

Développement de modules pour un nouveau système de gestion de la relation client basé sur le web et h5

CHEN, SHURUI

Branche : Informatique et Systèmes d'Information

Responsable pédagogique UTT :
???????

Semestre : Automne 2021

Résumé (150 mots)

J'ai effectué un stage de 6 mois au sein du département de fabrication intelligente de CONTEMPORARY AMPEREX TECHNOLOGY CO., LIMITED (CATL). Mon stage portait sur les deux domaines principaux suivants :

1. Aider l'équipe à développer des algorithmes d'apprentissage profond pour les types de traitement d'images. L'objectif est de donner à la ligne de production la capacité de détecter automatiquement les défauts.
2. Simulation et optimisation des AGV (AUTOMATED GUIDED VEHICL) utilisés dans l'usine à l'aide d'un logiciel de "plant simulation". L'objectif est d'améliorer l'efficacité des AGV dans l'usine.

Les connaissances informatiques que j'ai appliquées comprennent les langages de programmation : "PYTHON" et "JAVA", le langage de programmation de simulation "SIMTALK 2.0", la bibliothèque de traitement d'images "OPENCV", les algorithmes d'apprentissage profond et certains algorithmes de parcours, tels que A STAR.

Entreprise : CATL

Lieu : Fujian, Chine

Responsable :

Mots clés (CF Thésaurus)

- Logiciels de groupe
- Optimisation mathématique

Remerciement

Tout d'abord, je tiens à remercier M. Zhang Wei, directeur du département de fabrication intelligente de CATL, de m'avoir accepté comme stagiaire dans ce département.

Je tiens à remercier mes deux superviseurs de stage, Mme Lu Chun Wen et M. Lin Li Yong, pour leur aide tout au long de mon stage.

Je tiens également à remercier sincèrement l'équipe du système de répartition AGV, qui s'est montrée très sincère et amicale et qui était prête à m'aider à résoudre mes problèmes à tout moment.

Je voudrais adresser aussi ma profonde gratitude au groupe d'innovation émergente de SENSETIME pour son soutien dans mon travail. Bien que nous appartenions à des entreprises différentes, nous travaillons très bien ensemble dans notre travail. Je suis également très reconnaissant au Dr Ma Zheng pour ses conseils en matière d'apprentissage profond.

Je remercie bien évidemment notre responsable de stage François Iotti. Elle m'a vraiment aidé à résoudre de nombreux problèmes sur mon stage.

Je tiens à remercier tout le monde à CATL et tout le personnel qui m'a aidé. Je me souviendrai toujours de leur enthousiasme et de leur responsabilité.

Pour finir, je remercie ceux qui m'ont beaucoup appris au cours de ce stage, et même ceux qui ont eu la gentillesse de faire de ce stage un moment très profitable. Chacune de ces personnes ont rendu mon stage passionnant.

merci!

Sommaire

Remerciement	2
I. Présentation	4
I-1. Présentation de l'entreprise	4
CATL	4
I-2. Présentation du département	6
IMD (Intelligent Manufacturing Department)	6
I-3. Présentation de l'équipes	7
ILT (Intelligent Logistics Team)	7
EIG (Emerging Innovation Group)	7
II. Stage	8
II-1. Détection des défauts	8
II-1.1. Informations générales	8
II-1.2. Mes réalisations	11
II-2. Simulation de systèmes AGV	22
II-2.1. Informations générales	22
II-2.2. Mes réalisations	25
Conclusion	42
Bibliographie	43

I. Présentation

I-1. Présentation de l'entreprise

CATL

Fondée en 2011, CATL (CONTEMPORARY AMPEREX TECHNOLOGY CO., LIMITED) se concentre sur la recherche, le développement, la production et la vente de systèmes de batteries d'alimentation de véhicules à énergie nouvelle et de systèmes de stockage d'énergie. Elle s'engage à fournir des solutions de premier ordre pour les nouvelles applications énergétiques dans le monde entier. Ses technologies de base comprennent les capacités de développement et de fabrication de l'ensemble de la chaîne industrielle dans le domaine des batteries de puissance et de stockage d'énergie, telles que les matériaux, les batteries, les systèmes de batteries et le recyclage des batteries secondaires. [1]

CATL possède plus de 2 400 technologies brevetées et plus de 5 000 employés de R&D, dont au moins 2 000 maîtres. [2] En 2019 et 2018, elle a été classée parmi les "entreprises les plus innovantes de Chine" par Forbes Chine et s'est vu décerner le titre d'"entreprise de référence en matière de fabrication intelligente" par le ministère de l'Industrie et des Technologies de l'information de Chine.



Figure 1 Organigramme de l'entreprise

CATL dispose actuellement de quatre grands centres de R&D (1. Ningde, Fujian, 2. Liyang, Jiangsu, 3. Shanghai 4. Munich, Allemagne) et de cinq bases de production (1. Ningde, Fujian, 2. Xining, Qinghai, 3. Liyang, Jiangsu, 4. Yibin, Sichuan 5. Allemagne : Erfurt) [3], et a établi une coopération avec un certain nombre de grands constructeurs automobiles nationaux. Elle est devenue une entreprise de premier plan dans le secteur des nouvelles énergies en Chine.

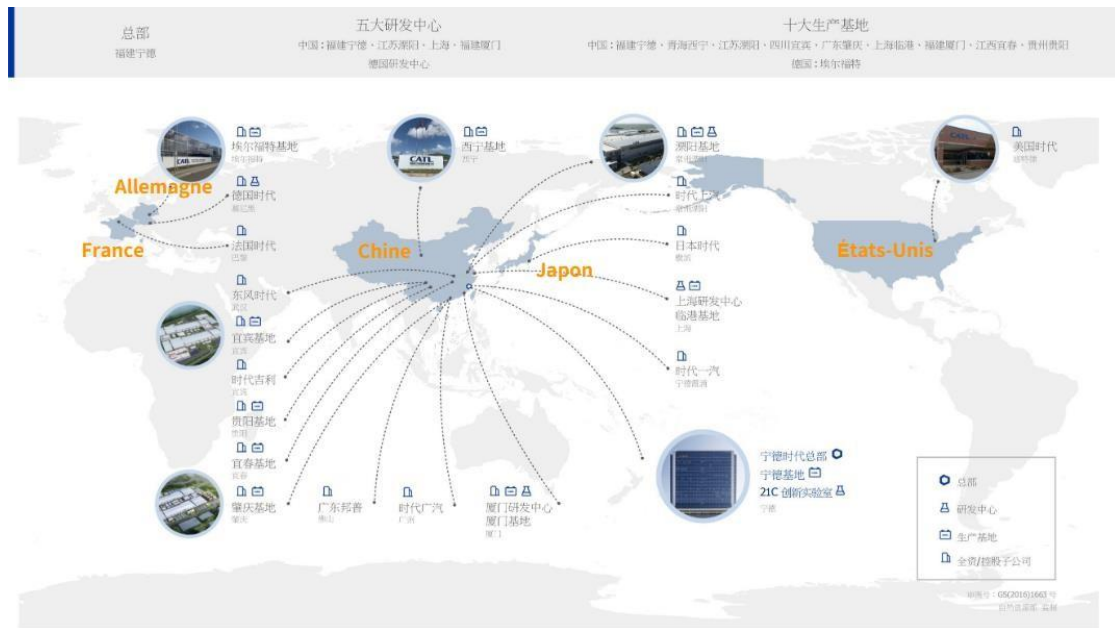


Figure 2 Distribution CATL

I-2. Présentation du département

IMD < Intelligent Manufacturing Department >

En mars 2017, CATL a élaboré un modèle de maturité de fabrication intelligente et un plan de développement. En août 2020, le département de fabrication intelligente (IMD) a été mis en place, visant à réduire les coûts de main-d'œuvre, à améliorer la qualité des produits et à augmenter le taux d'excellence grâce à l'intelligence (intelligence externe / intelligence autonome), et à atteindre à terme un système de fabrication intelligente à l'échelle de l'usine. [4]



Figure 3 Composition du secteur IMD

La division Fabrication intelligente est divisée en quatre sections principales. 1. Solutions intelligentes 2. Intelligence artificielle 3. Big Data 4. Détection intelligente.

Les Groupes orange sont ceux auxquels j'ai participé pendant mon stage.

I-3. Présentation de l'équipes

ILT (Intelligent Logistics Team)

ILT se concentre sur le développement de divers systèmes logiciels avancés de logistique, y compris le développement de divers systèmes de planification et le jumelage numérique des données logistiques. Nous construisons des plateformes d'usine définies par logiciel en combinant des calculs d'optimisation à grande échelle, des méthodes formelles, des big data et des algorithmes d'intelligence artificielle. Le résultat final est la construction automatique de plans, la génération automatique de codes de commande, l'ordonnancement en temps réel et la commande tolérante aux pannes, ainsi que l'apprentissage adaptatif. Aidez CATL à construire enfin une usine sans personnel et intelligente.

EIG (Emerging Innovation Group)

De SenseTime, Le EIG se concentre sur la mise en œuvre de l'intelligence artificielle dans des scénarios industriels. Ils espèrent combiner les capacités algorithmiques, les capacités d'ingénierie et les scénarios commerciaux, afin que l'IA puisse réellement renforcer la production industrielle, insister sur l'originalité et laisser l'IA guider le progrès humain. Leurs algorithmes d'IA ont réalisé une percée qualitative en matière de précision pour la détection des défauts des batteries d'énergie nouvelle et ont intégré leurs capacités algorithmiques dans un système qui permet aux clients finaux de réaliser une production de masse à faible coût avec une précision garantie. [5]

II. Stage

Mon sujet initial était le développement de la vision par ordinateur et depuis que j'ai rejoint le CATL, j'ai été responsable de deux tâches principales, qui ont toutes deux sont très enrichissantes, c'est pourquoi j'ai divisé mon sujet principal en deux parties ici.

II-1. Détection des défauts

II-1.1. Informations générales

A. Concepts et scénarios

L'une des étapes du processus de production d'une batterie consiste à souder le boîtier en aluminium au couvercle supérieur de la batterie. À ce stade, nous devons inspecter les cordons de soudure afin de détecter toute soudure défectueuse.

Voici quelques-uns des types de défauts dans les soudures prises avec une caméra 2D.

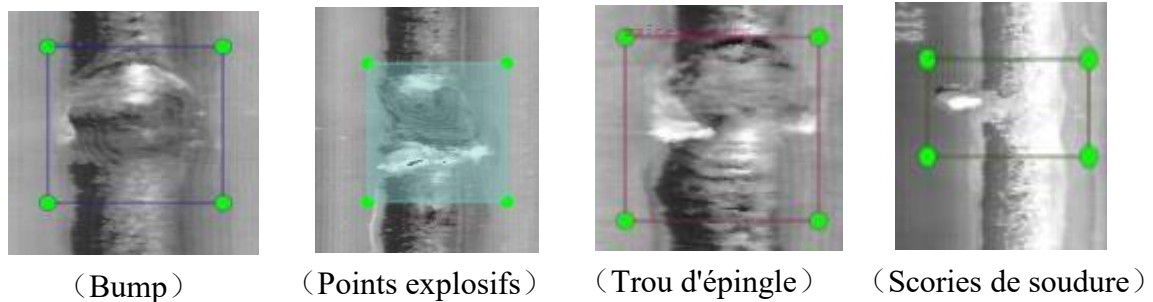


Figure 4 Différentes cartes de défauts

Au total, nous avons sept défauts : Bump, Soudure cassée, Points explosifs, Lumière polarisée, Trou d'épingle, Cratère, Scories de soudure. [6]

Dans le passé, nous avons utilisé une inspection visuelle manuelle pour filtrer les batteries défectueuses. Nous espérons maintenant utiliser la vision par ordinateur pour résoudre ce problème et améliorer la précision de l'inspection.

B. Terminologie

Avant de pouvoir comprendre l'ensemble du processus, nous devons connaître une certaine terminologie :

OK : Le produit est considéré comme normal par un objet (personne ou système de détection).

NG : Le produit est perçu comme défectueux par un objet (personne ou système de détection).

Overkill : Les produits normaux sont jugés par l'objet comme étant défectueux et ne passent pas le test.

Misskill : Les produits défectueux sont jugés par l'objet comme étant normaux et passent le test.

Taux d'overkill : Nombre d'overkill divisé par le total.

Taux d'misskill : Nombre d'misskill divisé par le total. [7]

Deep learning : Dans ce scénario, il peut être compris comme un réseau de neurones.

Réseau de neurones : Un réseau neuronal est un modèle opérationnel constitué d'un grand nombre de nœuds (ou "neurones") et d'interconnexions entre eux. [8] Il ajustera les poids entre les différents nœuds sur la base de l'ensemble de données original, de sorte que le réseau maximise l'ajustement à nos données. Nous utilisons cette méthode comme un classificateur d'images.

