EE16B
Designing Information Devices and Systems II

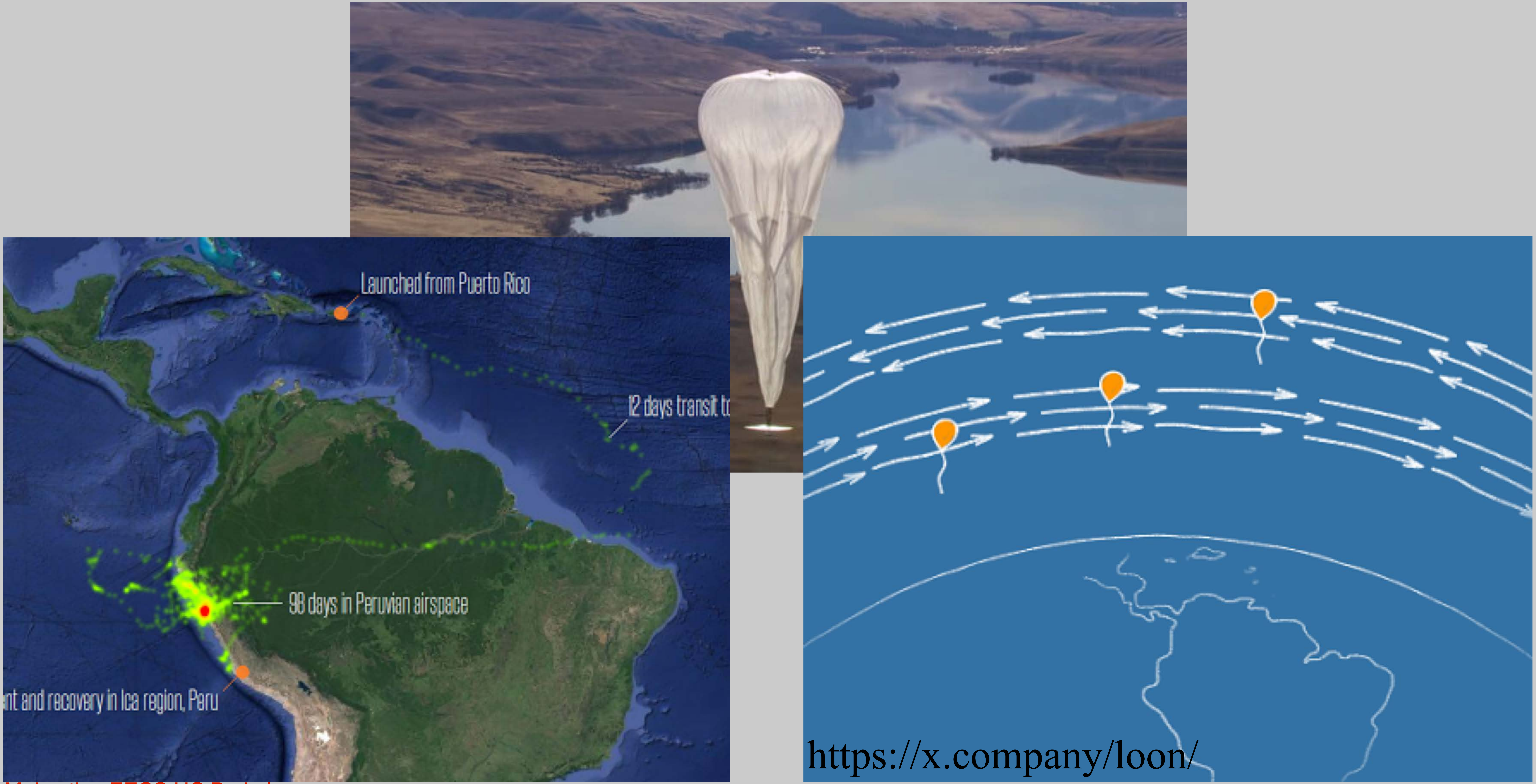
Lecture 8B
State Feedback Control Cont
System ID

Intro

- Last time
  - State feedback control
  - Eigen value asignment

- Today:
  - Continue state feedback control
  - Example of cooperative, adaptive cruise control
  - System ID

GOOGLE PROJECT LOON BALLOONS


Launched from Puerto Rico
12 days transit to
98 days in Peruvian airspace
nt and recovery in Ica region, Peru

https://x.company/loon/

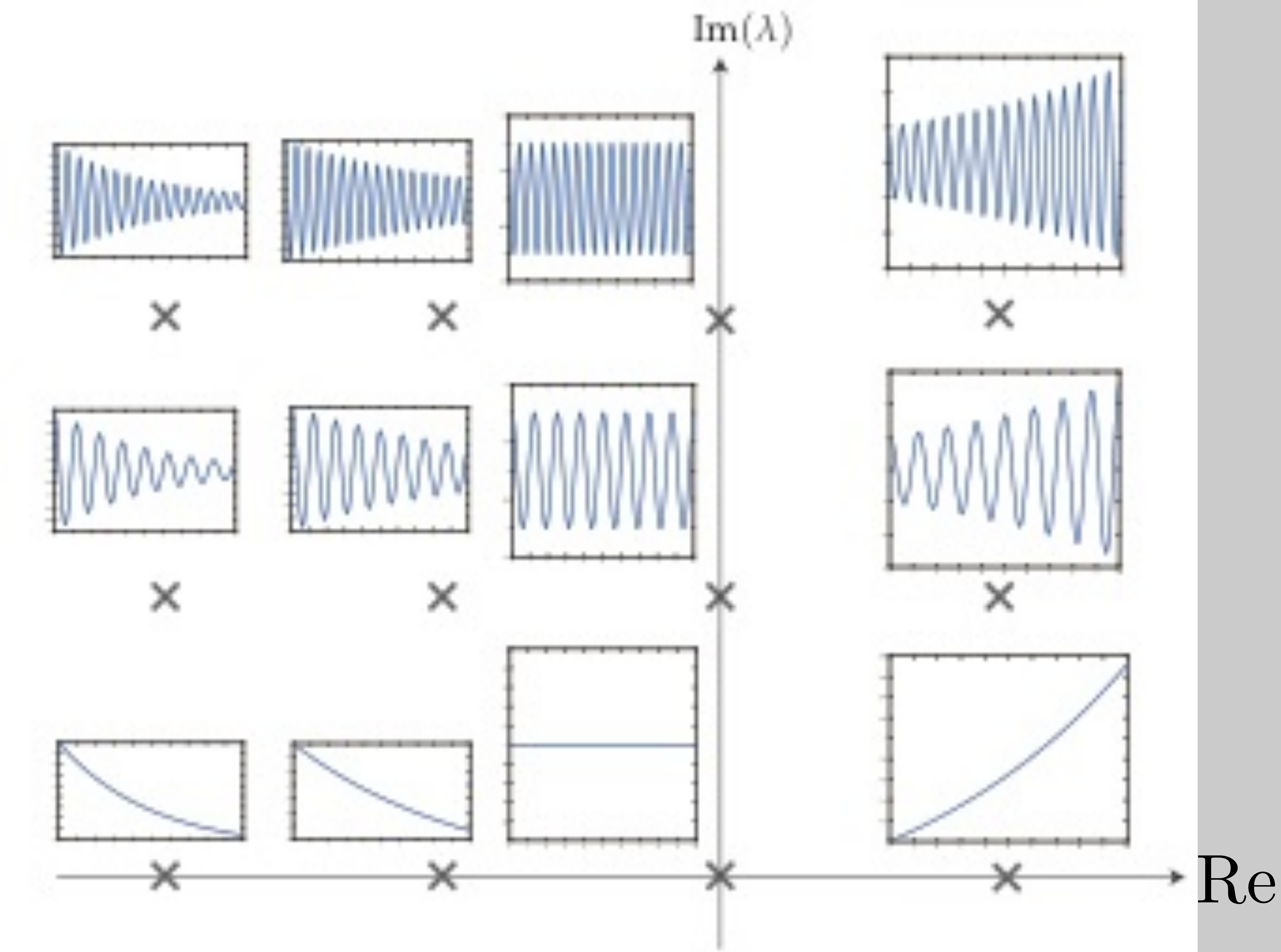
Continuous Time

$$\frac{d}{dt}\vec{x}(t) = A\vec{x}(t) + Bu(t)$$

$$u(t) = K\vec{x}(t)$$

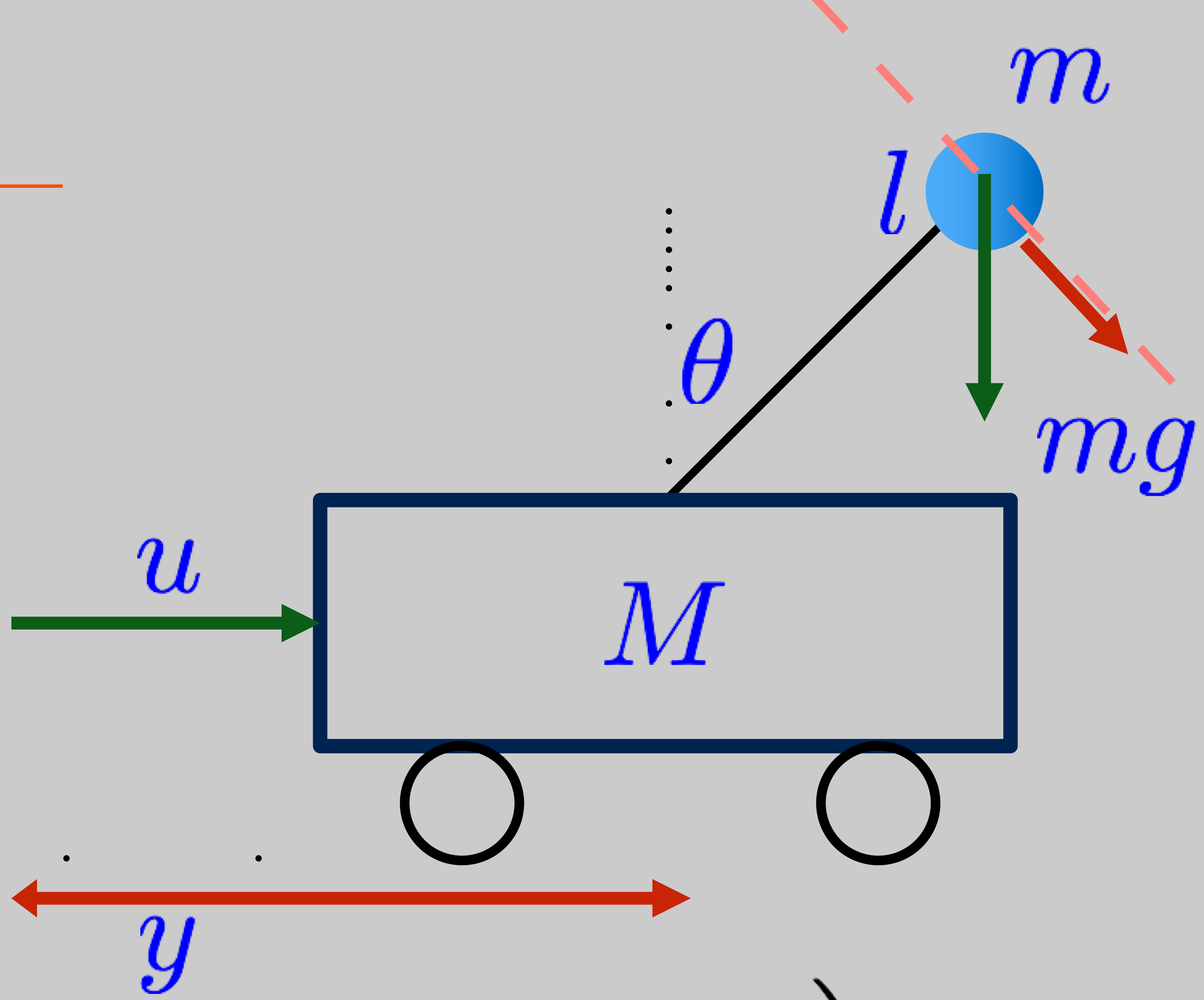
$$\frac{d}{dt}\vec{x}(t) = (A + BK)\vec{x}(t)$$

Choose K s.t. , Re λ_i(A+BK) < 0, i=1,2,3…n


Im(λ)
Re

Example 3: Pole on a Cart

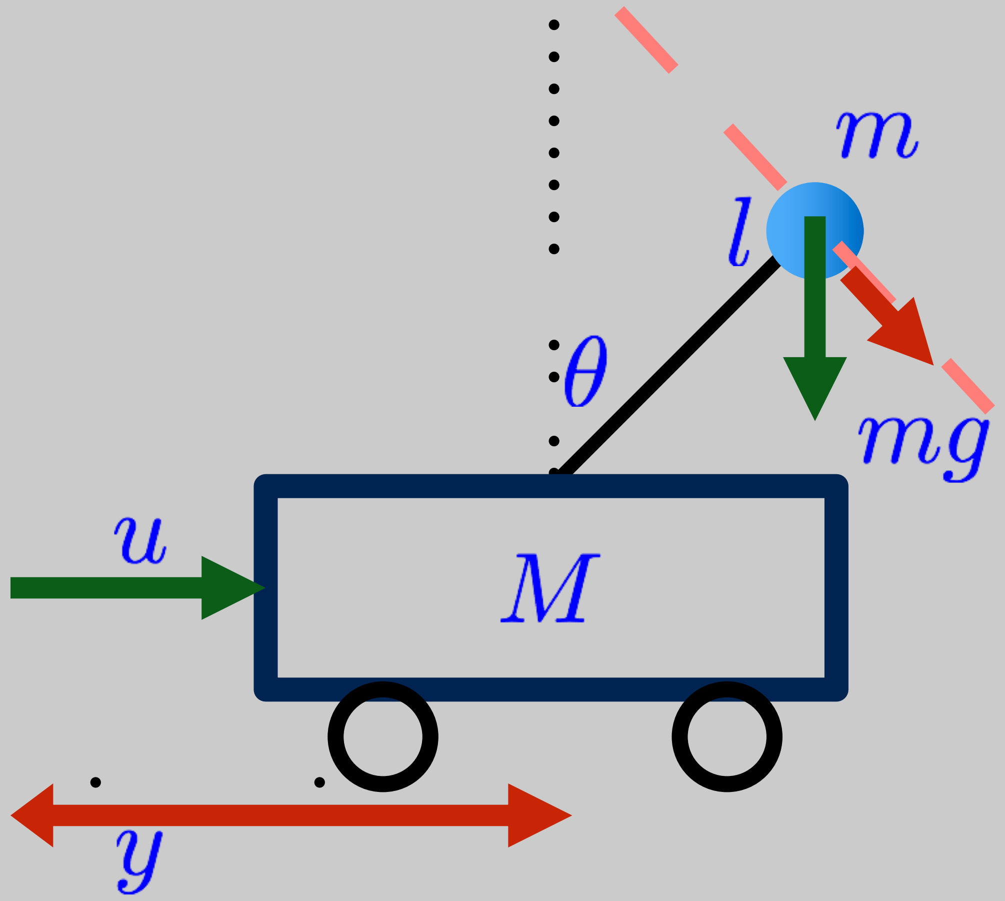
Design state-feedback control



$$\ddot{y} = \frac{1}{\frac{M}{m} + \sin^2\theta}\left(\frac{u}{m} + \dot{\theta}^2 l \sin\theta - g\sin\theta\cos\theta\right)$$

$$\ddot{\theta} = \frac{1}{l(\frac{M}{m} + \sin^2\theta)}\left(-\frac{u}{m}\cos\theta - \dot{\theta}^2 l\sin\theta\cos\theta + \frac{M+m}{m} g\sin\theta\right)$$

