

RÉPUBLIQUE DU SÉNÉGAL



ÉCOLE POLYTECHNIQUE DE THIÈS

Gm. 0362

PROJET DE FIN D'ETUDES

EN VUE DE L'OBTENTION DU DIPLOME D'INGÉNIEUR DE CONCEPTION

TITRE CONCEPTION ET REALISATION D'UN LOGICIEL
POUR L'ANALYSE ET LA SIMULATION
DES MECANISMES PLANS..

DATE : JUIN 1990

AUTEUR : Jérôme AVODAGBE
DIRECTEUR : Mouhamadou L. DIALLO
CO-DIRECTEUR : Jean Claude WARMOES

REPUBLIQUE DU SENEGAL

ECOLE POLYTECHNIQUE DE THIES

DEPARTEMENT DE GENIE MECANIQUE.

CONCEPTION ET REALISATION D'UN LOGICIEL
POUR L'ANALYSE ET LA SIMULATION DES
MECANISMES PLANS.

Projet de fin d'étude
en vue de l'obtention du diplôme
d'ingénieur de conception.

Présenté par: Jérôme AVODAGBE.

Sous la direction des professeurs:

Mohamadou L. DIALLO

Jean-Claude WARMOES

JUIN 1990

A mes parents...

Au Père Maurice ACHARD...

A tous ceux qui ont contribué à ma formation...

R E M E R C I E M E N T S .

Comme tout travail scientifique, le présent rapport n'a pas été élaboré en vase clos. Aussi, voudrions nous remercier tous ceux qui ont contribué à sa réalisation.

Aux professeurs L.M.DIALLO et J.C.WARMOES qui nous ont encadré nous témoignons notre gratitude.

Que toutes les personnes qui ont mis des moyens à notre disposition, ou qui nous ont prodigué des conseils trouvent ici l'expression de notre sincère reconnaissance.

TABLES DES MATIERES.

Sommaire.....	i
Liste des figures.....	ii
Liste des symboles.....	iv
INTRODUCTION.....	2
PREMIERE PARTIE: Concepts généraux et analyse des mécanismes plans.....	4
Chapitre 1: Définitions.....	4
1.1 Mécanisme.....	4
1.2 Membrure.....	4
Chapitre 2: Degré de liberté	6
2.1 Définition.....	6
2.2 Détermination du degré de liberté d'un ensemble de membrures.....	6
2.3 Exemple	7
2.4 Méthode de calcul	8
Chapitre 3: Mécanismes à quatre barres.....	7
3.1 Définition	9
3.2 Classification.....	9
3.2.1 Suivant Grashoff.....	10
3.2.2 Autres types de mécanisme à quatre membrures.....	10

Chapitre 4: Position limite et point mort.....	14
4.1 Positions pour un M4M.....	14
4.1.1 Position limite.....	14
4.1.2 Point mort.....	16
4.2 Positions pour un MC.....	18
4.2.1 Détermination de l'excentricité....	18
4.2.2 Position limite.....	19
4.2.3 Point mort.....	20
4.3 Positions pour un MCI.....	20
Chapitre 5: Angle de transmission.....	22
5.1 Angle de transmission pour un M4M.....	22
5.2 Angle de transmission pour un MC.....	24
5.3 Angle de transmission pour un MCI.....	25
 DEUXIEME PARTIE: Etude cinématique des mécanismes plans.....	 27
 Chapitre 1: Analyse cinématique des mécanismes M4M.....	 29
1.1 Positions dans les M4M.....	29
1.2 Vitesses dans les M4M	31
1.3 Accélérations dans les M4M.....	32
Chapitre 2: Analyse cinématique des mécanismes MC	34
2.1 Positions dans les MC.....	34
2.2 Vitesses dans les MC.....	35
2.3 Accélérations dans les MC.....	36

Chapitre 3: Analyse cinématique des mécanismes MCI.....	38
3.1 Positions des mécanismes MCI.....	38
3.2 Vitesses dans les MCI.....	39
3.3 Accélérations dans les MCI.....	40
 TROISIEME PARTIE: Présentation du progiciel SAMP1.....	 42
Chapitre 1: Exposé du problème.....	42
1. Introduction.....	42
2. Définition du problème.....	42
3. Analyse du problème.....	43
Chapitre 2: Mode d'utilisation du progiciel.....	51
1. Procédure de mise en oeuvre.....	51
2. Forme des données.....	51
3. Forme des résultats.....	51
Chapitre 3: Algorithmes.....	52
1 Les identificateurs.....	52
2 Structure des opérations.....	52
3 Exemple et mode d'utilisation de SAMP1.....	56
Chapitre 4: Analyse des méthodes et des résultats.....	77
 CONCLUSION ET RECOMMANDATIONS.....	 79
 ANNEXES.....	 81

LISTE DES FIGURES

Figure 1: chaine de membrures.....	7
Figure 2: chaine de membrures.....	7
Figure 3: manivelle-coulisseau.....	11
Figure 4: manivelle-coulisseau.....	11
Figure 5: manivelle-coulisseau inversé type 1.....	12
Figure 6: manivelle-coulisseau inversé	12
Figure 7: manivelle-coulisseau inversé type 2.....	12
Figure 8: coulisseau-coulisseau.....	13
Figure 9: position-limite 1 du M4M.....	15
Figure 10: position-limite 2 du M4M.....	15
Figure 11: point-mort 1 du M4M.....	17
Figure 12: point-mort 2 du M4M.....	17
Figure 13: position-limite du MC.....	18
Figure 14: point-mort du MC.....	18
Figure 15: position-limite 1 du MCI.....	21
Figure 16: position-limite 2 du MCI.....	21
Figure 17: angle de transmission du M4M.....	22
Figure 18: angle de transmission du MC.....	24
Figure 19: angle de transmission du MCI type 1.....	25
Figure 20: angle de transmission du MCI type 2.....	25
Figure 21: M4M.....	28
Figure 22: MC.....	34
Figure 23: MCI.....	38
Figure 24: M4M.....	43
Figure 25: M4M.....	43

Figure 26: M4M.....	44
Figure 27: M4M.....	45
Figure 28: M4M.....	45

S O M M A I R E

Dans ce projet nous allons élaborer un outil informatique pour l'étude des mécanismes plans.

Cet instrument aide le concepteur dans la réalisation, la vérification et le choix des mécanismes de base qui interviennent dans nombre de domaines de la vie courante. Dans la simulation et l'analyse nous utiliserons des méthodes comme celles des nombres complexes et des techniques graphiques.

Le programme mis au point permet de réaliser un schéma du mécanisme, d'étudier sa mobilité, de prédire la transformation du mouvement. Il détermine également les positions critiques et étudie la transmission et l'efficacité au cours du mouvement. Il sert en outre à faire l'analyse cinématique. Il constitue un simulateur du mouvement. Il peut montrer également la courbe de bielle lorsqu'on désire coupler un mécanisme à un autre pour réaliser des mécanismes complexes.

Le comportement des mécanismes est en réalité très complexe. Ce travail nous a montré que ces mécanismes ne peuvent pas occuper des positions quelconques au cours du mouvement. Ce dernier obéit à des lois et ne se produira pas pour certains paramètres sur la membrure motrice. Ces paramètres incompatibles entraîneraient dans la réalité la ruine du dispositif.

LISTE DES SYMBOLES

M4M: mécanisme à quatre membrure de longueur déterminée

MC: manivelle-coulisseau ou balancier coulisseau

MCI: manivelle-coulisseau inversé

θ_2 : position angulaire de la membrure d'entrée

θ_{2l} : position angulaire de la membrure d'entrée à la position limite

θ_{20} : position angulaire de la membrure d'entrée au point mort

INTRODUCTION

Avec les progrès considérables faits dans le design des instruments, les contrôles automatiques, les équipements automatisés, l'étude des mécanismes prend une nouvelle ampleur. Les mécanismes font partie de la branche de la mécanique qui concerne la cinématique des cames, des engrenages et des mécanismes à manivelle. Ces derniers se retrouvent dans les moteurs à combustion interne, les machines outils, les broyeurs et concasseurs de roches, les pompes etc...

Dans les dispositifs comme les instruments et les contrôles automatiques il s'avère important d'avoir des mouvements corrects et précis. La puissance dissipée par les divers éléments devra dès lors être aussi faible que possible, pour permettre aux composantes de suivre le mouvement donné à l'entrée, les déformations restant minimales. Cependant, il existe des mécanismes dont la conception ne s'arrête pas à l'analyse cinématique. Il faudra étudier non seulement la manière dont les composantes se déplacent, mais aussi les forces qui les actionnent pour faire le dimensionnement.

De nos jours, on ne peut plus se contenter d'avoir seulement un dispositif qui se met en mouvement. La précision et l'exactitude du mouvement deviennent des contraintes importantes du design. Aussi convient-il de doter le concepteur d'outil adéquat. Une approche assistée par ordinateur au design des mécanismes plans devra être envisagée à l'heure actuelle où l'ingénierie connaît une orientation informatique. Des études théoriques à la réalisation

du prototype, on disposera alors d'un instrument de simulation. Voilà le but que vise ce projet de fin d'études: la conception d'un progiciel de simulation et d'analyse des mécanismes plans. Il s'agit de mettre au point un programme permettant de faire l'analyse et l'étude cinématique des mécanismes plans.

Pour l'étude cinématique, nous utiliserons des méthodes analytiques et des méthodes graphiques aussi bien pour la détermination des positions que des vitesses et accélérations linéaires et angulaires. Dans l'analyse nous nous occuperons entre autres du nombre de degré de liberté, de l'identification du type de transformation de mouvement et de la définition des positions limites des points morts et de l'angle de transmission. Après avoir présenté dans une première partie les concepts fondamentaux relatifs aux mécanismes plans, nous ferons l'étude cinématique de certains mécanismes. Dans la troisième partie nous présenterons le progiciel SAMP1.

**PREMIERE PARTIE: ANALYSE ET CONCEPTS DE BASE DES
MECANISMES PLANS**

Chapitre 1: Définitions

Dans ce chapitre nous définirons un certain nombre de concepts qui seront utilisés dans notre étude.

1.1 Mécanisme:

Sous ce nom est désigné un ensemble de corps rigides ou flexibles disposés de façon à contrôler ou à produire un certain mouvement.

1.2 Membrure:

Nous appellerons membrure, un corps rigide ayant un ou plusieurs jointures grâce auxquelles il sera accouplé à d'autres composants du mécanisme pour produire un certain mouvement. Dans notre étude, seules les membrures à deux jointures ou membrures binaires seront utilisées.

*** Membrure d'entrée:**

C'est la membrure par laquelle le mouvement est communiqué au mécanisme. En général elle est actionnée par un moteur ou reliée à l'une des membrures mobiles d'un autre mécanisme en mouvement. On l'appelle également membrure entraînante ou membrure motrice.

* Membrure fixe:

Une membrure est fixe lorsqu'elle ne change pas de position au cours du mouvement du mécanisme. Elle sert de référence.

* Membrure intermédiaire:

Elle est reliée à la membrure motrice et transmet le mouvement à une autre membrure dite de sortie ou entraînée.

Chapitre 2: Le degré de liberté

2.1 Définition

Par degré de liberté, nous entendons le nombre de paramètres indépendants requis pour spécifier la position de chaque membrure par rapport à la membrure fixe.

2.2 Détermination du degré de liberté d'un ensemble de membrures

Lorsque les membrures sont coplanaires comme dans le cas de notre étude, la formule de KUTZBACH ou de GRUEBLER nous donne:

$$F=3*(N-1)-2*P1-P2$$

avec

F:nombre de degrés de liberté

N:nombre de membrures

P1:nombre de joints à un degré de liberté

P2:nombre de joints à deux degrés de liberté

Dans le cadre de la présente étude nous n'utiliserons que des joints à cheville: donc $P2=0$.

Dans la formule ci-dessus, un joint à cheville est supposé relié deux membrures. Or il arrive que plusieurs membrures soient reliées à un même joint à cheville. Une correction doit donc être portée à la formule ci-dessus pour tenir compte de ces cas.

Soit m_j le nombre de membrures reliées par le joint à cheville j . Ce joint supprimera $2*(m_j-1)$ degrés de liberté. Les P_1 joints à

cheville supprimeront donc $P_{it} = \sum(m_j - 1)$ degrés de liberté.

Avec ces considérations la formule devient:

$$F = 3 * (N - 1) - 2 * P_{it}$$

2.3 Exemples

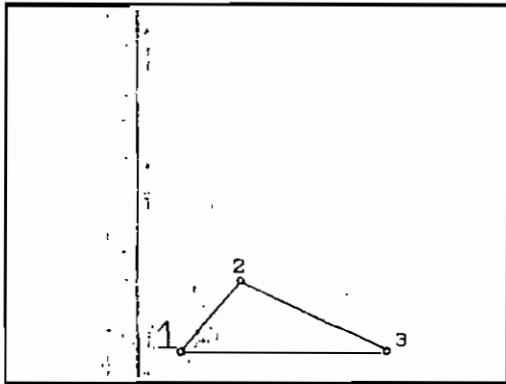


Figure 1 : Chaîne de membrures

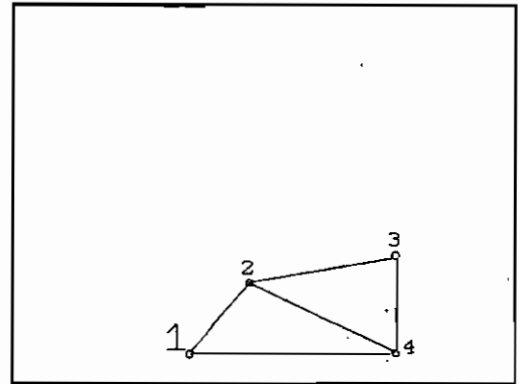


Figure 2: Chaîne de membrures

figure 1:

$$N = 3; m_1 = 2; m_2 = 2; m_3 = 2$$

$$\text{donc } P_{it} = (2-1) + (2-1) + (2-1) = 3$$

$$\text{d'où } F_1 = 3 * (3-1) - 2 * 3 = 0;$$

figure 2:

$$N = 5; m_1 = 2; m_2 = 3; m_3 = 2; m_4 = 3;$$

$$\text{donc } P_{it} = (2-1) + (3-1) + (2-1) + (3-1) = 6;$$

$$\text{d'où } F_1 = 3 * (5-1) - 2 * 6 = 0$$

2.4 Utilité du calcul du nombre de degré de liberté

Dans notre étude, une membrure est considérée comme rigide. Elle ne se déforme donc pas. Un ensemble de membrures reliées d'une façon quelconque ne produira pas nécessairement un mouvement. Pour obtenir un mouvement il faudrait que par rapport à une membrure fixe choisie, les autres puissent se mouvoir. Le nombre de degrés de liberté permet de prédire si les membrures peuvent se mettre en mouvement par rapport à la membrure fixe.

Lorsque le degré de liberté est inférieur ou égal à zéro le mouvement relatif est impossible. Dans le cas contraire les membrures peuvent mouvoir les unes par rapport aux autres: l'ensemble des membrures constitue alors un mécanisme.

Chapitre 3: Le mécanisme à quatre membrures

C'est un mécanisme de base très employé. De nombreux mécanismes couramment utilisés se résument à un mécanisme à quatre membrures ou à un ensemble de ceux-ci. Il peut-être parfois difficile à reconnaître, tel que le mécanisme manivelle-coulisseau utilisé dans les moteurs. Aussi, constitue-t-il l'objet de notre étude.

3.1 Définition

Le mécanisme à quatre membrures est une chaîne cinématique (c'est-à-dire un ensemble de corps rigides reliés entre eux) à liaison forcée (le mouvement relatif entre les membrures est unique et se répète d'un cycle à l'autre) dont les quatre membrures sont jointes par des chevilles et dont l'une des membrures est fixe.

3.2 Classification

La classification des mécanismes à quatre membrures peut se faire selon le type de mouvement des membrures d'entrée et de sortie.

Une membrure d'entrée ou de sortie constitue:

- une manivelle si elle peut faire un tour complet autour de son centre de rotation.
- un balancier si elle oscille une partie d'un tour.
- un coulisseau si elle effectue un mouvement de translation rectiligne guidé par une autre membrure.

3.2.1 Classification suivant la loi de GRASHOFF.

Cette loi s'applique pour identifier le type de transformation effectué par les mécanismes dont les quatre membrures sont de longueur donnée. Nous désignerons ces mécanismes sous le symbole de **M4M** dans notre étude.

Si l est la longueur de la membrure la plus longue,

s la longueur de la membrure la plus courte,

p et q les longueurs des autres membrures,

la loi de GRASHOFF stipule:

a- Si $l+s \leq p+q$ on aura:

* un manivelle-balancier si s est la longueur de la membrure adjacente à celle fixe;

* un manivelle-manivelle (ou double manivelle) si s est la longueur de la membrure fixe;

* un balancier-balancier (ou double balancier) si s est la longueur de la membrure opposée à celle fixe.

b- si $l+s > p+q$ on aura un balancier-balancier;

3.2.2. Autres types de mécanisme à quatre membrures

* manivelle-coulisseau

la membrure motrice fait un tour complet alors que la membrure de sortie (coulisseau) fait une translation

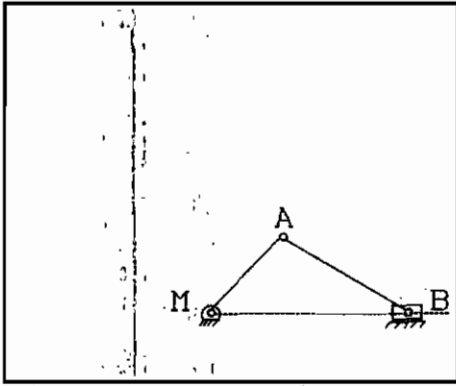


Figure 3: Mecanisme MC

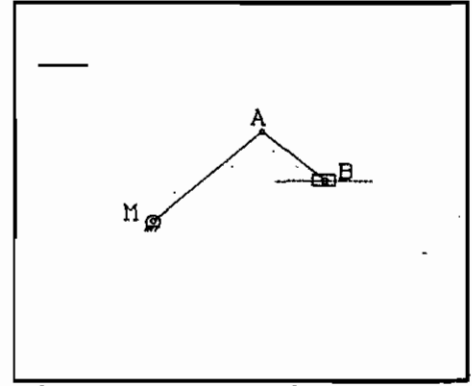


Figure 4: Mecanisme MC

***balancier-coulisseau**

la membrure d'entrée oscille autour de part et d'autre de deux positions angulaire sans faire un tour complet. Il ressemble au mécanisme manivelle-coulisseau à la seule différence que la membrure motrice est un balancier.

***manivelle-coulisseau inversé**

il s'obtient à partir d'un MC dans lequel la membrure qui était fixe devient mobile tandis-que l'une des membrures qui étaient mobilisées devient fixe. Nous le désignerons sous le symbole MCI.

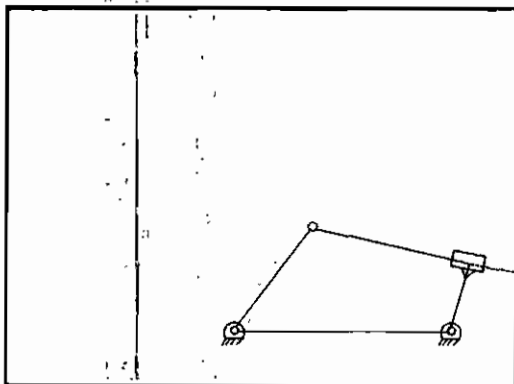


Figure 5: Mécanisme MCI 1

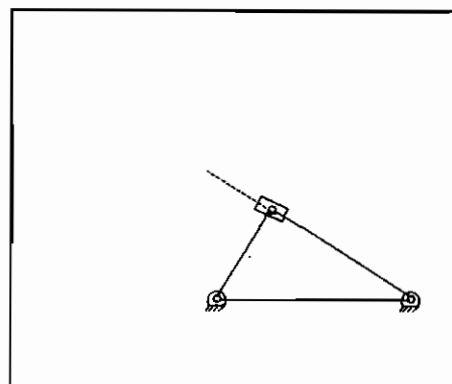


Figure 6 Mécanisme MCI

MECANISME MCI TYPE 1

MECANISME MCI (variante type2)

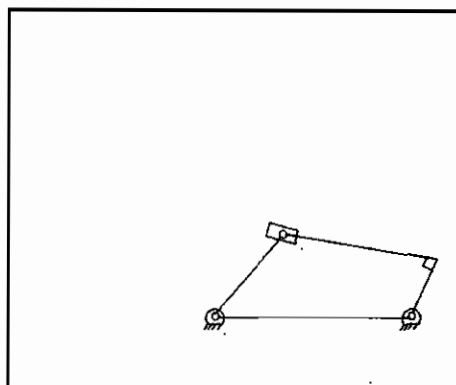


Figure 7: Mécanisme MCI 2

MECANISME MCI TYPE 2

Dans la suite, nous désignerons les mécanismes manivelle-coulisseau et balancier-coulisseau sous le symbole de MC.

*** coulisseau-coulisseau**

c'est un mécanisme à quatre membrures dont les membrures d'entrée et de sortie sont de longueur infinie. Un exemple en est le générateur d'ellipse. Ce mécanisme est peu utilisé dans la pratique. Aussi nous ne nous y appesantirons pas dans ce travail.

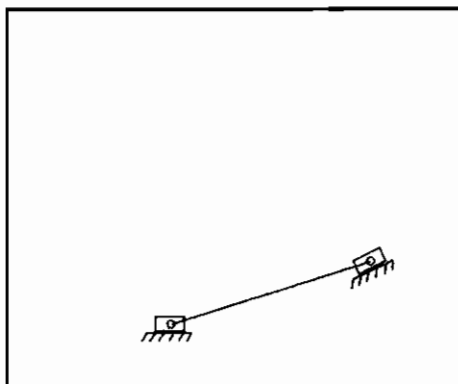


Figure 8

COULISSEAU-COULISSEAU

Chapitre 4: Positions limites et points morts.

Ce sont deux paramètres qui permettent de s'assurer de la mobilité effective du mécanisme à travers les différentes positions qu'il prend au cours du mouvement.

4.1 Positions limites et points morts pour un M4M.

4.1.1 Positions limites:

Cette position est caractérisée par la colinéarité entre la membrure motrice et la membrure intermédiaire.

Soient θ_2 l'orientation de la membrure d'entrée et θ_3 celle de la membrure intermédiaire ;

Le mécanisme sera dans une position limite si:

$$|\theta_2 - \theta_3| = 0 \quad \text{ou} \quad |\theta_2 - \theta_3| = 180 \text{ degrés}$$

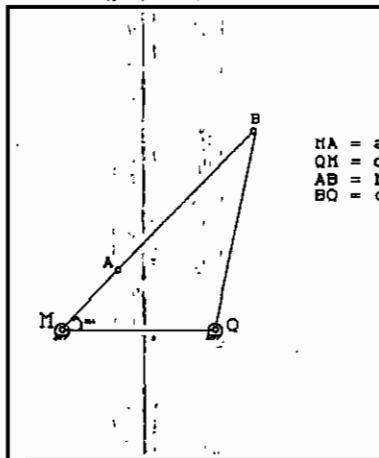


Figure 9

POSITION-LIMITE 1 DU M4M

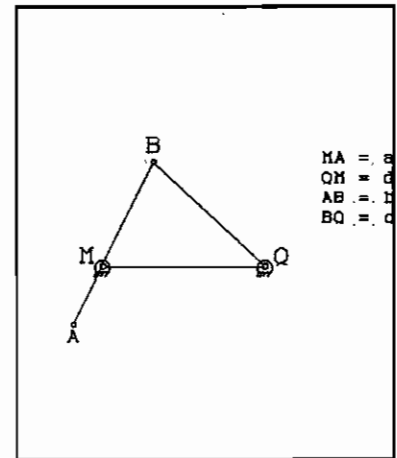


Figure 10

POSITION-LIMITE 2 DU M4M

De la figure 9 on déduit:

$$\cos(\theta_{2L1}) = \frac{d^2 + (a+b)^2 - c^2}{2 * (a+b) * d} \quad (1)$$

$$(1) \Rightarrow \theta_{2L1} = \left(\text{Arctg} \left(-1 + \frac{[2 * (a+b) * d]^2}{[(b+a)^2 + d^2 - c^2]^2} \right) \right)^{0.5}; \quad (2)$$

De la figure 10 on déduit:

$$\cos(\theta_{212}-180) = \frac{(b-a)^2+d^2-c^2}{2*(b-a)*d} \quad (3)$$

$$(3) \Rightarrow \theta_{212} = 180 + \left(\text{Arctan} \left(-1 + \frac{[2*(b-a)*d]^2}{[(b-a)^2+d^2-c^2]^2} \right) \right)^{0.5} \quad (4)$$

4.1.2. Point mort.

Dans cette position, la membrure intermédiaire et la membrure de sortie sont colinéaires.

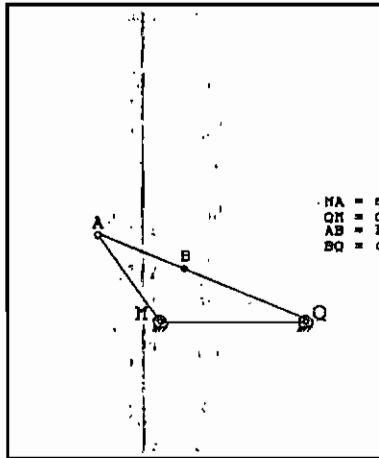


Figure 11

POINT-MORT 1 DU M4M

Sur la figure 11 on a:

$$\cos(\theta_{201}) = \frac{a^2 + d^2 - (b+c)^2}{2ad} \quad (5)$$

$$(5) \Rightarrow \theta_{201} = \left(\text{Arctan} \left(-1 + \frac{[2ad]^2}{[a^2 + d^2 - (b+c)^2]^2} \right) \right)^{0.5} \quad (6)$$

De la figure 12 on tire:

$$\cos(\theta_{202}) = \frac{a^2 + d^2 - (b-c)^2}{2*a*d} \quad (7)$$

$$(7) \Rightarrow \theta_{202} = \left(\text{Arctan} \left(-1 + \frac{[2ad]^2}{[a^2 + d^2 - (b-c)^2]^2} \right) \right)^{0.5} \quad (8)$$

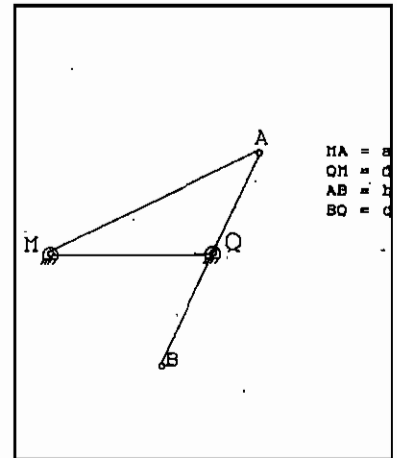


Figure 12

POINT-MORT 2 DU M4M

4.2 Positions limites et point mort d'un MC

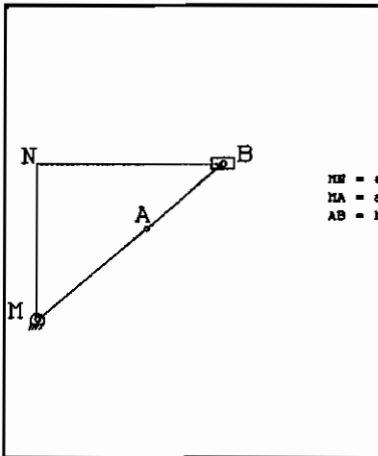


Figure 13

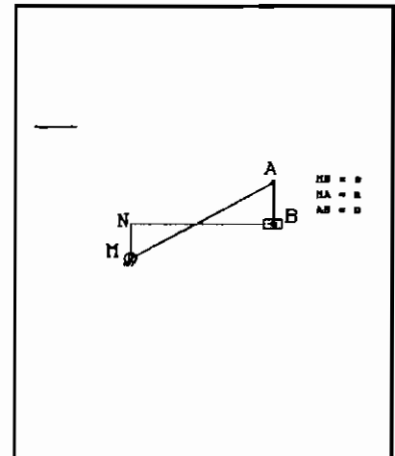


Figure 14

POSITION-LIMITE DU MC

POINT-MORT DU MC

4.2.1 Détermination de l'excentricité e

e sera déterminée graphiquement.

soit N l'intersection de l'axe du coulisseau et de la verticale passant par le noeud M; p l'angle que fait l'axe du coulisseau avec l'horizontale dans le sens trigonométrique (ici $p=0$);

les coordonnées de M sont x_M et y_M

les coordonnées de N sont x_N et y_N

les coordonnées de B sont x_B et y_B

On a:

$$x_N = x_M ; \quad (9)$$

$$y_N = x_N \cdot \text{tg}(p) + (y_B - x_B \cdot \text{tg}(p)) ; \quad (10)$$

du triangle MNM' on tire:

$$e=(y_N-y_M)*\sin(90-p); \quad (11)$$

e peut être négatif ou positif;

4.2.2 Position limite

Dans cette position la membrure motrice et la membrure intermédiaire sont colinéaires.

du triangle NMB' on tire:

$$\theta'=90+p-\phi; \quad (12)$$

figure 13 $\Rightarrow \sin(\phi_1)=\frac{e}{b+a} =x1$ (13)

$$(13)\Rightarrow\phi_1=(\text{Arctan}\left(\frac{x1^2}{1-x1^2}\right))^{0.5} \quad (14)$$

$$\theta_{L1}=\phi_1-p \quad (15)$$

figure 13 $\Rightarrow \sin(\phi_2)=\frac{e}{b-a} =x2$ (16)

$$(16)\Rightarrow\phi=(\text{Arctan}\left(\frac{x2^2}{1-x2^2}\right))^{0.5} \quad (17)$$

$$\theta_{L2}=\phi_2-p \quad (18)$$

4.2.3 Point mort

Ce mécanisme présente un point mort lorsque la membrure intermédiaire est perpendiculaire à l'axe du coulisseau, la manivelle étant motrice.

à partir de la figure 14 on déduit:

$$\sin(\theta') = \frac{b-e}{a} = x \quad (19)$$

$$(19) \Rightarrow \theta' = \left(\arctan\left(\frac{x^2}{1-x^2}\right) \right)^{0.5} \quad (20)$$

$$\theta = \theta' + p \quad (21)$$

4.3. Position limite de MCI

Elle correspond à celle où la manivelle est perpendiculaire à l'axe du coulisseau.

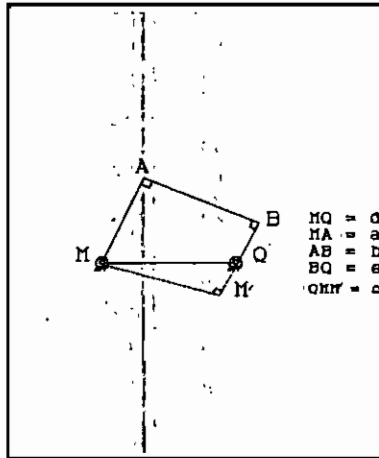


Figure 15

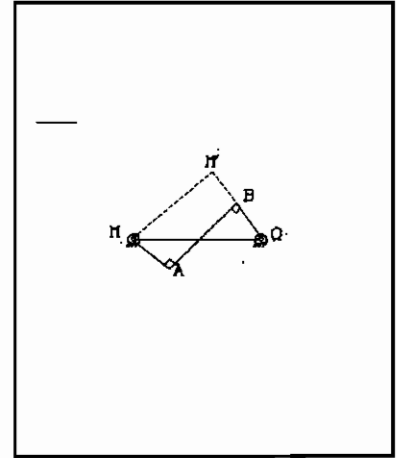


Figure 16

POSITION - LIMITE 1 DU MCI

de la figure 15 on déduit:

$$\sin(\theta') = \frac{e-a}{d} = x_1 \quad (22)$$

$$(22) \Rightarrow \theta' = (\text{Arctan}\left(\frac{x_1^2}{1-x_1^2}\right))^{0.5} \quad (23)$$

$$\theta_{2L1} = \theta' + 90 \quad (24)$$

de la figure 16 déduit:

$$\sin(\theta') = \frac{e+a}{d} = x_2 \quad (25)$$

$$\theta' = (\text{Arctan}\left(\frac{x_2^2}{1-x_2^2}\right))^{0.5} \quad (26)$$

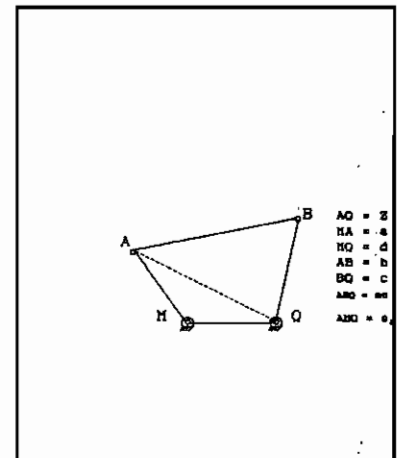
$$\theta_{2L2} = \theta' + 270 \quad (27)$$

Chapitre 5: Angle de transmission

Il s'agit de l'angle entre la direction du mouvement absolu T_a de la membrure de sortie et la direction du mouvement relatif T_r de la membrure intermédiaire par rapport à la membrure motrice. Nous le noterons μ . Pour une transmission efficace μ est en général compris entre 40° et 140° (SONI). Si μ est inférieur à 40° , le mécanisme tend à se déformer à cause du frottement au niveau des joints. Des accélérations excessives, des bruits et des saccades se créent dans ce cas à haute vitesse.

