



UNIVERSITÉ DE  
SHERBROOKE

FACULTÉ DE GÉNIE

---

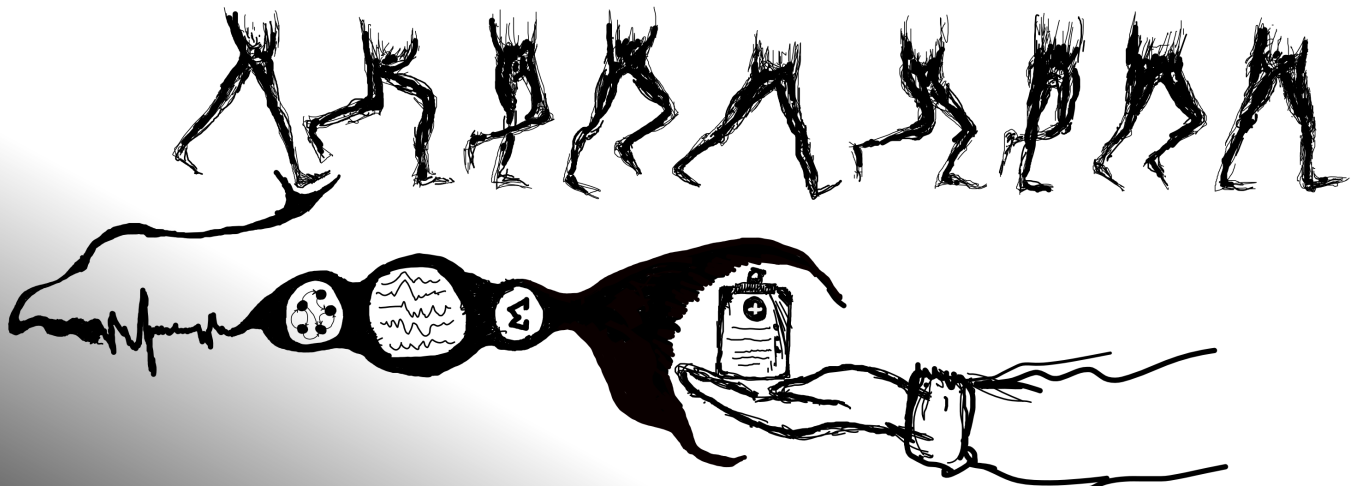
# Mise en application du reservoir computing pour l'analyse de la marche avec capteur IMU

---

	<i>Auteur</i>	>-12.0pt>-	<i>Email</i>	<i>Id</i>
12.0pt	Laurent Chiasson-Poirier		laurent.chiasson-poirier@usherbrooke.ca	15 098 528

Mémoire de maitrise présenté à :  
Pr Julien Sylvestre, PhD, Département Génie mécanique  
Pr. Karina Lebel, PhD, Département Génie électrique  
Pr. Abderraouf Boucherif, PhD, Département Génie mécanique

21 novembre 2022



## Sommaires

---

L'analyse de la marche est une science qui s'assure de comprendre le lien entre les mouvements impliqués lors de la marche et la progression de maladies dégénératives articulaires et musculaires des membres inférieurs, comme l'arthrose du genou. Des traitements cliniques non invasifs permettent maintenant de limiter cette progression en modifiant la démarche des patients. Afin d'améliorer ces interventions cliniques, les technologies portatives IMU ont pris une place importante permettant de suivre l'état des patients dans la vie quotidienne. Une grande variabilité de mouvement en milieu écologique nécessite toutefois l'utilisation d'algorithmes de traitement de plus en plus robustes pour un capteur portatif, en plus de considérer une mémoire, une consommation et un volume limités. La technologie de reservoir computing par système micro-électro-mécanique (RC-MEMS) développée à l'Institut Interdisciplinaire d'Innovation Technologique de l'université de Sherbrooke (3IT) est un capteur inertiel intelligent qui démontre des capacités de classification de signaux et de prédiction de fonctions temporelles non linéaires. Cette technologie se base sur le paradigme d'intelligence artificielle nommée le reservoir computing, conçue pour traiter des signaux temporels tout en limitant la complexité de la phase d'entraînement. La réduction de la taille du capteur et de la consommation énergétique de la technologie RC-MEMS fait de lui une alternative très intéressante pour un capteur portatif conventionnel. Dans ce projet, nous avons fait une preuve de concept de l'application du reservoir computing (RC) pour la résolution de trois tâches de prédictions biomécaniques pertinentes pour le traitement de l'arthrose : la détection des événements de marche, la classification de patrons de marche et la prédiction des forces aux sols. Nous avons testé un algorithme de reservoir computing numérique appelé le Echo states Network (ESN) pour tester ces tâches. Des critères de conception sur la taille et la vitesse de traitement du RC numérique ont été sélectionnés afin que ce traitement soit compatible avec l'application du RC-MEMS. Des recommandations sur l'implémentation dans le contexte clinique du réservoir computer sont identifiées.

Nos résultats montrent que l'algorithme du ESN est en mesure d'accomplir ces 3 types tâches. Les deux premières tâches, la prédiction des événements de marche (GED) et la classification de patron de marche (CPDM) sont accomplies avec un niveau de précision satisfaisant, et confirment que le RC peut réellement servir dans un contexte clinique. La troisième tâche, consistant à faire une prédiction des forces de réactions au sol (PFS), obtenait des résultats comparables à la moyenne des résultats obtenus dans la littérature, mais cela reste insuffisant en termes de précisions pour l'utiliser pour traiter l'arthrose. Une amélioration de la méthode d'entraînements, appelé le kernel training, spécifiquement appliqué pour le PFS permet de minimiser l'erreur en segmentant les données d'entraînement de la méthode RC sur des intervalles temporel de marche spécifique.

## Mots clés

---

Réseaux de neurone ; Reservoir computing ; Echo states Network ; Analyse de la marche ; Capteurs inertiels ; neuromorphic MEMS accelerometer

## Question de recherche

---

Comment le reservoir computing peut-il être utilisé pour l'analyse des signaux d'accélération de la marche ?

- Quelles sont les tâches de prédiction biomécanique utiles pour supporter un traitement clinique en lien avec la marche ?
- Comment chacune de ces tâches se définit-elle ? Quelles sont les données d'entrée et les données d'entraînement utilisées ainsi que les équipements nécessaires pour acquérir ces données ?
- Comment s'assurer que les tâches définies soient compatibles avec l'utilisation éventuelle d'un *reservoir computer physique* comme le MEMS-RC ?
- Quel est la performance, les avantages et les inconvénients du RC par rapport à d'autres types d'algorithmes implémentés pour ces tâches biomécaniques ?
- Quels sont les critères de succès de ces différentes tâches ?
- Quels sont les limitations du RC pour ces tâches biomécaniques ?

# Table des matières

<b>1</b>	<b>Mise en contexte sur l'utilisation du RC pour l'analyse de la marche</b>	<b>2</b>
1.1	Introduction à l'analyse de la marche . . . . .	3
1.1.1	Différentes conditions physiques liées à la marche . . . . .	3
1.1.2	Le rôle de l'analyse de la marche pour la prévention de ces conditions . . . . .	4
1.2	Concepts importants de l'analyse de la marche . . . . .	5
1.2.1	Phases fonctionnelles du cycle de marche . . . . .	5
1.2.2	Analyse de la fonction d'absorption dans le cycle de marche . . . . .	7
1.2.3	Diverses stratégies d'amélioration de la qualité de la marche . . . . .	9
1.3	Applications cliniques et intégration de technologies portatives . . . . .	11
1.3.1	Processus normal . . . . .	11
1.3.2	Intégration de technologies portatives . . . . .	11
1.3.3	Défis des technologies portatives en analyse de la marche . . . . .	12
1.3.4	Le reservoir computing comme méthode de traitement des données des capteurs portatifs . . . . .	13
1.4	Données expérimentales utilisées . . . . .	13
1.4.1	Bases de données . . . . .	13
1.4.2	Caractérisation des signaux IMU . . . . .	15
1.4.3	Caractérisation des événements de marche . . . . .	18
<b>2</b>	<b>Introduction au reservoir computing</b>	<b>21</b>
2.1	Mise en contexte sur les NN, RNN et RC . . . . .	22
2.1.1	Deep learning et réseaux de neurones (NN) . . . . .	22
2.1.2	Les réseaux de neurones récurrents (RNN) . . . . .	30
2.1.3	Principe émergent de RNN : les reservoirs computers (RC) . . . . .	33
2.2	Implémentation numérique de RC : echo state network . . . . .	34
2.2.1	Formulation du ESN . . . . .	34
2.2.2	Applications du ESN . . . . .	35
2.3	Implémentation physique d'un RC par MEMS-RC . . . . .	36
2.3.1	Similarité entre un oscillateur de duffing et une fonction d'activation numérique . . . . .	36
2.3.2	Méthode d'excitation et de lecture du reservoir MEMS . . . . .	36
2.3.3	Extraction et exploitation des données de sorties du RC-MEMS . . . . .	38
2.3.4	Avancements actuels de la technologie . . . . .	40
2.3.5	Retour sur les contraintes des RC-MEMS dans l'application des RC dans les tâches biomécaniques . . . . .	42
<b>3</b>	<b>Détection des événements de marche</b>	<b>44</b>
3.1	Abstract . . . . .	46
3.2	Introduction . . . . .	46
3.3	Methods . . . . .	48
3.3.1	Data Collection . . . . .	48
3.3.2	Data Formatting and Preprocessing . . . . .	51

