



HAL
open science

Etude des applications de Simulation Distribuée Interactive et l'algorithme Dead Reckoning

Akram Hakiri

► **To cite this version:**

Akram Hakiri. Etude des applications de Simulation Distribuée Interactive et l'algorithme Dead Reckoning. CONF EDSYS, May 2010, Toulouse, France. pp.103. hal-00608012

HAL Id: hal-00608012

<https://hal.science/hal-00608012>

Submitted on 11 Jul 2011

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Etude des applications de Simulation Distribuée Interactive et l'algorithme Dead Reckoning

Hakiri Akram (hakiri@laas.fr)

EDSYS, Toulouse

UNIV, Toulouse

Thèse encadrée par : Pascal Berthou¹² et Thierry Gayraud¹²

¹ CNRS ; LAAS, 7, avenue du Colonel Roche, 31077 Toulouse, France

² Université Toulouse ; UPS, INSA, INP, ISAE ; LAAS ; F-31077 Toulouse, France

Résumé La simulation distribuée vise à proposer une architecture de communication commune permettant l'intégration et l'interopérabilité d'un ensemble de simulateurs géographiquement distant. Les techniques de simulation distribuée permettent de réduire le temps et le coût de la conception de prototypes, leurs développements, leurs tests et le raffinement de leur cycle de vie. Ce domaine, aujourd'hui mature mais encore en pleine évolution, recèle encore des difficultés concernant la mise en oeuvre de simulateurs temps réels répartis sur des grandes distances. En effet, les effets du réseau sur la transmission d'information entre simulateurs sont gérés et compensés par des techniques relativement basiques. Les progrès fait dans le domaine de la gestion de la qualité de service dans les réseaux de nouvelle génération, ouvrent de nouvelles perspectives quand à l'adéquation de la simulation distribuée sur les nouveaux services de transmission offerts. Cette thèse à donc pour objectif d'aborder cette problématique. L'article présentera l'architecture que nous envisageons et se focalisera sur un point particulier qui est la compensation de l'erreur par une méthode de dead-reckoning originale.

A. INTRODUCTION

Une application de simulation distribuée interactive (DIS) est un système de réalité virtuelle interconnecté sur des réseaux grandes distances et qui partagent entre eux des informations via des interactions individuelles et coopératives. L'un des aspects les plus importants de ces applications est la capacité de chaque site de gérer en temps réel les états toutes les entités (orientation et position) participantes dans l'exercice.

L'architecture réseaux dans lesquels des entités distribuées sont intégrées n'a cessé d'évoluer. Nous focalisons sur cette évolution à travers la description des standards utilisés dans le cadre de nos travaux de thèse.

L'organisation de ce papier est comme suit : la section B introduit le projet PLATSIM. La section C présente les travaux antérieurs réalisés dans le cadre de thèse. Dans la section D nous focalisons sur une étude de l'algorithme Dead Reckoning. Une conclusion est donnée à la section E.

B. LE PROJET PLATSIM

Le projet PLATSIM a pour objet la conception et la réalisation de simulateurs génériques destinés à l'apprentissage de la conduite automobile ou motocycliste. En effet, les besoins en formation évoluent et les menaces actuelles qui s'y rattachent imposent l'entraînement collectif d'équipes complètes, utilisant des environnements quasi réels, afin d'être à même d'appréhender et de maîtriser ces nouvelles situations.

Le partenariat industriel-utilisateurs de PLATSIM a amené à la définition d'un contenu pédagogique applicatif adéquat, temps réel, donc le fonctionnement doit être supporté par une plate-forme intégrant un ensemble de simulateurs en réseau. Cette plate-forme doit pouvoir interconnecter jusqu'à 16 simulateurs, ces simulateurs pouvant se trouver sur le même site ou bien sur des sites distants. Un tel système permettrait alors l'entraînement d'équipes dans des situations d'urgence ou à risque.

Parmi les différentes problématiques s'y rattachant, deux d'entre elles concerne la mise en réseau de ces simulateurs et l'estimation des états des entités si-

mulés par chacun de ces simulateurs. En effet, ces simulateurs doivent pouvoir communiquer et partager entre eux plusieurs informations importantes, telles que les informations de position, d'état. . . Par exemple, une demi-seconde de latence sur des simulateurs en réseau, compte tenu de la vitesse des véhicules simulés, peut se traduire par un décalage de 15 mètres dans la base de données 3D. De plus, dans le cas général, ces simulateurs doivent être définis afin de s'interconnecter si possible aussi bien sur des réseaux locaux que sur des réseaux grandes distances, et dans ce dernier cas en utilisant des réseaux de cœur de type internet. Il s'agit ici de supporter par exemple le cas simple où deux utilisateurs sont connectés à deux sites équipés et éloignés et souhaitent s'entraîner ensemble.

