UNIVERSITÉ DE SHERBROOOKE

Faculté de génie

Département de génie électrique et génie informatique

APP 2 : Principes de dynamique et méthodes numériques

GEN441 02,GEL450,GEN441 01,GIF590

Présenté à

Maude Blondin et François Boone

Présenté par

Shawn Miller-Morneau

Alexis Juteau

Sherbrooke - 9 août 2024

Table des matières

Table des figures

Table des codes

1 Trajectoire et débit d'eau

1.1 Polynôme d'interpolation de la trajectoire de la glissade

Points	A	B	C	D	E
Coordonnées horizontales (m)	0	8	15	20	25
Coordonnées verticales (m)	30	19	20	16	$10 \leq y_f \leq 15$

Tableau 1 – Coordonnées de la trajectoire

Sachant que la glissade comporte une fonction non linéaire constituée de cinq points, il est possible de représenter les valeurs par une fonction polynomiale d'ordre 4. Autrement dit, l'ordre est déterminé par n-1, soit 5 points représentent un ordre 4. Pour déterminer les coefficients, il faut utiliser la méthode de la projection orthogonal et les manipulations matricielles.

$$g_4(x) = a_1x^0 + a_2x^1 + a_3x^2 + a_4x^3 + a_5x^4$$ **Note : Fonction candidate**

$$\begin{bmatrix} x_1^0 & x_1^1 & x_1^2 & x_1^3 & x_1^4 \\ x_2^0 & x_2^1 & x_2^2 & x_2^3 & x_2^4 \\ x_3^0 & x_3^1 & x_3^2 & x_3^3 & x_3^4 \\ x_4^0 & x_4^1 & x_4^2 & x_4^3 & x_4^4 \\ x_5^0 & x_5^1 & x_5^2 & x_5^3 & x_5^4 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ a_4 \end{bmatrix} = \begin{bmatrix} 30 \\ 19 \\ 20 \\ 16 \\ y_f \end{bmatrix}$$ **Note : Sous forme matricielle**

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & 8 & 64 & 512 & 4096 \\ 1 & 15 & 225 & 3375 & 50625 \\ 1 & 20 & 400 & 8000 & 160000 \\ 1 & 25 & 625 & 15625 & 390625 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ a_4 \end{bmatrix} = \begin{bmatrix} 30 \\ 19 \\ 20 \\ 16 \\ y_f \end{bmatrix}$$ **Note : Valeur calculé avec Tableau 1**

Pour déterminer les coefficients de la matrice A de dimension 5x1, il est nécessaire de calculer la pseudo-inverse de la matrice C à l'aide de MATLAB, puis de la multiplier par la matrice Y. Il convient de noter que la valeur de y_f est choisi arbitrairement en respectant les spécifications, sois 12.25.

$$A = (C^{\top}C)^{-1}C^{\top}Y$$

$$A = C^{\perp}Y$$

$$A = pinv(C)Y$$

$$A = \begin{bmatrix} 1.0000 & 0.0000 & -0.0000 & 0.0000 & -0.0000 \\ -0.2817 & 0.6565 & -0.7619 & 0.5000 & -0.1129 \\ 0.0276 & -0.1029 & 0.1638 & -0.1158 & 0.0273 \\ -0.0011 & 0.0053 & -0.0101 & 0.0080 & -0.0020 \\ 0.0000 & -0.0001 & 0.0002 & -0.0002 & 0.0000 \end{bmatrix} \begin{bmatrix} 30 \\ 19 \\ 20 \\ 16 \\ 12.25 \end{bmatrix}$$

$$A = \begin{bmatrix} 30.0000 \\ -4.5979 \\ 0.6305 \\ -0.0329 \\ 0.0006 \end{bmatrix}$$

$$\boxed{g(x) = 30 - 4.5979x^1 + 0.6305x^2 - 0.0329x^3 + 0.0006x^4}$$

Code 1 – Calcul de matrice pseudo inverse

```
C = [1, 0, 0, 0, 0;
     1, 8, 64, 512, 4096;
     1, 15, 225, 3375, 50625;
     1, 20, 400, 8000, 160000;
     1, 25, 625, 15625, 390625];
y = [30;
     19;
     20;
     16;
     12.25];
C_pinv = pinv(C);
A = C_pinv * y;
```

