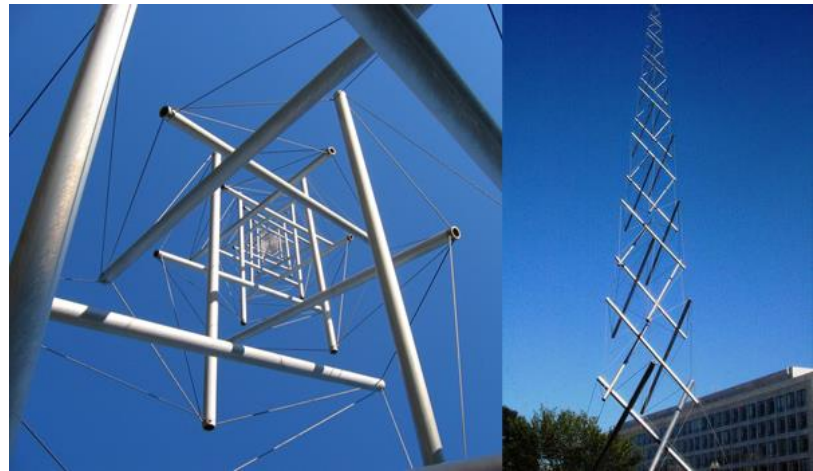


Étude et optimisation d'un système de tenségrité.

Plan d'étude

Une définition
Problématique
Les modules élémentaires
Réalisation de la table
Exploitations des données
Conclusion



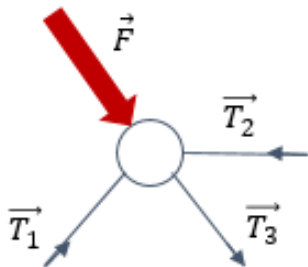
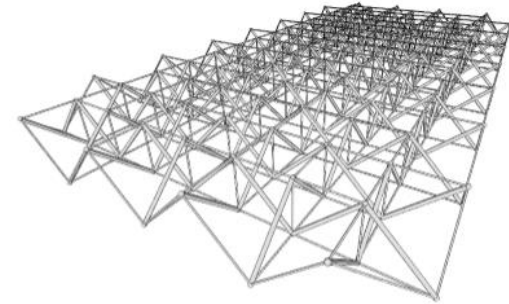
Tensegrity

Tensile

Integrity

Tenségrité :

- Etat d'autocontrainte stable.
- Composant comprimés dans un ensemble de composant tendus.



Stabilité si :

$$\sum_{i=1}^n \vec{T}_i + \vec{F} = \vec{0}$$

Etat d'autocontrainte :

- Etat apportant un équilibre stable et une certaine rigidité au système.
- Géométrie et autocontrainte sont étroitement liées.



Définition

Problématique

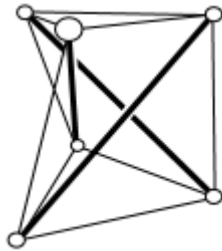
Les modules
élémentaires

Réalisation de
la table

Exploitation
des données

**Peut-on prévoir le comportement
dynamique d'un système de tenségrité
sous une charge extérieure?**

Les différents types de géométries :



Module simplex



Module quadruplex

Equation d'équilibre : $[A] \cdot \{t\} = \{0\}$

$[A]$: Matrice d'équilibre dépendante des paramètres géométriques

$\{t\}$: Vecteur d'autocontrainte

$\{0\}$: Vecteur nul

Les équations de Maxwell :

Pour un système à E éléments et n nœud :

$$E = 2n - 3 \text{ (En 2D)}$$

$$E = 3n - 6 \text{ (En 3D)}$$

Exemple en 2D :