L'efficacité de la transmission est le rapport du couple de sortie au couple d'entrée. On la nomme aussi l'indice de mérite. Pour les mécanismes M4M nous établirons l'expression de cet indice en fonction de μ .

5.1 Angle de transmission pour un M4M



ANGLE DE TRANSMISSION DU M4M

Figure 17

La direction du mouvement relatif Tr est perpendiculaire à la membrure intermédiaire AB et la direction du mouvement absolu Ta de la membrure de sortie QB est perpendiculaire à QB :

De la figure 17 on tire:

angle $MAB = \beta$

$$\text{triangle MAQ} \Rightarrow a^2 + d^2 - 2*a*d*\cos(\theta_2) = z^2 \quad (28)$$

$$\text{triangle ABQ} \Rightarrow b^2 + c^2 - 2*b*c*\cos(\mu) = z^2 \quad (29)$$

$$\text{d'où } \cos(\mu) = \frac{-a^2 - d^2 + 2*a*d*\cos(\theta_2) + b^2 + c^2}{2*b*c} = x \quad (30)$$

$$(30) \Rightarrow \mu = (\text{Arctan}\left(\frac{1}{x^2} - 1\right))^{0.5} \quad (31)$$

Si T_2 est le couple d'entrée, T_4 le couple de sortie, on établit
 $i = T_4/T_2 = (c*\sin(\mu))/(a*\sin(\beta))$; i : indice de mérite;

Les valeurs extrêmes de μ s'obtiennent en annulant la dérivée par rapport à θ_2 de l'équation (31). On trouve alors que μ est minimum ou maximum lorsque $\theta_2 = 0$ ou $\theta_2 = 180^\circ$.

5.2 Angle de transmission pour MC

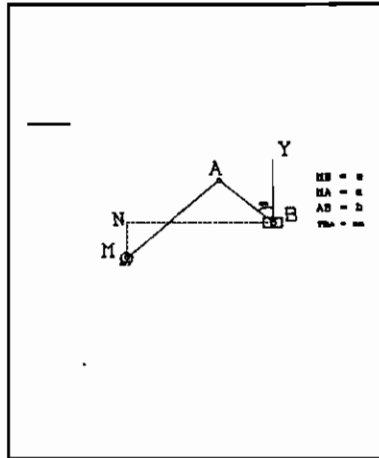


Figure 18

ANGLE DE TRANSMISSION DU MC

μ est l'angle entre la membrure intermédiaire AB et la normale y à l'axe du coulisseau.

Posons:

D :axe du coulisseau;

y:normale à D passant par B;

x:normale à D passant par A;

z:parallèle à D passant par M;

Y:point d'intersection des axes y et z;

X:point d'intersection des axes x et D;

Z:point d'intersection des axes x et z;

p:angle de D par rapport à l'horizontale (ici $p=0$);

De la figure 18 on observe:

μ est l'angle entre y et AB;

$e=ZA+AX$

(32)

triangle MZA => ZA=a*sin(p-θ₂) (33)

triangle AXB => AX=b*sin(90-μ)=b*cos(μ) (34)

(32), (33) et (34) => cos(μ) = $\frac{e-a*\sin(p-\theta_2)}{b}$ = x1 (35)

d'où μ = (Arctan($\frac{1}{x1^2} - 1$)^{0.5}) (36)

On peut alors remarquer que μ est maximum ou minimum lorsque θ₂=90 ou θ₂=270.

5:3 Angle de transmission pour MCI

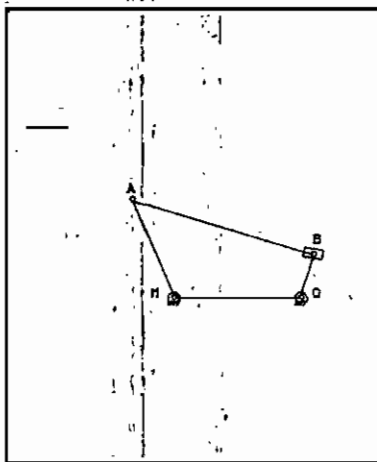


Figure 19

ANGLE DE TRANSMISSION
DU MCI TYPE 1

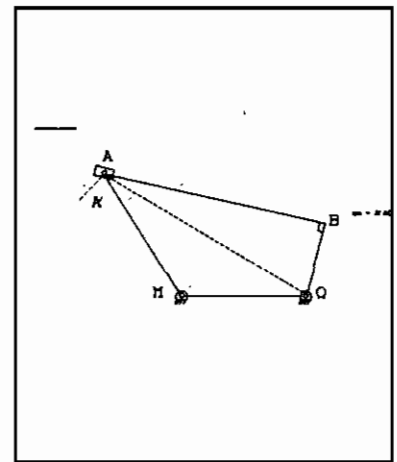


Figure 20

ANGLE DE TRANSMISSION
DU MCI TYPE 2

Dans le MCI de type 1 (figure 19), le coulisseau est relié à la membrure de sortie QB de manière à ce que l'angle de transmission soit maintenu à la valeur idéale de 90° .

Dans le MCI de type 2 (figure 20), le coulisseau est relié à la membrure d'entrée MA. L'angle de transmission est plus difficile à définir analytiquement en fonction de θ_2 . Graphiquement il correspond à l'angle entre le segment qui joint A et Q d'une part et la normale à l'axe du coulisseau et qui passe par A. Donc dans ce cas:

$$\mu = 90^\circ - \text{argument}(AB) + \text{argument}(AQ);$$

Après cette première étape consacrée à la définition des concepts généraux sur les mécanismes plans, nous aborderons dans la deuxième partie de ce rapport l'étude cinématique.

DEUXIEME PARTIE: ETUDE CINEMATIQUE DES MECANISMES PLANS

Le but de cette étude est de déterminer lorsqu'on connaît la position, la vitesse et l'accélération angulaire de la membrure d'entrée, les paramètres cinématiques (position, vitesse et accélération) des autres membrures et des différents joints. Il existe pour cela deux grandes approches: la méthode graphique et la méthode analytique.

La méthode graphique procède par le tracer des polygones de vitesse et d'accélération. Cette méthode est assez rapide. Cependant elle présente certains inconvénients. Les résultats qu'elle donne sont peu précis. Par ailleurs un polygone construit n'est valable que pour une position de la membrure d'entrée. La construction devra être reprise pour une autre position.

La méthode analytique est plus précise et permet de dériver en fonction des paramètres de la membrure d'entrée, les expressions des autres paramètres. Aussi choisirons nous cette méthode dans le cadre de notre programme. Les vecteurs seront notés sous la forme de nombre complexe.

Chapitre 1: Analyse cinématique des mécanismes M4M

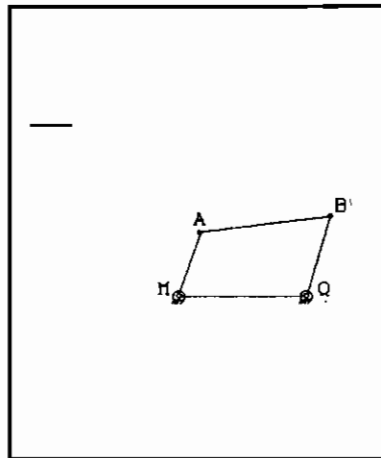


Figure 21

Mécanisme M4M

1.1 Position des mécanismes M4M

A partir de la figure 21 nous définissons les vecteurs suivants:

$$\text{membrure fixe:} \quad QM = d \cdot \exp(j \cdot \theta_1) \quad (37)$$

$$\text{membrure motrice :} \quad MA = a \cdot \exp(j \cdot \theta_2) \quad (38)$$

$$\text{membrure intermédiaire:} \quad AB = b \cdot \exp(j \cdot \theta_3) \quad (39)$$

$$\text{membrure de sortie:} \quad QB = c \cdot \exp(j \cdot \theta_4) \quad (40)$$

$$\text{polygone MABQ} \Rightarrow \text{QM} + \text{MA} + \text{AB} - \text{QB} = 0 \quad (41)$$

$$(41) \Rightarrow d \cdot \exp(j \cdot \theta_1) + a \cdot \exp(j \cdot \theta_2) + b \cdot \exp(j \cdot \theta_3) - c \cdot \exp(j \cdot \theta_4) = 0 \quad (42)$$

(42) \Rightarrow (43) et (44) avec:

$$d \cdot \cos(\theta_1) + a \cdot \cos(\theta_2) + b \cdot \cos(\theta_3) - c \cdot \cos(\theta_4) = 0 \quad (43)$$

$$d \cdot \sin(\theta_1) + a \cdot \sin(\theta_2) + b \cdot \sin(\theta_3) - c \cdot \sin(\theta_4) = 0 \quad (44)$$

en remarquant que $\theta_1 = 180^\circ$ on obtient:

$$(43) \Rightarrow -d + a \cdot \cos(\theta_2) + b \cdot \cos(\theta_3) - c \cdot \cos(\theta_4) = 0 \quad (45)$$

$$(44) \Rightarrow a \cdot \sin(\theta_2) + b \cdot \sin(\theta_3) - c \cdot \sin(\theta_4) = 0 \quad (46)$$

$$(45) \Rightarrow (-b \cdot \cos(\theta_3))^2 = (-d + a \cdot \cos(\theta_2) - c \cdot \cos(\theta_4))^2 \quad (47)$$

$$(46) \Rightarrow (-b \cdot \sin(\theta_3))^2 = (a \cdot \sin(\theta_2) - c \cdot \sin(\theta_4))^2 \quad (48)$$

en additionnant (47) et (48) et en posant:

$$K1 = \frac{d}{a} ; \quad K2 = \frac{d}{c} ; \quad K3 = \frac{a \cdot a - b \cdot b + c \cdot c + d \cdot d}{2 \cdot a \cdot c}$$

on obtient:

$$K1 \cdot \cos(\theta_4) - K2 \cdot \cos(\theta_2) + K3 = \cos(\theta_2 - \theta_4) \quad (49)$$

$$(49) \Rightarrow K1 \cdot \cos(\theta_4) - K2 \cdot \cos(\theta_2) + K3 = \cos(\theta_2) \cdot \cos(\theta_4) + \sin(\theta_4) \cdot \sin(\theta_2) \quad (50)$$

$$(50) \Rightarrow A \cdot (\tan(0.5 \cdot \theta_4))^2 + B \cdot \tan(0.5 \cdot \theta_4) + C = 0 \quad (51)$$

avec:

$$A = \cos(\theta_2) + K3 - K1 - K2 \cdot \cos(\theta_2) ;$$

$$B = -2 * \sin(\theta_2);$$

$$C = K_1 + K_3 - (1 + K_2) * \cos(\theta_2);$$

$$(51) \Rightarrow \theta_4 = 2 * \text{Arctan}\left(\frac{-B + (B^2 - 4 * A * C)^{0.5}}{2 * A}\right) \quad (52)$$

ou

$$\theta_4 = 2 * \text{Arctan}\left(\frac{-B - (B^2 - 4 * A * C)^{0.5}}{2 * A}\right) \quad (53)$$

De même:

$$(45) \Rightarrow (-d + a * \cos(\theta_2) + b * \cos(\theta_3))^2 = (c * \cos(\theta_4))^2 \quad (54)$$

$$(46) \Rightarrow (a * \sin(\theta_2) + b * \sin(\theta_3))^2 = (c * \sin(\theta_4))^2 \quad (55)$$

en additionnant (54) et (55) et en posant :

$$K_4 = d/b; \quad K_5 = \frac{c^2 - d^2 - a^2 - b^2}{2 * a * b} \quad \text{on obtient:}$$

$$K_4 * \cos(\theta_2) + K_1 * \cos(\theta_3) + K_5 = \cos(\theta_2 - \theta_3) \quad (56)$$

en posant:

$$D = K_4 * \cos(\theta_2) + \cos(\theta_2) + K_5 - K_1$$

$$E = -2 * \sin(\theta_2)$$

$$F = K_4 * \cos(\theta_2) - \cos(\theta_2) + K_5 + K_1$$

(56) donne:

$$\theta_3 = 2 * \text{Arctan}\left(\frac{-E + (E^2 - 4 * D * F)^{0.5}}{2 * D}\right) \quad (57)$$

ou

$$\theta_3 = 2 * \text{Arctan} \left(\frac{-E - (E^2 - 4 * D * F)^{0.5}}{2 * D} \right) \quad (58)$$

Conclusion

Les équations (57) et (58) d'une part, et (52) et (53) d'autre part permettent de déterminer les positions angulaires θ_3 et θ_4 des membrures 3 et 4.

1.2 Vitesse dans les mécanismes M4M

Ici nous établirons les expressions des vitesses angulaires des membrures ainsi que celles des noeuds.

$$(42) \Rightarrow d * \exp(j * \theta_1) + a * \exp(j * \theta_2) + b * \exp(j * \theta_3) = c * \exp(j * \theta_4) \quad (59)$$

en dérivant (59) par rapport au temps et en tenant compte que θ_1 est une constante valant 180° , on obtient:

$$j * a * w_2 * \exp(j * \theta_2) + j * b * w_3 * \exp(j * \theta_3) = j * c * w_4 * \exp(j * \theta_4) \quad (60)$$

$$\text{avec } w_2 = \frac{d\theta_2}{dt} ; \quad w_3 = \frac{d\theta_3}{dt} ; \quad w_4 = \frac{d\theta_4}{dt} ;$$

(60) \Rightarrow (61) et (62) avec:

$$c * w_4 * \sin(\theta_4) = a * w_2 * \sin(\theta_2) + b * w_3 * \sin(\theta_3) \quad (61)$$

$$c * w_4 * \cos(\theta_4) = a * w_2 * \cos(\theta_2) + b * w_3 * \cos(\theta_3) \quad (62)$$

d'où

$$w_4 = \frac{a * w_2 * \sin(\theta_2 - \theta_3)}{c * \sin(\theta_4 - \theta_3)} \quad (63)$$

$$w_3 = \frac{a * w_2 * \sin(\theta_4 - \theta_2)}{b * \sin(\theta_3 - \theta_4)} \quad (64)$$

De l'équation (60) on déduit les vitesses des joints A et B

$$V_A = a \cdot \omega_2 \cdot j \cdot \exp(j \cdot \theta_2) \quad (65)$$

$$V_B = c \cdot \omega_4 \cdot j \cdot \exp(j \cdot \theta_4) \quad (66)$$

$$V_{B/A} = b \cdot \omega_3 \cdot j \cdot \exp(j \cdot \theta_3) \quad (67)$$

1.3 Accélération dans les mécanismes M4M

Etablissons les équations qui nous permettront de calculer les différentes accélérations.

En dérivant (60) par rapport au temps on obtient:

$$j \cdot a \cdot \alpha_2 \cdot \exp(j \cdot \theta_2) + j^2 \cdot a \cdot \omega_2^2 \cdot \exp(j \cdot \theta_2) + j^2 \cdot b \cdot \omega_3^2 \cdot \exp(j \cdot \theta_3) + j \cdot b \cdot \alpha_3 \cdot \exp(j \cdot \theta_3) = j^2 \cdot c \cdot \omega_4^2 \cdot \exp(j \cdot \theta_4) + j \cdot c \cdot \alpha_4 \cdot \exp(j \cdot \theta_4) \quad (68)$$

avec:

$$\alpha_2 = d\omega_2/dt; \quad \alpha_3 = d\omega_3/dt; \quad \alpha_4 = d\omega_4/dt;$$

(68) => (69) et (70) avec

$$c \cdot \alpha_4 \cdot \sin(\theta_4) - b \cdot \alpha_3 \cdot \sin(\theta_3) = a \cdot \alpha_2 \cdot \sin(\theta_2) + a \cdot \omega_2^2 \cdot \cos(\theta_2) + b \cdot \omega_3^2 \cdot \cos(\theta_3) - c \cdot \omega_4^2 \cdot \cos(\theta_4) \quad (69)$$

$$c \cdot \alpha_4 \cdot \cos(\theta_4) - b \cdot \alpha_3 \cdot \cos(\theta_3) = a \cdot \alpha_2 \cdot \cos(\theta_2) - a \cdot \omega_2^2 \cdot \sin(\theta_2) - b \cdot \omega_3^2 \cdot \sin(\theta_3) + c \cdot \omega_4^2 \cdot \sin(\theta_4) \quad (70)$$

en posant:

$$A = c \cdot \sin(\theta_4); \quad B = b \cdot \sin(\theta_3);$$

$$C = a \cdot \alpha_2 \cdot \sin(\theta_2) + a \cdot \omega_2^2 \cdot \cos(\theta_2) + b \cdot \omega_3^2 \cdot \cos(\theta_3) - c \cdot \omega_4^2 \cdot \cos(\theta_4);$$

$$D = c \cdot \cos(\theta_4); \quad E = b \cdot \cos(\theta_3);$$

$$F = a \cdot \alpha_2 \cdot \cos(\theta_2) - a \cdot \omega_2^2 \cdot \sin(\theta_2) - b \cdot \omega_3^2 \cdot \sin(\theta_3) + c \cdot \omega_4^2 \cdot \sin(\theta_4);$$

on obtient:

$$\alpha_3 = \frac{C \cdot D - A \cdot F}{A \cdot E - B \cdot D} \quad (71)$$

$$\alpha_4 = \frac{C \cdot E - B \cdot F}{A \cdot E - B \cdot D} \quad (72)$$

de l'équation (68) on déduit les accélérations radiale A^r et tangentielle

A^t pour les joints A et B.

$$A^r(A) = j^2 \cdot a \cdot w^2 \cdot \exp(j \cdot \theta_2) \quad (73)$$

$$A^t(A) = j \cdot a \cdot \alpha_2 \cdot \exp(j \cdot \theta_2) \quad (74)$$

$$A^r(B) = j^2 \cdot c \cdot w^4 \cdot \exp(j \cdot \theta_2) \quad (75)$$

$$A^t(B) = j \cdot c \cdot \alpha_4 \cdot \exp(j \cdot \theta_4) \quad (76)$$

$$A^r(B/A) = j^2 \cdot b \cdot w^3 \cdot \exp(j \cdot \theta_3) \quad (77)$$

$$A^t(B/A) = j \cdot b \cdot \alpha_3 \cdot \exp(j \cdot \theta_3) \quad (78)$$

Chapitre 2: Analyse cinématique des mécanismes MC

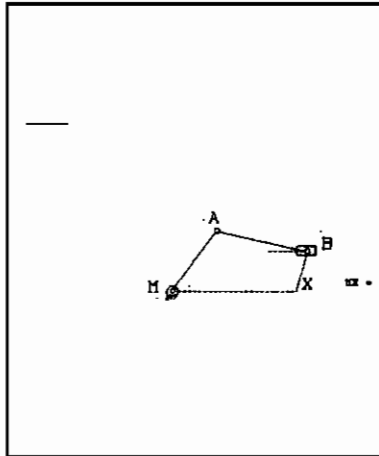


Figure 22

Mécanisme MC

Pour faire cette analyse, nous définissons comme sur la figure 22. les vecteurs suivants:

$$\begin{aligned} XM &= s \cdot \exp(j \cdot \theta_1); \quad MA = a \cdot \exp(j \cdot \theta_2); \quad BA = b \cdot \exp(j \cdot \theta_3); \quad XB = e \cdot \exp(j \cdot \theta_4); \\ \text{avec } \theta_1 &= 180; \quad \theta_4 = 90; \end{aligned}$$

2.1 Position dans les MC

Du polygone MABX on tire:

$$XM + MA - BA - XB = 0 \quad (79)$$

$$(79) \Rightarrow s \cdot \exp(j \cdot \theta_1) + a \cdot \exp(j \cdot \theta_2) - b \cdot \exp(j \cdot \theta_3) - e \cdot \exp(j \cdot \theta_4) = 0 \quad (80)$$

(80) \Rightarrow (81) et (82) avec:

$$a \cdot \cos(\theta_2) + s \cdot \cos(\theta_1) - e \cdot \cos(\theta_4) - b \cdot \cos(\theta_3) = 0 \quad (81)$$

$$a \cdot \sin(\theta_2) + s \cdot \sin(\theta_1) - e \cdot \sin(\theta_4) - b \cdot \sin(\theta_3) = 0 \quad (82)$$

$$(81) \Rightarrow a \cdot \cos(\theta_2) - s - b \cdot \cos(\theta_3) = 0 \quad (83)$$

$$(82) \Rightarrow a \cdot \sin(\theta_2) - e - b \cdot \sin(\theta_3) = 0 \quad (84)$$

$$(83) \Rightarrow (a \cdot \cos(\theta_2) - s)^2 = (b \cdot \cos(\theta_3))^2 \quad (85)$$

$$(84) \Rightarrow (a \cdot \sin(\theta_2) - e)^2 = (b \cdot \sin(\theta_3))^2 \quad (86)$$

en additionnant (46) et (47) on obtient:

$$L \cdot s^2 + M \cdot s + N = 0 \quad (87)$$

avec:

$$L = 1; M = -2 \cdot a \cdot \cos(\theta_2); N = a^2 + e^2 - b^2 - 2 \cdot a \cdot e \cdot \sin(\theta_2)$$

d'où

$$s_1 = \frac{-M + (M^2 - 4 \cdot L \cdot N)^{0.5}}{2 \cdot L} \quad (88)$$

$$s_2 = \frac{-M - (M^2 - 4 \cdot L \cdot N)^{0.5}}{2 \cdot L} \quad (89)$$

Les équations (49) et (50) permettent de connaître le déplacement du coulisseau en B.

2.2 Vitesse dans les mécanismes MC

Elaborons les expressions des différentes vitesses.

En dérivant (80) par rapport au temps on obtient:

$$j \cdot a \cdot w_2 \cdot \exp(j \cdot \theta_2) + s' \cdot \exp(j \cdot \theta_1) = j \cdot b \cdot w_3 \cdot \exp(j \cdot \theta_3) \quad (90)$$

(90) => (91) et (92) avec:

$$a \cdot w_2 \cdot \sin(\theta_2) - s' \cdot \cos(\theta_1) = b \cdot w_3 \cdot \sin(\theta_3) \quad (91)$$

$$a \cdot w_2 \cdot \cos(\theta_2) + s' \cdot \sin(\theta_1) = b \cdot w_3 \cdot \cos(\theta_3) \quad (92)$$

$$w_3 = \frac{a \cdot w_2 \cdot \cos(\theta_2)}{b \cdot \cos(\theta_3)} \quad (93)$$

$$(84) \Rightarrow \sin(\theta_3) = (e - a \cdot \sin(\theta_2)) / (-b) = x \quad (94)$$

$$(94) \Rightarrow \theta_3 = \left(\text{Arctan} \left(\frac{x^2}{1-x^2} \right) \right)^{0.5} \quad (95)$$

$$(91) \Rightarrow s' = a \cdot w_2 \cdot \frac{\sin(\theta_3 - \theta_2)}{\cos(\theta_3)} \quad (96)$$

Les équations (93) et (96) permettent de déterminer w_3 et s' .

2.3 Accélération dans les mécanismes MC

Pour déterminer α_3 et s'' nous allons dériver l'équation (90) par rapport au temps. Nous obtenons alors:

$$j \cdot a \cdot \alpha_2 \cdot \exp(j \cdot \theta_2) + j \cdot j \cdot a \cdot w_2^2 \cdot \exp(j \cdot \theta_2) + s'' \cdot \exp(j \cdot \theta_1) = j \cdot b \cdot \alpha_3 \cdot \exp(j \cdot \theta_3) + j^2 \cdot b \cdot w_3^2 \cdot \exp(j \cdot \theta_3) \quad (97)$$

(97) \Rightarrow (98) et (99) avec:

$$a \cdot \alpha_2 \cdot \sin(\theta_2) + a \cdot w_2^2 \cdot \cos(\theta_2) - s'' \cdot \cos(\theta_1) = b \cdot \alpha_3 \cdot \sin(\theta_3) + b \cdot w_3^2 \cdot \cos(\theta_3) \quad (98)$$

$$a \cdot \alpha_2 \cdot \cos(\theta_2) - a \cdot w_2^2 \cdot \sin(\theta_2) = b \cdot \alpha_3 \cdot \cos(\theta_3) - b \cdot w_3^2 \cdot \sin(\theta_3) \quad (99)$$

$$(99) \Rightarrow \alpha_3 = \frac{a \cdot \alpha_2 \cdot \cos(\theta_2) - a \cdot w_2^2 \cdot \sin(\theta_2) + b \cdot w_3^2 \cdot \sin(\theta_3)}{b \cdot \cos(\theta_3)} \quad (100)$$

(98) \Rightarrow (101) avec:

$$s'' = \frac{a \cdot \alpha_2 \cdot \sin(\theta_2) + a \cdot w_2^2 \cdot \cos(\theta_2) - b \cdot \alpha_3 \cdot \sin(\theta_3) - b \cdot w_3^2 \cdot \cos(\theta_3)}{\cos(\theta_1)} \quad (101)$$

α_3 est le module de l'accélération angulaire de la membrure
intermédiaire

s'' est le module de l'accélération linéaire du coulisseau.

Chapitre 3: Analyse cinématique des mécanismes MCI

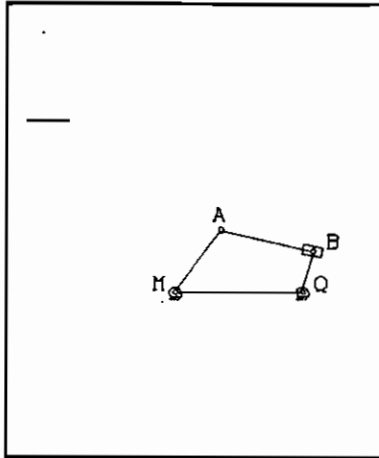


Figure 23

Mécanisme MCI

Pour faire cette étude, nous définissons les vecteurs suivants:

$$QM = d \cdot \exp(j \cdot \theta_1); \quad MA = a \cdot \exp(j \cdot \theta_2); \quad BA = b \cdot \exp(j \cdot \theta_3); \quad QB = e \cdot \exp(j \cdot \theta_4);$$

avec: $\theta_1 = 180^\circ$; $\theta_3 = 90 + \theta_4$; (voir figure 23)

3.1 Position dans les mécanismes MCI

Du polygone MABQ on déduit:

$$QM + MA - BA - QB = 0 \quad (102)$$

$$(102) \Rightarrow d \cdot \exp(j \cdot \theta_1) + a \cdot \exp(j \cdot \theta_2) - b \cdot \exp(j \cdot \theta_3) - e \cdot \exp(j \cdot \theta_4) = 0 \quad (103)$$

(103) \Rightarrow (104) et (105) avec:

$$d \cdot \cos(\theta_1) + a \cdot \cos(\theta_2) - b \cdot \cos(\theta_3) - e \cdot \cos(\theta_4) = 0 \quad (104)$$

$$d \cdot \sin(\theta_1) + a \cdot \sin(\theta_2) - b \cdot \sin(\theta_3) - e \cdot \sin(\theta_4) = 0 \quad (105)$$

$$(104) \Rightarrow a \cdot \cos(\theta_2) - d - e \cdot \cos(\theta_4) + b \cdot \sin(\theta_4) = 0 \quad (106)$$

$$(105) \Rightarrow a \cdot \sin(\theta_2) - e \cdot \sin(\theta_4) - b \cdot \cos(\theta_4) = 0 \quad (107)$$

$$(107) \Rightarrow b = \frac{a \cdot \sin(\theta_2) - e \cdot \sin(\theta_4)}{\cos(\theta_4)} \quad (108)$$

$$(106) \Rightarrow (a \cdot \cos(\theta_2) - d) \cdot \cos(\theta_4) + a \cdot \sin(\theta_2) \cdot \sin(\theta_4) - e = 0 \quad (109)$$

$$(109) \Rightarrow K_1 \cdot (\tan(\theta_4/2))^2 + K_2 \cdot \tan(\theta_4/2) + K_3 = 0 \quad (110)$$

avec:

$$K_1 = d - e - a \cdot \cos(\theta_2); \quad K_2 = 2 \cdot a \cdot \sin(\theta_2); \quad K_3 = a \cdot \cos(\theta_2) - d - e$$

$$(110) \Rightarrow \theta_4 = 2 \cdot \text{Arctan}\left(\frac{-K_2 - (K_2^2 - 4 \cdot K_1 \cdot K_3)^{0.5}}{2 \cdot K_1}\right) \quad (111)$$

ou

$$\theta_4 = 2 \cdot \text{Arctan}\left(\frac{-K_2 + (K_2^2 - 4 \cdot K_1 \cdot K_3)^{0.5}}{2 \cdot K_1}\right) \quad (112)$$

3.2 Vitesse dans les mécanismes MCI

La vitesse angulaire w_4 et la vitesse linéaire relative b' du coulisseau en B peuvent être obtenues en dérivant l'équation (103) par rapport au temps. On obtient alors:

$$j \cdot a \cdot w_2 \cdot \exp(j \cdot \theta_2) = b' \cdot \exp(j \cdot \theta_3) + j \cdot b \cdot w_3 \cdot \exp(j \cdot \theta_3) + j \cdot e \cdot w_4 \cdot \exp(j \cdot \theta_4) \quad (113)$$

(113) \Rightarrow (114) et (115) avec:

$$a \cdot w_2 \cdot \sin(\theta_2) = b' \cdot \sin(\theta_4) + w_4 \cdot (e \cdot \sin(\theta_4) + b \cdot \cos(\theta_4)) \quad (114)$$

$$a \cdot w_2 \cdot \cos(\theta_2) = b' \cdot \cos(\theta_4) + w_4 \cdot (e \cdot \cos(\theta_4) - b \cdot \sin(\theta_4)) \quad (115)$$

des équations (114) et (115) on déduit:

$$w_4 = \frac{a \cdot w_2}{b} \cdot \sin(\theta_2 - \theta_4) \quad (116)$$

$$b' = \frac{a \cdot \omega^2}{b} * (b \cdot \cos(\theta_4 - \theta_2) - e \cdot \sin(\theta_4 - \theta_2)) \quad (117)$$

3.3 Accélération dans les mécanismes MCI

L'accélération angulaire α_4 et l'accélération linéaire relative b'' peuvent être déterminées lorsqu'on dérive l'équation (113) par rapport au temps. On obtient alors:

$$\begin{aligned} j \cdot a \cdot \alpha_2 \cdot \exp(j \cdot \theta_2) + j^2 \cdot a \cdot \omega^2 \cdot \exp(j \cdot \theta_2) = \\ b'' \cdot \exp(j \cdot \theta_3) + \\ j \cdot b' \cdot \omega_3 \cdot \exp(j \cdot \theta_3) + j \cdot b' \cdot \omega_3 \cdot \exp(j \cdot \theta_3) + \\ j \cdot b' \cdot \alpha_3 \cdot \exp(j \cdot \theta_3) + j^2 \cdot b \cdot \omega_3^2 \cdot \exp(j \cdot \theta_3) + \\ j^2 \cdot e \cdot \alpha_4 \cdot \exp(j \cdot \theta_4) + j \cdot e \cdot \omega_4^2 \cdot \exp(j \cdot \theta_4) \end{aligned} \quad (118)$$

avec:

$$\theta_3 = \theta_4 + 90,$$

$$\omega_3 = \omega_4 \quad (119)$$

$$\alpha_3 = \alpha_4 \quad (120)$$

l'équation (118) => (121) et (122) avec:

$$\begin{aligned} b'' \cdot \sin(\theta_4) + (b \cdot \cos(\theta_4) + e \cdot \sin(\theta_4)) \cdot \alpha_4 = \\ a \cdot \omega^2 \cdot \cos(\theta_2) + a \cdot \alpha_2 \cdot \sin(\theta_2) - \\ 2 \cdot b' \cdot \omega_4 \cdot \cos(\theta_4) + b \cdot \omega_4^2 \cdot \sin(\theta_4) - \\ e \cdot \omega_4^2 \cdot \cos(\theta_4) \end{aligned} \quad (121)$$

$$\begin{aligned}
b'' \cdot \cos(\theta_4) + (-b \cdot \sin(\theta_4) + e \cdot \cos(\theta_4)) \cdot \alpha_4 = \\
-a \cdot w_2^2 \cdot \sin(\theta_2) + a \cdot \alpha_2 \cdot \cos(\theta_2) + \\
2 \cdot b' \cdot w_4 \cdot \sin(\theta_4) + b \cdot w_4^2 \cdot \cos(\theta_4) + \\
e \cdot w_4^2 \cdot \sin(\theta_4)
\end{aligned} \tag{122}$$

en posant:

$$A = \sin(\theta_4);$$

$$B = b \cdot \cos(\theta_4) + e \cdot \sin(\theta_4);$$

$$D = \cos(\theta_4);$$

$$E = -b \cdot \sin(\theta_4) + e \cdot \cos(\theta_4);$$

$$\begin{aligned}
C = a \cdot w_2^2 \cdot \cos(\theta_2) + a \cdot \alpha_2 \cdot \sin(\theta_2) - 2 \cdot b' \cdot w_4 \cdot \cos(\theta_4) + \\
b \cdot w_4^2 \cdot \sin(\theta_4) - e \cdot w_4^2 \cdot \cos(\theta_4)
\end{aligned}$$

$$\begin{aligned}
F = -a \cdot w_2^2 \cdot \sin(\theta_2) + a \cdot \alpha_2 \cdot \cos(\theta_2) + 2 \cdot b' \cdot w_4 \cdot \sin(\theta_4) + \\
b \cdot w_4^2 \cdot \cos(\theta_4) + e \cdot w_4^2 \cdot \sin(\theta_4)
\end{aligned}$$

on obtient:

$$\alpha_4 = \frac{A \cdot F - C \cdot D}{A \cdot E - B \cdot D} \tag{123}$$

$$b'' = \frac{C \cdot E - B \cdot F}{A \cdot E - B \cdot D} \tag{124}$$

Conclusion: Les deux premières parties nous ont permis d'élaborer les outils mathématiques qui seront le support du progiciel. Ce dernier peut être à présent étudié. C'est l'objectif de la dernière partie de ce rapport.