1.2 Polynôme d'approximation de μ_f en fonction de l'ouverture de la valve

Tests	1	2	3	4	5	6	7	8	9	10	11
Ouverture (%)	0	10	20	30	40	50	60	70	80	90	100
Coefficient	0.87	0.78	0.71	0.61	0.62	0.51	0.51	0.49	0.46	0.48	0.46

Tableau 2 – Coefficient de friction dynamique en fonction de l'ouverture de la valve

Pour définir la fonction candidate qui sera utilisée pour déterminer les coefficients, il est possible de déduire l'ordre de la fonction en définissant la forme de la pente de la courbe. Dans ce cas, puisque nous ne voulons pas interpolé une courbe avec tout les points candidats, il n'est pas nécessaire d'avoir un ordre 10. Ainsi, il est acceptable d'utiliser un modèle d'ordre 2 pour approximer une courbe exponentielle entre nos points. Pour déterminer les coefficients, il faut utiliser la méthode de la projection orthogonale et les manipulations matricielles.

$g(x) = a_0 x^0 + a_1 x^1 + a_0 x^2$ **Note : Fonction candidate**

$$\begin{bmatrix} x_1^0 & x_1^1 & x_1^2 \\ x_2^0 & x_2^1 & x_2^2 \\ x_3^0 & x_3^1 & x_3^2 \\ x_4^0 & x_4^1 & x_4^2 \\ x_5^0 & x_5^1 & x_5^2 \\ x_6^0 & x_6^1 & x_6^2 \\ x_7^0 & x_7^1 & x_7^2 \\ x_8^0 & x_8^1 & x_8^2 \\ x_9^0 & x_9^1 & x_9^2 \\ x_{10}^0 & x_{10}^1 & x_{10}^2 \\ x_{11}^0 & x_{11}^1 & x_{11}^2 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} 0.87 \\ 0.78 \\ 0.71 \\ 0.61 \\ 0.62 \\ 0.51 \\ 0.51 \\ 0.49 \\ 0.46 \\ 0.48 \\ 0.46 \end{bmatrix} \quad \textbf{Donc :} \begin{bmatrix} 1 & 0 & 0 \\ 1 & 10 & 100 \\ 1 & 20 & 400 \\ 1 & 30 & 900 \\ 1 & 40 & 1600 \\ 1 & 50 & 2500 \\ 1 & 60 & 3600 \\ 1 & 70 & 4900 \\ 1 & 80 & 6400 \\ 1 & 90 & 8100 \\ 1 & 100 & 10000 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} 0.87 \\ 0.78 \\ 0.71 \\ 0.61 \\ 0.62 \\ 0.51 \\ 0.51 \\ 0.49 \\ 0.46 \\ 0.48 \\ 0.46 \end{bmatrix}$$

$A = (C^\top C)^{-1} C^\top Y$

$A = C^\perp Y$

$$A = pinv(C)Y$$

$$A = \begin{bmatrix} 0.5804 & 0.3776 & 0.2098 & 0.0769 & -0.0210 & -0.0839 & -0.1119 & -0.1049 & -0.0629 & 0.0140 & 0.1259 \\ -0.0220 & -0.0106 & -0.0016 & 0.0052 & 0.0096 & 0.0117 & 0.0114 & 0.0088 & 0.0039 & -0.0034 & -0.0129 \\ 0.0002 & 0.0001 & -0.0000 & -0.0001 & -0.0001 & -0.0001 & -0.0001 & -0.0001 & -0.0000 & 0.0001 & 0.0002 \end{bmatrix} \begin{bmatrix} 0.87 \\ 0.78 \\ 0.71 \\ 0.61 \\ 0.62 \\ 0.51 \\ 0.51 \\ 0.49 \\ 0.46 \\ 0.48 \\ 0.46 \end{bmatrix}$$

$$A = \begin{bmatrix} 0.86678 \\ -0.00916 \\ 0.00005 \end{bmatrix}$$

$$\boxed{g(x) = 0.86678 - 0.00916x + 0.00005x^2}$$

1.3 Qualité d'approximation avec l'erreur RMS

$$RMSE = \sqrt{\frac{1}{N}E}$$

$$RMSE = \sqrt{\frac{1}{N}\sum_{n=1}^{N}[g(x_n) - y_n]^2}$$

En utilisant Matlab, il est possible de calculer les valeurs de l'erreur RMSE en variant l'ordre du modèle, donc en ajoutant davantage de coefficients.