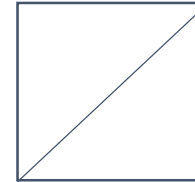


$$E = 4$$

$$n = 4$$

$E = 2n - 3$ non vérifiée,
Système **hyperstatique**

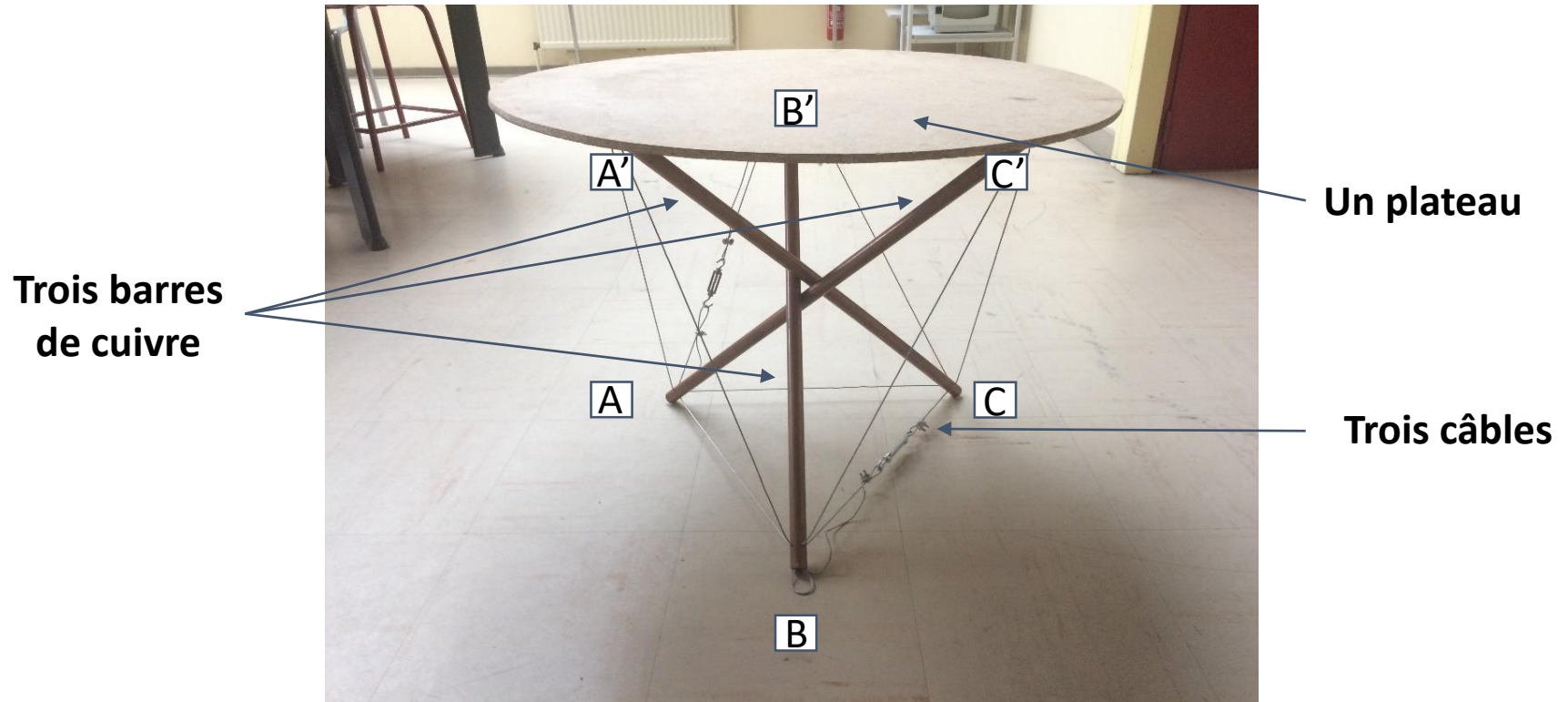
Ajout d'une
diagonale



$$E = 5$$

$$n = 4$$

$E = 2n - 3$ vérifiée,
Système **isostatique**





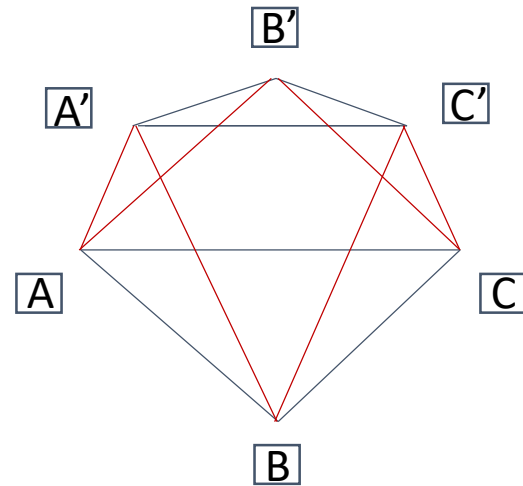
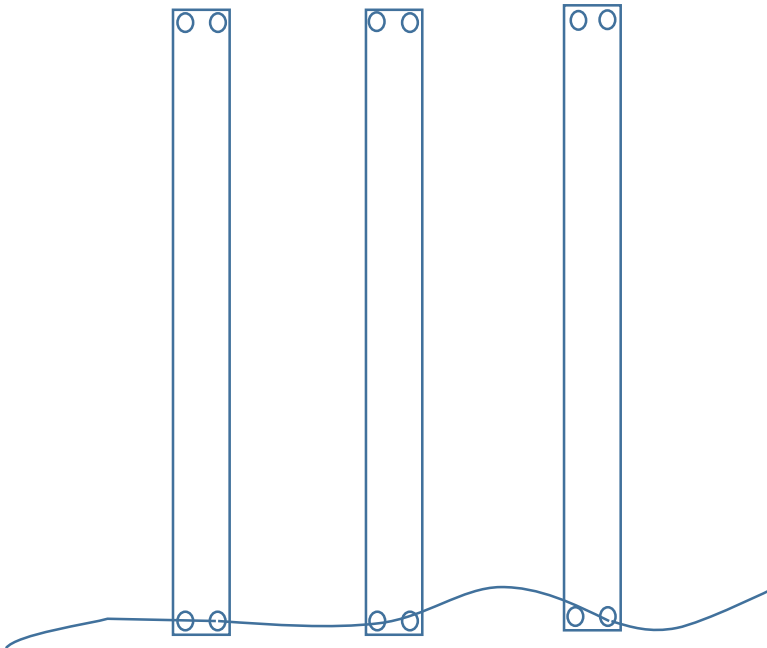
Pour notre table nous avons :

$$E = 3+3+6$$

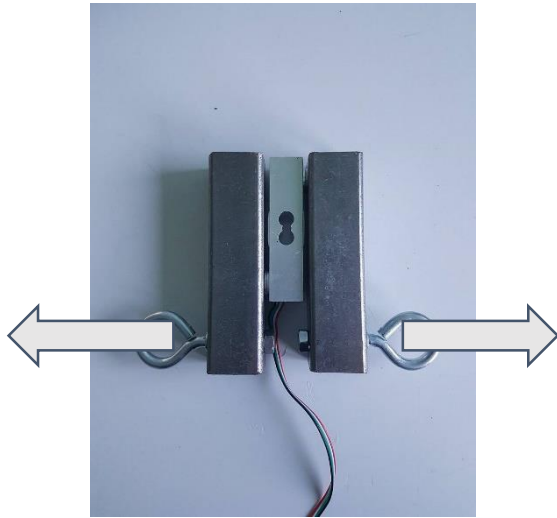
$$n = 6$$