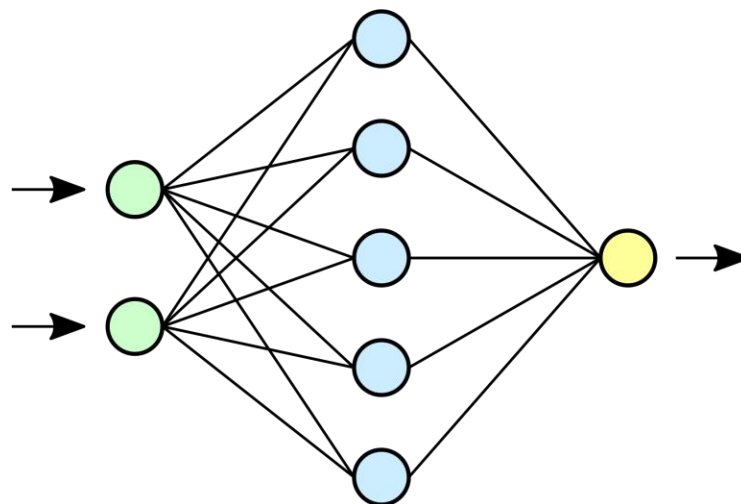


Figure 5 Carte conceptuelle du réseau neuronal

C. L'ensemble du processus

1. Nous devons définir le type de défaut, ici il y a sept types de défauts : Bump, Soudure cassée, Points explosifs, Lumière polarisée, Trou d'épingle, Cratère, Scories de soudure. (Pour différentes situations, il peut y avoir différents types de défauts, il faut donc les définir à l'avance.)
2. Importation dans le modèle d'ensembles de données annotées manuellement depuis la chaîne de production.
3. Utilisez cet ensemble de données pour appliquer un réseau neuronal afin de former un modèle.
4. Des tests préliminaires du modèle formé sont effectués à l'aide de jeux de données existants.
5. Le modèle a été mis en ligne de production pour être testé.
6. De nouvelles données sont constamment acquises pour améliorer le modèle et en augmenter la précision.

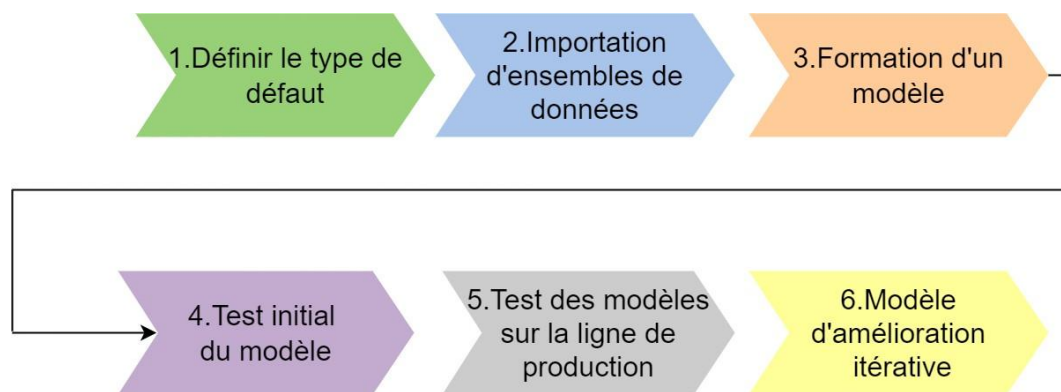


Figure 6 Diagramme de flux réel

D. Nos objectifs

L'utilisation du modèle dans la chaîne de production a permis de réduire le taux d'overkill du modèle à moins de 0,5 % et le taux de misskill est proche de 0,0 %. À terme, le modèle peut être utilisé pour remplacer tous les inspecteurs visuels de la qualité sur la chaîne de production. [9]

II-1.2. Mes réalisations

A. Logiciel de recherche de défauts

J'ai réalisé moi-même le développement et les tests du logiciel, pour lequel le Dr Ma Zheng m'a fourni de nombreuses suggestions d'amélioration, notamment l'adaptation de la langue chinoise et la sélection de chemins automatiques. La CATL m'a fourni un bon environnement de test.

B. Contexte

Lorsque l'on constate un overkill ou un misskill sur la ligne de production, il est nécessaire d'interroger les informations d'inspection de la batterie dans le modèle et d'observer l'image du défaut identifié par la machine. Ainsi, grâce à ce logiciel, la carte d'inspection des défauts d'une batterie ou de certaines batteries peut être automatiquement interrogée à partir des diagrammes 2D, 3D et des GLOG donnés, permettant ainsi au personnel de terrain de comparer l'image avec la réalité en temps réel.

a) Qu'est-ce qu'un fichier GLOG

Lorsque nous utilisons le modèle pour analyser les batteries de la chaîne de production, les mesures du modèle se traduisent par un fichier texte au contenu important. Chaque fois que le modèle filtre une batterie, il écrit dans ce fichier. Il est très grand et pas facile à observer.

```
I0910 04:35:34.997028 140 ind_api_impl.cpp:1412] fanbian: 0.177328
I0910 04:35:34.997028 140 ind_api_impl.cpp:1413] pianguang_length: 1.8
I0910 04:35:34.997028 140 ind_api_impl.cpp:1414] pianguang_length_harsh: 1.05
I0910 04:35:34.997028 140 ind_api_impl.cpp:1415] bolangbian_length: 0.63
I0910 04:35:34.997028 140 ind_api_impl.cpp:1416] bolangbian_length_harsh: 0
I0910 04:35:34.997028 140 ind_api_impl.cpp:1417] num results: 2
I0910 04:35:34.997028 140 ind_api_impl.cpp:1419] label: 2 conf: 0.897949 roi: 128 2191 259 2473 area: 4.73664 mid_to_low: -0.846669 mid_to_high: 0.556856 width: 0.177328 height: -0.847425
I0910 04:35:34.997028 140 ind_api_impl.cpp:1419] label: 3 conf: 0.9 roi: 0 398 562 2507 area: 0 mid_to_low: 0 mid_to_high: 0 width: 0 height: 0
I0910 04:35:34.997028 59476 ind_api_impl.cpp:1484] Load image from file: C:\software\0824-1\build\TempImage\001C86H0000001B9A7103787\001C86H0000001B9A7103787_CAM4_9096_B2_EDGE4.bmp
I0910 04:35:35.012650 59476 ind_api_impl.cpp:1489] Load image from file: C:\software\0824-1\build\TempImage\001C86H0000001B9A7103787\001C86H0000001B9A7103787_CAM4_9096_B2_EDGE4.bmp done.
I0910 04:35:35.012650 59476 ind_api_impl.cpp:1484] Load image from file: C:\software\0824-1\build\TempImage\001C86H0000001B9A7103787\001C86H0000001B9A7103787_CAM4_9096_B2_EDGE4.tif
I0910 04:35:35.028271 59476 ind_api_impl.cpp:1489] Load image from file: C:\software\0824-1\build\TempImage\001C86H0000001B9A7103787\001C86H0000001B9A7103787_CAM4_9096_B2_EDGE4.tif done.
I0910 04:35:35.106379 59476 ind_api_impl.cpp:1411] filename: C:\software\0824-1\build\TempImage\001C86H0000001B9A7103787\001C86H0000001B9A7103787_CAM4_9096_B2_EDGE4.bmp
I0910 04:35:35.106379 59476 ind_api_impl.cpp:1412] fanbian: 0.0484033
I0910 04:35:35.106379 59476 ind_api_impl.cpp:1413] pianguang_length: 0
I0910 04:35:35.106379 59476 ind_api_impl.cpp:1414] pianguang_length_harsh: 0
I0910 04:35:35.106379 59476 ind_api_impl.cpp:1415] bolangbian_length: 0
I0910 04:35:35.106379 59476 ind_api_impl.cpp:1416] bolangbian_length_harsh: 0
I0910 04:35:35.106379 59476 ind_api_impl.cpp:1417] num results: 0
I0910 04:35:39.246035 22188 ind_api_impl.cpp:1484] Load image from file: C:\software\0824-1\build\TempImage\001C86H0000001B9A7103781\001C86H0000001B9A7103781_CAM2_9096_B1_EDGE2.bmp
I0910 04:35:39.261684 22188 ind_api_impl.cpp:1489] Load image from file: C:\software\0824-1\build\TempImage\001C86H0000001B9A7103781\001C86H0000001B9A7103781_CAM2_9096_B1_EDGE2.bmp done.
I0910 04:35:39.261684 22188 ind_api_impl.cpp:1484] Load image from file: C:\software\0824-1\build\TempImage\001C86H0000001B9A7103781\001C86H0000001B9A7103781_CAM2_9096_B1_EDGE2.tif
I0910 04:35:39.308548 22188 ind_api_impl.cpp:1489] Load image from file: C:\software\0824-1\build\TempImage\001C86H0000001B9A7103781\001C86H0000001B9A7103781_CAM2_9096_B1_EDGE2.tif done.
I0910 04:35:39.495977 22188 ind_api_impl.cpp:1411] filename: C:\software\0824-1\build\TempImage\001C86H0000001B9A7103781\001C86H0000001B9A7103781_CAM2_9096_B1_EDGE2.bmp
I0910 04:35:39.495977 22188 ind_api_impl.cpp:1412] fanbian: 0.0878423
```

Figure 7 Schéma GLOG

b) Description du logiciel

Basé sur le développement de python3, utilisant PYINSTALLER pour emballer le fichier exe généré. Permet d'interroger rapidement les images de défauts des noyaux et prend également en charge l'exportation par lots des images de défauts.



Figure 8 Diagramme logiciel

c) Les fonctions

Fonction 1 : Effacer l'écran

Description : Effacez le contenu de la boîte de sortie en bas à gauche.

Mise en œuvre du code :

Créer des fonctions pour effacer les zones de texte.

```
def clear_text():
    text.delete('1.0', END)
```

Définir la fonction "clear_text" et définir la commande du bouton "Effacer l'écran" comme "clear_text".

```
b = Button(root, text='清屏', font=('微软雅黑', 14), width=10, height=1, command=clear_text)
b.place(x=800, y=250)
```

Fonction 2 : Recherche de l'existence d'un numéro de batterie

Description : Après avoir saisi le numéro de batterie ou le fichier txt avec les numéros de batterie, sélectionnez le fichier "GLOG" pour savoir si le numéro de batterie correspondant existe dans le fichier "GLOG" et exportez les résultats dans le fichier " Results.txt" généré.

Mise en œuvre du code :

Créer la méthode "check_existence".

```
def check_existence():
```

Déterminez d'abord si le fichier "GLOG" et le numéro de batterie ont été saisis. Si vous transmettez un fichier txt contenant le numéro de batterie, traitez-le et passez-le dans le tableau "barcodes".

```
log = get_text_input(t2)
barcodes = get_text_input(t3)
if barcodes == '':
    text.insert(END, "请输入电芯编号或者含电芯编号的txt文件!\n")
elif log == '':
    text.insert(END, "请选择正确的glog文件!\n")
else:
    if not os.path.isfile(log):
        text.insert(END, f'该 {log} 不存在!! \n')
        raise IOError(f'该 {log} 不存在!!')
    glog = open(log).readlines()
    if os.path.isfile(barcodes):
        barcodes = [
            line.strip()
            for line in open(barcodes, encoding='gbk').readlines()
        ]
        barcodes = [line[:24] for line in barcodes if line.startswith('001CB')]
    else:
        barcodes = [barcodes]
    print(barcodes)
```

Confirmez que l'entrée n'est pas vide

Confirmez que le fichier GLOG existe

Lancez la recherche transversale et enregistrez les résultats

```
for i, barcode in enumerate(barcodes):
    text.insert(END, f'[{i + 1}/{len(barcodes)}] {barcode}\n')
    file_object.write(f'[{i + 1}/{len(barcodes)}] {barcode}\n')
    first_line = 0
    for j in range(len(glog)):
        if barcode in glog[j]:
            first_line = j
            text.insert(END, f'{barcode} 在 {log} 的第 {j} 行!\n')
            file_object.write(f'{barcode} 在 {log} 的第 {j} 行!\n')
            break
    if first_line == 0:
        text.insert(END, f'警告! {barcode} 在 {log} 中未找到!\n')
        file_object.write(f'警告! {barcode} 在 {log} 中未找到!\n')
text.insert(END, f'对于{log}的查询结果已输出至查询结果.txt中!\n')
```

Boucle à travers chaque élément

Confirmer l'existence de chaque ligne

Finalement, on vous dira si chaque numéro peut être trouvé dans le fichier "GLOG". Comme suit, je peux trouver le numéro de batterie 001CB6H0000001B9A7100003 dans le fichier "GLOG" : glog.21210908-200154.37128 à la ligne 315595.



查询结果.txt - 记事本

文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

[1/1] 001CB6H0000001B9A7100003

001CB6H0000001B9A7100003 在 glog.20210908-200154.37128 的第 315595 行!

Fonction 3 : Générer des commandes python

Description : Génère directement la commande d'entrée qui peut être utilisée à la ligne de commande python, taper la commande dans python aura l'effet équivalent.

Mise en œuvre du code :

Obtenez les informations de position des trois boîtes d'entrée pour sortir des commandes python dans une certaine combinaison. Permet de suivre l'entrée de cette commande même en utilisant le terminal.

```
def get_command():
    global command
    str1 = get_text_input(t1)
    str2 = get_text_input(t2)
    str3 = get_text_input(t3)
    print(str3)
    str_all = "python kantu.py " + str1 + " " + str2 + " " + str3
    text.insert(END, "当前命令为:" + str_all + '\n')
    command = str_all
    print(command)
```

Combiné en format ligne de commande

Fonction 4 : Exécution

Description : Pour le nom du dossier d'images saisi, le fichier "GLOG" sélectionné et le numéro de la batterie à rechercher, la carte de recherche de défauts correspondante est générée. Comme la fonction 2, le texte avec plusieurs numéros de batterie est également pris en charge ici.

