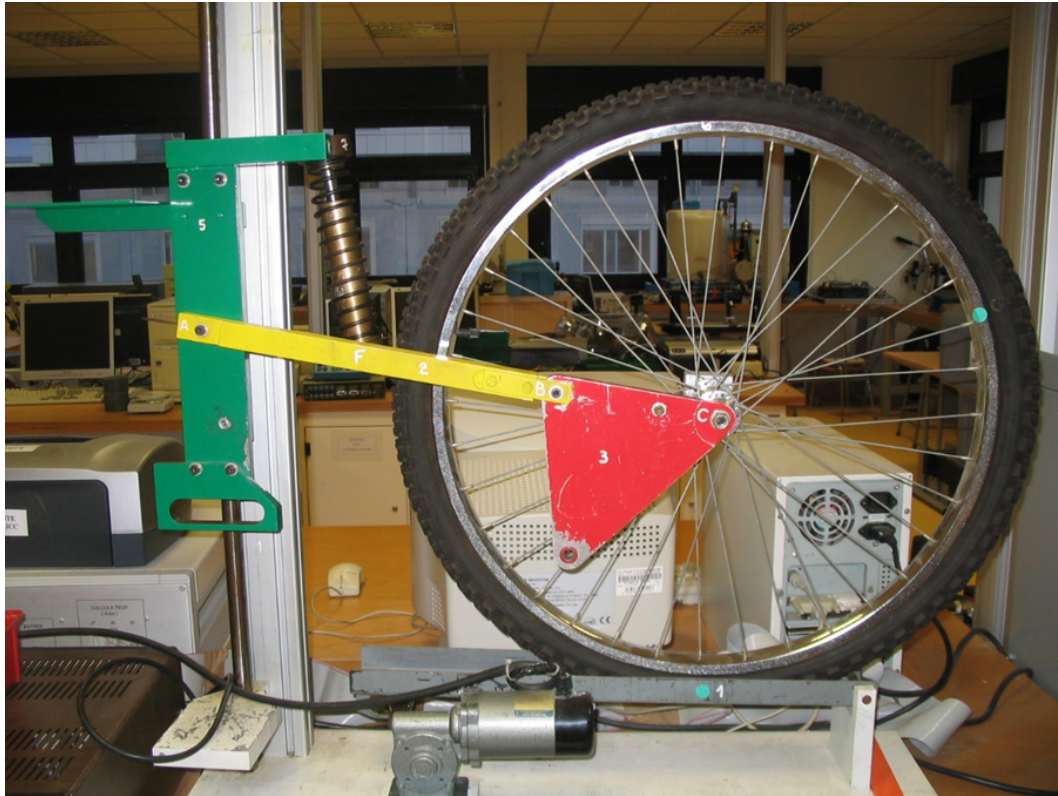




Suspension BMW



Objectif : déterminer la loi entrée-sortie de la suspension BMW selon trois approches :

- théorique ;
- expérimentale ;
- numérique.

Comparer les résultats et discuter les sources des écarts éventuels.

1 | Mise en situation

La firme BMW utilise deux types de suspension sur ces motos (présentées Figure 1) :

- Une suspension « classique » à bras oscillant nommée MONOLEVER, utilisée sur la K1200S.
- Une suspension à parallélogramme déformable nommée PARALEVER utilisée sur la R 1200 RT.



FIGURE 1 – A gauche : BLW K 1200 S à suspension Monolever. A droite : BMW R 1200 RT à suspension Paralever

La maquette utilisée au labo reprend exactement la géométrie à l'échelle 1 de la moto.

2 | Étude théorique

2.1 Modélisation

Le schéma cinématique issue de la modélisation est donnée Figure 2.

On pose

Paramètres	Valeur en mm
$\vec{OA} = y_A(t)\vec{y}_1$	
$\vec{IC} = R\vec{y}_1$	$R = 325$
$\vec{AC} = L_C\vec{x}_2$	$L_C = 515$
$\vec{AF} = L_F\vec{x}_2$	$L_F = 155$
$\vec{AK} = x_K\vec{x}_1 + y_K\vec{y}_1$	$x_K = 118; y_K = 170$
$\vec{FK} = \mu(t)\vec{y}_7$	

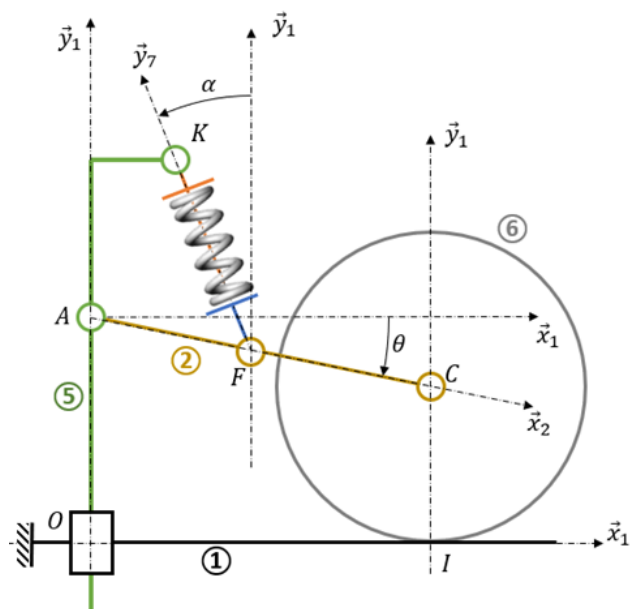


FIGURE 2 – Schéma cinématique de la suspension BMW en monolever

2.2 Étude cinématique

Question 1 : Réaliser le graphe des liaisons et la ou les figure(s) de changement de base.