3.3.3	ESN Procedure . . . . .	54
3.3.4	Comparative Procedure . . . . .	57
3.3.5	Performance Criteria . . . . .	58
3.4	Results . . . . .	60
3.4.1	Training Data . . . . .	60
3.4.2	Performance Under Various Walking Conditions . . . . .	62
3.4.3	Performance Between Orientations and Positions . . . . .	63
3.4.4	Performance According to Various Gait Event Classes . . . . .	64
3.4.5	Algorithms Comparison . . . . .	66
3.5	Discussion . . . . .	68
3.5.1	Training Characteristics to ensure good performance . . . . .	68
3.5.2	Robustness over Various Types of Input Data . . . . .	70
3.5.3	Robustness for Multi-events Detection . . . . .	71
3.5.4	Simplicity Compared to other Methods . . . . .	73
3.5.5	Clinical Perspectives and Further investigations . . . . .	73
3.6	Conclusion . . . . .	74
<b>4</b>	<b>Classification de patrons de marche</b>	<b>75</b>
4.1	Introduction . . . . .	76
4.1.1	Contexte d'application clinique . . . . .	76
4.1.2	Revue de la littérature sur la CPDM . . . . .	76
4.1.3	Perseptive de la méthode RC pour la classification . . . . .	79
4.1.4	Objectifs du chapitre . . . . .	80
4.2	Méthodologie . . . . .	81
4.2.1	Données expérimentales . . . . .	81
4.2.2	Définition des données d'entrées et objectives . . . . .	82
4.2.3	Configuration des données d'entraînement . . . . .	83
4.2.4	Construction des données par bloc continu . . . . .	84
4.2.5	Post-traitement des données de prédiction . . . . .	88
4.2.6	Métrique de performance . . . . .	91
4.3	Résultats . . . . .	91
4.3.1	Choix du nombre de points d'intégration . . . . .	91
4.3.2	Résultats globaux pour les différentes configurations . . . . .	92
4.3.3	Classification PDM avec des vitesses uniformes . . . . .	93
4.3.4	Classification PDM avec toutes vitesses confondue . . . . .	95
4.3.5	Classification selon la fréquence et la variabilité de la modification du PDM de la séquence continue . . . . .	96
4.4	Discussion . . . . .	98
4.4.1	Retour sur les performances globales du RC pour la CDPM . . . . .	99
4.4.2	Comparaison avec des travaux antérieurs de modèle AI . . . . .	100
4.4.3	Limitations et perspectives futures d'étude . . . . .	101
4.4.4	Fréquence de variation du PDM . . . . .	101
4.5	Conclusion . . . . .	102

<b>5</b>	<b>Prédiction des forces au sol</b>	<b>103</b>
5.1	Contexte . . . . .	104
5.1.1	Contexte d'application clinique . . . . .	104
5.1.2	Revue de la littérature sur l'estimation des forces au sol . . . . .	105
5.1.3	Perspective du RC pour la PFS . . . . .	109
5.1.4	Objectifs du chapitre . . . . .	110
5.2	Méthodologie . . . . .	111
5.2.1	Définition des données expérimentales . . . . .	111
5.2.2	Définition des données d'entrées et objectives du RC . . . . .	113
5.2.3	Rappel de la méthode d'entraînement RR standard . . . . .	115
5.2.4	Méthode d'entraînement par kernel . . . . .	116
5.2.5	Pondération multivariable . . . . .	124
5.2.6	Évaluation de la performance . . . . .	129
5.3	Résultats . . . . .	131
5.3.1	Comparaison globale des résultats sur les positions et les axes du corps . . . . .	134
5.3.2	Comparaison entre les variables de pondération . . . . .	135
5.3.3	Choix d'une position et d'un axe d'accélération optimale pour le capteur . . . . .	136
5.3.4	Résultats avec plusieurs variables de pondération . . . . .	137
5.4	Discussion . . . . .	143
5.4.1	Comparaison avec la littérature . . . . .	143
5.4.2	Performance globale et pistes d'amélioration . . . . .	143
5.5	Conclusion . . . . .	144
<b>6</b>	<b>Conclusion</b>	<b>146</b>
6.1	Rappel des objectifs . . . . .	146
6.2	Sommaire . . . . .	146
6.2.1	Définition de tâches d'analyse biomécaniques . . . . .	146
6.2.2	Définition des contraintes de conception du RC . . . . .	147
6.2.3	Résumé des performances en biomécanique . . . . .	147
6.2.4	Limitation de la méthode RC . . . . .	148
6.3	Contributions . . . . .	149
6.4	Travaux futurs . . . . .	149
6.4.1	Application du ESN dans une situation clinique . . . . .	150
6.4.2	Démonstration expérimentale sur le RC-MEMS Choix de la tâche de démonstration . . . . .	150
<b>A</b>	<b>Procédure de synchronisation des données de force aux sols</b>	<b>151</b>
A.1	Objectif . . . . .	151
A.1.1	Nature des données . . . . .	151
A.1.2	Étape 1 : Échantillonnage homogène et filtrage des signaux . . . . .	151
A.1.3	Principe de synchronisation . . . . .	151
A.1.4	Étape 2 : Synchronisation de l'accélération avec la force de réaction au sol . . . . .	153
A.1.5	Étape 3 : Synchronisation entre les accélérations . . . . .	157
A.1.6	Étape 4 : Remise à l'échantillonnage d'origine . . . . .	158

<b>B</b>	<b>Identifying gait events with Ground reaction force preprocessing</b>	<b>159</b>
B.1	MAREA . . . . .	159
B.2	UDS . . . . .	159
B.3	CIRRS . . . . .	160
<b>C</b>	<b>Hyperparameters optimization</b>	<b>162</b>
C.1	Influence of peaks detection threshold . . . . .	164
<b>D</b>	<b>All walking conditions results of TKEO and TKEO algorithms</b>	<b>165</b>
D.1	Processing comparison between ESN and TKEO . . . . .	165

## Table des figures

1.1	Modèle d'analyse de la marche développé par Perry [1]	6
1.2	Évènements de marche selon les signaux de force au sol verticale et d'accélération	7
1.3	Représentation de divers paramètres cinématiques associé à des stratégies d'amélioration de la marche	10
1.4	Processus de traitement de conditions de marche et intégration de technologies portatives.	11
1.5	Comparaison entre une situation en laboratoire et une en milieu écologique pour l'analyse de la marche. Le milieu écologique possède une quantité de données restreintes et est composé de plus de facteurs de variabilités	12
1.6	Représentation de la structure du mémoire sur l'application des algorithmes de reservoir computer (RC) pour l'analyse de la marche.	14
1.7	Allure des accélérations dans le domaine temporel sur un cycle de marche.	17
1.8	Contenu fréquentiel des signaux d'accélération dans les bases de données.	19
1.9	Distribution des occurrences des évènements de marche.	20
2.1	Situation du concept de reservoir computing (RC) en AI.	22
2.2	Comparaison de la performance du DL et des autres méthodes de Machine Learning.	23
2.3	Principe général d'utilisation du NN : transformation non linéaire et augmentation de la dimension des données pour faciliter la séparation de l'information. [2]	24
2.4	Représentation d'un perceptron	24
2.5	Fonction d'activation $\sigma(x) = \tanh(x)$ couramment utilisée pour un ANN.	25
2.6	Représentation développée (à gauche) et compacte (à droite) d'un NN, composé de $N_{in}$ entrées et d'une couche cachée de $N$ perceptrons.	26
2.7	Représentation d'un NN à plusieurs couches cachées (DNN).	27
2.8	Représentation d'un RNN (à une couche cachée).	30
2.9	Représentation développée d'un RNN à une couche cachée.	32
2.10	Représentation d'un RC.	33
2.11	Architecture du traitement d'un ESN.	34
2.12	Réponse de déplacement pour une micro poutre bi encastree dans un régime de grande amplitude non linéaire correspondant au modèle de Duffing [3].	37
2.13	Similarité de la réponse de la fonction d'activation $\tanh(x)$ par rapport à celle de la réponse en vibration de la poutre (RC-MEMS) développée [3].	37
2.14	Traitement du signal par le système physique.	38
2.15	Illustrations de l'influence de l'accélération sur le signal de réponse de la poutre.	39
2.16	États du reservoir $\mathbf{X}(t)$ pour l'accélération d'un cycle de marche complet.	39



2.17	Image au microscope du dispositif. La masse inertielle (A) est suspendue sur un ressort (C), sensible aux accélérations extérieures. La drive (B) est solidaire à la masse inertielle et excite la poutre (D) électriquement. Les jauges piézoélectriques (E) permettent la lecture de la déflexion de la poutre. [4]	40
2.18	Aperçu de la taille du RC-MEMS actuelle (version 2020-2021)	41
2.19	Performance du dispositif physique dans des tâches testées dans le groupe de recherche.	41
2.20	Délai d'utilisation d'un RC-MEMS en fonction du nombre de noeuds et de la période d'échantillonnage.	43
3.1	Procedure used to evaluate the performance of the ESN and of the TKEO algorithms.	49
3.2	IMU and FSR positioning for the three databases. In CIRRIIS tests the IMU was placed outside the shoes using straps. For MAREA tests, the three IMU were attached with straps at the three positions shown and the FSR were placed into the shoe soles. In UDS tests, the IMU and the FSR were placed on a sock inside the shoes. The orientations are defined as the antero-posterior (AP), the medio-lateral (ML) and the vertical (V) axis	51
3.3	Walking tracks of UDS walking tests.	52
3.4	Event targets identification process from foot pressure measurements for the three databases.	53
3.5	ESN algorithm representation considering the HS and TO gait event classes for the output prediction.	55
3.6	Average and $\pm 1$ standard deviation of the MAE for the values reported in reference [5] and our own implementation of the TKEO algorithm.	58
3.7	NRMSE and MAE as a function of the length of the training set.	61
3.8	MAE (ms) of the prediction with the ESN on HS and TO events as a function of the training / testing databases combinaison.	62
3.9	Mean absolute error (MAE), False positive rate (FPR), Positive predictive value (PPV) and T1 score of the ESN in function of the subsets of walking condition. TPR, PPV and T1 scores are put on the same scale (0 to 1) for all subfigures in order to compare performance between event and between databases.	63
3.10	MAREA database results of MAE, TPV, PPV and T1 performance of the ESN for GED on subset MAR <sub>all</sub> .	65
3.11	Same as figure 3.10 for the CIRRIIS database. The ESN for GED on subset CIR <sub>all</sub> with foot IMU position.	66
3.12	Same as figure 3.10 for the UDS database. The ESN for GED on subset UDS <sub>all</sub> with foot IMU position.	66
3.13	Comparison of MEA, FPR, PPV and T1 score on the three databases for the ESN and the TKEO algorithms.	67
3.14	Comparison of the TKEO and ESN algorithms to other algorithms in terms of the TPR, the SPF and the mean and $\pm 1$ standard deviation of the MAE.	69