Code 2 – Extrait de code pour le calcul RMSE selon différents ordres

```
% Calculs coefficients
Z_5 = pinv(C_5) * y;
Z_4 = pinv(C_4) * y;
Z_3 = pinv(C_3) * y;
% Polynomes differents ordres
g_5 = Z_5(1) + Z_5(2)*x + Z_5(3)*x.^2 + Z_5(4)*x.^3 + Z_5(5)*x.^4;
```

```
g_4 = Z_4(1) + Z_4(2)*x + Z_4(3)*x.^2 + Z_4(4)*x.^3;
g_3 = Z_3(1) + Z_3(2)*x + Z_3(3)*x.^2;
% Calculate RMSE pour chaque ordre
rmse_5coeff = sqrt((1/11) * sum((g_5 - y').^2))
rmse_4coeff = sqrt((1/11) * sum((g_4 - y').^2))
rmse_3coeff = sqrt((1/11) * sum((g_3 - y').^2))
```

$$RMSE_{ordre4} = 0.0176$$

$$RMSE_{ordre3} = 0.0178$$

$$RMSE_{ordre2} = 0.0180$$

1.4 Vitesse du participant le long de la trajectoire (en fonction de la coordonnée horizontale, x)

1.4.1 Force de frottement et calcul du travail

$\vec{F} = \mu_f mgcos(\theta)$ **Note : Définition du frottement**

$$W_{\vec{F}} = \int_A^E \vec{F} \cdot \vec{ds}$$

$$\vec{F} \cdot \vec{ds} = ||\vec{F}|| \cdot ||\vec{ds}||cos(\angle(\vec{F}, \vec{ds}))$$

$$\angle(\vec{F}, \vec{ds}) = \angle(\vec{F}, \vec{v}) = 180\,\mathrm{deg}$$

$$\vec{F} \cdot \vec{ds} = ||\vec{F}||dscos(180\,\mathrm{deg})$$

$$\vec{F} \cdot \vec{ds} = -||\vec{F}||ds$$

$$\vec{F} \cdot \vec{ds} = -\mu_f \cdot m \cdot g \cdot cos(\theta)ds$$

Donc :

$$W_{\vec{F}} = \int_A^E \vec{F} \cdot \vec{ds}$$

$$W_{\vec{F}} = \int_A^E -\mu_f \cdot m \cdot g \cdot cos(\theta)ds$$

$$W_{\vec{F}} = -\int_A^E \mu_f \cdot m \cdot g \cdot dx \qquad \textbf{Note : dx=cos(}\theta\textbf{)ds}$$

$$W_{\vec{F}} = -\mu_f \cdot m \cdot g \cdot [x]\Big|_a^b$$

$$W_{\vec{F}} = -\mu_f \cdot 80kg \cdot 9.81m/s \cdot [x]\Big|_a^b$$

$$\boxed{W_{\vec{F}} = -784.8 \cdot \mu_f \cdot x}$$

1.4.2 Théorème de la conservation de l'énergie

$$E_{ci} + E_{pgi} + \cancel{E_{pri}} + W_F = E_{cf} + E_{pgf} + \cancel{E_{prf}}$$

$$\cancel{\frac{1}{2} \cdot m \cdot v^2} + m \cdot g \cdot h_i + W_{\vec{F}} = \frac{1}{2} \cdot m \cdot v^2 + m \cdot g \cdot h_f$$

$$\cancel{m} \cdot g \cdot h_i - \cancel{m} \cdot g \cdot h_f + \cancel{m} \cdot g \cdot \mu_f \cdot x = \frac{1}{2} \cdot \cancel{m} \cdot v^2$$

