

Attributs animables

- Les scènes tridimensionnelles se composent de 3 types d'entités évoluant au cours du temps:
 - Objets
 - Caméras
 - Lumières

Attributs animables

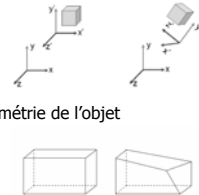
- Pour les objets:
 - position (automobile)
 - orientation (bras de robot)
 - taille (croissance)
 - forme (nuage, coeur humain)
 - couleur (plaque de cuisinière électrique qui chauffe)
 - transparence (simulation de brouillard évoluant)
 - coordonnées textures

Attributs animables

- Pour les caméras:
 - position de l'observateur (simulation de vol)
 - point d'intérêt
 - angle de vue (zoom in)
- Pour les sources de lumières:
 - intensité de la lumière
 - couleur d'une composante (diffuse, ambiante, etc.)
 - position de la lumière (simulation de phares de voiture)

Types d'animations

- Sur le plan géométrique
 - Mouvement rigide
 - Mise à jour continue de la position et de l'orientation d'objet dans la scène
 - Déformation
 - Mise à jour continue de la géométrie de l'objet



Types d'animations

- Natures des objets animables
 - Objets rigides
 - libres : chute de pierres, billard, etc.
 - articulés : robots industriels, mécanismes, etc.
 - Objets déformables : corps biologiques, tissus, etc.
 - Déformation locale
 - Déformation globale
 - Objets flexibles
 - Déformation permettant d'articuler certaines parties de l'objet
 - Systèmes de particules : fumé, neige, etc.
 - ...

Types d'animations

- L'animation peut être :
 - Temps réel vs non-temps réel
 - Calcul & génération
 - Visualisation doit être en temps réel
 - Interactive vs non-interactive
 - Apport d'une perturbation externe
 - L'opérateur « externe » est considéré comme élément de la scène et est modélisé comme entité géométrique et dynamique dans la scène

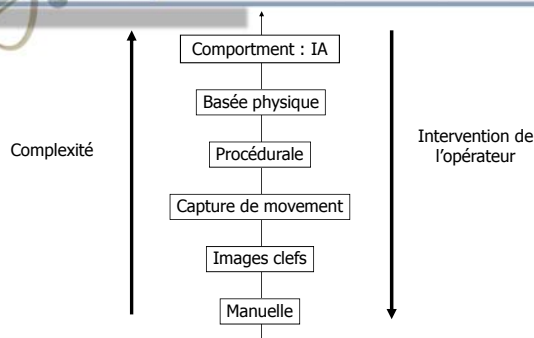
Types d'animations

- Cinématique vs dynamique
 - Animation cinématique
 - déterminée explicitement par l'animateur. [modèle descriptif]
 - fonction mathématique, interpolation de valeurs « clé », etc.
 - Animation dynamique
 - solution de l'équation du mouvement. [modèle générateur]
 - lois de la mécanique, thermodynamique, etc.
- Les animations cinématiques et dynamiques peuvent être directes ou inverses
 - Dynamique inverse
 - balançoire : on impose les rotations des jambes

Types d'animations

- Les modèles de contrôle du mouvement peuvent être classés en trois grandes familles générales :
 - Basée sur des modèles descriptifs ou phénoménologiques :
 - ils sont utilisés pour reproduire uniquement les effets (mouvement, déformation), sans aucune connaissance a priori sur les causes qui pourraient les avoir produits. Ils décrivent la cinématique des phénomènes dynamiques.
 - Basée sur des modèles générateurs ou fondés sur la physique :
 - ils décrivent les causes capables de produire un effet. Par exemple, les modèles utilisant la mécanique font partie des modèles générateurs.
 - Basée sur des modèles comportementaux :
 - ils simulent les organismes et les êtres vivants, leurs actions et leurs réponses à des stimulations extérieures.

Hierarchie des processus d'animation



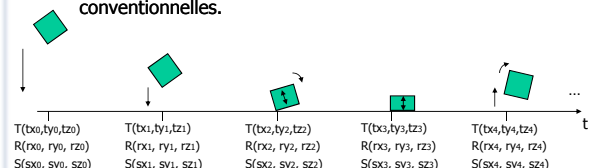
Techniques d'animation

- Manuelle
 - Fastidieux
- Interpolation entre positions clés
 - Très souple, moins fastidieux
- Capture de mouvements
 - Grande qualité mais spécifique
- Modèles générateurs
 - Automatique mais complexe
- Une animation peut être la combinaison de plusieurs de ces techniques

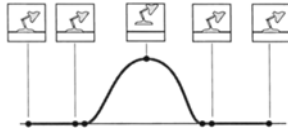
Animation manuelle

Animation manuelle

- Spécification, itération par itération, de l'ensemble des transformations et des modifications.
- Intérêts
 - Faibles coûts de développements
 - Impossibilité de générer l'animation avec les méthodes conventionnelles.



Animation par interpolation entre positions clés



Animation par interpolation entre positions clés

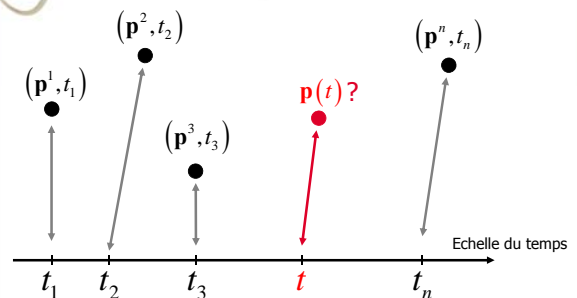
- Interpolation d'étapes « clés » : Keyframing
 - Le concepteur définit les étapes clés et le programme génère les configurations intermédiaires par interpolation.
- Une étape est une paire composée :
 - d'un instant
 - d'une valeur scalaire ou vectorielle
- L'interpolation consiste à évaluer une valeur intermédiaire cohérente entre deux étapes clés.

Animation par interpolation entre positions clés

- Ce qui est interpolable
 - Position
 - Orientation
 - Echelle
 - Forme : morphing
 - Couleur
 - Transparence
 - etc.

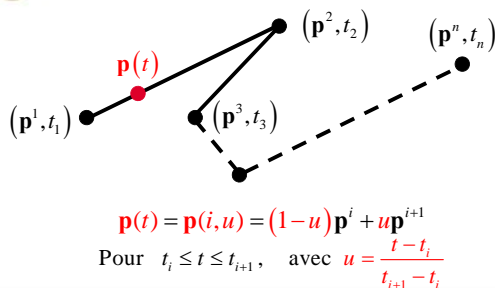
Animation par interpolation entre positions clés

- Exemple : positions « clés »



Animation par interpolation entre positions clés

- Interpolation linéaire des positions



Animation par interpolation entre positions clés

- Exemple :

- L'articulation d'un bras de robot est caractérisée par un angle a variant au cours du temps t ; on fixe les valeurs suivantes:

$$t=0 \quad a=10$$

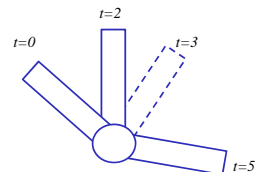
$$t=2 \quad a=20$$

$$t=5 \quad a=45$$

$$t=8 \quad a=100$$

$$t=3 \Rightarrow u = \frac{3-2}{5-2} = \frac{1}{3}$$

$$a(3) = a\left(1, \frac{1}{3}\right) = \left(1 - \frac{1}{3}\right) \cdot 20 + \frac{1}{3} \cdot 45 = 28.33$$

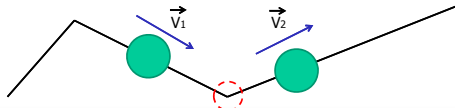


Animation par interpolation entre positions clés

- Paramétrage simple, rapide, uniforme.

$$p(i, 0) = p^i, \quad p(i, 1) = p^{i+1} \quad \text{et} \quad p(i, 1/2) = \frac{p^i + p^{i+1}}{2}$$

- L'interpolation linéaire occasionne des discontinuités au niveau de la dérivée, donc de la vitesse et par conséquent, l'animation va être saccadée.
- Changement de direction brutal à chaque étape.