4.1	Corrélation entre les deux métriques ( <i>Complexity</i> et <i>Mobility</i> ) établies avec l'accélération enregistrée avec un capteur inertiel sur l'humain et qui sont corrélées au niveau de douleur (VAS) établie pour chaque patient sur une échelle de 0 à 10 [6] . . . . .	77
4.2	Application d'un CNN logiciel pour la classification de patients selon trois conditions selon [7]. . . . .	79
4.3	Représentation des accélérations pour les différents PDM inclus dans la base de données CIRRIS20. . . . .	81
4.4	Deux configurations pour la construction des signaux utilisées dans la phase d'entraînement et la phase de test. . . . .	84
4.5	Traitement des données de marche pour la tâche AD. . . . .	84
4.6	Transition entre un PDM N vers un PDM RtTOTL . . . . .	85
4.7	Protocole de construction des séquences continues des signaux d'accélération	86
4.8	Construction d'une séquence continue de données. . . . .	86
4.9	Construction de séquences de données avec différentes longueurs d'intervalle $l_{interv}$ de PDM. . . . .	87
4.10	Exemple de détection de divers types de marche normale et pathologique .	88
4.11	Représentation de la courbe ROC et du calcul de l'aire sous la courbe (AUC) . . . . .	90
4.12	AUC moyen en fonction du nombre de points d'intégration utilisé en post traitement. . . . .	92
4.13	Résultats de classification pour des données de marche avec vitesse fixe (ISpeed). . . . .	94
4.14	Courbes ROC obtenues pour les tests de classification avec les deux participants considérés (MPart) et par blocs continus. . . . .	95
4.15	Résultats de classification pour des données de marche mélangeant toutes les vitesses pour tous les patients (MSpeed). . . . .	96
4.16	Courbe ROC sur les 5 classifications de séquences continues avec toutes vitesses confondues et en considérant les deux participants (MPart) . . . . .	97
4.17	Évaluation de la performance de classification en fonction de la longueur des intervalles de changement des PDM (paramètres $l_{interv}$ ). . . . .	98
4.18	Évaluation de la performance de classification en fonction de la variabilité des intervalles de changement des PDM (paramètres $v_{interv}$ ). . . . .	98
5.1	Orientation des forces de réaction au sol mesurées par le tapis roulant instrumenté (Bertec, 1000 Hz). . . . .	111
5.2	Allure des forces de réaction au sol pour l'ensemble des données de la base de données CIRRIS22. De gauche à droite, les forces au sol dans l'axe antéro-postérieur, médio-latéral et verticale. Ligne noire : la moyenne, Ligne rouge : la médiane, surface gris foncés [25-75e] percentile, surface gris pâle [5-95e] percentile. . . . .	112
5.3	Contenu fréquentiel dans les signaux de force au sol dans la base de donnée CIRRIS22. . . . .	113
5.4	Positions et orientations des IMU dans la base de donnée CIRRIS22. . . . .	114
5.5	Méthode d'entraînement standard pour la PFS. . . . .	116

5.6	Amplitude au 95 <sup>e</sup> percentile des signaux de forces au sol dans les 3 axes (V, AP et ML) en fonction de la durée du cycle de marche ( $\overline{\Delta T_{GCp}}$ ) pour les différents enregistrements du CIRRI22. . . . .	118
5.7	Amplitude au 95 <sup>e</sup> percentile des signaux de forces au sol dans les 3 axes (V, AP et ML) en fonction de la durée de la phase d'appui ( $\overline{\Delta T_{PAp}}$ ) pour les différents enregistrements du CIRRI22. . . . .	118
5.8	Amplitude au 95 <sup>e</sup> percentile des signaux de forces au sol dans les 3 axes (V, AP et ML) en fonction de la proportion de la phase d'appui sur la durée total du cycle de marche ( $\overline{Prop_{PAp}}$ ) pour les différents enregistrements du CIRRI22. . . . .	119
5.9	Exemple de coefficients de pondération calculé en fonction des valeurs du paramètre temporel exploité . . . . .	123
5.10	Attribution des kernels gaussiens dans l'ensemble de l'intervalle de variabilité pour un nombre de séparation $N_{sep}=25$ et $\sigma=0.022$ . . . . .	124
5.11	Protocole de la méthode Kernel training. La variable $\mathbf{P}_p$ est la variable de pondération utilisée, et elle peut représenter l'un des paramètres temporels $\Delta T_{GCp}$ , $\Delta T_{PAp}$ ou $Prop_{PAp}$ . . . . .	125
5.12	Estimation des forces au sol en mode opération avec un entraînement par kernel sur une variable de pondération. . . . .	125
5.13	Représentation des fonctions de pondération individuelles pour chaque variable de pondération $f_{pondk1}(v_p1, I_v1)$ et $f_{pondk2}(v_p2, I_v2)$ , avec les centres $\mu_{k1}=0.25$ (bleu) et $\mu_{k2}=0.5$ (orange) pour deux variables [1]v et [2]v, chacune avec un intervalle $I_v1$ et $I_v2$ . . . . .	127
5.14	Représentation de la Fonction de pondération globale $f_{pondk1}(v_p1, I_v1) \times f_{pondk2}(v_p2, I_v2)$ dans l'espace des variables de pondération, avec les centres $\mu_{k1}=0.25$ et $\mu_{k2}=0.5$ . . . . .	127
5.15	Méthode d'interpolation des noeuds. . . . .	128
5.16	Estimation des forces au sol en mode opération avec un entraînement par kernel sur plusieurs variables de pondération . . . . .	129
5.17	RMSE sur l'estimation des forces AP pour les différentes positions de capteurs et combinaison d'axes d'accélération comme données d'entrées. . . . .	132
5.18	RMSE sur l'estimation des forces ML pour les différentes positions de capteurs et combinaison d'axes d'accélération comme données d'entrées. . . . .	133
5.19	RMSE sur l'estimation des forces V pour les différentes positions de capteurs et combinaison d'axes d'accélération comme données d'entrées. . . . .	134
5.20	Proportion selon les positions et les choix de combinaison d'axes donnant la prédiction optimal des forces au sol selon les trois orientations et selon les trois intervalles du cycle de marche. . . . .	136
5.21	Comparaison entre les variables de pondération de la durée du cycle de marche, de la proportion de la phase d'appui et de la durée de la phase d'appui. . . . .	137
5.22	Résultat du $MAE_{BW}$ pour les trois prédictions (AP, ML et V) en fonction des variables de pondération utilisées. . . . .	138
5.23	Représentation de l'erreur de prédiction sur l'ensemble des signaux de force au sol antéro-postérieur avec la combinaison TS/AP comme signal d'entrée. . . . .	140

5.24	Représentation de l'erreur de prédiction sur l'ensemble des signaux de force au sol médio-latérale avec la combinaison TS/AP comme signal d'entrée. . . . .	141
5.25	Représentation de l'erreur de prédiction sur l'ensemble des signaux de force au sol vertical avec la combinaison TS/AP comme signal d'entrée . . . . .	142
A.1	Position des 5 capteurs d'accélération . . . . .	152
A.2	Filtrage du signal de force afin d'éliminer le bruit haute fréquence . . . . .	152
A.3	Évènements de marche détectés avec les forces au sol et mise en évidence des évènements de marche sur le signal d'accélération, ici provenant de l'axe AP sur le dessus du pied. . . . .	154
A.4	Choix du redoux dans le signal de force vertical selon le minimum des tous les minimum locaux apparaissant lors du foot flat dans la phase d'appui du pied au sol . . . . .	155
A.5	Valeur de la fonction de corrélation en fonction du lag n choisis . . . . .	156
A.6	Aperçu de la synchronisation entre les signaux de force et de l'accélération TS156	
A.7	Synchronisation de tous les autres signaux d'accélération afin de maximiser la fonction de corrélation entre TS et les autres individuellement . . . . .	157
C.1	Relations between the CHARC metrics and the mean absolute error of the GED task for all events. . . . .	163
C.2	GED performance according to the temporal threshold for peak detection .	164
D.1	MAE, TPR, PPV and T1 comparison between the ESN and TKEO algorithms for HS and TO events for multiple walking condition subsets . . . .	166
D.2	Examples of predictions on a timeseries for the UDS database for the ESN and the TKEO. . . . .	167

## Liste des tableaux

1.1	Énumération des différentes tâches de traitement par RC . . . . .	14
1.2	Caractéristiques principales des bases de données. . . . .	15
1.3	Caractéristiques des participants pour chaque base de données (moyenne) ± (écart-type). . . . .	15
1.4	Caractéristiques des capteurs IMU utilisés pour chaque base de données . .	16
3.1	Databases characteristics . . . . .	50
3.2	Echo State Network Hyperparameters . . . . .	56
3.3	Classification of the target events (TE) and the prediction events (PE) for FN, TP and FP. . . . .	59
4.1	Attribution des cas de classification pour un des PDM . . . . .	89
4.2	Comparaison des résultats de classification pour tous les types de configura- tions des données d'entraînement. . . . .	92
5.1	Résumé des différents modèles de prédictions des forces au sol lors de la marche et la course. . . . .	106
5.2	Définition des axes utilisés dans le vecteur d'entrée $\mathbf{u}(t)$ du RC . . . . .	114
5.3	Calcul des différents paramètres de pondération pour la méthode Kernel training . . . . .	117
5.4	Meilleurs résultats de prédiction pour l'axe AP des GRF . . . . .	132
5.5	Meilleurs résultats de prédiction pour l'axe ML des GRF . . . . .	133
5.6	Meilleurs résultats de prédiction pour l'axe V des GRF . . . . .	134
C.1	Echo State Network Hyperparameters . . . . .	162

# Chapitre 1

## Mise en contexte sur l'utilisation du RC pour l'analyse de la marche

### Éléments clé

Ce chapitre introduit les concepts principaux de l'analyse de la marche (AM) afin d'identifier comment les algorithmes RC peuvent améliorer le processus de traitement clinique d'un patient.