Example 3: Pole on a Cart



Linearization about $\theta = 0 \quad \dot{\theta} = 0$

State space model:

$$\frac{d}{dt}\begin{bmatrix} \theta(t) \\ \dot{\theta}(t) \\ \dot{y}(t) \end{bmatrix} = \begin{bmatrix} \frac{M+m}{Ml}g & 0 & 0 \\ -\frac{m}{M}g & 0 & 0 \end{bmatrix}$$

$$\frac{d}{dt}\begin{bmatrix} \theta(t) \\ \dot{\theta}(t) \\ \dot{y}(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ \frac{M+m}{Ml}g & 0 & 0 \\ -\frac{m}{M}g & 0 & 0 \end{bmatrix}\begin{bmatrix} \theta(t) \\ \dot{\theta}(t) \\ \dot{y}(t) \end{bmatrix} + \begin{bmatrix} 0 \\ -\frac{1}{Ml} \\ \frac{1}{M} \end{bmatrix} u(t)$$

$$M = 1$$
$$m = 0.1$$
$$l = 1$$
$$g = 1$$

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 11 & 0 & 0 \\ -1 & 0 & 0 \end{bmatrix} \qquad B = \begin{bmatrix} 0 \\ -1 \\ 1 \end{bmatrix}$$

Controller

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 11 & 0 & 0 \\ -1 & 0 & 0 \end{bmatrix} \quad B = \begin{bmatrix} 0 \\ -1 \\ 1 \end{bmatrix}$$

$$M = 1$$
$$m = 0.1$$
$$l = 1$$
$$g = 1$$

$$u(t) = k_1\theta(t) + k_2\dot{\theta}(t) + k_3\dot{y}(t)$$

$$A + BK = \begin{bmatrix} 0 & 1 & 0 \\ 11 - k_1 & -k_2 & -k_3 \\ -1 + k_1 & k_2 & k_3 \end{bmatrix}$$

Characteristic polynomial:

$$\lambda^3 + (k_2 - k_3)\lambda^2 + (k_1 - 11)\lambda + 10k_3 = 0$$

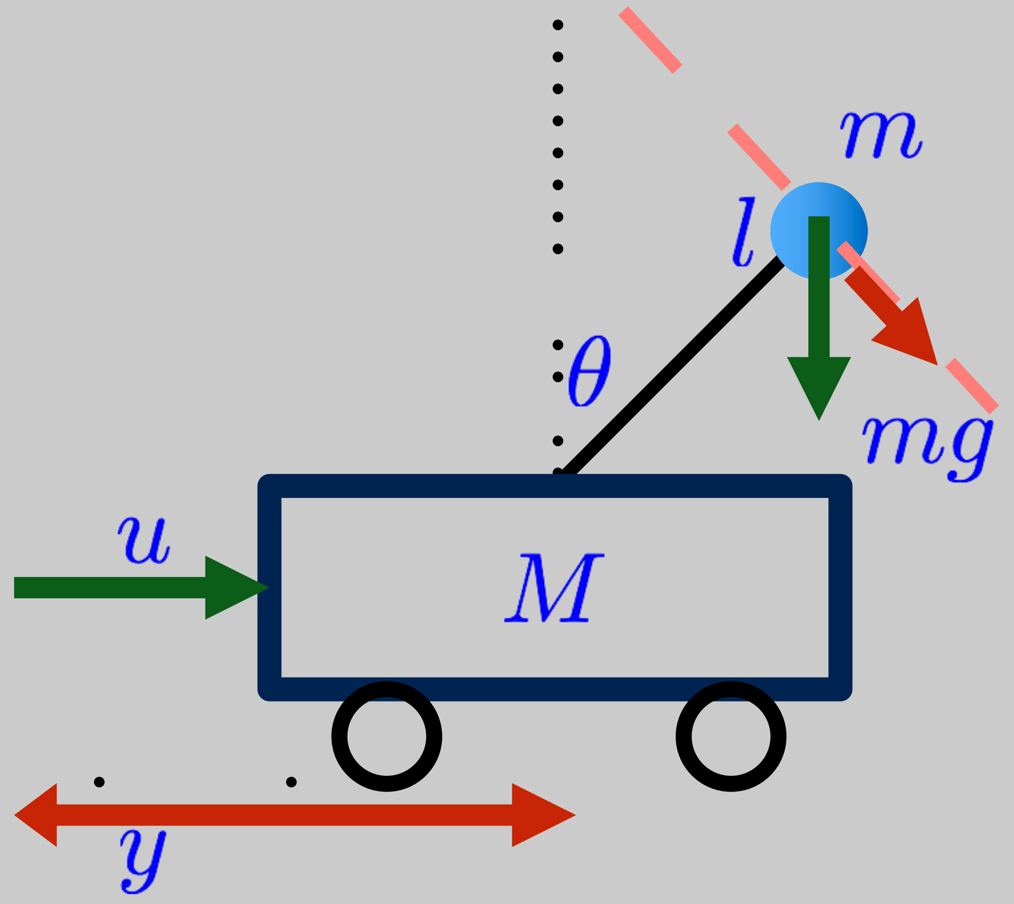
$$\Updownarrow$$

Desired: $\lambda_1, \lambda_2, \lambda_3$

$$(\lambda - \lambda_1)(\lambda - \lambda_2)(\lambda - \lambda_3)$$

Match coeff.

$$A + BK = \begin{bmatrix} 0 & 1 & 0 \\ 11 - k_1 & -k_2 & -k_3 \\ -1 + k_1 & k_2 & k_3 \end{bmatrix}$$



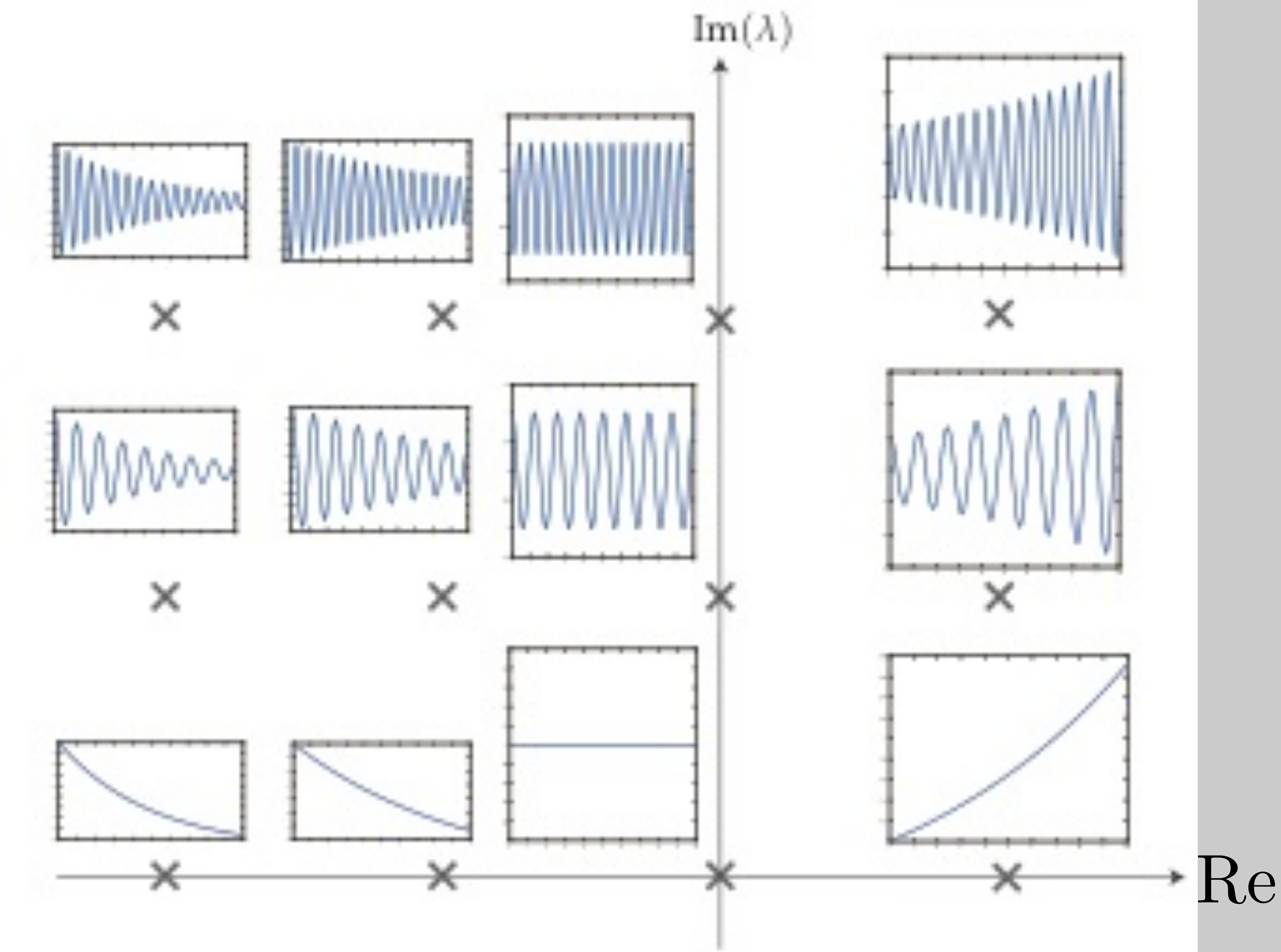
$$\lambda^3 + (k_2 - k_3)\lambda^2 + (k_1 - 11)\lambda + 10k_3 = 0$$

$$\lambda^3 - (\lambda_1 + \lambda_2 + \lambda_3)\lambda^2 - (\lambda_1\lambda_2 + \lambda_1\lambda_3 + \lambda_2\lambda_3)\lambda - \lambda_1\lambda_2\lambda_3 = 0$$

$$\begin{bmatrix} 0 & 1 & -1 \\ 1 & 0 & 0 \\ 0 & 0 & 10 \end{bmatrix} \begin{bmatrix} k_1 \\ k_2 \\ k_3 \end{bmatrix} = \begin{bmatrix} -(\lambda_1 + \lambda_2 + \lambda_3) \\ -(\lambda_1\lambda_2 + \lambda_1\lambda_3 + \lambda_2\lambda_3) + 11 \\ \lambda_1\lambda_2\lambda_3 \end{bmatrix}$$

inv

$$\begin{bmatrix} k_1 \\ k_2 \\ k_3 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0.1 \\ 0 & 0 & 0.1 \end{bmatrix} \begin{bmatrix} -(\lambda_1 + \lambda_2 + \lambda_3) \\ -(\lambda_1\lambda_2 + \lambda_1\lambda_3 + \lambda_2\lambda_3) + 11 \\ \text{-}\lambda_1\lambda_2\lambda_3 \end{bmatrix}$$


Im(λ)
Re

Controller

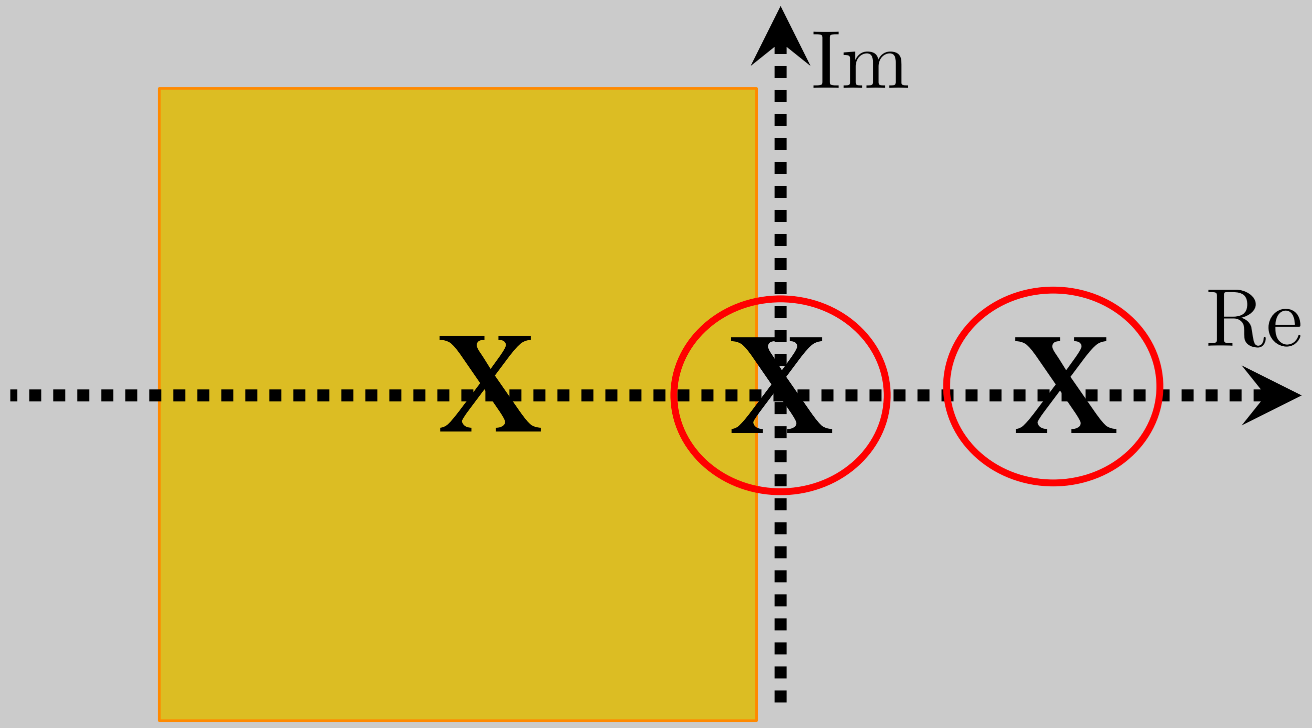
What is open loop? (no feedback control, k_i=0):

$$\lambda^3 + (k_2 - k_3)\lambda^2 + (k_1 - 11)\lambda + 10k_3 = 0$$

$$\lambda^3 - 11\lambda = 0$$

$$\lambda(\lambda^2 - 11) = 0$$

Ask yourself what if you can control just one, or two state variables?