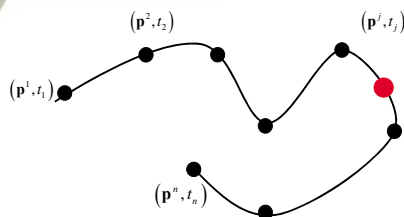


Animation par interpolation entre positions clés

- Une animation fluide nécessite
 - Continuité C^0 , C^1 et C^2

Animation par interpolation entre positions clés

- Interpolation polynomiale : Interpolation globale



$$P(t) = p_0 P_0(t) + \dots + p_i P_i(t) + \dots + p_n P_n(t) \quad \begin{cases} P_i(x) \text{ polynômes} \\ p_i(x) \text{ pts clefs} \end{cases}$$

Animation par interpolation entre positions clés

- Interpolation polynomiale : Interpolation globale

- Soient $n+1$ points distincts t_i réels et $n+1$ réels p_i
 - il existe un unique polynôme $p \in P_n$ tel que $p(t_i) = p_i$ pour $i = 0 \text{ à } n$

- Polynômes d'interpolation de Lagrange

$$p(t) = \sum_{i=0}^n L_{i,n}(t) p_i \quad \text{où} \quad L_{i,n}(t) = \prod_{\substack{1 \leq j \leq n \\ j \neq i}} \frac{t - t_j}{t_i - t_j}$$

- L_i polynôme de Lagrange

$$P(t_i) = y_0 P_0(t_i) + \dots + y_i P_i(t_i) + \dots + y_n P_n(t_i)$$

- Propriétés de L_i

- $L_i(t_i) = 1$
- $L_i(t_j) = 0 \quad (j \neq i)$

$$\begin{matrix} \downarrow & & \downarrow & & \downarrow \\ 0 & & 1 & & 0 \end{matrix}$$

Interpolation polynomiale

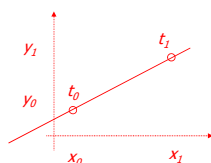
- Exemple avec $n=1$

- Nous disposons de deux points clefs $p_0(x_0, y_0)$ et $p_1(x_1, y_1)$

$$p(t) = p_0 \frac{t - t_1}{t_0 - t_1} + p_1 \frac{t - t_0}{t_1 - t_0}$$

$$x(t) = x_0 \frac{t - t_1}{t_0 - t_1} + x_1 \frac{t - t_0}{t_1 - t_0}$$

$$y(t) = y_0 \frac{t - t_1}{t_0 - t_1} + y_1 \frac{t - t_0}{t_1 - t_0}$$



Interpolation polynomiale

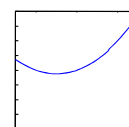
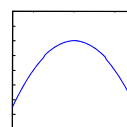
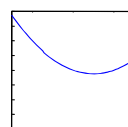
- Exemple avec $n=2$

- Nous disposons de 3 points clefs $p_0(0, 1)$, $p_1(2, 5)$ et $p_2(4, 17)$
- Instants $t_0 = 0$, $t_1 = 2$ et $t_2 = 4$
- polynômes de Lagrange associés :

$$L_0(t) = \frac{(t-2)(t-4)}{8}$$

$$L_1(t) = \frac{t(t-4)}{-4}$$

$$L_2(t) = \frac{t(t-2)}{8}$$



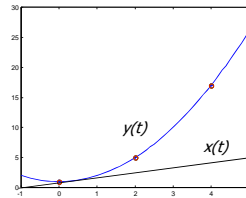
Interpolation polynomiale

- calcul du polynôme d'interpolation

- $y = L_0(x) + 5 \cdot L_1(x) + 17 \cdot L_2(x)$
- $x = 2 \cdot L_1(x) + 4 \cdot L_2(x)$

- Après simplification

- $y = t^2 + 1$
- $x = t$



Interpolation polynomiale : l'algorithme

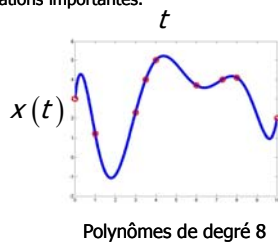
```

Lagrange  (t, p_i)
{
  pour i = 1 jusqu'à n
    pour j = 1 jusqu'à n, j ≠ i;
      l ← l * (t - t_j) / (t_i - t_j)
    fait
  p ← p + p_i * l
  fait
}
    
```

Complexité du calcul : n^2

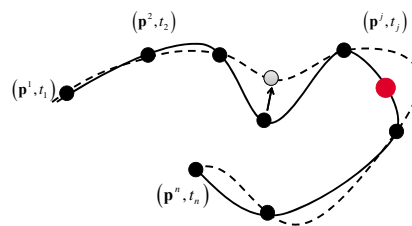
Interpolation polynomiale

- Interpolation polynomiale : Interpolation globale
 - Degré des polynômes égal au nombre de points de contrôle - 1
 - Oscillations importantes.



Interpolation polynomiale

- Interpolation polynomiale : Interpolation globale
 - Influence globale des points de contrôle.



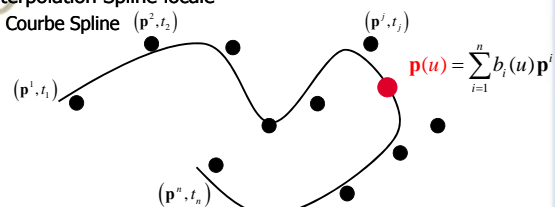
Animation par interpolation entre positions clés

- Interpolation Spline locale : Splines cubiques
 - on approche la courbe par morceaux (localement)
 - on prend des polynômes de degré faible (3) pour éviter les oscillations

Animation par interpolation entre positions clés

- Interpolation Spline locale

- Courbe Spline

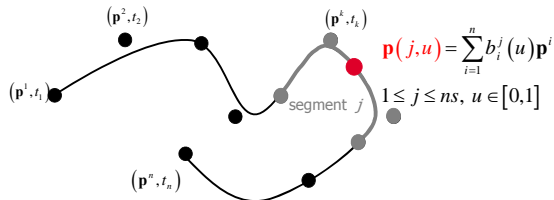


- Points de contrôle $p^1, p^2, p^3, \dots, p^n$
- Fonctions de mélange $b_1(u), b_2(u), b_3(u), \dots, b_n(u)$

Animation par interpolation entre positions clés

Interpolation Spline locale

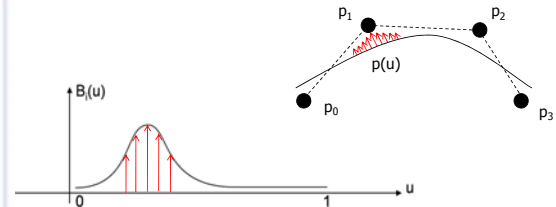
- Définition d'un segment dépend d'un nombre réduit de points de contrôle.