- L'AM est une science qui étudie le mouvement de la marche pour prédire la progression de conditions physiques, souvent dégénératives, qui altèrent la marche d'une personne, comme l'arthrose des membres inférieurs.
- Un élément fondamental de l'AM est la séparation du cycle de marche par segments temporels. Chaque segment du cycle sont associé à des fonctions principales du mouvement de la marche.
- Les évènements temporels de la marche permettent de séparer les segments du cycle de marche, en plus de fournir des informations temporelles comme la cadence et la période d'appui, utiles pour évaluer l'état physique du patient.
- L'absorption des chocs est la fonction la plus importante pour prévenir la progression de l'arthrose. Lors d'un processus clinique préventif de l'arthrose, des stratégies (mouvement forcés par le patient) peuvent être utilisées pour améliorer l'absorption des chocs pendant la marche.
- Le processus clinique est d'améliorer la détection de la posture ou une mesure de l'amortissement des jambes durant la marche à l'aide de capteurs portatifs permettant d'obtenir des informations continues sur la marche du patient, comme le suivi d'une cadence adéquate.
- Les capteurs portatifs étant contraints à des données limitées (données inertielles) incluant plus de variabilité, des algorithmes plus robustes doivent être utilisés afin d'analyser la marche. Le concept de RC est une méthode AI qui pourrait améliorer l'analyse de la marche avec des capteurs inertiels.
- Le RC sera donc utilisé dans ce mémoire pour prédire certaines informations pertinentes dans le contexte de réadaptation clinique pour faire la prédiction des évènements de marche, la classification de patrons de marche et l'évaluation des forces de réaction au sol. Le chapitre 2 introduit le concept de RC et les chapitre 3,4,5 démontrent son application dans chaque type de prédiction.

## 1.1 Introduction à l'analyse de la marche

---

Lorsqu'un humain apprend à marcher, il adapte naturellement son patron de marche de manière à minimiser son effort et les contraintes dans ses articulations. Des conditions physiques, souvent dégénératives, peuvent toutefois altérer la mobilité et la capacité musculaire d'une personne [1]. Ces conditions font apparaître des mouvements compensatoires qui modifient progressivement la démarche naturelle [1], et entraînent une augmentation significative des efforts ainsi qu'une limitation des déplacements au quotidien [1]. L'analyse de la marche permet de mieux comprendre comment évoluent ces diverses conditions et comment les traiter. Elle s'intéresse à la fois aux aspects cinématiques, dynamiques et énergétiques du corps en locomotion (en marche ou en course).

L'utilisation de technologies portatives pour l'analyse de la marche a grandement augmenté dans les dernières années, principalement en se basant sur des systèmes de mesure inertielle (IMU). Les IMU sont maintenant très abordables et compacts. Ces technologies donnent la possibilité d'analyser des patients dans des contextes réels, à l'extérieur du laboratoire et sur de longues durées en temps réel. Ils peuvent être utilisés dans le suivi de la progression d'une condition mais aussi comme système actif, comme un système de biofeedback ou un système de stimulation électrique fonctionnelle (FES).

Le développement d'algorithmes de traitement pour les données IMU est une sphère de recherche active en biomécanique. Les algorithmes issus du domaine de l'intelligence artificielle sont maintenant grandement exploités, pour leur simplicité et leur robustesse, ainsi qu'en raison du nombre important de données disponibles pour en faire l'entraînement. Ce mémoire s'intéresse à l'utilisation d'une branche de l'AI nommée le *reservoir computing* (RC) afin de faire le traitement de signaux d'accélération d'un capteur inertiel. On s'intéresse à des tâches d'analyse biomécanique aidant à la prévention de conditions comme l'arthrose des membres inférieurs ou des maladies neurologiques.

Ce chapitre résume diverses conditions physiques qui détériorent la marche et qui seront énumérées dans la section 1.1.1, ainsi que le rôle de l'analyse de la marche pour le suivi et le traitement de ces conditions. Ensuite, une brève introduction des concepts fondamentaux de la science de l'analyse de la marche est présentée dans la section 1.2. La section 1.3 aborde comment l'analyse de la marche permet de prévenir et de traiter la progression de ces conditions, et comment l'intégration des technologies portatives peut améliorer ce processus de suivi et traitement. On introduit finalement les défis liés à l'utilisation des technologies portatives, et comment les RC peuvent aider à améliorer le traitement des données dans les technologies portatives. Trois tâches de prédiction permettant de faire l'analyse biomécanique de la marche ont été sélectionnées : la détection des événements de marche, la classification de patrons de marche et de la prédiction de signaux de force. Une dernière section introduit les bases de données de marche utilisées afin de tester ces différentes tâches biomécaniques par le RC.

### 1.1.1 Différentes conditions physiques liées à la marche

---

De nombreuses conditions orthopédiques et neurologiques altèrent le mouvement de marche d'une personne. Au Canada, on compte environ 7 % de la population atteinte d'une condition d'arthrose dans les articulations des membres inférieurs (genoux et hanches) [8]. Cela représente environ 2,7 millions de canadiens, avec une prévalence élevée chez les per-

sonnes âgées de 65 ans et plus [8]. Cette condition produit de l'inflammation des tissus articulaires en lien avec la détérioration mécanique graduelle du cartilage, occasionnant un contact direct des extrémités osseuses dans l'articulation. La détérioration progressive du cartilage peut s'étaler sur plusieurs années, et certains patients peuvent ressentir des symptômes de douleurs jusqu'à 10 ans avant d'obtenir un diagnostic [8]. Plusieurs études observent que la vitesse de progression de la problématique d'arthrose est proportionnelle à l'amplitude des forces de contact imposées aux articulations lors de la marche [9]. Les personnes atteintes d'obésité, environ 28,2 % de la population canadienne en 2020 [10], sont grandement à risque de développer de l'arthrose au genou. Il en va de même pour les patients amputés, en raison d'un mouvement compensatoire développé dans leur démarche qui augmente significativement les contraintes articulaires dans leur membre sain [11, 12]. D'autres problèmes d'ordre neurologique, touchant environ 2.7% de la population canadienne, peuvent altérer la marche incluant notamment les accidents vasculaires cérébraux (0.9%), l'épilepsie (0.4%), la sclérose en plaque (0.3%), le parkinson (0.2 %) et la paralysie cérébrale (0.1 %) [8]. Les causes de maladies neurologiques sont principalement d'ordre génétique [13, 14].

### 1.1.2 Le rôle de l'analyse de la marche pour la prévention de ces conditions

---

L'AM peut être faite pour évaluer et soutenir le plan de traitement dans plusieurs cas. Par exemple [15] :

- Évaluation de l'état de progression de la pathologie d'un patient selon différentes conditions de paralysie cérébrale [16–18], de l'arthrose ou du parkinson [19–22].
- Évaluation de la déviation de la marche, des mouvements compensatoires, des pertes d'efficacité pour l'ajustement des prothèses pour des patients amputés [12, 23].
- Évaluation de la marche en pré-opération et en post-opération d'une arthroplastie totale des hanches ou des chevilles arthritiques (remplacement de l'articulation par une prothèse artificielle) [11, 12, 24].

Dans ces diverses situations, les méthodes préventives de rééducation à la marche jouent un rôle fondamental. Le travail d'un clinicien est de minimiser la gravité de ces conditions. On peut voir dans la section 1.1.1 que l'arthrose au genou est la condition la plus répandue parmi les problématiques reliées à la marche. Elle prend plusieurs années avant d'atteindre un niveau élevé de gravité ou de douleur. La rééducation à la marche permet de réduire la progression de ces conditions [9, 19]. Un ajustement adapté du patron de marche, en adoptant une position de pied externe [25] ou de tronc incliné, permet de réduire significativement la douleur dans l'articulation dans le cas de patients atteints d'arthrose.

D'autres démarches cliniques en lien avec le parkinson ont aussi été explorées pour modifier le patron de marche afin de réduire les risques associés (chutes) [21] et de caractériser la progression de cette condition [19, 20].

Des interventions clinique suite à des opérations d'amputations [11] ou d'arthropathies [24] peuvent aussi permettre de limiter les impacts résiduels après l'opération. Le clinicien corrige la démarche afin de favoriser la réduction des contraintes articulaires et la demande énergétique, favorisant la mobilité [15].



La section 1.2 présente les concepts utilisés en clinique pour caractériser et améliorer la qualité de marche d'un patient. La section 1.3 aborde comment se déroule une procédure rééducation à la marche et explique comment s'intégrerait un capteur portatif dans ce processus.

## 1.2 Concepts importants de l'analyse de la marche

---

Cette section introduit trois aspects importants de l'analyse de la marche, soit la séparation du cycle de marche en phases fonctionnelles, la caractérisation des impacts articulaires absorbés par le corps lors de la marche, et les différentes stratégies de marche utilisées en clinique pour le traitement de conditions dégénératives.

### 1.2.1 Phases fonctionnelles du cycle de marche

---

Le cycle de marche est un mouvement complexe dans son ensemble. Il implique le changement continu et périodique de l'alignement des parties inférieures du corps et de la sollicitation de différents groupes musculaires. Un concept fondamental de l'analyse de la marche consiste à segmenter le cycle de marche en différentes phases. Chaque phase représente une portion du mouvement global. Cela permet d'associer de manière plus spécifique chaque sous-mouvement en intervalles temporels et d'associer les muscles qui y sont sollicités. Le modèle d'analyse fonctionnelle (figure 1.1) de la marche proposé par Perry [1] est une référence bien établie dans le domaine.

Il est composé de deux phases principales (la phase d'appui et la phase de swing), et de huit phases secondaires. Des fonctions biomécaniques sont associées à chacun de ces sous-mouvements du cycle. Ces fonctions peuvent être liées à l'amortissement d'impact (réduction des contraintes articulaires), au support, à la stabilité, à la propulsion et à la progression [1]. Les phases du contact initial et celle du loading response sont considérées comme les plus exigeantes pour le corps d'un point de vue des contraintes articulaires, alors que la phase de swing est considérée comme la plus exigeante au niveau de l'effort musculaire [1]. Selon le modèle de Perry [1], la fonction d'amortissement est la plus importante pour les contraintes articulaires et joue un rôle majeur dans la progression de la condition d'arthrose.

