

Simulation d'un Système Automatisé de Stockage/Déstockage à Convoyeur Gravitationnel par Arena

L.Triqui, F.Kherbouche, Z.Sari

Laboratoire d'Automatique de Tlemcen (L. A.T), Université de Aboubekr Belkaid,
BP 119, Tlemcen 13000, Algerie
E-Mail : z-sari@mail.univ-tlemcen.dz

Résumé : l'un des plus importants critères de performances des systèmes automatisés de stockage/déstockage est le temps de cycle moyen de déstockage. Dans cet article, nous présentons un programme de simulation d'un AS\RS à convoyeur gravitationnel (AS\RS à CG) par Arena. Le but de ce travail consiste à concevoir un AS\RS à CG en spécifiant ses trois fonctions principales à savoir : le stockage, le déstockage, le restockage selon une loi aléatoire afin d'étudier ses caractéristiques et définir les performances intrinsèques du système.

Mots-Clé : système automatisée (Automated Storage/Retievel system (AS\RS)), systèmes Flexibles de Production (FMS), Simulation, Arena.

1. Introduction :

Les systèmes de manutention automatisée font partie intégrante des systèmes flexibles de production, leurs utilisations s'avèrent de plus en plus répandues dans l'industrie, leurs évolutions reposent sur les développements apportés aux véhicules autoguidés et aux systèmes automatisés de stockage/déstockage (AS\RS).

Les AS\RS sont une partie de l'industrie à part qui offre un environnement et un espace contrôlé pour les produits, les équipements et le personnel [1].

Ils sont composés d'un casier de stockage, un moyen de chargement/déchargement, des stations d'entrée/sortie et un système de contrôle.

Ils sont adoptés dans les FMS (flexible manufacturing system) pour diverses raisons : Très grande densité de stockage, accès directs aux charges à 100%, diminution des parcours des chariots élévateurs, minimisation du volume de l'entrepôt par palette, investissement minimum par palette stockée, exigence de main d'œuvre réduite pour diriger les activités, productivité améliorée avec le personnel existant, un mouvement rapide des produits, un minimum de manutention de produits, de plus les AS/RS offrent la flexibilité, la qualité, la précision dans les séquences désirées.

La méthodologie de stockage [1] spécifie l'emplacement du produit dans les zones de

stockage qui se fait selon deux modes, soit par un stockage dédié, soit par un stockage ouvert.

Dans le stockage dédié chaque produit possède son propre emplacement de Stockage, contrairement au stockage ouvert, dans ce mode le stockage des produits se fait de manière aléatoire et un système de commande permettant de mémoriser les emplacements des différents produits et les cases libres. Les études consacrées à la conception et le fonctionnement des AS\RS pendant plusieurs années a différencié plusieurs type d'AS\RS adaptée aux variances des produits en stockage, selon les dimensions et les volumes. Qu'on peut les citer comme suit [2] : AS\RS à charge unitaire, AS\RS à mini charge, AS\RS avec personne à bord, AS\RS à rack glissant, AS\RS à convoyeur gravitationnel.

Dans cet article nous présenterons un programme de simulation permettant de représenter les fonctions principale d'un AS\RS à CG.

2. Littérature :

Les performances des systèmes automatisés de stockage/déstockage (AS\RS) ont fait l'objet de plusieurs travaux, basés principalement sur la simulation. L'étude des AS\RS est relativement récente, Bozer et White [4] ont développé une conception de la forme du rack basée sur des techniques analytiques.

Le point de repos de la machine (S\R) a été largement étudié dans l'analyse des AS\RS.

Egbelu [5] a étudié les points de repos de la machine (S\R) dans les AS\RS, en développant ainsi des formulations pour minimiser le temps de cycle et le temps de réponse maximum, Hwang et Lim[6] ont montré que la formulation d'Egbelu pourrait être transformée en un problème d'emplacement de la station pour améliorer le temps de réponse. Egbelu et Wu [7] ont utilisé la simulation pour comparer les règles de positionnement du point de repos, en particulier, ils ont comparé les deux formulations d'Egbelu et les quatre règles statiques de Bozer et White.

Peter et al [8] ont développé une solution analytique pour la position du point de repos sous une variété de configurations d'AS\RS concluant que le meilleur point de repos de la machine se trouve au milieu du rack.