Donc $E = 3n-6$ est vérifiée,

Ainsi le système est **isostatique**.



- Deux triangles opposés
- Une couronne reliant les parties basses et hautes des barres



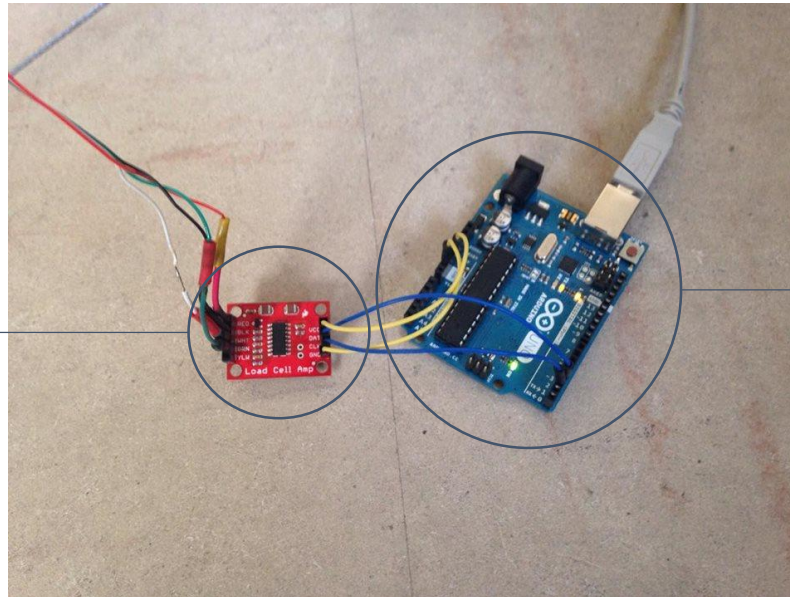
Modification de capteur de
force en capteur de force de
traction

Nous avons fixé un
capteur sur chaque câble



Pont de Wheatstone et carte Arduino :

Pont de Wheatstone



Carte Arduino

Réduit les variations parasites,
augmente la précision.