On utilise les événements de marche (colonne F) afin de repérer dans quelle phase du cycle se trouve le pied d'un marcheur. Ces événements définissent des repères temporels dans chaque cycle détectable par des capteurs de force au sol pour les événements de la phase d'appui ou par des mesures de position et de mouvement pour les événements de la phase de swing. Les deux événements principaux sont le heel strike (HS) et le toe off (TO). Ils permettent de faire la séparation entre les deux phases principales du cycle de marche. Ces phases permettent d'évaluer plusieurs paramètres temporels de la marche comme la cadence (nombre de cycles par seconde), la période d'appui et la proportion entre l'appui et le swing. Ces paramètres sont des indicateurs de base de la présence d'anomalies de la marche. En effet, des symptômes de douleur sur un membre inférieur entraînent généralement une baisse de la cadence et une diminution de la proportion de temps de la phase d'appui sur la jambe atteinte, ou une asymétrie de la durée des phases entre les deux jambes. La variation de ces paramètres dans le temps peut aussi être un indicateur pour certains problèmes neurologiques. La marche dite normale présente des

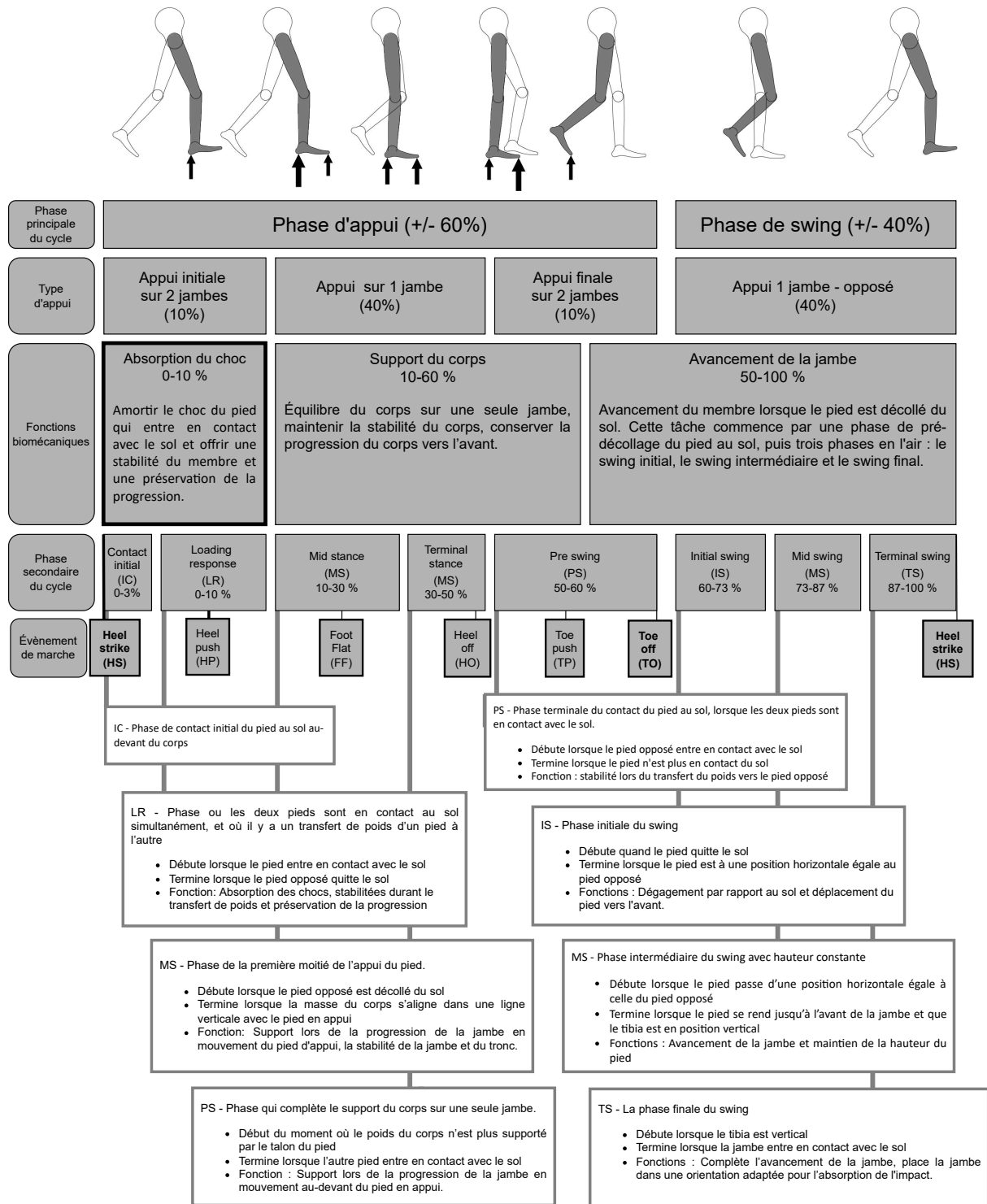
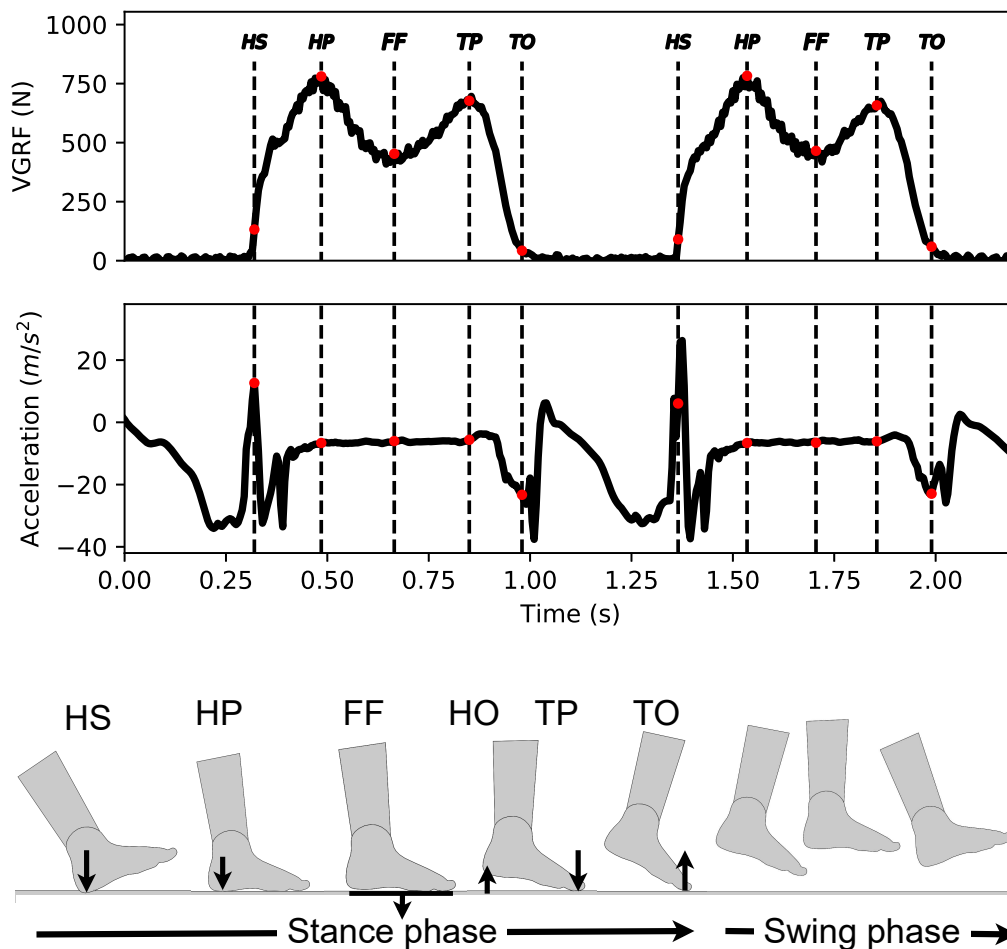


Figure 1.1 – Modèle d'analyse de la marche développé par Perry [1]

phases de durée stables dans le temps [26]. Les problèmes neurologiques entraînent souvent des phases de marche de durées inégales [19].

L'ensemble des évènements de marche permettent d'évaluer avec un peu plus de précision les phases secondaires de la phase d'appui. On voit où se situe ces évènements lors

du cycle de marche dans la figure 1.2. Le Heel-Push (HP) définit l'instant où la pression en dessous du pied est maximale. Le foot-flat (FF) représente un évènement qui se trouve environ au centre du mid-stance (MSt) et le heel-off (HO) correspond approximativement au début du mid-stance (MSw). Le toe-push (TP) est un évènement qui se situe au centre du pré-swing. Ces 4 autres évènements permettent de repérer dans le temps la contraction de différents groupes musculaires. Cela est utilisé en stimulation électrique fonctionnelle afin de synchroniser une impulsion électrique et la contraction d'un muscle dans le cycle de marche [27].



**Figure 1.2** – Évènements de marche selon les signaux de force au sol verticale et d'accélération

En somme, la détection des évènements de marche est une analyse fondamentale afin d'évaluer les paramètres temporels de la marche et de segmenter le cycle de marche pour d'autres analyses plus poussées.

### 1.2.2 Analyse de la fonction d'absorption dans le cycle de marche

La fonction d'absorption des chocs a pour but de réduire l'impact dans les articulations et de réduire la détérioration mécanique des tissus articulaires [1]. L'absorption est importante surtout durant la phase de loading response.

En effet, juste avant l'évènement de marche du contact initial du talon au sol (HS), le corps est en chute libre vers l'avant pour une distance d'environ 1 cm [1]. Malgré cette faible distance, cela crée un impact abrupt sur la jambe en progression. Le mouvement naturel de marche est adapté pour absorber ce choc, car la jambe crée une flexion de la cheville vers l'externe et une flexion du genou qui allongent la durée d'entrée en contact du pied avec le sol. Cela allonge le temps d'application de la charge sur le pied et minimise la contrainte ressentie dans l'articulation [1]. Des stratégies additionnelles peuvent permettre de réduire les contraintes articulaires, comme discuté dans la section 1.2.3.

Des métriques permettent de quantifier la qualité de l'absorption de choc durant la marche. Les trois métriques principales considérées sont :

1. la force de réaction au sol,
2. le moment abducteur au genou,
3. une mesure calculée des forces de contact articulaires.