TROISIEME PARTIE: Présentation du progiciel SAMP1

Chapitre 1: Exposé du problème.

1-Introduction

SAMP1 est la première version du progiciel de SIMULATION et d'ANALYSE des MECANISMES PLANS écrite dans le cadre de notre projet de fin d'étude.

2- Définition du problème

Ce rapport décrit la conception, la réalisation et le mode d'emploi du programme SAMP1. Ce dernier permet d'effectuer les opérations ci-dessous:

- dessiner un mécanisme plan à 4 membrures;
- déterminer le degré de liberté;
- identifier le type de transformation de mouvement possible;
- simuler le mécanisme en mouvement;
- Tracer la courbe de bielle manivelle
- Déterminer les conditions de positions limites et de points morts
- Déterminer l'angle de transmission
- Déterminer les variables cinématiques: position, vitesse et accélération
- Déterminer l'indice de mérite

-Reproduire le mécanisme dans une position donnée

3- Analyse du problème

Le problème essentiel à résoudre est celui de la représentation du mécanisme car tous les objectifs du projet partent du schéma du mécanisme.

* Le dessin du mécanisme.

Comme nous le soulignons plus haut, un mécanisme à quatre membrures est parfois difficile à reconnaître.

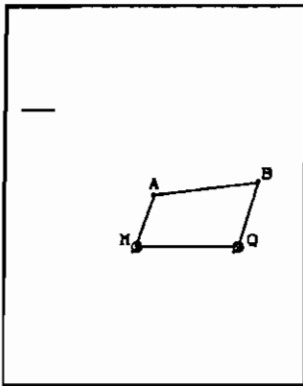


Figure 24

Mécanisme à 4 membrures

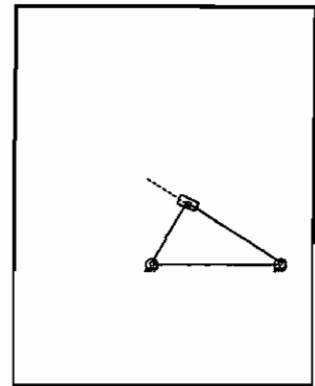


Figure 25

Mécanisme à 4 membrures

Les figures 24,25
et 26 représentent
chacune un mécanisme
à quatre membrures

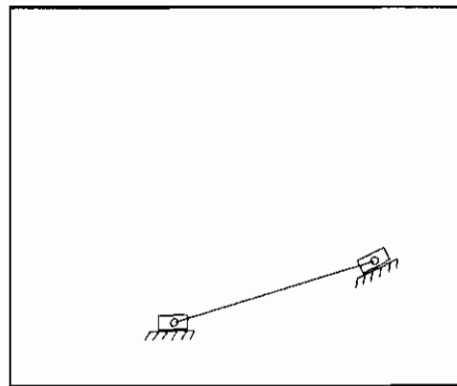


Figure 26

Mécanisme à 4 membrures

même si à vue d'oeil, une grande ressemblance ne se dégage entre eux. Or pour les besoins de la similitude il est nécessaire de représenter le mécanisme. Cette représentation doit être faite de façon à ce que le schéma résultant possède des caractéristiques identifiables par la suite.

Un moyen d'y parvenir serait de décomposer la figure à représenter en des segments de droite et de petits cercles.

Ainsi les figures ci-dessous se représenteraient en 9 segments et 2 cercles pour la figure 27 et 7 segments et 3 cercles pour la figure 28.

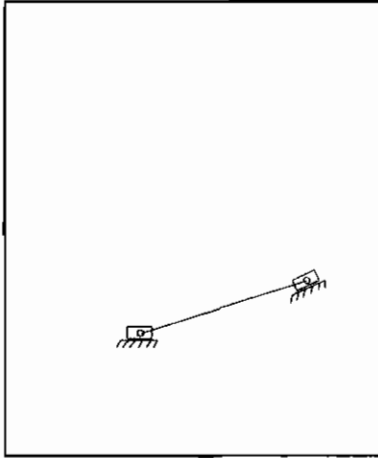


Figure 27

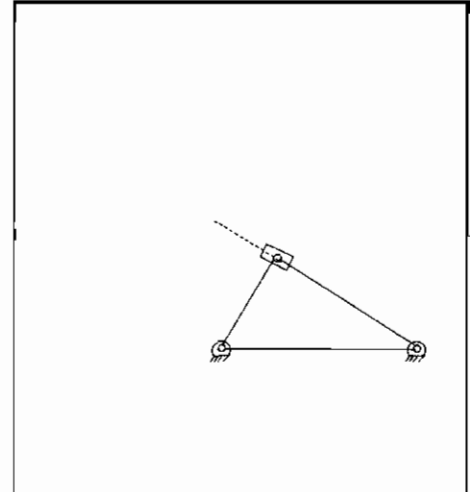


Figure 28

Mécanisme à 4 membrures

Mécanisme à 4 membrures

Un tel schéma serait très lent à réaliser à l'écran. Pire, ses caractéristiques seraient difficiles à définir. Sa précision serait faible.

Pour éviter ces inconvénients, nous assimilerons les coulisseaux à des "joints P" d'orientation (inclinaison de l'axe du coulisseau en degré) à définir. Les autres membrures sont considérées comme des segments de droite orientés (vecteur). Les chevilles sont des cercles de rayon fixe (joint R) :

Ainsi la figure 27 sera représentée par:

- un segment AB de longueur et de pente données
- un joint P d'orientation 0° en A
- un joint P d'orientation 45° (par exemple) en B

Alors que la figure 28 sera représentée par:

- un segment AB de longueur et pente données
- un segment BC de longueur et pente données
- un segment CA de longueur et pente données .
- un joint R en A
- un joint P d'orientation 130 degrés (par exemple) en B
- un joint R en C

Il s'agira donc, pour la représentation d'un mécanisme, de dessiner les membrures qui ne sont pas des coulisseaux à l'aide d'un segment de droite, de préciser ensuite les spécifications à chaque noeud: cheville => joint R, coulisseau => joint P.

La procédure 'SCHEMA' permet de représenter à l'écran le mécanisme souhaité. Il est à noter que dans cette représentation, les membrures binaires sont considérées comme des vecteurs. Ainsi la membrure AB diffère de la membrure BA alors que dans la réalité ces deux membrures sont identiques.

Il serait fastidieux de refaire le même dessin chaque fois qu'on aurait besoin d'analyser le mécanisme qu'il représente. C'est pourquoi le dessin réalisé devra être sauvegardé sur une disquette de données. Il pourra alors être rappelé au besoin.

* Le degré de liberté.

Le calcul du degré de liberté se fera à l'aide de l'équation (0). Il est à noter que ce calcul ne se fera que pour les dessins qui ne présentent pas de coulisseau.

* Type de transformation de mouvement.

Cette étude se fera en utilisant la loi de Grashoff pour les M4M.

* Simulation du mouvement.

Il s'agira de montrer le mécanisme dans les différentes positions qu'il occupe au cours d'un cycle. Suivant la valeur de la vitesse de la membrure motrice, le mouvement sera lent ou rapide. Nous ne pourrons, comme dans la réalité, montrer le mécanisme aux instants t , $t+dt$, $t+2dt$ etc., dt étant infiniment petit (le cycle durerait une éternité!). Etant donné une vitesse w_2 de la membrure motrice, il s'agira pour nous de montrer le mécanisme un certain nombre de fois au cours de son cycle. Par exemple, pour un mécanisme manivelle-manivelle nous pourrons procéder comme dans le tableau ci-dessous:

w_2	Nombre d'images du mécanisme au cours de cycle
1	360
2	180
3	120
4	90
.	.
10	36
.	.

.	.
180	2
360	1

La simulation pourra être interrompue par l'utilisateur pour visualiser une position du mécanisme au cours de son mouvement. Le mouvement sera poursuivi au gré de l'utilisateur. Par ailleurs les paramètres des différentes positions montrées seront imprimés de sorte que l'utilisateur pourra demander une visualisation du mécanisme dans une position qu'il choisirait. Ceci permettra de faire l'impression du dessin dans des positions désirées.

* Le tracé de la courbe de bielle manivelle.

Comme nous le rappelions ci-dessus, un mécanisme plan aussi complexe soit il, peut être décomposé en un ensemble de mécanismes à quatre membrures. Pour que le mouvement résultant de cette combinaison corresponde au besoin de l'utilisateur il est nécessaire de relier les mécanismes de base entre eux de manière bien déterminée. Une membrure appelée bielle servira de lien entre deux mécanismes afin de transmettre le mouvement du premier mécanisme au second. En général elle s'accouple à la membrure motrice du premier

mécanisme. Elle devra donc posséder des caractéristiques spécifiques (la longueur notamment) pour remplir sa mission et donner à la sortie le résultat escompté. Sa trajectoire varie en fonction de sa longueur. Afin de prévoir la bonne trajectoire nous visualiserons pour l'utilisateur le parcours de l'extrémité de cette membrure qui sera accouplé au second mécanisme. Cette trajectoire est appelée courbe de bielle manivelle.

* Détermination des positions limites et points morts.

Ces positions sont calculées par les équations (2), (4), (6), (8), (15), (18), (21), (24), (25), (27). Les paramètres de ces positions sont imprimés et une visualisation de ces positions pourra se faire au besoin.

* Détermination de l'angle de transmission.

Cet angle varie au cours du mouvement. Cette variation sera donnée à l'impression. Une position désirée pourra être visualisée. Et pour les M4M l'indice de mérite sera donné pour chaque angle de transmission.

* Variables cinématiques.

Ces variables seront déterminées à l'aide des équations:
(52), (53), (57), (58), (63), (64), (65), (66), (67), (71), (72),

(73), (74), (75), (76), (77), (78), (88), (89), (93), (95), (100),
(101), (111), (112), (116), (117), (119), (120), (123) et (124).

* Reproduction du mécanisme dans une position donnée.

(Impression)

Il s'agira de reproduire sur papier le mécanisme en un instant donné.

Chapitre 2 MODE D'UTILISATION DU PROGRAMME.

1- Procédure de mise en oeuvre.

Le programme est mis sous forme exécutable.

Pour son exécution à partir du système d'exploitation, il suffira de donner l'ordre suivant:

A> SAMP <cr>

L'utilisateur devra disposer d'une disquette de données qui sera, au cours de l'exécution, dans le lecteur B si le poste de travail a deux unités de lecture.

Le programme est interactif. L'utilisateur fournira des réponses adéquates aux questions qui lui seront posées. Il fera les choix qui lui conviennent.

2- Forme des données.

Les données représentant un dessin réalisé sont enregistrées dans un fichier de type file of dessin.

3- Forme des résultats.

Les résultats qui seront fournis à l'utilisateur sont nombreux. IL sera avantageux de choisir le mode "imprimante" pour la sortie de ces résultats au cours de l'exécution. Toutefois, au cas où on ne dispose pas d'imprimante, certains résultats apparaîtront à l'écran. *Ce dernier mode retarde l'exécution du travail*

l'écran. Ce dernier mode retarde l'exécution du travail.

Chapitre 3 : ALGORITHMES

1- Les identificateurs

Voir annexes.

2- Structure des opérations.

* Algorithme pour le dessin.

-répéter

.choisir un point de référence i

.présenter les coordonnées de ce point

.choisir un second point situé à une distance convenable et définissant avec le point précédent, l'inclinaison désirée pour la membrure.

.nommer et joindre les 2 points jusqu'à ce que toutes les membrures de longueur finie soient dessinées.

- A chaque noeud

.mettre un coulisseau si une membrure coulisseau est désirée en ce point dans la position où se trouve le mécanisme.

.mettre un joint cheville dans le contraire

- Sauvegarder le dessin sous un nom.

* algorithme pour le degré de liberté.

- Rappeler le dessin
- Déterminer le nombre de membrures m_j que relie chaque joint cheville
- Faire la somme de $(m_j - 1)$ pour les P_j joints cheville j
- Calculer le degré de liberté F grâce à l'équation (0)
- Si F est inférieur à 1 alors le dessin est une structure. Sinon, c'est un mécanisme

* Algorithme pour le type de transformation de mouvement

- Rappeler le dessin
- Identifier la longueur de chaque membrure
- Appliquer la loi de Grashoff
- Afficher la conclusion

* Simulation du mouvement.

- Rappeler le dessin
- Identifier le groupe du mécanisme M4M, MC ou MCI
- Lire la vitesse de la membrure motrice
- Pour certaines valeurs de l'angle de la membrure d'entrée, calculer les coordonnées des différents joints du mécanisme
- Pour chacune de ces positions sélectionnées,
 - .représenter le mécanisme
 - .effacer le mécanisme de l'écran

- Imprimer les paramètres des différentes positions sélectionnées
- Représenter le mécanisme dans sa position P si l'utilisateur en fait la demande
- Imprimer le mécanisme ainsi présenté.

* Algorithme pour la courbe de bielle.

- Rappeler le dessin
- Demander la longueur et l'angle par rapport à la bielle de la membrure d'accouplement
- Représenter le mécanisme et la membrure d'accouplement
- Pour différentes positions de la membrure motrice,
 - .déterminer les coordonnées x_c et y_c de l'extrémité c de la membrure d'accouplement
 - .représenter le point c.

* Algorithme pour les positions limite et point mort.

- Rappeler le dessin
- Identifier son groupe
- Calculer les paramètres de la position limite ou du point mort
- Imprimer le résultat et le mécanisme dans la position critique

* Algorithme pour l'angle de transmission.

- Rappeler le dessin
- Identifier son groupe
- Pour différentes positions de la membrure d'entrée, calculer l'angle de transmission
- Imprimer les résultats
- Imprimer le mécanisme

* Algorithme pour les variables cinématiques.

- Rappeler le dessin
- Identifier son groupe
- Demander la vitesse et l'accélération de la membrure motrice
- Déterminer les paramètres de la position, la vitesse et l'accélération pour les membrures mobiles
- Imprimer les résultats
- Imprimer le mécanisme.

* Algorithme pour la position instantanée

- Rappeler le dessin
- Identifier son groupe
- Demander la position angulaire de la membrure motrice
- Contrôler si le mécanisme passe dans cette position au cours du mouvement
- Afficher le mécanisme à cet instant

- Donner les paramètres du mécanisme à cet instant
- Imprimer le mécanisme au besoin

* Algorithme pour la visualisation des schéma.

- Rappeler le dessin.
- Représenter le dessin à l'écran.
- Afficher les paramètres du dessin.

3 - Un exemple d'analyse.

Tout en traitant l'exemple qui va suivre nous donnerons toutes les indications nécessaires pour utiliser le logiciel **SAMP1**. C'est/^{donc un}guide que nous élaborons à travers le présent exemple.

EXEMPLE

Considérons un mécanisme à quatre membrures MA, AB, BQ, QM de longueur respective 4 , 8, 5.5 et 5 unités. Considérons qu'à l'instant où l'argument du vecteur MA est $\theta_2 = 120^\circ$, celui du vecteur QB soit $\theta_4 = 62^\circ$ et celui du vecteur MQ=0.

Si MA et QM sont respectivement les membrures d'entrée et de sortie analysons ce mécanisme à l'aide du logiciel SAMP1.

Notre analyse se fera suivant les étapes suivantes.

ETAPE 0: Mise en marche du logiciel

- 1/ Mettre l'ordinateur en marche.
- 2/ Insérer la disquette #1 dans l'unité A.
- 3/ Taper A: et appuyer <Entrée>.
- 4/ Taper SAMP1 et appuyer <Entrée>.

Le menu principal du progiciel apparaît alors comme ci
dessous.

Faire le schéma du mécanisme

```
D : Dessin
V : Visualisation
L : Code de liberté
M : Type de transmission
C : Positions critiques
T : Type de transmission
I : Point de instabilité
G : Angle de crantage
E : Etape de couplage
S : Simulation
Q : Quitter
```

```
ECOLE POLYTECHNIQUE DE THIES
DEPARTEMENT DU GENIE MECANIQUE
PROJET SARA! POTENTE PAR AUDDABBE JEROME
```

REMARQUE PRELIMINAIRE:

Il se peut qu'au cours des prochaines étapes l'ordinateur se bloque. Cela sera dû à un manque d'espace mémoire. Il faut alors l'éteindre pour libérer la mémoire et le remettre en marche.

ETAPE 2:Représentation

Dans cette étape nous réaliserons le schéma du mécanisme. Pour ce faire il faudra:

1/ Appuyer sur la touche <D> ou positionner le rectangle ombragé sur Dessin et appuyer <Entrée>.

2/ Appuyer <ESC>.

Une feuille graphique apparaît alors à l'écran avec quelques instructions.

3/ A l'aide des touches $\uparrow, \downarrow, \rightarrow, \leftarrow, \uparrow, \downarrow, \swarrow$, sélectionner les points M, A, B et Q qui permettent de respecter au mieux les contraintes de longueur et d'orientation des membrures. Comment procéder?

Les touches \uparrow, \downarrow sont celles d'amplification (elles permettent d'augmenter ou de diminuer le pas de déplacement du curseur graphique).

Les touches \leftarrow, \rightarrow sont celles de déplacement.

La touche \swarrow permet de changer d'origine. C'est elle qui nous permettra surtout de faire une esquisse du schéma (sélection des différents noeuds).

Par une combinaison des touches de déplacement et d'amplification positionnons le curseur au point de coordonnées (305, 165). Ce sera l'emplacement du noeud M. Appuyons sur la touche \swarrow . Le point M devient alors l'origine de notre repère (le prochain vecteur ou membrure sera tracé à partir de M). Actionnons encore les touches de déplacement en surveillant les distance (longueur) et angle indiqués au bas du graphique. Nous constatons que le segment de longueur 4 unités est trop petit. Il nous faut donc choisir un facteur d'échelle par lequel seront multipliées toutes les longueurs de membrures afin que notre schéma soit plus visible (aggrandissement). Choisissons comme facteur 20. Les longueurs à représenter deviennent MA=80, AB=160, BQ=110 et QM=100.

Nous recherchons à l'aide des flèches le point A placé à une position telle que l'affichage des distances et angles au bas de l'écran approche le plus possible des valeurs 80 et 120. On

parvient à distance=79.8 et angle=120.1 pour le point A de coordonnées (265,96). Retenons ces coordonnées. Recherchons l'emplacement du point Q. Pour cela ramenons le curseur graphique en M et appuyons sur la touche \wedge pour reconsidérer le point M comme point de départ. Sélectionnons à l'aide des flèches $\downarrow \uparrow \leftarrow \rightarrow$ le point Q tel que la distance et l'angle affichés deviennent 100 et 0. Nous lisons à ce moment les coordonnées du point Q (405,165). Le dernier point à chercher est B tel que l'affichage approche le plus des valeurs distance=110 et angle=62°. On parvient à l'aide des flèches $\uparrow \downarrow \leftarrow \rightarrow$ aux valeurs suivantes: distance=110.1; angle=61.8 correspondant au point B de coordonnées (457,68). Nous connaissons maintenant les coordonnées de tous les noeuds du schéma. Nous pouvons tracer les différents segments qui le constituent. Comment faire le tracer?

Ramener le curseur en M (305,165).

Appuyer sur la touche \wedge : M devient le point de départ.

Amener le curseur en A (265,96).

Appuyer sur <Entrée>: à droite et en bas de l'écran se lit le message: donner le nom du joint 1

Taper M .

On lit alors le message: donner le nom du joint 2

Taper A : le segment MA se trace alors.

Amener le curseur en B (457,68) et appuyer <Entrée>.

Taper B: le segment AB se trace.

Amener le curseur en Q (405,165) et appuyer <Entrée>.

Taper Q: le segment BQ se trace.

Amener le curseur en M (305,165) et appuyer <Entrée>:le segment QM se trace.

IL nous reste à placer aux noeuds adéquats les joints chevillés ou les coulisseaux. Dans le cas du présent mécanisme nous avons un joint cheville à chaque noeud. Il faut appuyer sur <F10> pour commencer et taper 1 à chaque question " type de joint en M (A, B, Q)".

Après la mise en place de ces joints (et ou cheville), il faut nommer le dessin réalisé après avoir placé la disquette de données dans le lecteur B. Donnons le nom "UN" à ce dessin.

ETAPE 3: Visualisation du schéma réalisé

Cette étape nous permet de revoir le schéma que nous venons de faire et de l'imprimer au besoin.

1/ Taper V ou positionner le rectangle ombragé sur VISUALISATION et appuyer <Entrée>.

2/ Taper esc.

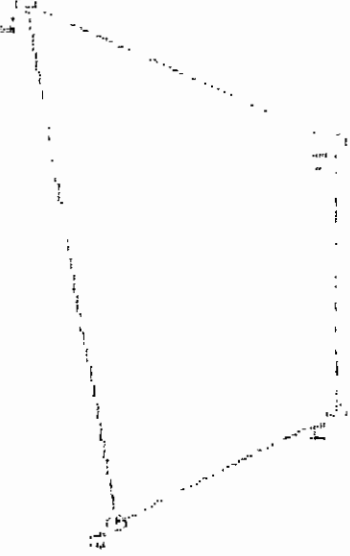
3/ Taper 1 (notre mécanisme n'a pas de coulisseau) et appuyer <Entrée>.

4/ Taper UN (c'est le nom que nous avons donné au dessin réalisé) et appuyer <Entrée>.

5/ Taper O pour faire une impression ou N dans le cas contraire.

6/ Appuyer sur une touche (pour lancer l'impression) ou sur esc (dans le cas contraire).

On obtient le résultat à la page suivante.



1901
 1902
 1903
 1904
 1905
 1906
 1907
 1908
 1909
 1910
 1911
 1912
 1913
 1914
 1915
 1916
 1917
 1918
 1919
 1920
 1921
 1922
 1923
 1924
 1925
 1926
 1927
 1928
 1929
 1930
 1931
 1932
 1933
 1934
 1935
 1936
 1937
 1938
 1939
 1940
 1941
 1942
 1943
 1944
 1945
 1946
 1947
 1948
 1949
 1950
 1951
 1952
 1953
 1954
 1955
 1956
 1957
 1958
 1959
 1960
 1961
 1962
 1963
 1964
 1965
 1966
 1967
 1968
 1969
 1970
 1971
 1972
 1973
 1974
 1975
 1976
 1977
 1978
 1979
 1980
 1981
 1982
 1983
 1984
 1985
 1986
 1987
 1988
 1989
 1990
 1991
 1992
 1993
 1994
 1995
 1996
 1997
 1998
 1999
 2000

ETAPE 4: Analyse de la mobilité du schéma.

Ici nous étudierons le schéma UN pour vérifier s'il représente effectivement un mécanisme ou non.

1/ Taper L ou positionner le rectangle ombragé sur DEGRE DE LIBERTE et appuyer <Entrée>.

2/ Appuyer esc.

3/ Taper UN et appuyer <Entrée>.

4/ Appuyer sur <Impr écran> pour imprimer l'écran.
On obtient le résultat ci-dessous.

LE DEGRE DE LIBERTE EST: 1

VOUS AVEZ UN MECANISME

appuyer une touche pour continuer

5/ Appuyer sur une touche pour retourner au menu principal.

ETAPE 5: Etude du type de mouvement.

Cette étape permet d'étudier le type de mouvement de notre mécanisme.

1/ Taper M ou positionner le rectangle ombragé sur TYPE DE TRANSMISSION et appuyer <Entrée>.

2/ Appuyer esc.

3/ Taper UN et appuyer <Entrée>.

4/ Répondre correctement aux questions éventuelles (pour le mécanisme UN il n'y a pas de questions mais pour d'autres mécanismes l'utilisateur pourrait avoir à préciser les différentes membrures).

5/ Imprimer l'écran.

On obtient le résultat ci-dessous.

ce dessin est un balancier-balancier

appuyer une touche pour continuer

6/ Appuyer une touche pour revenir au menu principal.

ETAPE 5: Analyse des positions particulières.

Ici nous étudierons les positions de point mort et les positions limites. Une imprimante est requise pour la sortie des résultats.

1/ Taper C ou positionner le rectangle ombragé sur POSITIONS CRITIQUES et appuyer sur <Entrée>.

2/ Appuyer sur esc.

3/ Taper 1 et appuyer <Entrée>.

4/ Taper UN et appuyer <Entrée>.

5/ Préciser les différentes membrures en tapant le nom de la membrure suivi de <Entrée>. (entrée MA, intermédiaire:AB, sortie:BQ, fixe: MQ).

6/ Appuyer esc.

On obtient:

A la position limite 2
la membrure motrice a une inclinaison de 244.7 degrés

Au point mort 2
la membrure motrice a une inclinaison de 59.9 degrés

7/ Taper 0 pour revenir au menu principal.

ETAPE 7: Etude de l'angle de transmission.

Il s'agit d'étudier les angles de transmission et l'efficacité de la transmission en fonction des différentes positions angulaires que traversera la membrure d'entrée au cours du mouvement (nous n'avons pas envisagé toutes les positions angulaires; cela est pratiquement impossible).

1/ Taper T ou positionner le rectangle ombragé sur ANGLE DE TRANSMISSION et appuyer <Entrée>.

2/ Mettre la disquette #2 en A et appuyer esc.

3/ Taper 1 et appuyer <Entrée>.

4/ Taper UN. et appuyer <Entrée>.

5/ Préciser les différentes membrures.

6/ Appuyer esc.

ON obtient le résultat:

Angle de transmission μ et efficacité η
 position entrée

position entrée	μ	η
121.1	59.6	-35.4
122.1	59.9	-33.1
123.1	60.2	-24.1
124.1	60.5	-22.7
125.1	60.8	-18.5
127.1	61.4	-12.0
128.1	61.7	-12.2
129.1	61.9	-11.8
130.1	62.2	-10.5
131.1	62.4	-9.5
132.1	62.6	-9.5
133.1	62.9	-8.6
135.1	63.3	-7.4
137.1	63.7	-6.9
139.1	63.9	-6.5
139.1	64.0	-6.1
140.1	64.2	-5.8
142.1	64.5	-5.5
144.1	64.8	-5.0
145.1	65.0	-4.9
146.1	65.1	-4.8
148.1	65.4	-4.4
149.1	65.5	-4.3
151.1	65.7	-4.0
152.1	65.8	-3.8
153.1	65.9	-3.9
154.1	66.0	-3.8
155.1	66.1	-3.5
156.1	66.2	-3.5
158.1	66.3	-3.4
160.1	66.5	-3.2
162.1	66.6	-3.1
163.1	66.7	-3.0
164.1	66.7	-2.9
166.1	66.8	-2.8
167.1	66.9	-2.8
168.1	66.9	-2.7
169.1	66.9	-2.7
170.1	67.0	-2.7
171.1	67.0	-2.5
172.1	67.0	-2.6
173.1	67.1	-2.7
176.1	67.1	-2.5
180.1	67.1	-2.3
181.1	67.1	2.8
182.1	67.1	2.3
183.1	67.1	2.1
184.1	67.1	2.3
185.1	67.1	2.3
186.1	67.1	2.6
187.1	67.1	2.7
188.1	67.0	2.5

167.1	67.0	3.7
170.1	67.0	3.7
191.1	66.9	3.7
192.1	66.9	3.8
193.1	66.9	3.8
194.1	66.8	3.8
195.1	66.8	3.9
196.1	66.7	3.9
197.1	66.7	3.9
198.1	66.6	3.1
199.1	66.5	3.2
200.1	66.5	3.2
201.1	66.4	3.2
202.1	66.3	3.3
203.1	66.3	3.5
204.1	66.2	3.5
205.1	66.1	3.6
206.1	65.9	3.9
207.1	65.9	4.0
208.1	65.8	4.0
209.1	65.7	4.1
211.1	65.5	4.4
212.1	65.3	4.5
213.1	65.2	4.5
214.1	65.1	4.7
215.1	65.0	4.9
216.1	64.8	5.1
217.1	64.7	5.3
221.1	64.0	6.2
222.1	63.9	6.6
224.1	63.5	7.3
225.1	63.2	7.6
226.1	63.0	8.1
227.1	62.8	8.8
228.1	62.6	9.7
229.1	62.4	9.7
230.1	62.1	10.9
231.1	61.9	12.3
232.1	61.6	12.6
233.1	61.4	14.6
234.1	61.1	17.2
235.1	60.8	19.3
236.1	60.5	20.3
237.1	60.2	26.0
238.1	59.9	39.1
239.1	59.5	52.4
240.1	59.2	121.9
241.1	58.8	188.3
242.1	58.5	-176.4
244.1	57.7	-50.5
245.1	57.3	-27.1
246.1	56.9	-23.7
247.1	56.5	-17.4
249.1	55.6	-14.6
250.1	55.2	-12.8
253.1	53.7	-9.1

254.1	53.2	-8.1
255.1	52.6	-7.4
257.1	51.5	-6.3
258.1	51.0	-5.8
259.1	50.4	-5.3
260.1	49.8	-5.1
261.1	49.2	-4.9
264.1	47.2	-4.0
265.1	46.6	-3.7
267.1	45.2	-3.4
268.1	44.5	-3.3
269.1	43.8	-3.1
270.1	43.0	-2.9
271.1	42.3	-2.8
272.1	41.5	-2.6
273.1	40.7	-2.5
274.1	40.0	-2.3
275.1	39.2	-2.2
276.1	38.4	-2.1
280.1	35.0	-1.7
281.1	34.2	-1.6
282.1	33.3	-1.5
283.1	32.4	-1.5
284.1	31.5	-1.4
285.1	30.6	-1.3
286.1	29.7	-1.2
288.1	27.8	-1.1
289.1	26.9	-1.0
290.1	25.9	-1.0
292.1	23.8	-0.9
294.1	21.7	-0.7
295.1	20.6	-0.7
297.1	18.4	-0.6
299.1	15.9	-0.5
301.1	13.1	-0.4
302.1	11.5	-0.3
303.1	9.7	-0.3
304.1	7.6	-0.2
305.1	4.5	-0.1
55.1	3.3	0.1
57.1	10.1	0.3
58.1	11.3	0.3
59.1	13.4	0.4
60.1	14.8	0.4
61.1	16.1	0.5
62.1	17.4	0.5
63.1	18.4	0.6
64.1	19.3	0.6
65.1	20.0	0.7
66.1	22.0	0.8
67.1	23.0	0.8
68.1	24.1	0.9
69.1	25.1	0.9
71.1	27.0	1.0

72.1	28.0	1.1
73.1	28.9	1.2
74.1	29.5	1.3
75.1	30.6	1.3
76.1	31.7	1.4
77.1	32.6	1.5
78.1	33.5	1.6
79.1	34.3	1.6
80.1	35.2	1.7
81.1	36.0	1.8
82.1	36.9	1.9
83.1	37.7	2.0
84.1	38.5	2.1
85.1	39.3	2.2
86.1	40.1	2.4
87.1	40.9	2.5
88.1	41.7	2.7
89.1	42.4	2.8
90.1	43.2	2.9
91.1	43.9	3.1
92.1	44.6	3.3
93.1	45.3	3.5
94.1	46.0	3.6
95.1	46.7	3.8
96.1	47.4	4.1
99.1	49.3	4.9
100.1	49.9	5.4
101.1	50.5	5.6
102.1	51.1	6.0
103.1	51.7	6.4
104.1	52.2	7.0
105.1	52.8	7.6
106.1	53.3	8.3
107.1	53.9	9.4
108.1	54.3	10.3
109.1	54.8	11.0
110.1	55.2	13.3
111.1	55.7	13.3
112.1	56.2	16.7
113.1	56.6	18.4
114.1	57.0	26.9
115.1	57.4	29.5
116.1	57.8	39.6
117.1	58.2	73.1
118.1	58.6	116.3
119.1	58.7	130.4
120.1	59.3	139.5

7/ Taper 0 pour revenir au menu principal.

ETAPE 8 : Position instantannée.

Les résultats de l'étape précédente nous montrent quelques positions angulaires qu'occupe la membrure d'entrée au cours du mouvement. La présente étape nous permet de visualiser le mécanisme lorsque la position angulaire de la membrure d'entrée est connue. Supposons que nous voulions visualiser le mécanisme UN lorsque la position angulaire de la membrure d'entrée est de 65.1 degrés dans le sens trigonométrique. Nous procéderions comme suit:

1/ Taper I ou positionner le rectangle ombragé sur POSITION INSTANTANNEE et appuyer <Entrée>.

2/ Appuyer esc.

3/ Taper 1 et appuyer <Entrée>.

4/ Taper UN et appuyer <Entrée>.

5/ Préciser les différentes membrures.

6/ Taper 0 (sortie sur imprimante).

7/ Taper 65.1 (position angulaire de la membrure d'entrée).

8/ Appuyer une touche pour lancer l'impression du mécanisme.

On obtient le résultat:

LES PARAMETRES DU MAM				
	entrée	inter.	sortie	fixe
	ma	ab	bq	qm
longueur	79.8	194.0	110.1	100.0
penche	65.1	220.2	90.5	150.0

segment	length	angle
ab	79.8	65.1
bc	194.0	290.2
cd	119.1	90.5
da	100.0	180.0



9/ Appuyer esc (pour revenir au menu principal).

ETAPE 9: Analyse cinématique.

Ici nous nous intéressons aux variables cinématiques du mécanisme à un instant où la position (dégrés), la vitesse (tours par seconde) et l'accélération (tours par seconde carré) angulaires de la membrure d'entrée sont connues. Faisons cette analyse lorsque la vitesse de la membrure d'entrée est de 15 tours par seconde (signe positif dans le sens anti horaire et signe négatif dans le sens horaire), son accélération angulaire est de 0 tours par seconde carré et sa position 120.1 dégrés.