En plus de méthodes classiques de prédiction et d'anticipation, et de hiérarchisation des flux de données, un réseau temps réel est à développer tout en offrant des performances suffisantes aux différents simulateurs et équipements qui participeront à la réalisation de l'application simulée. Le but est d'assurer que le système ainsi conçu est capable de garantir la QoS (Quality of Service) des communications, de calculateurs applicatifs à calculateurs applicatifs, afin de garantir le déroulement synchrone des simulations et la transmission en temps réel de tous les événements présents dans cette simulation.

C. TRAVAUX ACTUEL

La problématique générale visée par cette thèse consiste à définir comment intégrer les protocoles de Transport et les nouveaux systèmes de fournitures de QoS IP dans une architecture de communication de bout en bout à garantie de QoS reliant des applications de simulation interactive distribuée. Trois aspects sont adressés dans ce papier : le premier aspect vise à proposer un framework pour lier la QoS à la simulation distribuée afin d'être utilisé dans le cadre du projet PLATSIM. Pour ce faire, Les framework HLA [IEE00] et DDS [OMG07] sont étudiés et des interfaces de programmation (API) sont proposées afin de pouvoir caractériser les flux de la simulation et établir un schéma de synchronisation permettant de satisfaire les contraintes de QoS.

Le deuxième aspect s'adresse à la synchronisation entre les flux de communication issus d'un simulateur et leur présentation sur le site récepteur. Une modélisation par des réseaux de Petri temporel à flux hiérarchique a été proposée. Le troisième aspect est lié à la procédure de mise à jour de d'une entité simulée par un ensemble d'algorithmes de Dead Reckoning.

1. High Level Architecture (HLA)

HLA [IEE00] est une méthodologie qui se focalise sur l'utilisation d'une simulation distribuée de haut

niveau pour faciliter l'interopérabilité des modèles et des simulations partielles (simulation temps réel, événementielle, de contrôle commande, . . .) et faciliter la réutilisation des composants de modélisation et de simulation (M&S) existants.

Elle permet aussi d'exploiter de multiples programmes (simulateurs) produits indépendamment pour créer une simulation à grande envergure, ceci en rajoutant la spécification de mécanismes permettant la distribution nécessaire à la communication globale.

Chaque simulation participante est appelée un fédéré; elle interagit avec d'autres fédérés au sein d'une collection de simulations, dite fédération HLA. La spécification décrit la manière avec laquelle les fédérés communiquent dans la fédération au travers d'un middleware indispensable pour implémenter l'architecture des services de HLA, dite HLA-RTI (Run-Time Infrastructure). Les fédérés interagissent en utilisant les services proposés par HLA-RTI. Chaque fédéré peut publier (produire) une variable pour informer d'une intention d'envoyer des données à d'autres fédérés au sein d'une même fédération et souscrire (consommer) à une variable pour refléter certaines informations créées et mise à jour par d'autres fédérés.

Les implémentations commerciales du HLA-RTI ne fournissent pas des API qui permettent la garantie de la QoS de bout en bout entre des simulateurs distants. Nous proposons des interfaces de programmations permettant de garantir les communications et la gestion des flux sur un framework HLA.

2. Data Distribution Service (DDS)

DDS est un standard spécifié par l'OMG (Object Management Group) [OMG07], et son rôle est de proposer une technologie évoluée d'échange de données sur des réseaux allant des systèmes embarqués aux réseaux grande distance.

En se basant sur une architecture producteur/consommateur, cette spécification définit une interface logicielle permettant à des applications hétérogènes de communiquer entre elles. Elle s'adresse principalement aux industries soumises à de fortes contraintes de fiabilité et de performances telles que l'aéronautique, la défense ou encore les télécommunications. Les API et la qualité du service (QoS) incluses dans DDS sont choisies pour avoir à la fois un comportement prédictible et une implémentation efficace et performante. La spécification DDS offre deux niveaux d'interface : L'un de bas niveau, Data Centric Publish Subscribe (DCPS) [15], hautement configurable, étroitement lié aux données et riche de nombreux paramètres QoS pour déterminer le comportement requis [16]. Un niveau plus élevé, Data Local Reconstruction Layer (DLRL), qui offre une approche simplifiée du modèle OMT de HLA et qui permet une meilleure intégration au niveau applicatif. Le standard OMG-

DDS cible les communications temps réel dans des domaines allant des systèmes embarqués de contrôle commande jusqu’aux systèmes largement distribués sur des réseaux étendus. Chaque profil DDS ajoute des fonctionnalités distinctes qui définissent le niveau de service offert par DDS pour réaliser le paradigme ” bonnes données au bon moment et au bon endroit ”.

