ \mathrm{Tr}\left(Q_e r_{ke} d_k^T M^{-1} D(\mathcal{G})\Lambda\right) \end{bmatrix} = \begin{bmatrix} d_1^T M^{-1} F r_{1e} + \|\dot{r}_{1e}\|^2 \\ \vdots \\ d_k^T M^{-1} F r_{ke} + \|\dot{r}_{ke}\|^2 \end{bmatrix}$$

Notice the ith row of the left side of the equation is the trace of the outer product of the vectors $Q_e r_{ie}$ and $D(\mathcal{G})^T M^{-1} d_i$ times the diagonal matrix Λ. This is equivalent to the Hadamard (elementwise) product of $Q_e r_{ie}$ and $D(\mathcal{G})^T M^{-1} d_i$, inner product with λ.

$$\begin{bmatrix} \left(D(\mathcal{G})^T M^{-1} d_1 \odot Q_e r_{1e}\right)^T \\ \vdots \\ \left(D(\mathcal{G})^T M^{-1} d_k \odot Q_e r_{ke}\right)^T \end{bmatrix} \lambda = \begin{bmatrix} d_1^T M^{-1} F r_{1e} + \|\dot{r}_{1e}\|^2 \\ \vdots \\ d_k^T M^{-1} F r_{ke} + \|\dot{r}_{ke}\|^2 \end{bmatrix}$$

$$\left(D(\mathcal{G})^T M^{-1} D(\mathcal{G}) \odot Q_e Q_e^T\right)\lambda = \mathrm{diag}\left(D(\mathcal{G})^T M^{-1} F Q_e^T + \dot{Q}_e \dot{Q}_e^T\right)$$

If the constraints are independent, a solution for λ exists.

$$\lambda = \left(D(\mathcal{G})^T M^{-1} D(\mathcal{G}) \odot Q_e Q_e^T\right)^{-1} \mathrm{diag}\left(D(\mathcal{G})^T M^{-1} F Q_e^T + \dot{Q}_e \dot{Q}_e^T\right) \tag{2}$$

Note the equation for λ depends only on edge coordinates and their derivatives.

IV. Network Control

Sometimes it may be useful to directly control the edge coordinates and their velocities without regard for the node coordinates. For example, we may want to control the orientation of the rods that connect masses together and their angular velocities while paying no attention to where the robot exists in space or what forces may be acting on the entire robot (such as gravity). We propose a modification to the Euler-Lagrange equation given by equation 1 by multiplying it by the edge transformation matrix $D(\mathcal{G})^T$:

$$\ddot{Q}_e + D(\mathcal{G})^T M^{-1}\Gamma(Q_e, \dot{Q}_e) = D(\mathcal{G})^T M^{-1} F$$

Now, since the equation only contains edge coordinates, their velocities, and force inputs, we can employ a variety of methods to control the edge coordinates to some desired trajectory given by $Q_{e,d}$. We will use the feedback linearization technique in this paper for simplicity. We propose the controller:

$$\begin{aligned} D(\mathcal{G})^T M^{-1} F = -k_1(Q_e - Q_{e,d}) - k_2(\dot{Q}_e - \dot{Q}_{e,d}) + \ddot{Q}_{e,d} \\ + D(\mathcal{G})^T M^{-1}\Gamma(Q_e, \dot{Q}_e) \end{aligned} \tag{3}$$

In general, $D(\mathcal{G})$ is not full rank, therefore we cannot write an equation for F. We will look at how to solve for the control input F through examples.