1/ Taper A ou positionner le rectangle ombragé sur ANALYSE CINEMATIQUE.

2/ Appuyer esc.

3/ Taper 1 et appuyer <Entrée>.

4/ Taper UN et appuyer <Entrée>.

5/ Préciser les différentes membrures.

6/ Taper 120.1 et appuyer <Entrée>.

7/ Taper 15 et appuyer <Entrée>.

8/ Taper 0 et appuyer <Entrée>.

9/ Imprimer l'écran au besoin.

10/ Suivre les instructions au bas de l'écran.

On obtient le résultat:

```
donner la position angulaire de la membrure d'entrée: 120.1
donner la vitesse angulaire de la membrure d'entrée : 15
donner l'accélération angulaire de la membrure d'entrée: 0
```

déplacement

entité	module	argument
noeud m/g	1007.0	180.0
noeud a	79.8	120.1
noeud b	194.0	8.3

appuyer sur <BARRE D'ESPACEMENT> pour continuer

vitesse

entité	module	argument
noeud a	1196.3	149.9
noeud b/a	1266.3	261.7
noeud s	1392.0	208.2

appuyer sur <BARRE D'ESPACEMENT> pour continuer

accélération radiale

entité	module	argument
noeud a	17945.1	59.9
noeud b/a	8260.5	171.7
noeud b	17345.0	0.0

appuyer sur <BARRE D'ESPACEMENT> pour continuer

accélération tangentielle

entité	module	argument
noeud a	0.0	0.0
noeud b/a	3733.1	119.2
noeud b	10846.2	29.2

appuyer sur <BARRE D'ESPACEMENT> pour continuer

entité	position ang.	itesse ang.	accélération ang.
membrure 1	120.1	15.0	0.0
membrure 2	7.7	6.7	0.0
membrure 3	41.3	12.6	0.0

Appuyer sur <SPACE> pour continuer

11/ Taper 0 pour revenir au menu principal.

Remarque: Il faut retenir que dans le schéma du mécanisme nous avons choisi un facteur multiplicatif de 20 pour les longueurs des membrures. Les variables cinématiques telles que les modules de déplacement, vitesse, accélération radiale, accélération tangentielle ont subi aussi l'effet de ce facteur multiplicatif. Pour connaître donc leur valeur dans le mécanisme de départ il faut diviser leur valeur présente par le facteur 20. Les autres paramètres sont indépendantes des longueurs.

ETAPE 10: ETUDE DU COUPLAGE

Cette étape permet de tracer la courbe de bielle en vue du couplage à d'autres mécanismes.

1/ Taper E ou positionner le rectangle ombragé sur ETUDE DE COUPLAGE et appuyer <Entrée>.

2/ Appuyer esc.

3/ Taper 1 et appuyer <Entrée>.

4/ Taper UN et appuyer <Entrée>.

5/ Préciser les différentes membrures.

6/ Suivre les instructions affichées à l'écran graphique et suivre le curseur graphique lors de la combinaison des flèches.

7/ Appuyer sur F10 pour quitter.

ETAPE 11: La simulation du mouvement.

Cette étape nous permet de simuler le mouvement du mécanisme lorsque la membrure d'entrée est en rotation uniforme.

1/ Taper S ou positionner le rectangle ombragé sur SIMULATION et appuyer <Entrée>.

2/ Appuyer esc.

3/ Taper 1 et appuyer sur <Entrée>.

4/ Taper UN et appuyer sur <Entrée>.

5/ Préciser les différentes membrures et donner la vitesse en tours par seconde de la membrure d'entrée.

6/ Appuyer une touche pour revenir au menu principal.

Chapitre 4: Analyse des méthodes utilisées et des résultats

Le support fondamental de ce logiciel est le graphisme. Notre travail repose sur l'analyse d'un schéma du mécanisme. De la précision de ce schéma dépend l'exactitude des résultats. L'utilisateur pourrait être surpris de se voir refuser des analyses qu'il désire faire sur certains mécanismes notamment les mécanismes manivelle-coulisseau inversés. Ceci est dû au fait que ces mécanismes ont des propriétés qu'il aurait négligées lors de la réalisation du schéma. En effet pour aider l'utilisateur dans la réalisation du schéma, les coordonnées du point courant ainsi que la longueur et la pente que fait le segment en cours sont affichées. Si les coordonnées sont les vrais paramètres du point, l'angle et la longueur lus sont en fait les valeurs arrondies de calcul. L'utilisateur devra s'en rappeler. Il pourra sélectionner des points précis pour la réalisation de schémas exacts.

Dans la simulation, une oscillation de un ou deux millimètres peut se noter sur des trajectoires précises comme celles d'un coulisseau dans un mécanisme-coulisseau. Cela est lié au fait que plusieurs variables représentant des coordonnées des noeuds du dessin ont été arrondies aux entiers les plus proches au cours du calcul pour respecter les contraintes du graphisme.

Le choix l'échelle et les unités de mesure a été laissé libre à l'utilisateur. Pour obtenir des résultats cohérents dans l'analyse, nous suggérons de donner les positions angulaires dans

le repère trigonométrique et en degré, les vitesses angulaires de la membrure motrice en tours par seconde, les accélérations angulaires en tours par seconde carré.

La simulation a été faite en mouvement uniforme. Cela constitue une limite pour la présente version du progiciel.

le repère trigonométrique et en degré, les vitesses angulaires de la membrure motrice en tours par seconde, les accélérations angulaires en tours par seconde carré.

La simulation a été faite en mouvement uniforme. Cela constitue une limite pour la présente version du logiciel.

CONCLUSION ET RECOMMANDATIONS

RECOMMANDATIONS

1. Utiliser une imprimante pour exploiter toutes les ressources de ce logiciel.
2. Exécuter le travail de préférence sur un poste XT IBM PC monochrome.
3. Visualiser le mécanisme pour éviter les incompatibilités de données dues à une mauvaise sélection de dessin.
4. Sélectionner une membrure de longueur finie comme membrure motrice.
5. Ce travail pourra être complété par la simulation en régime stationnaire des mécanismes plans.

CONCLUSION

En somme la présente étude a permis de réaliser un outil interactif de design des mécanismes.

La détermination des positions critiques indique à l'utilisateur les points particuliers qui pourraient interrompre le mouvement. L'analyse des angles de transmission constitue une étape importante dans la conception et la vérification des mécanismes. De cet angle dépendra l'efficacité de la transmission. L'étude cinématique offre les moyens de connaître les paramètres des membrures lorsque les variables d'entrée sont connues. Le programme vérifie la cohérence des variables imposées à la membrure motrice avec la nature même du mécanisme. Il réalise également la simulation du mécanisme. Enfin l'étude du couplage offre un élément de base (la courbe de bielle) pour passer des mécanismes à quatre membrures à des mécanismes complexes.

C'est dire que l'étude cinématique se réalise bien avec le logiciel SAMPl. En exploitant au mieux les avantages de précision de calculs et de vitesse d'exécution des opérations, permet de visualiser le mouvement des mécanismes de base. Cependant ce travail se fait sous l'hypothèse de la rigidité des membrures ce qui ne correspond pas souvent à la réalité.

Voilà pourquoi ce travail devra être complété et amélioré pour embrasser dans un proche avenir, l'étude dynamique et la synthèse des mécanismes ainsi que l'analyse assistée par ordinateur de leurs déformations élastiques.

ANNEXES

- A Bibliographie
- B Identificateur
- C Listing du progiciel.

A/ BIBLIOGRAPHIE

-Atmaran H. Soni,

"Mechanism synthesis and Analysis",

Mc Graw-Hill Book Company USA

1974

-Diallo M. L., Notes supplémentaires,

EPT 1958.

-Ferdinand P. Beer and E. Russel Johnston;

"Vector Mechanics for Engineers Dynamics" fourth
edition

Mc Graw-Hill Book Company USA

1984

- H. E. Barnacle and G. E. Walker,

"Mechanisms of machines" vol2.

Pergamon Press London

1965

-H. H. Mable and F. W. Dyrvik,

"Mechanisms and Dynamics of Machinery"

third edition SI version

John Wiley & sons, Inc USA

1976

-Shigley, J. E.,

"Kinematic Analysis of Mechanisms",

Mc Graw-Hill USA

1969

- Youssef, Y. A.,

Notes de cours: Mécanismes, EPT

1982

- Donald HEARN and M. Pauline BAKER,

"Computer Graphics",

Prentice-Hall, Englewood Cliffs, New Jersey 07632

1986

- Gabriel Cuellar,

"Graphismes sur IBMPC /XT/ compatibles"

3^e Edition,

Eyrolles Paris

-Herbert Schildt,

"Advanced Turbo Pascal"

Borland-Osborne/McGraw-Hill

-Jacques Boisgontier,

"Turbo Pascal version 4"

Édico Nathan France

1989

B- Les identificateurs.

NOM	TYPE	DESCRIPTION
driver	integer	paramètre de la carte graphique
t,i,r,v	integer	compteurs
w,p	integer	compteurs
g	integer	indice de la membrure motrice
k	integer	indice de la membrure intermédiaire
h	integer	indice de la membrure de sortie
f	integer	indice de la membrure fixe
j	integer	nombre de noeuds
s	integer	nombre de segments
pas	integer	incrément du curseur graphique
orig	integer	indice de l'origine d'un vecteur
extr	integer	indice de l'extrémité d'un vecteur
jointA	integer	indice de l'extrémité de la membrure motrice
jointB	integer	indice de l'extrémité de la membrure auxilliaire
jointQ	integer	indice de l'extrémité de la membrure fixe

jointM	integer	indice de l'origine de la membrure motrice
xi	longint	abscisse de l'intersection de deux courbes
yi	longint	ordonnée de l'intersection de deux courbes
x	longint	abscisse du point courant
y	longint	ordonnée du point courant
xa	longint	abscisse du point précédent
ya	longint	ordonnée du point précédent
xjoint	longint	abscisse du joint
yjoint	longint	ordonnée du joint
absc	array[1..20] of longint	tableau des abscisses des noeuds
ordo	array[1..20] of longint	tableau des ordonnées des noeuds
abscA	array[1..360] of longint	tableau des abscisses successives du noeud A
ordoA	array[1..360] of longint	tableau des ordonnées successives du noeud A
long	array[1..190] of real	tableau des longueurs des membrures
angle	array[1..190] of real	tableau des angles des membrures
prisme	array[1..20] of real	tableau des angles des axes coulisseau
angleG	array[1..360] of real	tableau des positions angulaires successives de la membrure d'entrée
angleK	array[1..360] of real	tableau des positions angulaires de la membrure intermédiaire
angleH	array[1..360] of real	tableau des positions

		angulaires successives de la membrure de sortie
module	real	module d'un vecteur
argument	real	argument d'un vecteur
rp	real	argument du vecteur courant
rr	real	module du vecteur courant
delta	real	angle ou discriminant
zeta	real	angle
theta	real	angle
pen	real	angle
alpha	real	angle
car	char	donnée à lire
sort	char	caractère de sortie
joint	array[1..20] of char	tableau des noms des noeuds
typejoint	array[1..20] of char	tableau des noms de type de joint
ecrase	char	caractère de controle pour supprimer un dessin
position	char	caractère
nomdessin	string[14]	nom du dessin
xst	string[14]	nombre converti en mot
yst	string[14]	nombre converti en mot
ext	string[14]	nombre converti en mot
jo	string[14]	nombre converti en mot
se	string[14]	nombre converti en mot
pen	string[14]	nombre converti en mot
resultat	string	phrase réponse
fichier	file	fichier
nom	file of dessin	fichier contenant les paramètres du dessin
ok	boolean	contrôle une action

cadre	procedure	elle trace le cadre du dessin
gommer	procedure	elle efface le dessin courant
lire	procedure	elle lit le nom du dessin
trigo	procedure	elle donne l'angle défini par le vecteur courant
interarcs	procedure	elle donne les coordonnées de l'intersection de deux arcs
fenetre	procedure	elle définit une fenêtre graphique
sortie	procedure	elle permet de sortir d'un menu
segment	procedure	elle baptise le segment défini par deux points
coordonne	procedure	elle donne les coordonnées du point, la longueur et l'argument du vecteur courants
point	procedure	elle baptise les noeuds
existe	fonction	elle vérifie l'existence d'un dessin
photo	procedure	elle capte l'image instantanée de l'écran
cliche	procedure	elle restitue l'image captée
nommer	procedure	elle baptise un dessin
appeler	procedure	elle appelle un dessin
sauver	procedure	elle sauvegarde un dessin
charger	procedure	elle transfère le dessin

PLMCI	procedure	elle détermine les positions limites d'un manivelle coulisseau inversé
ATM4M	procedure	elle détermine les angles de transmission d'un mécanisme M4M
ATMC	procedure	elle détermine les angles de transmission d'un manivelle-coulisseau
ATMCI	procedure	elle détermine les angles de transmission d'un manivelle coulisseau inversé
complexe	procedure	elle calcule le module et l'argument d'un nombre complexe
dist	procedure	elle calcule la distance entre deux points
interdroites	procedure	elle détermine l'intersection de deux droites
PVA-M4M	procedure	elle fait l'analyse cinématique d'un mécanisme M4M
PVA-MC	procedure	elle fait l'analyse cinématique d'un manivelle coulisseau
PVA-MCI	procedure	elle fait l'analyse cinématique d'un manivelle coulisseau inversé
menupos	procedure	elle affiche le menu pour l'étude des positions critiques
menuAT	procedure	elle affiche le menu pour l'analyse des angles de transmission

		appelé en mémoire
coulisseau	procedure	elle dessine une membrure coulisseau
restituer	procedure	elle montre le dessin appelé à l'écran
prismatique	procedure	elle identifie une membrure coulisseau
placerjoint	procedure	elle place les joints chevilles et les membrures coulisseaux
schema	procedure	elle permet de réaliser un dessin
natureM4M	procedure	elle détermine le type de mouvement d'un M4M
degre	procedure	elle donne le degré de liberté d'un dessin treillis
visualisation_M4M	procedure	elle visualise un M4M
PLPM4M	procedure	elle détermine les positions limites et points morts d'un M4M
Visualisation_MC	procedure	elle visualise un mécanisme manivelle-coulisseau
tg	fonction	elle définit la fonction tangente
interarcdroite	procedure	elle détermine l'intersection d'une droite et d'un arc de cercle
PLPMC	procedure	elle détermine la position limites et point mort d'un manivelle coulisseau
veriMC12	procedure	elle identifie un manivelle coulisseau inversé
visualisationMC1	procedure	elle visualise un manivelle coulisseau inversé

menucine	procedure	elle affiche le menu pour l'étude cinématique
menupoin	procedure	elle affiche le menu pour la visualisation de position instantanée
menusim	procedure	elle affiche le menu pour la simulation
menucou	procedure	elle affiche le menu pour du couplage
POINM4M	procedure	elle visualise un mécanisme M4M dans une position instantanée
POINMC	procedure	elle visualise un manivelle coulisseau dans une position instantanée
POINMCI	procedure	elle visualise un manivelle coulisseau inversé dans une position instantanée
simM4M	procedure	elle simule un mécanisme M4M dans son mouvement
simMC	procedure	elle simule un manivelle-coulisseau dans son mouvement
simMCI	procedure	elle simule un manivelle coulisseau inversé dans son mouvement
couM4M	procedure	elle étudie le couplage à un mécanisme M4M
couMC	procedure	elle étudie le couplage à un manivelle-coulisseau
couMCI	procedure	elle étudie le couplage à un manivelle coulisseau inversé
menu1	procedure	elle affiche le menu principal

C/ LE LISTING.

MODULE 1: LES VARIABLES GLOBALES (unit0)

```
UNIT Unit0;
```

INTERFACE

```
Uses crt,graph;
```

```
type
```

```
dessin=record
```

```
    sj,ss:integer;
```

```
    sx,sy:longint;
```

```
    sabsc,sordo:array[1..20]of longint;
```

```
    sangle,slong:array[1..190] of real;
```

```
    sjoint,stypejoint:array[1..20] of char;
```

```
    sseg:array[1..190] of string[2];
```

```
    sprisme:array[1..20] of real;
```

```
end;
```

```
var
```

```
    grmode,driver:integer;
```

```
    t,r,i,j,pas,h,s,orig,extr,v:integer;
```

```
    p,ce,g,w,jointA,jointB,jointQ,jointM,f,k:integer;
```

```
    memx,memy:array[1..4] of integer;
```

```
    taille,tail,tailecran:word;
```

```
    xi,yi,x,y,xa,ya,xjoint,yjoint:longint;
```

```

absc:array[1..20] of longint;
ordo:array[1..20] of longint;
abscA,ordoA,abscB,ordoB:array[1..360] of longint;
imecran,pimage,ppim,curs_im,memCurs_im:pointer;
angle:array[1..190] of real;
long:array[1..190] of real;
prisme:array[1..20] of real;
angleG,angleK,angleH:array[1..360] of real;
module,argument,rp,rr:real;
delta,zeta,theta,pente,alpha:real;
car,sort :char;
ecran :array[1..16000] of char absolute $8800:$0000;
joint:array[1..20] of char;
typejoint:array[1..20] of char;
ecrase,position:char;
ndessin,xst,yst,ext,jo,se,pen:string[14];
seg:array[1..190] of string[2];
nomdessin,resultat:string;
fichier:file;
nom:file of dessin;
ok:boolean;

```

IMPLEMENTATION

Begin

End.

MODULE 2: LES PROCEDURES D'ORDRE GENERAL (progene)

UNIT Progene;

INTERFACE

Uses crt,graph,unit0,printer;

Procedure cadre;

Procedure gommer;

Procedure lire(var chaîne:string);

procedure trigo(x1,y1,x2,y2:longint);

Procedure interarcs(x1,y1,x2,y2:longint;r1,r2:real);

Procedure fenetre;

Procedure sortie;

Procedure segment(xa,ya,x,y:longint);

procedure coordonne(x,y:longint);

procedure point(xa,ya,x,y:longint);

Function existe(nomdessin:string):boolean;

procedure photo;

procedure cliche;

procedure nommer;

Procedure appeler;

Procedure sauver(nomdessin:string);

Procedure charger(nomdessin:string);

Procedure coulisseau(xjoint,yjoint:longint;p:real);

Procedure prismatique2;

Procedure restituer (nomdessin:string);

Procedure prismatique(x,y:longint);

Procedure placerjoint(xjoint,yjoint:longint);

Procedure visualisation_M4M (x1,y1,x2,y2,x3,y3,x4,y4:longint);

Procedure visualisation_MC(x1,y1,x2,y2,x3,y3:longint);

function tg(p:real):real;

Procedure interarcdroite(x1,y1,x2,y2:longint;r,p:real);

Procedure veriMC12(nomdessin:string);

Function veri2_MCI(nomdessin:string):boolean;


```

Procedure visualisationMCI(x1,y1,x2,y2,x3,y3,x4,y4:longint);
Procedure complexe(a,b:longint);
Function dist(x1,y1,x2,y2:longint):real;
Procedure interdroides(x1,y1,x2,y2:longint;p1,p2:real);

```

IMPLEMENTATION

```

Procedure cadre;
(elle initialise l'écran graphique)
begin
  detectgraph(driver,grmode);
  initgraph(driver,grmode,'');
  setlinestyle(0,0,3);
  rectangle(0,0,getmaxx,getmaxy);
  setlinestyle(0,0,1);
end;

Procedure gommer;
(elle efface le dessin)
begin
  setviewport(105,12,695,305,false);
  clearviewport;
  setviewport(0,0,getmaxx,getmaxy,false);
end;

Procedure lire(var chaine:string);
var
  car:char;
begin
  chaine:='';
  car:=readkey;
  while car<>#013 do

```

```

begin
  write(car);
  chaine:=chaine+car;
  car:=readkey;
end;
end;

procedure trigo(x1,y1,x2,y2:longint);
{elle détermine l'angle entre les points 1 et 2}
begin
  if x2=x1 then
    begin
      if y2<y1 then zeta:=90;
      if y2>y1 then zeta:=270;
      if y2=y1 then zeta:=0;
    end
  else
    begin
      zeta:=(180/pi)*(arctan((y2-y1)/(x2-x1)));
      if ((x2>x1)and(y2<=y1)) then zeta:=-zeta;
      if ((x2>x1)and(y2>y1)) then zeta:=360-zeta;
      if ((x2>x1)and(y2=y1)) then zeta:=0;
      if (x2<x1) then zeta:=180-zeta;
    end;
end;

Procédure interarcs(x1,y1,x2,y2:longint;r1,r2:real);
{elle détermine l'intersection (xi,yi) de 2 cercles (xi étant au max)}

```

```

var
.
.
b,a, a1,b1,c1:real;
.
begin
if y2<>y1 then
.
begin
a:=(x1-x2)/(y2-y1);b:=(r1*r1-r2*r2+x2*x2-x1*x1+y2*y2-y1*y1)/(2*(y2-y1));
a1:=a*a+1;
b1:=a*(b-y1)-x1;
c1:=(b-y1)*(b-y1)+x1*x1-r1*r1;
delta:=b1*b1-a1*c1;
if delta>=0 then
begin
if ((x1<x2)and((-b1-sqrt(delta))/a1>=0)) then xi:=round((-b1-sqrt(delta))/a1)
else xi:=round((-b1+sqrt(delta))/a1);
yi:=round(a*xi+b);
end;
end
else
begin
if x1<>x2 then
begin
xi:=round(((r1*r1-r2*r2)/(x2-x1)+x1+x2)/2);
if (r1>abs((xi-x1))) then
yi:=round(y1+sqrt(r1*r1-(xi-x1)*(xi-x1)));
end;
end;
end;
end;

```

Procedure fenetre;

```
begin
```

```

setviewport(1,320,300,340,false);
clearviewport;
setviewport(0,0,getmaxx,getmaxy,false);
end;

Procedure sortie;
begin
gotoXY(30,24);write('appuyer esc pour sortir');
sort:=readkey;
end;

Procedure segment(xa,ya,x,y:longint);
begin
s:=s+1;
str(s,se);
angle[s]:=rp;
long[s]:=rr;
orig:=0;
repeat
orig:=orig+1;
until ((absc[orig]=xa) and (ordo[orig]=ya));
extr:=0;
repeat
extr:=extr+1;
str(extr,ext);
until ((absc[extr]=x) and (ordo[extr]=y));
seg[s]:=joint[orig]+joint[extr];
end;

procedure coordonne(x,y:longint);

```

(elle donne les coordonnées du point courant et la distance qui le sépare du dernier point)

```
begin
  str(x,xst);
  str(y,yst);
  rr:=(sqrt((x-xa)*(x-xa)+(y-ya)*(y-ya)));
  if x=xa then
    begin
      if y<ya then rp:=90;
      if y>ya then rp:=270;
      if y=ya then rp:=0;
    end
  else
    begin
      rp:=(180/pi)*(arctan((y-ya)/(x-xa)));
      if ((x>xa)and(y<=ya)) then rp:=-rp;
      if ((x>xa)and(y>ya)) then rp:=360-rp;
      if ((x>xa)and(y=ya)) then rp:=0;
      if (x<xa) then rp:=180-rp;
    end;
  setviewport(300,320,650,340,false);
  clearviewport;
  setviewport(0,0,getmaxx,getmaxy,false);
  outtextxy(350,325,'ABSCISSE = '+xst+' ORDONNEE = '+yst);
  str(rr:4:1,xst); str(rp:4:1,yst);
  outtextxy(350,333,'DISTANCE = '+xst+' ANGLE = '+yst);
end;
```

```
procedure point(xa,ya,x,y:longint);
```

```
begin
```

```

v:=0;
for orig:=1 to j do
begin
if ( (absc[orig]=xa) and (ordo[orig]=ya) ) then
begin
v:=1;
h:=orig;
end;
end;
if v=0 then
begin
j:=j+1;
str(j,jo);
outtextxy(10,325,'donner le nom du joint'+jo);
joint[j]:=readkey;
setviewport(1,320,300,340,false);
clearviewport;
setviewport(0,0,getmaxx,getmaxy,false);
absc[j]:=xa;
ordo[j]:=ya;
outtextxy(absc[j]-10,ordo[j]-10,joint[j]);
end;
w:=0;
for extr:=1 to j do
begin
if ((absc[extr]=x) and (ordo[extr]=y)) then
begin
w:=1;
g:=extr;
end;
end;
if w=0 then
begin

```

```

    j:=j+1;
    str(j,jo);
    outtextxy(10,325,'donner le nom du joint'+jo);
    joint[j]:=readkey;
    setviewport(1,320,300,340,false);
    clearviewport;
    setviewport(0,0,getmaxx,getmaxy,false);
    absc[j]:=x;
    ordo[j]:=y;
    outtextxy(absc[j]-10,ordo[j]-10,joint[j]);
end;
end;

```

Function existe(nomdessin:string):boolean;

```

var
    fich:text;
begin
    assign(fich,'B:'+nomdessin);
    {$I-}
    reset(fich);
    {$I+}
    if ioresult=0 then
        begin
            existe:=true;
            close(fich);
        end
    else existe:=false;
end;

```

procedure photo;

```

begin

```

```

tailecran:=imagesize(0,0,getmaxx,getmaxy);
getmom(imecran,tailecran);
getimage(0,0,getmaxx,getmaxy,imecran^);
closegraph;
restorecrtmode;
end;

procedure cliche;
begin
detectgraph(driver,grmode);
initgraph(driver,grmode,'');
putimage(0,0,imecran^,normalput);
end;

procedure nommer;
begin
clrscr;
gotoXY(20,12); write('donner un nom au dessin ');
lire(nomdessin);
if existe(nomdessin)=true then
begin
gotoXY(20,14); writeln('un autre dessin porte ce nom');
gotoXY(20,16); writeln('conserver Y/N');
ecrase:=upcase(readkey);
if ecrase='Y' then nommer;
end;
end;

Procedure appeler;
begin
clrscr;
gotoXY(20,12); write('donner le nom du dessin : ');
lire(nomdessin);

```



```

if existe(nomdessin)=false then
begin
  clrscr;
  gotoXY(20,14);writeln('ce dessin ne se trouve pas dans B');
  delay(1000);
  appeler;
end;
end;

Procedure sauver(nomdessin:string);
var
  donnees:dessin;
begin
  assign(nom,'b:'+nomdessin);
  rewrite(nom);
  with donnees do
  begin
    sj:=j;
    ss:=s;
    sx:=x;
    sy:=y;
    for i:=1 to j do
      begin
        sabsc[i]:=absc[i];
        sordo[i]:=ordo[i];
        sjoint[i]:=joint[i];
        stypejoint[i]:=typejoint[i];
        sprisme[i]:=prisme[i];
      end;
    for i:=1 to s do
      begin
        sseg[i]:=seg[i];
        sangle[i]:=angle[i];
      end;
    end;
  end;
end;

```

```

    slong[i]:=long[i];
end;
write(nom,donnees);
end;
close(nom);
end;

```

Procedure charger(nomdessin:string);

```

var
  donnees:dessin;
begin
  assign(nom,'b:'+nomdessin);
  reset(nom);
  read(nom,donnees);
  with donnees do
  begin
    j:=sj;
    s:=ss;
    x:=sx;
    y:=sy;
    for i:=1 to j do
    begin
      absc[i]:=sabsc[i];
      ordo[i]:=sordo[i];
      joint[i]:=sjoint[i];
      typejoint[i]:=stypejoint[i];
      prisme[i]:=sprisme[i];
    end;
    for i:=1 to s do
    begin
      seg[i]:=sseg[i];
      angle[i]:=sangle[i];
      long[i]:=slong[i];
    end;
  end;
end;

```

```

    end;
  end;
  close(nom);
end;

```

Procédure coulisseau(xjoint,yjoint:longint;p:real);

(elle permet de placer un coulisseau d'axe incliné de p degré trigo au point (xjoint,yjoint))

var

```

  coord:arccoordstype;
  co:word;
  a1,b1,a2,b2,a3,b3,a4,b4,temp:longint;

```

begin

```

  co:=getcolor;
  temp:=round(p);
  setcolor(getbkcolor);
  arc(xjoint,yjoint,temp+30,temp+150,10);
  getarccoords( coord);
  a1:=coord.xstart;b1:=coord.ystart;
  a2:=coord.xend;b2:=coord.yend;
  arc(xjoint,yjoint,temp+210,temp+330,10);
  getarccoords(coord);
  a3:=coord.xstart;b3:=coord.ystart;
  a4:=coord.xend;b4:=coord.yend;
  setcolor(co);
  setlinestyle(0,0,2);
  line(a1,b1,a2,b2);
  line(a2,b2,a3,b3);
  line(a3,b3,a4,b4);
  line(a4,b4,a1,b1);
  setcolor(getbkcolor);
  arc(xjoint,yjoint,temp,temp+180,10);
  getarccoords(coord);
  setlinestyle(1,0,2);

```

```

setcolor(co);
line(coord.xstart,coord.ystart,coord.xend,coord.yend);
setlinestyle(0,0,1);
circle(xjoint,yjoint,3);
end;

```

```

Procedure prismatique2;

```

```

begin
setlinestyle(0,0,3);
line(round(absc[i]-20*cos(prisme[i]*pi/180)),round(ordo[i]+20*sin(prisme[i]*pi/180)),
      round(absc[i]+20*cos(prisme[i]*pi/180)),round(ordo[i]-20*sin(prisme[i]*pi/180)));
setlinestyle(0,0,1);
end;

```

```

Procedure restituer (nomdessin:string);

```

```

begin
for i:=1 to s do
begin
for h:=1 to j do
begin
if joint[h]=copy(seg[i],1,1) then orig:=h;
if joint[h]=copy(seg[i],2,1) then extr:=h;
end;
line(absc[orig],ordo[orig],absc[extr],ordo[extr]);
end;
for i:=1 to j do
begin

```

```

outtextxy(absc[i]-10,ordo[i]-10, joint[i]);

if typejoint[i]='R' then circle(absc[i],ordo[i],3);
if typejoint[i]='P' then
begin
coulisseau(absc[i],ordo[i],prisme[i]);
end;
end;
end;

Procédure prismatic(x,y:longint);
:
begin
clrscr;
gotoxy(20,12); write('donner son orientation en degrés ');
read(prisme[i]);
:
cliche;
:
coulisseau(x,y,prisme[i]);
:
end;

Procédure placerjoint(xjoint,yjoint:longint);
:
var
ty:integer;
begin
:
photo;
gotoxy(18,10);writeln(' nature du joint en '+joint[i]+' ? ');
gotoxy(25,14);write('1 : joint cheville');
gotoxy(25,16);write('2 : coulisseau');
gotoxy(25,18);write('0 : pas de joint');
gotoxy(45,22);write(' VOTRE CHOIX');
gotoxy(65,22);read(ty);
:
if ty=1 then typejoint[i]='R'
else if ty=2 then typejoint[i]='P'
else typejoint[i]='N';

```

```

if typejoint[i]='R' then
begin
clique;
circle(xjoint,yjoint,3);
end;
if typejoint[i]='P' then
begin
prismatique(xjoint,yjoint);
end;
if typejoint[i]='M' then clique;
end;

```

Procedure visualisation_M4M (x1,y1,x2,y2,x3,y3,x4,y4:longint);

(elle montre l'image instantannée d'un M4M en mouvement)

```

begin
gommer;
circle(absc[jointM],ordo[jointM],3);
line(absc[jointM],ordo[jointM],abscA[i],ordoA[i]);
circle(abscA[i],ordoA[i],3);
line(abscA[i],ordoA[i],abscB[i],ordoB[i]);
circle(abscB[i],ordoB[i],3);
line(abscB[i],ordoB[i],absc[jointQ],ordo[jointQ]);
circle(absc[jointQ],ordo[jointQ],3);
line(absc[jointQ],ordo[jointQ],absc[jointM],ordo[jointM]);
end;

```

Procedure visualisation_MC(x1,y1,x2,y2,x3,y3:longint);

(elle permet de visualiser un mécanisme manivelle coulisseau dans une certaine position)

```

var
  coord:arccoordstype;
  ci:integer;
begin
  gommer;
  ci:=getcolor;
  setcolor(getbkcolor);
  arc(abscl[joint8],ordo[joint8],round(prisme[joint8]),round(prisme[joint8])+180,200);
  getarccoords(coord);
  setlinestyle(1,0,2);
  setcolor(ci);
  line(coord.xstart,coord.ystart,coord.xend,coord.yend);
  setlinestyle(0,0,1);
  circle(x1,y1,3);
  line(x1,y1,x2,y2);
  circle(x2,y2,3);
  line(x2,y2,x3,y3);
  coulisseau(x3,y3,prisme[joint8]);
end;

```

Function tg(p:real):real;

(elle calcule la tangente de l'angle p exprimé en degré)

```

begin
  if ((p<>90) or(p<>270)) then
    begin
      tg:=(sin(p*pi/180))/(cos(p*pi/180));
    end;
end;

```

Procédure interarcdroite(x1,y1,x2,y2:longint;r,p:real);

(elle détermine l'intersection (xi,yi) d'un arc de centre (x1,y1) et de rayon r

et de la droite passant par le point (x_2, y_2) et de pente p degré trigo x_i ayant sa plus petite valeur)

```
var
a, a1, b1, c1: real;
begin
  if ((p=90) or (p=270)) then
  begin
    if abs(x2-x1)*1.0=r then
    begin
      xi:=x2;
      yi:=y1;
      delta:=0; (existence de l'intersection)
    end;
  end
  else
  begin
    a:=tg(-p);
    a1:=a*a+1;
    b1:=a*(y2-y1-x2*a)-x1;
    c1:=(y2-a*x2-y1)*(y2-a*x2-y1)+x1*x1-r*r;
    delta:=b1*b1-a1*c1;
    if delta>=0 then
    begin
      xi:=round((-b1+sqrt(delta))/a1);
      yi:=round(a*(xi-x2)+y2);
    end;
  end;
end;
```