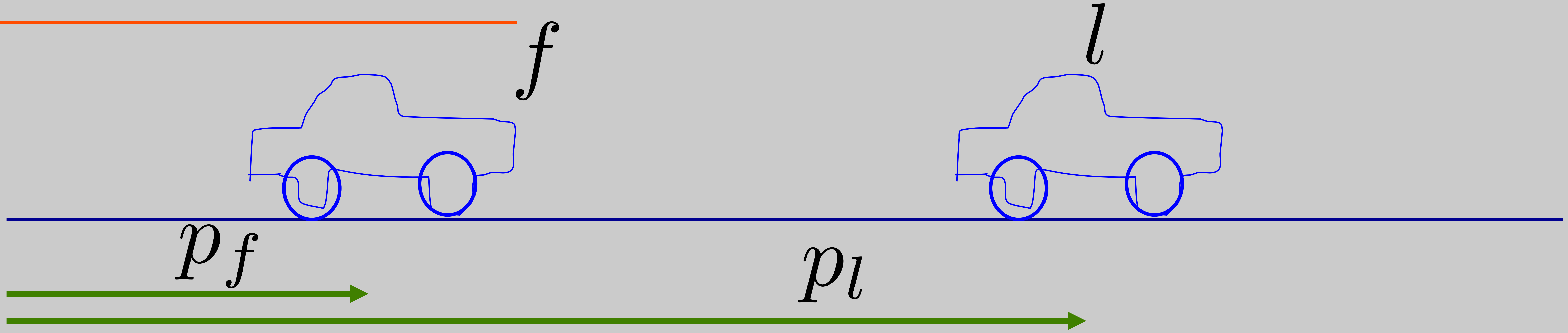


Controller moves bad eigen-values left!

PLATOONING

Cooperative Adaptive Cruise control

Example:



$$\frac{d}{dt}p_l(t) = v_l(t) \qquad \frac{d}{dt}p_f(t) = v_f(t)$$

$$\frac{d}{dt}v_l(t) = u_l(t) \qquad \frac{d}{dt}v_f(t) = u_f(t)$$

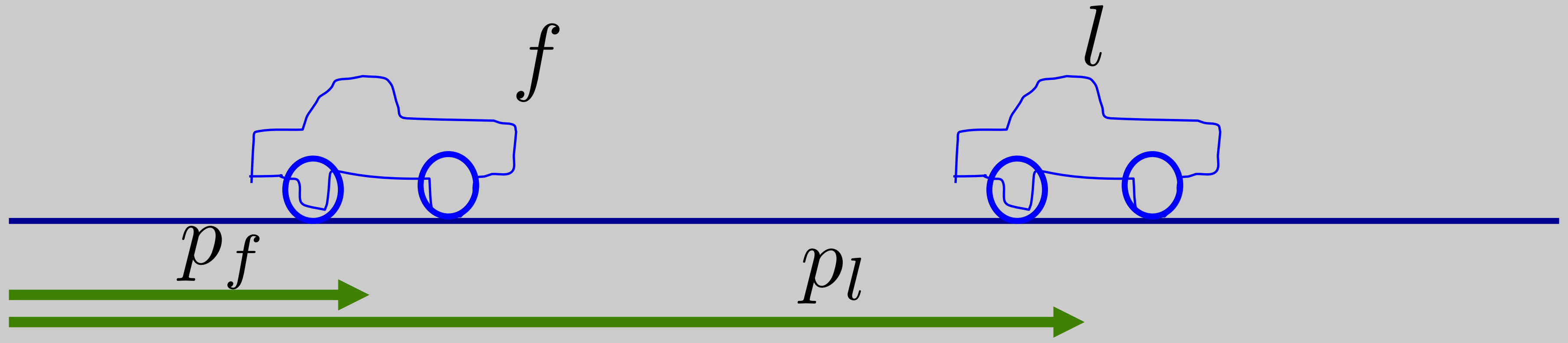
$$x_1(t) = p_l(t) - p_f(t) - \delta$$

$$x_2(t) = v_l(t) - v_f(t)$$

$$\frac{d}{dt}x_1(t) = v_l(t) - v_f(t)$$

$$\frac{d}{dt}x_2(t) = u_l(t) - u_f(t) \triangleq u(t)$$

Cooperative Adaptive Cruise control



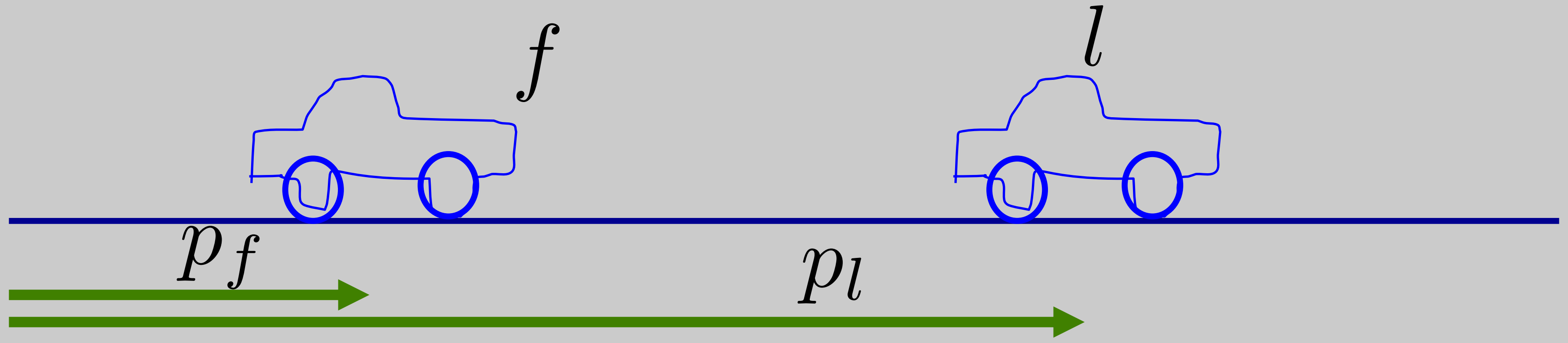
$$\frac{d}{dt}x_1(t) = v_l(t) - v_f(t)$$

$$\frac{d}{dt}x_2(t) = u_l(t) - u_f(t) \triangleq u(t)$$

$$A = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \qquad B = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \qquad \Rightarrow A + BK = \begin{bmatrix} 0 & 1 \\ k_1 & k_2 \end{bmatrix}$$

Q) What eigen-values will you want here?

Let's look at input more closely…



$$u(t) = k_1 x_1(t) + k_2 x_2(t)$$

$$\Rightarrow u(t) = k_1(p_l(t) - p_f(t) - \delta) + k_2(v_l(t) - v_f(t))$$

But leader chooses his own acceleration u_l(t)

$$u(t) = u_l(t) - u_f(t)$$

$$\begin{aligned} u_f(t) &= u_l(t) - u(t) \\ &= u_l(t) - k_1(p_l(t) - p_f(t) - \delta) - k_2(v_l(t) - v_f(t)) \end{aligned}$$

Q) What does the follower need to know to implement?

A) Cooperative (vehicle2vehicle comm.)
range sensor (for distance and velocity)

Outputs

$$\vec{x}(t+1) = A\vec{x}(t) + Bu(t)$$

Can’t always measure state directly or all states…

Define output:

$$\vec{y}(t) = C\vec{x}(t)$$

p x n matrix for p outputs

Outputs

$$\vec{x}(t+1) = A\vec{x}(t) + Bu(t)$$

Can’t always measure state directly or all states…

Define output:

$$\vec{y}(t) = C\vec{x}(t)$$

p x n matrix for p outputs

$$y = x_1 \quad \Rightarrow C = [1\ 0\ 0\ \cdots\ 0]$$

$$y = x_1 + x_2 \quad \Rightarrow C = [1\ 1\ 0\ \cdots\ 0]$$

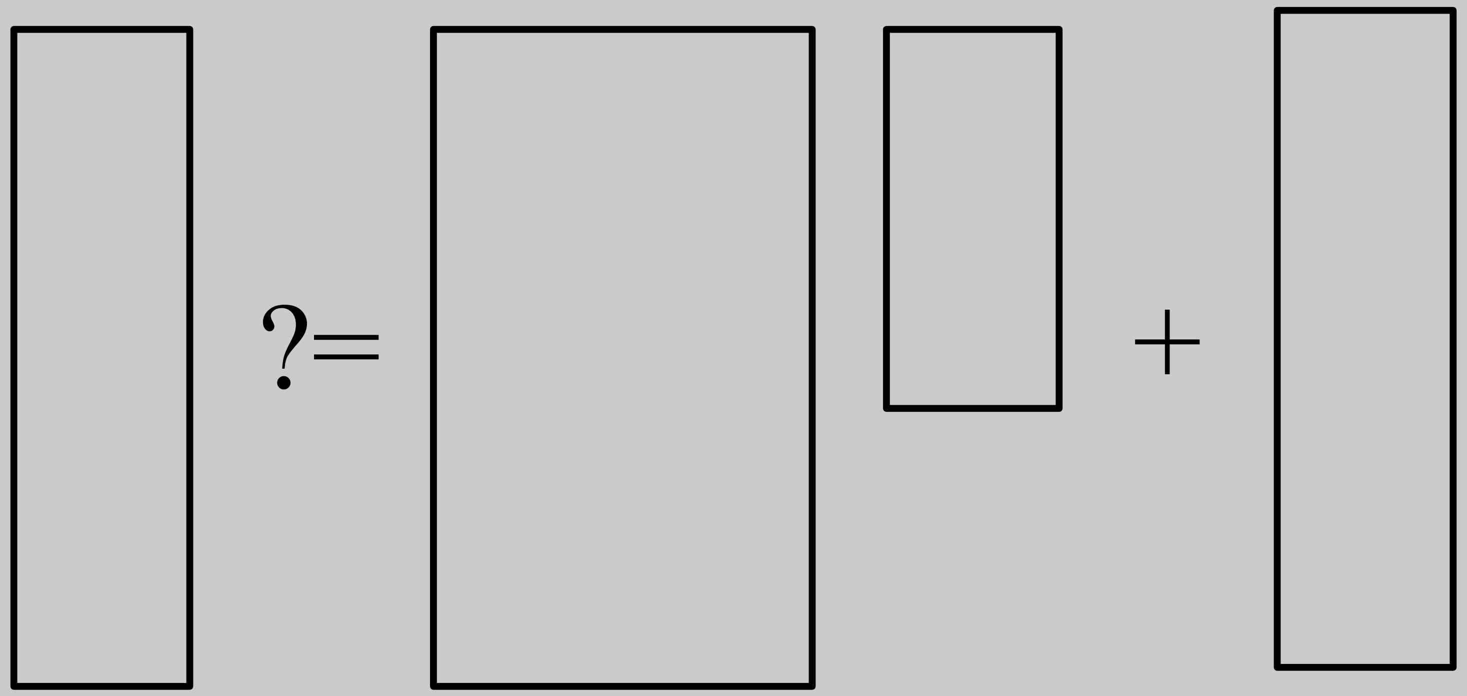
$$\vec{y} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \quad \Rightarrow C = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \end{bmatrix}$$

