

Présentation des Sciences de l'Ingénieur

Que sont les sciences de l'ingénieur ?

La société a besoin d'innovations sociales, technologiques et environnementales



la santé



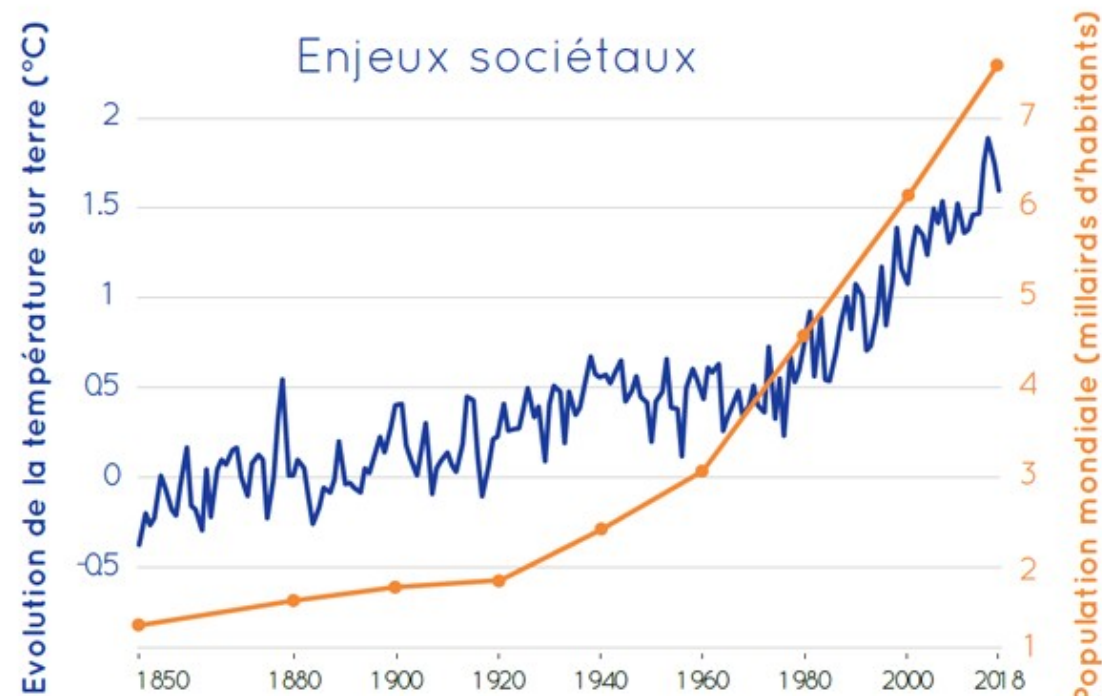
l'énergie



l'information



les transports



Présentation des Sciences de l'Ingénieur

Les Sciences de l'ingénieur : une réponse au besoin d'innovations

Un enseignement scientifique et transversal centré sur l'innovation technologique alliant notamment l'Ingénierie **Mécanique, Électrique, Automatique** et **Numérique**.

Des compétences développées en s'appuyant sur des **systèmes pluri-technologiques réels** et complexes répondant à des **besoins** de la société.

Une mise en perspective de l'enseignement pour former des ingénieurs **éthiques et responsables** qui seront aptes à relever les défis actuels et futurs de la société.

Une pédagogie variée avec entre autres des **projets collaboratifs** et de la classe inversée permettant de développer des compétences d'**autonomie** et de **communication** fondamentales pour les futurs ingénieurs scientifiques et techniques.

L'utilisation de **démarches d'ingénierie modernes** pour valider et améliorer les performances des systèmes.

L'**innovation** est l'ADN de cet enseignement en permanente évolution, avec par exemple l'**Ingénierie Numérique** (Big Data, Réseaux de communications, Intelligence Artificielle, ...)

Robot NAO

Extrait du cahier des charges :

«**requirément**»
Aider la personne à se lever
nd =1
Poids de la personne < 100kg
Temps d'exécution < 30s

The figure consists of three parts: a kinematic diagram, an electrical equivalent circuit, and a set of equations.