$$2 \cdot g((h_i - h_f) - \mu_f x) = v^2$$

$$\boxed{v = \sqrt{2g((h_i - h_f) - \mu_f x)}}$$

1.4.3 Graphique de la vitesse du participant le long de la trajectoire (en fonction de la coordonnée horizontale, x)

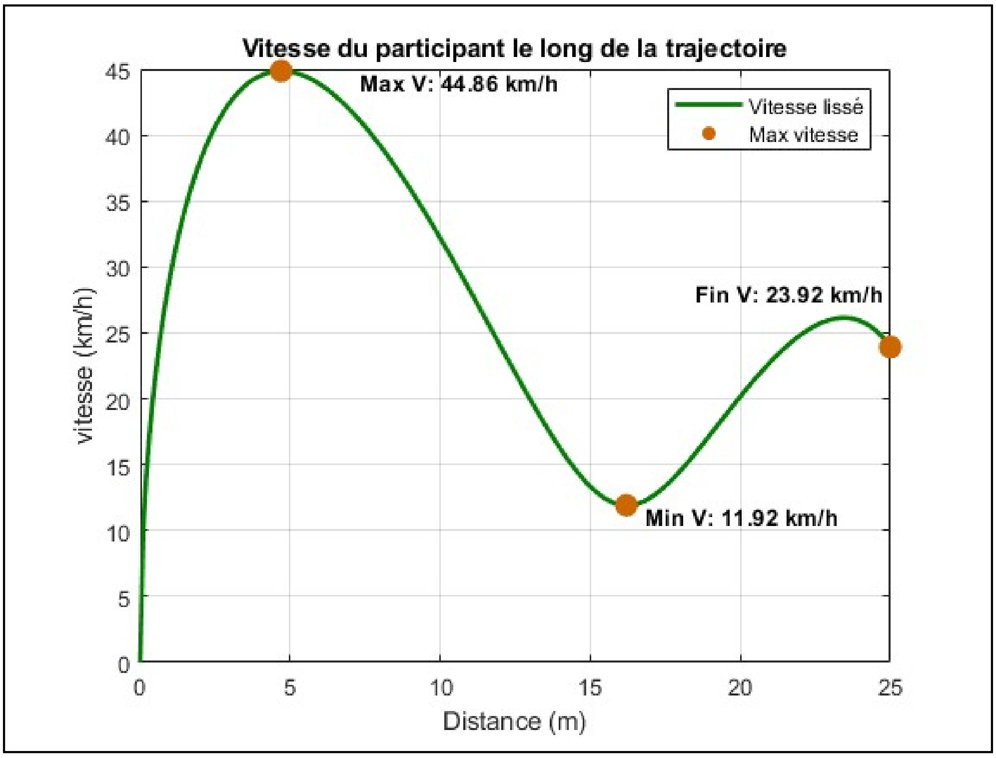

FIGURE 1 – Vitesse du participant

1.4.4 Calcul et affichage de la vitesse avec MatLab

Code 3 – Calcul et affichage de la vitesse du participant

```
% [...] Code generation matrice
A = C_pinv * y;
x = 0:0.1:25;
g = A(1) + A(2)*x + A(3)*x.^2 + A(4)*x.^3 + A(5)*x.^4;
%%%%%%%%%%%%CONSTANTES%%%%%%%%%%%%%%%%%%%%
kms=3.6;       % Transfer vers km/s
hi = 30;       % Hauteur initiale
mu_f = 0.62; % Coefficient de friction
x = 0:0.1:25;
v = sqrt(2 * 9.81 * ((hi - g)- mu_f * x ))*kms;
%%%%%%%%%%%%%POINT IMPORTANT%%%%%%%%%%%%%
```

```
[max_v, max_idx] = max(v);
min_v = min(v(max_idx+1:end)); % Minimum apres le maximum
min_idx = find(v == min_v, 1);
end_x = x(end);
end_v = v(end);
%%%%%%%%%%%%TRACAGE DE LA COURBE%%%%%%%%
figure(2);
plot(x, v, 'Color', VERT_SHERB, 'LineWidth', 2);
xlabel('Distance (m)');
ylabel('vitesse (km/h)');
title('Vitesse du participant le long de la trajectoire');
hold on;
scatter(x(max_idx), max_v, 100, OR_SHERB, 'filled');   %MAX
scatter(x(min_idx), min_v, 100, OR_SHERB, 'filled');   %MIN
scatter(end_x, end_v, 100, OR_SHERB, 'filled');        %FIN
legend('Vitesse lisse', 'Max vitesse');
%%%%%%%%%%%%Points importants%%%%%%%%%%%%
hold off;
grid on;
% [...] Affichage de textes adjacents aux points.
```