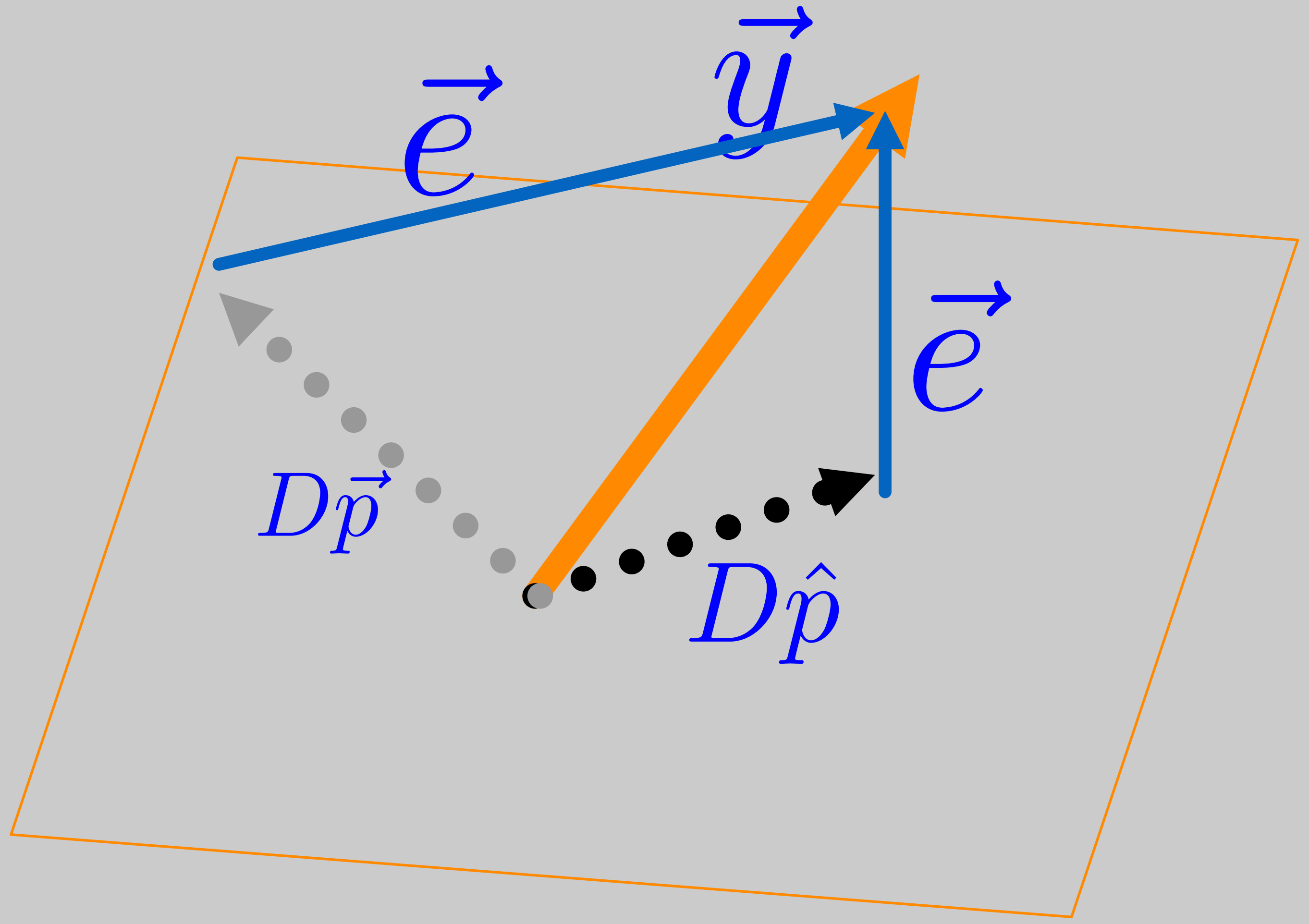
$$\vec{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

Least Squares

$$\vec{y} = D\vec{p} + \vec{e}$$

$p \in \mathcal{R}^k \qquad y \in \mathcal{R}^l$





Least Squares, minimizes $||\vec{e}||$, which is orthogonal to columns of D

$$D^T\vec{e} = 0 \quad \Rightarrow D^T(\vec{y} - D\hat{p}) = 0 \quad \Rightarrow D^T\vec{y} - D^T D\hat{p} = 0$$

Least Squares

$$\vec{y} = D\vec{p} + \vec{e}$$

$$\Rightarrow D^T\vec{y} - D^T D\hat{p} = 0$$

$$D^T D\hat{p} = D^T\vec{y}$$

If $D^T D$ is invertible,

$$\hat{p} = (D^T D)^{-1} D^T\vec{y}$$

Scalar System ID

$$x(t+1) = \lambda x(t) + bu(t) + w(t) \qquad y(t) = 1 \cdot x(t)$$

$$x(1) = \lambda x(0) + bu(0) + w(0), \quad y(1) = x(1)$$

$$x(2) = \lambda x(1) + bu(1) + w(1), \quad y(2) = x(2)$$

$$\vdots$$

$$x(l) = \lambda x(l-1) + bu(l-1) + w(l-1), \quad y(l) = x(l)$$

$$\text{argmin}_{\lambda,b} \;\; ||\vec{w}|| \;\Rightarrow\; \text{argmin}_{\vec{p}} \left\| \underbrace{\begin{bmatrix} \quad \\ \quad \\ \quad \end{bmatrix}}_{D} \underbrace{\begin{bmatrix} \lambda \\ b \end{bmatrix}}_{\vec{p}} - \underbrace{\begin{bmatrix} \quad \\ \quad \\ \quad \end{bmatrix}}_{\vec{y}} \right\|$$

Scalar System ID

$$\Rightarrow \operatorname{argmin}_{\vec{p}} \left\| \underset{D}{\begin{bmatrix} x(0) & u(0) \\ x(1) & (1) \\ \vdots & \\ x(l) & u(l) \end{bmatrix}} \underset{\vec{p}}{\begin{bmatrix} \lambda \\ b \end{bmatrix}} - \underset{\vec{y}}{\begin{bmatrix} y(1) \\ y(2) \\ \vdots \\ y(l) \end{bmatrix}} \right\|$$

Least Squares Estimate:

$$\hat{p} = (D^T D)^{-1} D^T \vec{y}$$

2x2 Matrix System ID

$$\vec{x}(t+1) = A\vec{x}(t) + Bu(t) + \vec{w}(t)$$

$$\begin{bmatrix} \boxed{y_0(t+1)} \\ \boxed{y_1(t+1)} \end{bmatrix} \approx \begin{bmatrix} \boxed{a_{00} \quad a_{01}} \\ \boxed{a_{10} \quad a_{11}} \end{bmatrix} \boxed{\begin{bmatrix} x_0(t) \\ x_1(t) \end{bmatrix}} + \begin{bmatrix} \boxed{b_0 u(t)} \\ \boxed{b_1 u(t)} \end{bmatrix}$$

$$\begin{bmatrix} \quad \\ \quad \\ \quad \end{bmatrix} \begin{bmatrix} a_{00} \\ a_{01} \\ b_0 \end{bmatrix} = \begin{bmatrix} \quad \\ \quad \\ \quad \end{bmatrix} \qquad \begin{bmatrix} \quad \\ \quad \\ \quad \end{bmatrix} \begin{bmatrix} a_{10} \\ a_{11} \\ b_1 \end{bmatrix} = \begin{bmatrix} \quad \\ \quad \\ \quad \end{bmatrix}$$

2x2 Matrix System ID

$$\underset{D}{\begin{bmatrix} x_0(0) & x_1(0) & u(0) \\ x_0(1) & x_1(1) & u(1) \\ \vdots & & \\ x_0(l) & x_1(l) & u(l) \end{bmatrix}} \underset{\vec{p}_0}{\begin{bmatrix} a_{00} \\ a_{01} \\ b_0 \end{bmatrix}} \approx \underset{\vec{y}_0}{\begin{bmatrix} y_0(1) \\ y_0(2) \\ \vdots \\ y(l) \end{bmatrix}}$$

$$\underset{D}{\begin{bmatrix} x_0(0) & x_1(0) & u(0) \\ x_0(1) & x_1(1) & u(1) \\ \vdots & & \\ x_0(l) & x_1(l) & u(l) \end{bmatrix}} \underset{\vec{p}_1}{\begin{bmatrix} a_{10} \\ a_{11} \\ b_1 \end{bmatrix}} \approx \underset{\vec{y}_0}{\begin{bmatrix} y_1(1) \\ y_1(2) \\ \vdots \\ y(l) \end{bmatrix}}$$

$$\hat{p}_0 = (D^T D)^{-1} D^T \vec{y}_0$$

$$\hat{p}_1 = (D^T D)^{-1} D^T \vec{y}_1$$

General Matrix System ID

$$\underset{D}{\begin{bmatrix} x_0(0) & x_1(0) & u(0) \\ x_0(1) & x_1(1) & u(1) \\ \vdots & & \\ x_0(l) & x_1(l) & u(l) \end{bmatrix}} \underset{\vec{p}_0}{\begin{bmatrix} a_{00} \\ a_{01} \\ b_0 \end{bmatrix}} \approx \underset{\vec{y}_0}{\begin{bmatrix} y_0(1) \\ y_0(2) \\ \vdots \\ y(l) \end{bmatrix}} \qquad \underset{D}{\begin{bmatrix} x_0(0) & x_1(0) & u(0) \\ x_0(1) & x_1(1) & u(1) \\ \vdots & & \\ x_0(l) & x_1(l) & u(l) \end{bmatrix}} \underset{\vec{p}_1}{\begin{bmatrix} a_{10} \\ a_{11} \\ b_1 \end{bmatrix}} \approx \underset{\vec{y}_0}{\begin{bmatrix} y_1(1) \\ y_1(2) \\ \vdots \\ y(l) \end{bmatrix}}$$

$$\begin{bmatrix} x_0(0) & x_1(0) & u(0) \\ x_0(1) & x_1(1) & u(1) \\ \vdots & & \\ x_0(l) & x_1(l) & u(l) \end{bmatrix} \left[\begin{bmatrix} a_{00} \\ a_{01} \\ b_0 \end{bmatrix} \begin{bmatrix} a_{10} \\ a_{11} \\ b_1 \end{bmatrix}\right] \approx \left[\begin{bmatrix} y_0(1) \\ y_0(2) \\ \vdots \\ y(l) \end{bmatrix} \begin{bmatrix} y_1(1) \\ y_1(2) \\ \vdots \\ y(l) \end{bmatrix}\right]$$

$$\underset{D}{\begin{bmatrix} \vec{x}^T(0) & u(0) \\ \vec{x}^T(1) & u(1) \\ \vdots & \\ \vec{x}^T(l) & u(l) \end{bmatrix}} \underset{P}{\begin{bmatrix} A^T \\ B^T \end{bmatrix}} = \underset{Y}{\begin{bmatrix} \vec{y}^T(0) \\ \vec{y}^T(1) \\ \vdots \\ \vec{y}^T(l) \end{bmatrix}} \Rightarrow \hat{P} = (D^T D)^{-1} D^T \vec{Y}$$

Control Recap

- Controllability:

$$\vec{x}(n) - A^n\vec{x}(0) = \begin{bmatrix} A^{n-1}B & A^{n-2}B & \cdots & AB & B \end{bmatrix} \begin{bmatrix} u(0) \\ u(1) \\ \vdots \\ u(n-1) \end{bmatrix}$$

If Rn is full rank then we can move to any target value

Same rank test for continuous time

- Open loop control:

Can use the above equation to design an input sequence – and apply it blindly. Accuracy of result will depend on accuracy of model.

Control Recap – State Feedback

$$u(t) = K\vec{x}(t)$$

Closed-loop system: $\Rightarrow \quad \vec{x}(t+1) = (A + BK)\vec{x}(t)$

Must choose K s.t. A+BK has eigenvalues inside the unit circle (or left half-plane for continuous time)

If controllable, can assign eigenvalues for A+BK arbitrarily

If not, some eigenvalues of A can not be changed!

(could be OK, if stable, bad news if not)

System Identification

Given a system:

$$\vec{x}(t+1) = A\vec{x}(t) + Bu(t)$$

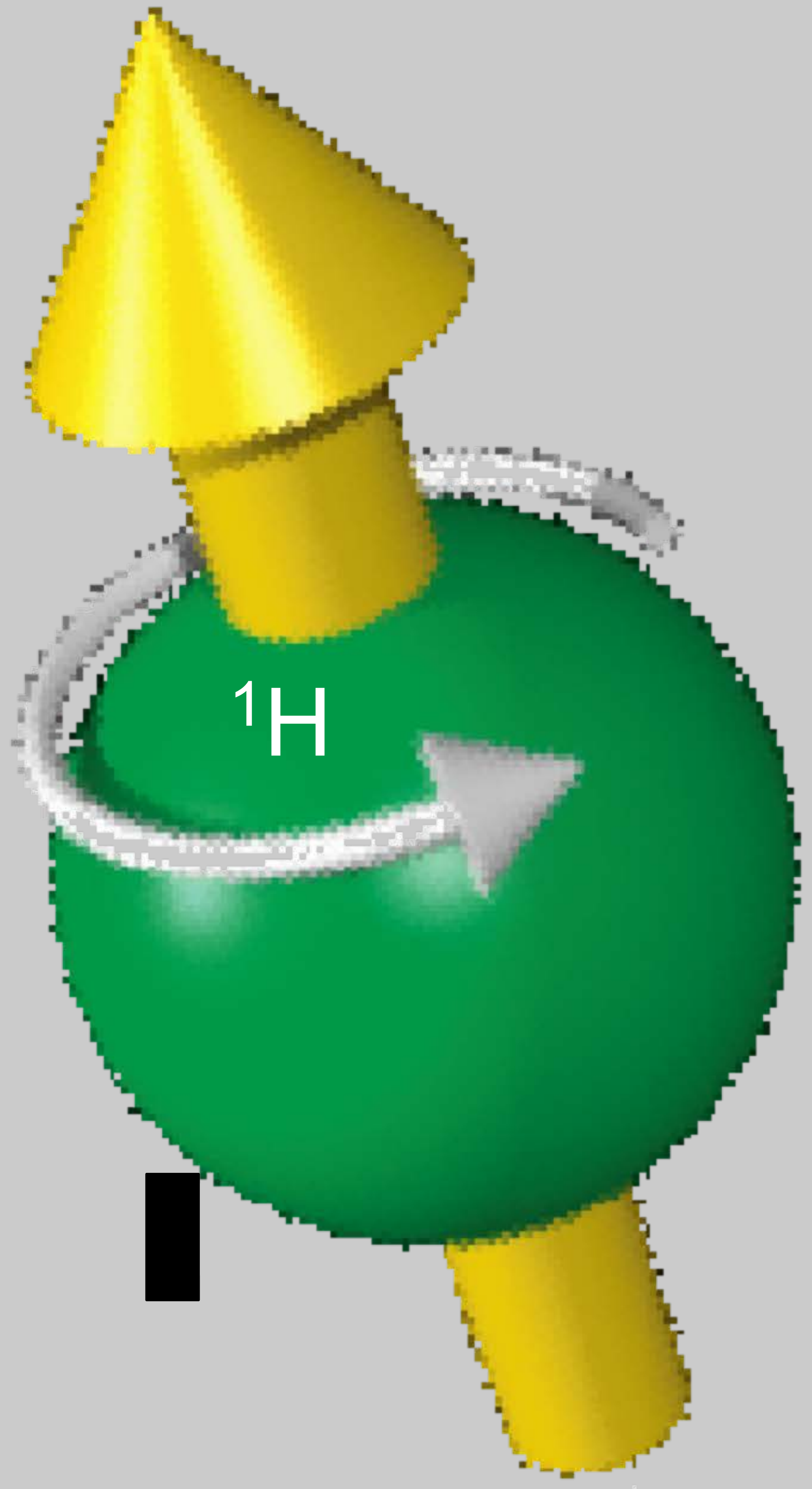
Can we learn A, and B by applying inputs and observing resulting states?