La modélisation du temps de cycle de la machine S\R est devenue l'un des plus importants domaines de la recherche dans l'étude des AS\RS. Bozer et White ont développé des expressions analytiques pour les temps moyens des simple et double cycles d'un AS\RS. Bien que les résultats ont été établis au cas spécifique des AS\RS à charge unitaire, le travail de Bozer et White est devenu une source d'inspiration pour le développement d'autres expressions pour différents types d'AS\RS

Sari et al [3] ont établi l'expression du temps de cycle moyen d'un AS\RS à convoyeur gravitationnel, pour quatre différentes positions du point de restockage et P\D.

3. Description du système AS\RS à CG :

Dans un certain nombre d'entreprises, on cherche à optimiser le stockage d'un grand nombre de produits : matière première, bruts, composants, outillages, produits finis ou semi-finis... Le stockage horizontal (charge unitaire, charge réduite ou mini charge) revient cher par la surface au sol qu'il mobilise (stockage pur mais aussi les allées permettant à un chariot élévateur d'accéder partout) que par sa difficulté de gestion (car le contenu du stock change sans arrêt car ce ne sont que des stockages temporaires).

Une solution intéressante est le stockage à convoyeur gravitationnel qui peut être représenté par :

a) Un Rack :

Un rack (de hauteur H, largeur L et profondeur D) composé de casiers (dont le support est un convoyeur gravitationnel, CG)

comportant chacun M emplacements, disposés en matrice multicouche composée de Nl casiers horizontaux, Nh casiers verticaux et M couche.

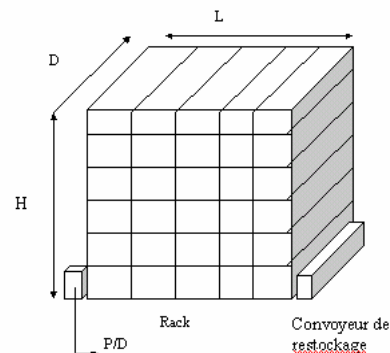
b) Deux machines S/R parallèles :

L'une sur la face avant du rack (MS) pour le stockage et l'autre sur la face arrière du rack (MD) pour le déstockage, les machines peuvent avancer/reculer sur les Nh cases, monter/descendre aux Nv étages, elles peuvent aussi accéder aux postes d'entrée-sortie du stock.

c) Un convoyeur du restockage :

C'est un convoyeur accumulateur c'est à dire : il retient le produit jusqu'à ce que le nombre de produit soit inférieur ou égale à la moitié de la capacité total de la file et alors le produit est libéré vers sa destination, en plus il sert de liaison entre les deux faces du rack.

Rq : le point de dépôt est sur la face avant du rack et le point de retrait est sur la face arrière du rack.



4. Fonctionnement :

Pour stocker un produit la machine MS le récupère du point repos et le dépose aléatoirement dans l'un des casiers de la face avant du rack, le produit glisse sur un convoyeur gravitationnel et se positionne dans l'emplacement libre de la face arrière du rack.

Le convoyeur gravitationnel est utilisé pour minimiser l'espace de passage et garantir les opérations FIFO (First In/ First Out).

Pour le déstockage d'un produit déjà positionné dans l'emplacement d'un casier, la machine MD désigne la cellule aléatoirement déstocke successivement tous les produits qui précèdent et les dépose au point de restockage ou ils seront stockés de nouveau aléatoirement dans l'une des cellules sur la face avant du rack par la machine MS, jusqu'à ce qu'elle

recupère le produit désiré et le dépose au point de livraison.

4.1 La notation utilisée pour définir le système :

$E(SC)$: le temps moyen de stockage.

$E(RC)$: le temps moyen de déstockage.

L, H, D : longueur, hauteur, profondeur du rack.

l, h, d : Longueur, hauteur, profondeur de la cellule.

M : nombre d emplacement dans un casier.

m : rang de couche.

Nl : nombre de casiers horizontaux dans le rack.

Nh : nombre de casiers verticaux.

tv : temps de déplacement vertical depuis le point P/D jusqu'à la dernière ligne.

tv' : temps de déplacement du casier au suivant.

th : temps de déplacement horizontal depuis le point P/D jusqu'à la dernière Cologne

th' : temps de déplacement horizontal d'un point au point suivant.

T : facteur de normalisation

b : facteur de forme.

