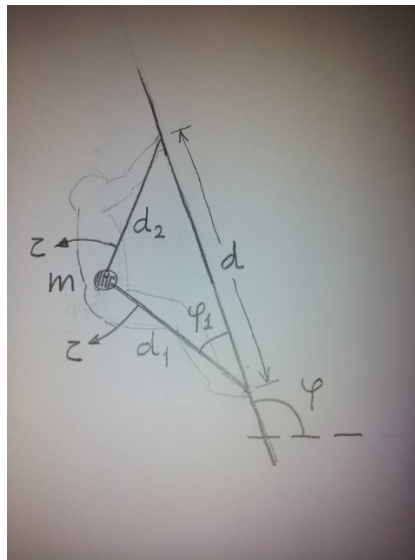


## Control 2. Computación I

Jose M. Soler  
Universidad Autónoma de Madrid

13 de febrero de 2017

1. En un modelo simplificado, consideramos un escalador como una masa puntual  $m$  con dos extremidades rígidas (piernas y brazos, sin masa) de longitudes  $d_1$  y  $d_2$ . El escalador se agarra a una pared inclinada un ángulo  $\phi$  con respecto a la horizontal, de modo que sus pies y manos están separados una distancia  $d$ . Además, con los músculos del tronco, el escalador ejerce un par de fuerzas  $\tau$  sobre sus extremidades, en sentidos opuestos (ver figura), para minimizar la fuerza ejercida por las manos, que son su punto más débil.



- Escribir una función con la siguiente interfaz (2 puntos)

```
function [r1,r2] = climberPosition( d1, d2, d, phi )  
% Climber's feet and hands positions, from the body center  
% Input:  
% d1 : legs length (m)  
% d2 : arms length (m)  
% d : distance between feet and hands (m)  
% phi : wall inclination from horizontal (rad)
```

```

% Output:
%   r1(2,1) : feet position vector, from body center (m)
%   r2(2,1) : hands position vector, from body center (m)
% Algorithm:
%   Use the triangular equation  $d_2^2 = d^2 + d_1^2 - 2*d*d_1*\cos(\phi_1)$  to
%   obtain  $\phi_1$ . Find position vectors  $r_0$  and  $r_2$  of body center and
%   hands, from feet. Subtract  $r_0$  to change origin to body center.

```

- Escribir un script *climber.m* que defina  $d_1 = 1,2$  m,  $d_2 = 1,0$  m,  $d = 2,0$  m,  $\phi = 120^\circ$ , llame a *climberPosition* y dibuje el triángulo formado por el punto central del escalador, sus pies y sus manos (1 punto).
- Escribir una función con la siguiente interfaz (2 puntos):

```

function [f1,f2] = climberForce( m, d1, d2, d, phi, tau )
% Force vectors on the climber's feet and hands
% Input:
%   m       : climber mass (kg)
%   d1      : climber legs length (m)
%   d2      : climber arms length (m)
%   d       : distance between feet and hands (m)
%   phi     : wall inclination from horizontal (rad)
%   tau     : body torque between arms and legs (N*m)
% Output:
%   f1(2,1) : column vector of feet force (N)
%   f2(2,1) : column vector of hands force (N)
% Algorithm:
%   Call climberPosition to find r1 and r2. Then solve
%   the equilibrium eqs:
%       f1x+f2x = 0           horizontal force
%       f1y+f2y-m*g = 0      vertical force
%       x1*f1y-y1*f1x-tau = 0 legs torque
%       x2*f2y-y2*f2x+tau = 0, arms torque
%   as  $A*f=b \Rightarrow f=A\b$ , with  $f=[f1x;f1y;f2x;f2y]$ .
%   Finally, separate f into f1 and f2

```

- En *climber.m*, definir  $m = 70$  kg y, llamando a *climberForce* dentro de un bucle, calcular la fuerza  $|f_2|$  de las manos, para valores de  $\tau$  entre -100 y +500 N m, a intervalos de 10 N m, y dibujar  $|f_2|$  en función de  $\tau$  (1 punto).
- Usando *find*, obtener el valor de  $\tau$  para el que se minimiza  $|f_2|$ , y escribir los dos valores (1 punto).
- En la primera figura, usando *quiver*, dibujar los vectores de las fuerzas sobre el cuerpo, los pies y las manos, divididas por  $mg$  (1 punto).

2. En el modelo dipolar más sencillo, el campo magnético terrestre es

$$\mathbf{B}(\mathbf{r}) = \left( \frac{\mu_0 m}{4\pi} \right) \frac{\hat{\mathbf{m}} - (\hat{\mathbf{m}} \cdot \hat{\mathbf{r}})\hat{\mathbf{r}}}{r^3} \quad (1)$$

siendo  $\mu_0 m / 4\pi = 8,06 \times 10^{15} \text{ Tm}^3$ ,  $\mathbf{r}$  el vector posición desde el centro de la Tierra y  $\mathbf{m}$  el momento magnético terrestre, que supondremos dirigido según el eje  $z$ .  $\hat{\mathbf{m}}$  y  $\hat{\mathbf{r}}$  son vectores unitarios en las direcciones  $\mathbf{m}$  y  $\mathbf{r}$ . La fuerza de Lorentz de dicho campo sobre una partícula cargada es  $\mathbf{f} = q \mathbf{v} \times \mathbf{B}$ , siendo  $q$  su carga y  $\mathbf{v}$  su velocidad.

- Escribir una función con las siguiente interfaz (2 puntos):

```
function f = earthMagneticForce(q,r,v)
% Lorentz force due to Earth's magnetic field,
% with z axis along the north pole
% Input:
%   q       : particle charge (C)
%   r(3,1)  : particle position vector (m)
%   v(3,1)  : particle velocity (m/s)
% Output:
%   f(3,1)  : particle magnetic force (N)
```

- En un script *protonTrajectory.m*, definir la masa y carga de un protón de rayos cósmicos, sus condiciones iniciales  $\mathbf{r}_0 = (1, 2, 3) \times 10^8 \text{ m}$ ,  $\mathbf{v}_0 = -(1, 2, 3) \times 10^7 \text{ m/s}$ , la función de fuerza *myForce=@(r,v)magneticForce(q,r,v)*, llamar a la función *rungeKutta* de clase para obtener la trayectoria durante 30 s, y dibujarla en tres dimensiones (2 puntos).

### Notas:

- Los programas deben estar documentados en inglés y no deben escribir nada que no se pida.
- Las figuras deben incluir etiquetas en cada eje.
- Enviar los ficheros *climberPosition.m*, *climberForces.m*, *climber.m*, *earthMagneticForce.m* y *protonTrajectory.m* a jose.soler@uam.es antes de terminar el examen.
- Para cada función y programa, se valorará:
  1. Que tenga la interfaz que se pide, y haga lo que se pide sin errores.
  2. Que esté bien documentado.
  3. Que esté programado de forma sencilla y fácil de entender.