Kinematic Diagram: Shows a mechanism with a rotating disk (1) and a slider (2) on a curved guide (3). The disk has a radius R_1 and a moment of inertia J_1 . The slider has a mass m and a moment of inertia J_2 . The guide has a radius R_2 and a moment of inertia J_3 . The mechanism is driven by a voltage source E through a series of resistors $R_1, R_2, R_3, R_4, R_5, R_6, R_7, R_8, R_9, R_{10}, R_{11}, R_{12}, R_{13}, R_{14}, R_{15}, R_{16}, R_{17}, R_{18}, R_{19}, R_{20}, R_{21}, R_{22}, R_{23}, R_{24}, R_{25}, R_{26}, R_{27}, R_{28}, R_{29}, R_{30}, R_{31}, R_{32}, R_{33}, R_{34}, R_{35}, R_{36}, R_{37}, R_{38}, R_{39}, R_{40}, R_{41}, R_{42}, R_{43}, R_{44}, R_{45}, R_{46}, R_{47}, R_{48}, R_{49}, R_{50}, R_{51}, R_{52}, R_{53}, R_{54}, R_{55}, R_{56}, R_{57}, R_{58}, R_{59}, R_{60}, R_{61}, R_{62}, R_{63}, R_{64}, R_{65}, R_{66}, R_{67}, R_{68}, R_{69}, R_{70}, R_{71}, R_{72}, R_{73}, R_{74}, R_{75}, R_{76}, R_{77}, R_{78}, R_{79}, R_{80}, R_{81}, R_{82}, R_{83}, R_{84}, R_{85}, R_{86}, R_{87}, R_{88}, R_{89}, R_{90}, R_{91}, R_{92}, R_{93}, R_{94}, R_{95}, R_{96}, R_{97}, R_{98}, R_{99}, R_{100}$.

Electrical Equivalent Circuit: A circuit diagram showing a voltage source E connected to a series of resistors $R_1, R_2, R_3, R_4, R_5, R_6, R_7, R_8, R_9, R_{10}, R_{11}, R_{12}, R_{13}, R_{14}, R_{15}, R_{16}, R_{17}, R_{18}, R_{19}, R_{20}, R_{21}, R_{22}, R_{23}, R_{24}, R_{25}, R_{26}, R_{27}, R_{28}, R_{29}, R_{30}, R_{31}, R_{32}, R_{33}, R_{34}, R_{35}, R_{36}, R_{37}, R_{38}, R_{39}, R_{40}, R_{41}, R_{42}, R_{43}, R_{44}, R_{45}, R_{46}, R_{47}, R_{48}, R_{49}, R_{50}, R_{51}, R_{52}, R_{53}, R_{54}, R_{55}, R_{56}, R_{57}, R_{58}, R_{59}, R_{60}, R_{61}, R_{62}, R_{63}, R_{64}, R_{65}, R_{66}, R_{67}, R_{68}, R_{69}, R_{70}, R_{71}, R_{72}, R_{73}, R_{74}, R_{75}, R_{76}, R_{77}, R_{78}, R_{79}, R_{80}, R_{81}, R_{82}, R_{83}, R_{84}, R_{85}, R_{86}, R_{87}, R_{88}, R_{89}, R_{90}, R_{91}, R_{92}, R_{93}, R_{94}, R_{95}, R_{96}, R_{97}, R_{98}, R_{99}, R_{100}$ and a capacitor C_1 .

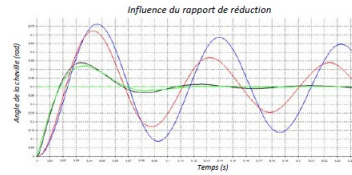
Equations:

$$\begin{aligned} \ddot{\varphi} &= a(\varphi) = R - R\ddot{\varphi} + L \frac{d\dot{\varphi}}{dt} \\ \ddot{\varphi} &= \ddot{\varphi}_m(\varphi) + \ddot{\varphi}_c(\varphi) - b\dot{\varphi} \\ \ddot{\varphi} &= K_1 \cdot i(t) \\ a(t) &= K_2 \cdot u(t) \end{aligned}$$

```

1 import time
2 import random
3 import sys
4 from math import sqrt
5
6 class MyCircle(MyCircleBase):
7     def __init__(
8         self,
9         center_x, center_y, radius,
10         color, stroke_color, stroke_width):
11         self.x = x
12         self.y = y
13         self.r = r
14         self.c = c
15         self.sc = stroke_color
16         self.sw = stroke_width
17
18     def __str__(self):
19         return "x: %f, y: %f, r: %f, c: %s, sc: %s, sw: %d" % (
20             self.x, self.y, self.r, self.c, self.sc, self.sw)
21
22     def __repr__(self):
23         return "x: %f, y: %f, r: %f, c: %s, sc: %s, sw: %d" % (
24             self.x, self.y, self.r, self.c, self.sc, self.sw)
25
26     def __eq__(self, other):
27         return self.x == other.x and self.y == other.y and self.r == other.r and self.c == other.c and self.sc == other.sc and self.sw == other.sw
28
29     def __neq__(self, other):
30         return not self.__eq__(other)
31
32     def __lt__(self, other):
33         return self.x < other.x and self.y < other.y and self.r < other.r and self.c < other.c and self.sc < other.sc and self.sw < other.sw
34
35     def __gt__(self, other):
36         return self.x > other.x and self.y > other.y and self.r > other.r and self.c > other.c and self.sc > other.sc and self.sw > other.sw
37
38     def __le__(self, other):
39         return self.x <= other.x and self.y <= other.y and self.r <= other.r and self.c <= other.c and self.sc <= other.sc and self.sw <= other.sw
40
41     def __ge__(self, other):
42         return self.x >= other.x and self.y >= other.y and self.r >= other.r and self.c >= other.c and self.sc >= other.sc and self.sw >= other.sw
43
44     def __add__(self, other):
45         return MyCircle(self.x + other.x, self.y + other.y, self.r + other.r, self.c + other.c, self.sc + other.sc, self.sw + other.sw)
46
47     def __sub__(self, other):
48         return MyCircle(self.x - other.x, self.y - other.y, self.r - other.r, self.c - other.c, self.sc - other.sc, self.sw - other.sw)
49
50     def __mul__(self, other):
51         return MyCircle(self.x * other.x, self.y * other.y, self.r * other.r, self.c * other.c, self.sc * other.sc, self.sw * other.sw)
52
53     def __div__(self, other):
54         return MyCircle(self.x / other.x, self.y / other.y, self.r / other.r, self.c / other.c, self.sc / other.sc, self.sw / other.sw)
55
56     def __mod__(self, other):
57         return MyCircle(self.x % other.x, self.y % other.y, self.r % other.r, self.c % other.c, self.sc % other.sc, self.sw % other.sw)
58
59     def __and__(self, other):
60         return MyCircle(self.x & other.x, self.y & other.y, self.r & other.r, self.c & other.c, self.sc & other.sc, self.sw & other.sw)
61
62     def __or__(self, other):
63         return MyCircle(self.x | other.x, self.y | other.y, self.r | other.r, self.c | other.c, self.sc | other.sc, self.sw | other.sw)
64
65     def __xor__(self, other):
66         return MyCircle(self.x ^ other.x, self.y ^ other.y, self.r ^ other.r, self.c ^ other.c, self.sc ^ other.sc, self.sw ^ other.sw)
67
68     def __lshift__(self, other):
69         return MyCircle(self.x << other.x, self.y << other.y, self.r << other.r, self.c << other.c, self.sc << other.sc, self.sw << other.sw)
70
71     def __rshift__(self, other):
72         return MyCircle(self.x >> other.x, self.y >> other.y, self.r >> other.r, self.c >> other.c, self.sc >> other.sc, self.sw >> other.sw)