ϕ : taux de remplissage.

5. Description du système :

Le système a étudié comportant :

Deux ensembles de stations : chaque ensemble est composé de 36 stations réparties en six lignes placées l'une au-dessus de l'autre et chaque ligne est constituée de six stations placées l'une à côté de l'autre. Les deux ensembles sont reliés par des files d'attente à M emplacement.

Plus :

- Une station d'entrée (dépôt) : elle reçoit les entités pour les stocker.
- Une station de sortie : Elle reçoit les entités pour la livraison.
- Une station de restockage : elle se constitue de deux stations (acce, resto) qui sont reliées entre eux par une file d'attente à M emplacement.

6. Modélisation du temps de cycle pour L'AS/RS à CG:

Considérons un AS/RS à CG composé d'un rack de largeur L de hauteur H , et de profondeur P . La largeur, la hauteur et la profondeur de chaque emplacement sont respectivement l , h et p . De ce fait, le rack est composé de Nl casiers horizontaux, Nh casiers verticaux, et M emplacement Le point

de repos de la machine de déstockage se trouve au milieu du rack.

Le comportement de la machine de stockage dans le système AS/RS à CG est le même que dans le système classique, ainsi le temps moyen de stockage reste inchangé par rapport au AS/RS à charge unitaire. Cependant, l'emplacement des points de dépôt/livraison et le convoyeur de restockage joue un rôle dans la détermination de l'expression des temps moyens de déstockage.

Sari et al [3] ont donné l'expressions du temps moyen de cycle de déstockage de la machine S/R pour le convoyeur de restockage et le point de dépôt\livraison sont aux coins bas du rack :

$$E(RC) = T \left(\frac{b^2}{3} + 1 \right) \left(\frac{1}{4} + \frac{1}{2} \rho M \right)$$

7. Simulation par ARENA :

Afin d'étudier l'emplacement des différentes positions des stations P/D et le point de restockage on a modélisé notre système et simulé par ARENA qui est parmi les logiciels les plus récents conçus pour la simulation des systèmes flexibles de production.

Le système ARENA est un support de modélisation intégré permettant de construire des modèles plus ou moins détaillés pour un large éventail d'applications possibles. ARENA intègre également toutes les fonctions liées à la simulation, telles que l'animation, l'analyse des données d'entrée, la vérification du modèle et l'analyse de résultats, dans un seul environnement de modélisation.

7.1 Structure d'ARENA :

Modéliser en ARENA représente toutes les activités ou situations par des blocs qui sont reliés logiquement par des liens, ces blocs sont renseignés par des données. Tous les blocs se trouvent stockés dans des bibliothèques du logiciel. Chaque bloc est constitué des éléments suivants :

- Une icône représentative de l'activité représentée.
- Des connecteurs pour assurer la liaison entre les blocs.
- Une boîte de dialogue.
- Un programme.
- Un fichier d'aide

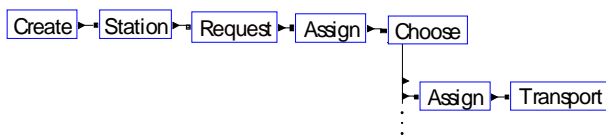
Chaque système de fabrication a son modèle de simulation spécifique et par conséquent il a ses propres paramètres et données. Dans un programme de simulation d'un système de

production l'entité doit être créée, transformée puis détruite

8. Modèle logique :

Notre travail consiste à concevoir et simuler un AS/RS à CG par ARENA en spécifiant trois fonctions principales : le stockage, le déstockage, le restockage selon une loi aléatoire afin d'étudier ses caractéristiques et définir les performances intrinsèques du système en utilisant certains modules d'ARENA tels que : Create, Station, Request, Transport, Scan, pour expliquer la chaîne de stockage, le déstockage et le restockage.

8.1. Pour le stockage :



→ **Create** : Ce module se charge de la création des entités qui entrent au système selon une loi de distribution exponentielle. Dans notre cas, il se charge de la création de cinq produits différents, donc on a utilisé cinq modules de *create* et on a assigné à chacun un attribut qui spécifie le type de produit et un ensemble d'images dont chaque image représente un type d'entité.