Charge maximale : 65Kg à cause du plateau

Définition

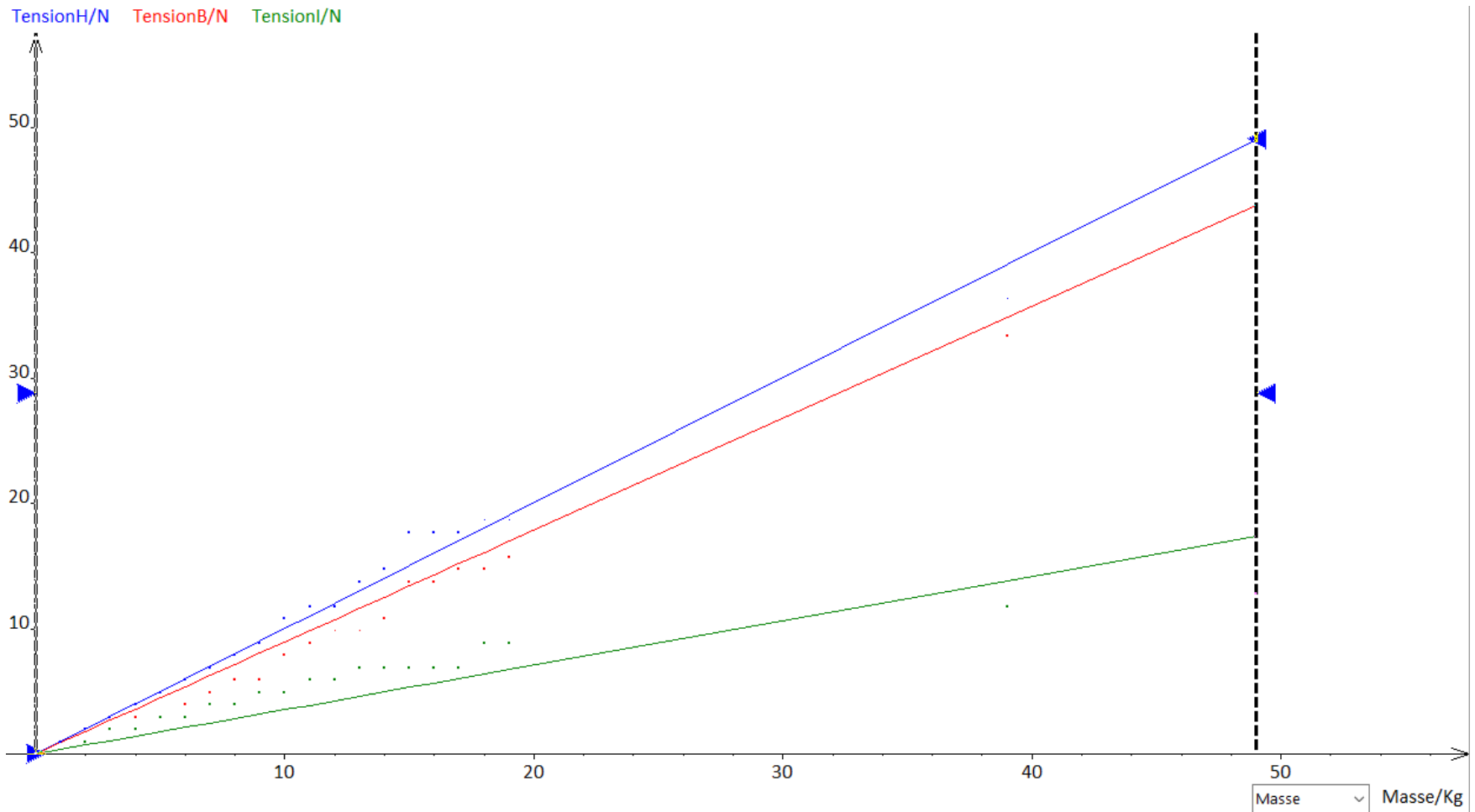
Problématique

Les modules
élémentaires

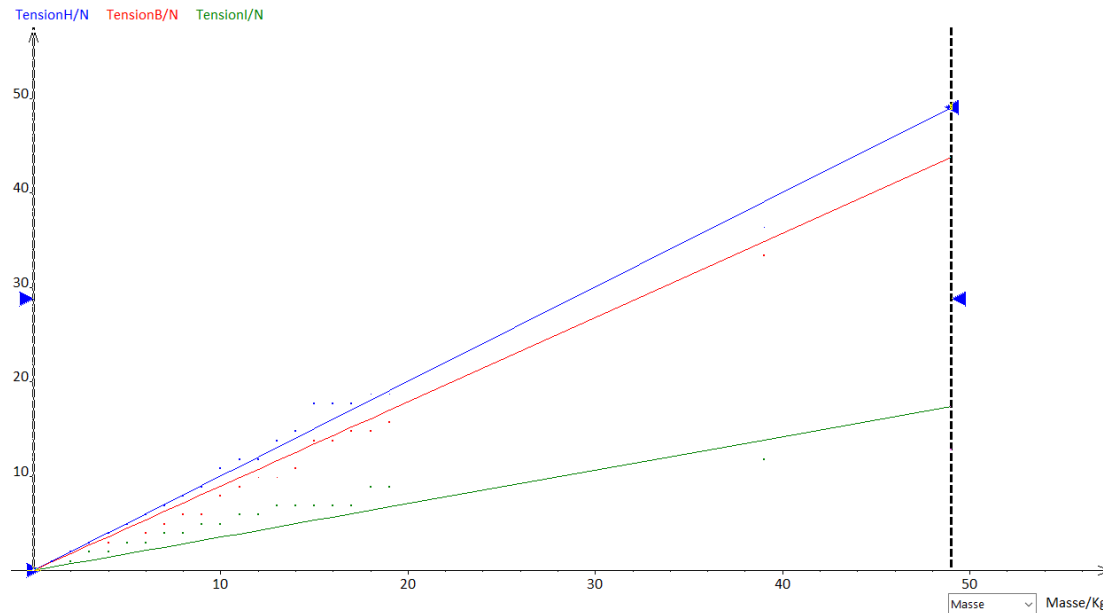
Réalisation de
la table

Exploitation
des données

Masse (Kg)	1	2	5	8	10	15	19	39	49
Tension Haut (Kg)	0,1	0,2	0,5	0,8	1,1	1,8	1,9	3,4	5,0
Tension Bas (Kg)	0,1	0,1	0,3	0,6	0,8	1,3	1,6	3,7	5,0
Tension Int (Kg)	0,1	0,1	0,3	0,4	0,5	0,7	0,9	1,2	1,3



Tension dans les câbles en fonction de la
masse présente sur le plateau de la table



Rapport d'environ **1/10** entre la masse sur la table et la masse mesurée dans les câbles du haut et du bas.

Le câble intermédiaire est, lui encore moins touché par l'effort représenté.

Conclusion



Les efforts se répartissent correctement, la table peut supporter une charge très lourde.

Dans notre cas c'est le plateau qui empêchait de mettre une charge supérieure.



Avantages de cette table de tensegrité : structure légère mais tout aussi stable.

Annexe

```
#include "HX711.h"
#define DOUT 3
#define CLK 2

HX711 scale(DOUT, CLK);

float calibration_factor = -7050;

void setup() {
  Serial.begin(9600);
  Serial.println("HX711 calibration sketch");
  Serial.println("Remove all weight from scale");
  Serial.println("After readings begin, place known weight on scale");
  Serial.println("Press + or a to increase calibration factor");
  Serial.println("Press - or z to decrease calibration factor");

  scale.set_scale();
  scale.tare(); //Reset the scale to 0

  long zero_factor = scale.read_average();
  Serial.print("Zero factor: "); //On tare à chaque lancement
  Serial.println(zero_factor);
}

void loop() {

  scale.set_scale(calibration_factor);

  Serial.print("Reading: ");
  Serial.print(scale.get_units()*0.453592*0.33333, 1); //On passe du lbs au Kg
  Serial.print(" kg");
  Serial.print(" calibration_factor: ");
  Serial.print(calibration_factor);
  Serial.println();

  if(Serial.available())
  {
    char temp = Serial.read();
    if(temp == '+' || temp == 'a')
      calibration_factor += 10;
    else if(temp == '-' || temp == 'z')
      calibration_factor -= 10;
  }
}
```