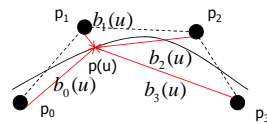
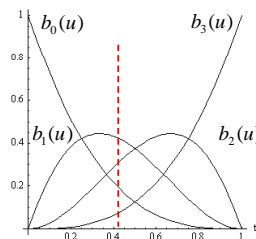
Animation par interpolation entre positions clés

Fonctions de bases

- $P(u) = B_0(u) p_0 + B_1(u) p_1 + B_2(u) p_2 + \dots$



Animation par interpolation entre positions clés



Animation par interpolation entre positions clés

- B_1 polynôme degré 3 :

$$B_1(u) = b_{13} u^3 + b_{12} u^2 + b_{11} u^1 + b_{10} u^0 \\ = (u^3 \ u^2 \ u^1 \ 1) (b_{13} \ b_{12} \ b_{11} \ b_{10})$$

- Écriture matricielle :

$$P(u) = \begin{pmatrix} u^3 & u^2 & u^1 & 1 \end{pmatrix} \begin{pmatrix} 4 & 1 & 0 & 0 \\ 1 & 4 & 1 & 0 \\ 0 & 1 & 4 & 1 \\ 0 & 0 & 1 & 4 \end{pmatrix} \begin{pmatrix} p_0 \\ p_1 \\ p_2 \\ p_3 \end{pmatrix}$$

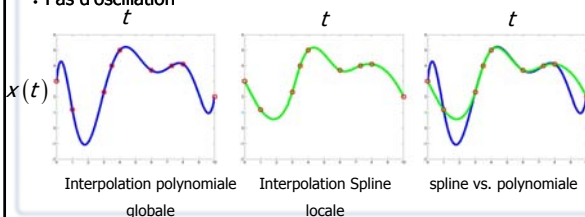
- $P(u) = (u^3 \ u^2 \ u^1 \ 1) [b_{ij}] (p_0 \ p_1 \ p_2 \ p_3)^t$

$$P(u) = [u] M_{\text{spline}} [p_j]^t$$

Animation par interpolation entre positions clés

Interpolation Spline

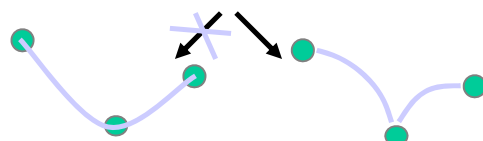
- Polynômes de degré faible (3)
- Meilleure maîtrise de l'interpolation (influence locale des points de contrôle)
- Pas d'oscillation



Animation par interpolation entre positions clés

- La continuité du mouvement en direction et vitesse n'est pas toujours souhaitée.

- l'animation d'une balle rebondissant sur un mur demande l'introduction d'une discontinuité du mouvement au point d'impact
- on ne souhaite pas voir la balle freiner avant de rencontrer le mur.



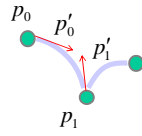
Animation par interpolation entre positions clés

Solution

- Interpolation des positions et des vitesses séparément
- Utilisation des Splines Hermite pour un meilleur contrôle de l'interpolation

$$P(u) = U \cdot M_{\text{Spline_Hermite}} \cdot P'$$

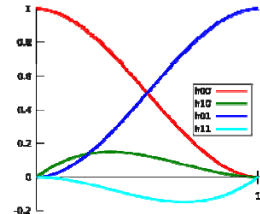
$$P(u) = \begin{pmatrix} u^3 & u^2 & u^1 & 1 \end{pmatrix} \cdot \begin{pmatrix} 2 & -2 & 0 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & -2 & 1 & 0 \\ 1 & -1 & 0 & 0 \end{pmatrix} \cdot \begin{pmatrix} p_0 \\ p_1 \\ p'_0 \\ p'_1 \end{pmatrix}$$



$$p(t) = (2t^3 - 3t^2 + 1)p_0 + (t^3 - 2t^2 + t)m_0 + (-2t^3 + 3t^2)p_1 + (t^3 - t^2)m_1$$

Animation par interpolation entre positions clés

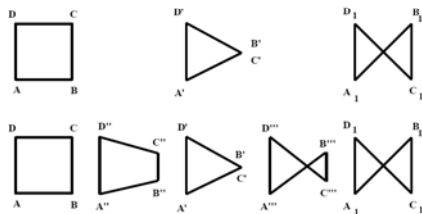
$$\begin{aligned} h_{00}(t) &= 2t^3 - 3t^2 + 1 \\ h_{10}(t) &= t^3 - 2t^2 + t \\ h_{01}(t) &= -2t^3 + 3t^2 \\ h_{11}(t) &= t^3 - t^2 \end{aligned}$$



Animation par interpolation entre positions clés

Morphing 3D (ou Morphage 3D)

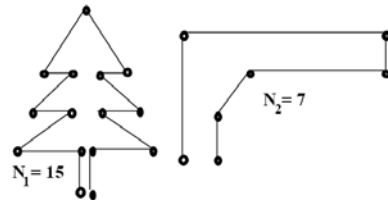
- Animation permettant de transformer une forme géométrique en une autre.



Animation par interpolation entre positions clés

Problème

- Les deux configurations clés ne présentent pas le même nombre de sommets



Animation par interpolation entre positions clés

- il faut d'abord procéder à un prétraitement consistant en l'égalisation du nombre de sommets des deux dessins. Plusieurs algorithmes sont possibles. Le plus simple est le suivant:

Soient N1 et N2 les nombres de sommets des 2 dessins.

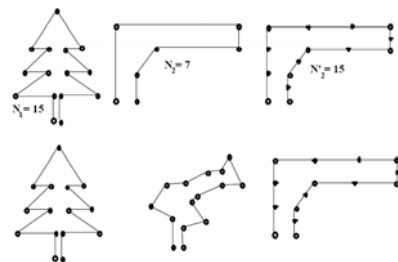
Si N1 > N2 alors

- on calcule le rapport RT:=(N1-1) div (N2-1) et le reste RS:=(N1-1) mod (N2-1)
- on ajoute RT points aux RS premiers segments et RT-1 aux autres.

Animation par interpolation entre positions clés

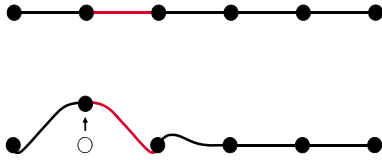
- N1=15 et N2=7 => RS=2 et RT=2.

- On ajoute donc 2 points aux 2 premiers segments et 1 point aux 4 autres.



Animation par interpolation entre positions clés **

- Interpolation Spline locale
 - Influence d'un point de contrôle réduite à un petit nombre de segments splines.



Animation par interpolation entre positions clés

- Avantages
 - Confère un haut degré de contrôle sur l'animation
 - Interpolation des paramètres ... calculs simples
- Inconvénients
 - Construction de l'animation couteuse en temps
 - Difficile de designer des animations réalistes
 - Définir toutes les expressions avant

Animation par capture de mouvements MoCap



Animation par capture de mouvements

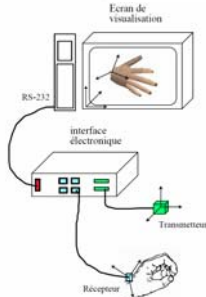
- Principe
 - Mesurer les actions directes d'un acteur pour l'analyse et la réplique immédiate ou différée
- Correspondances:
 - Directe : bras humain contrôlant le mouvement du bras de l'acteur de synthèse
 - Indirecte : mouvement de souris contrôlant les yeux d'un personnage et la direction de la tête.
- Ce qui peut être traqué :
 - tête : champ de perception (visuel, auditif, etc.)
 - pupilles : point de vue plus précis
 - main(s) : Commande et interaction avec l'EV
 - autres organes : représentation entière de l'opérateur dans l'EV.
 - autres objets de la scène réel : application mélangeant réel et virtuel (RA)

Animation par capture de mouvements

- Capteurs de position et d'orientation 3D :
 - Donne la position et l'orientation d'une partie du corps de l'opérateur dans un repère globale ou définit par le concepteur.
- Type des capteurs (systèmes de tracking)
 - Magnétique
 - Optique
 - Acoustique
 - Mécanique
- Nature de systèmes
 - Capteurs passifs
 - Les capteurs sont détectés
 - Capteurs actifs
 - L'information est envoyée par les capteurs

Systèmes de tracking

Principe

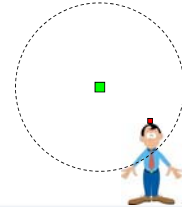
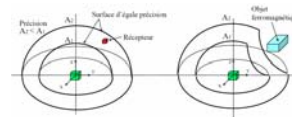


Systèmes de tracking

Traqueurs magnétiques

Principe

- Formé de deux ensembles d'antennes magnétiques orthogonales entre elles.
- Le premier ensemble est placé à l'intérieur d'un émetteur et le second à l'intérieur d'un récepteur.
- Le signal (champ magnétique) reçu par le récepteur permet de déduire la position et l'orientation de celui-ci par rapport à l'émetteur.