Au niveau de ce framework, nos travaux de recherche sont focalisés sur l’adaptabilité de la QoS fournie par DDS à des réseaux de communication grande distance à garantie de QoS.

3. Synchronisation des flux

La synchronisation intra-flux permet d’assurer le respect des contraintes temporelles portant sur les unités de synchronisation de chaque flux pris individuellement. La synchronisation est en général définie pour une unité de synchronisation, qui est définie pour un ensemble de données de base, ou objets, par exemple k échantillons d’audio, ou m trames de vidéo. Le niveau synchronisation intra-flux définit, pour chaque flux, les intervalles de validité temporelle (durée nominale, gigue admissible positive et négative,...) acceptable pour chaque unité de synchronisation. Un deuxième problème récurrent, et induit par le premier, concerne la synchronisation entre plusieurs flux. Par exemple les applications de simulation distribuée interactive intègrent des flux multimedia et des flux interactifs issus des consoles de jeux des joueurs en réseau. Pour assurer une vision cohérente de l’ensemble de ces flux, il convient d’assurer la synchronisation entre tous ces flux. Ceci pourra se faire en garantissant la synchronisation entre les flux deux à deux, par exemple en assurant tout d’abord la synchronisation entre les flux multimedia, et ensuite assurer la synchronisation entre un de ces flux et le flux interactif de la simulation distribuée. Nous avons proposés un modèle de synchronisation basé sur les réseaux de Petri temporels à flux.

D. PRINCIPE DU DEAD RECKONING (DR)

1. Vue d’ensemble

Afin de réduire le nombre de messages de mise à jour des états de ses entités, chaque site maintien, en plus de la représentation réelle, un modèle de haute fidélité (représentation extrapolée ou modèle Dead Reckoning) pour estimer les états des entités simulées localement [MMN04]. Les états anticipés sont calculés à partir des informations d’états de leur passé en utilisant des équations d’extrapolation. L’écart entre l’état extrapolé et l’état réel ne doit pas excéder l’un des seuils définis dans le standard (Th_{pos} pour la position et Th_{or} pour l’orientation).

Sur tous les autres sites distants, la réception d’un paquet contenant une nouvelle position engendre la mise à jour de l’état de l’entité concernée par une technique de dérivation polynomiale. Nous focalisons à la dérivation du 2^{eme} ordre : si l’on dispose de la position P_i , de la vitesse V_i et de l’accélération A_i à l’instant t_i , on peut alors estimer la position, $P_{DR}(t)$, de l’entité à tout instant $t > t_i$ grace à une extrapolation quadratique exprimée par la relation 1 :

$$P_{DR}(t) = P_i + V_i(t - t_i) + A_i(t - t_i)^2 \quad (1)$$

Une simulation interactive distribuée présente, non seulement des contraintes liées à la nature distribuée et interactive, mais aussi des exigences qu’elle impose au réseau de communication sous-jacent : tout d’abord ce support à la simulation distribué doit être capable de supporter l’ensemble du trafic et d’écouler le débit correspondant et ensuite, il doit assurer le transfert des données suffisamment rapide et suffisamment fiable afin d’avoir une simulation cohérente de point de vue simulateur et de point de vue simulation. Cette cohésion se manifeste par la représentation cohérente de l’état de toutes les entités simulées et tout événement survenus.

Deux aspects fondamentaux doivent être pris en compte afin d’assurer cette cohésion : d’une part, la cohérence spatiale qui exige la connaissance de toute occurrence des événements survenant sur les sites distants ; d’autre part, la cohérence spatiale sollicite la perception de l’état des entités distantes à erreur maximale admissible. Outre, pour les applications DIS, la QoS est exprimée au moyen de trois termes :

- Fiabilité ou taux perte de messages maximum admissible, noté τ , qui est fortement lié à l’erreur d’extrapolation maximale admissible pour assurer la cohérence spatiale de la simulation.
- Latence (Délai de transit maximum) admissible sur le réseau, noté DT_{max} , qui permet d’assurer la cohérence temporelle de la simulation.
- Variation maximale de ce délai ou gigue, notée ΔDT .

Le RFC 1667 [SWP94] spécifie ces paramètres selon le couplage entre les entités. Un couplage faible peut survenir si les entités simulées ne sont assez nombreuses et, la distance qui les sépare est suffisamment grande pour tolérer les erreurs de transmissions ; les valeurs admissibles de ces paramètres : $D \leq 300$ ms et $\tau \leq 5\%$.