Ces métriques sont mesurables par l'intermédiaire d'instruments de mesures directes non invasives comme :

- des marqueurs de positions infrarouges,
- des tapis roulants instrumentés,
- des électromyogrammes (mesure l'activité électrique en surface des muscles),
- des caméras de capture d'images haute résolution,
- des IMUs.

Ces instruments permettent d'établir des mesures du mouvement des membres, de l'activité musculaire et de la pression au sol. Différents outils numériques développés en analyse de la marche nous permettent ensuite d'utiliser ces données afin de remonter aux métriques de qualité de la marche. Chacune de ces métriques présente un compromis entre la pertinence dans l'évaluation de l'arthrose et la complexité de mesure.

La force de réaction se mesure facilement à l'aide d'une plateforme de force ou d'un tapis roulant instrumenté. Toutefois, ces instruments permettent d'évaluer la force au point de contact au sol du pied et n'est pas directement représentative des contraintes dans les articulations du genou.

Une mesure des contraintes internes de l'articulation est, à l'opposé, très pertinente, mesurant directement la valeur de la contrainte mécanique dans le cartilage. Elle implique toutefois un grand niveau de complexité, combinant beaucoup de mesures expérimentales, incluant des mesures des forces au sol, des mesures électromyographiques (EMG) et des mesures de position par marqueurs infrarouges, avec un modèle musculosquelettique très lourd en calcul numérique [28]. Ce processus est donc couteux en instrumentation, en expertise pour manipuler ces instruments et en temps de calcul.

Le moment abducteur du genou (KAM) est considéré comme la métrique offrant le meilleur compromis entre la pertinence et la complexité d'évaluation. Elle demande de mesurer les forces de réaction au sol ainsi que la position des membres inférieurs, à l'aide d'instruments de capture d'image ou de caméra avec des marqueurs infrarouges. La mesure du KAM exige donc un niveau de complexité modéré au niveau de l'instrumentation et assez faible au niveau du calcul. De plus, contrairement aux forces au sol, le KAM, est directement associé à la progression de l'arthrose [29] et se définit comme une mesure de

substitution des contraintes articulaires en clinique [29]. Le KAM est l'une des mesures la plus utilisée dans le domaine du traitement de l'arthrose du genou pour évaluer la qualité de la marche d'un patient [9].

En somme, plusieurs signaux biomécaniques peuvent être utilisés lors d'une évaluation afin d'établir de manière quantitative la qualité de la marche et de la fonction d'amortissement de choc. Le KAM est le signal le plus utilisé en clinique pour évaluer la qualité de la marche, car il est relativement facile à mesurer et représentatif des contraintes articulaires. Les forces au sol sont très faciles à mesurer, mais sont moins pertinentes, alors qu'une mesure directe des contraintes articulaires est très pertinente, mais très complexe à obtenir. Ces différents signaux demeurent normalement mesurables dans un contexte de laboratoire, mais l'utilisation de capteurs portatifs pour la prédiction de ces signaux est très pertinente. En effet, le laboratoire permet de capturer la marche seulement un court instant, dans un environnement précis et dans des tâches contrôlées, alors qu'un suivi continue, en plus d'évaluer ces mesures dans un contexte écologique réel, capture la variabilité dans le temps de la marche (comme la douleur qui change quotidiennement).

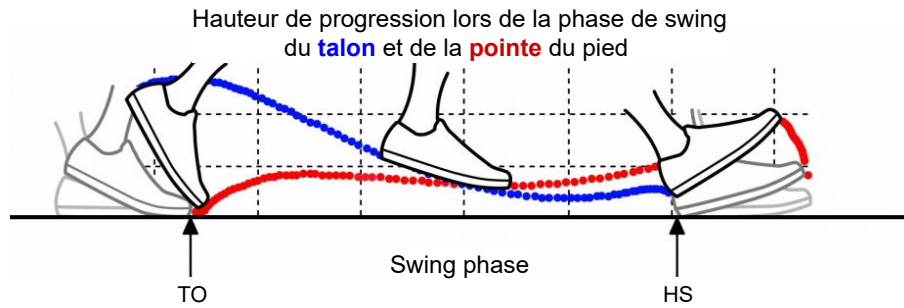
### 1.2.3 Diverses stratégies d'amélioration de la qualité de la marche

Beaucoup d'études ont établi des liens entre différents paramètres cinématiques de la marche (représenté dans la figure 1.3) et leur impact sur la condition d'un patient.

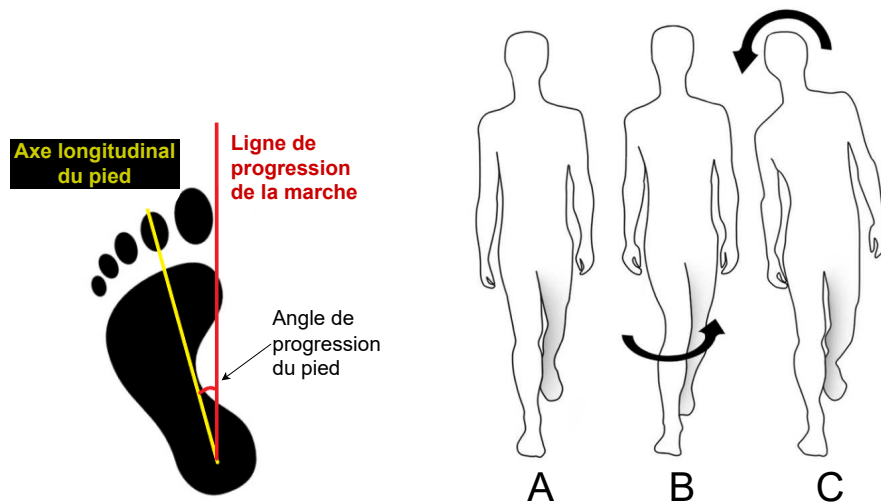
Par exemple, la mesure de la hauteur de progression (figure 1.3a) du pied est associée au risque de chute chez les personnes âgées [33]. De plus, des études [25, 34] montrent la corrélation entre l'angle de progression du pied (figure 1.3b) adopté lors de la marche et la détérioration du cartilage articulaire. Elles ont démontré que l'utilisation d'une démarche à angle ouvert ou fermé peut réduire la contrainte dans le genou. L'inclinaison latérale du tronc et le déplacement du genou le plus affecté vers l'intérieur pendant la phase d'appui, aussi appelé la médialisation du genou (figure 1.3c), sont d'autres postures qui permettent de ralentir la progression de l'arthrose au membre inférieur [31]. L'angle du genou dans le plan frontal et sagittal (figure 1.3d) a aussi été identifié comme un paramètre significatif dans la réduction des moments abducteurs du genou [32, 35].

Des travaux ont aussi accordé une importance à la régularité de la vitesse et de la cadence de marche, notamment chez des patients atteints de paralysie cérébrale [36] ou pour des populations vieillissantes [26]. L'augmentation légère de la cadence de marche autour de 100 – 120 pas par minute permet notamment de réduire le KAM [9]. Un paramètre nommé le *Gait deviation index* est aussi utilisé afin d'évaluer la corrélation entre les accélérations du corps et divers problèmes en lien avec la marche, notamment l'arthrose et la paralysie cérébrale [37].

On peut fournir des piste de solution dans un traitement clinique basées sur ces caractéristiques du mouvement identifiées dans la littérature, comme la longueur de foulée, la hauteur du pied dans la phase de swing, l'angle de progression du pied durant la foulée, la durée des diverses phases de la marche, la force d'impact du pied mesurée au sol ou la symétrie du mouvement entre les deux pieds. Dans un contexte clinique, un capteur portatif pouvant détecter l'adoption de ces divers mouvements du patient à un potentiel d'augmentation de l'efficacité d'un traitement clinique de rééducation à la marche.

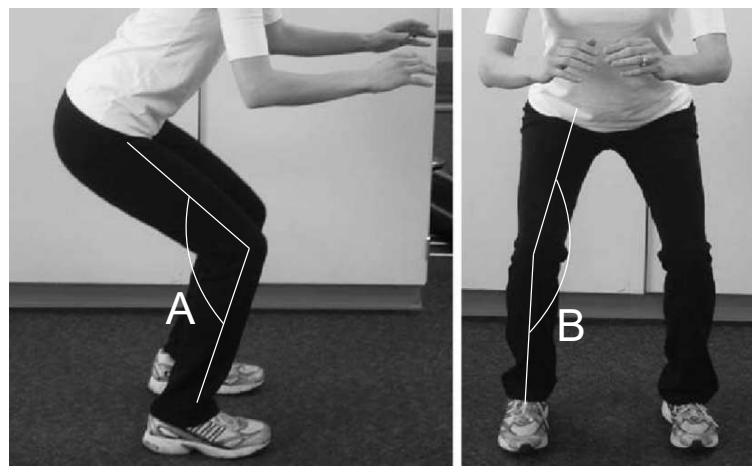


(a) Hauteur de progression du talon et de la pointe du pied lors de la marche, durant la phase de swing [30]



(b) Représentation de l'angle de progression du pied [25]

(c) Marche normale (A), marche avec une médialisation du genou (B) et marche avec une inclinaison du tronc (C) [31]



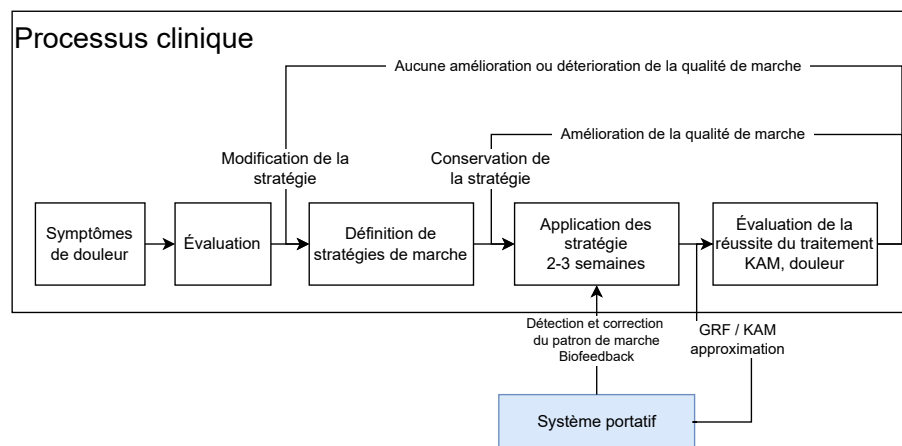
(d) Angle sagittal (A) et frontal (B) du genou [32]

**Figure 1.3** – Représentation de divers paramètres cinématiques associé à des stratégies d'amélioration de la marche

## 1.3 Applications cliniques et intégration de technologies portatives

### 1.3.1 Processus normal

Une personne qui débute une condition dégénérative comme l'arthrose peut consulter des cliniciens (physiothérapeute, kinésologue) lorsqu'une douleur est ressentie. L'intervention faite par le clinicien, représentée dans la case blanche de la figure 1.4, comprend généralement une première phase d'évaluation et d'adoption de stratégies correctives de la marche afin de prévenir la progression de la condition. Le patient applique ensuite les modifications dans sa marche de manière autonome pendant 2 à 3 semaines (variable selon la situation). Des consultations additionnelles après cette période permettent d'évaluer à nouveau la qualité de la marche et le niveau d'amélioration de la condition (posture, mobilité, douleur ...). De nouveaux correctifs peuvent être appliqués s'il n'y a pas d'amélioration du patron de marche. La rééducation à la marche peut donc prendre plusieurs itérations cliniques et de correction [15].