1.5 Commentaires vitesse du participant et choix de μ_f

Pour le choix de μ_f, la valeur a été décidée empiriquement en fonction des spécifications mentionnées dans le devis, à savoir une vitesse maximale de 45 km/h, une vitesse minimale de 10 km/h, ainsi qu'une vitesse de sortie au point E devant se situer entre 20 km/h et 25 km/h. Avec une hauteur finale de 12,25 mètres (choix arbitraire) et un coefficient de friction de 0,62 (soit 32,5 %), nous obtenons une vitesse maximale de 44,86 km/h, une vitesse minimale de 11,92 km/h, et une vitesse au point E de 23,92 km/h. Pour aller plus loin, si l'on tient compte de l'erreur RMS de 0,018, nous constatons que les valeurs obtenues dépassent les plages spécifiées dans le devis. Une recommandation pour le client serait de réduire la hauteur initiale et de réduire l'angle de la pente de la glissade.

1.6 Coefficient de friction choisi en fonction de l'ouverture de la valve en % (graphique)

Le graphique représente la trajectoire du polynôme d'approximation mentionné au point 1.2 de ce rapport. De plus, chaque point sur la courbe représente le pourcentage d'ouverture de la valve en fonction du coefficient choisi empiriquement.

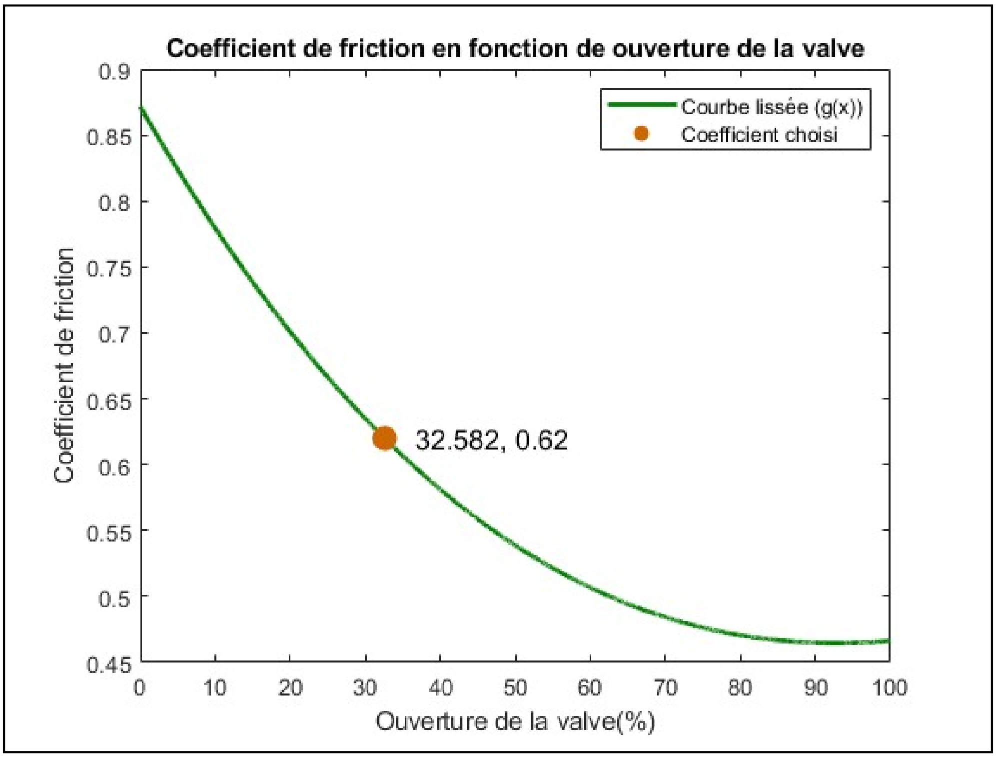

FIGURE 2 – Coefficient de friction en fonction de l'ouverture de la valve(%)

2 Ballon-mousse et minuterie

2.1 Suppositions

1. L'impact est direct et central ; il a lieu au point E, avant de parcourir la longueur $L_T = 3m$ de la trappe.
2. La friction entre le participant et la surface de la trappe est négligeable (pour simplifier).
3. Il y a conservation de la quantité du mouvement (aucune impulsion externe).