Un couplage fort se manifeste lorsque plusieurs entités se trouvent dans une zone étroite et, les paquets qu’ils transmettent nécessitent des performances plus grandes pour assurer la cohérence et la consistance de la simulation. Les valeurs admissibles de ces paramètres sont : $D \leq 100$ ms et $\tau \leq 2\%$.

L'expression de la QoS à partir du degré du couplage entre entités présente trois limites principales :

- L1 : s'agissant entre tout couple d'entités présentes dans l'exercice, la détermination du degré de couplage (susceptible d'évoluer dans le temps) paraît coûteuse en terme de temps de calcul.
- L2 : la QoS basée sur le degré du couplage, c'est à dire dépendant de chaque couple (site émetteur, site récepteur), s'accorde mal avec un transport multicast.
- L3 : le standard ignore l'influence du délai de transit (latence) sur l'erreur de la position/orientation et néglige les contraintes de la cohérence spatiale. Nous illustrons les limites de cette troisième contraintes.

Considérant le cas de la figure 1, où deux Sites S_e et S_r appartenant à un même exercice de simulation échangent des informations concernant une entité simulée qu'on note A. Nous focalisons sur le comportement aperçu de A sur le site distant en supposant que l'approche DR est utilisée pour alléger le trafic sur le réseau. La figure 1 illustre l'erreur d'extrapolation de la position de l'entité E_s sur le site émetteur et l'erreur E_r sur le site récepteur.

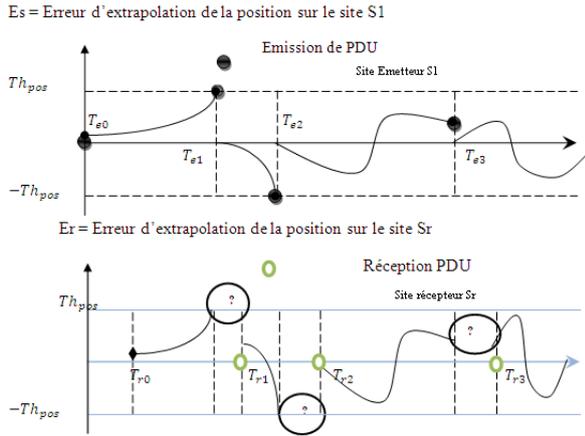


Figure 1. Erreur d'extrapolation de la position

Nous avons choisis, les cercles en noir pour désigner les PDU transmis par le site S_e et, les cercles en couleur gris pour marquer les PDU reçus par le site récepteur S_r . Nous avons choisi pour origine des temps une date d'émission (T_{e0}).

Les dates T_{e0} , T_{e1} , T_{e2} et T_{e3} correspondent aux dates d'émission des PDU et T_{r0} , T_{r1} , T_{r2} et T_{r3} les dates de réception des ces PDU.

- A partir de la date T_{e0} , l'erreur d'extrapolation de la position E_s augmente et atteint la valeur maximale (Th_{pos} ou seuil DR) à la date T_{e1} : le mécanisme DR provoque alors l'émission d'un PDU contenant la position réelle de A : l'erreur E_s s'annule ;
- A partir de T_{e1} , le scénario précédent se reproduit jusqu'à la date T_{e2} ;

- A partir de T_{e2} , l'erreur oscille entre Th_{pos} et $-Th_{pos}$ sans jamais quitter l'intervalle autorisé, au bout du HEART BEAT TIMER, soit $T_{e3}=T_{e2} + 5$ s, le DR émet un PDU et l'erreur revient à zéro.

Analysons l'évolution de l'erreur E_r sur le site récepteur :

- A partir de T_{r0} , l'erreur d'extrapolation de la position E_r commise sur le site récepteur est identique à celle faite au même instant sur le site S_e ; en particulier elle atteint la valeur maximale admissible (Th_{pos} à T_{e1} , date d'émission du PDU de rafraîchissement de de A. Cependant, la mise à jour de la position de A n'est effective que qu'à la date T_{r1} , l'intervalle séparant T_{e1} et T_{r1} correspondant au délai de transit à travers le réseau. Il apparaît donc une indétermination quant à la valeur de E entre les instants T_{e1} et T_{r1} , période durant la quelle E_r peut dépasser la valeur du seuil DR (Th_{pos}) engendrant une violation de la cohérence spatiale ;
- de manière générale, on met en évidence une absence de maîtrise de E_r durant les intervalles de temps indiqués par un point d'interrogation.

Deux questions se posent alors : (1) l'absence de garantie de cohérence spatiale est-elle funeste au déroulement correcte d'une exercice de simulation distribuée interactive ? (2) si oui, peut-on remédier à ce problème, c'est-à-dire maîtriser l'excès transitoire de E_r ?