Systèmes de tracking

Caractéristiques

- Besoin de synchroniser les récepteurs
- Limitation concernant l'espace capturé, sensible à l'environnement
- Pour le corps humain, 11 capteurs sont nécessaires
- Utilisation de la cinématique inverse pour calculer le reste de l'information



Flock of Birds – Ascension
3.05m & ; $\pm 180^\circ$ Azimut & Roll, $\pm 90^\circ$ Élévation
Résolution statique: Position: 0.5mm ; Orientation: 0.1°
Rafraîchissement : 144 mesures/s



Systèmes de tracking

Traqueurs optiques

Principe

- Des marqueurs sont placés sur l'objet à tracker
 - Marqueurs passifs (retro-réfléchissants)
 - Marqueurs actifs (diodes infrarouge)
- Les capteurs (caméra infrarouge munies de flash infrarouges) suivent les groupes de points formant les objets à tracker
- Avantages
 - Sans fil (marqueurs passifs), Suivi d'un grand nombre de cibles
- Inconvénient
 - Prix élevé



Systèmes de tracking

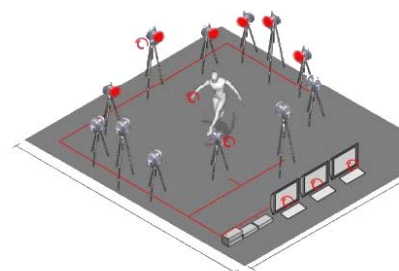
Marqueurs passifs

- Marqueurs réfléchissants
- Repérage des positions des marqueurs avec plusieurs caméras IR
- Liberté de mouvement (pas besoin de câblage)
- Problème quand il y a des occlusions (marqueurs cachés)
- Confusion entre deux marqueurs quand ils sont trop proches
- 2-6 caméras



ARTTrack2 - AR-tracking
Résolution en translation : 0.4 mm
Résolution en Orientation : 0.12°
60 fps
Capteur passifs : 4m d'actions

Systèmes de tracking



Systèmes de tracking

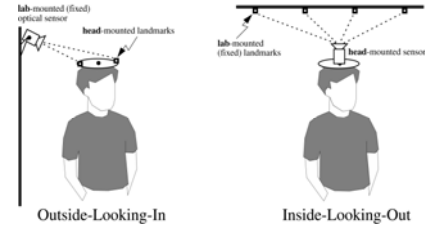
Marqueurs actifs

- Marqueurs LED synchronisés
 - Besoin de câblage pour la synchronisation des marqueurs
- Pas de confusion entre les marqueurs (modulation de signal)



Le récepteur (caméra) peut être

- Fixe => marqueurs embarqués
- Embarqué => marqueurs fixe

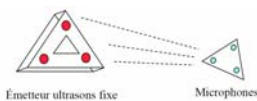


Systèmes de tracking

Traqueurs à ultrason

Principe

- Constitué d'un émetteur, d'un récepteur et d'une unité électronique.
- L'émetteur est constitué de hauts parleurs ultrasonores et le récepteur de microphones montés sur un bâti.
- Prix peu élevé par rapport aux autres systèmes.
- Sensible aux bruits de fond de la pièce est obligé une vision mutuelle directe entre le récepteur et l'émetteur.



Systèmes de tracking

InterSense : inneriel + acoustic



Systèmes de tracking

Traqueurs Mécaniques

Principe

- Les traqueurs mécaniques mesure les variations d'un système mécanique relié à l'opérateur.
- Mesure par potentiomètres ou codeurs optiques.
- Précis, pas très coûteux mais assez encombrants.



Évaluation technologique

Système	Précision	Réponse	Robustesse	Sociabilité	Commentaires
Mécanique	Bonne 0.1-2.5 mm	Bonne	Bonne Faible sensibilité à l'environnement	Encombrement assez important	Bien adapté pour le retour d'effort. Application téléopération
Optique	Bonne 0.1-0.5 mm La précision et la résolution diminuent quand le volume de travail augmente	Bonne Ces systèmes sont bien adaptés au temps réel.	Bonne Certains systèmes peuvent être affectés par la lumière ambiante	Sensible au masquage des marqueurs ou contrastes Encombrement assez important	Peut demander des installations complexes pour obtenir de bons résultats.
Magnétiques	Bonne ~ 5mm Bonne dans les petits espaces de travail Dépend fortement de la présence de métal dans la zone de travail	Fréquence d'échantillonnage relativement faible	Grande sensibilité au métal dans la zone utile	Bonne dans les petits espaces de travail.	Le plus couramment utilisé. Possibilités de suivre plusieurs objets simultanément en rajoutant des capteurs au système.
Acoustique	Bonne ~ 1mm	Dépendant de la distance	Dépendant de la distance	Sensible au masquage des capteurs ou des sources.	Extrêmement stable dans la zone utile. Peu onéreux et peu encombrant.

MoCap

Avantage

- Mouvement très réaliste : provient directement de la réalité.

Inconvénient

- Pour un nouveau mouvement, il est nécessaire d'enregistrer de nouveau la réalité.
- La capture de mouvement n'est donc pas possible dans les cas suivants :
 - simulation en temps réel, où la situation et les actions des personnages virtuels ne peuvent pas être prédits à l'avance.
 - dans les situations dangereuses, où on ne peut pas impliquer une vraie personne.

Modèles générateurs

Modèles générateurs

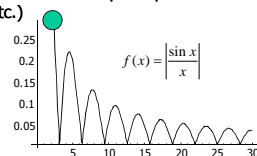
- Génération automatique du mouvement
- Génération du mouvement sur la base d'un modèle physique

Animation procédurale

Animation procédurale

- Le mouvement est décrit par une équation mathématique explicite
 - Approximation d'une loi (physique, etc.)

- Rebond d'une balle



- Pendule d'une horloge
 - $\alpha = A \sin(\omega \cdot t + \varphi)$

- Avantages :

- Simplicité d'implémentation et rapidité d'exécution

Modèles de contrôle du mouvement

Les modèles de contrôle du mouvement forment le cœur de tout système d'animation. Ils déterminent la classe des mouvements et/ou déformations réalisables par le système ainsi que la convivialité de l'interface homme machine et définissent le champ d'application du système.

Les modèles de contrôle du mouvement peuvent être classés en trois grandes familles générales :

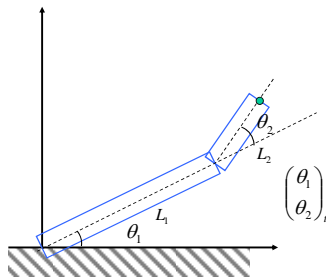
- les modèles descriptifs ou phénoménologiques : ils sont utilisés pour reproduire uniquement les effets (mouvement, déformation), sans aucune connaissance a priori sur les causes qui pourraient les avoir produits. Ils décrivent la cinématique des phénomènes dynamiques.
- les modèles générateurs ou fondés sur la physique : ils décrivent les causes capables de produire un effet. Par exemple, les modèles utilisant la mécanique font partie des modèles générateurs.
- les modèles comportementaux : ils simulent les organismes et les êtres vivants, leurs actions et leurs réponses à des stimulations extérieures.