73
74     def __pow__(self, other):
75         return MyCircle(self.x ** other.x, self.y ** other.y, self.r ** other.r, self.c ** other.c, self.sc ** other.sc, self.sw ** other.sw)
76
77     def __neg__(self):
78         return MyCircle(-self.x, -self.y, self.r, self.c, self.sc, self.sw)
79
80     def __pos__(self):
81         return MyCircle(self.x, self.y, self.r, self.c, self.sc, self.sw)
82
83     def __abs__(self):
84         return MyCircle(abs(self.x), abs(self.y), self.r, self.c, self.sc, self.sw)
85
86     def __negabs__(self):
87         return MyCircle(-abs(self.x), -abs(self.y), self.r, self.c, self.sc, self.sw)
88
89     def __posabs__(self):
90         return MyCircle(abs(self.x), abs(self.y), self.r, self.c, self.sc, self.sw)
91
92     def __negpos__(self):
93         return MyCircle(-abs(self.x), abs(self.y), self.r, self.c, self.sc, self.sw)
94
95     def __posneg__(self):
96         return MyCircle(abs(self.x), -abs(self.y), self.r, self.c, self.sc, self.sw)
97
98     def __negneg__(self):
99         return MyCircle(-abs(self.x), -abs(self.y), self.r, self.c, self.sc, self.sw)
100
101     def __pospos__(self):
102         return MyCircle(abs(self.x), abs(self.y), self.r, self.c, self.sc, self.sw)
103
104     def __negpospos__(self):
105         return MyCircle(-abs(self.x), abs(self.y), self.r, self.c, self.sc, self.sw)
106
107     def __posnegneg__(self):
108         return MyCircle(abs(self.x), -abs(self.y), self.r, self.c, self.sc, self.sw)
109
110     def __negnegneg__(self):
111         return MyCircle(-abs(self.x), -abs(self.y), self.r, self.c, self.sc, self.sw)
112
113     def __pospospos__(self):
114         return MyCircle(abs(self.x), abs(self.y), self.r, self.c, self.sc, self.sw)
115
116     def __negpospospos__(self):
117         return MyCircle(-abs(self.x), abs(self.y), self.r, self.c, self.sc, self.sw)
118
119     def __posnegnegneg__(self):
120         return MyCircle(abs(self.x), -abs(self.y), self.r, self.c, self.sc, self.sw)
121
122     def __negnegnegneg__(self):
123         return MyCircle(-abs(self.x), -abs(self.y), self.r, self.c, self.sc, self.sw)
124
125     def __pospospospos__(self):
126         return MyCircle(abs(self.x), abs(self.y), self.r, self.c, self.sc, self.sw)
127
128     def __negpospospospos__(self):
129         return MyCircle(-abs(self.x), abs(self.y), self.r, self.c, self.sc, self.sw)
130
131     def __posnegnegnegneg__(self):
132         return MyCircle(abs(self.x), -abs(self.y), self.r, self.c, self.sc, self.sw)
133
134     def __negnegnegnegneg__(self):
135         return MyCircle(-abs(self.x), -abs(self.y), self.r, self.c, self.sc, self.sw)
136
137     def __pospospospospos__(self):
138         return MyCircle(abs(self.x), abs(self.y), self.r, self.c, self.sc, self.sw)
139
140     def __negpospospospospos__(self):
141         return MyCircle(-abs(self.x), abs(self.y), self.r, self.c, self.sc, self.sw)
142
143     def __posnegnegnegnegneg__(self):
144         return MyCircle(abs(self.x), -abs(self.y), self.r, self.c, self.sc, self.sw)
145
146     def __negnegnegnegnegneg__(self):
147         return MyCircle(-abs(self.x), -abs(self.y), self.r, self.c, self.sc, self.sw)
148
149     def __pospospospospospos__(self):
150         return MyCircle(abs(self.x), abs(self.y), self.r, self.c, self.sc, self.sw)
151
152     def __negpospospospospospos__(self):
153         return MyCircle(-abs(self.x), abs(self.y), self.r, self.c, self.sc, self.sw)
154
155     def __posnegnegnegnegnegneg__(self):
156         return MyCircle(abs(self.x), -abs(self.y), self.r, self.c, self.sc, self.sw)
157