Procédure veriMC12(nomdessin:string);

(elle vérifie si le dessin appelé est un manivelle coulisseau inversé)


```

var
pr:integer;
veri:boolean;
entree,fixe:string;
begin
if (((s<>4)or(j<>4))and((s<>3)or (j<>3))) then veri:=false;
if ((s=3)and(j=3))then
begin
pr:=0;
for i:=1 to 3 do if typejoint[i]='P' then
begin
v:=i;
pr:=pr+1;
end;
if (pr<>1) then veri:=false
else
begin
clrscr;
gotoXY(10,8);write('les membrures sont : ');
for i:=1 to 3 do write(seg[i],' ');
gotoXY(20,14);write('donner la membrure d''entrée: ');
gotoXY(20,16);write('donner la membrure fixe: ');
gotoXY(55,14);lire(entree);
gotoXY(55,16);lire(fixe);
for i:=1 to 3 do
begin
if ((seg[i]=entree)or(seg[i]=copy(entree,2,1)+copy(entree,1,1))) then g:=i;
if ((seg[i]=fixe) or (seg[i]=copy(fixe,2,1)+copy(fixe,1,1))) then f:=i;
end;
for i:=1 to 3 do
begin
if joint[i]=copy(seg[g],1,1) then jointM:=i;
if joint[i]=copy(seg[g],2,1) then jointA:=i;

```

```

    if joint[i]=copy(seg[f],1,1) then jointQ:=i;
end;
joint[4]:=joint[jointQ];
typejoint[4]:='R';long[4]:=0;seg[4]:=joint[4]+joint[4];jointB:=4;
for i:=1 to 4 do
begin
    if (seg[i]=joint[jointA]+joint[jointB]) then k:=i;
    if (seg[i]=joint[jointB]+joint[jointQ]) then h:=i;
end;
angle[h]:=angle[k]-90; veri:=true ;
end;
end;
if ((s=4)and(j=4)) then
begin
pr:=0;
for i:=1 to 4 do if typejoint[i]='P' then
begin
v:=i;
pr:=pr+1;
end;
if (pr<>1) then veri:=false
else
begin
clrscr;
gotoXY(10,8);write('les membrures sont : ');
for i:=1 to 4 do write(seg[i],' ');
gotoXY(20,14);write('donner la membrure d'entrée: ');
gotoXY(20,16);write('donner la membrure fixe: ');
gotoXY(55,14);lire(entree);
gotoXY(55,16);lire(fixe);
for i:=1 to 4 do
begin
if ((seg[i]=entree)or(seg[i]=copy(entree,2,1)+copy(entree,1,1))) then g:=i;

```

```

        if ((seg[i]=fixe) or (seg[i]=copy(fixe,2,1)+copy(fixe,1,1))) then f:=i;
    end;
for i:=1 to 4 do
    begin
        if joint[i]=copy(seg[g],1,1) then jointM:=i;
        if joint[i]=copy(seg[g],2,1) then jointA:=i;
        if joint[i]=copy(seg[f],1,1) then jointQ:=i;
    end;
for i:=1 to 4 do
    begin
        if ((i<>jointM) and (i<>jointA) and (i<>jointQ)) then jointB:=i;
    end;
for i:=1 to 4 do
    begin
        if (seg[i]=joint[jointA]+joint[jointB]) then k:=i;
        if (seg[i]=joint[jointB]+joint[jointQ]) then h:=i;
    end;
    if ((abs(angle[k]-angle[h])=90) or (abs(angle[k]-angle[h])=270)) then veri:=true
    else veri:=false;
end;
end;
ok:=veri;
end;

```

```

Function veri2_MCI(nomdessin:string):boolean;
{elle précise si le schéma nomdessin est un MCI}
begin
    veriMCI2(nomdessin);
    veri2_MCI:=ok;
end;

```

```
Procédure visualisationMCI(x1,y1,x2,y2,x3,y3,x4,y4:longint);
```

```
{elle montre le mécanisme dans une position donnée}
```

```
begin
```

```
  donner;
```

```
  coulisseau(xjoint,yjoint,prisme[jointB]);
```

```
  circle(x2,y2,3);
```

```
  line(x1,y1,x2,y2);
```

```
  circle(x1,y1,3);
```

```
  line(x2,y2,x3,y3);
```

```
  line(x3,y3,x4,y4);
```

```
  line(x4,y4,x1,y1);
```

```
  circle(x4,y4,3);
```

```
end;
```

```
Procédure complexe(a,b:longint);
```

```
{elle permet de calculer le module et l'argument d'un nombre complexe}
```

```
begin
```

```
  module:=sqrt(a*a+b*b);
```

```
  trigo(0,0,a,b);
```

```
  argument:=zeta;
```

```
end;
```

```
Function dist(x1,y1,x2,y2:longint):real;
```

```
{elle permet de calculer la distance entre les points (x1,y1) et (x2,y2)}
```

```
begin
```

```
  dist:=sqrt((x2-x1)*(x2-x1)+(y2-y1)*(y2-y1));
```

```
end;
```

```
Procédure interdroites (x1,y1,x2,y2:longint;p1,p2:real);
```

(elle permet de déterminer l'intersection (xi,yi) des droites d'angle p1 et p2 qui passent par les points 1 et 2)

```
var
```

```
  a1,a2,b1,b2:real;
```

```
begin
```

```
  if ((abs(p1-p2) <> 0) and (abs(p1-p2) <> 180)) then
```

```
    begin
```

```
      if ((p1 >= 0) and (p1 < 90)) then a1 := tg(-p1);
```

```
      if ((p1 > 90) and (p1 < 270)) then a1 := tg(180-p1);
```

```
      if ((p1 > 270) and (p1 < 360)) then a1 := tg(360-p1);
```

```
      if ((p2 >= 0) and (p2 < 90)) then a2 := tg(-p2);
```

```
      if ((p2 > 90) and (p2 < 270)) then a2 := tg(180-p2);
```

```
      if ((p2 > 270) and (p2 < 360)) then a2 := tg(360-p2);
```

```
    if ((p1=90) or (p1=270)) or ((p2=90) or (p2=270)) then
```

```
      begin
```

```
        if ((p1=90) or (p1=270)) then
```

```
          begin
```

```
            xi := x1;
```

```
            yi := round(a2*(xi-x2)+y2);
```

```
          end
```

```
        else b1 := y1 - a1*x1;
```

```
      if ((p2=90) or (p2=270)) then
```

```
        begin
```

```
          xi := x2;
```

```
          yi := round(a1*(xi-x1)+y1);
```

```
        end
```

```
      else b2 := y2 - a2*x2;
```

```
    end
```

```
  else
```

```
    begin
```

```
      b1 := y1 - a1*x1;
```

```
      b2 := y2 - a2*x2;
```

```
xi:=round((b2-b1)/(a1-a2));  
yi:=round(a1*xi+b1);  
end;  
end;  
end;
```

Begin

End.

MODULE 3: REALISATION DU SCHEMA (schema)

```
Uses crt,dos,graph,unit0,progene,printer;
```

```
Procedure schema;
```

```
var  
    fini:boolean;
```

```
PROCEDURE BrowseFile(InFile:string);
```

```
Var  
    line :integer;
```

```
CONST  
    MaxPage = 2 ;
```

```
TYPE  
    BufferType = array [1..MaxPage,1..24] of String;
```

```
VAR  
    InChar,ch           :Char;
```

```
J, I,K,nl,temp,  
    Enreg              :Byte;
```

```
    CurrentPage,TempPage,
```

```

FirstPage,LastPageInBuffer,Status,
LastPageInFile,blinking,bold,LineNum      :integer;

TempStr,TopLine,
TempLine                                    :String;

InText                                     :Text;
Buffer                                     : BufferType;
BufferEmpty                               : Boolean;
Ens                                         : Set of Char;

```

```

PROCEDURE SetVideo(Attribute:integer);

```

```

Begin (procedure SetVideo)

```

```

    blinking:=(attribute and 4) * 4;

```

```

    if (attribute and 1) = 1 Then

```

```

        Begin

```

```

            bold:=(Attribute and 2) * 7;

```

```

        TextColor (1 + blinking + bold);

```

```

            TextBackground (3);

```

```

        End

```

```

    Else

```

```

        Begin

```

```

            bold:= (attribute and 2) * 5 div 2;

```

```

            TextColor (7 + blinking + bold);

```

```

            TextBackground (0);

```

```

        End;

```

```

End; (procedure SetVideo)

```



```
PROCEDURE PutString(OutString:string;  
                   Line,Col,Attrib:Integer);
```

```
Begin  
    SetVideo(Attrib);  
    Gotoxy(col,line);  
    Write(OutString);  
    SetVideo(0);  
End;
```

```
FUNCTION TextFileSize (TextFile:String):integer;
```

```
Begin (procedure TextFileSize)  
    {$I-} Assign (InText,TextFile);  
    Reset(InText);  
    If IOResult>0 then TextFileSize:=0  
    Else  
        Begin (Else)  
            LineNum:=0;  
            While not eof (InText) Do  
                Begin  
                    readln (InText,TempLine);  
                    LineNum:= LineNum +1;  
                End;  
            TextFileSize:= LineNum;  
        End; (Else)  
    Close (InText); {$I+}  
End; (procedure TextFileSize)
```

Begin (procedure Browsefile)

```
($I-) Assign(InText,InFile);
SetTextBuf(InText,Buffer);

Reset(InText);

If IOResult <> 0 Then
  Begin
    ClrScr; Gotoxy(10,10);

    Write ('INSEREZ LA DISQUETTE PROGRAMME');

    Gotoxy(10,12);write('PUIS APPUYER SUR UNE TOUCHE ');

    Ch:=ReadKey; Reset(InText);

  End;

LastPageInFile:=TextFileSize(InFile) div 24 + 1;

CurrentPage:=1;

BufferEmpty:= true;

InChar:='j';

While InChar <> chr(27) Do
  Begin (While InChar)
    if bufferEmpty Then
      Begin (if BufferEmpty)
        BufferEmpty:=false;

        FirstPage:=CurrentPage - MaxPage div 2;

        if FirstPage < 1 then FirstPage:=1;

        TempPage :=1;

        reset(InText);

        While(TempPage < FirstPage) Do
          Begin
            For Line :=1 to 24 do
              readln(InText,TempStr);

              TempPage:=TempPage+1;

            End;

            Line :=1; TempPage:=1;

            While (FirstPage + TempPage-1 <= LastPageInFile)
```

```

And (TempPage <= MaxPage) do
  Begin
    readln(inText,Buffer[TempPage,Line]);
    Line := Line + 1;
    if Line > 24 then
      Begin
        TempPage:= TempPage + 1;
        Line := 1;
      End;
    End;(while and)

    LastPageInBuffer:=FirstPage + TempPage - 2;
  End;(if BufferEmpty)
  ClrScr;
  Str (CurrentPage,TopLine);
  Str(LastPageInFile,TempStr);
  Line := 1 ;
  While (Line<=24) And (Not KeyPressed) Do
    Begin
      PutString (Buffer[CurrentPage - FirstPage +1,Line],
                Line+1,1,0);
      Line:= Line + 1;
    End;
  InChar:=readkey;
  if (InChar = chr(81)) or (InChar = Chr(73)) then
    Begin
      if (InChar = chr(81)) And
        (CurrentPage < LastPageInFile) then
        Begin
          CurrentPage:= Currentpage + 1;
          If CurrentPage > LastPageInBuffer Then
            BufferEmpty :=true;
        End
    End

```

```

        Else if (InChar = Chr(73)) and
            (CurrentPage > 1) then
            Begin
                CurrentPage := CurrentPage -1;
                if CurrentPage < FirstPage then
                    BufferEmpty:= true;
                End;
                InChar:='j';
            End (if chr(27))
        Else
            InChar := UpCase (InChar);
        End; (While InChar)
    Close(InText); ($!+)
End; (procedure Browsefile)

```

BEGIN

```

detectgraph(driver,grmode);
initgraph(driver,grmode,'');
x:=305; y:=102; pas:=1; xa:=x; ya:=y; j:=0; s:=0; fini:=false;
rectangle(100,10,700,310); outtextxy(1,30,' F1: ');
outtextxy(1,45,' Aide');
outtextxy(1,65,' Entr: ');
outtextxy(1,80,' Droite');
outtextxy(1,100,' Home: ');
outtextxy(1,115,' Origine');
outtextxy(1,135,' Fleches:');
outtextxy(1,150,' Deplacement');
outtextxy(1,170,' Pgdn:');
outtextxy(1,185,' Desincrement');
outtextxy(1,205,' Pgup:');
outtextxy(1,220,' Increment');

```

```

outtextxy(1,240,' F4:');
outtextxy(1,255,' Reprise');
outtextxy(1,275,' F10: Fin!');
setlinestyle(0,0,3);
Rectangle(1,10,700,310);
setlinestyle(0,0,1);
rectangle(300,100,310,104);
  tail:=imagesize(300,100,310,105);
getmem(curs_im,tail);
getmem(ppim,tail);
getmem(memcurs_im,tail);
getimage((x-5),(y-2),(x+5),(y+2),curs_im^);
getimage((x-5),(y-2),(x+5),(y+2),memcurs_im^);
setviewport(300,100,310,104,false);
clearviewport;
setviewport(0,0,getmaxx,getmaxy,false);
ce:=getcolor;
REPEAT
  p:=Getpixel(x,y);
  getimage((x-5),(y-2),(x+5),(y+2),ppim^);
  Repeat
    putpixel(x,y,0);putimage((x-5),(y-2),curs_im^,normalput);
    putpixel(x,y,ce);delay(2);
  until keypressed;
  putpixel(x,y,p);
  putimage((x-5),(y-2),ppim^,normalput);
  car:=uppercase(readkey);
  if car =#0 then
  begin
    car:=readkey;
    case car of
      #59:begin
        closegraph;

```

```

        restorecrtmode;

        clrscr;

        browsefile('aschema.pas');

        clrscr;

        exit;

    end;

#75: if x>(100+pas) then x:=x-pas;
#77: if x<(695-pas) then x:=x+pas;
#80: if y<(305-pas) then y:=y+pas;
#72: if y>(12+pas) then y:=y-pas;
#73: if pas<70 then pas:=pas+10;
#81: if pas>1 then pas:=pas-10;
#68: fini:=true;
#62:begin
    s:=0;
    j:=0;
    setviewport(105,12,695,305,false);
    clearviewport;
    setviewport(0,0,getmaxx,getmaxy,false);
    putpixel(x,y,ce);
end;

#71:begin
    putpixel(x,y,ce); xa:=x; ya:=y;
end;
end;
end
else
begin
    case
    car of
    #13:begin

```

```

if j=0 then
begin
  absc[1]:=xa;
  ordo[1]:=ya;
  absc[2]:=x;
  ordo[2]:=y;
  long[1]:=rr;
  angle[1]:=rp;
  str(j+1,jo);
  outtextxy(10,325,'donner le nom du joint'+jo);
  joint[1]:=readkey;
  setviewport(1,320,300,340,false);
  clearviewport;
  setviewport(0,0,getmaxx,getmaxy,false);
  outtextxy(absc[1]-10,ordo[1]-10,joint[1]);
  str(j+2,jo);
  outtextxy(10,325,'donner le nom du joint'+jo);
  joint[2]:=readkey ;
  seg[1]:=joint[1]+joint[2];
  s:=1;
  setviewport(1,320,300,340,false);
  clearviewport;
  setviewport(0,0,getmaxx,getmaxy,false);
  outtextxy(absc[2]-10,ordo[2]-10,joint[2]);
  j:=2;
end
else
begin
  point (xa,ya,x,y);
  segment (xa,ya,x,y);
end;
setcolor(ce);line(xa,ya,x,y); xa:=x; ya:=y;
end;

```

```
        end;
    end;
    coordonne(x,y);
UNTIL fini;
for i:=1 to j do
    begin
        xjoint:=absc[i];
        yjoint:=ordo[i];
        placerjoint(xjoint,yjoint);
    end;
photo;
nommer;
sauver(nomdessin);

closegraph;
```

End;

```
Begin
    schema;
end.
```


MODULE 4: VISUALISATION DU SCHEMA (Visualis)

```
Uses crt,graph,unit0,grimpr,progene,printer;
```

```
var
```

```
  impres:char;
```

```
  code_retour:byte;
```

```
begin
```

```
  appeler;
```

```
  charger(nomdessin);
```

```
  clrscr;
```

```
  gotoxy(12,12);writeln('Dsirez-vous imprimer ? O (oui) ou N (non) ');
```

```
  gotoxy(12,60);impres:=upcase(readkey);
```

```
  cadre;
```

```
  restituer(nomdessin);
```

```
  outtextxy(5,5 , 'segment long. angle');
```

```
  for i:=1 to s do
```

```
    begin
```

```
      str(long[i]:4:1,xst);str(angle[i]:4:1,yst);
```

```
      outtextxy(5,10+10*i,' '+seg[i]+' '+xst+' '+ yst);
```

```
    end;
```

```
  if impres='O' then imprime_ecran(code_retour);
```

```
  delay(40000);
```

```
  closegraph;
```

End.

MODULE 5: ETUDE DE LA MOBILITE DU SCHEMA (Mobilite)

Program Mobilite;

{Il determine le degr de libert d'une chaine de membrures de longueur finie}

uses crt,progene,unit0;

var

m:array[1..20] of integer;

DL,n:integer;

ccc:char;

begin

appeler;

charger(nomdessin);

n:=0;

for i:=1 to j do

begin

m[i]:=0;

for h:=1 to s do

begin

if((joint[i]=copy(seg[h],1,1))or(joint[i]=copy(seg[h],2,1))) then m[i]:=m[i]+1;

end;

m[i]:=m[i]-1;

n:=n+m[i];

end;

DL:=3*(s-1)-2*n;

clrscr;

gotoXY(20,10);write('LE DEGRE DE LIBERTE EST: ',DL);

if DL>=1 then

begin

gotoXY(20,15);write('VOUS AVEZ UN MECANISME');

end

```
else
  begin
    gotoXY(20,15);write('VOUS AVEZ UNE STRUCTURE');
  end;
gotoxy(40,23);write('appuyer une touche pour continuer');
ccc:=readkey;
end.
```

MODULE 6: ETUDE DE LA TRANSFORMATION DU MOUVEMENT D'UN M4M (typeM4M)

```
Program typeM4M;
  {elle determine la nature d'un mecanisme quatre membrures }
  uses crt,unit0,progene;

var
  inv:boolean;
  temp:real;
  entree,fixe,memb:string;
  m:array[1..20] of integer;
  DL,n,pr:integer;
  ccc:char;
begin
  resultat:=' '; pr:=0;
  appeler;
  charger(nomdessin);
  n:=0;
  if ((s=2)and(j=3)) then
  begin
    pr:=0;
    for i:=1 to 3 do
    begin
      if typejoint[i]='P' then pr:=pr+1;
    end;
    if (pr=1) then resultat:=' est un manivelle-coulisseau ';
  end;
  if (((j=3)and(s=3))or((j=4)and(s=4))) then
  if veri2_mci(nomdessin) then
  resultat:=' est un manivelle-coulisseau-invers ';
  for i:=1 to j do
  begin
    if typejoint[i]='P' then pr:=pr+1;
  end;
```

```

if pr=0 then
begin
for i:=1 to j do
begin
m[i]:=0;
for h:=1 to s do
begin
if((joint[i]=copy(seg[h],1,1))or(joint[i]=copy(seg[h],2,1))) then m[i]:=m[i]+1;
end;
m[i]:=m[i]-1;
n:=n+m[i];
end;
DL:=3*(s-1)-2*n;
if DL <0 then resultat:='est une structure'
else
begin
repeat
inv:=false;
for i:=1 to 3 do
begin
if long[i+1]<long[i] then
begin
temp:=long[i];
long[i]:=long[i+1];
long[i+1]:=temp;
memb:=seg[i];
seg[i]:=seg[i+1];
seg[i+1]:=memb;
inv:=true;
end;
end;
until(inv=false);
if long[1]+long[4]>=long[2]+long[3] then resultat:='est un balancier-balancier'

```

```

else
begin
  clrscr;
  gotoXY( 11,4);write('Les membrures sont :');
  for i:=1 to 4 do
  begin
    gotoXY(5+6*i,6);write(seg[i],',');
  end;
  gotoXY(20,12);write('donner la membrure d'entree: ');
  gotoXY(20,14);write('donner la membrure fixe: ');
  gotoxy(60,12);lire(entree);
  gotoxy(60,14); lire(fixe);
  clrscr;
  for i:=1 to 4 do
  begin
    if seg[i]=entree then h:=i;
    if seg[i]=fixe then g:=i
  end;
  if long[h]=long[1] then resultat:='est un manivelle-balancier'
else
  begin
    if long[g]=long[1] then resultat:='est un manivelle-manivelle'
    else resultat:='est un balancier-balancier';
  end;
end;
end;
end;
clrscr;
gotoXY(20,12);writeln('ce dessin:', resultat);
gotoxy(40,23);write('appuyer une touche pour continuer!');
ccc:=readkey;
end.^Z

```

MODULE 7: ETUDE DES POSITIONS PARTICULIERES (critique)

Program critique;

(il donne les positions limites et points morts des M4M, MC, MC1)

uses crt,unit0,progene,printer,graph;

Procedure PLPMM4M;

(elle détermine les positions limites et points morts d'un M4M)

var

entree,sortie,inter,fixe:string;

a,b,c,d,theta21,theta22,y1,y2:real;

begin

appeler;

charger(nomdessin);

if ((s<>4) or (j<>4)) then

begin

clrscr;

gotoxy(20,12);writeln('ce dessin n''a pas 4 membrures');

delay(10000);

end

else

begin

clrscr;

gotoxy(11,4);write('les membrures sont :');

for i:=1 to 4 do

begin

gotoxy(5+6*i,6);write(seg[i],';');

end;

gotoxy(20,10);write('membrane d'entrée ? ');

gotoxy(20,12);write('membrane intermediaire ? ');

```

gotoXY(20,14);write('membrure sortie ? ');
gotoXY(20,16);write('membrure fixe ? ');
gotoXY(45,10);lire(entree);
gotoXY(45,12);lire(inter);
gotoXY(45,14);lire(sortie);
gotoXY(45,16);lire(fixe);
for i:=1 to 4 do
begin
  if seg[i]=entree then g:=i;
  if seg[i]=inter then k:=i;
  if seg[i]=sortie then h:=i;
  if seg[i]=fixe then f:=i;
end;
a:=long[g];b:=long[k];c:=long[h];d:=long[f]; theta21:=600;theta22:=600;
y1:=(2*(a+b)*d)*(2*(a+b)*d);y2:=((a+b)*(a+b)+d*d-c*c)*((a+b)*(a+b)+d*d-c*c);
if y1>=y2 then
begin
  if y2=0 then theta21:=90
  else theta21:=180/pi*sqrt(Arctan(y1/y2-1));
  if theta21<>600 then theta21:=-angle[f]+180+theta21;
end;
y1:=(2*(b-a)*d)*(2*(b-a)*d);y2:=((b-a)*(b-a)+d*d-c*c)*((b-a)*(b-a)+d*d-c*c);
if y1>=y2 then
begin
  if y2=0 then theta22:=90
  else theta22:=180+180/pi*sqrt(Arctan(y1/y2-1));
  if theta22<>600 then theta22:=-angle[f]+180+theta22;
end;
clrscr;
if ((theta21<>600) or (theta22<>600)) then

begin

```



```

gotoxy(10,10);writeln(LST, ' _____ ');
  if theta21<>600 then
    begin
gotoxy(10,11);writeln(LST, ' |                A la position limite 1                | ');
gotoxy(10,12);writeln(LST, ' | la membrure motrice a une inclinaison de ',theta21:4:1,' degrés
| ');

    end;
  if theta22<> 600 then
    begin
gotoxy(10,13);writeln(LST, ' |                | ');
gotoxy(10,14);writeln(LST, ' |                A la position limite 2                | ');
gotoxy(10,15);writeln(LST, ' | la membrure motrice a une inclinaison de ',theta22:4:1,' degrés
| ');

    end;
gotoxy(10,16);writeln(LST, ' |                | ');
gotoxy(10,17);writeln(LST, ' _____ ');

  end;
theta21:=600; theta22:=600;
y1:=(2*a*d)*(2*a*d);y2:=(-(b+c)*(b+c)+d*d+a*a)*(-(b+c)*(b+c)+d*d+a*a);
  if y1>=y2 then
    begin
      if y2=0 then theta21:=90
        else theta21:=180/pi*sqrt(Arctan(y1/y2-1));
      if theta21<>600 then theta21:=-angle[f]+180+theta21;
    end;
y1:=(2*a*d)*(2*a*d);y2:=(-(b-c)*(b-c)+d*d+a*a)*(-(b-c)*(b-c)+d*d+a*a);
  if y1>=y2 then
    begin
      if y2=0 then theta22:=90
        else theta22:=180/pi*sqrt(Arctan(y1/y2-1));
      if theta22<>600 then theta22:=-angle[f]+180+theta22;
    end;
clrscr;

```

```

if ((theta21<>600) or (theta22<>600)) then
begin
gotoxy(10,10);writeln(LST, ' _____ ');
if theta21<>600 then
begin
gotoxy(10,11);writeln(LST, ' | Au point mort 1 | ');
gotoxy(10,12);writeln(LST, ' | la membrure motrice a une inclinaison de ',theta21:4:1,' degrés
| ');
gotoxy(10,13);writeln(LST, ' | | ');
end;
if theta22<>600 then
begin
gotoxy(10,14);writeln(LST, ' | Au point mort 2 | ');
gotoxy(10,15);writeln(LST, ' | la membrure motrice a une inclinaison de ',theta22:4:1,' degrés
| ');
gotoxy(10,16);writeln(LST, ' | | ');
end;
gotoxy(10,17);writeln(LST, ' _____ ');
end;
for i:=1 to 4 do
begin
if joint[i]=copy(seg[k],1,1) then jointA:=i;
if joint[i]=copy(seg[k],2,1) then jointB:=i;
if joint[i]=copy(seg[f],1,1) then jointQ:=i;
if joint[i]=copy(seg[f],2,1) then jointM:=i;
end;
end;
end;
end;

```

Procédure PLPMMC;

(elle détermine les positions limites et points morts d'un manivelle coulisseau)

var

xn,yn:longint;

a0:string[1];

temp:integer;

y1,y2,a,b,theta21,theta22,an,e:real;

begin

temp:=0;

for i:=1 to 3 do

begin

if typejoint[i]='P' then temp:=temp+1;

end;

if (temp<>1) then

begin

clrscr;

gotoXY(20,12);write('vous n'avez pas un manivelle coulisseau');

end

else

begin

for i:=1 to 3 do if typejoint[i]='P' then jointB:=i;

for i:=1 to 2 do

begin

if copy(seg[i],1,1)=joint[jointB] then

begin

a0:=copy(seg[i],2,1);

k:=i;

end;

if copy(seg[i],2,1)=joint[jointB] then

begin

a0:=copy(seg[i],1,1);

```

    k:=i;
  end;
end;
for i:=1 to 3 do if joint[i]=a0 then jointA:=i;
for i:=1 to 3 do if ((i<>jointA) and (i<>jointB)) then jointM:=i;
if k=1 then g:=2;
if k=2 then g:=1;
xn:=absc[jointM];
an:=(1.0*xn-1.0*absc[jointB])*tg(prisme[jointB]);
yn:=ordo[jointB]+trunc(an);
e:=trunc(round((yn-ordo[jointM])*sin(90-prisme[jointB])));
a:=long[g];b:=long[k];
theta21:=600;theta22:=600;
y1:=(e/(a+b))*(e/(a+b));
  if y1=1 then theta21:=90
  else if y1<1 then theta21:=180/pi*sqrt(Arctan(y1/(1-y1)));
if theta21<>600 then theta21:=theta21-prisme[jointB];
if b<>a then
begin
  y2:=(e/(b-a))*(e/(b-a));
  if y2=1 then theta22:=90
  else if y2<1 then theta22:=180/pi*sqrt(Arctan(y2/(1-y2)));
  if theta22<>600 then theta22:=theta22-prisme[jointB];
end;
clrscr;
if ((theta21<>600) or (theta22<>600)) then
begin
gotoxy(10,10);writeln(LST, ' _____ ');
  if theta21<>600 then
begin
gotoxy(10,11);writeln(LST, ' | A la position limite 1 | ');
gotoxy(10,12);writeln(LST, ' | la membrure motrice a une inclinaison de ',theta21:4:1,' degrés
| ');

```

```

end;
if theta22<> 600 then
begin
gotoxy(10,13);writeln(LST, ' | | ');
gotoxy(10,14);writeln(LST, ' | A la position limite 2 | ');
gotoxy(10,15);writeln(LST, ' | la membrure motrice a une inclinaison de ',theta22:4:1,' degrés
| ');
end;
gotoxy(10,16);writeln(LST, ' | | ');
gotoxy(10,17);writeln(LST, ' |-----| ');
end;
theta21:=600; theta22:=600;
y1:=((b-e)/a)*((b-e)/a);
if y1=1 then theta21:=90
else if y1<1 then theta21:=180/pi*sqrt(Arctan(y1/(1-y1)));
if theta21<>600 then theta21:=theta21+prisme[joint8];
y2:=((b+e)/a)*((b+e)/a);
if y2=1 then theta22:=90
else if y2<1 then theta22:=180/pi*sqrt(Arctan(y2/(1-y2)));
if theta22<>600 then theta22:=360-theta22+prisme[joint8];
clrscr;
if ((theta21<>600) or (theta22<>600)) then
begin
gotoxy(10,10);writeln(LST, ' |-----| ');
if theta21<>600 then
begin
gotoxy(10,11);writeln(LST, ' | Au point mort 1 | ');
gotoxy(10,12);writeln(LST, ' | la membrure motrice a une inclinaison de ',theta21:4:1,' degrés
| ');
gotoxy(10,13);writeln(LST, ' | | ');
end;
if theta22<>600 then
begin

```

```

gotoxy(10,14);writeln(LST, ' |                               Au point mort 2                               | ');
gotoxy(10,15);writeln(LST, ' | la membrure motrice a une inclinaison de ',theta22:4:1,' degrés
| ');
gotoxy(10,16);writeln(LST, ' |                               | ');
    end;
gotoxy(10,17);writeln(LST, ' |-----| ');
    end;
    end;
end;

```

Procédure PLMCI;

(elle détermine la position limite d'un manivelle coulisseau inversé)

var

theta21,theta22,theta2,theta3,theta4:real;

longK:array[1..360] of real;

y1,y2,a,b,e,d,AQ2,k1,k2,k3:real;

begin

if veri2_MCI(nomdessin)=false then

begin

clrscr;

gotoxy(20,12);write('vous n'avez pas le bon mécanisme');

end

else

begin

a:=long[g];b:=long[k];e:=long[h];d:=long[f];

theta21:=600;theta22:=600;

y1:=((e-a)/d)*((e-a)/d);

if y1=1 then theta21:=90

else if y1<1 then theta21:=180/pi*sqrt(Arctan(y1/(1-y1)));

if e>=a then

begin

if theta21<>600 then theta21:=theta21+90

```

end

else if theta21<>600 then theta21:=90-theta21;
if theta21 <>600 then theta21:=theta21+angle[f]-180;
y2:=((e+a)/d)*((e+a)/d);
if y2=1 then theta22:=90
else if y2<1 then theta22:=180/pi*sqrt(Arctan(y2/(1-y2)));
if theta22<>600 then theta22:=theta22+270;
if theta22<>600 then theta22:=angle[f]-180+theta22;

clrscr;
if ((theta21<>600) or (theta22<>600)) then
begin
gotoxy(10,10);writeln(LST, ' _____ ');
if theta21<>600 then
begin
gotoxy(10,11);writeln(LST, ' | A la position limite 1 | ');
gotoxy(10,12);writeln(LST, ' | la membrure motrice a une inclinaison de ',theta21:4:1,' degrés
| ');
end;
if theta22<> 600 then
begin
gotoxy(10,13);writeln(LST, ' | | ');
gotoxy(10,14);writeln(LST, ' | A la position limite 2 | ');
gotoxy(10,15);writeln(LST, ' | la membrure motrice a une inclinaison de ',theta22:4:1,' degrés
| ');
end;
gotoxy(10,16);writeln(LST, ' | | ');
gotoxy(10,17);writeln(LST, ' _____ ');
end;
end;
end;
end;

```

Procedure menupos;

```

(elle donne le menu secondaire pour les positions limites et point morts)

var
  choix:integer;

begin
  clrscr;

  gotoXY(5,5); write('1:POSITIONS POUR MECANISME A QUATRE MEMBRURES SANS COULISSEAU');
  gotoXY(5,7); write('2:POSITIONS POUR MECANISME A QUATRE MEMBRURES AVEC COULISSEAU');
  gotoxy(5,9); write('0:pour sortir');
  gotoXY(50,12);write('VOTRE CHOIX ? ');read(choix);

  case choix of
    1:begin
      PLPMH4M;
      sortie;
      if sort=#27 then menupos
      else sortie;
    end;
    2:begin
      appeler;
      charger(nomdessin);
      if (( s=2)and(j=3)) then PLPMMC ;
      if(((s=4)and(j=4)) or ((s=3)and(j=3)))then PLMCI ;
      sortie;
      if sort=#27 then menupos
      else sortie;
    end;
  end;

end;

begin
  menupos;
end.