Mise en œuvre du code :

Enregistrez les informations numérotées des éléments de la batterie dans la liste. Certaines exceptions sont faites, notamment le fait que le fichier n'existe pas, que le numéro ne comporte pas 24 chiffres, etc.

```
if os.path.isfile(barcode):
    save_dir_flag = True
    barcodes = [
        line.strip()
        for line in open(barcode, encoding='gbk').readlines()
    ]
elif "." in barcodes:
    text.insert(END, f'未找到文件 :{barcodes} 。 \n')
else:
    if (not barcodes.startswith('001CB')) and len(barcodes) >= 6:
        new_barcode = []
        save_dir_flag = True
        for key in name_to_path_map.keys():
            if barcodes in key:
                new_barcode.append(key)
        if len(new_barcode) == 0:
            text.insert(END, f'未找到电芯编号 :{barcodes} 。 \n')
            raise Exception(f'未找到电芯编号 : {barcodes}。')
        barcodes = new_barcode
    elif len(barcodes) == 24:
        save_dir_flag = True
        barcodes = [barcodes]
    else:
        text.insert(END, '请输入正确编号:至少后六位或全部!\n')
```


Itérer sur chaque valeur dans les "barcodes", appeler la fonction2 et voir si elle peut être trouvée dans le "GLOG".

```
for i, barcode in enumerate(barcode):
    print(f'[{i + 1}/{len(barcode)}] {barcode}')
    text.insert(END, f'[{i + 1}/{len(barcode)}] {barcode}\n')
    first_line = 0
    for j in range(len(glog)):
        if barcode in glog[j]:
            first_line = j
            break
    if first_line == 0:
        print(f'警告! {barcode} 在 {log} 中未找到!')
        text.insert(END, f'警告! {barcode} 在 {log} 中未找到!\n')
```

Sur la base du numéro de cellule, trouvez la ligne correspondante contenant "num results" et utilisez la fonction "extract_defect_info" pour décomposer les informations de la ligne.

```
if lines[k].find('num results') != -1:
    num_results = int(lines[k].split(':')[1])
    for m in range(k + 1, k + 1 + num_results): Décomposez les informations de chaque ligne
        defect_info = extract_defect_info(lines[m])
        defects.append(defect_info)
```

Exemple : Les informations de la ligne trouvée sont affichées comme suit :

```
481728 18910 04:35:39.777163 23052 ind_api_impl.cpp:1416] bolangbian_length_harsh: 0
481729 18910 04:35:39.777163 23052 ind_api_impl.cpp:1417] num_results: 1
481730 18910 04:35:39.777163 23052 ind_api_impl.cpp:1419] label: 0 conf: 0.45752 roi: 363 568 525 670 area: 3.2208 mid_to_low: -0.0342034 mid_to_high: 0.628347 width: 0.0603325 height: 0
481731 18910 04:35:42.339802 62276 ind_api_impl.cpp:1484] Load image from file: C:\software\0824-1\build\TempImage\001C86H000000189A7183787\001C86H000000189A7183787_CAV1_9096_B2_EDGE2.bmp
481732 18910 04:35:42.339802 62276 ind_api_impl.cpp:1489] Load image from file: C:\software\0824-1\build\TempImage\001C86H000000189A7183787\001C86H000000189A7183787_CAV1_9096_B2_EDGE2.bmp done.
481733 18910 04:35:42.339802 62276 ind_api_impl.cpp:1484] Load image from file: C:\software\0824-1\build\TempImage\001C86H000000189A7183787\001C86H000000189A7183787_CAV1_9096_B2_EDGE2.tif
481734 18910 04:35:42.461547 62276 ind_api_impl.cpp:1489] Load image from file: C:\software\0824-1\build\TempImage\001C86H000000189A7183787\001C86H000000189A7183787_CAV1_9096_B2_EDGE2.tif done.
481735 18910 04:35:42.557762 62276 ind_api_impl.cpp:1411] filename: 0.887735
481736 18910 04:35:42.557762 62276 ind_api_impl.cpp:1412] fanliam: 0.887735
481737 18910 04:35:42.557762 62276 ind_api_impl.cpp:1415] piangwang_length: 0
481738 18910 04:35:42.557762 62276 ind_api_impl.cpp:1414] piangwang_length_harsh: 0
481739 18910 04:35:42.557762 62276 ind_api_impl.cpp:1415] bolangbian_length: 0
481740 18910 04:35:42.557762 62276 ind_api_impl.cpp:1416] bolangbian_length_harsh: 0
481741 18910 04:35:42.557762 62276 ind_api_impl.cpp:1417] num_results: 0
481742 18910 04:35:42.557762 9840 ind_api_impl.cpp:1484] Load image from file: C:\software\0824-1\build\TempImage\001C86H000000189A7183787\001C86H000000189A7183787_CAV1_9096_B2_EDGE1.bmp
481743 18910 04:35:42.557762 9840 ind_api_impl.cpp:1489] Load image from file: C:\software\0824-1\build\TempImage\001C86H000000189A7183787\001C86H000000189A7183787_CAV1_9096_B2_EDGE1.bmp done.
481744 18910 04:35:42.557762 9840 ind_api_impl.cpp:1484] Load image from file: C:\software\0824-1\build\TempImage\001C86H000000189A7183787\001C86H000000189A7183787_CAV1_9096_B2_EDGE1.tif
481745 18910 04:35:42.620273 9840 ind_api_impl.cpp:1489] Load image from file: C:\software\0824-1\build\TempImage\001C86H000000189A7183787\001C86H000000189A7183787_CAV1_9096_B2_EDGE1.tif done.
481746 18910 04:35:42.792189 9840 ind_api_impl.cpp:1411] filename: 0.8889995
481747 18910 04:35:42.792189 9840 ind_api_impl.cpp:1412] fanliam: 0.8889995
```

Séparer les informations sur les tâches dans le "boxex" et "labels".

```
bboxes = np.array(
    [d['roi'] + [d['conf']] for d in defects])
labels = np.array([d['label'] for d in defects])
```

Il est passé dans cette fonction "imshow_det_bboxes" et les informations sont données sur l'image par la méthode opencv2 de cette fonction. (cv2.rectangle and cv2_add_chinese_text)

```
imshow_det_bboxes(
    tif_img_tmp,
    bbox2,
    np.array([label]),
    bbox_color=(0, 0, 255),
    text_color=(255, 0, 0),
    thickness=2,
    font_scale=0.7,
    class_names=CLASSES,
    out_file=os.path.join(save_dir, save_name),
    show=False,
    zhixin_flag=True)

cv2.rectangle(
    img, left_top, right_bottom, bbox_color, thickness=thickness)

img = cv2_add_chinese_text(img, label_text,
    (bbox_int[0] + new_w, bbox_int[1] - 25 + new_h),
    cv2.FONT_HERSHEY_COMPLEX, text_color)
```


Nous nous retrouvons avec une image qui contient des informations clés. Cela comprend la surface du défaut (area), le point le plus bas du défaut (mid_to_low), le point le plus haut du défaut (mid_to_high), la largeur du défaut (width) et la hauteur du défaut (height).

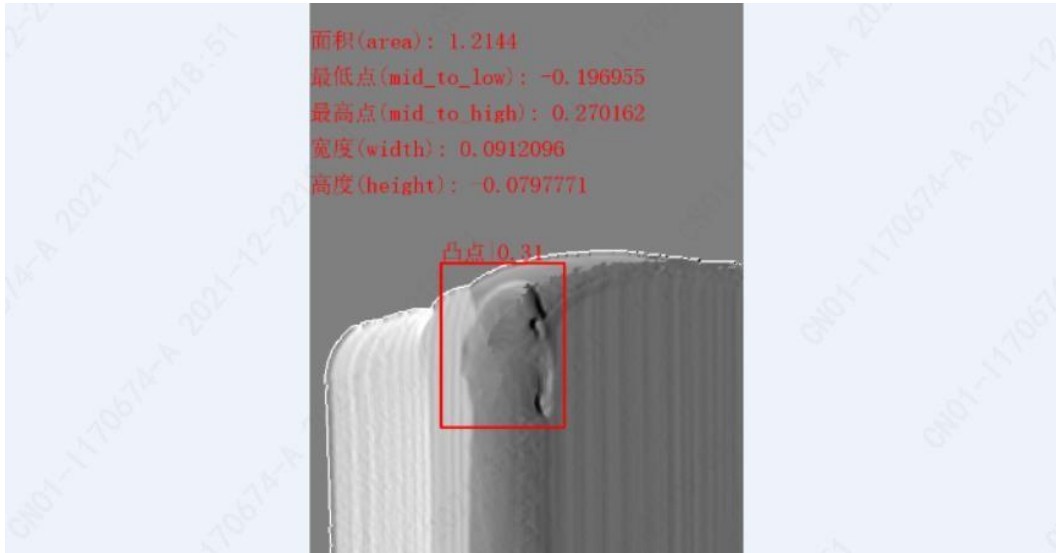


Figure 9 Résultats générés par le logiciel

d) Utilisez-le sur la ligne de production

Pour appliquer cela à la ligne de production, je dois montrer les lacunes de l'inspection visuelle manuelle actuelle des défauts sur la ligne de production, puis démontrer par l'expérimentation pourquoi nous avons besoin d'un étiquetage assisté par des mesures d'IA pour nous aider à déterminer les défauts, et enfin la nécessité de l'appliquer à la ligne de production.

Contexte :

La méthode artificielle actuelle pour déterminer l'OK, NG d'un produit sur la ligne de production est irrationnelle. (Les jugements sont faits en utilisant purement la subjectivité humaine)

Raisons :

1) Lorsque le même noyau est inspecté par différentes personnes, celles-ci peuvent émettre des jugements différents quant à la présence d'un défaut dans le noyau. Il n'est pas possible d'être totalement précis selon les normes de qualité requises par l'AQ pour la soudure du couvercle supérieur d'une carotte.

2) la même personne peut également se tromper sur l'état du même noyau en raison de la lumière, d'une vision réduite, etc.

Comme le plus petit objet visible à l'œil est de 0,1 mm à 0,2 mm, l'œil humain seul ne peut plus juger des défauts lorsque la profondeur d'un défaut central tel qu'un trou d'épingle est inférieure à cette valeur fluctuante. L'appréciation humaine des défauts fondamentaux est trop subjective et volatile, nous avons donc besoin d'un outil quantitatif pour nous aider dans ce processus. Ace stade, il serait utile que la mesure de chaque dimension du défaut soit donnée au moment du jugement.

Expérimentation et analyse :

Préparation préliminaire : Puisque nous ne considérons les résultats des mesures AI que comme une aide au jugement, les anomalies telles que le bruit de l'image et le décollement des scories de soudure doivent être éliminées. Ainsi, sur un total de 304 échantillons, nous avons sélectionné 272 échantillons.

J'ai utilisé trois méthodes pour réaliser l'expérience.

1. l'observation du diagramme 2D ;
2. l'observation du diagramme 3D ;
3. l'utilisation de mon logiciel plus l'observation du diagramme 3D.

Les résultats des expériences sont les suivants :

Méthodes	Total	Taux d'overkill	Taux d'misskill
2D	272	18.0%	10.2%
3D	272	15.8%	8.1%
3D avec logiciel	272	6.3% (17)	1.8%

Nous pouvons constater que nos taux d'overkill et d'misskill ont considérablement diminué après l'utilisation du logiciel.

Mais même avec ces améliorations, pourquoi avons-nous toujours un taux aussi élevé d'overkill et de misskill ?

Revenons au concept d'overkill et de misskill :

Jugement de QA	Mon jugement	Résultats
OK	NG	OVERKILL
NG	OK	MISSKILL

Puisque le jugement final est pertinent pour les deux parties (moi et QA : Gestion de la qualité), j'ai décidé de revenir à l'ontologie des données pour prendre la justesse des observations.

J'ai compté toutes les données de l'overkill et j'ai constaté que 15 des 17 échantillons étaient des trous d'épingles.

J'ai compté toutes les mesures des trous d'épingles et j'ai constaté qu'elles dépassaient toutes le seuil donné (0.32). Cela signifie qu'ils doivent être définis comme des échantillons NG !