158     def __negnegnegnegnegnegneg__(self):
159         return MyCircle(-abs(self.x), -abs(self.y), self.r, self.c, self.sc, self.sw)
160
161     def __pospospospospospospos__(self):
162         return MyCircle(abs(self.x), abs(self.y), self.r, self.c, self.sc, self.sw)
163
164     def __negpospospospospospospos__(self):
165         return MyCircle(-abs(self.x), abs(self.y), self.r, self.c, self.sc, self.sw)
166
167     def __posnegnegnegnegnegnegneg__(self):
168         return MyCircle(abs(self.x), -abs(self.y), self.r, self.c, self.sc, self.sw)
169
170     def __negnegnegnegnegnegnegneg__(self):
171         return MyCircle(-abs(self.x), -abs(self.y), self.r, self.c, self.sc, self.sw)
172
173     def __pospospospospospospospos__(self):
174         return MyCircle(abs(self.x), abs(self.y), self.r, self.c, self.sc, self.sw)
175
176     def __negpospospospospospospospos__(self):
177         return MyCircle(-abs(self.x), abs(self.y), self.r, self.c, self.sc, self.sw)
178
179     def __posnegnegnegnegnegnegnegneg__(self):
180         return MyCircle(abs(self.x), -abs(self.y), self.r, self.c, self.sc, self.sw)
181
182     def __negnegnegnegnegnegnegnegneg__(self):
183         return MyCircle(-abs(self.x), -abs(self.y), self.r, self.c, self.sc, self.sw)
184
185     def __pospospospospospospospospos__(self):
186         return MyCircle(abs(self.x), abs(self.y), self.r, self.c, self.sc, self.sw)
187
188     def __negpospospospospospospospospos__(self):
189         return MyCircle(-abs(self.x), abs(self.y), self.r, self.c, self.sc, self.sw)
190
191     def __posnegnegnegnegnegnegnegnegneg__(self):
192         return MyCircle(abs(self.x), -abs(self.y), self.r, self.c, self.sc, self.sw)
193
194     def __negnegnegnegnegnegnegnegnegneg__(self):
195         return MyCircle(-abs(self.x), -abs(self.y), self.r, self.c, self.sc, self.sw)
196
197     def __pospospospospospospospospospos__(self):
198         return MyCircle(abs(self.x), abs(self.y), self.r, self.c, self.sc, self.sw)
199
200     def __negpospospospospospospospospospos__(self):
201         return MyCircle(-abs(self.x), abs(self.y), self.r, self.c, self.sc, self.sw)
202
203     def __posnegnegnegnegnegnegnegnegnegneg__(self):
204         return MyCircle(abs(self.x), -abs(self.y), self.r, self.c, self.sc, self.sw)
205
206     def __negnegnegnegnegnegnegnegnegnegneg__(self):
207         return MyCircle(-abs(self.x), -abs(self.y), self.r, self.c, self.sc, self.sw)
208
209     def __pospospospospospospospospospospos__(self):
210         return MyCircle(abs(self.x), abs(self.y), self.r, self.c, self.sc, self.sw)
211
212     def __negpospospospospospospospospospospos__(self):
213         return MyCircle(-abs(self.x), abs(self.y), self.r, self.c, self.sc, self.sw)
214
215     def __posnegnegnegnegnegnegnegnegnegnegneg__(self):
216         return MyCircle(abs(self.x), -abs(self.y), self.r, self.c, self.sc, self.sw)
217
218     def __negnegnegnegnegnegnegnegnegnegnegneg__(self):
219         return MyCircle(-abs(self.x), -abs(self.y), self.r, self.c, self.sc, self.sw)
220
221     def __pospospospospospospospospospospospos__(self):
222         return MyCircle(abs(self.x), abs(self.y), self.r, self.c, self.sc, self.sw)
223
224     def __negpospospospospospospospospospospospos__(self):
225         return MyCircle(-abs(self.x), abs(self.y), self.r, self.c, self.sc, self.sw)
226
227     def __posnegnegnegnegnegnegnegnegnegnegnegneg__(self):
228         return MyCircle(abs(self.x), -abs(self.y), self.r, self.c, self.sc, self.sw)
229
230     def __negnegnegnegnegnegnegnegnegnegnegnegneg__(self):
231         return MyCircle(-abs(self.x), -abs(self.y), self.r, self.c, self.sc, self.sw)
232