Concernant la question (1), l'excès transitoire potentiellement observable devient préjudiciable dès lors que le délai de transit DT n'est plus négligeable devant l'intervalle de temps séparant deux réceptions consécutives de PDU DR relatifs à une même entité. Si c'est le cas, les sites récepteurs n'auront plus une vue spatialement cohérente de l'état de l'entité durant la période de temps non négligeable, au pire "presque tout le temps" si la période de rafraîchissement de l'entité d'une entité est voisine au inférieure au délai de transit des PDU qui lui sont associés dans le réseau. Ce risque qui n'apparaît pas dans les réseaux locaux, peut apparaître de façon plus importante dans les réseaux grande distance.

Pour la deuxième question, plusieurs travaux présentaient des moyens de mettre en œuvre la maîtrise de l'excès transitoire : Les auteurs dans [KK06] ont proposé une approche basée sur le filtre de Kalman pour l'estimation des paramètres des entités mobiles dans un réseau ad-hoc afin de réduire le trafic entre les différents nœuds. Bien qu'elle permettait de gagner 10% de la bande passante du réseau, cette approche ne corrige pas efficacement l'erreur d'extrapolation.

Dans [LC05] un algorithme DR flou est appliqué dans une simulation HLA. Cet algorithme prend

en compte le degrés de corrélation flou lors des mesures des relations entre les entités (position, taille, angle de vision,...) et considère une approche multi-niveaux pour estimer le niveau du seuil que l'application doit respecter. Également, une approche Neuro-Reckoning pour améliorer l'extrapolation du prédicateur des entités simulées a été proposée [AMSM07]. Un banc de réseaux de neurones (prédicateur de position, de vitesse, d'orientation et de vélocité) permet d'estimer la nouvelle propriété estimée de l'entité. Nous proposons une approche sous-jacente à ces travaux en utilisant une approche neuro-flou intelligente [JS95] qui permet de réduire considérablement l'erreur d'extrapolation. A la différence de ces propositions, notre approche implique tout d'abord l'abandon de la notion du couplage et ensuite, la détermination de la valeur de l'erreur à partir des seules informations contenues dans chaque PDU. Cette proposition remédie à la la contrainte L1 de manière à garantir à tout instant une erreur maximale admissible, en particulier durant les périodes transitoires : cela permet de remédier à la limite L3. Notre approche est basé sur un algorithme d'inférence neuro-flou adaptatif intelligent qui émule le comportement humain lors de la détermination de l'erreur maximale admissible satisfaisant les contraintes de la QoS évoqués précédemment.

E. ANFIS DEAD RECKONING

1. Modèle théorique

Comme démontré dans la section précédente, l'erreur d'extrapolation de la position sur un site récepteur n'est inférieure au seuil Th_{pos} que dans l'intervalle de temps $[T_{ei}+DT, T_{ei+1}]$. Dans l'intervalle $[T_{ei}+DT, T_{ei}+DT]$, l'erreur peut excéder de manière drastique et conduire à une incohérence.

Une façon de garantir une borne supérieure Er_{max} à l'excès transitoire Er est de disposer d'une garantie de la part du réseau que le délai de transition DT des PDU ne dépasse pas une valeur limite DT_{max} , comme l'illustre la figure 2. Notre approche permet de réduire le délai pendant lequel l'information est utilisable en bornant l'erreur maximale admissible, et ce jusqu'à ce que la nouvelle position soit mise à jour par un nouveau PDU. La position estimer est maintenant calculée par un réseau neuro-flou adaptatif pré-configuré. L'influence du délai de transit DT sur la connaissance de l'écart des entités simulées par chaque site est liée à l'expression de l'écart $e_p(t)$ (en valeur absolue) entre la position réelle $P_A(t)$ de l'entité et sa position extrapolée (extrapolation de second ordre) par le mécanisme du Dead Reckoning $P_{DR}(t)$ (relation 2).

$$e_p(t) = \| P_{DR}(t) - P_A(t) \| \leq E_{max}; E_{max} \geq 0 \quad (2)$$

$$\begin{aligned} e_p(Te_{i+1} + DT) = & \| \int_{Te_i}^{Te_{i+1}} du \int_{Te_i}^{Te_{i+1}} [A_a(\tau) - A_i] d\tau \| \\ & + \| \int_{Te_i}^{Te_{i+1}+DT} du \int_{Te_{i+1}}^{Te_{i+1}+DT} [A_a(\tau) - A_i] d\tau \| + \\ & + \| \int_{Te_i}^{Te_{i+1}+DT} du \int_{Te_{i+1}}^u [A_a(\tau) - A_i] d\tau \| \leq M \quad (3) \end{aligned}$$