Modélisation cinématique directe

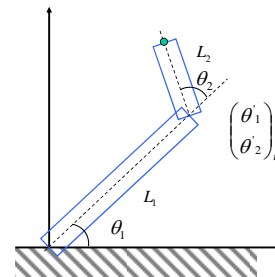
Animation par cinématique directe

- Gestion fine, bas niveau du mouvement
- Spécifier directement des vitesses
 - Le programme calcule les positions et les angles : par intégration numérique
- Compatible avec
 - Animation manuelle
 - Animation par interpolation entre positions clés
 - Animation par capture de mouvements
 - etc.
- Avantages
 - contrôle complet de l'objet
 - suivre un scénario précis
 - Utile pour des objets simples

Animation par cinématique directe



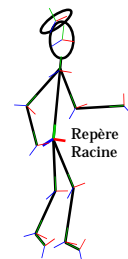
Animation par cinématique directe



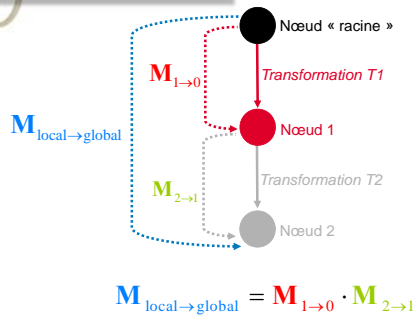
Animation par cinématique directe

- Objet décomposé en une hiérarchie de repères : Humain, animaux, robots...
- Chaque itération
 - Spécification de l'ensemble des paramètres du mouvement
 - Composition des transformation pour l'animation de chaque segment

Animation par cinématique directe



Animation par cinématique directe



Animation par cinématique directe

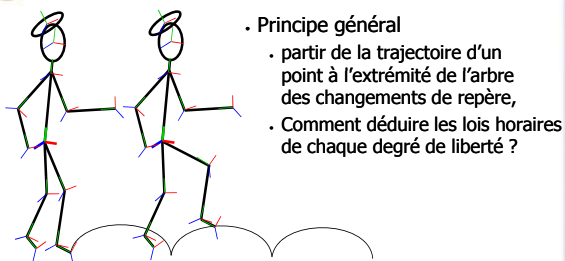
- Inconvénients
 - Animation par cinématique directe devient très complexe lorsque les chaînes hiérarchiques du squelette deviennent trop longues.
 - besoin du contrôle de tout les DDL de l'objet
 - inadaptée pour la génération de mouvements complexes

Animation par cinématique inverse

Animation par cinématique inverse

- Technique consistant à partir du résultat voulu et à calculer le meilleur mouvement possible des angles pour y parvenir.
- Spécification de haut niveau
 - point à atteindre, trajectoire d'une extrémité, ...
- D'un point de vue mathématique cela revient à faire l'inverse de la cinématique directe.
- Au lieu de partir des angles des articulations, et de trouver la position des éléments, on part des positions voulues, pour ensuite reconstruire les angles des rotations nécessaires à ce résultat.
 - Résolution indirecte d'un problème non linéaire

Animation par cinématique inverse



Animation par cinématique inverse

- Étude locale
 - L'extrémité est identifiée par sa position et son orientation rassemblées dans un vecteur \mathbf{x}
 - On cherche un vecteur de degrés de liberté $\boldsymbol{\eta}$ qui vérifie l'équation suivante :

$$\mathbf{x} = f(\boldsymbol{\eta})$$
 - f est la fonction résultant de la composition des changements de repères depuis l'extrémité jusqu'à la racine

Animation par cinématique inverse

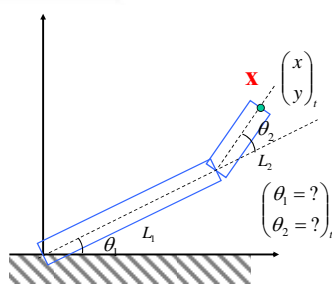
- La fonction f est une composée de rotations et de translations $(\theta_1, \theta_2, \theta_3, \dots, t_1, t_2, \dots)$

Animation par cinématique inverse

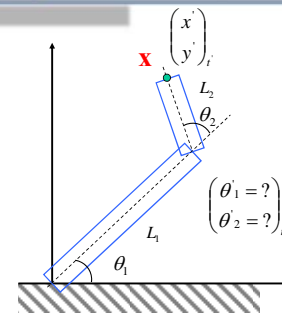
- En résumé
 - Donnée : position à atteindre $\mathbf{X} = (x_1, x_2, \dots)$
 - Sortie : valeurs des paramètres des articulations
 - $\boldsymbol{\eta}$ = vecteur des paramètres du modèle
 - $\boldsymbol{\eta} = (\theta_1, \theta_2, \theta_3, \dots)$
 - Trouver :

$$\mathbf{X} = f(\boldsymbol{\eta})$$

Animation par cinématique inverse

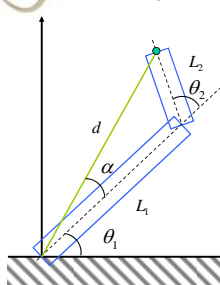


Animation par cinématique inverse



Animation par cinématique inverse

- Solution directe (deux rotations) $d = \sqrt{x^2 + y^2}$



$$d^2 = (L_1 + L_2 \cos \theta_2)^2 + (L_2 \sin \theta_2)^2$$

$$d^2 = L_1^2 + L_2^2 + 2L_1L_2 \cos \theta_2$$

$$\cos \theta_2 = \frac{d^2 - L_1^2 - L_2^2}{2L_1L_2}$$

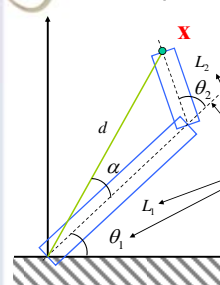
$$\tan(\alpha) = \frac{L_2 \sin \theta_2}{L_1 + L_2 \cos \theta_2}$$

$$\tan(\theta_1 + \alpha) = \frac{y}{x}$$

$$\theta_1 = \arctan\left(\frac{y}{x}\right) - \arctan\left(\frac{L_2 \sin \theta_2}{L_1 + L_2 \cos \theta_2}\right)$$

Animation par cinématique inverse

- Solution directe (deux rotations) $d = \sqrt{x^2 + y^2}$



$$d^2 = (L_1 + L_2 \cos \theta_2)^2 + (L_2 \sin \theta_2)^2$$

$$d^2 = L_1^2 + L_2^2 + 2L_1L_2 \cos \theta_2$$

$$\cos \theta_2 = \frac{d^2 - L_1^2 - L_2^2}{2L_1L_2}$$

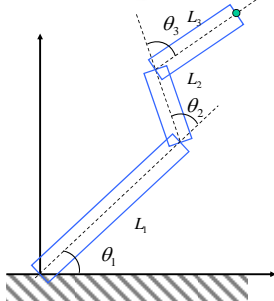
$$\tan(\alpha) = \frac{L_2 \sin \theta_2}{L_1 + L_2 \cos \theta_2}$$

$$\tan(\theta_1 + \alpha) = \frac{y}{x}$$

$$\theta_1 = \arctan\left(\frac{y}{x}\right) - \arctan\left(\frac{L_2 \sin \theta_2}{L_1 + L_2 \cos \theta_2}\right)$$