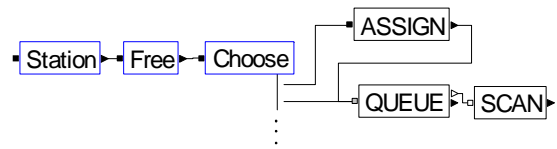
→ **Station** : pour représenter la station d'entrée du système, c'est-à-dire tous les produits à stocker se trouvent dans cette station.

→ **Request** : Pour appeler le transporteur 1 (machine de stockage MS) qui transporte les charges selon une loi aléatoire qui est déclarée dans le bloc *Assign*.

→ **Assign** : Dans ce bloc, on déclare une variable qui génère des nombres aléatoires.

→ **Choose** : On déclare un vecteur $TAL(i)$ qui comporte 36 éléments ($i = 1..36$) ; dont chaque élément désigne le nombre d'une station.

→ **Transport** : Une fois que la valeur donnée par *RND* correspond à la valeur de $TAL(\text{indice})$, l'entité est transportée vers sa destination, c'est-à-dire l'une des stations du premier ensemble.



→ **Station set** : Pour représenter l'ensemble des stations où les entités se logent avant d'être déstockées.

→ **Free** : ce module sert à libérer le transporteur 1 au niveau des stations.

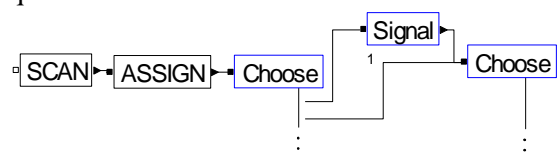
→ **Choose** : Pour tester les emplacements libres dans les stations et si une cellule est pleine, il change la station aléatoirement jusqu'à ce qu'il trouve un emplacement libre.

→ **Queue blocs** : Queue représente l'ensemble des files d'attente assignées à chaque station avec une capacité finie égale à *capa*, dans notre exemple, elle est égale à huit (8).

→ **Scan** : pour bloquer les entités dans les files d'attente jusqu'à ce que certaines conditions soient vérifiées (le taux de remplissage est égal à 75% de la capacité du système global).

8.2. Pour le déstockage :

Le déstockage sera déclenché lorsque les conditions de scan seront vérifiées, c'est-à-dire que :



- Le taux de remplissage est égal à 75% de la capacité du système global.

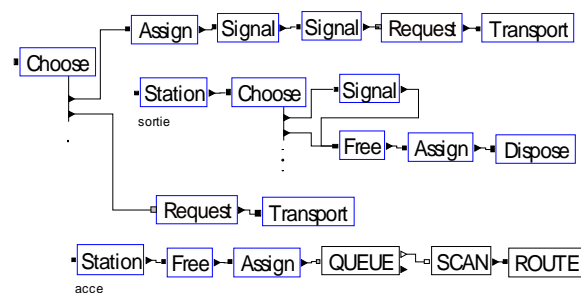
- Le déstockage doit se faire aléatoirement, donc pour déstocker un produit, il faut désigner la file d'attente et le rang aléatoirement.

➤ Pour la file d'attente, on déclare une variable appelée 'file' dans le module Variable qui génère des valeurs selon l'expression : $File = ANINT((RA * 35) + 1)$ avec 35 : (nombre de station - 1)

➤ Pour le rang, on déclare une variable appelé 'RANG' dans le module Variable qui génère des valeurs selon l'expression : $RANG=(ANINT(RA*7)+1)$ avec $7 = \text{capa}-1$.

- Une fois la file et le rang sont trouvés on teste le nombre d'entité dans cette file :

➤ Si le nombre d'entité est inférieure au rang on envoie un signal pour changer la file d'attente jusqu'à ce qu'on trouve une file qui a le nombre d'entité supérieur ou égale au rang donné par la variable aléatoire.



➤ A ce niveau on teste le rang par le module *choose* s'il correspond au rang désigné par la règle aléatoire on appelle le transporteur 2 par *request* pour livrer le produit a la station de sortie, Sinon on les transporte vers la station de restockage (acce).