图片编号	人工判断结果bmp	人工判断结果TII	人工判断结果 (AI辅助测量)	真实!
001CB6H0000001B9A7118598	NG	NG	NG(针孔深度0.324309)	OK
001CB6H0000001B9B7100630	NG	NG	NG(针孔深度0.32427)	OK
001CB6H0000001B9B7104346	NG	OK	NG(针孔深度0.414925)	OK
001CB6H0000001B9B7105341	NG	OK	NG(针孔深度0.3411)	OK
001CB6H0000001B9B7105820	NG	OK	NG(针孔深度0.353125)	OK
001CB6H0000001B9B7106830	NG	NG	NG(针孔深度0.321827)	OK
001CB6H0000001B9B7107318	NG	NG	NG(针孔深度0.334921)	OK
001CB6H0000001B9B7107458	NG	NG	NG(针孔深度0.383703)	OK
001CB6H0000001B9B7107590	OK	NG	NG(针孔深度0.330906)	OK
001CB6H0000001B9B7108808	NG	NG	NG(针孔深度0.381704)	OK
001CB6H0000001B9B7109524	NG	NG	NG(针孔深度0.351238)	OK
001CB6H0000001B9B7110352	NG	NG	NG(针孔深度0.339817)	OK
001CB6H0000001B9B7110548	NG	OK	NG(针孔深度0.35601)	OK
001CB6H0000001B9B7112920	NG	OK	NG(针孔深度0.339244)	OK
001CB6H0000001B9B7113774	NG	NG	NG(针孔深度0.351238)	OK

Valeurs mesurées > 0.32

Figure 10 Résultats expérimentaux

Ensuite, il nous suffit de prouver que les mesures sont absolument exactes et que le jugement du QA est erroné. Nous avons utilisé une image de caméra simulée et effectué une mesure manuelle de la ligne de traction (comme figure ci-dessous), après 10 mesures, nos résultats d'erreur sont fondamentalement conformes aux mesures données par la machine, avec un écart de 0,01 mm

1	2	3	4	5	6	7	8	9	10	Min	Max	Avg
-0.113	-0.107	-0.110	-0.103	-0.105	-0.109	-0.108	-0.106	-0.106	-0.110	-0.103	-0.113	-0.108

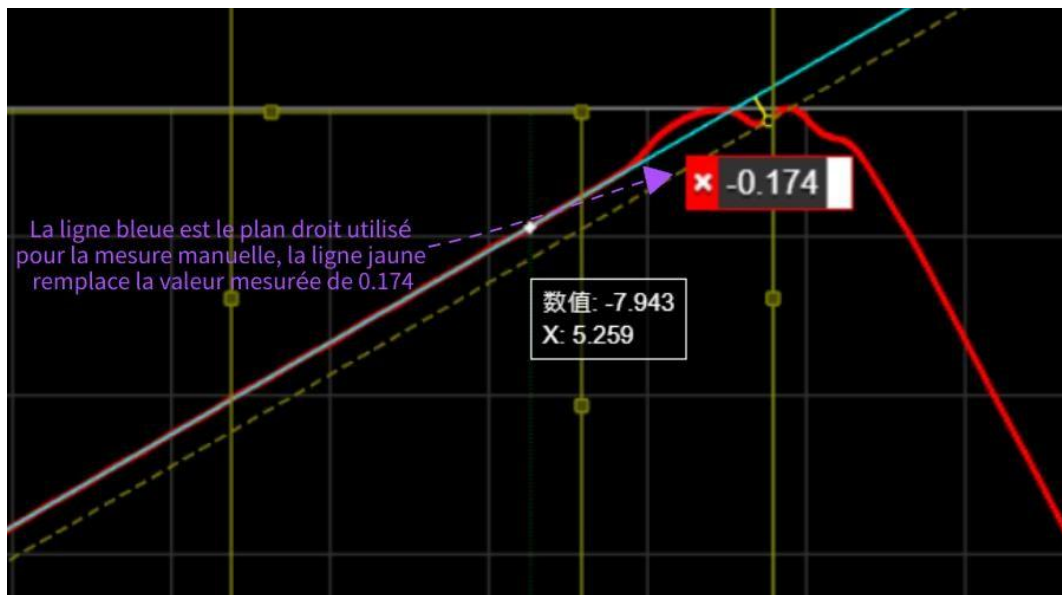


Figure 11 Schéma d'essai

e) Conclusion et résultats

Avec cette conclusion, nous ne pouvons pas prouver que le jugement de QA est absolument correct. En outre, la mesure assistée par l'IA est une aide précieuse pour l'inspection visuelle manuelle en termes de résultats. Mais comment obtenir un résultat vraiment précis ? L'échantillon de carotte NG doit être remis au QA avec la carte des défauts qui en découle : laissez le QA confirmer que le type de défaut saisi dans la batterie est exact et vérifiez la précision de ses mesures. Si l'on pense qu'il peut y avoir un problème avec l'angle de la caméra, etc., il peut être remis dans la chaîne de production pour être testé à nouveau. Cela permet de garantir l'authenticité des valeurs mesurées, d'éviter les problèmes de capture de la caméra et de réduire l'influence des facteurs subjectifs du jugement humain, ce qui permet d'obtenir un véritable échantillon NG et un échantillon OK.

Finalement, nous l'avons appliqué à la chaîne de production, ce qui a permis de réduire le taux d'overkill à 0,3 % et le taux d'misskill à 0,0 %.

C. Amélioration et test de la plateforme AE

a) Informations générales

AUTO EVOLUTION est un système de production de modèles IA pour la détection des défauts industriels.[10] Comme l'inspection de la qualité des batteries d'énergie nouvelle se caractérise par des exigences de précision élevées, des types de batteries variables et de nouveaux défauts qui apparaissent souvent avec des changements dans le processus de la chaîne de production, nous espérons que cette plateforme donnera aux opérateurs non professionnels la possibilité de modéliser des modèles d'IA. Il améliore considérablement la précision de l'inspection de la qualité des batteries d'alimentation et réduit le coût des itérations de longue durée des modèles d'IA.

Ses principaux objectifs sont les suivants :

1. Spécification des données du projet et des itérations du modèle pour faciliter la communication et l'interface du projet.
2. Affiner et optimiser les principaux processus du projet en fonctionnant sans code.
3. Réduire les opérations manuelles, améliorer la précision de la détection et renforcer l'efficacité.
4. l'optimisation du taux d'utilisation des ressources informatiques.

À ce stade, le système AE se concentre sur l'archivage, l'itération et la documentation des ensembles de données, la formation des modèles, les tests, etc.

b) Mes réalisations

Depuis la version 10.15, l'équipe du GIE et moi-même avons des discussions chaque lundi sur les améliorations à apporter à la plate-forme. J'ai donné à l'équipe du GIE 40 suggestions d'améliorations, dont 39 ont été acceptées et 10 bugs identifiés, dont les détails sont confidentiels pour d'autres entreprises et ne peuvent donc pas être divulgués.

Feedback sur l'expérience AE

AE 使用体验反馈

我的空间 | 最近修改: 12月21日 21:28

任务列表

优先级 价值

所有任务 字段配置 2 筛选 1 分组 排序 行为 提醒 生成表单 评论 添加记录

改进项目描述 Description du projet d'amélioration

优先级 价值 AE 详细描述 Description détaillée 反馈日期 状态 图片

改进项目描述	优先级	价值	AE 详细描述	反馈日期	状态	图片
1. 模型测试结果可视化修正	P0	4-期望型需求		11-02	排期中	
2. 模型类型是否支持多类, 否则测试中会有不同的模板	P0	4-期望型需求	输入数据在基本上仅使用一模板	11-08	已入需求池	
3. 测试(excel)中加入gt的标签信息	P0	4-期望型需求	在Total表1中写入标签类型	11-08	已入需求池	
4. 2. 筛选图片操作	P0	5-必备型需求	比如: 从所有数据中里面筛选出所有11孔的数据的反馈	11-15	已入需求池	
5. 数据问题	P0	5-必备型需求	本地Windows未发现问题, 产线有问题	11-22	已入需求池	
6. 结果页面增加type属性	P0	4-期望型需求		11-22	已入需求池	
7. 下载的压缩包, 增加说明文档, 提供该压缩包的一些信息	P0	4-期望型需求	压缩包名称改为 det-\$TEST_TASK_NAME.zip	11-22	已入需求池	
8. CV处理需求增加	P0	5-必备型需求	cv后需求增加, 应该是测试应该系数有问题, 另外要在测量值...	11-22	已入需求池	
1. 在右侧信息栏输出模型对模型的测量值	P1	5-必备型需求	能在图片右侧输出模型的测量值结果	10-15	排期中	
2. 模型对比功能	P1	5-必备型需求	1.对比两个模型的测试结果	11-01	排期中	
3. 作为初次筛选, 2个模型增加操作	P1		模型对比的功能	11-01	未启动	
4. 自动筛选模板, 灯板设置	P1			11-08	已入需求池	

51 条记录

Figure 12 Commentaires sur AE

II-2. Simulation de systèmes AGV

II-2.1. Informations générales

A. Concepts et scénarios commerciaux

L'outil logiciel Plant Simulation de Tecnomatix, un logiciel de simulation sur la production, la logistique et l'ingénierie implémenté en C++, est un outil de modélisation et de simulation orienté objet, graphique et intégré. La structure et la mise en œuvre du système répondent toutes deux aux exigences de l'orientation objet. La simulation d'usine, classée par l'académie, est une classe typique d'outils logiciels de simulation d'événements discrets. [11]



Figure 13 Caractéristiques de Plant Simulation

B. Terminologie

AGV (Automated Guided Vehicle) : Véhicules de transport équipés de dispositifs de navigation automatique, capables de suivre un chemin de navigation défini, avec protection de sécurité et diverses fonctions de transfert de charge. [12]



Figure 14 Modèle de AGV

Entrée des matériaux : Envoyez une demande pour que l'AGV livre le matériel requis ici.

Sortie de matériau : Envoyez une demande pour que l'AGV ramasse le matériel généré.

AGVbroker : Recevoir les demandes générées et envoyer des instructions à l'AGV pour le faire fonctionner.

Simulation : Utiliser des modèles pour reproduire les processus essentiels qui se produisent dans les systèmes réels.

Jumeau numérique : Le jumeau numérique est un processus de simulation qui intègre de multiples disciplines, quantités physiques, échelles et probabilités en exploitant pleinement les données provenant de modèles physiques, de mises à jour de capteurs et d'historiques opérationnels. [13]

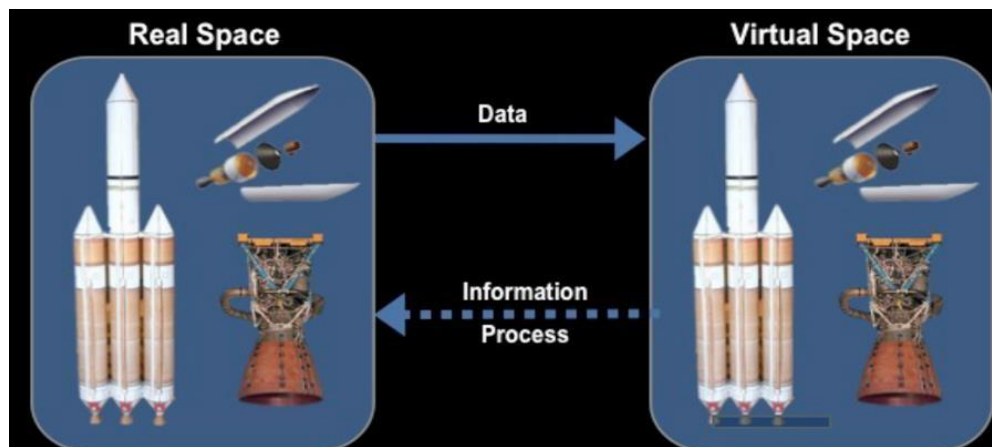


Figure 15 Le concept de jumeau numérique de Grieves et Vickers

C. Langage : Simtalk 2.0

Le langage de programmation utilisé dans Plant Simulation est appelé SimTalk et est basé sur le langage de programmation C.

En écrivant des programmes, l'utilisateur peut :

1. Répondre à des événements spécifiques et effectuer des contrôles personnalisés pour une modélisation plus précise.
2. Accéder aux propriétés des objets pour une vue flexible de l'état d'exécution de la simulation.
3. Définir des propriétés définies par l'utilisateur qui peuvent être manipulées pour fournir une représentation plus détaillée de l'état de l'objet.
4. La définition de boîtes de dialogue pour faciliter l'utilisation du modèle de simulation par l'utilisateur.

D. Objectifs

Le modèle de simulation de l'AGV est établi par le biais d'une simulation d'usine, et la répartition des tâches, le mode de chargement et l'itinéraire d'exploitation de l'AGV sont optimisés dans une certaine mesure.

II-2.2. Mes réalisations

A. Simulation du système AGV

Deux types de simulation d'AGV pour des usines à grande échelle ont été réalisés indépendamment, ce qui constitue une bonne référence pour l'ordonnancement des AGV sur les lignes de production.

