Magnetic Forces and Torques using Dipole Model

Dr. Axel Krieger, Transcribed by Naveed Riaziat

November 18th, 2021

1 Preliminaries

$$\begin{bmatrix} \omega \\ v \end{bmatrix} = AI \tag{1}$$

$$I = A^{-1} * \begin{bmatrix} \omega_{desired} \\ v_{desired} \end{bmatrix} \tag{2}$$

Goal: Control angular velocity and linear velocity as a function of coil current.

2 Single Electromagnet

$$m = \frac{1}{\mu_0} VR\hat{h} \tag{3}$$

where m is the magnetic moment of the permanent magnet robot, $\hat{h}$ is the unit vector heading, μ_0 is the permeability of vacuum, V is the volume, and R is the magnetic remanence (Tesla).

$$B(m_c, r) = \frac{\mu_0}{4\pi ||r||^3}(3(m_c \cdot \hat{r})\hat{r} - m_c) \tag{4}$$

where B is the magnetic field, m_c is the magnetic moment of the EM coil,

$$m_c = NSI\hat{n} \tag{5}$$

where N is the number of coils, S is the cross-sectional surface area of the coil, I is the coil curent (amps), and $\hat{n}$ is the unit vector normal to the loop.
Then, the unit magnetic field is as follows:

$$\tilde{B}(\tilde{m_c}, r) = \frac{\mu_0}{4\pi ||r||^3}(3(\tilde{m_c} \cdot \hat{r})\hat{r} - \tilde{m_c}) \tag{6}$$

where the unit magnetic moment is

$$\tilde{m}_c = NS(1A)\hat{n} \tag{7}$$

As such,

$$B = \tilde{B} \cdot i \tag{8}$$

where i is the unit-less current ratio.

3 Multiple Electromagnets

Now, with our magnetic robot in the middle of 4 independent coils, each with their own unique magnetic moments and currents, we extend the calculation.

As shown before, for a single coil k with a distance r_k from the robot in this setup, we have:

$$B_k(r_k) = \tilde{B_k}(r_k) \cdot i \tag{9}$$

For multiple coils:

$$B(p) = \sum_{k=1}^{n} B_k(r_k) = \sum_{k=1}^{n} \tilde{B_k}(r_k) \cdot i_k \tag{10}$$

$$B(p) = [\tilde{B_1}(r_1)\tilde{B_2}(r_2)\ldots\tilde{B_n}(r_n)] \begin{bmatrix} i_1 \\ i_2 \\ \vdots \\ i_n \end{bmatrix} \tag{11}$$

The magnetic forces and torques are as follows:

$$T = m \times B(p) \tag{12}$$

$$F = \nabla(B(p) \cdot m) \tag{13}$$

$$F = \begin{bmatrix} \frac{\delta B(p)}{\delta x} \\ \frac{\delta B(p)}{\delta y} \\ \frac{\delta B(p)}{\delta z} \end{bmatrix} \cdot m \tag{14}$$

Therefore,

$$\frac{\delta B(p)}{\delta x} = [\frac{\delta B_1\tilde{(}r_1)}{\delta x} \ldots \frac{\delta B_n\tilde{(}r_n)}{\delta x}] \begin{bmatrix} i_1 \\ i_2 \\ \vdots \\ i_n \end{bmatrix} = \tilde{B_x}(p)I \tag{15}$$

$$\begin{bmatrix} T \\ F \end{bmatrix} = \begin{bmatrix} m \times \tilde{B}(p) \\ m^T\tilde{B_x}(p) \\ m^T\tilde{B_y}(p) \\ m^T\tilde{B_z}(p) \end{bmatrix} \begin{bmatrix} i_1 \\ i_2 \\ \vdots \\ i_n \end{bmatrix} \tag{16}$$

$$\begin{bmatrix} T \\ F \end{bmatrix} = AI \tag{17}$$

if A is square and det(A) != 0,

$$I = A^{-1} \begin{bmatrix} T \\ F \end{bmatrix} \tag{18}$$

otherwise, if A is not square or A is singular,

$$I = A^{+} \begin{bmatrix} T \\ F \end{bmatrix} \tag{19}$$

where $A^+ = (A^TA)^{-1}A^T$

Given T and F, we can find the desired outputs:

$$T = C\omega \tag{20}$$

$$F = bv \tag{21}$$

Therefore,

$$I = A^{+} \begin{bmatrix} c\omega \\ bv \end{bmatrix} \tag{22}$$

4 Challenges and Implementation

1. $m^T \tilde{B_x}(p)$ requires numerical solution
2. Magnetic robot aligns very quickly with B, so it is hard to control Θ, ω with I

It is often easier to ignore the angular dynamics and align the robot in the direction of the magnetic field $B(p)$, like a compass.