```

MODULE 8: ETUDE DE LA TRANSMISSION (transmis)


```

gotoXY(45,10);lire(entree);
gotoXY(45,12);lire(inter);
gotoXY(45,14);lire(sortie);
gotoXY(45,16);lire(fixe);
for i:=1 to 4 do
begin
  if seg[i]=entree then g:=i;
  if seg[i]=inter then k:=i;
  if seg[i]=sortie then h:=i;
  if seg[i]=fixe then f:=i;
end;
a:=long[g];b:=long[k];c:=long[h];d:=long[f];
for i:=1 to 4 do
begin
  if joint[i]=copy(seg[k],1,1) then jointA:=i;
  if joint[i]=copy(seg[k],2,1) then jointB:=i;
  if joint[i]=copy(seg[f],1,1) then jointQ:=i;
  if joint[i]=copy(seg[f],2,1) then jointM:=i;
end;
t:=0;clrscr;
gotoxy(10,10);writeln(LST, ' _____ ');
gotoxy(10,12);writeln(LST, ' | Angle de transmission  $\mu$  et efficacité eff | ');
gotoxy(10,14);writeln(LST, ' | position entrée  $\mu$  eff | ');
for i:=1 to 360 do
begin
  angleG[i]:=angle[g]+i; if angleG[i]>360 then angleG[i]:=angleG[i]-360;
  if angleG[i]>360 then angleG[i]:=angleG[i]-360;
  abscA[i]:=absc[jointM]+round(long[g]*cos(angleG[i]*pi/180));
  ordoA[i]:=ordo[jointM]-round(long[g]*sin(angleG[i]*pi/180));
  interarcs(abscA[i],ordoA[i],absc[jointQ],ordo[jointQ],long[k],long[h]);
  if delta>=0 then
  begin
    t:=t+1;angleG[t]:=angleG[i];abscA[t]:=abscA[i];ordoA[t]:=ordoA[i];

```

```

x1:=(-a*a-d*d+b*b+c*c+2*a*d*cos(pi/180*(angleG[t]+180-angle[f])))/(2*b*c);x1:=x1*x1;
if x1=0 then mu:=90
else if x1<1 then mu:=180/pi*sqrt(Arctan(1/x1-1));
clrscr;
abscB[t]:=xi; ordoB[t]:=yi;trigo(abscA[t],ordoA[t],abscB[t],ordoB[t]);
if (a*sin(pi/180*(-angleK[t]+angleG[t])) <>0 then
begin
angleK[t]:=zeta;eff:=c*sin(pi/180*mu)/(a*sin(pi/180*(-angleK[t]+angleG[t])));
gotoxy(10,16);
writeln(LST, ' |', angleG[t]:4:1, ' ', ' ', mu:4:1, ' ', ' ',
',eff:4:1, ' |');
trigo(abscB[t],ordoB[t],absc[jointQ],ordo[jointQ]);
angleH[t]:=zeta;
end;
end;
end;
gotoxy(10,17);writeln(LST, ' ──────────────────────────────────── ');end;
end;

```

Procedure ATMC;

{elle détermine l'angle de transmission d'un manivelle coulisseau}

var

```

xn,yn:longint;
a0:string[1];
temp:integer;
an,e:real;
a,b,x1,mu,theta2:real;

```

begin

```
temp:=0;
```

```
for i:=1 to 3 do
```

```
begin
```

```
if typejoint[i]='P' then temp:=temp+1;
```

```
end;
```

```

if (temp<>1) then
begin
  clrscr;
  gotoxy(20,12);write('vous n'avez pas un manivelle coulisseau');
end
else
begin
  for i:=1 to 3 do if typejoint[i]='P' then jointB:=i;
  for i:=1 to 2 do
  begin
    if copy(seg[i],1,1)=joint[jointB] then
    begin
      a0:=copy(seg[i],2,1);
      k:=i;
    end;
    if copy(seg[i],2,1)=joint[jointB] then
    begin
      a0:=copy(seg[i],1,1);
      k:=i;
    end;
  end;
  for i:=1 to 3 do if joint[i]=a0 then jointA:=i;
  for i:=1 to 3 do if ((i<>jointA) and (i<>jointB)) then jointM:=i;
  if k=1 then g:=2;
  if k=2 then g:=1;
  xn:=absc[jointM];a:=long[g];b:=angle[k];
  an:=(1.0*xn-1.0*absc[jointB])*tg(prisme[jointB]);
  yn:=ordo[jointB]+trunc(an);
  e:=trunc(round((yn-ordo[jointM])*sin(90-prisme[jointB])));
  t:=0;clrscr;
  gotoxy(10,10);writeln(LST, ' _____ ');
  gotoxy(10,12);writeln(LST, ' | Angle de transmission  $\mu$  MC | ');
  gotoxy(10,14);writeln(LST, ' | position entrée  $\mu$  | ');

```

Program transmis;

Uses crt,graph,unit0,progene,printer;

Procedure ATM4M;

(elle détermine l'angle de transmission d'un M4M)

var

entree,sortie,inter,fixe:string;

x1,a,b,c,d,mu,theta2,eff:real;

begin

appeler;

charger(nomdessin);

if ((s<>4) or (j<>4)) then

begin

clrscr;

gotoxy(20,12);writeln('ce dessin n''a pas 4 membrures');

delay(10000);

end

else

begin

clrscr;

gotoxy(11,4);write('les membrures sont :');

for i:=1 to 4 do

begin

gotoxy(5+6*i,6);write(seg[i],';');

end;

gotoxy(20,10);write('membrane d'entrée ? ');

gotoxy(20,12);write('membrane intermédiaire ? ');

gotoxy(20,14);write('membrane de sortie ? ');

gotoxy(20,16);write('membrane fixe ? ');

```

for i:=1 to 360 do
begin
angleG[i]:=angle[g]+i;if angleG[i]>360 then angleG[i]:=angleG[i]-360;
abscA[i]:=absc[jointM]+trunc(round(long[g]*cos(angleG[i]*pi/180)));
ordoA[i]:=ordo[jointM]-trunc(round(long[g]*sin(angleG[i]*pi/180)));
if ((prisme[jointB]=90)or (prisme[jointB]=270)) then
begin
t:=t+1;
angleG[t]:=angleG[i];abscA[t]:=abscA[i];ordoA[t]:=ordoA[i];
x1:=(e-a*sin(pi/180*(prisme[jointB]-angleG[t])))/b;x1:=x1*x1;
if x1=0 then mu:=90
else if x1<1 then mu:=180/pi*sqrt(Arctan(1/x1-1));
gotoxy(10,16);
writeln(LST, ' | ',angleG[t]:4:1,' ',',',mu:4:1,' ',',',
',', ' | ');
abscB[t]:=absc[jointB];
ordoB[t]:=ordo[i]+round(sqrt(sqrt(long[k])-sqrt(abscB[t]-abscA[t])));
trigo(abscA[t],ordoA[t],abscB[t],ordoB[t]);
angleK[t]:=zeta;
end
else
begin
interarcdroite(abscA[i],ordoA[i],absc[jointB],ordo[jointB],long[k],prisme[jointB]);
if delta>=0 then
begin
t:=t+1;
angleG[t]:=angleG[i];abscA[t]:=abscA[i];ordoA[t]:=ordoA[i];
x1:=(e-a*sin(pi/180*(prisme[jointB]-angleG[t])))/b;x1:=x1*x1;
if x1=0 then mu:=90
else if x1<1 then mu:=180/pi*sqrt(Arctan(1/x1-1));
gotoxy(10,16);

```

```

        writeln(LST, ' |', angleG[t]:4:1, ' ', ' ', mu:4:1, ' ', ' ',
        ', ' |');
        abscB[t]:=xi;
        ordoB[t]:=yi;
        trigo(abscA[t],ordoA[t],abscB[t],ordoB[t]);
        angleK[t]:=zeta;
    end;
end;
end;
gotoxy(10,17);writeln(LST, ' |-----| '); end;
end;

```

Procedure ATMC1;

(elle détermine l'angle de transmission d'un manivelle coulisseau inversé)

var

e,a,b,d,alpha,beta,mu,theta2:real;

pro:integer;

AQ2:real;

longK:array[1..360] of real;

begin

if veri2_MCI(nomdessin)=false then

begin

clrscr;

gotoXY(20,12);write('vous n'avez pas le bon mécanisme');

end

else

begin

t:=0;

for i:=1 to 360 do

begin

```

angleG[i]:=angle[g]+i; if angleG[i]>360 then angleG[i]:=angleG[i]-360;
theta2:=angleG[i]-angle[f]+180; d:=long[f];
a:=long[g];b:=long[k];e:=long[h];
abscA[i]:=absc[jointM]+round(long[g]*cos(pi/180*theta2));
ordoA[i]:=ordo[jointM]-round(long[g]*sin(pi/180*theta2));
AQ2:=sqr(abscA[i]-absc[jointQ])+sqr(ordoA[i]-ordo[jointQ]);
if AQ2-e*e>=0 then
begin
longK[i]:=AQ2-e*e;longK[i]:=sqrt(longK[i]);
interarcs(abscA[i],ordoA[i],absc[jointQ],ordo[jointQ],longK[i],e);
if delta>=0 then
begin
t:=t+1;
angleG[t]:=angleG[i];abscA[t]:=abscA[i];ordoA[t]:=ordoA[i];longK[t]:=longK[i];
abscB[t]:=xi;ordoB[t]:=yi;
trigo(abscA[t],ordoA[t],abscB[t],ordoB[t]);
angleK[t]:=zeta;
trigo(abscB[t],ordoB[t],absc[jointQ],ordo[jointQ]);
angleH[t]:=zeta;
end;
end;
end;
clrscr;
gotoxy(10,10);writeln(LST, ' _____ ');
gotoxy(10,12);writeln(LST, ' | Angle de transmission  $\mu$  MCI | ');
gotoxy(10,14);writeln(LST, ' | position entrée  $\mu$  | ');
for i:=1 to 360 do
end;
end;

```

Procédure menuAT;

(elle donne le menu secondaire pour les angles de transmission)

```

var
  choix:integer;
begin
  clrscr;
  gotoXY(5,5); write('1:ANGLE DE TRANSMISSION POUR MECANISME A QUATRE MEMBRURES SANS COULISSEAU');
  gotoXY(5,7); write('2:ANGLE DE TRANSMISSION POUR MECANISME A QUATRE MEMBRURES AVEC COULISSEAU');
  gotoXY(5,9); write('0:pour sortir');
  gotoXY(50,12);write('VOTRE CHOIX ? ');read(choix);
  case choix of
    1:begin
      ATM4M;
      sortie;
      if sort=#27 then menuAT
      else sortie;
    end;
    2:begin
      appeler;
      charger(nomdessin);
      if (( s=2)and(j=3)) then ATMC
      else
        begin
          if(((s=4)and(j=4))or((s=3)and(j=3))) then ATMC1
          else
            begin
              clrscr;
              gotoXY(10,12);write('vous n'avez pas le bon mécanisme');
            end;
          end;
        sortie;
        if sort=#27 then menuAT
        else sortie;
      end;
    end;
  end;
end;

```



```
end;  
end;
```

```
Begin  
  clrscr;  
  menuAT;  
End.
```

MODULE 9: ANALYSE DE LA POSITION A UN INSTANT (instant)

```

Program instant;
  {il étudie le mécanisme à un instant donné}

Uses crt,graph,grimpr,unit0,progene,printer;

var
  impres:char;
  code_retour:byte;

Procedure POINM4M;
  {elle affiche un M4M dans une position instantannée}

  var
    entree,sortie,inter,fixe:string;
    x1,a,b,c,d,mu,theta2,eff:real;
  begin
    appeler;
    charger(nomdessin);
    if ((s<>4) or (j<>4)) then
      begin
        clrscr;
        gotoxy(20,12);writeln('ce dessin n'a pas 4 membrures!');
        delay(10000);
      end
    else
      begin
        clrscr;
        gotoXY( 11,4);write('les membrures sont :');
        for i:=1 to 4 do
          begin
            gotoXY(5+6*i,6);write(seg[i],',');
          end
        end
      end
    end
  end

```

```

end;
gotoXY(20,10);write('membrure d'entrée ? ');
gotoXY(20,12);write('membrure intermédiaire ? ');
gotoXY(20,14);write('membrure de sortie ? ');
gotoXY(20,16);write('membrure fixe ? ');
gotoxy(20,18);write('sortie à l'imprimante O/N ');
gotoXY(45,10);lire(entree);
gotoXY(45,12);lire(inter);
gotoXY(45,14);lire(sortie);
gotoXY(45,16);lire(fixe);
gotoxy(55,18);impres:=upcase(readkey);
for i:=1 to 4 do
begin
if seg[i]=entree then g:=i;
if seg[i]=inter then k:=i;
if seg[i]=sortie then h:=i;
if seg[i]=fixe then f:=i;
end;
a:=long[g];b:=long[k];c:=long[h];d:=long[f];
for i:=1 to 4 do
begin
if joint[i]=copy(seg[k],1,1) then jointA:=i;
if joint[i]=copy(seg[k],2,1) then jointB:=i;
if joint[i]=copy(seg[f],1,1) then jointQ:=i;
if joint[i]=copy(seg[f],2,1) then jointM:=i;
end;
t:=0;clrscr;
for i:=1 to 1 do
begin
gotoxy(10,12);write(' donner la position (dégré) de la membrure motrice à cet instant ');
gotoxy(12,70);read(angleG[i]);
if angleG[i]>360 then angleG[i]:=angleG[i]-360;
abscA[i]:=absc[jointM]+round(long[g]*cos(angleG[i]*pi/180));

```

```

ordoA[i]:=ordo[jointM]-round(long[g]*sin(angleG[i]*pi/180));
interarcs(abscA[i],ordoA[i],absc[jointQ],ordo[jointQ],long[k],long[h]);
if delta<0 then
begin
  clrscr;gotoxy(12,10);write('CE N'EST PAS UNE POSITION DE CE MECANISME');
end
else
begin
  t:=t+1;angleG[t]:=angleG[i];abscA[t]:=abscA[i];ordoA[t]:=ordoA[i];
  abscB[t]:=xi; ordoB[t]:=yi;trigo(abscA[t],ordoA[t],abscB[t],ordoB[t]);
  angleK[t]:=zeta;
  trigo(abscB[t],ordoB[t],absc[jointQ],ordo[jointQ]);
  angleH[t]:=zeta;
end;
end;
clrscr;
gotoxy(10,10);writeln(LST, ' _____ ');
gotoxy(10,12);writeln(LST, ' |          LES PARAMETRES DU M4M          | ');
gotoxy(10,14);writeln(LST, ' |      entrée   inter.   sortie   fixe      | ');
gotoxy(10,16);
writeln(LST, ' |          ',seg[g],',          ',seg[k],',          ',
seg[h],',          ',seg[f],',          ', ' ', ' | ');
gotoxy(10,18);
writeln(LST, ' | longueur ',long[g]:4:1,',          ',long[k]:4:1,
' ',long[h]:4:1,',          ',long[f]:4:1,',          ', ' | ');
gotoxy(10,20);
writeln(LST, ' | pente ',angleG[t]:4:1,',          ',angleK[t]:4:1,
' ',angleH[t]:4:1,',          ',angle[f]:4:1,',          ', ' | ');
gotoxy(10,21);writeln(LST, ' |          | ');
gotoxy(10,23);writeln(LST, ' _____ ');

cadre;
for i:=1 to t do

```

```

begin
  visualisation_M4M
(absc[jointM],ordo[jointM],abscA[i],ordoA[i],abscB[i],ordoB[i],absc[jointQ],ordo[jointQ]);

  outtextxy(5,5 , 'segment long. angle');
  str(long[g]:4:1,xst);str(angleG[i]:4:1,yst);
  outtextxy(5,20,' '+seg[g]+' '+xst+' '+ yst);
  str(long[k]:4:1,xst);str(angleK[i]:4:1,yst);
  outtextxy(5,30,' '+seg[k]+' '+xst+' '+ yst);
  str(long[h]:4:1,xst);str(angleH[i]:4:1,yst);
  outtextxy(5,40,' '+seg[h]+' '+xst+' '+ yst);
  str(long[f]:4:1,xst);str(angle[f]:4:1,yst);
  outtextxy(5,50,' '+seg[f]+' '+xst+' '+ yst);
  outtextxy(absc[jointM]-15,ordo[jointM]-15,''+joint[jointM]);
  outtextxy(absc[jointQ]-15,ordo[jointQ]-15,''+joint[jointQ]);
  outtextxy(abscA[i]-15,ordoA[i]-15,''+joint[jointA]);
  outtextxy(abscB[i]-15,ordoB[i]-15,''+joint[jointB]);

  if impres='0' then imprime_ecran(code_retour);
  delay(20000);
  end;
  closegraph;
end;
end;

```

Procédure POINMC;

(elle affiche un manivelle coulisseau dans une position instantannée)

```

var
  xn,yn:longint;
  a0:string[1];
  temp:integer;
  an,e:real;
  a,b,x1,mu,theta2:real;
begin

```

```

temp:=0;
for i:=1 to 3 do
begin
if typejoint[i]='P' then temp:=temp+1;
end;
if (temp<>1) then
begin
clrscr;
gotoxy(20,12);write('vous n'avez pas un manivelle coulisseau');
end
else
begin
gotoxy(20,18);write('sortie à l'imprimante O/N ');
gotoxy(45,60);impres:=upcase(readkey);

for i:=1 to 3 do if typejoint[i]='P' then jointB:=i;
for i:=1 to 2 do
begin
if copy(seg[i],1,1)=joint[jointB] then
begin
a0:=copy(seg[i],2,1);
k:=i;
end;
if copy(seg[i],2,1)=joint[jointB] then
begin
a0:=copy(seg[i],1,1);
k:=i;
end;
end;

for i:=1 to 3 do if joint[i]=a0 then jointA:=i;
for i:=1 to 3 do if ((i<>jointA) and (i<>jointB)) then jointM:=i;
if k=1 then g:=2;
if k=2 then g:=1;

```

```

xn:=absc[jointM];a:=long[g];b:=angle[k];
an:=(1.0*xn-1.0*absc[jointB])*tg(prisme[jointB]);
yn:=ordo[jointB]+trunc(an);
e:=trunc(round((yn-ordo[jointM])*sin(90-prisme[jointB])));
t:=0;clrscr;
gotoxy(12,10);write('donner la position (dégré) de la manivelle à cet instant!');
gotoxy(70,10);read(angleG[1]);
i:=1;
  if angleG[i]>360 then angleG[i]:=angleG[i]-360;
  abscA[i]:=absc[jointM]+trunc(round(long[g]*cos(angleG[i]*pi/180)));
  ordoA[i]:=ordo[jointM]-trunc(round(long[g]*sin(angleG[i]*pi/180)));
  if ((prisme[jointB]=90)or (prisme[jointB]=270)) then
  begin
    t:=t+1;
    angleG[t]:=angleG[i];abscA[t]:=abscA[i];ordoA[t]:=ordoA[i];
    abscB[t]:=absc[jointB];
    if (sqr(long[k])-sqr(abscB[t]-abscA[t])) >=0 then
    begin
      ordoB[t]:=ordo[i]+round(sqrt(sqr(long[k])-sqr(abscB[t]-abscA[t])));
      trigo(abscA[t],ordoA[t],abscB[t],ordoB[t]);
      angleK[t]:=zeta;
    end
    else t:=t-1;
  end
  else
  begin
    interarcdroite(abscA[i],ordoA[i],absc[jointB],ordo[jointB],long[k],prisme[jointB]);
    if delta>=0 then
    begin
      t:=t+1;
      angleG[t]:=angleG[i];abscA[t]:=abscA[i];ordoA[t]:=ordoA[i];
      abscB[t]:=xi;
    end
  end

```

```

        ordoB[t]:=yi;
        trigo(abscA[t],ordoA[t],abscB[t],ordoB[t]);
        angleK[t]:=zeta;
    end;
end;

gotoxy(10,10);writeln(LST, ' _____ ');
gotoxy(10,12);writeln(LST, ' |           LES PARAMETRES DU MC           | ');
gotoxy(10,14);writeln(LST, ' |           entrée   auxil.           coulisseau | ');
gotoxy(10,16);
writeln(LST, ' |           ',seg[g],',           ',seg[k],',           ',
' ' ',',           ',joint[jointB],',           | ');
gotoxy(10,18);
writeln(LST, ' | longueur ',long[g]:4:1,',           ',long[k]:4:1,
' ',',           ',           ',           ',           ',           ',           ',           | ');
gotoxy(10,20);
writeln(LST, ' | pente ',angleG[t]:4:1,',           ',angleK[t]:4:1,
' ',',           ',           ',           ',           ',           ',           | ');
gotoxy(10,21);writeln(LST, ' |           | ');
gotoxy(10,23);writeln(LST, ' _____ ');

clrscr;

cadre;

for i:=1 to t do
begin
    visualisation_MC(absc[jointM],ordo[jointM],abscA[i],ordoA[i],abscB[i],ordoB[i]);
    outtextxy(5,5 , 'segment long. angle');
    str(long[g]:4:1,xst);str(angleG[i]:4:1,yst);
    outtextxy(5,20,' '+seg[g]+' '+xst+' '+ yst);
    str(long[k]:4:1,xst);str(angleK[i]:4:1,yst);
    outtextxy(5,30,' '+seg[k]+' '+xst+' '+ yst);
    str(prisme[jointB]:4:1,xst);
    outtextxy(5,40,' coulisseau: '+' '+xst+' '+ 'degre');
    outtextxy(absc[jointM]-15,ordo[jointM]-15,''+joint[jointM]);
end;

```



```

    outtextxy(abscA[i]-15,ordoA[i]-15,''+joint[jointA]);
    outtextxy(abscB[i]-15,ordoB[i]-15,''+joint[jointB]);
    if impres='0' then imprime_ecran(code_retour);
    delay(20000);

    end;
closegraph;
end;
end;

Procedure POINMCI;
(elle affiche la position instantannée d'un manivelle coulisseau inversé)
var
    e,a,b,d,alpha,beta,mu,theta2:real;
    pro:integer;
    Aq2:real;
    longK:array[1..360] of real;

begin
    if veri2_MCI(nondessin)=false then
        begin
            clrscr;
            gotoxy(20,12);write('vous n'avez pas le bon mécanisme');
        end
    else
        begin
            clrscr;
            gotoxy(5,12);write(' donner l'angle (dégrés) de la manivelle à cet instant');
            gotoxy(70,12);read(angleG[1]);
            gotoxy(20,18);write('sortie à l'imprimante O/N ');
            gotoxy(45,60);impres:=upcase(readkey);
        end
    end;
end;

```

```

t:=0;

theta2:=angleG[1]-angle[f]+180; d:=long[f];
a:=long[g];b:=long[k];e:=long[h];
abscA[1]:=absc[jointM]+round(long[g]*cos(pi/180*theta2));
ordoA[1]:=ordo[jointM]-round(long[g]*sin(pi/180*theta2));
AQ2:=sqr(abscA[1]-absc[jointQ])+sqr(ordoA[1]-ordo[jointQ]);
if AQ2-e*e>=0 then
begin
longK[1]:=AQ2-e*e;longK[1]:=sqrt(longK[1]);
interarcs(abscA[1],ordoA[1],absc[jointQ],ordo[jointQ],longK[1],e);
if delta<0 then
begin
clrscr;
gotoxy(12,10);write('ce n'est pas une position du mécanisme');
end
else
begin
t:=t+1;
angleG[t]:=angleG[1];abscA[t]:=abscA[1];ordoA[t]:=ordoA[1];longK[t]:=longK[1];
abscB[t]:=xi;ordoB[t]:=yi;
trigo(abscA[t],ordoA[t],abscB[t],ordoB[t]);
angleK[t]:=zeta;
trigo(abscB[t],ordoB[t],absc[jointQ],ordo[jointQ]);
angleH[t]:=zeta;
end;
end;
clrscr; gotoxy(20,18);write('sortie à l'imprimante O/N ');

clrscr;
gotoxy(10,10);writeln(LST, ' _____ ');
gotoxy(10,12);writeln(LST, ' | LES PARAMETRES DU MCI | ');
gotoxy(10,14);writeln(LST, ' | entrée auxil. sortie fixe coulis. | ');
gotoxy(10,16);

```

```

writeln(LST, ' | ',seg[g],', ' ,seg[k],', ' ',
seg[h],', ' ',seg[f],', ' ',joint[jointB],', ' | ');
gotoxy(10,18);
writeln(LST, ' | longueur ',long[g]:4:1,', ' ,longK[t]:4:1,
' ',long[h]:4:1,', ' ',long[f]:4:1,',', ' | ');
gotoxy(10,20);
writeln(LST, ' | pente ',angleG[t]:4:1,', ' ,angleK[t]:4:1,
' ',angleH[t]:4:1,', ' ',angle[f]:4:1,', ' ,angleK[t]:4:1,', ' | ');
gotoxy(10,21);writeln(LST, ' | ');
gotoxy(10,23);writeln(LST, ' | ');
cadre;
for i:=1 to t do
begin
prisme[jointB]:=angleK[i];
if v=jointA then
begin
xjoint:=abscA[i];yjoint:=ordoA[i];
end;
if v=jointB then
begin
xjoint:=abscB[i];yjoint:=ordoB[i];
end;
if v=jointQ then
begin
xjoint:=absc[jointQ];yjoint:=ordo[jointQ];
end;
visualisationMC1
(absc[jointM],ordo[jointM],abscA[i],ordoA[i],abscB[i],ordoB[i],absc[jointQ],ordo[jointQ]);
outtextxy(5,5 , 'segment long. angle');
str(long[g]:4:1,xst);str(angleG[i]:4:1,ystr);
outtextxy(5,20,' '+seg[g]+' '+xst+' '+ ystr);
str(longK[i]:4:1,xst);str(angleK[i]:4:1,ystr);
outtextxy(5,30,' '+seg[k]+' '+xst+' '+ ystr);

```

```

    str(long[h]:4:1,xst);str(angleH[i]:4:1,yst);
outtextxy(5,40,' '+seg[h]+' '+xst+' '+ yst);
    str(long[f]:4:1,xst);str(angle[f]:4:1,yst);
outtextxy(5,50,' '+seg[f]+' '+xst+' '+ yst);
    str(prisme[jointB]:4:1,xst);
outtextxy(5,60,' coulisseau '+xst+' degre');
outtextxy(absc[jointM]-15,ordo[jointM]-15,''+joint[jointM]);
outtextxy(absc[jointQ]-15,ordo[jointQ]-15,''+joint[jointQ]);
outtextxy(abscA[i]-15,ordoA[i]-15,''+joint[jointA]);
outtextxy(abscB[i]-15,ordoB[i]-15,''+joint[jointB]);
    if impres='0' then imprime_ecran(code_retour);
    delay(20000);
    end;
    closegraph;
    end;
end;

```

Procédure menupoin;

(elle donne le menu secondaire pour la sélection d'une position instantannée)

```

var
    choix:integer;
begin
    clrscr;
    gotoXY(5,5); write('1:POSITION INSTANTANNEE POUR MECANISME A QUATRE MEMBRURES SANS COULISSEAU');
    gotoXY(5,7); write('2:POSITION INSTANTANNEE POUR MECANISME A QUATRE MEMBRURES AVEC COULISSEAU');
    gotoxy(5,9); write('0: SORTIE');
    gotoXY(50,12);write('VOTRE CHOIX ? ');read(choix);
    case choix of
        1:begin
            POINM4M;
            sortie;
            if sort=#27 then menupoin

```

```

    else sortie;
end;
2:begin
    appeler;
    charger(nomdessin);
    if (( s=2)and(j=3)) then POINMC
    else
        begin
            if(((s=4)and(j=4))or((s=3)and(j=3))) then POINMC1
            else
                begin
                    clrscr;
                    gotoXY(10,12);write('vous n'avez pas le bon mécanisme');
                    end;
                end;
            sortie;
            if sort=#27 then menupoin
            else sortie;
        end;
    end;
end;
end;
end;

```

Begin

```
clrscr;
```

```
menupoin;
```

End.

MODULE 10: ETUDE CINEMATIQUE (cinemati)

```
Uses crt,graph,unit0,progene,printer ;
```

```
Procedure PVA_M4M;
```

```
{elle détermine les vitesses des joints et les vitesses angulaires des  
membrures lorsque la vitesse de la membrure d'entrée est connue}
```

```
var
```

```
OO,MO,AO,BO,entree,sortie,inter,fixe:string;
```

```
modV,argV:array[1..3] of real;
```

```
modP,argP:array[1..4] of real;
```

```
modgama,arggama:array[1..6] of real;
```

```
a,b,c,d:real;
```

```
theta2,theta3,theta4:real;
```

```
alpha2,alpha3,alpha4:real;
```

```
w2,w3,w4:real;
```

```
K1,K2,K3,K4,K5:real;
```

```
ap,bp,cp,dp,ep,fp,ac,bc,cc,dc,ec,fc:real;
```

```
begin
```

```
  appeler;
```

```
  charger(nomdessin);
```

```
  if ((s<>4) or (j<>4)) then
```

```
    begin
```

```
      clrscr;
```

```
      gotoxy(20,12);writeln('ce dessin n'a pas 4 membrures');
```

```
      delay(10000);
```

```
    end
```

```
  else
```

```
    begin
```

```
      clrscr;
```

```
      gotoXY( 11,4);write('les membrures sont :');
```

```

for i:=1 to 4 do
  begin
    gotoXY(5+6*i,6);write(seg[i],');
  end;
gotoXY(20,10);write('membrure d'entrée ? ');
gotoXY(20,12);write('membrure intermédiaire ? ');
gotoXY(20,14);write('membrure de sortie ? ');
gotoXY(20,16);write('membrure fixe ? ');
gotoXY(45,10);lire(entree);
gotoXY(45,12);lire(inter);
gotoXY(45,14);lire(sortie);
gotoXY(45,16);lire(fixe);
for i:=1 to 4 do
  begin
    if seg[i]=entree then g:=i;
    if seg[i]=inter then k:=i;
    if seg[i]=sortie then h:=i;
    if seg[i]=fixe then f:=i;
  end;

Q0:=copy(seg[f],1,1);M0:=copy(seg[g],1,1);A0:=copy(seg[k],1,1);B0:=copy(seg[h],1,1);
for i:=1 to 4 do
  begin
    if joint[i]=Q0 then jointQ:=i;
    if joint[i]=M0 then jointM:=i;
    if joint[i]=A0 then jointA:=i;
    if joint[i]=B0 then jointB:=i;
  end;
clrscr;
d:=long[f];b:=long[k];c:=long[h];a:=long[g];
gotoxy(12,12);write('donner la position angulaire de la membrure d'entrée: ');
gotoxy(12,14);write('donner la vitesse angulaire de la membrure d'entrée: ');
gotoxy(12,16);write('donner l'accélération angulaire de la membrure d'entrée: ');

```

```

gotoxy(70,12);read(theta2);theta2:=theta2-angle[f]+180;
gotoxy(70,14);read(w2);
gotoxy(70,16);read(alpha2);
if ((a<>0) and (c<>0) and (b<>0)) then
begin
K1:=d/a;
K2:=d/c;
K3:=(a*a-b*b+c*c+d*d)/(2*a*c);
ap:=cos(theta2*pi/180)+K3-K1-K2*cos(theta2*pi/180);
bp:=-2*sin(theta2*pi/180);
cp:=K1+K3-(1+K2)*cos(theta2*pi/180);
if ap<>0 then
begin
if (bp*bp-4*ap*cp)>=0 then
begin
if bp>0 then theta4:=180/pi*2*arctan((-bp+sqrt(bp*bp-4*ap*cp))/(2*ap))
else
theta4:=180/pi*2*arctan((-bp-sqrt(bp*bp-4*ap*cp))/(2*ap));
K4:=d/b;
K5:=(c*c-d*d-a*a-b*b)/(2*a*b);
dp:=K4*cos(theta2*pi/180)+cos(theta2*pi/180)+K5-K1;
ep:=-2*sin(theta2*pi/180);
fp:=K4*cos(theta2*pi/180)-cos(theta2*pi/180)+K5+K1;
if dp<>0 then
begin
if (ep*ep-4*dp*fp) >=0 then
begin
if ep<0 then theta3:=180/pi*2*arctan((-ep-sqrt(ep*ep-4*dp*fp))/(2*dp))
else
theta3:=180/pi*2*arctan((-ep+sqrt(ep*ep-4*dp*fp))/(2*dp));
if (theta4<>theta3) then
begin
w4:=a*w2/c*sin((pi/180)*(theta2-theta3))/sin(pi/180*(theta4-theta3));