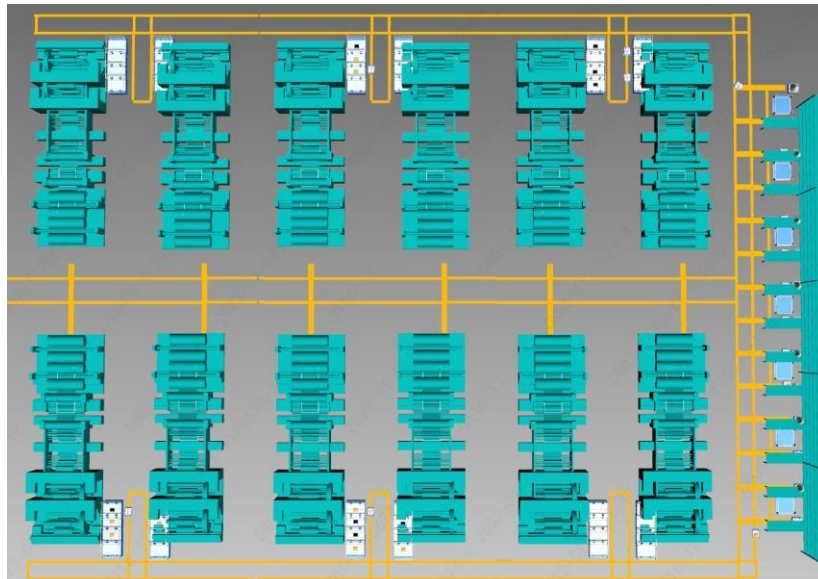


Figure 17 Conception de l'AGV de la section d'enroulement

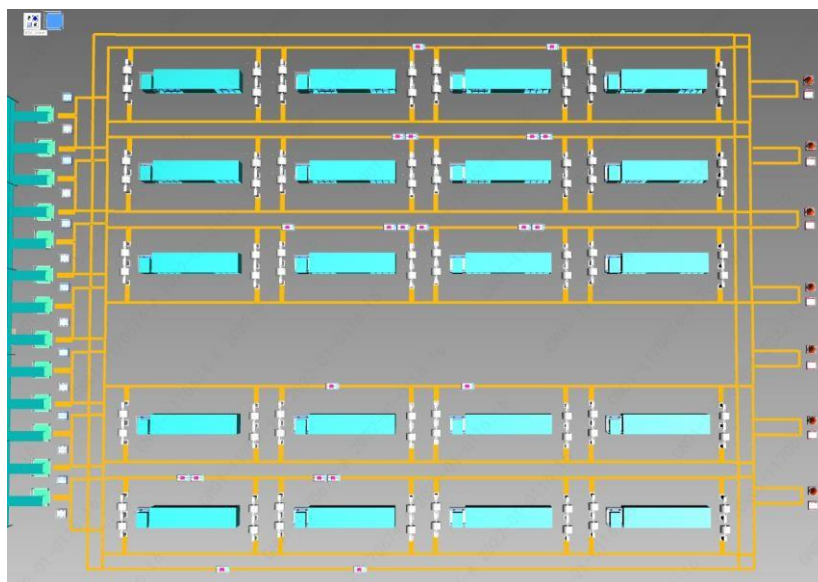


Figure 16 Conception d'un AGV pour la section de découpage à l'emporte-pièce

B. Logique du système AGV

Avant d'expliquer comment construire le code, nous devons comprendre la logique d'ordonnancement de l'ensemble du système. Comme indiqué ci-dessous :

Description du processus

1. L'entrée du matériel donne des instructions pour demander du matériel.
2. L'AGVbroker reçoit cette commande et l'envoie au AGV.
3. Les AGV transportent les matériaux vers la machine.
4. Production de machines qui génèrent de nouveaux matériaux à l'exportation.
5. Envoyez une nouvelle demande de chariot pour le ramassage du matériel.
6. L'AGVbroker envoie la commande au AGV.
7. le chariot prend le matériel et passe au processus suivant.

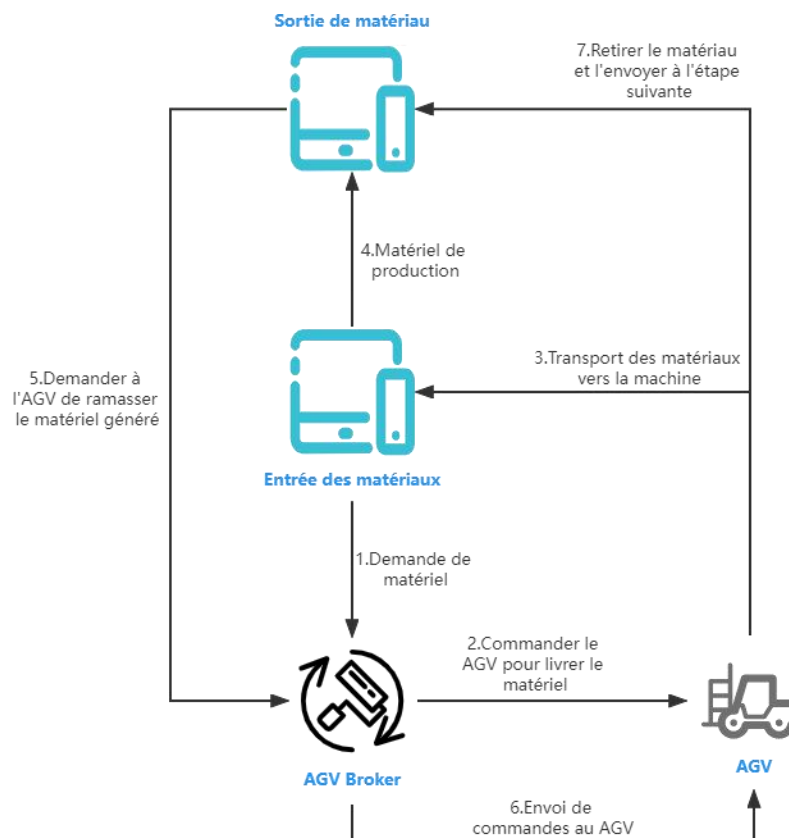


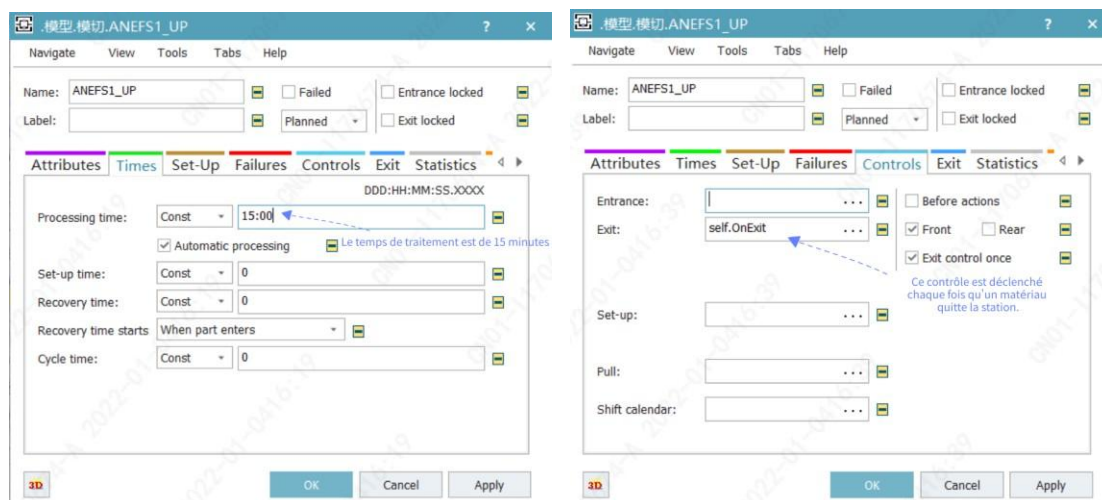
Figure 18 Diagramme de flux logique

C. Mise en œuvre programmée

Je vais suivre chaque étape du processus en me basant sur l'organigramme ci-dessus pour montrer comment le système peut être mis en œuvre de manière programmatique.

a) Demande de matériel

Pour chaque station, lorsque son matériel est consommé, elle envoie une demande de matériel au système. Dans le système de simulation, chaque station dispose de son propre temps de traitement (comme le montre la figure xx). Lorsque son matériel est épuisé, il quitte la station pour l'étape suivante, et la station doit alors faire une demande de nouveau matériel. Nous utilisons le contrôle des exportations pour simuler une demande. (Cette commande de sortie est déclenchée chaque fois qu'une ressource sort de la station).



Le code pour "self.OnExit" est le suivant. Où "self" désigne la méthode elle-même, "self. ~" désigne la couche au-dessus de self, pour la station de travail ANEFS1_UP, "self.~.broker" désigne le coordinateur de la station de travail, et "self.~.broker.m_request" désigne la méthode du coordinateur. Les 5 derniers paramètres lui sont passés.

```
self.~.broker.m_request(void,self.~, "上料",1,self.~.行号)
```

b) Commander le AGV pour livrer le matériel

Les fonctions suivantes sont exécutées lorsque "m_request" est déclenché, où "NumGoods" indique le nombre total de tâches et "NumOrder" indique le nombre de tâches effectuées pour le compte de la tâche. Les informations concernant la tâche seront écrites dans la table "t_request".

```

-- Param 表示输入参数 mu: 物料 loc: 请求的位置 task需要的任务:
Param mu,Loc:Object,Task:string,priority,pos_line:integer

var obj := self.~
var t_request := root.t_request
var timenow := root.eventController.simTime
obj.NumGoods += 1
obj.NumOrder += 1

t_request.AppendRow(obj.NumGoods,mu,Loc,Task,priority,pos_line,timenow)

obj.m_Doorder()
  
```

Ensuite, comme dans la dernière ligne du schéma ci-dessus, nous allons déclencher la fonction "m_Doorder" et d'abord nous allons trier la table "t_request" par temps et par priorité pour obtenir l'ordre d'exécution des tâches. Ensuite, nous utiliserons "m_getAGV" pour sélectionner le chariot qui exécutera la tâche.

```

.模型.模拟.AGV_broker.m_Doorder
var obj := self.~
var veh :object
var t_request := root.t_request
var t_order := root.t_order
var timenow := root.eventController.simTime
var t_AGV := root.t_AGV
var t_all := root.t_all

var num := 1
t_request.sort("task生成时间","UP")
t_request.sort("优先级","Down")
veh := obj.m_getAGV()

.模型.模拟.AGV_broker.m_getAGV
->object
var obj:= self.~
var t_AGV := root.t_AGV

for var i := 1 to t_AGV.ydim
  if t_AGV["Status",i] = true
    result := t_AGV["AGV",i]
    t_AGV["Status",i] :=false
  exitloop
end
next
  
```

Nous avons acquis le AGV qui va effectuer la tâche, et nous devons maintenant informer le AGV de la tâche. Dans le cas de l'Entrée des matériaux, nous devons aller chercher le matériel nécessaire à la station avant de le livrer à la station, donc la première destination du AGV est le lieu où le matériel est pris et la deuxième destination est la station.


```

if task = "上料" or task = "SP_空盘"
  for var m := 1 to t_all.ydim
    if line = t_all["行号",m]
      veh.Destination := t_all[task,m]
      exitloop
    end
  next
  veh.next_destination := t_request["请求位置",num]
elseif task = "下料" or task = "UP_空盘"
  for var m := 1 to t_all.ydim
    if line = t_all["行号",m]
      veh.next_destination := t_all[task,m]
      exitloop
    end
  next
  veh.Destination := t_request["请求位置",num]
end

```

Le contenu de cette section exécuté

Les informations relatives à la tâche sont ensuite affectées au véhicule et retirées de la table "t_request" pour être stockées dans la table "t_order".

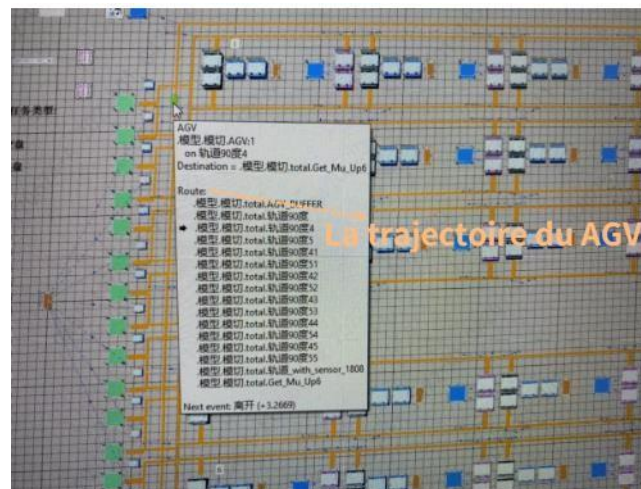
```

veh.TaskNu := t_request["NU",num]
veh.Task := t_request["任务类型",num]
veh.Task_number := t_request["任务编号",num]
t_order.AppendRow(t_request["任务编号",num],t_request["NU",num],t_request["请求位置",num],t_request["任务类型",num],t_request["优先级",num],t_request["关联行号",num],t_request["
--root.Transferline(lnum,veh,timeword)
obj.NumOrder --1
obj.NumFreeAGV --1
t_request.cutRow(num)
veh.go