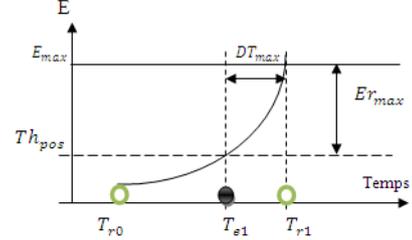


Figure 2. Garantie de l'excès transitoire maximal

A partir de cette relation 2 et 3, il est possible de trouver une valeur de M qui satisfait les contraintes de la QoS décrites précédemment. Néanmoins, même si cette valeur permet de borner l'erreur d'extrapolation, elle n'est pas une solution optimale pour résoudre l'inéquation. Afin d'assurer l'optimalité de la solution, nous proposons dans la suite de ce papier une méthode de calcul de l'erreur basée sur le système d'inférence flou pour le choix du seuil Th_{pos} et des réseaux de neurones adaptatifs permettant d'optimiser l'erreur d'extrapolation. Le modèle d'inférences neuro-flou adaptatif (Adaptive Neuro-Fuzzy Inference System :ANFIS) est basé tout d'abord, sur un système d'inférence flou pour la formulation des termes linguistiques des règles flou ensuite, entraîné via un algorithme d'apprentissage des réseaux de neurones artificiels pour la prise de décision.

Pour simplifier l'illustration du modèle, on se contente d'utiliser le modèle DR de second ordre pour estimer la position d'une entité en se basant sur ces paramètres de corrélation flou (position, vitesse, accélération).

Le système d'inférence flou considéré admet trois entrée (a_1, a_2, a_3); et une seule sortie f (notée aussi O). Dans notre exemple a_1 représente la position, a_2 représente la vitesse et a_3 représente l'orientation. La fonction de mapping qui permet d'avoir la sortie f est donnée par l'équation 4 :

$$f^k = f(a^k) = f(a_1^k, a_2^k, \dots, a_n^k); \quad (4)$$

Avec : $k \in 1..K$ (temps discret) et les ensembles d'entraînement flou sont donnés par la relation 5 :

$$\{(a^1, a^2, a^3, f^1), \dots, (a_1^k, a_2^k, a_3^k, f^k)\} \quad (5)$$

La méthode qui permet de décrire les termes linguistiques, les fonctions d'appartenance et les règles des prémisses consiste à utiliser les règles SI "... ALORS ..." des relations 1. :

$$\begin{cases} R_1 = SIa_1^1 est A_{i1} et \dots et a_1^n est A_{in} ALORS f = z_1 \\ R_2 = SIa_2^1 est B_{i1} et \dots et a_2^n est B_{in} ALORS f = z_2 \\ R_3 = SIa_3^1 est C_{i1} et \dots et a_3^n est C_{in} ALORS f = z_3 \end{cases}$$

Avec : $i = 1, \dots, m$, A_{in} et B_{in} les degrés d'appartenances de la fonction triangulaire et z_i des nombres réels. Nous Supposons que chaque variable linguistique admet 7 termes linguistiques : NB, NM, NS, ZE, PS, PM, PB (Negative Big, Negative Medium, Negative Small, ZERO, Positive Small, Positive Medium, Positive Big), leurs fonctions d'appartenances respectives sont de forme triangulaire (figure 3) et que chaque terme représente un paramètre de corrélation de l'entité dans l'intervall $[-Th_{pos}, Th_{pos}]$.

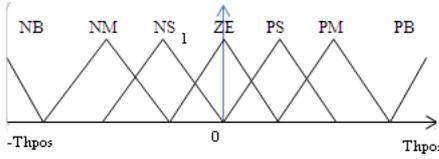


Figure 3. Termes linguistiques de la variable d'entrée x

Le modèle d'inférence flou utilisé est celui de Takagi-Sugeno (figure 4).

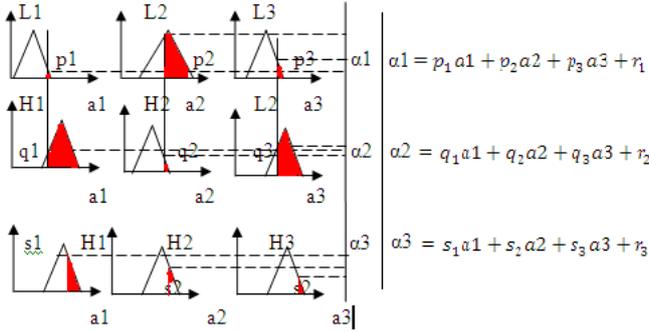


Figure 4. Système d'inférences flou de Takagi-Sugeno avec 3 règles

Les paramètres des prémisses sont utilisés pour réaliser l'apprentissage selon la méthode du gradient descendant. La figure 5 illustre le système ANFIS à une entrée et une sortie.