**Figure 1.4** – Processus de traitement de conditions de marche et intégration de technologies portatives.

### 1.3.2 Intégration de technologies portatives

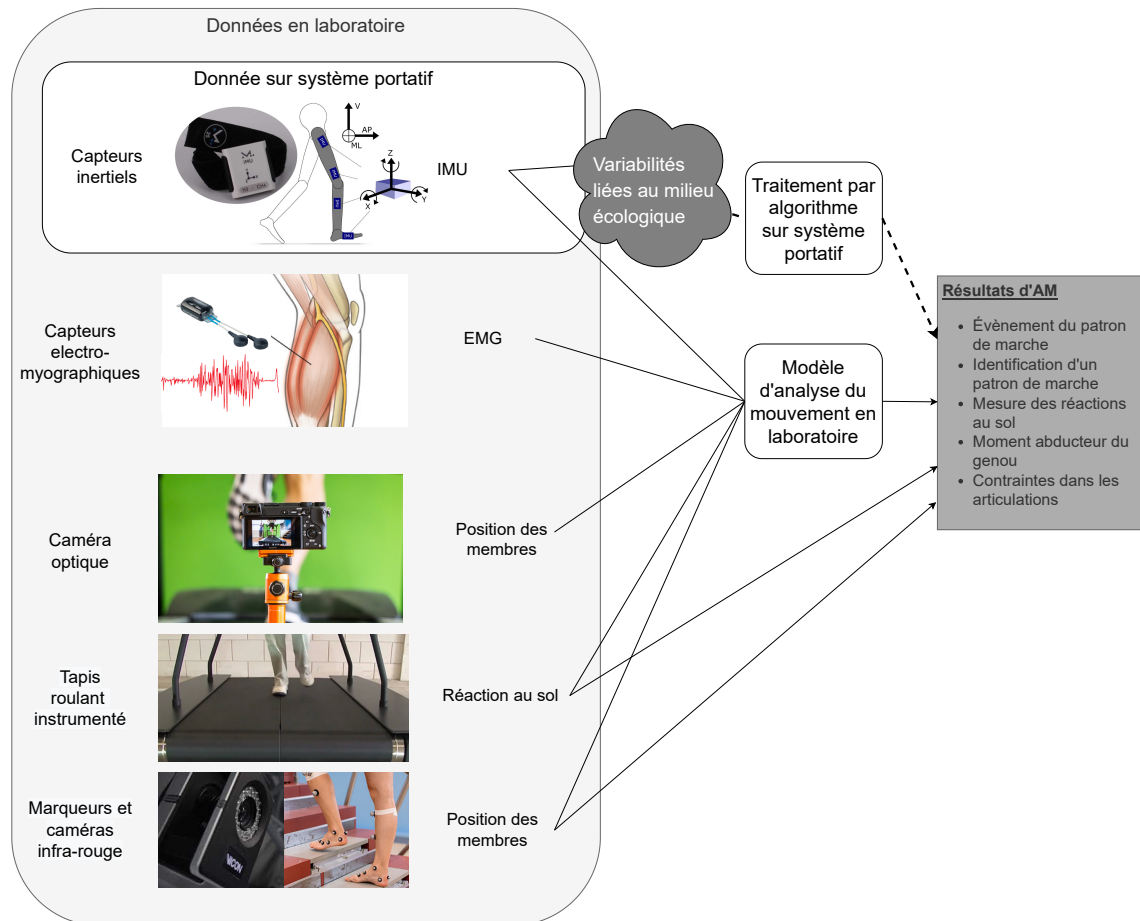
À l'extérieur des séances cliniques, il est difficile de s'assurer que les correctifs de la marche proposés par le clinicien sont bien appliqués par le patient. Les technologies portatives peuvent vérifier le degré d'adoption à l'extérieur des séances clinique.

Ainsi, les systèmes portatifs offrent un potentiel d'étendre l'analyse de la marche :

- En évaluant la qualité de la marche dans un milieu écologique en estimant des mesures comme la force au sol et le KAM associées à la progression de l'arthrose.
- En donnant un biofeedback pouvant aider le patient à corriger en temps réel le mouvement, sans la présence constante d'un expert clinique (reste à développer et à développer).

### 1.3.3 Défis des technologies portatives en analyse de la marche

L'utilisation des technologies portatives pour effectuer les différentes analyses biomécaniques reste un défi. L'analyse de la marche dans un contexte écologique comporte des contraintes et limitations incontournables illustré dans la figure figure 1.5.



**Figure 1.5** – Comparaison entre une situation en laboratoire et une en milieu écologique pour l'analyse de la marche. Le milieu écologique possède une quantité de données restreintes et est composé de plus de facteurs de variabilités

Les deux principales limitations sont énumérées ci-dessous :

1. **Accessibilité restreinte des appareils de mesure** : Le principal défi des technologies portatives est la variété restreinte des appareils de mesure par rapport à ce qui est mesurable normalement en laboratoire. Les systèmes inertiels ou les systèmes d'électromyogramme de surface sont les seuls parmi les appareils de mesures mentionnés dans la section 1.2.2 qui peuvent facilement s'incorporer dans des appareils portatifs. L'analyse du mouvement se limite donc à une nature des données beaucoup plus restreinte, soit l'activité électrique en surface des muscles via les EMG ou les accélérations et la vitesse angulaire via les capteurs inertiels.
2. **Augmentation des facteurs de variabilité dans les données** : Le processus d'analyse de la marche mentionné à la figure 1.5 se réduit à un environnement



contrôlé et souvent peu représentatif d'un environnement de la vie courante (écologique). Les systèmes portatifs permettent d'évaluer des variables biomécaniques (cadence, force de réaction au sol, cinématique de la jambe ou du pied) directement sur le patient pour suivre leur état de santé en continu. Toutefois, ces mesures sont beaucoup plus difficiles à établir en conditions écologiques en raison de la très grande variabilité du contexte de marche. L'état physique du patient, le type de sol et d'environnement peuvent influencer la marche du patient et par conséquent l'estimation de ces différentes variables biomécaniques de manière à réduire leur fiabilité ou leur interprétation.

Dans la figure 1.5, on fait l'hypothèse que les données disponibles dans un contexte écologique ne sont que les capteurs inertiels (accéléromètre, gyroscope et magnétomètre). On peut toutefois noter que dans certaines applications, on considère aussi d'autres types de capteurs pour des applications portatives comme les EMG ou encore des capteurs de pression (force sensitive resistor, FSR).

### 1.3.4 Le reservoir computing comme méthode de traitement des données des capteurs portatifs

---

Ce mémoire s'intéresse à une nouvelle méthode de traitement AI, le reservoir computing (RC). Le paradigme des algorithmes RC est une méthode black box versatile permettant des prédictions de nature variée. Un RC peut être entraîné sur une large variabilité de données temporels, comme les accélérations du pied, pour prédire une grande variabilité de prédiction. Dans ce projet, on démontre l'application de la méthode RC en analyse de la marche avec trois types de prédiction, comme représenté dans la figure 1.6 : la détection des événements de marche, la détection de patron de marche et la prédiction de signaux biomécanique des forces au sol. Chacune de ces tâches forme des sujets indépendants abordés dans un chapitre dédié dans ce mémoire, et ayant des utilités spécifiques en clinique, comme résumé dans la tableau 1.1. En pratique, l'intérêt de la méthode RC est aussi de permettre d'accomplir chacune de ces tâches en parallèle.

## 1.4 Données expérimentales utilisées

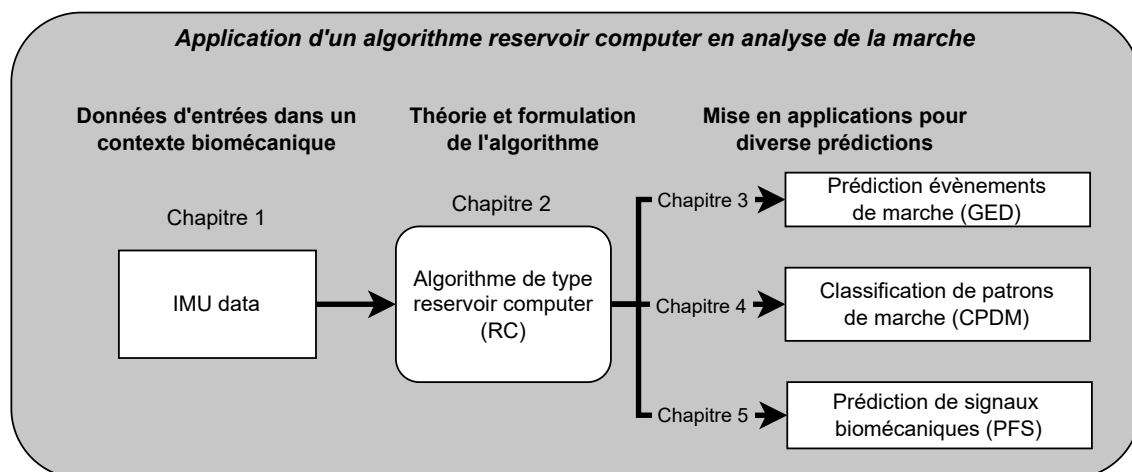
---

Cette section présente les quatre bases de données de marche utilisées dans ce mémoire. On présente aussi une caractérisation des signaux d'accélération (allure, amplitude, fréquence) dans la section 1.4.2 ainsi que la distribution de la position temporelle des événements de marche dans le cycle dans la section 1.4.3.