Question 2 : Réaliser une fermeture géométrique permettant d'exprimer l'angle α en fonction de l'angle θ et des constantes du système, ainsi qu'une expression de μ en fonction de θ et des constantes du

système.

Question 3 : Réaliser une fermeture géométrique permettant d'exprimer la longueur y_A en fonction de l'angle θ et des constantes du système

2.3 Détermination avec Python

-  Ouvrir le fichier **TPC1-6 - BMW - Etudiant.py**.
-  Compléter la fonction `Theta_to_mu_th` qui prend en argument l'angle θ et renvoie la valeur de μ .
-  Faire de même pour la fonction `Theta_to_Ya_th` qui prend en argument l'angle θ et renvoie la valeur de y_A .

3 | Étude expérimentale

3.1 Manipulation

-  A l'aide d'un grand réglé, faire une série de mesures en positionnant des masses différentes sur le plateau de la pièce 5 (verte).
-  Pour chaque masse relever :
 - La hauteur du point A ;
 - L'angle θ à l'aide d'un rapporteur (ou mieux) ;
 - La distance FK .
-  Utiliser des masses à utiliser sont :
 - Une masse utilisée pour le robot Ericc
 - Les masses utilisées pour Maxpid
 - Tout autre objet ne risquant pas de tomber ; une balance est mise à disposition pour mesurer leur masse.
-  Réaliser 10 mesures, pour 10 masses différentes (totales) sur le plateau de la pièce 5 en essayant d'espacer le plus possible les points de mesure. *Si nécessaire, il est possible d'appuyer avec la main pour obtenir de nouveaux points si les masses ne le permettent pas.*
-  Il peut être intéressant de refaire les mesures en retirant les masses afin de les doubler et d'augmenter la fiabilité des mesures.

3.2 Tracer des courbes avec Python

-  Ouvrir le fichier **TPC1-6 - BMW - Etudiant.py**.
-  Enregistrer les valeurs expérimentales dans trois listes `Theta_liste`, `mu_exp_liste` et `Ya_exp_liste`.
-  Tracer, sur deux figures différentes, les longueurs expérimentales y_A en fonction de θ et μ en fonction de θ .
-  En utilisant la même liste `Theta_liste`, tracer sur ces mêmes figures et à l'aide des fonctions précédemment créées, les évolutions des longueurs théoriques $y_{A_{th}}$ et μ_{th} en fonction de θ .

4 | Étude numérique

4.1 Assemblage

-  Ouvrir le fichier **BMW Monolever.sldasm**.
-  La liaison entre le support et le bâti est déjà réalisée (plus difficile de faire une glissière avec ces formes). De même une coaxiale est déjà présente pour le ressort.
-  Avec les contraintes, réaliser les **liaisons pivots** entre :
 - Le support et le bras ;
 - Le support et une coupelle ;
 - Le bras et l'autre coupelle ;

- Le bras et la roue.

 Réaliser une **liaison appui plan** entre la roue et le bâti (les liaisons ponctuelles étant plus délicates en simulation).

4.2 Animation

 Si ce n'est pas déjà le cas, activer Meca3D.

 Sous Meca3D, faire un clic droit sur **Mécanisme** puis **Construction automatique**.

 Si les contraintes sont bien définies, il devrait y avoir 2 liaisons appui plan, 2 liaisons pivot glissant et 4 liaisons pivot.

 Pour que l'angle θ soit bien défini, changer la position initiale de sorte que $\theta = 0$. *Par exemple en ajoutant une contrainte de parallélisme ou d'angle.*

 Réaliser le **Calcul mécanique**. Pour cela faire une **Étude géométrique** en faisant déplacer la pivot glissant entre le bâti et le support de la distance déterminée par les expérimentations. Prendre un temps de simulation adéquat et 10 positions par exemple.

 Lancer le calcul et vérifier le bon fonctionnement via l'animation.

 Ajouter la courbe du déplacement de la pivot glissant entre la bâti et le support en fonction de θ puis exporter les données au format texte.

 Faire de même pour la pivot glissant représentant l'allongement du ressort.

4.3 Tracer des courbes avec Python

Question 4 : Dans le script TPC1-6 - BMW - Etudiant.py :

 Dans le programme principal, section Simulation, appeler la fonction `donneesSldW` afin extraire les données issues de Solidworks.

 Tracer la courbe simulée de y_A en fonction de θ sur le même graphique que la théorique et l'expérimentale. *Attention aux unités et à ce qui est vraiment donné par Solidworks.*

 Tracer la courbe simulée de μ en fonction de θ sur le même graphique que la théorique et l'expérimentale.

 Dans l'onglet **Evaluer**, il est possible d'utiliser l'outil **Mesurer** pour mesurer des distances dans la position initiale.