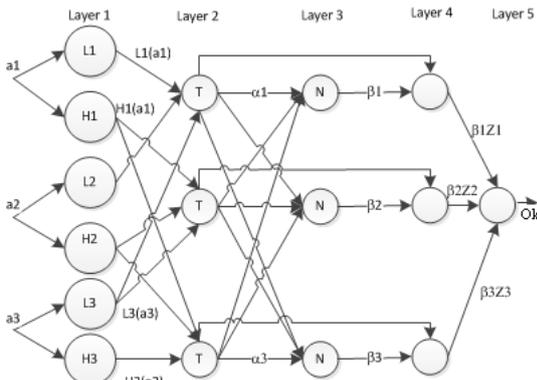


Figure 5. Système ANFIS à une entrée et une sortie

- Couche 1 : La sortie d'un nœud est le degré d'appartenance de l'entrée satisfaisant les variables linguistiques associées à ce nœud (c.f figure 3).
- Couche 2 : chaque nœud réalise une fonction T-norme. Les sorties respectives des nœuds, appelés "rule nodes" (haut, milieu et bas) sont données par les équations suivantes (c.f figure 4) :

$$\alpha_1 = L1(a1) \wedge L2(a2) \wedge L3(a3) \quad (6)$$

$$\alpha_2 = H1(a1) \wedge H2(a2) \wedge L3(a3) \quad (7)$$

$$\alpha_3 = H1(a1) \wedge H2(a2) \wedge H3(a3) \quad (8)$$

- Couche 3 : La sortie des T-normes sont normalisés. Après combinaison linéaire des variables d'entrée dans la couche 4, la sortie prédite est obtenue dans la couche 3 par une moyenne pondérée des sorties des différentes règles, comme le présente les relations 9, 10, 11 :

$$\beta_1 = \frac{\alpha_1}{\alpha_1 + \alpha_2 + \alpha_3} \quad (9)$$

$$\beta_2 = \frac{\alpha_2}{\alpha_1 + \alpha_2 + \alpha_3} \quad (10)$$

$$\beta_3 = \frac{\alpha_3}{\alpha_1 + \alpha_2 + \alpha_3} \quad (11)$$

- Couche 4 : la phase d'apprentissage se réalise dans cette couche. Les données stockées dans une base de données de l'expert sont comparées à celles issues des couches précédentes pour réaliser l'apprentissage hors ligne. Une autre alternative (non discutée ici) correspond à l'apprentissage en ligne et qui consiste à comparer les données au fur à mesure de l'évolution de l'algorithme. La sortie de chacun des trois neurones est le produit normalisé de la couche du tir (niveau apprentissage) et de la règle particulière correspondante, données par les équations 12, 13 et 14 :

$$\beta_1 z_1 = \beta_1 \vee B^{-1}(\alpha_1) \quad (12)$$

$$\beta_2 z_2 = \beta_2 \vee B^{-1}(\alpha_2) \quad (13)$$

$$\beta_3 z_3 = \beta_3 \vee B^{-1}(\alpha_3) \quad (14)$$

- Couche 5 : un seul nœud calcule la sortie du système en réalisant la somme de toutes les entrées, comme le démontre l'équation 15

$$O_k = \beta_1 z_1 + \beta_2 z_2 + \beta_3 z_3 \quad (15)$$

On suppose que nous avons l'ensemble $\{(x_1, y_1), \dots, (x_k, y_k)\}$, avec y_k le vecteur d'état d'une entité simulée stocké dans la base de données et x_k la valeur réelle au cours de la simulation de ce même vecteur à l'instant k . On définit l'erreur quadratique entre de l'entraînement par l'équation 16 :

$$e_p(t, t + DT) = E_k = \frac{1}{2} \times (y_k - o_k)^2 \quad (16)$$

Avec o_k est la sortie calculée du système flou \mathfrak{R} correspondant à la sortie instantanée y_k au instants $k=1,\dots,K$. La méthode du gradient descendant de la relation 17 est utiliser pour l'apprentissage des paramètres des parties conditions et conséquences des règles flous.

$$b_i(t+1) = b_i(t) - \eta \times \frac{\partial E}{\partial b_4} = b_i(t) - \frac{\eta}{b_i^2} \times \delta_k \frac{\alpha_1 + \alpha_2 - \alpha_3}{\alpha_1 + \alpha_2 + \alpha_3} \quad (17)$$

$\delta_k = (y_k - o_k)$ dénote l'erreur. η est la pas d'apprentissage (toujours positif) et t le nombre d'ajustements.