How Does MRI Work? (some today – more later!)

- Magnetic Polarization
  -- Very strong uniform magnet
- Excitation
  -- Very powerful RF transmitter
- Acquisition
  -- Location is encoded by gradient magnetic fields
  -- Very powerful audio amps

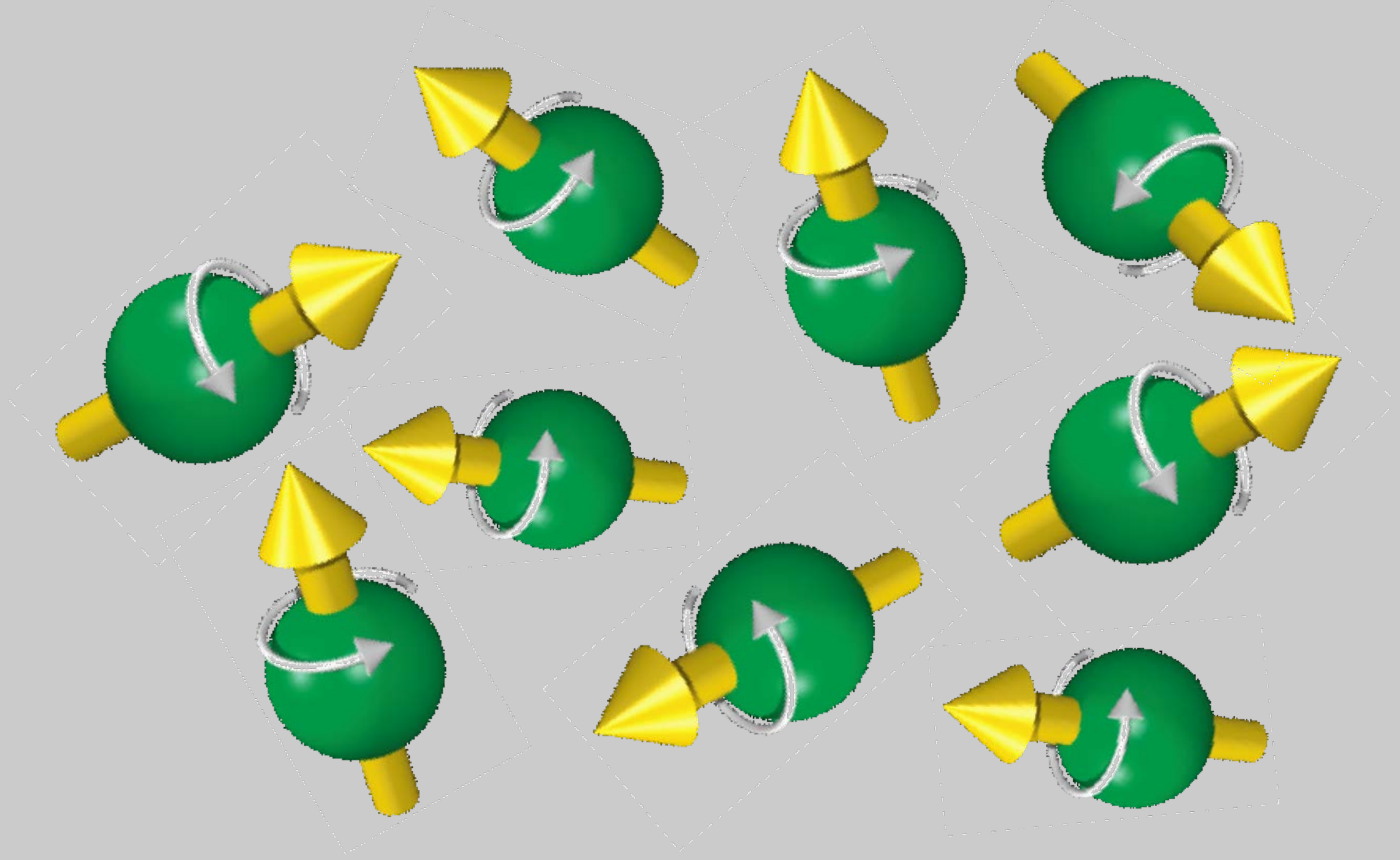
Polarization

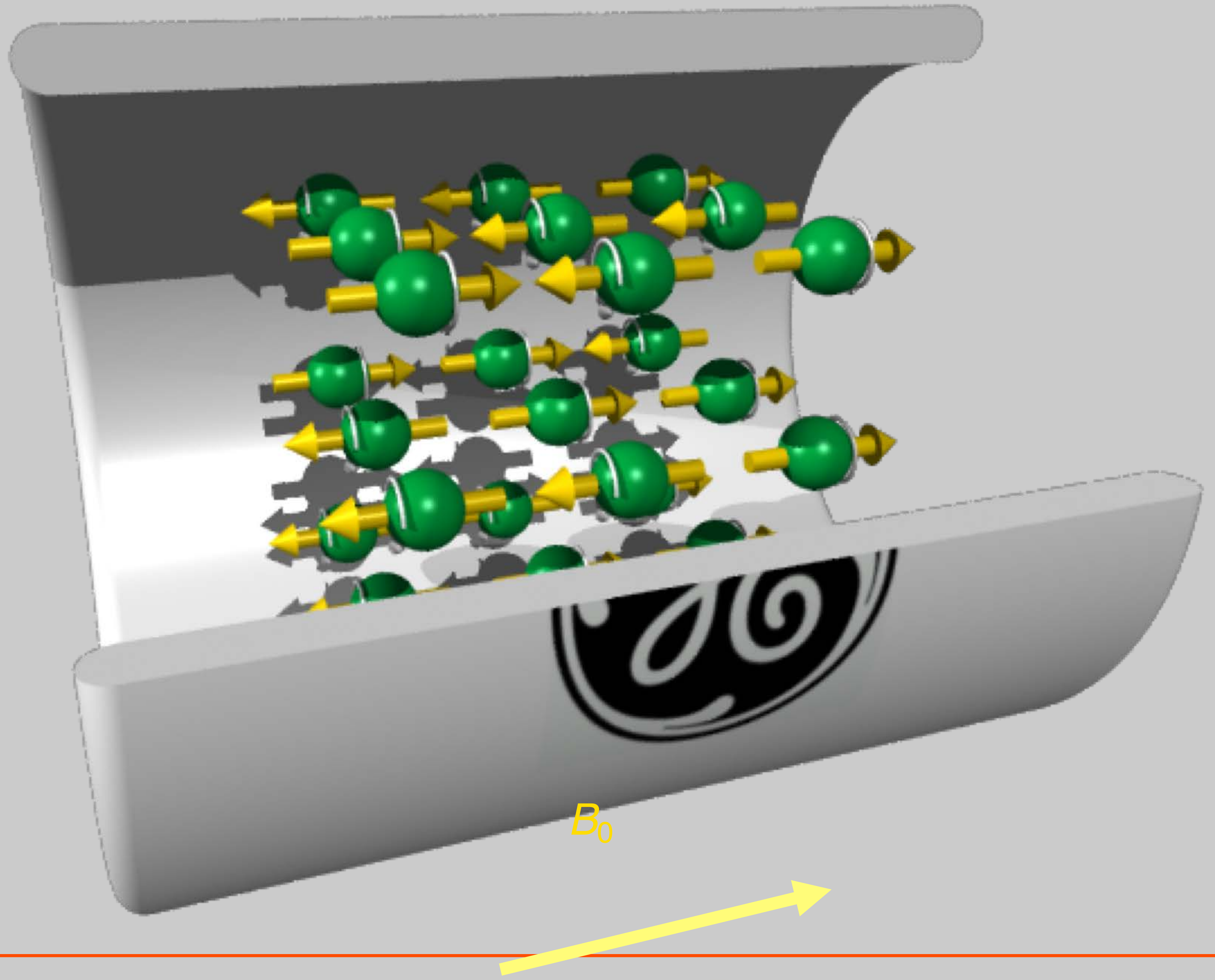
- Protons have a magnetic moment
- Protons have spins
- Like rotating magnets



Polarization

- Body has a lot of protons
- In a strong magnetic field B0, spins align with B0 giving a net magnetization