```



Présentation des Sciences de l'Ingénieur

Pour aller en première sciences de l'ingénieur, il est souhaitable d'avoir :

1- un projet professionnel dans le domaine des sciences de l'ingénieur

- ingénieur en informatique
- ingénieur en mécanique
- ingénieur en aéronautique
- Etc.

ou un projet d'études dans les sciences de l'ingénieur

- BTS, BUT, Bachelor, Master, etc.

2- Avoir un niveau en **Math** et en physique correct

Malgré le fait qu'on travaille sur des systèmes (produits grand public, didactisés), on a quand même besoin de « manipuler » des concepts (mathématiques et physiques).

Nb : Il ne faudra pour autant ne pas négliger :

- le français : - rédaction des projets (première et en terminale)
- l'anglais : - informatique (python et recherche d'information sur le net)

Présentation des Sciences de l'Ingénieur

Classe de première Sciences de l'Ingénieur

- 4 heures/semaine avec :
 - des activités pratiques (sur des systèmes présents dans le laboratoire)
 - des cours
 - 1 challenge de 12h durant lequel les élèves réalisent un projet.

Classe de terminale Sciences de l'Ingénieur

- 6 heures/semaine avec :
 - des activités pratiques (sur des systèmes présents dans le laboratoire)
 - des cours
 - 1 projet de 48h sur lequel les élèves peuvent s'appuyer pour le grand oral

NB : En plus des 6h de SI en terminale, les élèves ont aussi 2h de physique obligatoire. (Donc ils n'abandonnent pas la physique)

Présentation des Sciences de l'Ingénieur

Idées reçues sur les Sciences de l'Ingénieur

Les Sciences de l'Ingénieur ne sont que dans les filières technologiques ?

FAUX Les élèves qui s'orientent en 1ère puis Terminale de la voie générale peuvent choisir la spécialité Sciences de l'Ingénieur : c'est une spécialité idéale si vous voulez préparer au mieux votre parcours vers les métiers d'ingénieur-e-s ou de technicien-ne-s par exemple. Les Sciences de l'Ingénieur sont aussi présentes dans les filières technologiques appelées STI2D avec une pédagogie adaptée.

Les Sciences de l'Ingénieur sont des sciences appliquées ?

VRAI Les Sciences de l'Ingénieur étudient les performances des systèmes innovants qui nous entourent, à travers des résolutions scientifiques rigoureuses et des concepts spécifiques à la discipline. En complément des Travaux Dirigés en classe, les projets en équipe organisés dans les laboratoires de Sciences de l'Ingénieur permettent aussi d'analyser concrètement et en temps réel les performances des systèmes, ou de concevoir des prototypes innovants répondant à un cahier des charges. Bref, si vous aimez les sciences et la résolution de problèmes scientifiques appliqués sur des systèmes innovants, vous aimerez les Sciences de l'Ingénieur !

En Sciences de l'Ingénieur, on ne fait que de la mécanique ?

FAUX Les systèmes innovants étudiés en Sciences de l'Ingénieur suivent les évolutions technologiques du monde qui nous entoure et combinent des technologies complexes et des champs d'études variés : électronique, informatique, analyse des mouvements et des actions mécaniques, résistance et choix des matériaux, réseaux de communication, intelligence artificielle.... On retrouve dans l'enseignement des Sciences de l'Ingénieur toutes les compétences qui seront nécessaires au futur technicien ou ingénieur, pour appréhender les systèmes pluri-technologiques complexes actuels, et innover dans cet environnement.

Les filles apprécient autant les Sciences de l'Ingénieur que les garçons ?

VRAI Les filles s'épanouissent tout autant dans les filières comportant des Sciences de l'Ingénieur que les garçons, et y réussissent très bien! Les carrières scientifiques féminines sont d'ailleurs grandement valorisées par les entreprises, qui cherchent aujourd'hui davantage de mixité dans leurs services. Pour plus d'information à ce sujet, consulter la page

[F.I.R.S.T : Femmes & Ingénieures - Réussir en Sciences et Technologies.](#)