$$e = \frac{v_{Bf} - v_{Af}}{v_{Ai} - v_{Bi}} \tag{1}$$

$$m_A v_{Ai} + m_B v_{Bi} = m_A v_{Af} + m_B v_{Bf} \tag{2}$$

2.2 Calcule vitesse et temps pour le ballon attrapé (G1)

$$v_{Af} = v_{Bf}$$

$$m_A v_{Ai} + m_B v_{Bi} = m_A v_{Af} + m_B v_{Bf}$$

$$m_A v_{Ai} + m_B v_{Bi} = v_{Af}(m_A + m_B)$$

$$v_{Af} = \frac{m_A v_{Ai} + m_B v_{Bi}}{(m_A + m_B)} = \frac{80kg * 6.42m/s + 8kg * -1m/s}{(80kg + 8kg)} = \boxed{5.745m/s}$$

$$\boxed{t_{G1} = \frac{L_T}{v_{Af}} = \frac{3m}{5.745m/s} = 0.522s} \tag{3}$$

2.3 Calcule vitesse et temps pour le ballon rebondit (G2)

$$v_{Bf} = \frac{v_{Ai} + \frac{m_B}{m_A} v_{Bi} + e(v_{Ai} - v_{Bi})}{1 + \frac{m_B}{m_A}} \tag{4}$$

$$v_{Af} = v_{Bf} - e(v_{Ai} - v_{Bi}) \tag{5}$$

$$v_{Af} = \frac{v_{Ai} + \frac{m_B}{m_A} v_{Bi} + e(v_{Ai} - v_{Bi})}{1 + \frac{m_B}{m_A}} - e(v_{Ai} - v_{Bi})$$

$$v_{Af} = \frac{6.42m/s + \frac{8kg}{80kg} * -1m/s + 0.8(6.42m/s - -1m/s)}{1 + \frac{8kg}{80kg}} - 0.8(6.42m/s - -1m/s)$$

$$\boxed{v_{Af} = 5.2058m/s}$$

$$t_{G2} = \frac{L_T}{v_{Af}} = \frac{3m}{5.2058m/s} = \boxed{0.5763s} \tag{6}$$

2.4 Durée de la minuterie avant que la trappe ne s'ouvre

$$\Delta t = \frac{(t_{G1} - 0.02) + (t_{G2} + 0.02)}{2} = \frac{(0.522s - 0.02) + (0.5763s + 0.02)}{2} = \boxed{0.5492s} \tag{7}$$

2.4.1 Marge obtenue

$$\Delta t - t_G 1 > 0.02 \longrightarrow 0.5492s - 0.522s > 0.02 \longrightarrow \boxed{0.029 > 0.02}$$

$$t_G 2 - \Delta t > 0.02 \longrightarrow 0.5763s - 0.5492s > 0.02 \longrightarrow \boxed{0.0244 > 0.02}$$

2.5 Code MATLAB du Ballon-mousse et minuterie

Code 4 – Justification calculs Ballon-mousse et minuterie

```
L_t = 3;
e_G2 = 0.8;
m_b = 8;
m_p = 80;
v_pi = 6.42;
v_bi = -1;
%G1
v_pf = (m_p*v_pi + m_b*v_bi)/(m_b+m_p)
t_G1 = L_t/v_pf
```

```
%G2
v_bf = (v_pi + (m_b/m_p)*v_bi + e_G2*(v_pi - v_bi))/(1+(m_b/m_p));
v_pf = v_bf - e_G2*(v_pi-v_bi)
t_G2 = L_t/v_pf
dt = ((t_G1 - 0.02)+(t_G2 + 0.02))/2
```

3 Coussin trampoline

3.1 Suppositions

1. Participant-ballon tombe d'une hauteur $h_0 = 5m$ avec vitesse initiale nulle
2. Coussin-trampoline considéré un ressort linéaire de constante $k_c = 6000N/m$.
3. Masse du coussin-trampoline négligeable.
4. Trainée aérodynamique pendant la tombée négligeable.
5. Amortissement dans le coussin-trampoline négligeable.
6. Participant-ballon collé au coussin-trampoline après l'impact (pas de rebond du participantballon).