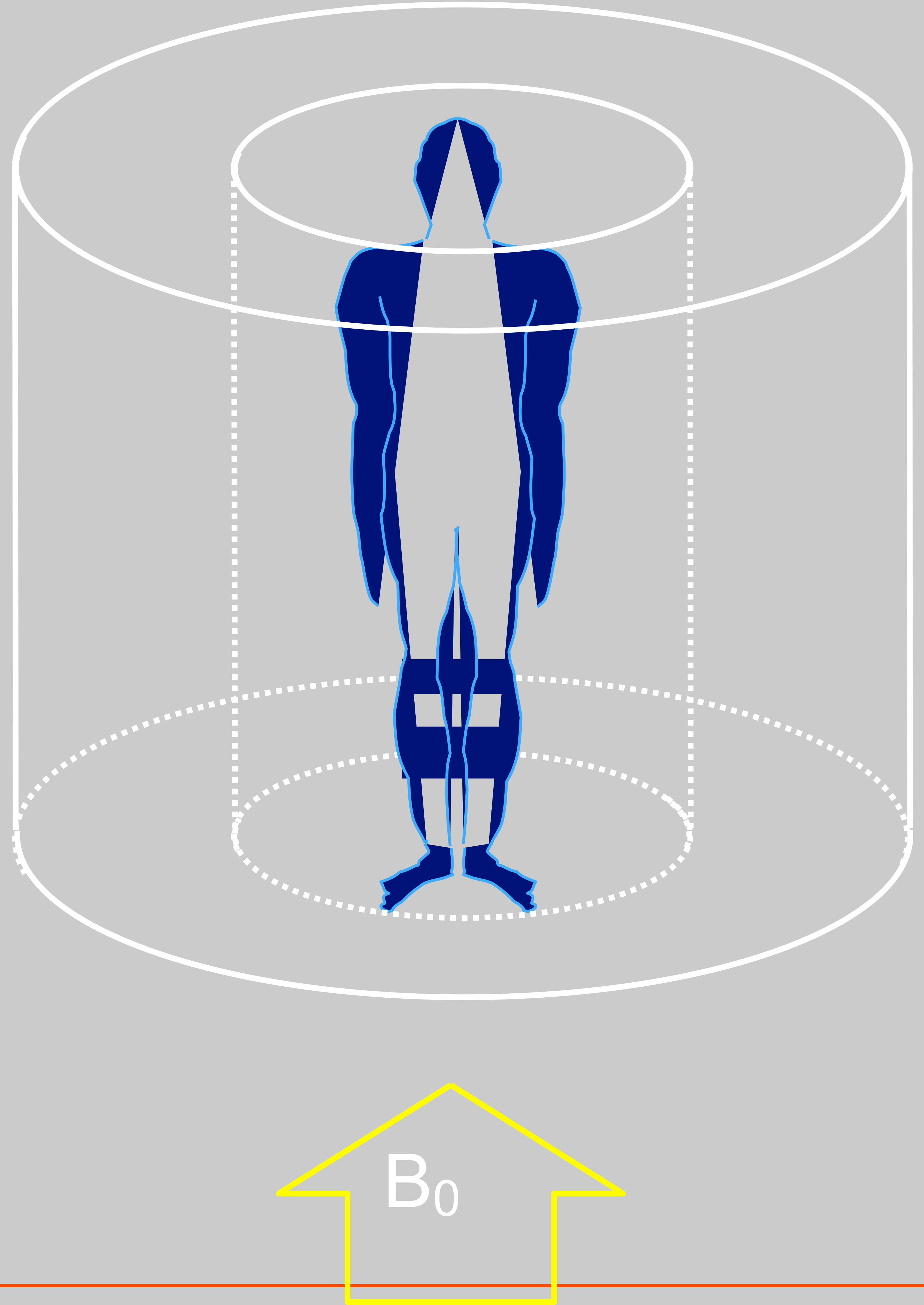




*Graphic rendering Bill Overall

Polarizing Magnet

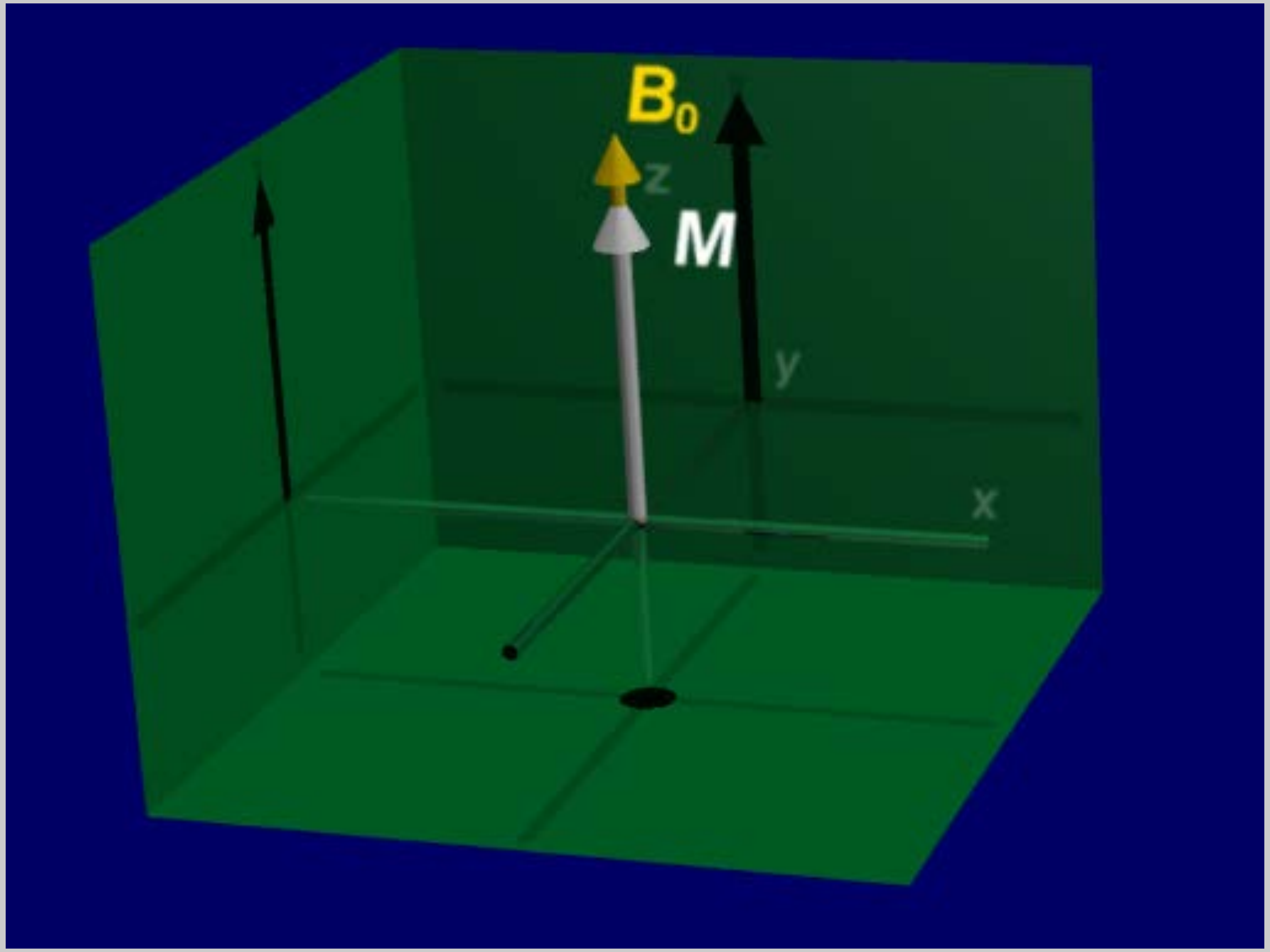
- 0.1 to 12 Tesla
- 0.5 to 3 T common
- 1 T is 10,000 Gauss
- Earth's field is 0.5G
- Typically a superconducting magnet

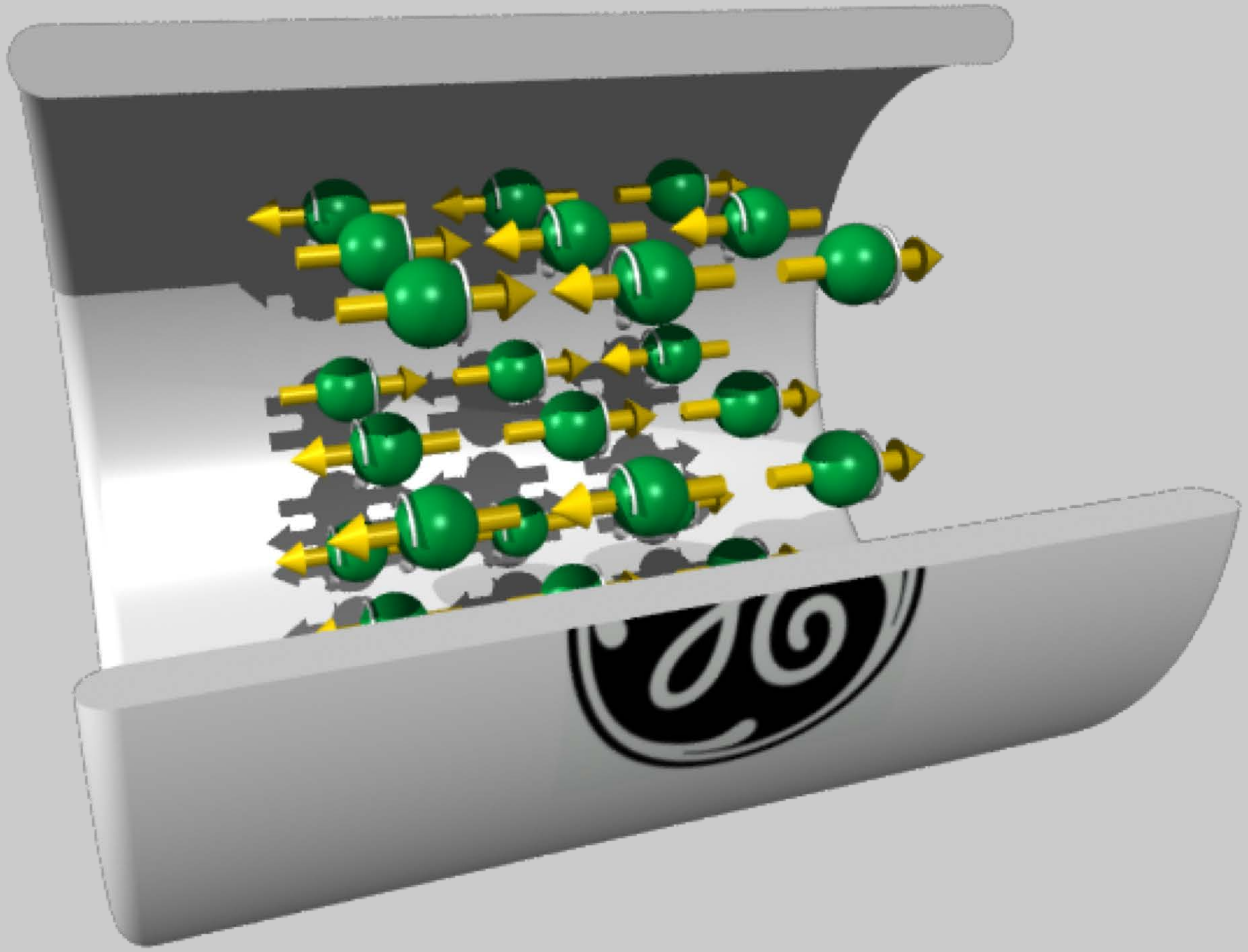


Typical MRI Scanner

A magnetic object

Polarizaion

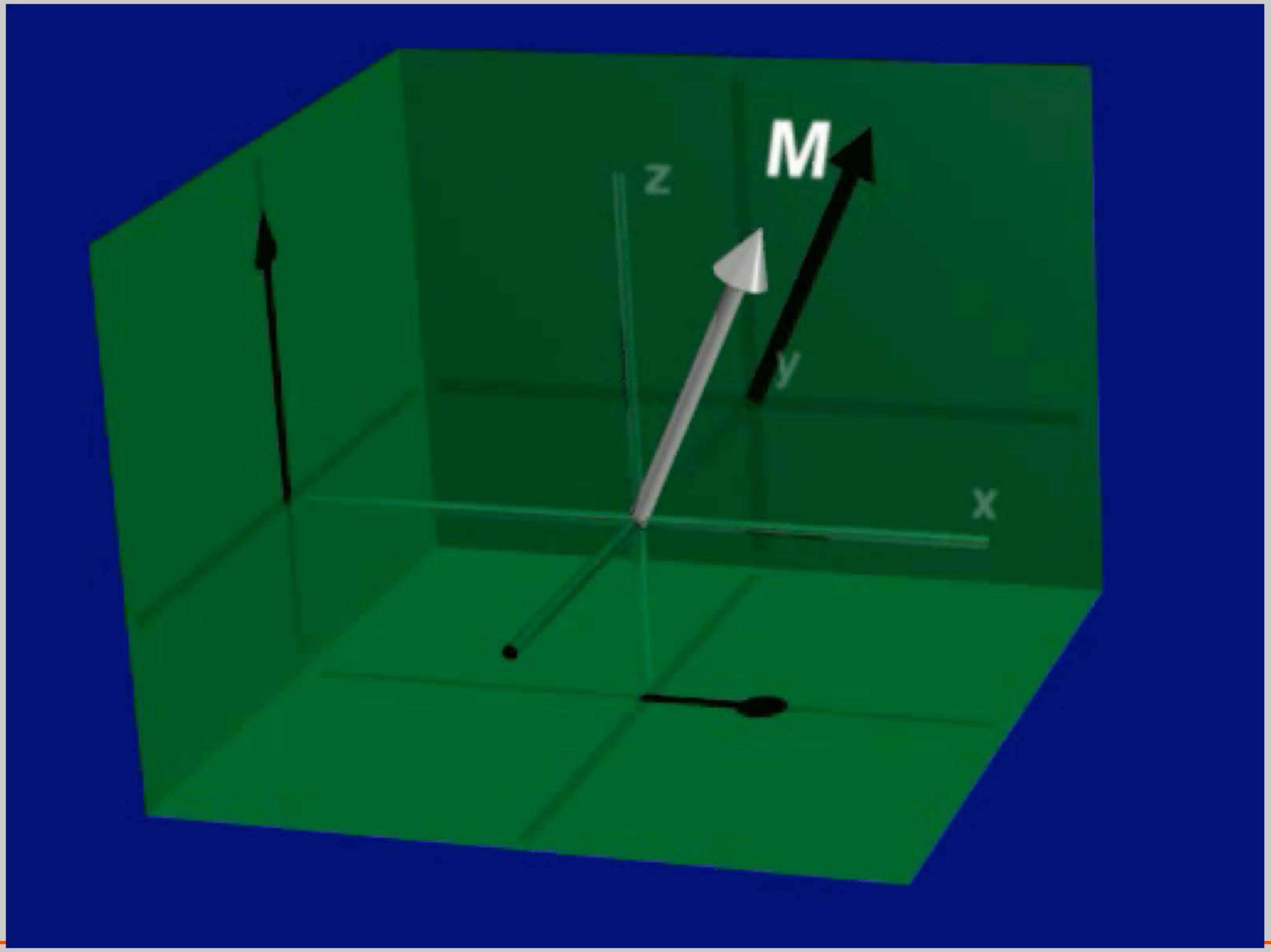




Free Precession

- Much like a spinning top
- Frequency proportional to the field
- f = 127MhZ @ 3T

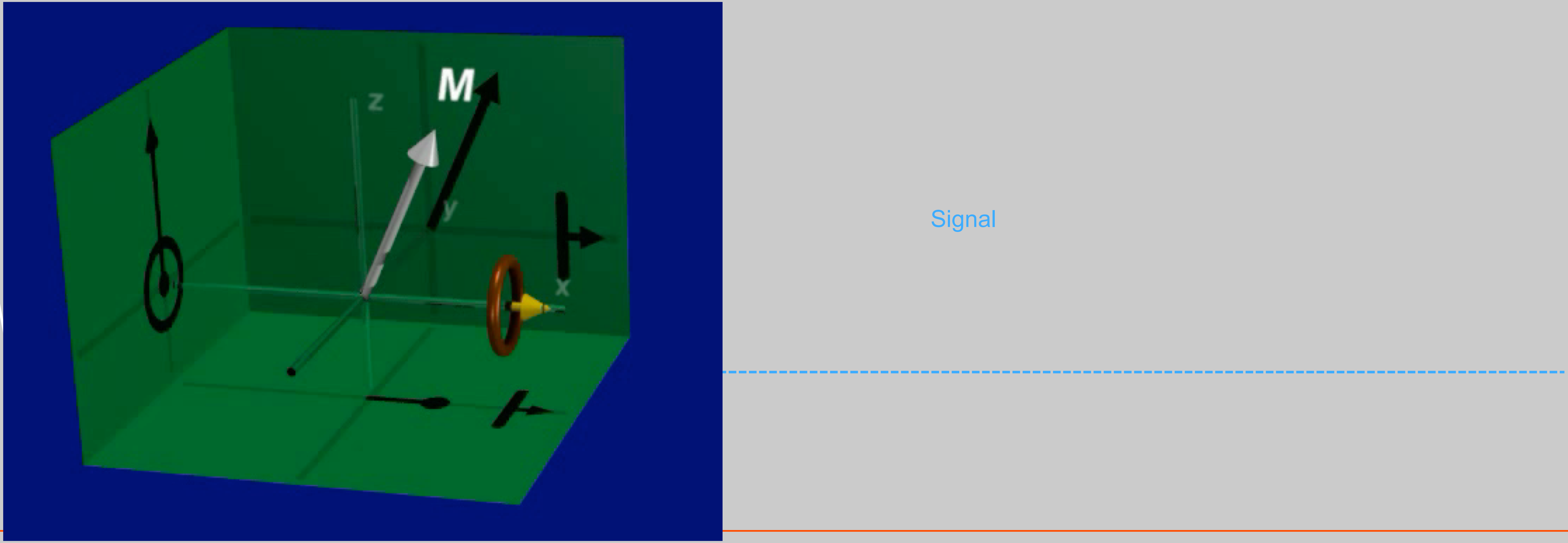
$$\frac{d\vec{M}}{dt} = \vec{M} \times \gamma \vec{B}$$