```

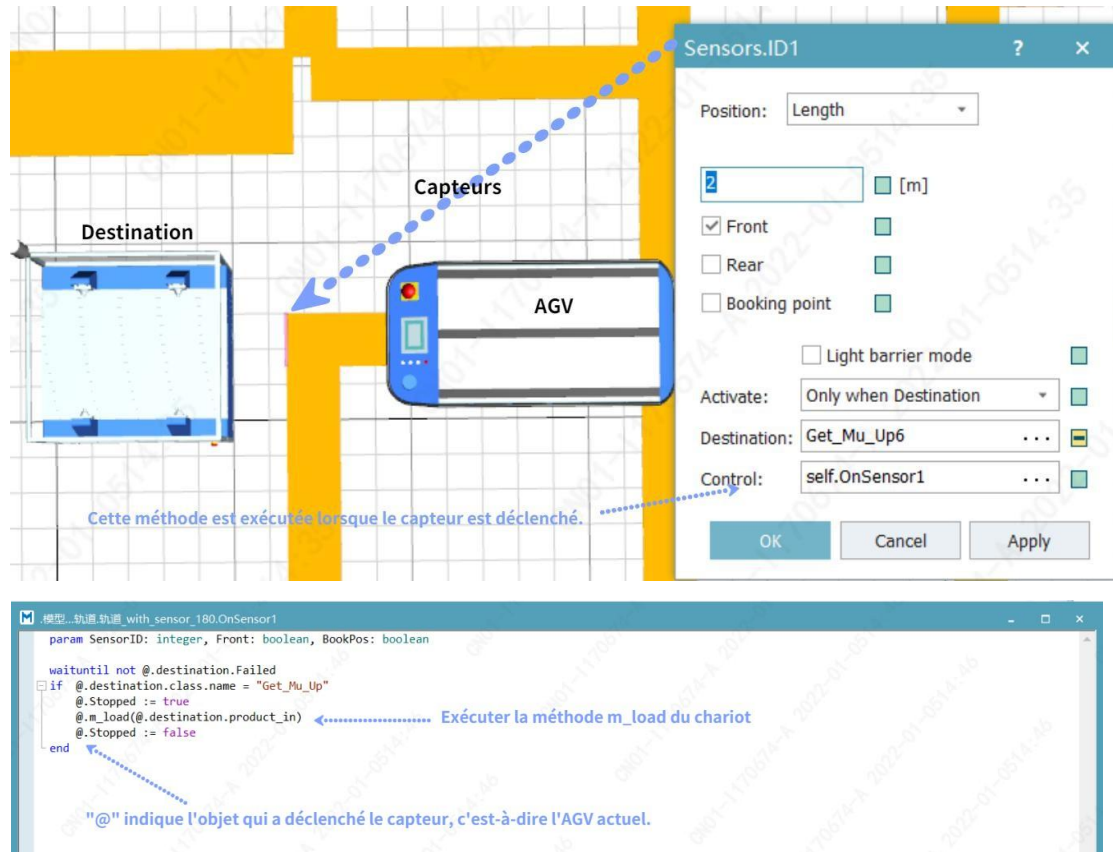
Sortez l'AGV de la cache et bougez.

Grace a le système de trajectoire fourni avec PLANT SIMULATION permet à l'AGV de sélectionner automatiquement une trajectoire en fonction de la route définie.



c) Transport des matériaux vers la machine

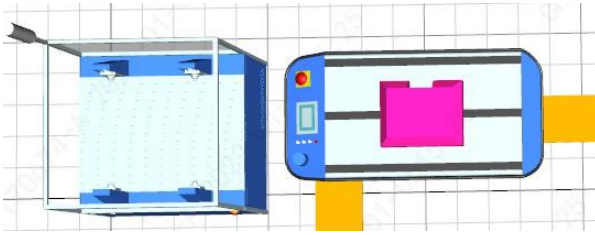
Nous savons déjà que l'AGV choisira automatiquement son chemin, donc lorsque la voiture atteint sa destination, nous simulons un capteur sur la route devant cette destination pour déclencher l'action de l'AGV.



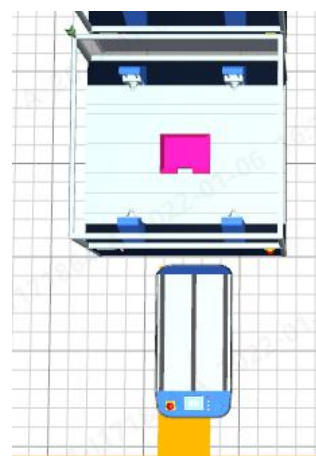
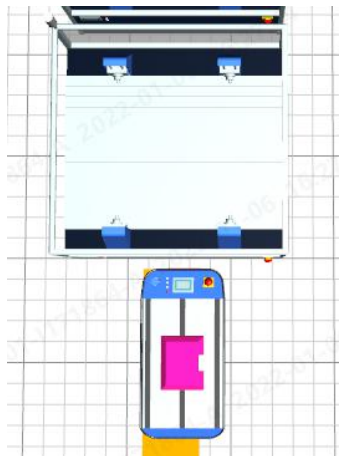
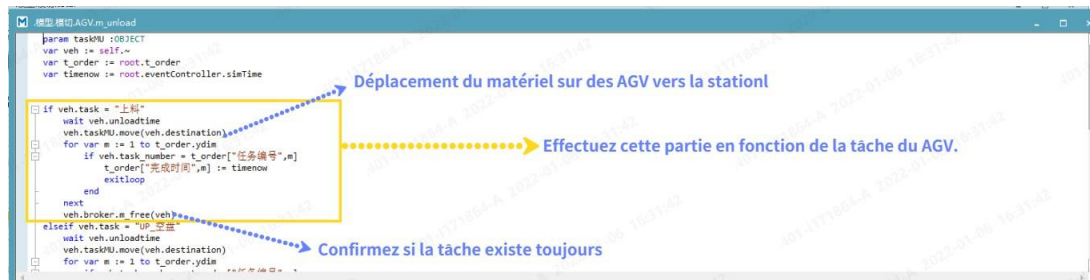
L'AGV arrive d'abord à l'endroit où le matériel a été obtenu et se procure le matériel requis par le poste de travail. À ce stade, le chariot exécute la méthode "m_load", en créant un matériau sur l'AGV et en modifiant la destination de l'AGV pour qu'elle corresponde à la station qui a besoin du matériau.



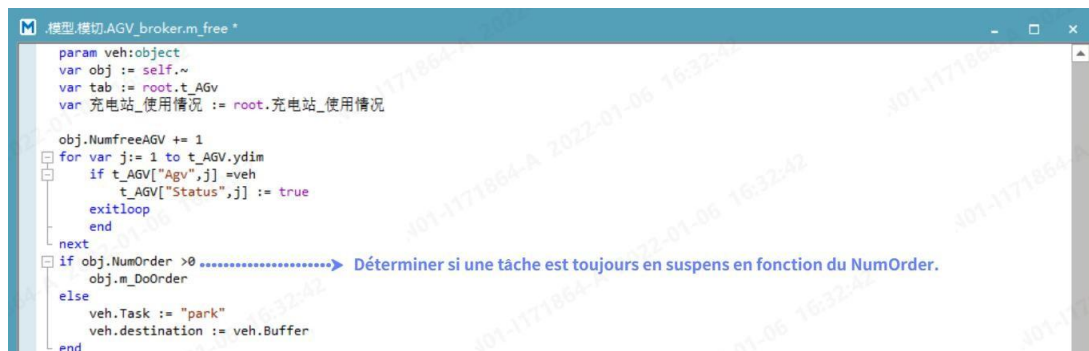
Nous pouvons voir que l'AGV est déjà chargé de matériel.



L'AGV se rend ensuite à la station où le matériel doit être chargé. Lorsque l'AGV se rend à la station, nous utilisons également des capteurs pour déclencher la méthode de "unload" de l'AGV afin de déplacer le matériel sur l'AGV vers la station.

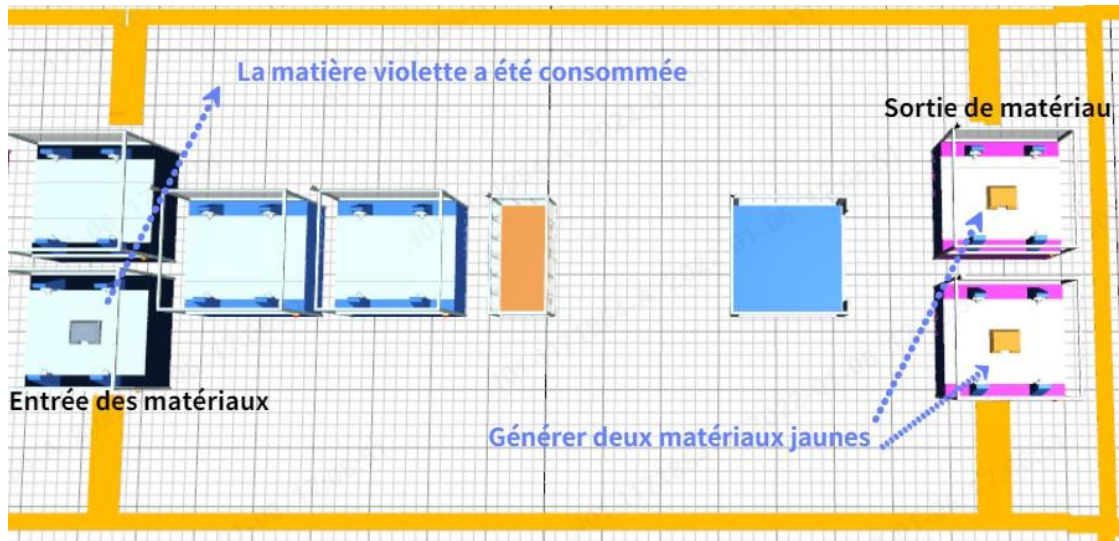


L'AGV a terminé sa tâche et exécute alors la méthode "m_free" pour demander au "AGVBroker" s'il a d'autres tâches à accomplir. Si c'est le cas, l'AGV récupère les informations relatives à la tâche pour l'exécuter, sinon, l'AGV retourne au Buffer et attend la prochaine commande à envoyer.



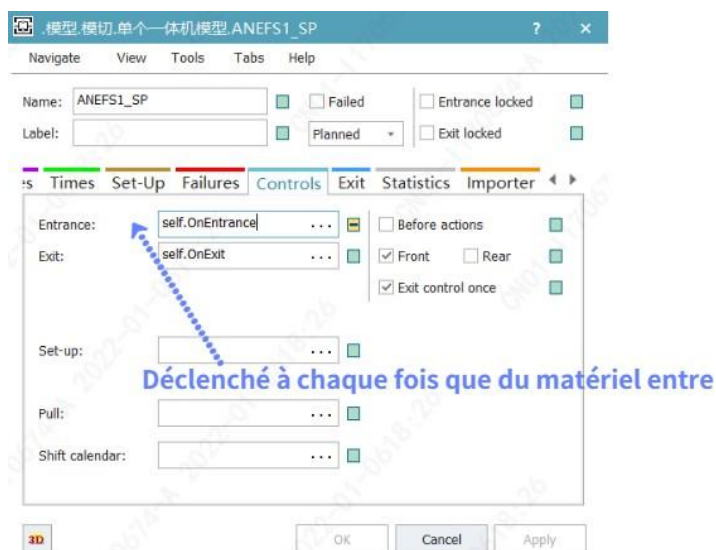
d) Production de machines

Après qu'une période de temps prédéterminée se soit écoulée dans la station (en fonction du temps de fonctionnement réel de l'usine), deux nouveaux produits sont fabriqués à la sortie de matériau.



e) Envoyez une nouvelle demande à l'AGVbroker

Sur la base du diagramme ci-dessus, nous savons que deux nouveaux matériaux sont générés à ce stade et qu'une nouvelle tâche doit être émise pour que le chariot les livre. Nous utilisons le contrôle d'entrée pour simuler les demandes émises (chaque fois qu'un article est sur le point d'entrer dans la station, une nouvelle demande est générée).



f) Commander le AGV pour ramassage le matériel

Effectuez la même opération que le point 2, sélectionnez un chariot et effectuez la tâche de ramassage du matériel, mais cette fois, la première destination de AGV doit être la station et la deuxième destination doit être l'endroit où le matériel généré par la station est stocké.