3.2 Équation pour le coussin trampoline

Déterminer l'énergie cinétique du participant avec ballon avant impact et la vitesse initiale lors du contact avec le coussin trampoline.

$$\varepsilon_c = (m_b + m_p)g * h_0 \tag{8}$$

$$\varepsilon_c = \frac{(m_b + m_p)v_i^2}{2} \tag{9}$$

$$v_i = \sqrt{\frac{2\varepsilon_c}{(m_b + m_p)}} \tag{10}$$

$$\varepsilon_c = (m_b + m_p)g * h_0 = (8kg + 80kg) * 9.81m/s * 5m = \boxed{4316,4J}$$

$$v_i = \sqrt{\frac{2\varepsilon_c}{(m_b + m_p)}} = \sqrt{\frac{2 * 4316,4J}{(8kg + 80kg)}} = \boxed{9,904m/s}$$

Avec ces paramètre, il est possible d'identifier la hauteur sécuritaire du coussin avec la conservation de l'énergie où T est utilisé pour l'énergie cinétique, V est utilisé pour l'énergie potentielle et W est utilisé pour le travail d'une force

$$\varepsilon_c = mg\Delta x \qquad \varepsilon_r = \frac{k_c \Delta x^2}{2} \qquad W = \frac{(m_b + m_p)v_i^2}{2}$$

$$\varepsilon_{ci} + \varepsilon_{ri} + W = \varepsilon_{cf} + \varepsilon_{rf} \tag{11}$$

$$mg\Delta h + \frac{k_c \Delta h^2}{2} + W = mg\Delta h + \frac{k_c \Delta h^2}{2}$$

$$0 + 0 + W = mg\Delta h + \frac{k_c \Delta h^2}{2}$$

$$W = \frac{k_c \Delta h^2}{2} + mg\Delta h \tag{12}$$

$$0 = \frac{k_c \Delta h^2}{2} + mg\Delta h - W$$

À l'aide de l'équation quadratique, il est possible de déterminer Δh

$$\boxed{\Delta h = [-1.3520, 1.0642]}$$

La hauteur sécuritaire pour le coussin trampoline serai de minimum 1.35m pour une masse maximale d'un participant de 80kg. Pour une question de sécurité, la hauteur du coussin trampoline devrai être de 1.75m.

3.3 Code MATLAB du coussin trampoline

Code 5 – Justification calculs coussin trampoline

```
h_0 = 5;
g = 9.81;
m_b = 8;
```

```
m_p = 80;
m = m_b + m_p;
k_c = 6000;

%Energie cinetique
E_c = m * g * h_0
%Vitesse
v_i = sqrt(2 * E_c / m)
%Hauteur
h = roots([k_c/2 m*g -E_c])
%Verification Energie
E = k_c * h.^2 / 2 + m * g .* h
```

4 Bassin d'eau

4.1 Suppositions

1. On néglige la traînée aérodynamique pendant la tombée dans l'air
2. On ignore la forme physique du participant ; on ne considère que le déplacement de son centre de masse.
3. Les forces appliquées sur le participant une fois dans l'eau sont la gravité (mg), la flottabilité de forme $k_f mg$ et la trainée hydrodynamique de la forme bv^2, les deux dernièress'opposant à la vitesse de descente dans l'eau, où b est le coefficient hydrodynamique, k_f le facteur de flottabilité et v la vitesse dans l'eau
4. Les calculs se font sur la base de l'équation différentielle linéarisée
5. On suppose que le participant tombe dans l'eau d'une hauteur h_1 avec une vitesse initiale nulle
6. La profondeur sécuritaire z_{bassin} du bassin d'eau est la profondeur à laquelle la vitesse duparticipant est réduite à une vitesse d'impact v_f sécuritaire

4.2 Équation pour le bassin

Déterminer l'énergie cinétique du participant avant impact $z = 0, h_1 = 10m$ et la vitesse initiale lors de l'entrée dans l'eau.

$$\varepsilon_c = m_{pg}g * h_1 \tag{13}$$

$$\varepsilon_c = \frac{m_{pg}v_i^2}{2} \tag{14}$$

$$v_i = \sqrt{\frac{2\varepsilon_c}{m_{pg}}} \tag{15}$$

$$\varepsilon_c = m_{pg}g * h_1 = 88kg * 9.81m/s * 10m = \boxed{7848J}$$

$$v_i = \sqrt{\frac{2\varepsilon_c}{m_{pg}}} = \sqrt{\frac{2 * 7848J}{80kg}} = \boxed{14.007m/s}$$