Animation par cinématique inverse

- Trois rotations : calcul direct également

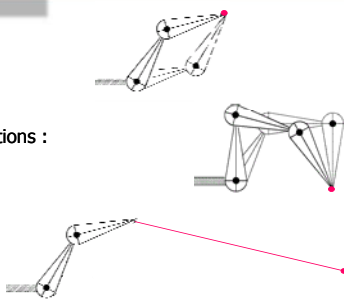


Animation par cinématique inverse

- N articulations (au-delà de 3 rotations)
 - dépendance non linéaire vis-à-vis des degrés de liberté
 - non-linéaire à cause des rotations
- Contraintes de la cinématique inverse :
 - Problème non-linéaire
 - Plusieurs solutions
 - Pas toujours bien conditionné
 - Limites des articulations

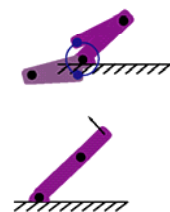
Animation par cinématique inverse

- Non-unicité
 - Deux solutions :
- Intervalle de solutions :
- Pas de solutions :



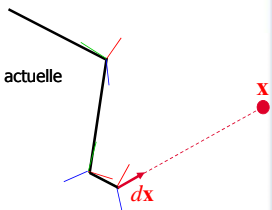
Animation par cinématique inverse

- Mauvais conditionnement
 - Petite différence sur \mathbf{x} , grande différence sur $\boldsymbol{\eta}$:
- Limites des articulations
 - Changer $\boldsymbol{\eta}$ ne rapproche pas de \mathbf{x} :



Animation par cinématique inverse

- Approche différentielle
 - Petits pas par petits pas
 - À chaque étape :
 - Choix entre plusieurs solutions
 - Prendre solution proche position actuelle
 - Taille des pas :
 - Chercher taille optimale ?
 - Rotations : pour $x < 2$ degrés:
 - $\sin(x) \approx x$
 - $\cos(x) \approx 1$
 - Garder pas < 2 degrés



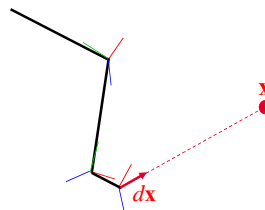
Animation par cinématique inverse

Approche différentielle

$$\mathbf{x} = f(\boldsymbol{\eta})$$

$$\downarrow$$

$$d\mathbf{x} = \mathbf{J}(\boldsymbol{\eta})d\boldsymbol{\eta}$$



Animation par cinématique inverse

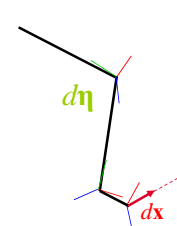
$\dot{\mathbf{x}} = \mathbf{J}(\boldsymbol{\eta}) \dot{\boldsymbol{\eta}}$
 $\dot{\mathbf{x}} = d\mathbf{x} / dt$ $\dot{\boldsymbol{\eta}} = d\boldsymbol{\eta} / dt$

$$\mathbf{J}(\boldsymbol{\eta}) = \begin{pmatrix} \frac{\partial f_1}{\partial \eta_1} & \dots & \frac{\partial f_1}{\partial \eta_q} \\ \vdots & \dots & \vdots \\ \frac{\partial f_p}{\partial \eta_1} & \dots & \frac{\partial f_p}{\partial \eta_q} \end{pmatrix}$$

$\frac{\partial f}{\partial \eta_1} \frac{\partial f}{\partial \theta_1} = \frac{\partial \mathcal{R}_1}{\partial \theta_1} T_1 R_2 T_2 R_3 T_3 \dots$

Animation par cinématique inverse

Approche différentielle



$\dot{\mathbf{x}} = \mathbf{J}(\boldsymbol{\eta}) \dot{\boldsymbol{\eta}}$
 $\dot{\boldsymbol{\eta}} = \mathbf{J}(\boldsymbol{\eta})^{-1} \dot{\mathbf{x}}$

Animation par cinématique inverse

- J n'est pas inversible
- J n'est pas carrée : J matrice 2*n

$$\begin{bmatrix} \frac{\partial f_x}{\partial \theta_0} & \frac{\partial f_x}{\partial \theta_1} & \dots \\ \frac{\partial f_y}{\partial \theta_0} & \frac{\partial f_y}{\partial \theta_1} & \dots \end{bmatrix} \begin{bmatrix} \theta_0 \\ \theta_1 \\ \vdots \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix}$$

- l'inverse de J n'existe pas.

Animation par cinématique inverse

- Pseudo-Inverse
- J n'est pas carrée par contre $\mathbf{J}^T \mathbf{J}$ est carrée (n*n)

$\dot{\mathbf{x}} = \mathbf{J}(\boldsymbol{\eta}) \dot{\boldsymbol{\eta}}$
 $\dot{\mathbf{x}} = \mathbf{J} \dot{\boldsymbol{\eta}}$
 $\mathbf{J}^T \dot{\mathbf{x}} = \mathbf{J}^T \mathbf{J} \dot{\boldsymbol{\eta}}$
 $(\mathbf{J}^T \mathbf{J})^{-1} \mathbf{J}^T \dot{\mathbf{x}} = (\mathbf{J}^T \mathbf{J})^{-1} \mathbf{J}^T \mathbf{J} \dot{\boldsymbol{\eta}}$
 $\dot{\boldsymbol{\eta}} = (\mathbf{J}^T \mathbf{J})^{-1} \mathbf{J}^T \dot{\mathbf{x}}$

Animation par cinématique inverse

- La solution

$\dot{\boldsymbol{\eta}} = (\mathbf{J}^T \mathbf{J})^{-1} \mathbf{J}^T \dot{\mathbf{x}}$
 $\dot{\boldsymbol{\eta}} = \mathbf{J}^* \dot{\mathbf{x}}$

- Avec

$\mathbf{J}^* = (\mathbf{J}^T \mathbf{J})^{-1} \mathbf{J}^T$ (pseudo-inverse)

- Au final

$\boldsymbol{\eta} = \boldsymbol{\eta} + \dot{\boldsymbol{\eta}}$

Animation par cinématique inverse

- Comment calculer \mathbf{J}^+ ?
- Si $\mathbf{J}^T \mathbf{J}$ n'est pas inversible

Singular Values Decomposition

- Toute matrice $m \times n$ peut s'exprimer par SVD:

$$\mathbf{A} = \mathbf{U} \mathbf{S} \mathbf{V}^T$$

- \mathbf{U} est une matrice $m \times m$, orthonormale
- \mathbf{S} est une matrice $n \times n$, diagonale, *singular values*
- \mathbf{V} est une matrice $n \times n$ orthonormale

$$\mathbf{A} = (\mathbf{h}_1 | \mathbf{h}_2 | \dots | \mathbf{h}_n) \cdot \begin{pmatrix} s_1 & 0 & 0 & 0 \\ 0 & s_2 & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & s_n \end{pmatrix} \cdot \begin{pmatrix} \mathbf{a}_1 \\ \mathbf{a}_2 \\ \vdots \\ \mathbf{a}_n \end{pmatrix}$$

$m \times n \quad m \times m \quad m \times n \quad n \times n$

- Exemple

$$\begin{pmatrix} 2 & 0 \\ 0 & -3 \\ 0 & 0 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 2 & 0 \\ 0 & 3 \\ 0 & 0 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

Singular Values Decomposition

- Les s_i sont les valeurs singulières de \mathbf{A}
- Si la matrice \mathbf{A} est singulière, il y a des s_i nulles
- Rang de \mathbf{A} : nombre de s_i non nulles
- La décomposition SVD est unique, mis à part l'ordre des s_i ou s'il y a égalité parmi des s_i
- Déterminant : produit des valeurs
- \mathbf{U}, \mathbf{V} : colonnes orthogonales

Calcul de la matrice inverse

$$\mathbf{A}^{-1} = (\mathbf{V}^T)^{-1} \mathbf{S}^{-1} \mathbf{U}^{-1} = \mathbf{V} \mathbf{S}^{-1} \mathbf{U}^T$$