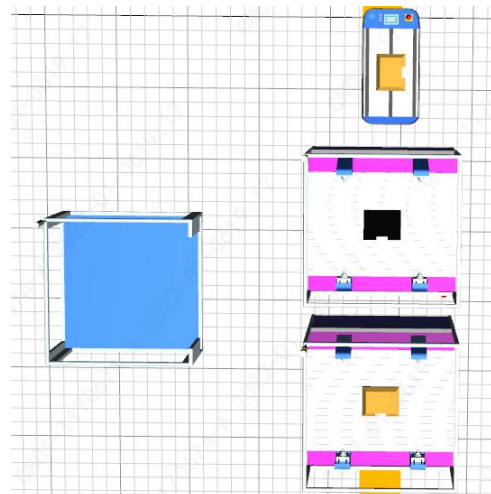
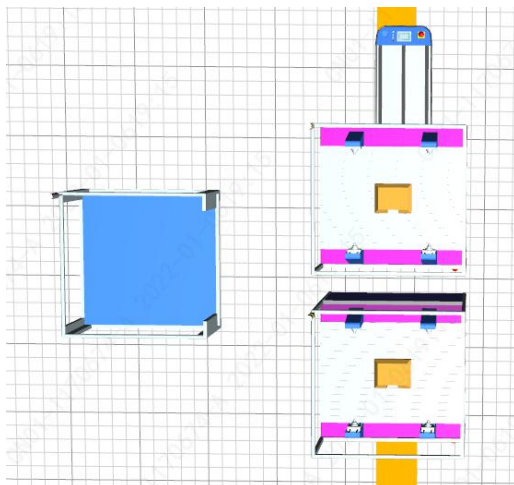
```
if task = "上料" or task = "SP_空盘"
  for var m := 1 to t_all.ydim
    if line = t_all["行号",m]
      veh.Destination := t_all[task,m]
      exitloop
    end
  next
  veh.next_destination := t_request["请求位置",num]
elseif task = "下料" or task = "UP_空盘"
  for var m := 1 to t_all.ydim
    if line = t_all["行号",m]
      veh.next_destination := t_all[task,m]
      exitloop
    end
  next
  veh.Destination := t_request["请求位置",num]
end
```

Le contenu de cette section exécuté

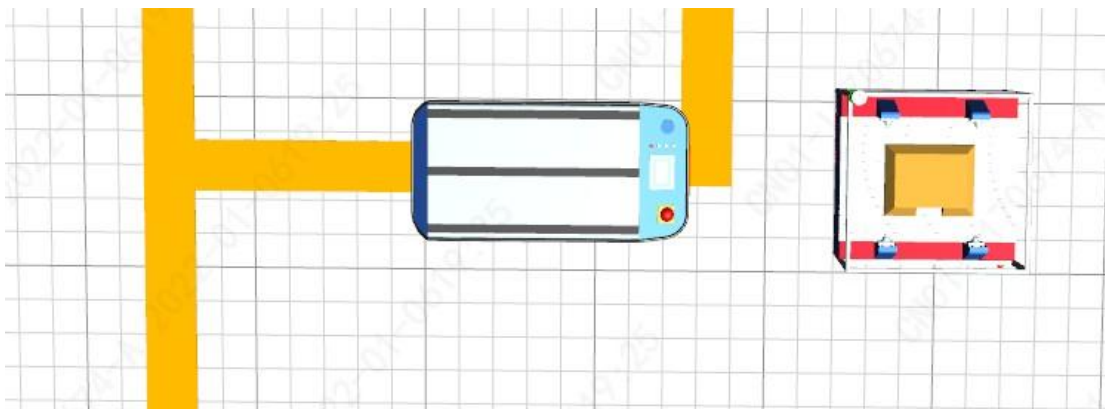
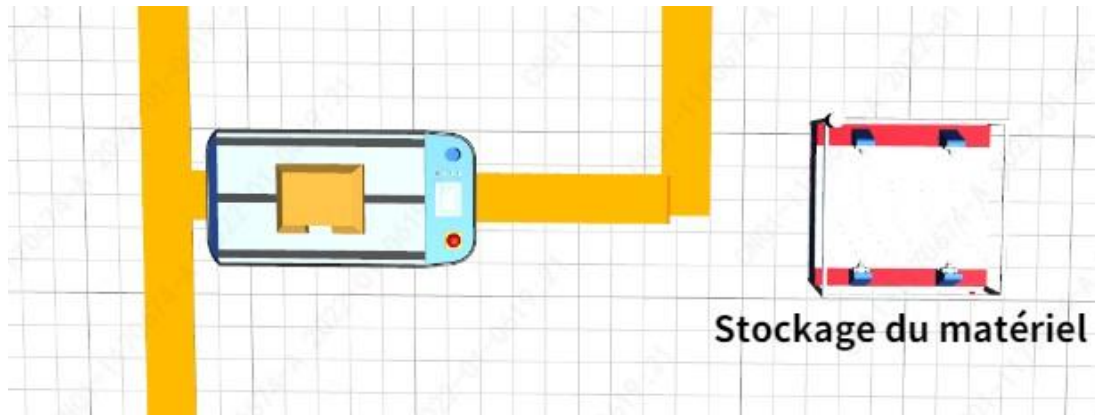
g) Ramassage des matériaux

L'AGV arrive d'abord à la station et acquiert le matériel généré par le poste de travail. À ce stade, le capteur exécute la méthode de chargement de l'AGV, déplaçant le produit du poste de travail vers l'AGV. La destination de l'AGV est alors modifiée pour correspondre à l'endroit où le matériel est stocké.

Nous pouvons voir que le matériel a été déplacé vers l'AGV, qui se dirigera ensuite vers la zone de stockage du matériel.



Les AGV stockent les matériaux dans la zone de stockage



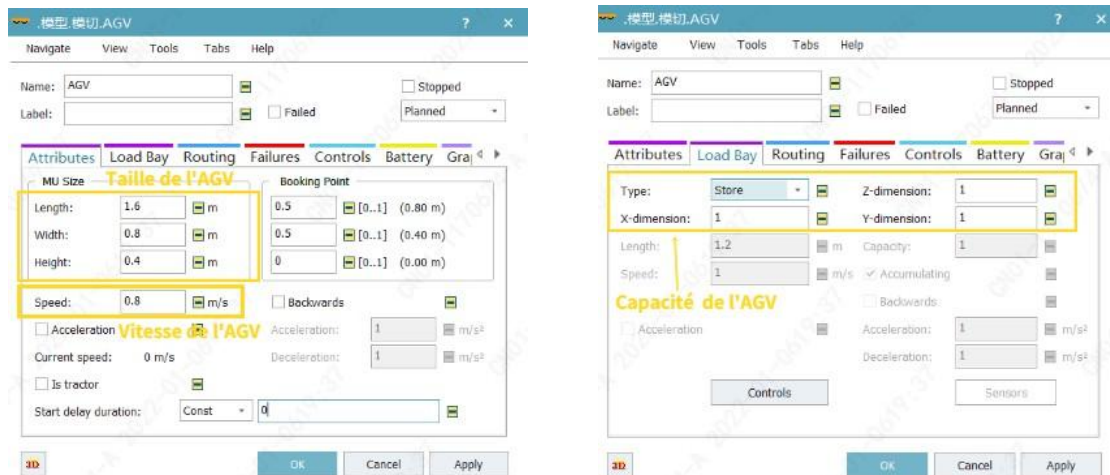
Une fois que l'AGV a terminé sa tâche, il effectue le même processus qu'à l'étape 3, en appelant la méthode "m_free" et en demandant au "AGV_broker" s'il a d'autres tâches à effectuer, en prenant différentes actions en fonction du retour d'information.

D. Optimisation de la simulation

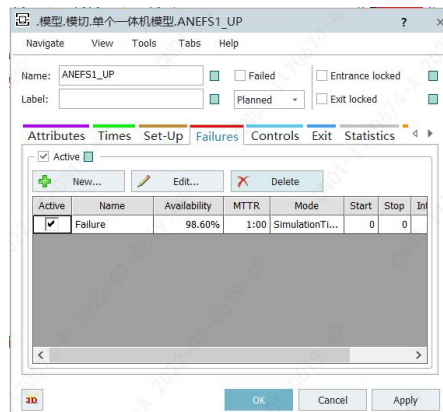
Pour l'instant, j'ai mis en œuvre le processus de base du fonctionnement de l'AGV, mais nous devons encore ajouter quelques informations supplémentaires pour le rendre aussi réaliste que possible.

a) Sélection des paramètres de l'équipement

Nous devons d'abord ajuster la longueur, la largeur, la vitesse et la capacité de chargement de l'AGV.



Sachant que les équipements peuvent aussi tomber en panne en cours de fonctionnement, nous avons compté les pannes des lignes de production existantes et sommes arrivés à un taux d'échec d'environ 98,6 %. Ceci est pris en compte dans le modèle de simulation.



À cela s'ajoutent de nombreux paramètres à régler, comme le temps que met la machine à traiter le matériau, le temps que met le chariot à prendre le matériau, etc.

b) Logique de chargement pour les AGV.

Les AGV consomment constamment de l'énergie pendant leur fonctionnement et nous devons en tenir compte dans nos simulations. Nous avons donc équipé les AGV de différents paramètres pour la situation énergétique.

Model: AGV

Label:

Attributes Load Bay Routing Failures Controls Battery Gra

Charge: 4 Ah

Basic consumption: 0 A

Capacity: 4 Ah

Charge current: 20 A

Charge control: self.OnCharge

Driving consumption: 4 A

Reserve: 0 Ah

Réglage des paramètres de la batterie pour chaque aspect de l'AGV

OK Cancel Apply

Afin d'éviter que le chariot ne soit incapable d'accomplir ses tâches en raison d'un manque d'énergie lors de leur exécution, le niveau d'énergie actuel du chariot doit être évalué après que le chariot a accompli chaque tâche, et s'il est supérieur au seuil, il continuera à fonctionner, et s'il est inférieur au seuil, il se rendra à une station de recharge pour se recharger.

```

if (veh.BatCharge / veh.BatCapacity) < root.BatReserve_rate
  for var j:= 1 to t_AGV.ydim
    if t_AGV["Agv",j] =veh
      t_AGV["Status",j] := false
      exitloop
    end
  next
  veh.Task := "充电"
  veh.destination := 充电站_使用情况["充电站",1]
  充电站_使用情况["统计",1] += 1
  充电站_使用情况.sort("统计","UP")
elseif obj.NumOrder >0
  obj.m_DoOrder
else
  veh.Task := "park"
  veh.destination := veh.Buffer
end

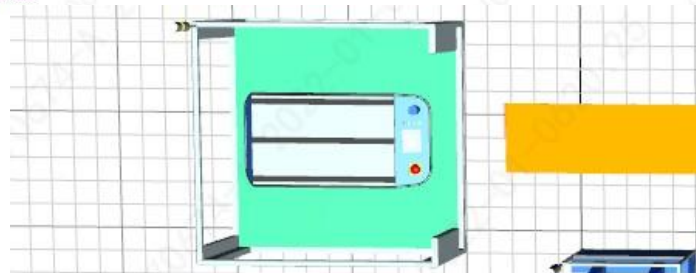
```

Comparaison des conditions de puissance et de seuil

Modification de la destination de l'AGV vers une station de recharge

Lorsque l'AGV entre dans la station de charge, nous activons l'interrupteur de charge de l'AGV et le laissons partir lorsqu'il est complètement chargé.

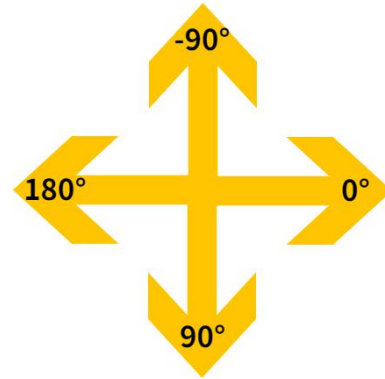
```
@.BatCharging := true
```



c) Simulation des virages de l'AGV

En plus des méthodes d'optimisation plus conventionnelles mentionnées ci-dessus, j'ai également ajouté une simulation spéciale des virages de l'AGV, ce qui a considérablement amélioré le réalisme de la simulation.

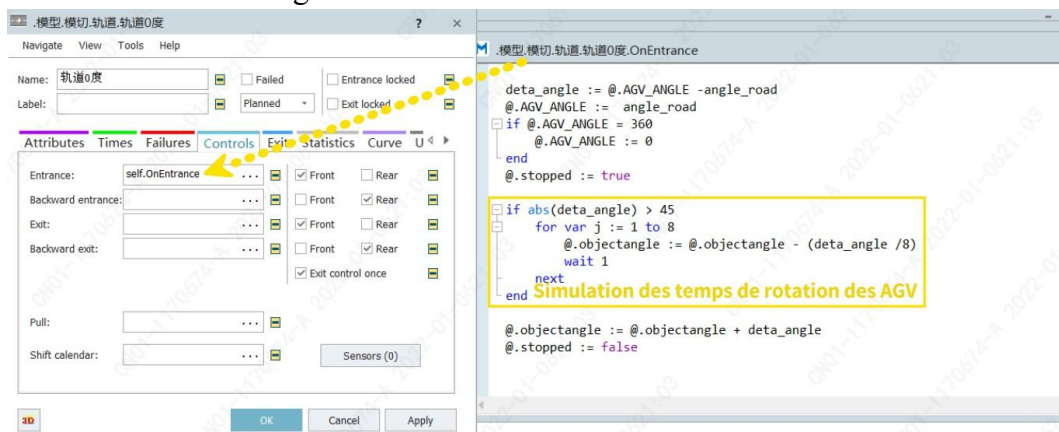
Notre route n'a que quatre directions : haut, bas, gauche et droite. J'ai donc donné les angles pour toutes les directions de déplacement en me basant sur le diagramme de droite.



Lorsque l'AGV doit tourner, nous calculons la différence d'angle en soustrayant l'angle actuel de l'angle après le virage, comme indiqué dans le tableau ci-dessous.

Angle initial \ Angle suivant	0°	90°	180°	270°(-90°)
0°	0°	-90° = 0° - 90°	-180°	90°
90°	90°	0°	-90°	-180°
180°	180°	90°	0°	-90°
270° (-90°)	-90°	180°	90°	0°

Ajoutez des contrôles d'entrée à l'entrée de chaque section de la route pour calculer l'angle auquel l'AGV doit tourner lorsqu'il entre, puis calculez le temps en fonction de la vitesse angulaire de l'AGV.



E. Méthodes d'optimisation

a) Exprimment Management :

La " Exprimment Management " permet d'analyser et de comparer les performances des systèmes avec différentes configurations de paramètres. Il permet d'exécuter facilement plusieurs simulations avec autant de paramètres que possible et d'analyser l'effet de chaque paramètre sur les performances du système, en générant les graphiques correspondants. [15] Je vais utiliser cet outil pour vous montrer comment choisir les meilleurs paramètres.