➤ L'opération se répète après chaque entité sortante avec émission de deux signaux :

➤ l'un change la valeur de la file (fig 1)

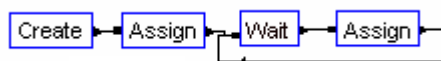


Fig 1

➤ et l'autre change la valeur du rang (fig 2)

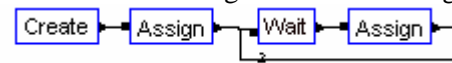
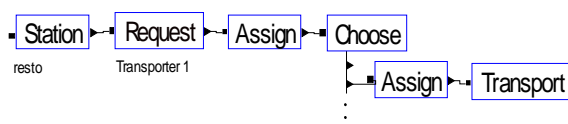


fig 2

➔ **wait** : ce module place et bloque les entités dans une file d'attente interne jusqu'à ce qu'il reçoive le signal adéquat par le module signal.

➔ **Signal** : Quand une entité arrive à ce bloc, le code du signal est envoyé et a ce moment l'entité est libérée de la file d'attente interne de wait jusqu'à l'interruption du signal par un délai ou une instruction.

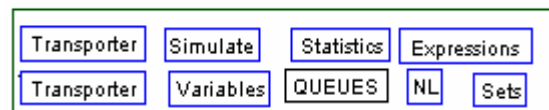
8.3. Pour le Restockage :



➤ Les produits sont déposés dans une file d'attente qui relie la station 'acce' jusqu'à ce que le nombre de produit soit inférieur ou égale à la moitié de la capacité total de la file, alors on appelle le transporteur 1 pour stocker à nouveau les produits selon la même règle aléatoire et le cycle reprend.

➔ Déclaration :

Certain module ne fonctionnent pas sans être déclarés donc on a utilisé quelque module tel que : Variable, Expressions, QUEUES.



➔ **Transporter** : On déclare deux transporteur , le premier pour les opérations de stockage et restockage , le deuxième se charge du déstockage.

Les deux transporteurs ont la même vitesse et ce déplace selon des chemins bien déterminés.

➔ **NL** : Permet de définir les intersections et les chemins des transporteurs, en plus, il fait la connections entre les différentes stations de stockage.

➔ **Set** : Pour déclarée les différents ensembles :

- Ensemble des stations :
- Ensemble des images
- Ensemble des files d'attente.

➔ **Variable** : Déclaration des différentes variables du système telles que : capa , file, rang....etc.

➔ **Expressions** : Pour formuler les expressions du système exp. : $\text{file} = ANINT((RA*35)+1)$.

➔ **Statistics** : Ce module est utilisé pour définir les informations collectées durant la simulation et spécifie les sortie et les observations concernant le model simulé.

➔ **QUEUES** : Pour déclarer les différentes files d'attente du système en spécifiant la rotation par FIFO (first in first out).

9. Conclusion:

Dans cet article nous avons présenté un programme de simulation que nous avons développés pour la simulation d'un système automatisé de stockage\déstockage à convoyeur gravitationnel en spécifions ces trois fonctions principales: le stockage, le déstockage, le restockage selon une loi aléatoire . Nous nous sommes intéressés plus particulièrement au fonctionnement aléatoire qui reste riche en phénomènes complexes car l'aléatoire ne possède pas de loi. Ceci simplifie la tâche aux future étudiants qui feront des études dans ce domaine.

10. Reference:

- [1] ASKIN, R. G., STANDRIDGE, C. R., (1993) Modeling and Analysis of Manufacturing Systems. John Wiley & Sons.
- [2] SINGH, N., (1996) Computer -Integrated Design and Manufacturing
- [3] SARI, Z., CHETTO, H., GHOUALI, N., Travel time models of a flow rack automated storage\retrieval system. Submitted to computer and Industrial Engineering, (1999).
- [4] BOZER, Y.A. WHITE, J. A., Travel time models for automated storage\retrieval system. IIE Transactions, Vol. 16, 4, pp.329-338, (1984).
- [5] P.J Egbelu, Framework foe dynamic positioning of storage/retrieval system, International Journal of Production Research, 29(1), pp. 17-37, (1991).
- [6] H. Hwang, J.M. Lim, Deriving an optimal dwell point of the storage/retrieval systems, International Journal of Productions Research,31, 11, pp. 2591-2602, (1993).
- [7] P.J Egbelu, C. T. Wu, A comparison of dwell point rules in storage/retrieval systems. International Journal of Productions Research, 31(11),pp. 2515-2530, (1993)
- [8] B.A. Pters, J.S. Smith, T.S. Hale, closed form models for optimal dwell point location in automated storage and retrieval systems. International Journal of Productions Research, 34(6), pp.1757-1771, (1996).