Pour identifier la profondeur sécuritaire du bassin, on utilise les équations définis par les requis du devis reçu. L'équation est non linéaire, donc à l'aide de la séries de Taylor, on linéarise l'équation.

$$\Sigma F = ma = m\frac{dv}{dz} = m\vec{v}\frac{dv}{dz} \tag{16}$$

$$\Sigma F = mg - k_f mg - bv^2 \tag{17}$$

$$m\vec{v}\frac{dv}{dz} = mg - k_f mg - bv^2 \rightarrow \frac{dv}{dz} = \frac{g(1-k_f)}{v} - \frac{bv}{m} \tag{18}$$

$$\frac{d^2v}{dz^2} = -\frac{g(1-k_f)}{v^2} - \frac{b}{m} \tag{19}$$

$$f(x_0 + \Delta x) = f(x_0) + \left.\frac{df(x)}{dx}\right|_{x=x_0} \Delta x$$

$$\frac{d\Delta v}{dz} = \frac{g(1-k_f)}{v_0} - \frac{bv_0}{m} + (-\frac{g(1-k_f)}{v_0^2} - \frac{b}{m})\Delta v \tag{20}$$

Pour identifier le point v_0 le plus proche de $v_f = 1m/s$ on pose que $\frac{dv}{dz} = 0m/s^2$ Ceci représente l'équation d'équilibre.

$$\frac{dv}{dz} = \frac{g(1-k_f)}{v_0} - \frac{bv_0}{m} \to 0 = \frac{g(1-k_f)}{v_0} - \frac{bv_0}{m} \to v_0 = \sqrt{\frac{mg(1-k_f)}{b}} \tag{21}$$

$$v_0 = \sqrt{\frac{mg(1-k_f)}{b}} = \sqrt{\frac{80kg * 9.81m/s(1-0.95)}{47kg/m}} = \boxed{0.9137m/s}$$

Dans le cas ou $\frac{dv}{dz} = 0m/s^2$ on peut simplifier l'équation linéarisé par :

$$\frac{d\Delta v}{dz} = (-\frac{g(1-k_f)}{v_0^2} - \frac{b}{m})\Delta v$$

Il est alors possible d'isoler la profondeur z_{bassin} nécéssaire pour le bassin.

$$\frac{d\Delta v}{dv} = (-\frac{g(1-k_f)}{v_0^2} - \frac{b}{m})\Delta z$$

$$\int_{v_i}^{v_f} \frac{d\Delta v}{dv} = \int (-\frac{g(1-k_f)}{v_0^2} - \frac{b}{m})\Delta z$$

$$ln(\Delta v)\Big|_{v_i}^{v_f} = (-\frac{g(1-k_f)}{v_0^2} - \frac{b}{m})z_{bassin}$$

$$z_{bassin} = \frac{ln(v_i - v_0) - ln(v_f - v_0)}{-\frac{g(1-k_f)}{v_0^2} - \frac{b}{m}} \tag{22}$$

$$z_{bassin} = \frac{ln(v_i - v_0) - ln(v_f - v_0)}{-\frac{g(1-k_f)}{v_0^2} - \frac{b}{m}} = \frac{ln(14.007m/s - 0.9137m/s) - ln(1m/s - 0.9137m/s)}{-\frac{9.81m/s(1-0.95)}{0.9137^2 m/s} - \frac{47kg/m}{80kg}}$$

$$\boxed{z_{bassin} = -4.27m \approx -5m}$$

4.3 Code MATLAB du bassin

Code 6 – Justification calculs bassin

```
h = 10;
g = 9.81;
m_b = 8;
m_p = 80;
m = m_b + m_p;
k_f = 0.95;
b = 47;
v_f = 1;
%Energie cinetique
E = m * g * h %8.6328kJ
%Vitesse
v_i = sqrt(2 * E / m) %14.0071 m/s
v_0 = sqrt(m * g * (1 - k_f) / b) %0.9583 m/s
%Profondeur Bassin
D_2 = -g*(1 - k_f)/v_0^2 - b/m
z = (log(v_i-v_0)- log(v_f-v_0))/D_2 % -5.3797m
```

Références

[1] Jean de Lafontaine, ing, *Guide de l'étudiant - Principes de dynamique et méthodes numériques - S5 Électrique et S5 Informatique – APP 2*, Département de génie électrique et de génie informatique, Université de Sherbrooke