2. Simulation

Cette section présente les résultats de la simulation de l'architecture ANFIS pour le modèle Sugeno de la figure 4 testé sur un exemple simple de la trajectoire de l'entité. ANFIS est utilisé pour modéliser deux fonctions non linéaires ; les résultats sont comparés à ceux achevé par le modèle de retro- propagation du gradient pour un apprentissage hors-ligne. Le comportement de l'entité simulée est stocké dans une base de données d'apprentissage. Le nombre de règles d'inférence flou est sensé être connu au préalable par un expert qui est supposé connaître en avance le système cible. Hors, Nous ne disposons pas de cet expert, le nombre de règle flou est choisi de façon empirique, par l'examen des données et le choix de nombre de test et examen des erreurs. Ceci reste valable puisque le nombre d'entrée est choisi égale à 3. La trajectoire réelle de l'entité et celle donnée par le modèle de haute fidélité du modèle ANFIS sont très proches. Le modèle ANFIS a permis de reproduire (le plus proche possible) le modèle réel. Le tableau 1 illustre l'erreur entre le modèle prédit et le modèle réel. Les résultats sont très prometteuses pour le cas d'une seule entité.

Tableau 1. Erreur d'extrapolation de la position d'une entité

Pas	Erreur
1	0.317
2	0.313
3	0.312
4	0.310
5	0.309
6	0.306
7	0.305
8	0.304
9	0.301
10	0.30

L'erreur demeure proche de 0 et l'algorithme de prédiction converge au bout de 10 itération. En effet, les valeurs initiales des paramètres des prémisses choisi lors de la phase de fuzzification à l'entrée de la couche 1 sont suffisamment bien répartis pour recouvrir la trajectoire de l'entité. Ceci est expliqué par le fait que pour chaque entrée du système ANFIS il existe au moins une condition exhaustive dite ϵ Completeness, qui veut dire que $\mu_A(x) \geq \epsilon = 0,5$. Il y a une assurance que l'entité ne pourra éventuellement pas heurter un obstacle puisqu'il elle suit la trajectoire presque réelle. De plus, il n'y aura plus de message de correction généré par le site l'émetteur, et ce pour les raisons précédentes. Ainsi, la bande passante du réseau ne sera éventuellement pas détériorée, et en plus la latence est gérée par le site récepteur.

F. CONCLUSION

Dans ce travail nous avons présenté un panorama des travaux de recherche que nous avons fait, et nous avons proposés une vue d'ensemble de l'algorithme Dead Reckoning rencontré dans les applications de la simulation interactive distribuée. Nous avons montré son intérêt pour satisfaire les besoins des applications de simulations interactives distribuées. Enfin, nous avons proposé une contribution basée sur les systèmes d'inférences neuro-flou adaptatifs (ANFIS) pour résoudre le problème d'inconsistance et d'incohérence des simulations. Plusieurs idées, protocoles et produits ont déjà implémenté certains algorithmes DR. Cependant, cette contribution fait partie d'un travail en cours et les résultats que nous avons aboutis sont très prometteurs.

RÉFÉRENCES

- [AMSM07] T. WARD A. McCOY and D. DELANEY S. McLOONE. Multistep-ahead neural-network predictors for network traffic reduction in distributed interactive applications. *ACM Transactions on Modeling and Computer Simulation (TOMACS)*, Septembre 2007.
- [IEE00] HLA IEEE-1516. Ieee standard for modeling and simulation high level architecture (hla). <http://standards.ieee.org>, Janvier 2000.
- [JS95] JYH-SHING ROGER JANG and CHUEN-TSAI SUN. Neuro-fuzzy modeling and control. *Proceedings of the IEEE*, Mars 1995.
- [KK06] Seong-Whan Kim and Ki-Hong Ko. Kalman filter based dead reckoning algorithm for minimizing network traffic between mobile nodes in wireless grid. *Springer Berlin*, Octobre 2006.
- [LC05] Gencai Chen Ling Chen. A fuzzy dead reckoning algorithm for distributed interactive applications. *Springer Berlin*, August 2005.
- [MMN04] J.Liu W.Lyou M. McAuliffe, R. Long and D. Nocera. Testing the implementation of dis dead reckoning algorithms. *SISOSTD*, Octobre 2004.
- [OMG07] OMG. Data distribution service for real-time systems (dds). <http://www.omg.org/technology/documents/>, Janvier 2007.
- [SWP94] S. Symington, D. Wood, and M. Pullen. Modeling and simulation requirements for ipng. In *Network Working Group*, <http://tools.ietf.org/html/rfc1667>, 1994.