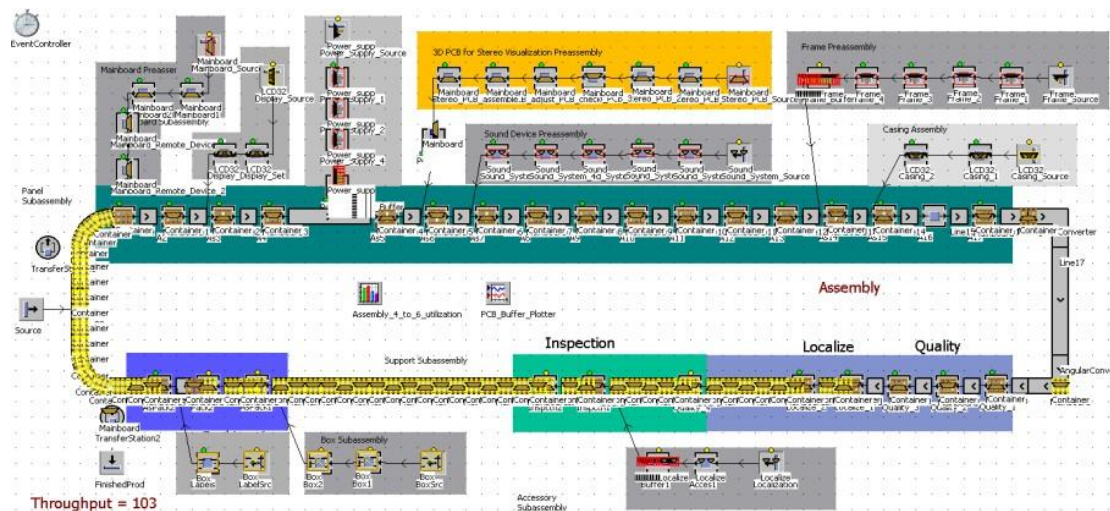


Figure 19 Dessin du modèle d'usine

Vous trouverez ci-dessous un modèle de simulation d'une ligne d'assemblage où nous voulons optimiser le nombre de palettes situées sur la ligne et la vitesse de la ligne pour lui donner un débit plus élevé.

- Le nombre de patins de transport a été évalué entre 10 et 100 par pas de 10.
- Vitesse de transport du convoyeur entre 1 et 5 m/s par pas de 1.

.Models.LCD_03.ExperimentManager1.GeneralDesign

For the multi-level experimental design you define

10

	Input value	root.Source.Number	.MaterialFlow.Line.Speed
1	Lower level	10	1
2	Upper level	100	5
3	Increment	10	1

Figure 20 Expérience 1

Nous allons générer le tableau expérimental suivant.

Numéro d'expérience	Nombre de patins	Vitesse	Sortie
Exp01	10	1	
Exp02	10	2	
Exp03	10	3	
...	
Exp49	100	4	
Exp50	100	5	

Après les calculs du logiciel, nous obtenons le graphique suivant, que nous pouvons trouver. Le rendement atteint son maximum dans le cas de l'expérience 12(30 patins, 2 m/s), puis commence à diminuer progressivement.

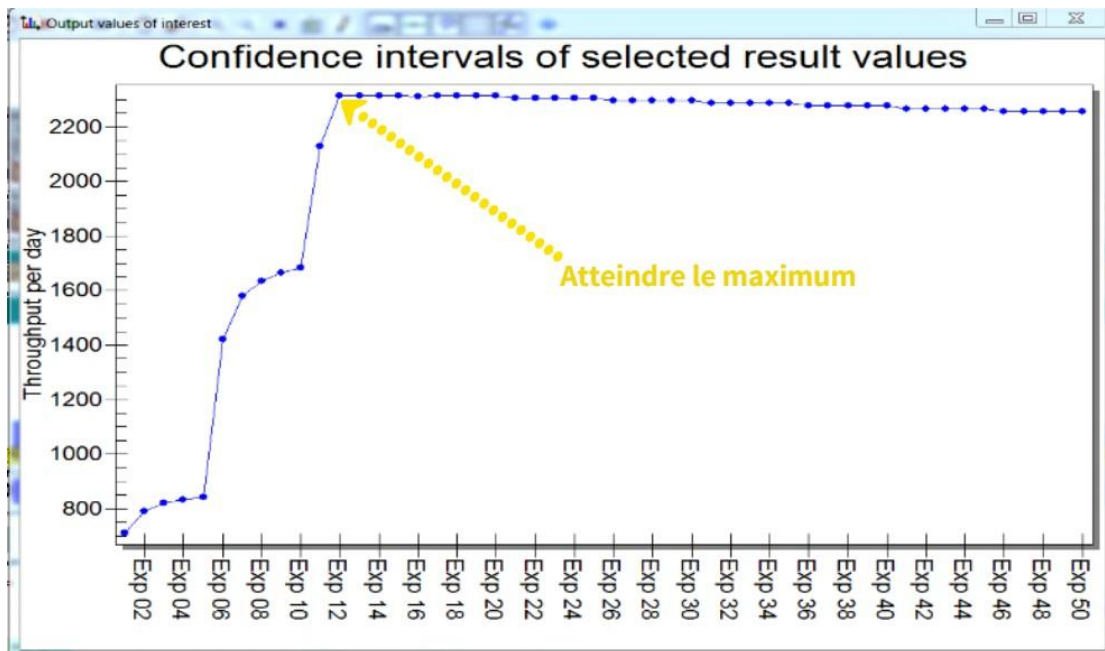


Figure 21 Résultats de l'expérience 1

Puisque nous avons vu que le nombre optimal de patins se situe autour de 30, nous exécutons les 441 combinaisons pour un nombre de patins compris entre 20 et 40 et une vitesse de transport comprise entre 2 et 4 m/s par pas de 0,1 m/s. de transport entre 2 et 4 m/s par pas de 0,1 m/s.

	Input value	root.Source.Number	.MaterialFlow.Line.Speed
1	Lower level	20	2
2	Upper level	40	4
3	Increment	1	0.1

Figure 22 Expérience 2

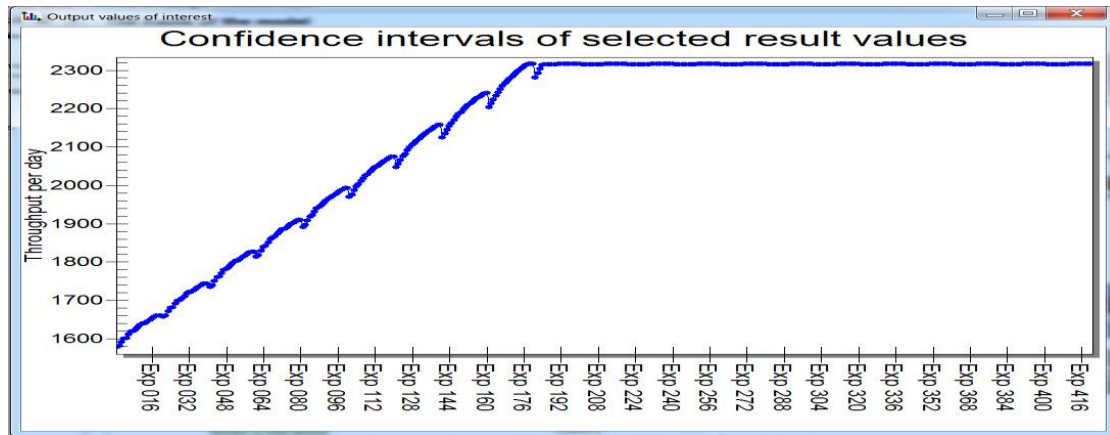
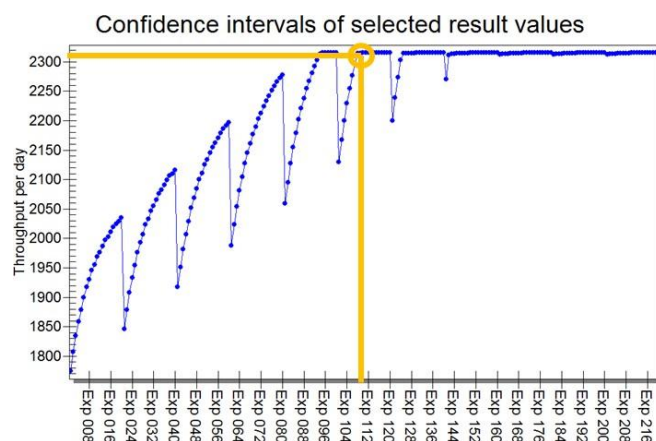


Figure 23 Résultats de l'expérience 2

Nous pouvons voir que jusqu'à l'expérience 179, la quantité de production augmente significativement en augmentant le nombre de patins et la vitesse du convoyeur. Cependant, à partir de l'expérience 185, il est difficile d'augmenter le rendement en augmentant à la fois la vitesse du convoyeur et le nombre de supports.

Nous avons compté le rendement le plus élevé de 2315, puis nous avons effectué des combinaisons entre 25 et 35 et 1 et 3 m/s pour trouver toutes les combinaisons ayant le même rendement.

	Input value	root.Source.Number	.MaterialFlow.Line.Speed
1	Lower level	20	2
2	Upper level	40	4
3	Increment	1	0.1



Skids	Conveyor speed	Output
30	1.9	2315
31	1.9	2315
32	1.9	2315
33	1.9	2315
34	1.9	2315
35	1.9	2315
30	2	2315
31	2	2315
32	2	2315
33	2	2315
34	2	2315
35	2	2315
30	2.1	2315
31	2.1	2315
32	2.1	2315
33	2.1	2315
34	2.1	2315
35	2.1	2315
30	2.2	2315
31	2.2	2315
32	2.2	2315
33	2.2	2315
34	2.2	2315
35	2.2	2315
30	2.3	2315
31	2.3	2315
32	2.3	2315

Figure 24 Expérience 3 et les résultats

La combinaison optimale combine un rendement maximal avec une vitesse minimale du convoyeur (sécurité, économies d'énergie) et un nombre minimal de patins et le nombre minimum de patins. Donc La combinaison optimale est

- ✓ **30 patins**
- ✓ **1,9 m/s Vitesse du convoyeur**

Conclusion

Pour conclure, ce stage de six mois a été une expérience enrichissante qui m'a permis d'apprendre beaucoup de choses nouvelles et m'a donné une vision approfondie du secteur des énergies nouvelles. J'ai été très inspiré par la possibilité d'utiliser mes connaissances pour développer des logiciels et des systèmes et de contribuer à l'entreprise en faisant le lien entre ce que j'ai appris à l'école et ce qui se passe réellement dans l'entreprise. J'ai beaucoup appris sur les algorithmes d'apprentissage profond en résolvant le traitement des images en python dans le cadre du projet. J'ai rencontré beaucoup de personnes très compétentes, ce qui a grandement amélioré mes connaissances du secteur de l'internet et m'a aidé à planifier ma future carrière.

En outre, grâce au système de simulation que j'ai construit, j'ai eu l'occasion d'observer des équipements physiques réels et j'ai pu me faire une idée des connexions entre les différents équipements et de la manière d'optimiser les paramètres des équipements pour tirer le meilleur parti de l'usine.

Les principales tâches que j'ai accomplies sont les suivantes.

1. L'ajout d'un logiciel d'inspection supplémentaire au processus de détection des défauts dans l'usine a permis de réduire de manière significative les taux de sur- et de sous-abattage.
2. achevé les modèles de simulation pour deux segments de l'usine et accéléré le processus de numérisation de l'usine.
3. fourni 40 suggestions d'amélioration du logiciel industriel de Sensetime et les a aidés à les mettre en œuvre.

Pour le secteur de l'internet, les projets à vocation industrielle requièrent non seulement nos connaissances frontales et dorsales, mais aussi la capacité d'aller plus loin dans la chaîne de production, de comprendre nos modèles de chaîne de production et de développer des systèmes réellement applicables à la chaîne de production, ce sur quoi j'espère travailler à l'avenir.

Enfin, je tiens à remercier mes tuteurs, mes collègues et ceux qui m'ont gentiment accueilli pendant mon stage. J'ai également beaucoup apprécié le cours.

- Et ceci conclut le rapport. Merci de votre attention !

Bibliographie

- [1] CATL. Présentation CATL. 2022*
- [2] site officiel de CATL. <https://www.catl.com/>*
- [3] CATL Classe de 2022 Document de présentation 2021.01*
- [4] Introduction au département IMD 2021.08*
- [5] Sensetime. Présentation de l'équipe. 2021.05*
- [6] CATL. Introduction aux principes de base des cellules de batterie V3.0, 2021*
- [7] CATL. Équipements et procédés de fabrication de cellules de batteries V2.0, 2021*
- [8] François Blayo et Michel Verleysen, Les réseaux de neurones artificiels, PUF, Que Sais-je No 3042 , 1^{re} éd., 1996*
- [9] Sensetime. Spécification du projet de soudage du couvercle supérieur, 2021*
- [10] Sensetime. Manuel d'utilisation du système AE V1.0, 2021*
- [11] Siemens PLM. Plant Simulation 2014.*
- [12] https://en.wikipedia.org/wiki/Automated_guided_vehicle*
- [13] « Technologie de fabrication aéronautique » Exploration de l'utilisation des modèles jumeaux numériques dans la gestion de la configuration des produits, 2017*
- [14] Siemens PLM. Plant Simulation and HEEDS Integrated Optimization-EN, 2019*