MIT physics demos

Free Precession

- Precession induces magnetic flux
- Flux induces voltage in a coil



Audacity
Default Input Source
L
R
-21 0
-21 0
15.0
16.0
17.0
18.0
19.0
20.0
21.0
22.0
23.0
24.0
25.0
26.0
27.0
28.0
Audio Trac
Mute
Solo
L
R
0.05
0.00
0.05
0.00

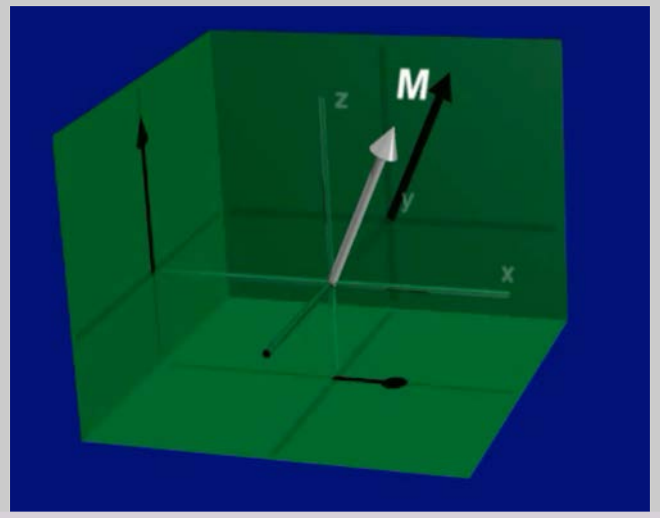
tereo, 8192Hz
2-bit float

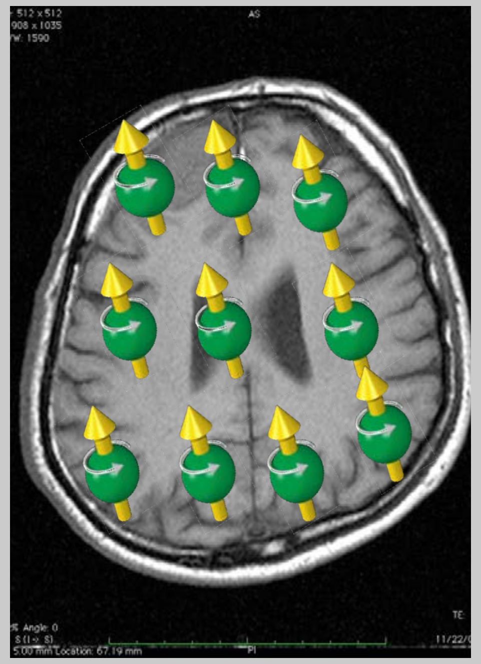
isk space remains for recording 761 hours and 21 minutes

oject rate: 8192 Cursor: 0:00.000000 min:sec [Snap-To Off]

Free precession

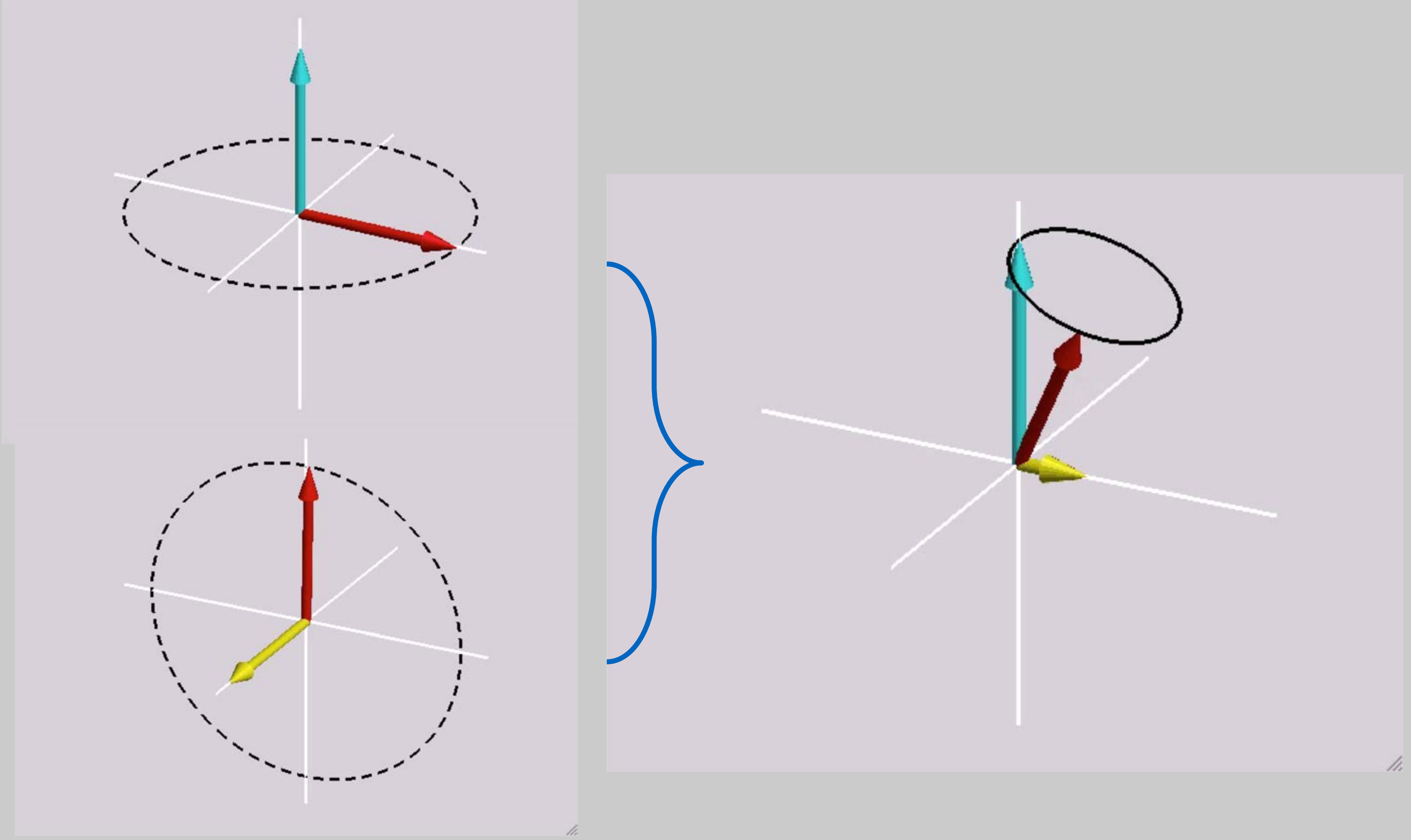
$$\begin{bmatrix} \dot{M}_x(\vec{r},t) \\ \dot{M}_y(\vec{r},t) \\ \dot{M}_z(\vec{r},t) \end{bmatrix} = \begin{bmatrix} 0 & \gamma B_0 & 0 \\ -\gamma B_0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} M_x(\vec{r},t) \\ M_y(\vec{r},t) \\ M_z(\vec{r},t) \end{bmatrix}$$





Gradient encoding

$$\begin{bmatrix} \dot{M}_x(\vec{r},t) \\ \dot{M}_y(\vec{r},t) \\ \dot{M}_z(\vec{r},t) \end{bmatrix} = \begin{bmatrix} 0 & \gamma \vec{G}(t)\cdot\vec{r} & 0 \\ -\gamma \vec{G}(t)\cdot\vec{r} & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} M_x(\vec{r},t) \\ M_y(\vec{r},t) \\ M_z(\vec{r},t) \end{bmatrix}$$

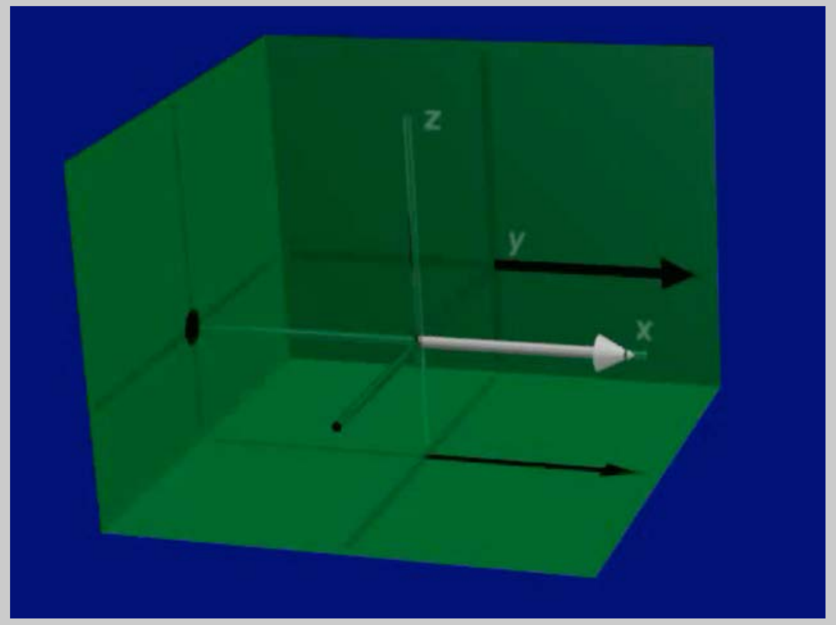


RF excitation

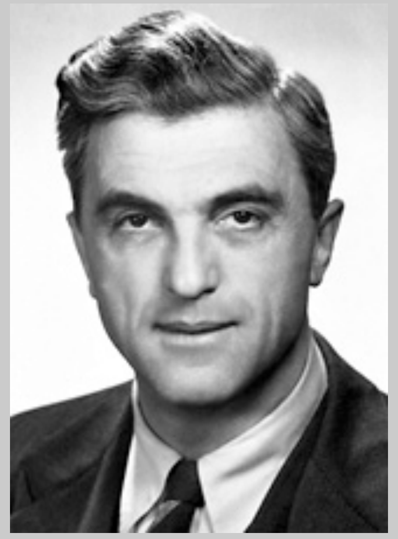
$$\begin{bmatrix} \dot{M}_x(\vec{r},t) \\ \dot{M}_y(\vec{r},t) \\ \dot{M}_z(\vec{r},t) \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & \gamma B_{1x}(t) \\ 0 & -\gamma B_{1x}(t) & 0 \end{bmatrix} \begin{bmatrix} M_x(\vec{r},t) \\ M_y(\vec{r},t) \\ M_z(\vec{r},t) \end{bmatrix}$$

Relaxation

$$\begin{bmatrix} \dot{M}_x(\vec{r},t) \\ \dot{M}_y(\vec{r},t) \\ \dot{M}_z(\vec{r},t) \end{bmatrix} = \begin{bmatrix} -\frac{1}{T_2} & 0 & 0 \\ 0 & -\frac{1}{T_2} & 0 \\ 0 & 0 & -\frac{1}{T_1} \end{bmatrix} \begin{bmatrix} M_x(\vec{r},t) \\ M_y(\vec{r},t) \\ M_z(\vec{r},t) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \frac{1}{T_1} \end{bmatrix} M_0(\vec{r})$$



The Bloch Equation



$$\begin{bmatrix} \dot{M}_x(\vec{r},t) \\ \dot{M}_y(\vec{r},t) \\ \dot{M}_z(\vec{r},t) \end{bmatrix} = \begin{bmatrix} -\frac{1}{T_2}(\vec{r}) & \gamma \vec{G}(t) \cdot \vec{r} & -\gamma B_{1y}(t) \\ -\gamma \vec{G}(t) \cdot \vec{r} & -\frac{1}{T_2}(\vec{r}) & \gamma B_{1x}(t) \\ \gamma B_{1y}(t) & -\gamma B_{1x}(t) & -\frac{1}{T_1}(\vec{r}) \end{bmatrix} \begin{bmatrix} M_x(\vec{r},t) \\ M_y(\vec{r},t) \\ M_z(\vec{r},t) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \frac{1}{T_1} \end{bmatrix} M_0(\vec{r})$$

$$C = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

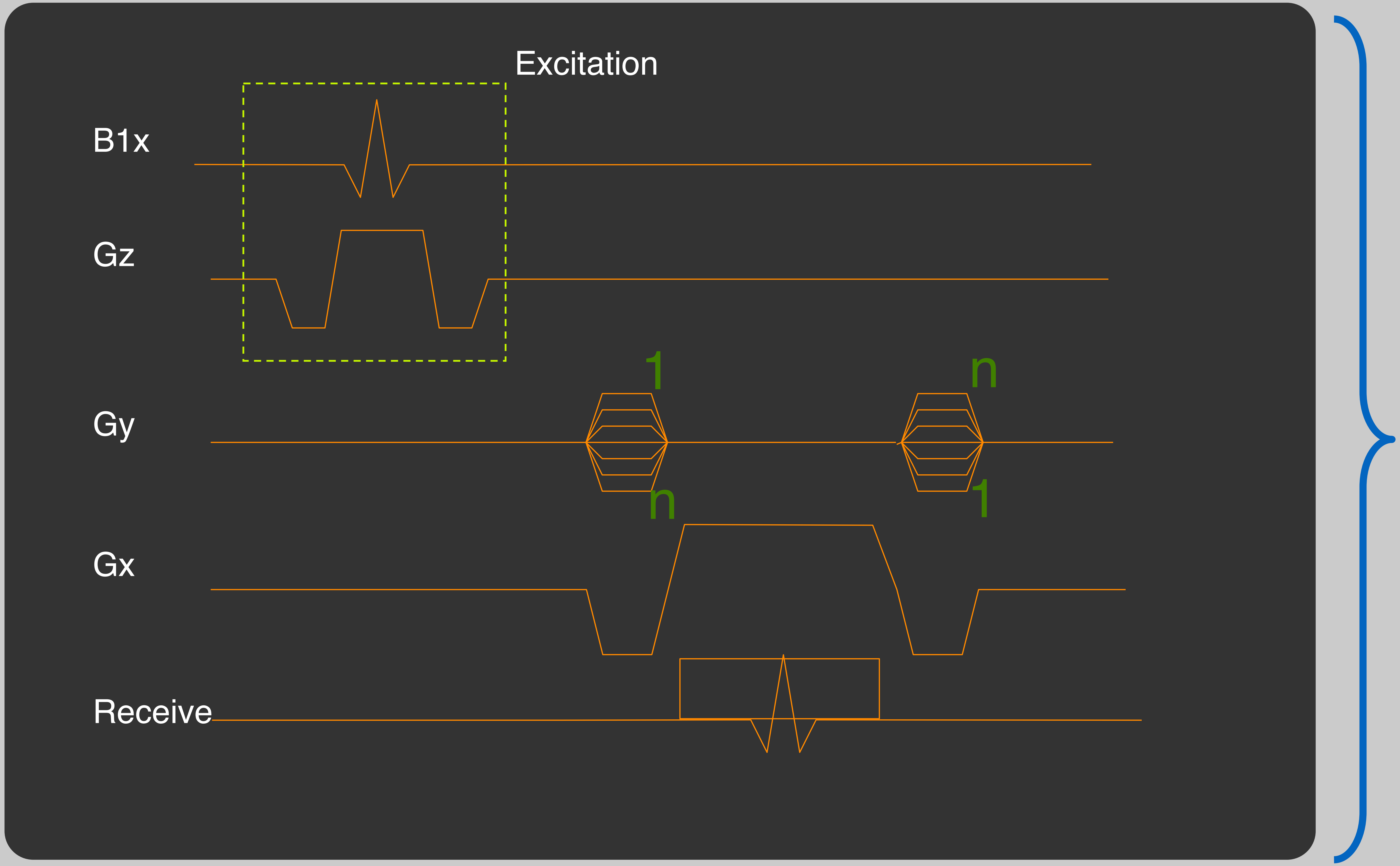
Observe: $\vec{y}(t) = \int_{\vec{R}} CM(\vec{r},t)d\vec{r}$

Inputs: $G_x(t), G_y(t), G_z(t), B_{1x}(t), B_{1y}(t)$

Unknown: $M_0(\vec{r})$ The image!