```



```

w3:=a*w2/b*sin(pi/180*(theta4-theta2))/sin(pi/180*(theta3-theta4));
ac:=c*sin(pi/180*theta4);
bc:=b*sin(pi/180*theta3);
cc:=a*alpha2*sin(pi/180*theta2)+a*w2*w2*cos(pi/180*theta2)+
      b*w3*w3*cos(pi/180*theta3)-c*w4*w4*cos(pi/180*theta4);
dc:=c*cos(pi/180*theta4);
ec:=b*cos(pi/180*theta3);
fc:=a*alpha2*cos(pi/180*theta2)-a*w2*w2*sin(pi/180*theta2)-
      b*w3*w3*sin(pi/180*theta3)+c*w4*w4*sin(pi/180*theta4);
if (ac*ec<>bc*dc) then
begin
  alpha3:=(cc*dc-ac*fc)/(ac*ec-bc*dc);
  alpha4:=(cc*ec-bc*fc)/(ac*ec-bc*dc);
  modP[1]:=d;argP[1]:=180;
  modP[2]:=a;argP[2]:=theta2;
  modP[3]:=b;argP[3]:=theta3;
  modP[4]:=c;argP[4]:=theta4;
  complexe(round(-a*w2*sin(pi/180*theta2)),round(a*w2*cos(pi/180*theta2)));
  modV[1]:=module;argV[1]:=argument;
  complexe(round(-b*w3*sin(pi/180*theta3)),round(b*w3*cos(pi/180*theta3)));
  modV[2]:=module;argV[2]:=argument;
  complexe(round(-c*w4*sin(pi/180*theta4)),round(c*w4*cos(pi/180*theta4)));
  modV[3]:=module;argV[3]:=argument;
  complexe(round(-a*w2*w2*cos(pi/180*theta2)),round(-a*w2*w2*sin(pi/180*theta2)));
  modgama[1]:=module;arggama[1]:=argument;
  complexe(round(-a*alpha2*sin(pi/180*theta2)),round(a*alpha2*cos(pi/180*theta2)));
  modgama[2]:=module;arggama[2]:=argument;
  complexe(round(-b*w3*w3*cos(pi/180*theta3)),round(-b*w3*w3*sin(pi/180*theta3)));
  modgama[3]:=module;arggama[3]:=argument;
  complexe(round(-b*alpha3*sin(pi/180*theta3)),round(b*alpha3*cos(pi/180*theta3)));
  modgama[4]:=module;arggama[4]:=argument;
  complexe(round(-c*w4*w4*cos(pi/180*theta4)),round(-c*w4*w4*sin(pi/180*theta4)));
  modgama[5]:=module;arggama[4]:=argument;

```

```

complexe(round(-c*alpha4*sin(pi/180*theta4)),round(c*alpha4*cos(pi/180*theta4)));
modgama[6]:=module;arggama[6]:=argument;

clrscr;

gotoxy(40,5);write('déplacement');

gotoxy(12,8);write('entité');gotoxy(36,8);write('module');gotoxy(61,8);write('argument');

gotoxy(10,11);write('noeud ',M0,'/',O0);

gotoxy(37,11);write(modP[1]:4:1);gotoxy(63,11);write(argP[1]:4:1);

gotoxy(10,13);write('noeud ',A0);

gotoxy(37,13);write(modP[2]:4:1);gotoxy(63,13);write(argP[2]:4:1);

gotoxy(10,15);write('noeud ',B0);

gotoxy(37,15);write(modP[3]:4:1);gotoxy(63,15);write(argP[3]:4:1);

gotoxy(28,21);write('appuyer sur <BARRE D'ESPACEMENT> pour continuer');

repeat until ReadKey = ' ';

clrscr;

gotoxy(40,5);write('vitesse');

gotoxy(12,8);write('entité');gotoxy(36,8);write('module');gotoxy(61,8);write('argument');

gotoxy(10,11);write('noeud ',A0);

gotoxy(37,11);write(modV[1]:4:1);gotoxy(63,11);write(argV[1]:4:1);

gotoxy(10,13);write('noeud ',B0,'/',A0);

gotoxy(37,13);write(modV[2]:4:1);gotoxy(63,13);write(argV[2]:4:1);

gotoxy(10,15);write('noeud ',B0);

gotoxy(37,15);write(modV[3]:4:1);gotoxy(63,15);write(argV[3]:4:1);

gotoxy(28,21);write('appuyer sur <BARRE D'ESPACEMENT> pour continuer');

repeat until ReadKey = ' ';

clrscr;

gotoxy(40,5);write('accélération radiale');

gotoxy(12,8);write('entité');gotoxy(36,8);write('module');gotoxy(61,8);write('argument');

gotoxy(10,11);write('noeud ',A0); gotoxy(37,11);write(modgama[1]:4:1);gotoxy(63,11);
write(arggama[1]:4:1);

gotoxy(10,13);write('noeud ',B0,'/',A0);gotoxy(37,13);write(modgama[3]:4:1);gotoxy(63,13);
write(arggama[3]:4:1);

gotoxy(10,15);write('noeud ',B0); gotoxy(37,15);write(modgama[5]:4:1);gotoxy(63,15);
write(arggama[5]:4:1);

```

```

gotoxy(28,21);write('appuyer sur <BARRE D'ESPACEMENT> pour continuer');
repeat until ReadKey = ' ';
clrscr;
gotoxy(40,5);write('accélération tangentielle');
gotoxy(10,8);write('entité');gotoxy(36,8);write('module');gotoxy(61,8);write('argument');
gotoxy(12,11);write('noeud ',A0); gotoxy(37,11);write(modgama[2]:4:1);gotoxy(63,11);
write(aggama[2]:4:1);
gotoxy(12,13);write('noeud ',B0,'/',A0);gotoxy(37,13);write(modgama[4]:4:1);gotoxy(63,13);
write(aggama[4]:4:1);
gotoxy(12,15);write('noeud ',B0); gotoxy(37,15);write(modgama[6]:4:1);gotoxy(63,15);
write(aggama[6]:4:1);
gotoxy(28,21);write('appuyer sur <BARRE D'ESPACEMENT> pour continuer');
repeat until ReadKey = ' ';
clrscr;
gotoxy(40,5);write('paramètres angulaires');
gotoxy(10,8);write('entité');gotoxy(21,8);write('position
ang. ');gotoxy(40,8);write('vitesse ang. ');
gotoxy(55,8);
write('accélération ang. ');
theta2:=theta2+angle[f]-180;
gotoxy(9,11);write('membrane:',M0,A0);gotoxy(24,11);write(theta2:4:1);gotoxy(43,11);
write(w2:4:1);gotoxy(59,11);
write(alpha2:4:1);
gotoxy(9,13);
write('membrane:',A0,B0);gotoxy(24,13);write(theta3:4:1);gotoxy(43,13);write(w3:4:1);
gotoxy(59,13);write(alpha3:4:1);
gotoxy(9,15);
write('membrane:',Q0,B0);gotoxy(24,15);write(theta4:4:1);gotoxy(43,15);write(w4:4:1);
gotoxy(59,15); write(alpha4:4:1);
gotoxy(28,21);write('appuyer sur <BARRE D'ESPACEMENT> pour continuer');
repeat until ReadKey = ' ';
end
else

```

```

begin
  clrscr;
  gotoxy(10,12);write('vos données sont incompatibles avec ce mécanisme');
  gotoxy(10,16);write('suggestion: VISUALISER LE MECANISME, SIMULER SON MOUVEMENT');
  gotoxy(10,18);write('          POUR REVISER VOS DONNEES');
end;

end
else
begin
  clrscr;
  gotoxy(10,12);write('vos données sont incompatibles avec ce mécanisme');
  gotoxy(10,16);write('suggestion: VISUALISER LE MECANISME, SIMULER SON MOUVEMENT');
  gotoxy(10,18);write('          POUR REVISER VOS DONNEES');
end;

end
else
begin
  clrscr;
  gotoxy(10,12);write('vos données sont incompatibles avec ce mécanisme');
  gotoxy(10,16);write('suggestion: VISUALISER LE MECANISME, SIMULER SON MOUVEMENT');
  gotoxy(10,18);write('          POUR REVISER VOS DONNEES');
end;

end
else
begin
  clrscr;
  gotoxy(10,12);write('vos données sont incompatibles avec ce mécanisme');
  gotoxy(10,16);write('suggestion: VISUALISER LE MECANISME, SIMULER SON MOUVEMENT');
  gotoxy(10,18);write('          POUR REVISER VOS DONNEES');
end;

```

```

end
else
  begin
    clrscr;
    gotoxy(10,12);write('vos données sont incompatibles avec ce mécanisme');
    gotoxy(10,16);write('suggestion: VISUALISER LE MECANISME, SIMULER SON MOUVEMENT');
    gotoxy(10,18);write('          POUR REVISER VOS DONNEES');
  end;
end
else
  begin
    clrscr;
    gotoxy(10,12);write('vos données sont incompatibles avec ce mécanisme');
    gotoxy(10,16);write('suggestion: VISUALISER LE MECANISME, SIMULER SON MOUVEMENT');
    gotoxy(10,18);write('          POUR REVISER VOS DONNEES');
  end;

end

end;
end;

Procedure PVA_MC;

```

(elle détermine les position,vitesse et accélération angulaires pour un manivelle-coulisseau)

var

```
xn,yn:longint;
a0,MO,A0,B0:string[1];
temp:integer;
an,e,a,b:real;
modP,argP,modV,argV:array[1..3] of real;
modgama,arggama:array[1..5] of real;
theta1,theta2,theta3,theta4:real;
alpha2,alpha3:real;
w2,w3:real;
mp,np,sp:real;
sprime,sseconde:real;

begin
temp:=0;
for i:=1 to 3 do
begin
if typejoint[i]='P' then temp:=temp+1;
end;
if (temp<>1) then
begin
clrscr;
gotoXY(20,12);write('vous n'avez pas un manivelle coulisseau');
end
else
begin
for i:=1 to 3 do
begin
if typejoint[i]='P' then jointB:=i;
end;
end;
```

```

for i:=1 to 2 do
begin
  if copy(seg[i],1,1)=joint[jointB] then
  begin
    a0:=copy(seg[i],2,1);
    k:=i;
  end;
  if copy(seg[i],2,1)=joint[jointB] then
  begin
    a0:=copy(seg[i],1,1);
    k:=i;
  end;
end;
for i:=1 to 3 do if joint[i]=a0 then jointA:=i;
for i:=1 to 3 do if ((i<>jointA) and (i<>jointB)) then jointM:=i;
AO:=joint[jointA];BO:=joint[jointB];MO:=joint[jointM];
if k=1 then g:=2;
if k=2 then g:=1;
xn:=absc[jointM];
an:=(1.0*xn-1.0*absc[jointB])*tg(prisme[jointB]);
yn:=ordo[jointB]+trunc(an);
e:=trunc(round((yn-ordo[jointM])*sin((pi/180)*(90-prisme[jointB]))));
if e<0 then e:=-e;
theta1:=180+prisme[jointB];
theta4:=90+prisme[jointB];
clrscr;
a:=long[g];b:=long[k];
gotoxy(12,12);write('donner la position angulaire de la membrure d'entrée: ');
gotoxy(12,14);write('donner la vitesse angulaire de la membrure d'entrée: ');
gotoxy(12,16);write('donner l'accélération angulaire de la membrure d'entrée: ');
gotoxy(70,12);read(theta2);theta2:=theta2-prisme[jointB];
gotoxy(70,14);read(w2);
gotoxy(70,16);read(alpha2);

```

```

mp:=-2*a*cos(theta2*pi/180);
np:=a*a+e*e-b*b-2*a*e*sin(theta2*pi/180);
if (mp*mp-4*np)>=0 then
begin
sp:=(-mp+sqrt(mp*mp-4*np))/2;
if (a*cos(theta2*pi/180)-sp) <>0 then
begin
theta3:=180/pi*arctan((a*sin(theta2*pi/180)-e)/(a*cos(theta2*pi/180)-sp));
if ((b>0)and(cos(theta3*pi/180)<>0)) then
begin
w3:=a*w2/b*cos(theta2*pi/180)/cos(theta3*pi/180);
sprime:=a*w2*sin(pi/180*(theta3-theta2))/cos(pi/180*theta3);
alpha3:=1/(b*cos(theta3*pi/180));
alpha3:=alpha3*(a*alpha2*cos(theta2*pi/180)
-a*w2*w2*sin(theta2*pi/180)+b*w3*w3*sin(theta3*pi/180));
sseconde:=-a*alpha2*sin(pi/180*theta2)
-a*w2*w2*cos(pi/180*theta2)+b*alpha3*sin(pi/180*theta3);
sseconde:=sseconde+b*w3*w3*cos(pi/180*theta3);
modP[1]:=a;argP[1]:=theta2;
modP[2]:=b;argP[2]:=theta3;
modP[3]:=e;argP[3]:=90;
complexe(round(-a*w2*sin(pi/180*theta2)),round(a*w2*cos(pi/180*theta2)));
modV[1]:=module;argV[1]:=argument;
modV[3]:=-sprime;
complexe(round(-b*w3*sin(pi/180*theta3)),round(b*w3*cos(pi/180*theta3)));
modV[2]:=module;argV[2]:=argument;
complexe(round(-a*alpha2*sin(pi/180*theta2)),round(a*alpha2*cos(pi/180*theta2)));
modgama[1]:=module;arggama[1]:=argument;
complexe(round(-a*w2*w2*cos(pi/180*theta2)),round(-a*w2*w2*sin(pi/180*theta2)));
modgama[2]:=module;arggama[2]:=argument;
complexe(round(-b*alpha3*sin(pi/180*theta3)),round(b*alpha3*cos(pi/180*theta3)));
modgama[3]:=module; arggama[3]:=argument;
complexe(round(-b*w3*w3*cos(pi/180*theta3)),round(-b*w3*w3*sin(pi/180*theta3)));

```



```

modgama[4]:=module;arggama[4]:=argument;
modgama[5]:=sseconde;arggama[5]:=180;
clrscr;
gotoxy(40,5);write('déplacement');
gotoxy(10,8);
write('entité');gotoxy(36,8);write('module');gotoxy(61,8);write('argument');
gotoxy(12,11);write('noeud ',AO,'/',MO);
gotoxy(37,11);write(modP[1]:4:1);gotoxy(63,11);write(argP[1]:4:1);
gotoxy(12,13);write('noeud ',AO,'/',BO);
gotoxy(37,13);write(modP[2]:4:1);gotoxy(63,13);write(argP[2]:4:1);
gotoxy(12,15);write('noeud ',BO,'/XO');
gotoxy(63,15);write(modP[3]:4:1);gotoxy(63,15);write(argP[3]:4:1);
gotoxy(28,21);write('appuyer sur <BARRE D'ESPACEMENT> pour continuer');
repeat until ReadKey = ' ';
clrscr;
gotoxy(40,5);write('vitesse');
gotoxy(10,8);write('entité');gotoxy(36,8);write('module');gotoxy(61,8);write('argument');
gotoxy(12,11);write('noeud ',AO,'/',MO);
gotoxy(37,11);write(modV[1]:4:1);gotoxy(63,11);write(argV[1]:4:1);
gotoxy(12,13);write('noeud ',AO,'/',BO);
gotoxy(37,13);write(modV[2]:4:1);gotoxy(63,13);write(argV[2]:4:1);
gotoxy(12,15);write('noeud ',BO);gotoxy(37,15);write(modV[3]:4:1);
gotoxy(28,21);write('appuyer sur <BARRE D'ESPACEMENT> pour continuer');
repeat until ReadKey = ' ';
clrscr;
gotoxy(40,5);write('accélération radiale');
gotoxy(10,8);write('entité');gotoxy(36,8);write('module');
gotoxy(61,8);write('argument');
gotoxy(12,11);write('noeud ',AO,'/',MO);gotoxy(37,11);
write(modgama[2]:4:1);gotoxy(63,11);
write(arggama[2]:4:1);
gotoxy(12,13);write('noeud ',AO,'/',BO);gotoxy(37,13);write(modgama[4]:4:1);
gotoxy(63,13);

```

```

write(aggama[4]:4:1);
gotoxy(28,21);write('appuyer sur <BARRE D''ESPACEMENT> pour continuer');
repeat until ReadKey = ' ';
clrscr;
gotoxy(40,5);write('accélération tangentielle');
gotoxy(10,8);write('entité');gotoxy(36,8);write('module');gotoxy(61,8);write('argument');
gotoxy(12,11);write('noeud ,A0,','/',MO);gotoxy(37,11);
write(modgama[1]:4:1);gotoxy(63,11);
write(aggama[1]:4:1);
gotoxy(12,13);write('noeud ,A0,','/',BO);gotoxy(37,13);
write(modgama[3]:4:1);gotoxy(63,13);
write(aggama[3]:4:1);
gotoxy(12,15);write('noeud ',BO);gotoxy(37,15);write(modgama[5]:4:1);gotoxy(63,15);
write(aggama[5]:4:1);
gotoxy(28,21);write('appuyer sur <BARRE D''ESPACEMENT> pour continuer');
repeat until ReadKey = ' ';
clrscr;
gotoxy(40,5);write('paramètres angulaires');
gotoxy(7,8);write('entité');gotoxy(21,8);write('position ang. ');gotoxy(40,8);
write('vitesse ang. ');gotoxy(55,8);
write('accélération ang. ');
theta2:=theta2+prisme[jointB];
gotoxy(9,11);write('membrane:',MO,AO);gotoxy(24,11);write(theta2:4:1);gotoxy(43,11);
write(w2:4:1);gotoxy(59,11);
write(alpha2:4:1);
gotoxy(9,13);write('membrane:',BO,AO);gotoxy(24,13);write(theta3:4:1);gotoxy(43,13);
write(w3:4:1);gotoxy(59,13);
write(alpha3:4:1);
gotoxy(28,21);write('appuyer sur <BARRE D''ESPACEMENT> pour continuer');
repeat until ReadKey = ' ';
end
else
begin

```

```

    clrscr;
    gotoxy(10,12);write('vos données sont incompatibles avec ce mécanisme');
    gotoxy(10,16);write('suggestion: VISUALISER LE MECANISME, SIMULER SON MOUVEMENT');
    gotoxy(10,18);write('          POUR REVISER VOS DONNEES');
end;

end
else
begin
    clrscr;
    gotoxy(10,12);write('vos données sont incompatibles avec ce mécanisme');
    gotoxy(10,16);write('suggestion: VISUALISER LE MECANISME, SIMULER SON MOUVEMENT');
    gotoxy(10,18);write('          POUR REVISER VOS DONNEES');
end;
end
else
begin
    clrscr;
    gotoxy(10,12);write('vos données sont incompatibles avec ce mécanisme');
    gotoxy(10,16);write('suggestion: VISUALISER LE MECANISME, SIMULER SON MOUVEMENT');
    gotoxy(10,18);write('          POUR REVISER VOS DONNEES');
end;

end;
end;

```

Procédure PVA_MCI;

{elle détermine les position,vitesse et accélération angulaires d'un manivelle coulisseau inversé}

var

Q0,M0,A0,B0:string;

```

modP,argP :array[1..4] of real;
modV,argV:array[1..3] of real;
modgama,arggama:array[1..6] of real;
a,b,d,e,k1,k2,k3,theta4,theta3,w4,bprime:real;
ac,bc,cc,dc,ec,fc,alpha4,bseconde,w3:real;
theta2,w2,alpha2,alpha3,temp1,temp2:real;
begin
  if veri2_MCI(nomdessin)=false then
    begin
      clrscr;
      gotoXY(20,12);write('vous n'avez pas le bon mécanisme!');
    end
  else
    begin
      a:=long[g]; AO:=joint[jointA];
      b:=long[k]; BO:=joint[jointB];
      d:=long[f]; MO:=joint[jointM];
      e:=long[h]; BO:=joint[jointO];
      clrscr;
      gotoxy(12,12);write('donner la position angulaire de la membrure d'entrée: ');
      gotoxy(12,14);write('donner la vitesse angulaire de la membrure d'entrée : ');
      gotoxy(12,16);write('donner l'accélération angulaire de la membrure d'entrée: ');
      gotoxy(70,12);read(theta2);theta2:=theta2-angle[f]+180;
      gotoxy(70,14);read(w2);
      gotoxy(70,16);read(alpha2);
      k1:=d-e-a*cos(theta2*pi/180);
      k2:=2*a*sin(theta2*pi/180);
      k3:=a*cos(theta2*pi/180)-d-e;
      if k1<>0 then
        begin
          if (k2*k2-4*k1*k3)>=0 then
            begin
              if k2>0 then theta4:=2*180/pi*arctan((-k2+sqrt(k2*k2-4*k1*k3))/(2*k1))

```

```

else
theta4:=2*180/pi*arctan((-k2-sqrt(k2*k2-4*k1*k3))/(2*k1));
theta3:=90+theta4;
if b<>0 then
begin
w4:=a*w2/b*sin(pi/180*(theta2-theta4));w3:=w4;
bprime:=a*w2/b*(b*cos(pi/180*(theta4-theta2))-e*sin(pi/180*(theta4-theta2)));
ac:=sin(pi/180*theta4);
bc:=b*cos(pi/180*theta4)+e*sin(pi/180*theta4);
cc:=a*w2*w2*cos(pi/180*theta2)+a*alpha2*sin(pi/180*theta2);
cc:=cc-2*bprime*w4*cos(pi/180*theta4)+b*w4*w4*sin(pi/180*theta4)-e*w4*w4*cos(pi/180*theta4);
dc:=cos(pi/180*theta4);
ec:=-b*sin(pi/180*theta4)+e*cos(pi/180*theta4);
fc:=-a*w2*w2*sin(pi/180*theta2)+a*alpha2*cos(pi/180*theta2);
fc:=fc+2*bprime*w4*sin(pi/180*theta4)+b*w4*w4*cos(pi/180*theta4)+e*w4*w4*sin(pi/180*theta4);
if (ac*ec-bc*dc)<>0 then
begin
alpha4:=(ac*fc-cc*dc)/(ac*ec-bc*dc);alpha3:=alpha4;
bseconde:=(cc*ec-bc*fc)/(ac*ec-bc*dc);
modP[1]:=d;argP[1]:=180;
modP[2]:=a;argP[2]:=theta2;
modP[3]:=b;argP[3]:=theta3;
modP[4]:=e;argP[4]:=theta4;
complexe(round(-a*w2*sin(pi/180*theta2)),round(a*w2*cos(pi/180*theta2)));
modV[1]:=module;argV[1]:=argument;
complexe(round(bprime*cos(pi/180*theta3)-b*w3*sin(pi/180*theta3)),
round(bprime*sin(pi/180*theta3)+bprime*w3*cos(pi/180*theta3)));
modV[2]:=module;argV[2]:=argument;
complexe(round(-e*w4*sin(pi/180*theta4)),round(e*w4*cos(pi/180*theta4)));
modV[3]:=module;argV[3]:=argument;
temp1:=bseconde*cos(pi/180*theta3)-b*w3*w3*cos(pi/180*theta3) ;
temp2:=bseconde*sin(pi/180*theta3)-b*w3*w3*sin(pi/180*theta3);
complexe(round(temp1),round(temp2));

```

```

modgama[6]:=module; arggama[6]:=argument;
temp1:=-2*bprime*w3*sin(pi/180*theta3)-bprime*alpha3*sin(pi/180*theta3);
temp2:=2*bprime*w3*cos(pi/180*theta3)+bprime*alpha3*cos(pi/180*theta3);
complexe(round(temp1),round(temp2));
modgama[3]:=module; arggama[3]:=argument;
complexe(round(-a*w2*w2*cos(pi/180*theta2)),round(-a*w2*w2*sin(pi/180*theta2)));
modgama[1]:=module; arggama[1]:=argument;
complexe(round(-e*alpha4*cos(pi/180*theta4)),round(-e*alpha4*sin(pi/180*theta4)));
modgama[4]:=module; arggama[4]:=argument;
complexe(round(-e*w4*w4*sin(pi/180*theta4)),round(e*w4*w4*cos(pi/180*theta4)));
modgama[5]:=module; arggama[5]:=argument;
complexe(round(-a*alpha2*sin(pi/180*theta2)),round(a*alpha2*cos(pi/180*theta2)));
modgama[2]:=module; arggama[2]:=argument;

clrscr;
gotoxy(40,5);write('déplacement');
gotoxy(10,8);write('entité');gotoxy(36,8);write('module');gotoxy(61,8);write('argument');
gotoxy(12,11);write('noeud ,MO, '/',00);gotoxy(37,11);write(modP[1]:4:1);
gotoxy(63,11);write(argP[1]:4:1);
gotoxy(12,13);write('noeud ,AO, '/',MO);gotoxy(37,13);write(modP[2]:4:1);
gotoxy(63,13);write(argP[2]:4:1);
gotoxy(12,15);write('noeud ,AO, '/',80);gotoxy(37,15);write(modP[3]:4:1);
gotoxy(63,15);write(argP[3]:4:1);
gotoxy(12,17);write('noeud ,BO, '/',00);gotoxy(37,17);write(modP[4]:4:1);
gotoxy(63,17);write(argP[4]:4:1);
gotoxy(28,21);write('appuyer sur <BARRE D'ESPACEMENT> pour continuer');
repeat until ReadKey = ' ';

clrscr;
gotoxy(40,5);write('vitesse');
gotoxy(10,8);write('entité');gotoxy(36,8);write('module');gotoxy(61,8);write('argument');
gotoxy(12,11);write('noeud ,AO, '/',MO);gotoxy(37,11);
write(modV[1]:4:1);gotoxy(63,11);write(argV[1]:4:1);
gotoxy(12,13);write('noeud ,AO, '/',80);gotoxy(37,13);write(modV[2]:4:1);
gotoxy(63,13);write(argV[2]:4:1);

```

```

gotoxy(12,15);write('noeud',80,'/',00);gotoxy(37,15);write(modV[3]:4:1);
gotoxy(63,15);write(argV[3]:4:1);
gotoxy(28,21);write('appuyer sur <BARRE D'ESPACEMENT> pour continuer');
repeat until ReadKey = ' ';
clrscr;
gotoxy(40,5);write('accélération radiale');
gotoxy(10,8);write('entité');gotoxy(36,8);write('module');gotoxy(61,8);write('argument');
gotoxy(12,11);write('noeud ',80,'/',M0);gotoxy(37,11);write(modgama[1]:4:1);
gotoxy(63,11);write(arggama[1]:4:1);
gotoxy(12,13);write('noeud ',80,'/',00);gotoxy(37,13);write(modgama[4]:4:1);
gotoxy(63,13);write(arggama[4]:4:1);
gotoxy(28,21);write('appuyer sur <BARRE D'ESPACEMENT> pour continuer');
repeat until ReadKey = ' ';
clrscr;
gotoxy(40,5);write('accélération tangentielle');
gotoxy(10,8);write('entité');gotoxy(36,8);write('module');gotoxy(61,8);write('argument');
gotoxy(12,11);write('noeud ',80,'/',M0);gotoxy(37,11);write(modgama[2]:4:1);
gotoxy(63,11);write(arggama[2]:4:1);
gotoxy(12,13);write('noeud ',80,'/',00);gotoxy(37,13);write(modgama[5]:4:1);
gotoxy(63,13);write(arggama[5]:4:1);
gotoxy(28,21);write('appuyer sur <BARRE D'ESPACEMENT> pour continuer');
repeat until ReadKey = ' ';
clrscr;
gotoxy(40,5);write('paramètres angulaires');
gotoxy(7,8);write('entité');gotoxy(21,8);write('position ang. ');gotoxy(40,8);
write('vitesse ang. ');gotoxy(55,8);
write('accélération ang. ');
theta2:=theta2+angle[f]-180;
gotoxy(9,11);write('membrane:',M0,A0);gotoxy(24,11);write(theta2:4:1);
gotoxy(43,11);write(w2:4:1);gotoxy(59,11);
write(alpha2:4:1);
gotoxy(9,13);write('membrane:',80,A0);gotoxy(24,13);write(theta3:4:1);
gotoxy(43,13);write(w3:4:1);gotoxy(59,13);

```

```

write(alpha3:4:1);
gotoxy(9,15);write('membrure:',Q0,B0);gotoxy(24,15);write(theta4:4:1);
gotoxy(43,15);write(w4:4:1);gotoxy(59,15);
write(alpha4:4:1);
gotoxy(28,21);write('appuyer sur <BARRE D'ESPACEMENT> pour continuer');
repeat until ReadKey = ' ';
end
else
begin
clrscr;
gotoxy(10,12);write('vos données sont incompatibles avec ce mécanisme');
gotoxy(10,16);write('suggestion: VISUALISER LE MECANISME, SIMULER SON MOUVEMENT');
gotoxy(10,18);write('          POUR REVISER VOS DONNEES');
end;
end
else
begin
clrscr;
gotoxy(10,12);write('vos données sont incompatibles avec ce mécanisme');
gotoxy(10,16);write('suggestion: VISUALISER LE MECANISME, SIMULER SON MOUVEMENT');
gotoxy(10,18);write('          POUR REVISER VOS DONNEES');
end;
end
end
else
begin
clrscr;
gotoxy(10,12);write('vos données sont incompatibles avec ce mécanisme');
gotoxy(10,16);write('suggestion: VISUALISER LE MECANISME, SIMULER SON MOUVEMENT');
gotoxy(10,18);write('          POUR REVISER VOS DONNEES');
end;
end
else

```



```

begin
  clrscr;
  gotoxy(10,12);write('vos données sont incompatibles avec ce mécanisme');
  gotoxy(10,16);write('suggestion: VISUALISER LE MECANISME, SIMULER SON MOUVEMENT');
  gotoxy(10,18);write('          POUR REVISER VOS DONNEES');
end;

end;
end;

Procédure menucine;
(elle donne le menu secondaire pour l'analyse cinématique des mécanismes)
var
  choix:integer;
begin
  clrscr;
  gotoXY(5,5); write('1:ANALYSE CINEMATIQUE POUR MECANISME A QUATRE MEMBRURES SANS COULISSEAU');
  gotoXY(5,7); write('2:ANALYSE CINEMATIQUE POUR MECANISME A QUATRE MEMBRURES AVEC COULISSEAU');
  gotoXY(5,9); write('0:pour sortir');
  gotoXY(50,12);write('VOTRE CHOIX ? ');read(choix);
  case choix of
    1:begin
      PVA_M4M;
      sortie;
      if sort=#27 then menucine
      else sortie;
    end;
    2:begin

```

```

appeler;
charger(nomdessin);
if ((s=2)and(j=3)) then PVA_MC
else
begin
if(((s=4)and(j=4))or((s=3)and(j=3))) then PVA_MCI
else
begin
clrscr;
gotoXY(10,12);write('vous n'avez pas le bon mécanisme');
end;
end;
end;
end;
end;
end;

```

BEGIN

menucine;

END.