- L'inverse d'une matrice orthonormale est sa transposée
- Puisque \mathbf{S} est diagonale, \mathbf{S}^{-1} est aussi diagonale avec les inverses multiplicatifs des s_i

$$\begin{pmatrix} s_1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & s_n \end{pmatrix}^{-1} = \begin{pmatrix} \frac{1}{s_1} & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \frac{1}{s_n} \end{pmatrix}$$

Calcul de la matrice inverse

$$\mathbf{A}^{-1} = (\mathbf{V}^T)^{-1} \mathbf{S}^{-1} \mathbf{U}^{-1} = \mathbf{V} \mathbf{S}^{-1} \mathbf{U}^T$$

- Ce n'est pas vrai quand des $s_i = 0$
 - La matrice est singulière
- Pseudoinverse: si $s_i = 0$, on fixe $1/s_i = 0$
 - C'est la matrice "la plus près" de l'inverse
 - Existe pour toutes les matrices, même singulières ou rectangulaires
 - Égale à $(\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T$ si $\mathbf{A}^T \mathbf{A}$ est inversible

Pseudo-Inverse avec SVD

- Calculer SVD: $\mathbf{A} = \mathbf{U} \mathbf{S} \mathbf{V}^T$

$$\begin{pmatrix} \mathbf{A} \end{pmatrix} = \begin{pmatrix} \mathbf{U} \end{pmatrix} \begin{pmatrix} s_1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & s_n \end{pmatrix} \begin{pmatrix} \mathbf{V} \end{pmatrix}^T$$

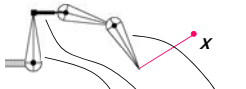
- Pseudo-inverse : $\mathbf{A}^+ = \mathbf{V} \mathbf{S}^{-1} \mathbf{U}^T$
 - Si \mathbf{S} n'est pas singulière

$$\begin{pmatrix} \mathbf{A}^+ \end{pmatrix} = \begin{pmatrix} \mathbf{V} \end{pmatrix} \begin{pmatrix} \frac{1}{s_1} & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \frac{1}{s_n} \end{pmatrix} \begin{pmatrix} \mathbf{U} \end{pmatrix}^T$$

Pseudo-Inverse avec SVD

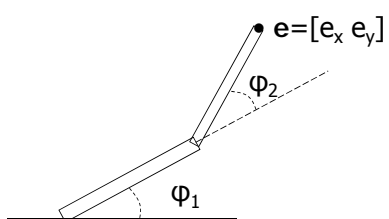
- Si S est :
 - Singulière : $s_j = 0$
 - Mal conditionnée ($s_j \ll s_0$)
 - On vérifie le ratio de la valeur singulière max sur la valeur min (condition number).
 - C'est une indication de la stabilité du système
 - Test : $s_j < \varepsilon s_0$
- Alors:
 - Prendre 0 au lieu de $1/s_j$ pour ces valeurs

Calcul du Jacobien

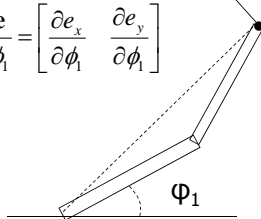


$$\begin{bmatrix} \frac{de_x}{d\varphi_0} & \frac{de_x}{d\varphi_1} & \dots & \frac{de_x}{d\varphi_N} \\ \frac{de_y}{d\varphi_0} & \frac{de_y}{d\varphi_1} & \dots & \frac{de_y}{d\varphi_N} \end{bmatrix}$$

Calcul du Jacobien

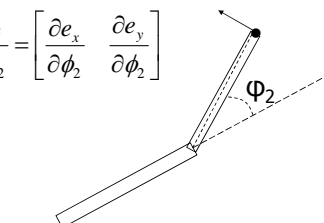


Calcul du Jacobien



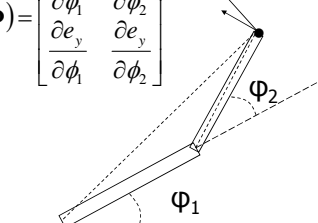
$$\frac{\partial e}{\partial \phi_1} = \begin{bmatrix} \frac{\partial e_x}{\partial \phi_1} & \frac{\partial e_y}{\partial \phi_1} \end{bmatrix}$$

Calcul du Jacobien



$$\frac{\partial e}{\partial \phi_2} = \begin{bmatrix} \frac{\partial e_x}{\partial \phi_2} & \frac{\partial e_y}{\partial \phi_2} \end{bmatrix}$$

Calcul du Jacobien



$$J(e, \Phi) = \begin{bmatrix} \frac{\partial e_x}{\partial \phi_1} & \frac{\partial e_x}{\partial \phi_2} \\ \frac{\partial e_y}{\partial \phi_1} & \frac{\partial e_y}{\partial \phi_2} \end{bmatrix}$$

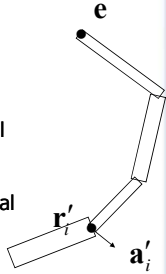
Calcul du Jacobien: DDL Rotation

- Dans un premier temps il faut calculer le pivot et l'axe de rotation

$$\mathbf{a}'_i = \mathbf{a}_i \cdot \mathbf{W}_{i-\text{parent}}$$

$$\mathbf{r}'_i = \mathbf{r}_i \cdot \mathbf{W}_{i-\text{parent}}$$

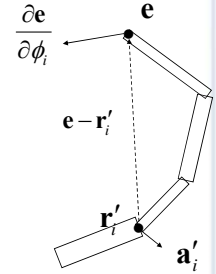
- \mathbf{a}_i resp. \mathbf{a}'_i : axe de rotation dans le repère local resp. global (vecteur unitaire)
 - \mathbf{a} est un vecteur unitaire $[a_x \ a_y \ a_z \ 0]$
- \mathbf{r}_i resp. \mathbf{r}'_i : pivot dans le repère local resp. global
 - \mathbf{r} est une position $[r_x \ r_y \ r_z \ 1]$
- $\mathbf{W}_{i-\text{parent}}$: changement de repère du nœud concerné vers le repère parent



Calcul du Jacobien: DDL Rotation

- Le Jacobien du nœud est égal à

$$\frac{\partial \mathbf{e}}{\partial \phi_i} = \mathbf{a}'_i \times (\mathbf{e} - \mathbf{r}'_i)$$



Calcul du Jacobien: DDL Rotation

- Avec une rotation 3DDL : XYZ dans l'ordre

$$x-ddl : \mathbf{a}'_i = [1 \ 0 \ 0 \ 0] \cdot \mathbf{R}_y(\theta_y) \cdot \mathbf{R}_z(\theta_z) \cdot \mathbf{W}_{\text{parent}}$$

$$y-ddl : \mathbf{a}'_i = [0 \ 1 \ 0 \ 0] \cdot \mathbf{R}_z(\theta_z) \cdot \mathbf{W}_{\text{parent}}$$

$$z-ddl : \mathbf{a}'_i = [0 \ 0 \ 1 \ 0] \cdot \mathbf{W}_{\text{parent}}$$

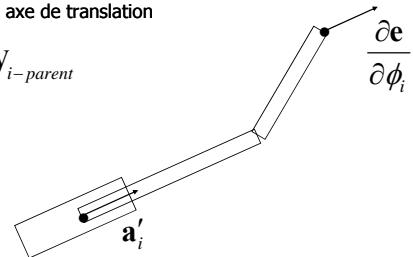
- $\mathbf{R}_y(\theta_y)$ et $\mathbf{R}_z(\theta_z)$ sont les matrices de rotation suivant y et z