MRI Pulse Sequence

$$\begin{bmatrix} \dot M_x(\vec r,t) \\ \dot M_y(\vec r,t) \\ \dot M_z(\vec r,t) \end{bmatrix} = \begin{bmatrix} -\frac{1}{T_2(\vec r)} & \gamma \vec G(t)\cdot \vec r & -\gamma B_{1y}(t) \\ -\gamma \vec G(t)\cdot \vec r & -\frac{1}{T_2(\vec r)} & \gamma B_{1x}(t) \\ \gamma B_{1y}(t) & -\gamma B_{1x}(t) & -\frac{1}{T_1(\vec r)} \end{bmatrix} \begin{bmatrix} M_x(\vec r,t) \\ M_y(\vec r,t) \\ M_z(\vec r,t) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \frac{1}{T_1} \end{bmatrix} M_0(\vec r)$$



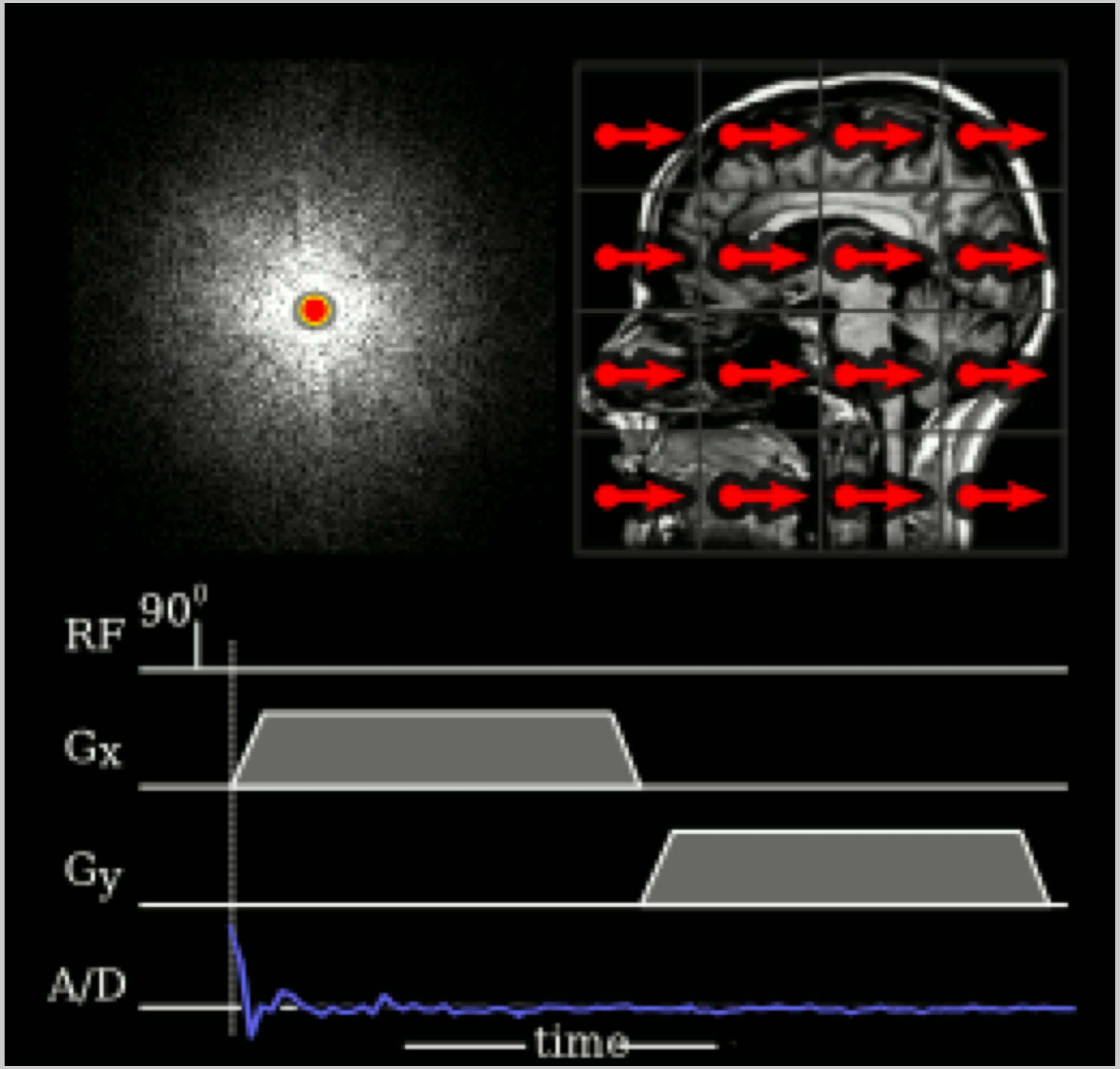
Repeat n times
rate = TR seconds

Acquisition

Observe:

$$\vec{y}(t) = \int_{\vec{R}} CM(\vec{r},t)d\vec{r}$$

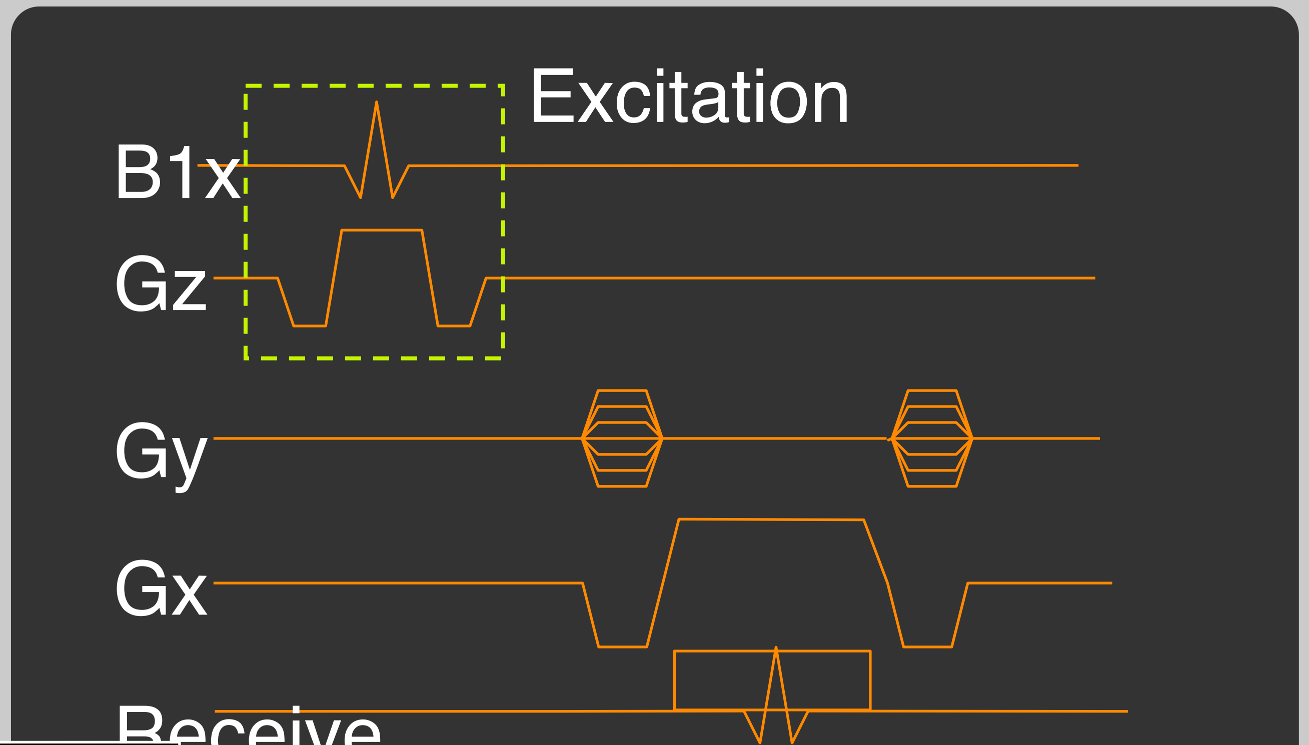
$$C = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

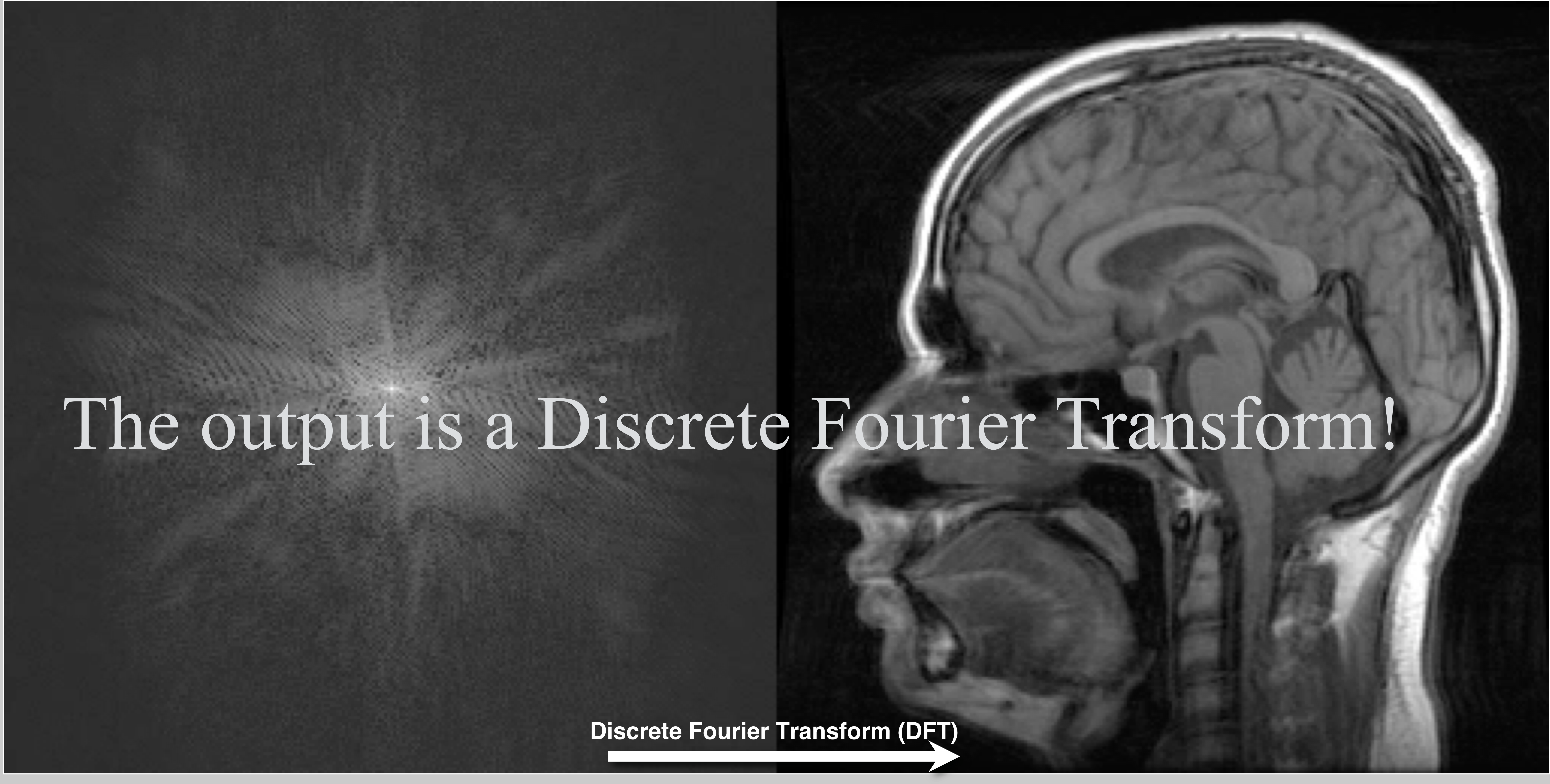


Fourier Reconstruction

$$y(t) = \int_R M_x(r,t) + jM_y(r,t)dr$$

Data (showing just magnitude)





The output is a Discrete Fourier Transform!