$$\tilde{B}(\tilde{m_c}, r) = \frac{\mu_0}{4\pi||r||^3}(3(\tilde{m_c} \cdot \hat{r})\hat{r} - \tilde{m_c}) \tag{23}$$

with

$$\tilde{F}(r, \tilde{m_c}, m_m) = \nabla(\tilde{B}(r) \cdot m_m) = \frac{3\mu_0}{4\pi||r||^5}[(\tilde{m_c} \cdot r)m_m + (m_m \cdot r)\tilde{m_c} + (\tilde{m_c} \cdot m_m)r - \frac{5(\tilde{m_c} \cdot r)(m_m \cdot r)}{||r||^2}r] \tag{24}$$

is the unit force from a single EM onto a robot with magnetic dipole moment m_m. Thus,

$$\begin{bmatrix} B \\ F \end{bmatrix} = \begin{bmatrix} \tilde{B}(p) \\ \tilde{F_1}(\tilde{m_{c_1}}, r_1, m) \dots \tilde{F_m}(\tilde{m_{c_n}}, r_n, m) \end{bmatrix} \begin{bmatrix} i_1 \\ \vdots \\ i_n \end{bmatrix} = CI \tag{25}$$

We will implement this directly for control in lab 4. If C is square and non-singular,

$$I = C^{-1} \begin{bmatrix} B_{desired} \\ F_{desired} \end{bmatrix} = C^{-1} \begin{bmatrix} \alpha\hat{h}_{des} \\ bv_{des} \end{bmatrix} \tag{26}$$

Otherwise,

$$I = C^{+} \begin{bmatrix} B_{desired} \\ F_{desired} \end{bmatrix} = C^{+} \begin{bmatrix} \alpha\hat{h}_{des} \\ bv_{des} \end{bmatrix} \tag{27}$$

This is paired with our simplified dynamics, $F = b\dot{x}$ and $\hat{h}||B$. We have to tune α so you do not saturate the maximum coil current for orientation. Experimentally, this should be approximately 0.2 Amps or 20% of the maximum coil current.

5 Control Strategies

1. **[Lab 3]** PID Direct Current Control
   Problems
   (a) No orientation control
   (b) Robot movement depends strongly on position.
2. **[Lab 4] ...**

6 PID Force Control with Orientation

The equations for the PID control are as follows.

$$e = p_{des} - p_m \tag{28}$$

$$F = Pe + D\dot{e} + I\int edt \tag{29}$$

$$I = C(m,p)^{-1} * \begin{bmatrix} \alpha\hat{h}_{des} \\ F \end{bmatrix} \tag{30}$$

You will need to return P, I, D and α. α should be selected to not use very much current (10-20% of max). This will run open-loop orientation/heading control and result in position-independent performance.

A problem with this control is that the dipole model breaks down significantly near the coils. Additionally, estimating the robot as a single point can become a problematic assumption as well. To mitigate this, there are numerical FEA Biot-Savart law solvers. The implementation is of course slower, but much more accurate.

Additionally, there are points where the matrix C is ill-conditioned and can lead to "sticking points". This arises from attempting to control the heading and direction at once, where the coils cannot both produce the desired field direction and ∇B. To mitigate this, we can add more EM coils or just avoid certain configurations.

7 Recap: Velocity Kinematics and Jacobians

Consider a robot with end effector position $\vec{x}$ and joint angles $\theta_1 \ldots \theta_i$. Given $\vec{x} = f(\theta)$, what is the velocity $\dot{\vec{x}}$?

Definition 1 *A function is smooth if it is continuous and all orders of its derivative are continuous.*

Definition 2 *The Jacobian of f is the array of partial derivatives.*

$$J(\theta) \equiv \begin{pmatrix} \frac{\delta x_1}{\delta \theta_1} & \cdots & \frac{\delta x_1}{\delta \theta_m} \\ \vdots & & \vdots \\ \frac{\delta x_m}{\delta \theta_1} & \cdots & \frac{\delta x_m}{\delta \theta_m} \end{pmatrix} \tag{31}$$

$$\dot{\vec{x}} = J(\theta)\dot{\theta} \tag{32}$$

For manipulation, J relates cartesian linear and angular velocities to joint rates.

$$\begin{bmatrix} v \\ \omega \end{bmatrix} = J_m \dot{\theta} \tag{33}$$

Theorem 1 $\dot{\theta} = J^{-1}(\theta)\dot{x}$ *iff* J *is invertible.*

Many robots have configurations at which the Jacobian is singular. Singularities represent configurations for which certain directions of motion are unattainable. This acts like a loss of DoF. At singularities, bounded end-effector velocities correspond to unbounded joint velocities.

Consider our magnetic robot with four electromagnets, from equation 25 we get

$$\begin{bmatrix} B \\ F \end{bmatrix} = C(m,p)I \rightarrow \begin{bmatrix} \alpha\hat{h} \\ bv \end{bmatrix} = C(m,p)I \tag{34}$$

If $det(C(m,p) \neq 0$,

$$I = C^{-1}(m,p) \begin{bmatrix} \alpha\hat{h} \\ bv \end{bmatrix} \tag{35}$$

Otherwise, the robot is in a singularity and cannot be controlled with finite currents. As the C matrix becomes ill-conditioned, the coil currents will rapidly increase. Thus, we should always compute the determinant to ensure safe operation.

7.1 Example

Given a revolute-revolute manipulator,

$$v = \begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = J(\theta) \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix} \tag{36}$$

This robot has singularities at $\theta_2 = 0, \pi$.