V. Results

A. Two link pendulum

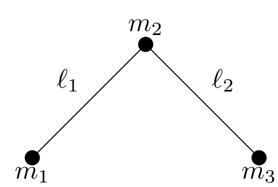

Fig. 1. Figure

$$n = 3,\ k = 2$$

$$D(\mathcal{G}) = \begin{bmatrix} 1 & 0 \\ -1 & -1 \\ 0 & 1 \end{bmatrix}$$

In this example, define the forces on each of the masses as:

$$f_1 = m_1 u_1 R_{90} \frac{r_{1e}}{\ell_1} + m_1 \begin{bmatrix} 0 \\ g \end{bmatrix}$$

$$f_2 = m_2 \begin{bmatrix} 0 \\ g \end{bmatrix}$$

$$f_3 = m_3 u_2 R_{90} \frac{r_{2e}}{\ell_2} + m_3 \begin{bmatrix} 0 \\ g \end{bmatrix}$$

$$F = \begin{bmatrix} f_1^T \\ f_2^T \\ f_3^T \end{bmatrix}$$

where:

$$R_{90} = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix},\ u_1, u_2 \in \mathbb{R}$$

We choose the forces to be perpendicular to the bars they are connected to plus a term to offset gravity. This simplifies the dynamics by cancelling out the force term in equation 2. Rewrite equation 3 with these forces, and represent the rows of the right side of the equation with b_1^T, b_2^T:

$$\begin{bmatrix} u_1 (R_{90} \frac{r_{1e}}{\ell_1})^T \\ u_2 (R_{90} \frac{r_{2e}}{\ell_2})^T \end{bmatrix} = \begin{bmatrix} b_1^T \\ b_2^T \end{bmatrix}$$

The rows of both sides of this equation are not multiples of each other, so we use a projection to find u_i such that the equation is close:

$$u_1 = b_1 \cdot R_{90} \frac{r_{1e}}{\ell_1},\ u_2 = b_2 \cdot R_{90} \frac{r_{2e}}{\ell_2}$$

This controller will converge $Q_e \to Q_{e,d}$ as $t \to 0$, without regard to the absolute positions of the coordinates Q. We will use another controller to control the absolute position of the robot, by applying additional force to node 2. Define the desired position of node 2 as $r_{2,d}$, where $\ddot{r}_{2,d} = 0$, and redefine f_2 as:

$$f_2 = -k_3(r_2 - r_{2,d}) - k_4(\dot{r}_2 - \dot{r}_{2,d}) + m_2 \begin{bmatrix} 0 \\ g \end{bmatrix}$$

As $r_2 \to r_{2,d}$ and $\dot{r}_2 \to \dot{r}_{2,d}$, f_2 will only exert force to resist gravity and the force term can be cancelled out of the control equation as before. The trade-off is that we sacrifice edge control accuracy while node control converges. Now, we are able to control the relative positions between adjacent nodes and control the absolute position of the robot to a certain degree simultaneously.

B. Four link pendulum

VI. Conclusion

VII. Summary

- Overparameterization you can extract the graph, why?
- Matrix form of the Euler-Lagrange equation
- Derivation of Γ (constraint force vectors) and λ (constraint force magnitudes)
- Example of simple overparameterized system with distance constraints 2-bar
- What info is needed for λ (when you cancel out forces or not?)
- PD control of edge coordinates
- Adding control of global variables with edge control (leader-follower)
- More examples: jointed wing, star, X-wing

References

[1] G. Eason, B. Noble, and I. N. Sneddon, “On certain integrals of Lipschitz-Hankel type involving products of Bessel functions,” Phil. Trans. Roy. Soc. London, vol. A247, pp. 529–551, April 1955.

[2] J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.

[3] I. S. Jacobs and C. P. Bean, “Fine particles, thin films and exchange anisotropy,” in Magnetism, vol. III, G. T. Rado and H. Suhl, Eds. New York: Academic, 1963, pp. 271–350.

[4] K. Elissa, “Title of paper if known,” unpublished.

[5] R. Nicole, “Title of paper with only first word capitalized,” J. Name Stand. Abbrev., in press.

[6] Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, “Electron spectroscopy studies on magneto-optical media and plastic substrate interface,” IEEE Transl. J. Magn. Japan, vol. 2, pp. 740–741, August 1987 [Digests 9th Annual Conf. Magnetics Japan, p. 301, 1982].

[7] M. Young, The Technical Writer’s Handbook. Mill Valley, CA: University Science, 1989.