MODULE 11: ETUDE DU COUPLAGE A D'AUTRES MECANISMES:

LA COURBE DE BIELLE (couplage)

```
Uses crt, graph, unit0, grimpr, progene, printer;
type tabl=array[1..360] of longint;
var
  code_retour:byte;

ProcEDURE CALM4M (var xcou,ycou:tabl);
  (elle fait les calculs pour la courbe de bielle de couplage à un M4M)
  Var
    cou:real;

  begin
    t:=0;
    for i:=1 to 360 do
      begin
        angleG[i]:=angle[g]+i;
        if angleG[i]>360 then angleG[i]:=angleG[i]-360;
        abscA[i]:=absc[jointM]+round(long[g]*cos(angleG[i]*pi/180));
        ordoA[i]:=ordo[jointM]-round(long[g]*sin(angleG[i]*pi/180));
        interarcs(abscA[i],ordoA[i],absc[joint0],ordo[joint0],long[k],long[h]);
        if delta>=0 then
          begin
            t:=t+1;angleG[t]:=angleG[i];abscA[t]:=abscA[i];ordoA[t]:=ordoA[i];
            abscB[t]:=xi; ordoB[t]:=yi;trigo(abscA[t],ordoA[t],abscB[t],ordoB[t]);
            angleK[t]:=zeta;cou:=zeta+rp-angle[k];
            trigo(abscB[t],ordoB[t],absc[joint0],ordo[joint0]);
            angleH[t]:=zeta; rp:=theta+angleK[t];
            xcou[t]:=abscA[t]+round(rp*cos(pi/180*cou));
            ycou[t]:=ordoA[t]-round(rp*sin(pi/180*cou));
          end;
      end;
    end;
```

```
end;  
end;
```

```
Procedure CALMC(var xcou,ycou:tabl);
```

```
{elle fait les calculs pour la courbe de bielle d'un MC}
```

```
Var
```

```
cou:real;
```

```
begin
```

```
  t:=0;
```

```
  for i:=1 to 360 do
```

```
    begin
```

```
      angleG[i]:=angle[g]+i;if angleG[i]>360 then angleG[i]:=angleG[i]-360;
```

```
      abscA[i]:=absc[jointM]+trunc(round(long[g]*cos(angleG[i]*pi/180)));
```

```
      ordoA[i]:=ordo[jointM]-trunc(round(long[g]*sin(angleG[i]*pi/180)));
```

```
      if ((prisme[jointB]=90)or (prisme[jointB]=270)) then
```

```
        begin
```

```
          t:=t+1;
```

```
          angleG[t]:=angleG[i];abscA[t]:=abscA[i];ordoA[t]:=ordoA[i];
```

```
          abscB[t]:=absc[jointB];
```

```
          ordoB[t]:=ordo[i]+round(sqrt(sqr(long[k])-sqr(abscB[t]-abscA[t])));
```

```
          trigo(abscA[t],ordoA[t],abscB[t],ordoB[t]);
```

```
          angleK[t]:=zeta;
```

```
        end
```

```
      else
```

```
        begin
```

```
          interarcdroite(abscA[i],ordoA[i],absc[jointB],ordo[jointB],long[k],prisme[jointB]);
```

```
          if delta>=0 then
```

```
            begin
```

```
              t:=t+1;
```

```
              angleG[t]:=angleG[i];abscA[t]:=abscA[i];ordoA[t]:=ordoA[i];
```

```

    abscB[t]:=xi;
    ordoB[t]:=yi;
    trigo(abscA[t],ordoA[t],abscB[t],ordoB[t]);
    angleK[t]:=zeta;cou:=zeta+rp-angle[k];
    xcou[t]:=abscA[t]+round(rr*cos(pi/180*cou));
    ycou[t]:=ordoA[t]-round(rr*sin(pi/180*cou));
    end;
  end;
end;
end;

```

Procedure CALMCI (var xcou,ycou:tabl);

{elle fait les calculs relatifs à la courbe de bielle d'un MCI}

var

theta2,cou:real;

d,a,b,AQ2,e:real;

longK:array[1..360] of real;

begin

t:=0;

for i:=1 to 360 do

begin

angleG[i]:=angle[g]+i;

theta2:=angleG[i]-angle[f]+180; d:=long[f];

a:=long[g];b:=long[k];e:=long[h];

abscA[i]:=absc[jointM]+round(long[g]*cos(pi/180*theta2));

ordoA[i]:=ordo[jointM]-round(long[g]*sin(pi/180*theta2));

AQ2:=sqr(abscA[i]-absc[jointQ])+sqr(ordoA[i]-ordo[jointQ]);

if AQ2-e*e>=0 then

begin

longK[i]:=AQ2-e*e;longK[i]:=sqrt(longK[i]);

```

interarcs(abscA[i],ordoA[i],absc[joint0],ordo[joint0],longK[i],e);
if delta>=0 then
begin
t:=t+1;
angleG[t]:=angleG[i];abscA[t]:=abscA[i];ordoA[t]:=ordoA[i];longK[t]:=longK[i];
abscB[t]:=xi;ordoB[t]:=yi;
trigo(abscA[t],ordoA[t],abscB[t],ordoB[t]);
angleK[t]:=zeta;cou:=zeta+rp-angle[k];
trigo(abscB[t],ordoB[t],absc[joint0],ordo[joint0]);
angleH[t]:=zeta;
xcou[t]:=abscA[t]+round(rr*cos(pi/180*cou));
ycou[t]:=ordoA[t]-round(rr*sin(pi/180*cou));
end;
end;
end;
end;

```

```

Procedure schemaco(groupe:string;var theta:real);

```

```

var
code_retour:byte;fini:boolean;
vi1,vi2:word;
xcou,ycou:tabl;
begin
for i:=1 to 360 do
begin
xcou[i]:=0;ycou[i]:=0;
end;
detectgraph(driver,grmode);
initgraph(driver,grmode,'');
vi1:=getcolor;vi2:=getbkcolor;
x:=absc[jointA]; y:=ordo[jointA]; pas:=1; xa:=x; ya:=y; fini:=false;
outtextxy(1,65,' Entr: ');

```

```

outtextxy(1,80,' Droite');
outtextxy(1,135,' Fleches:');
outtextxy(1,150,' Deplacement');
outtextxy(1,170,' Pgdn:');
outtextxy(1,185,' Desincrement');
outtextxy(1,205,' Pgup:');
outtextxy(1,220,' Increment');
outtextxy(1,240,' F4:');
outtextxy(1,255,' Reprise');
outtextxy(1,275,' F10: Fin');
setlinestyle(0,0,3);
Rectangle(1,10,700,310);
setlinestyle(0,0,1);
rectangle(300,100,310,104);
  tail:=imagesize(300,100,310,105);
getmem(curs_im,tail);
getmem(ppim,tail);
getmem(memcurs_im,tail);
getimage((x-5),(y-2),(x+5),(y+2),curs_im^);
getimage((x-5),(y-2),(x+5),(y+2),memcurs_im^);
setviewport(300,100,310,104,false);
clearviewport;
setviewport(0,0,getmaxx,getmaxy,false);
ce:=getcolor;
REPEAT
  restituer(nondessin);
  p:=Getpixel(x,y);
  getimage((x-5),(y-2),(x+5),(y+2),ppim^);
  Repeat
    putpixel(x,y,0);putimage((x-5),(y-2),curs_im^,normalput);
    putpixel(x,y,ce);delay(2);
  until keypressed;
  putpixel(x,y,p);

```

```

putimage((x-5),(y-2),ppim^,normalput);
car:=upcase(readkey);
if car =#0 then
begin
  car:=readkey;
  case car of
    #75: if x>(100+pas) then
      begin
        x:=x-pas;
      end;
    #77: if x<(695-pas) then
      begin
        x:=x+pas;
      end;
    #80: if y<(305-pas) then
      begin
        y:=y+pas;
      end;
    #72: if y>(12+pas) then
      begin
        y:=y-pas;
      end;
    #73: if pas<70 then pas:=pas+10;
    #81: if pas>1 then pas:=pas-10;
    #68: fini:=true;
    #62:begin
      setviewport(105,12,695,305,false);
      clearviewport;
      setviewport(0,0,getmaxx,getmaxy,false);
      restituer(nomdessin);

```




```

var
  groupe,entree,sortie,inter,fixe:string;
  x1,a,b,c,d,mu,theta2,eff:real;
begin
  appeler;
  charger(nomdessin);
  if ((s<>4) or (j<>4)) then
    begin
      clrscr;
      gotoxy(20,12);writeln('ce dessin n''a pas 4 membrures!');
      delay(10000);
    end
  else
    begin
      groupe:='M4M';
      clrscr;
      gotoxy( 11,4);write('les membrures sont :');
      for i:=1 to 4 do
        begin
          gotoxy(5+6*i,6);write(seg[i],',');
        end;
      gotoxy(20,10);write('membrure d''entrée ? ');
      gotoxy(20,12);write('membrure intermédiaire ? ');
      gotoxy(20,14);write('membrure de sortie ? ');
      gotoxy(20,16);write('membrure fixe ? ');
      gotoxy(45,10);lire(entree);
      gotoxy(45,12);lire(inter);
      gotoxy(45,14);lire(sortie);
      gotoxy(45,16);lire(fixe);
      for i:=1 to 4 do
        begin
          if seg[i]=entree then g:=i;
          if seg[i]=inter then k:=i;

```

```

    if seg[i]=sortie then h:=i;
    if seg[i]=fixe then f:=i;
end;
a:=long[g];b:=long[k];c:=long[h];d:=long[f];
for i:=1 to 4 do
begin
    if joint[i]=copy(seg[k],1,1) then jointA:=i;
    if joint[i]=copy(seg[k],2,1) then jointB:=i;
    if joint[i]=copy(seg[f],1,1) then jointQ:=i;
    if joint[i]=copy(seg[f],2,1) then jointM:=i;
end;
schemaco(groupe,theta);
end;
end;

```

Procedure COUMC;

(elle réalise le couplage à un manivelle coulisseau)

var

```

xn,yn:longint;
a0:string[1];
temp:integer;
an,e:real;groupe:string;
a,b,x1,mu,theta2:real;

```

begin

```
temp:=0;
```

```
for i:=1 to 3 do
```

```
begin
```

```
    if typejoint[i]='P' then temp:=temp+1;
```

```
end;
```

```
if (temp<>1) then
```

```
begin
```

```
    clrscr;
```

```

gotoXY(20,12);write('vous n'avez pas un manivelle coulisseau');
end
else
begin
groupe:='MC';
for i:=1 to 3 do if typejoint[i]='P' then jointB:=i;
for i:=1 to 2 do
begin
if copy(seg[i],1,1)=joint[jointB] then
begin
a0:=copy(seg[i],2,1);
k:=i;
end;
if copy(seg[i],2,1)=joint[jointB] then
begin
a0:=copy(seg[i],1,1);
k:=i;
end;
end;
for i:=1 to 3 do if joint[i]=a0 then jointA:=i;
for i:=1 to 3 do if ((i<>jointA) and (i<>jointB)) then jointM:=i;
if k=1 then g:=2;
if k=2 then g:=1;
xn:=absc[jointM];a:=long[g];b:=angle[k];
an:=(1.0*xn-1.0*absc[jointB])*tg(prisme[jointB]);
yn:=ordo[jointB]+trunc(an);
e:=trunc(round((yn-ordo[jointM])*sin(90-prisme[jointB])));
schemaco(groupe,theta);
end;
end;

```



```

Procedure COUMCI;
  {elle réalise le couplage à un manivelle coulisseau inversé}
var
  e,a,b,d,alpha,beta,mu,theta2:real;
  pro:integer;groupe:string;
  A02:real;
  longK:array[1..360] of real;
begin
  if veri2_MCI(nomdessin)=false then
    begin
      clrscr;
      gotoXY(20,12);write('vous n''avez pas le bon mécanisme');
    end
  else
    begin
      groupe:='MCI';
      schemaco(groupe,theta);
    end;
end;

```

```

Procedure menucou;
  {elle donne le menu secondaire pour le couplage à un mécanisme}
var
  choix:integer;
  ccc:char;
begin
  clrscr;
  gotoXY(5,5); write('1:COUPLAGE A UN MECANISME A QUATRE MEMBRURES SANS COULISSEAU');
  gotoXY(5,7); write('2:COUPLAGE A UN MECANISME A QUATRE MEMBRURES AVEC COULISSEAU');
  gotoXY(50,12);write('VOTRE CHOIX ? ');read(choix);
  case choix of
    1:begin

```

```

    COUM4M;
end;
2:begin
    appeler;
    charger(nomdessin);
    if (( s=2)and(j=3)) then COUMC
    else
        begin
            if(((s=4)and(j=4))or((s=3)and(j=3))) then COUMC1
            else
                begin
                    clrscr;
                    gotoXY(10,12);write('vous n'avez pas le bon mécanisme');
                end;
            end;
        end;
    end;
end;
clrscr;
gotoxy(40,23);write('appuyer une touche pour continuer');
ccc:=readkey;
end;

```

```

Begin
    clrscr;
    menucou;
End.

```

MODULE 12: SIMULATION DE MOUVEMENT (simulati)

```
Uses crt,graph,unit0,progene,printer;
```

```
var  
  vis:integer;  
  ang,vit:real;  
  col1,col2:word;  
Procedure SIMM4M;  
  (elle simule le mouvement d'un M4M)  
  var  
    entree,sortie,inter,fixe:string;  
    x1,a,b,c,d,mu,theta2,eff:real;  
  begin  
    appeler;  
    charger(nomdessin);  
    if ((s<>4) or (j<>4)) then  
      begin  
        clrscr;  
        gotoxy(20,12);writeln('ce dessin n'a pas 4 membrures');  
        delay(10000);  
      end  
    else  
      begin  
        clrscr;  
        gotoxy( 11,4);write('les membrures sont :');  
        for i:=1 to 4 do  
          begin  
            gotoxy(5+6*i,6);write(seg[i],',');  
          end;  
        gotoxy(20,10);write('membrane d'entrée ? ');  
        gotoxy(20,12);write('membrane intermédiaire ? ');
```

```

gotoXY(20,14);write('membrure de sortie ? ');
gotoXY(20,16);write('membrure fixe ? ');
gotoxy(20,18);write('vitesse de rotation de la membrure motrice ? ');
gotoXY(45,10);lire(entree);
gotoXY(45,12);lire(inter);
gotoXY(45,14);lire(sortie);
gotoXY(45,16);lire(fixe);
gotoxy(65,18);read(vit);
if vit=0 then
begin
  clrscr;gotoxy(12,14);write(' Il n''y a pas de mouvement si la membrure motrice est fixe');
  delay(2000);halt;
end;

for i:=1 to 4 do
begin
  if seg[i]=entree then g:=i;
  if seg[i]=inter then k:=i;
  if seg[i]=sortie then h:=i;
  if seg[i]=fixe then f:=i;
end;
a:=long[g];b:=long[k];c:=long[h];d:=long[f];
for i:=1 to 4 do
begin
  if joint[i]=copy(seg[k],1,1) then jointA:=i;
  if joint[i]=copy(seg[k],2,1) then jointB:=i;
  if joint[i]=copy(seg[f],1,1) then jointQ:=i;
  if joint[i]=copy(seg[f],2,1) then jointM:=i;
end;
t:=0;clrscr;gotoxy(10,12);write('un instant ');
ang:=angle[g];vis:=round(vit);
for i:=1 to abs(round(360/vit)) do
begin

```



```

ang:=ang+vis;
  abscA[i]:=absc[jointM]+round(long[g]*cos(ang*pi/180));
  ordoA[i]:=ordo[jointM]-round(long[g]*sin(ang*pi/180));
  interarcs(abscA[i],ordoA[i],absc[jointQ],ordo[jointQ],long[k],long[h]);
  if delta>=0 then
    begin
      t:=t+1;angleG[t]:=ang;abscA[t]:=abscA[i];ordoA[t]:=ordoA[i];

      abscB[t]:=xi; ordoB[t]:=yi;trigo(abscA[t],ordoA[t],abscB[t],ordoB[t]);
      angleK[t]:=zeta;
      trigo(abscB[t],ordoB[t],absc[jointQ],ordo[jointQ]);
      angleH[t]:=zeta;
    end
  else
    begin
      vis:=-vis;ang:=ang+vis;
      abscA[i]:=absc[jointM]+round(long[g]*cos(ang*pi/180));
      ordoA[i]:=ordo[jointM]-round(long[g]*sin(ang*pi/180));
      interarcs(abscA[i],ordoA[i],absc[jointQ],ordo[jointQ],long[k],long[h]);
      if delta>=0 then
        begin
          t:=t+1;angleG[t]:=ang;abscA[t]:=abscA[i];ordoA[t]:=ordoA[i];

          abscB[t]:=xi; ordoB[t]:=yi;trigo(abscA[t],ordoA[t],abscB[t],ordoB[t]);
          angleK[t]:=zeta;
          trigo(abscB[t],ordoB[t],absc[jointQ],ordo[jointQ]);
          angleH[t]:=zeta;
        end;
      end;
    end;
  end;
end;
end;

cadre;

```

```

col1:=getcolor;
col2:=getbkcolor;
for i:=1 to t do
begin
setcolor(col1);
visualisation_M4M
(absc[jointM],ordo[jointM],abscA[i],ordoA[i],abscB[i],ordoB[i],absc[jointQ],ordo[jointQ]);
setcolor(col2);
visualisation_M4M
(absc[jointM],ordo[jointM],abscA[i],ordoA[i],abscB[i],ordoB[i],absc[jointQ],ordo[jointQ]);
end;
closegraph;
end;
end;

```

Procedure SIMMC;

{elle simule le mouvement d'un manivelle coulisseau}

var

xn,yn:longint;

a0:string[1];

temp:integer;

an,e:real;

a,b,x1,mu,theta2:real;

begin

temp:=0;

for i:=1 to 3 do

begin

if typejoint[i]='P' then temp:=temp+1;

end;

if (temp<>1) then

begin

clrscr;

```

gotoXY(20,12);write('vous n'avez pas un manivelle coulisseau');
end
else
begin
  clrscr;gotoxy(12,14);write('vitesse de rotation de la membrure motrice ?');
  gotoxy(70,14);read(vit);vis:=round(vit);
  if vit=0 then
  begin
    clrscr;gotoxy(12,14);write(' il n'y a pas de mouvement si la membrure motrice est fixe');
    delay(2000);halt;
  end;

  for i:=1 to 3 do if typejoint[i]='P' then jointB:=i;
  for i:=1 to 2 do
  begin
    if copy(seg[i],1,1)=joint[jointB] then
    begin
      a0:=copy(seg[i],2,1);
      k:=i;
      end;
    if copy(seg[i],2,1)=joint[jointB] then
    begin
      a0:=copy(seg[i],1,1);
      k:=i;
      end;
    end;
  for i:=1 to 3 do if joint[i]=a0 then jointA:=i;
  for i:=1 to 3 do if ((i<>jointA) and (i<>jointB)) then jointM:=i;
  if k=1 then g:=2;
  if k=2 then g:=1;
  xn:=absc[jointM];a:=long[g];b:=angle[k];
  an:=(1.0*xn-1.0*absc[jointB])*tg(prisme[jointB]);
  yn:=ordol[jointB]+trunc(an);

```

```

e:=trunc(round((yn-ordo[jointM])*sin(90-prisme[jointB])));
t:=0;clrscr;gotoxy(10,12);write('un instant ');
ang:=angle[g];
for i:=1 to abs(round(360/vit)) do
begin
  ang:=ang+vis;
  angleG[i]:=ang;if angleG[i]>360 then angleG[i]:=ang-360;
  abscA[i]:=absc[jointM]+trunc(round(long[g]*cos(angleG[i]*pi/180)));
  ordoA[i]:=ordo[jointM]-trunc(round(long[g]*sin(angleG[i]*pi/180)));
  if ((prisme[jointB]=90)or (prisme[jointB]=270)) then
  begin
    t:=t+1;
    angleG[t]:=angleG[i];abscA[t]:=abscA[i];ordoA[t]:=ordoA[i];
    abscB[t]:=absc[jointB];
    if (sqr(long[k])-sqr(abscB[t]-abscA[t])) >=0 then
    begin
      ordoB[t]:=ordoA[i]+round(sqrt(sqr(long[k])-sqr(abscB[t]-abscA[t])));
      trigo(abscA[t],ordoA[t],abscB[t],ordoB[t]);
      angleK[t]:=zeta;
    end;
  end
else
begin
  interarcdroite(abscA[i],ordoA[i],absc[jointB],ordo[jointB],long[k],prisme[jointB]);
  if delta>=0 then
  begin
    t:=t+1;
    angleG[t]:=angleG[i];abscA[t]:=abscA[i];ordoA[t]:=ordoA[i];
    abscB[t]:=xi;
    ordoB[t]:=yi;
    trigo(abscA[t],ordoA[t],abscB[t],ordoB[t]);
    angleK[t]:=zeta;
  end
end

```

```

else
begin
vis:=-vis;
ang:=ang+vis;
t:=t+1;
angleG[t]:=ang;abscA[t]:=abscA[i];ordoA[t]:=ordoA[i];
abscB[t]:=absc[jointB];
if sqrt(long[k])-sqrt(abscB[t]-abscA[t]) >=0 then
begin
ordoB[t]:=ordoA[i]+round(sqrt(sqrt(long[k])-sqrt(abscB[t]-abscA[t])));
trigo(abscA[t],ordoA[t],abscB[t],ordoB[t]);
angleK[t]:=zeta;
interarcdroite(abscA[i],ordoA[i],absc[jointB],ordo[jointB],long[k],prisme[jointB]);
if delta>=0 then
begin
angleG[t]:=angleG[i];abscA[t]:=abscA[i];ordoA[t]:=ordoA[i];
abscB[t]:=xi;
ordoB[t]:=yi;
trigo(abscA[t],ordoA[t],abscB[t],ordoB[t]);
angleK[t]:=zeta;
end;
end;
end;
end;
end;
cadre;col1:=getcolor;col2:=getbkcolor;
for i:=1 to t do
begin
setcolor(col1);
visualisation_MC(absc[jointM],ordo[jointM],abscA[i],ordoA[i],abscB[i],ordoB[i]);
setcolor(col2);
visualisation_MC(absc[jointM],ordo[jointM],abscA[i],ordoA[i],abscB[i],ordoB[i]);

```

```

    end;
    closegraph;
  end;
end;

```

Procedure SIMMCI;

{elle simule le mouvement d'un manivelle coulisseau inversé}

var

e,a,b,d,alpha,beta,mu,theta2:real;

pro:integer;

AQ2:real;

longK:array[1..360] of real;

begin

if veri2_MCI(nomdessin)=false then

begin

clrscr;

gotoxy(20,12);write('vous n'avez pas le bon mécanisme');

end

else

begin

clrscr;gotoxy(5,12);write('vitesse de la membrure motrice ?');

gotoxy(70,12);read(vit);vis:=round(vit);

if vit=0 then

begin

clrscr;gotoxy(12,14);write(' Il n'y a pas de mouvement si la membrure motrice est fixe');

delay(2000);halt;

end;

clrscr;gotoxy(10,12);write('un instant ');

t:=0; ang:=angle[g];

for i:=1 to abs(round(360/vit)) do

begin

```

ang:=ang+vis;
angleG[i]:=ang;
theta2:=angleG[i]-angle[f]+180; d:=long[f];
a:=long[g];b:=long[k];e:=long[h];
abscA[i]:=absc[jointM]+round(long[g]*cos(pi/180*theta2));
ordoA[i]:=ordo[jointM]-round(long[g]*sin(pi/180*theta2));
AQ2:=sqr(abscA[i]-absc[jointQ])+sqr(ordoA[i]-ordo[jointQ]);
if AQ2-e*e>=0 then
begin
longK[i]:=AQ2-e*e;longK[i]:=sqrt(longK[i]);
interarcs(abscA[i],ordoA[i],absc[jointQ],ordo[jointQ],longK[i],e);
if delta>=0 then
begin
t:=t+1;
angleG[t]:=angleG[i];abscA[t]:=abscA[i];ordoA[t]:=ordoA[i];longK[t]:=longK[i];
abscB[t]:=xi;ordoB[t]:=yi;
trigo(abscA[t],ordoA[t],abscB[t],ordoB[t]);
angleK[t]:=zeta;
trigo(abscB[t],ordoB[t],absc[jointQ],ordo[jointQ]);
angleH[t]:=zeta;
end
else
begin
vis:=-vis;
ang:=ang+vis;
t:=t+1;
angleG[t]:=ang;
theta2:=angleG[t]-angle[f]+180; d:=long[f];
a:=long[g];b:=long[k];e:=long[h];
abscA[i]:=absc[jointM]+round(long[g]*cos(pi/180*theta2));
ordoA[i]:=ordo[jointM]-round(long[g]*sin(pi/180*theta2));
AQ2:=sqr(abscA[i]-absc[jointQ])+sqr(ordoA[i]-ordo[jointQ]);
if AQ2-e*e>=0 then

```

```

begin
  longK[i]:=A02-e*e;longK[i]:=sqrt(longK[i]);
  interarcs(abscA[i],ordoA[i],absc[jointQ],ordo[jointQ],longK[i],e);
  if delta>=0 then
    begin
      angleG[t]:=angleG[i];abscA[t]:=abscA[i];ordoA[t]:=ordoA[i];longK[t]:=longK[i];
      abscB[t]:=xi;ordoB[t]:=yi;
      trigo(abscA[t],ordoA[t],abscB[t],ordoB[t]);
      angleK[t]:=zeta;
      trigo(abscB[t],ordoB[t],absc[jointQ],ordo[jointQ]);
      angleH[t]:=zeta;
    end;
  end;
end;
end;
end;
end;
clrscr;
cadre;col1:=getcolor;col2:=getbkcolor;
for i:=1 to t do
  begin
    prisme[jointB]:=angleK[i];
    if v=jointA then
      begin
        xjoint:=abscA[i];yjoint:=ordoA[i];
      end;
    if v=jointB then
      begin
        xjoint:=abscB[i];yjoint:=ordoB[i];
      end;
    if v=jointQ then
      begin
        xjoint:=absc[jointQ];yjoint:=ordo[jointQ];
      end;
  end;

```



```

        setcolor(col1);
        visualisationMCI
        (absc[jointM],ordo[jointM],abscA[i],ordoA[i],abscB[i],ordoB[i],absc[jointQ],ordo[jointQ]);
        setcolor(col2);
        visualisationMCI
        (absc[jointM],ordo[jointM],abscA[i],ordoA[i],abscB[i],ordoB[i],absc[jointQ],ordo[jointQ]);

    end;

    closegraph;

end;
end;

```

Procedure menusim;

(elle donne le menu secondaire pour la simulation du mouvement)

var

choix:integer;

ccc:char;

begin

clrscr;

gotoXY(5,5); write('1:SIMULATION POUR MECANISME A QUATRE MEMBRURES SANS COULISSEAU');

gotoXY(5,7); write('2:SIMULATION POUR MECANISME A QUATRE MEMBRURES AVEC COULISSEAU');

gotoXY(5,9); write('0:pour sortir');

gotoXY(50,12);write('VOTRE CHOIX ? ');read(choix);

case choix of

1:begin

 SIMM4M;

end;

2:begin

 appeler;

```
charger(nomdessin);
if (( s=2)and(j=3)) then SIMMC
else
begin
if(((s=4)and(j=4))or((s=3)and(j=3))) then SIMMCI
else
begin
clrscr;
gotoXY(10,12);write('vous n'avez pas le bon mécanisme!');
end;
end;
end;
end;
clrscr;gotoxy(45,23);write('appuyer une touche pour continuer!');
ccc:=readkey;
end;
```

```
Begin
clrscr;
menusim;
End.
```

MODULE 13: PROGRAMME MAITRE (SAMP1)

Program Menuprog;

(\$M \$4000,0,\$4000)

Uses crt,dos,GerEcran;

Type

String14=String[14];

Const

FICHEXE: Array[1..11] of string14=('schema.exe','visualis.exe',
'mobilite.exe','typeM4M.exe','critique.exe','Transmis.exe',
'instant.exe','cinemati.exe','couplage.exe','simulati.exe',
'trtrtrtrtt.exe');

var

x:byte; code:string;

Function Existe(Nomfich:string14):boolean;

var fich:text;

begin

assign(fich,Nomfich);

{\$1-} Reset(fich); {\$1+}

if ioread=0 then

begin

existe:=True;

close(fich);

end

else existe:=false;

end;

Procedure ExecFile(Nomfich:String14;var CodeRetour:String);

var ok:boolean;

disq:char;

begin

```

ok:=false;
if ((x=1)or(x=2)or(x=3)or(x=4)or(x=5)) then
begin
gotoxy(15,25);clreol;
repeat
gotoxy(15,25);write(' PLACER LA DISQUETTE #1 EN A ET APPUYER ESC');
disq:=readkey;
until disq=#27;
disq:=' ';
end;
if ((x=6)or(x=7)or(x=8)or(x=9)or(x=10)) then
begin
gotoxy(15,25);clreol;
repeat
gotoxy(15,25);write(' PLACER LA DISQUETTE #2 EN A ET APPUYER ESC');
disq:=readkey;
until disq=#27;
disq:=' ';
end;

if existe(NomFich) then
Begin
    Exec(Nomfich,'');
    CodeRetour:='';
end
else
codeRetour:=NomFich+' Fichier inexistant ou entraineur non valide ';
end;

Procedure Texte;
begin
textcolor(7); textbackground(0);
gotoxy(25,6);write(' D : DESSIN ');

```

```

gotoxy(25,7);write(' V : VISUALISATION      ');
gotoxy(25,8);write(' L : DEGRE DE LIBERTE      ');
gotoxy(25,9);write(' M : TYPE DE TRANSMISSION  ');
gotoxy(25,10);write(' C : POSITIONS CRITIQUES  ');
gotoxy(25,11);write(' T : ANGLE DE TRANSMISSION ');
gotoxy(25,12);write(' I : POSITION INSTANTANEE  ');
gotoxy(25,13);write(' A : ANALYSE CINEMATIQUE  ');
gotoxy(25,14);write(' E : ETUDE DE COUPLAGE    ');
gotoxy(25,15);write(' S : SIMULATION           ');
gotoxy(25,16);write(' Q : QUITTER              ');

Normvideo;

end;

Procédure Menu(var y:byte);
var i:byte;
    car:char; clsp,fini,choisi:boolean;
begin
    Textcolor(0); clrscr;
    normvideo;
    SetTimeColorAndPosition(45,1,0,7);
    Remplissage:=false;
    Doublecadre:=true;
    Cadrage(20,4,60,18,0,7);
    TEXTE;
    Remplissage:=true;
    Cadrage(10,20,70,24,0,7);
    textcolor(0); textbackground(7);
    gotoxy(20,21); write('      ECOLE POLYTECHNIQUE DE THIES ');
    gotoxy(20,22); write('      DEPARTEMENT DU GENIE MECANIQUE ');
    gotoxy(20,23); write(' PROJET SAMP1 PRESENTE PAR AVODAGBE JEROME ');
    Normvideo; {textcolor(0+128); textbackground(7);
    gotoxy(47,23); write('AVODAGBE JEROME ');}

```

```

fini:=false; choisi:=false;
Repeat
  if choisi then fini:=true;
  if clsp then
    case car of
      #72: if y>1 then y:=y-1
          else y:=11;
      #80: if y<11 then y:=y+1
          else y:=1;
    end
  else
    begin
      case car of
        'D': begin y:=1;choisi:=true; end;
        'V': begin y:=2;choisi:=true; end;
        'L': begin y:=3;choisi:=true; end;
        'M': begin y:=4;choisi:=true; end;
        'C': begin y:=5;choisi:=true; end;
        'T': begin y:=6;choisi:=true; end;
        'I': begin y:=7;choisi:=true; end;
        'A': begin y:=8;choisi:=true; end;
        'E': begin y:=9;choisi:=true; end;
        'S': begin y:=10;choisi:=true; end;
        'O': begin y:=11;choisi:=true; end;
        #13: begin choisi:=true; end;
      end;
    end;
  end;

If y=1 then
begin
  texte;
  textcolor(0); textbackground(7);
  gotoxy(25,6);write(' D : DESSIN ');

```

```

Normvideo;
gotoxy(10,3);clreol;
textcolor(7); textbackground(0);
gotoxy(10,3);write(' Faire le schéma du mécanisme ');
end;
If y=2 then
begin
texte;
textcolor(0); textbackground(7);
gotoxy(25,7);write(' V : VISUALISATION ');
Normvideo;
gotoxy(10,3);clreol;
textcolor(7); textbackground(0);
gotoxy(10,3);write(' Revoir le schéma du mécanisme et ses paramètres');
end;
If y=3 then
begin
texte;
textcolor(0); textbackground(7);
gotoxy(25,8);write(' L : DEGRE DE LIBERTE ');
Normvideo;
gotoxy(10,3);clreol;
textcolor(7); textbackground(0);
gotoxy(10,3);write(' Etudier la mobilité ');
end;
If y=4 then
begin
texte;
textcolor(0); textbackground(7);
gotoxy(25,9);write(' M : TYPE DE MECANISME ');
Normvideo;
gotoxy(10,3);clreol;
textcolor(7); textbackground(0);

```

```

    textcolor(7); textbackground(0);
    gotoxy(10,3);write(' Faire le schéma du mécanisme ');
end;
If y=2 then
begin
    texte;
    textcolor(0); textbackground(7);
    gotoxy(25,7);write(' V : VISUALISATION ');
    Normvideo;
    gotoxy(10,3);clreol;
    textcolor(7); textbackground(0);
    gotoxy(10,3);write(' Revoir le schéma du mécanisme et ses paramètres');
end;
If y=3 then
begin
    texte;
    textcolor(0); textbackground(7);
    gotoxy(25,8);write(' L : DEGRE DE LIBERTE ');
    Normvideo;
    gotoxy(10,3);clreol;
    textcolor(7); textbackground(0);
    gotoxy(10,3);write(' Etudier la mobilité ');
end;
If y=4 then
begin
    texte;
    textcolor(0); textbackground(7);
    gotoxy(25,9);write(' M : TYPE DE MECANISME ');
    Normvideo;
    gotoxy(10,3);clreol;
    textcolor(7); textbackground(0);
    gotoxy(10,3);write(' Connaître la nature du mécanisme ');
end;

```



```

If y=5 then
begin
  texte;
  textcolor(0); textbackground(7);
  gotoxy(25,10);write(' C : POSITIONS CRITIQUES ');
  Normvideo;
  gotoxy(10,3);clreol;
  textcolor(7); textbackground(0);
  gotoxy(10,3);write(' Déterminer les points morts et les positions limites ');
end;

If y=6 then
begin
  texte;
  textcolor(0); textbackground(7);
  gotoxy(25,11);write(' T : ANGLE DE TRANSMISSION ');
  Normvideo;
  gotoxy(10,3);clreol;
  textcolor(7); textbackground(0);
  gotoxy(10,3);write(' Variation des angles de transmission au cours du mouvement ');
end;

If y=7 then
begin
  texte;
  textcolor(0); textbackground(7);
  gotoxy(25,12);write(' I : POSITION INSTANTANEE ');
  Normvideo;
  gotoxy(10,3);clreol;
  textcolor(7); textbackground(0);
  gotoxy(10,3);write(' Avoir les paramètres instantanés ');
end;

If y=8 then
begin
  texte;

```

```

textcolor(0); textbackground(7);
gotoxy(25,13);write(' A : ANALYSE CINEMATIQUE ');
Normvideo;
gotoxy(10,3);clreol;
textcolor(7); textbackground(0);
gotoxy(10,3);write(' Etudier la position, la vitesse et l''accélération ');
end;
If y=9 then
begin
texte;
textcolor(0); textbackground(7);
gotoxy(25,14);write(' E : ETUDE DE COUPLAGE ');
Normvideo;
gotoxy(10,3);clreol;
textcolor(7); textbackground(0);
gotoxy(10,3);write(' Analyser le couplage à d''autres mécanismes ');
end;
If y=10 then
begin
texte;
textcolor(0); textbackground(7);
gotoxy(25,15);write(' S : SIMULATION ');
Normvideo;
gotoxy(10,3);clreol;
textcolor(7); textbackground(0);
gotoxy(10,3);write(' Simuler le mouvement du mécanisme ');
end;
If y=11 then
begin
texte;
textcolor(0); textbackground(7);
gotoxy(25,16);write(' Q : QUITTER ');
Normvideo;

```

```

        gotoxy(10,3);clreol;
        textcolor(7); textbackground(0);
        gotoxy(10,3);write(' Quitter le logiciel SAMP1 ');
    end;
    if not(choisi) then Car:=Uppcase(readcar(clsp));
until fini ;

end;

Procedure Directeur(var x:byte);
begin
    Menu(x);
    if x=11 then
    begin
        clrscr;
        exit;
    end
    else
    begin
        Execfile(FICHEXE [x],code);
        if code<> '' then
        begin
            gotoxy(5,25);write(' ERREUR ==> ',code);
            Sound(1500); delay(100); nosound;
            Sound(1000); delay(100); nosound;
            delay(1500);
            gotoxy(10,25); clreol;
        end;
        Directeur(x);
    end;
end;
end;

```

```
begin
```

```
  x:=1;
```

```
  directeur(x);
```

```
end.
```