Calcul du Jacobien : DDL translation

- Pour les DDL en translation uniquement
 - Pas de pivot
 - Uniquement un axe de translation

$$\mathbf{a}'_i = \mathbf{a}_i \cdot \mathbf{W}_{i-\text{parent}}$$

$$\frac{\partial \mathbf{e}}{\partial \phi_i} = \mathbf{a}'_i$$



Calcul du Jacobien : DDL translation

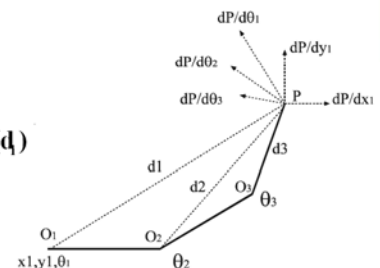
- Translation en 3D XYZ

- On applique la même démarche suivant chaque axe
- La translation suivant un axe n'influe pas les translations suivant les autres axes
- Le résultat est un vecteur unitaire dans la même direction que l'axe de translation

Calcul du Jacobien

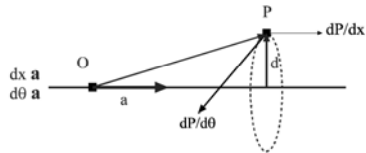
$$\frac{d\mathbf{P}}{dx_i} = \mathbf{axis}(x_i)$$

$$\frac{d\mathbf{P}}{d\theta_i} = \|\mathbf{d}_i\| \mathbf{norm}(\mathbf{d}_i)$$



Calcul du Jacobien

Exemple : rotation et translation

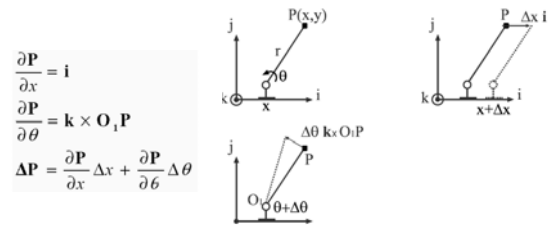


$$\frac{dP}{dx} = a$$

$$\frac{dP}{d\theta} = a \times d = a \times OP$$

Calcul du Jacobien

Exemple rotation et translation



$$\frac{\partial P}{\partial x} = i$$

$$\frac{\partial P}{\partial \theta} = k \times O_1 P$$

$$\Delta P = \frac{\partial P}{\partial x} \Delta x + \frac{\partial P}{\partial \theta} \Delta \theta$$

Intégration de contraintes

$$P_1 - P = \Delta P = 0$$

$$g = \begin{pmatrix} \Delta P \cdot i \\ \Delta P \cdot j \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

$$J = \frac{dg}{d\theta}$$

$$J = \begin{bmatrix} \partial g_1 / \partial \theta_1 & \dots & \partial g_1 / \partial \theta_n \\ \dots & \dots & \dots \\ \partial g_m / \partial \theta_1 & \dots & \partial g_m / \partial \theta_n \end{bmatrix}$$

Intégration de contraintes

$$\begin{bmatrix} \frac{\partial P}{\partial \theta_1} \cdot i & \frac{\partial P}{\partial \theta_2} \cdot i & \frac{\partial P}{\partial \theta_3} \cdot i \\ \frac{\partial P}{\partial \theta_1} \cdot j & \frac{\partial P}{\partial \theta_2} \cdot j & \frac{\partial P}{\partial \theta_3} \cdot j \end{bmatrix} \begin{bmatrix} \Delta \theta_1 \\ \Delta \theta_2 \\ \Delta \theta_3 \end{bmatrix} = \begin{bmatrix} x_1 - x \\ y_1 - y \end{bmatrix}$$

Intégration de contraintes

Contrainte point

$$\Delta P = \frac{\partial P}{\partial x} \Delta x + \frac{\partial P}{\partial \theta} \Delta \theta = g = P_1 - P$$

$$\begin{bmatrix} \frac{\partial P}{\partial x} \cdot i & \frac{\partial P}{\partial \theta} \cdot i \\ \frac{\partial P}{\partial x} \cdot j & \frac{\partial P}{\partial \theta} \cdot j \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta \theta \end{bmatrix} = \begin{bmatrix} (P_1 - P) \cdot i \\ (P_1 - P) \cdot j \end{bmatrix}$$

$$\begin{bmatrix} 1 & (k \times O_1 P) \cdot i \\ 0 & (k \times O_1 P) \cdot j \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta \theta \end{bmatrix} = \begin{bmatrix} x_1 - x \\ y_1 - y \end{bmatrix}$$

Intégration de contraintes

Contrainte point-ligne

$$\Delta P \cdot n = \left(\frac{\partial P}{\partial x} \Delta x + \frac{\partial P}{\partial \theta} \Delta \theta \right) \cdot n = -d = (P_1 - P) \cdot n$$

$$\begin{bmatrix} \frac{\partial P}{\partial x} \cdot n & \frac{\partial P}{\partial \theta} \cdot n \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta \theta \end{bmatrix} = -d$$

$$\begin{bmatrix} i \cdot n & (k \times O_1 P) \cdot n \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta \theta \end{bmatrix} = -d$$

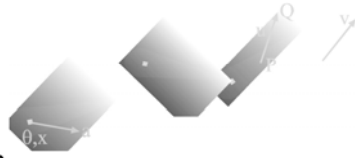
Intégration de contraintes

- Contrainte sur les orientation : vecteur d'alignement

$$\mathbf{g} = \mathbf{v} - \mathbf{u}$$

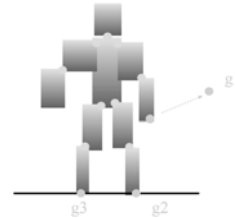
$$\frac{\partial \mathbf{PQ}}{\partial x} = \frac{\partial \mathbf{Q}}{\partial x} - \frac{\partial \mathbf{P}}{\partial x} = \mathbf{a} - \mathbf{a} = \mathbf{0}$$

$$\frac{\partial \mathbf{PQ}}{\partial \theta} = \frac{\partial \mathbf{Q}}{\partial \theta} - \frac{\partial \mathbf{P}}{\partial \theta} = \mathbf{a} \times \mathbf{O}_i \mathbf{B} - \mathbf{a} \times \mathbf{O}_i \mathbf{A} = \mathbf{a} \times \mathbf{AB}$$



Contraintes multiples

- On regroupe toutes les contraintes dans la même équation



$$\begin{bmatrix} \mathbf{J}_1 \\ \mathbf{J}_2 \\ \mathbf{J}_3 \end{bmatrix} \Delta \theta = \begin{bmatrix} \mathbf{g}_1 \\ \mathbf{g}_2 \\ \mathbf{g}_3 \end{bmatrix}$$

Limites aux articulations

- Chaque articulation a un intervalle limité
 - Par ex. le coude : varie sur $[0, \pi]$
 - Élément important du réalisme
- Pour forcer les limites :
 - Tester si dépassement
 - Annuler paramètre i
 - Recalculer sans i
 - Vérifier les autres paramètres

Limites aux articulations

- Algorithme modifié :
 - Après avoir calculé \mathbf{dn} , test pour ch. articulation:
 - $\theta_{min_i} < (\theta + \mathbf{dn}_i) < \theta_{max_i}$
 - Si ça sort de l'intervalle :
 - Annuler colonne i de \mathbf{J}
 - Revient à annuler paramètre i
 - Recalculer \mathbf{J}^+
 - Moindres carrés : $\mathbf{dn}_i \approx 0$
 - Pour plus de robustesse, forcer $\mathbf{dn}_i = 0$
 - Trouver \mathbf{dn} , itérer

Algorithme

```
inverseKinematics()
{
    Vector n = getLinkParameters();
    Vector x = target - computeEndPoint();
    for(k=0; k<kmax ; k++){
        Matrix J = computeJacobian();
        Matrix J* = pseudoInverse(J);
        Vector dn = J* x ;
        n = n+ dn;
        putLinkParameters();
    }
}
```