### 1.4.1 Bases de données

---

Dans le cadre de ce projet, nous avons développé plusieurs bases de données permettant d'entraîner le RC sur les différentes tâches de prédiction biomécanique. Cette section décrit les bases de données utilisées. Chaque base de données possède des données objectives d'entraînement, synchronisées avec les données d'accélération. Ces données représentent soit les événements de marche (GE), la mesure de la force de réaction au sol (GRF) ou une labélisation de patrons de marche (WS).



**Figure 1.6** – Représentation de la structure du mémoire sur l'application des algorithmes de reservoir computer (RC) pour l'analyse de la marche. Le chapitre 1 (section 1.4) introduit les données IMU utilisées en entrée de l'algorithme, le chapitre 2 introduit les éléments théoriques et la formulation de la méthode RC, puis trois applications sont développées dans les chapitres 3 à 5. Le chapitre 3 traitera de la tâche de détection des évènements de marche, le chapitre 4 aborde la tâche de classification de patrons de marche et le chapitre 5 traitera de la tâche de prédiction de signal.

Type de prédiction du RC	Information clinique extraite
Détection d'évènements de marche (GED)	<ul style="list-style-type: none"> <li>• Estimation de la cadence</li> <li>• Estimation de la proportion de temps en phase d'appui ou en phase de swing</li> <li>• Valeur intermédiaire exploitable pour les autres tâches</li> </ul>
La classification de patrons de marche (AD)	<ul style="list-style-type: none"> <li>• Détection d'une position adéquate</li> <li>• Correction lors d'une stratégie mal effectuée</li> </ul>
La prédiction de signaux biomécaniques indirectes	<ul style="list-style-type: none"> <li>• Démonstration avec la prédiction des forces au sol (PFS)</li> <li>• Permettrait aussi l'estimation de signaux indirectes mesurant la qualité de la fonction d'absorption de la marche (KAM) et contrainte articulaire</li> </ul>

**Table 1.1** – Énumération des différentes tâches de traitement par RC

Les données du CIRRI20 et CIRRI22 ont été acquises au laboratoire d'analyse du mouvement du CIRRI, à Québec. CIRRI20 est une base de données où deux participants ont marché sur un tapis roulant selon cinq vitesses différentes et cinq patrons de marche différents, utiliser pour la prédiction des évènements de marche (chapitre 3) et la classification des patrons de marche (chapitre 4). Pour CIRRI22, les participants étaient à la fois des participants sains (CIRRI22a) et des patients atteints d'arthrose du genou

(CIRRIS22b), utilisé pour la prédiction des forces au sol (chapitre 5).

La base de données de UDS a été développée à l'université de Sherbrooke spécifiquement pour évaluer la tâche de GED (chapitre 3).

Le MAREA est une base donnée accessible en ligne pour la communauté scientifique acquise et distribuée par l'université de Halmstad, en Suède, [38], utiliser aussi pour le GED.

La tableau 1.2 présent les caractéristiques des bases de données. Chacune se compose d'enregistrements d'accélération mesurés par un IMU sur le pied pour CIRRIS20 et UDS, différentes positions de la jambe pour CIRRIS22 et sur le pied, le bras et les hanches pour MAREA. Les caractéristiques dans le domaine temporel et fréquentiel de signaux d'accélération seront présentées dans la section 1.4.2.

Base de données	Type de données d'entraînement associées	Nombre de participants	Durée totale d'acquisition (min)	Conditions de marche	Condition des participants
CIRRIS20	GE, GRF, WS	2	112	TRI	Normal
CIRRIS22a	GE, GRF	27	162	TRI	Normal
CIRRIS22b	GE, GRF	18	100	TRI	Arthrose
UDS	GE	14	448	MI	Normal
MAREA	GE	19	362	MI et ME	Normal

GE : Gait event / évènement de marche

GRF : Ground reaction force / force de réaction au sol

WS : Classifié par stratégies de marche

TRI : Sur tapis roulant instrumenté

MI : Marche intérieure

ME : Marche extérieure

**Table 1.2** – Caractéristiques principales des bases de données.

Base de données	Âge des patients	Poids des participants (kg)	Taille des patients (cm)	Vitesse de marche (m/s)
CIRRIS 1	36/31	61/70	156/177	0.8 à 1.6
CIRRIS 2a	$36.7 \pm 4.2$	$73.9 \pm 16.7$	$172 \pm 8$	$1.25 \pm 0.12$
CIRRIS 2b	$65.3 \pm 7.3$	$79.3 \pm 13.9$	$165 \pm 12$	$0.75 \pm 0.23$
UDS	$29.1 \pm 13.0$	$65.2 \pm 13.0$	$169.7 \pm 9.0$	-
MAREA	$33.4 \pm 7$	$73.2 \pm 10.9$	$172.6 \pm 9.5$	-

**Table 1.3** – Caractéristiques des participants pour chaque base de données (moyenne)  $\pm$  (écart-type).

## 1.4.2 Caractérisation des signaux IMU

Les accélérations représentent les données d'entrée du RC. Dans chacune des bases de données, elles ont été enregistrées sur le pied avec un capteur IMU 3 axes dont les caractéristiques sont présentées dans le tableau 1.4. Les accélérations ont été ré-échantillonnées à une fréquence uniforme de 200 Hz.

L'allure générale des signaux d'accélération est périodique dans le cycle de marche d'une personne. L'allure des accélérations à chaque cycle est montrée dans la figure 1.7.

Base de données	Capteurs	fréquence d'acquisition (Hz)	plage d'amplitude (g)
CIRRIS	Physilog 4	200	$\pm 8$
UDS	Adafruit MMA8451	250	$\pm 2$
MAREA	Shimmer3	125	$\pm 8$

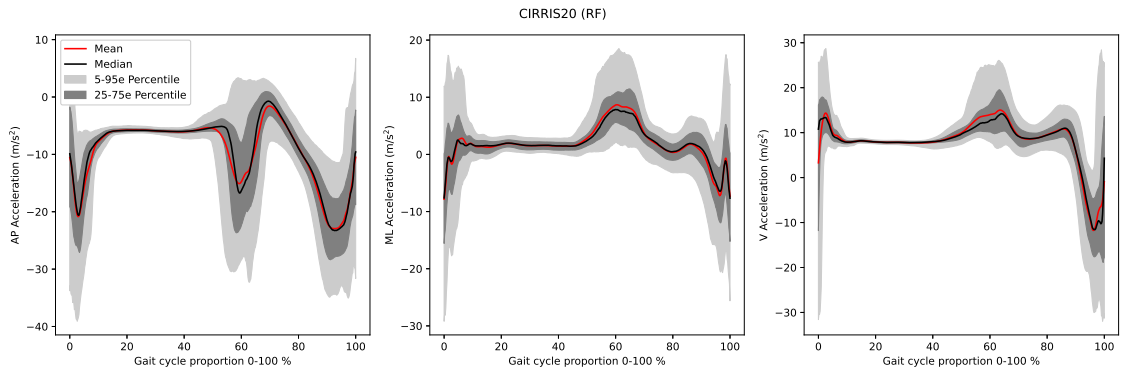
**Table 1.4** – Caractéristiques des capteurs IMU utilisés pour chaque base de données

L'amplitude des signaux dans chaque cycle possède une certaine variabilité qui est différentes en fonction de la portion du cycle de marche.

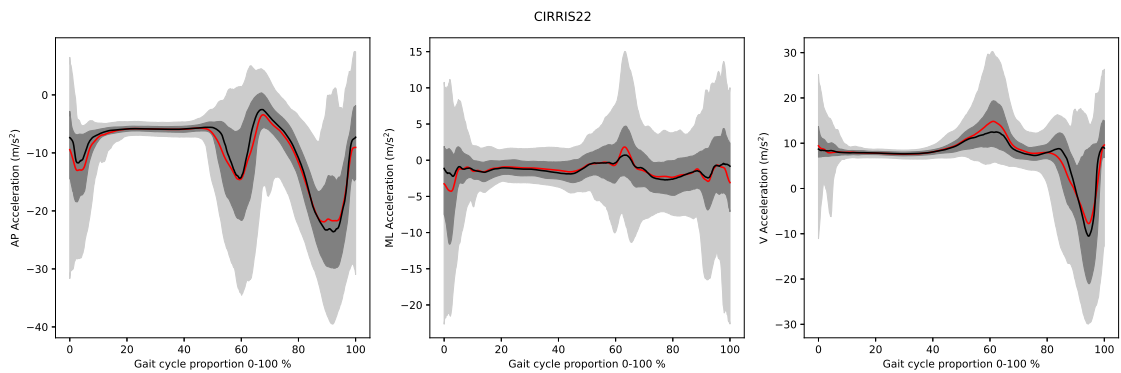
On voit dans la figure 1.7 que la variabilité est grande lors de la phase de swing (lorsque le pied est en mouvement, environ 50-100 % du cycle). Cette variabilité est de 5 et 30  $\frac{m}{s^2}$  de largeur d'intervalle entre le 5<sup>e</sup> et le 95<sup>e</sup> percentile sur l'ensemble des signaux. On voit aussi que la variabilité est faible durant la phase d'appui (lorsque le pied ne bouge pas, environ 10-50% du cycle), soit de l'ordre de 1-3  $\frac{m}{s^2}$  de largeur d'intervalle entre le 5<sup>e</sup> et le 95<sup>e</sup> percentile.

On peut donc interpréter que deux sources de variabilité existent dans les signaux d'accélération de ces bases de données. La différence de mouvement entre les participants occasionne une grande variabilité, alors que la différence d'orientation et de position des IMU sur le corps occasionne une plus petite variabilité.

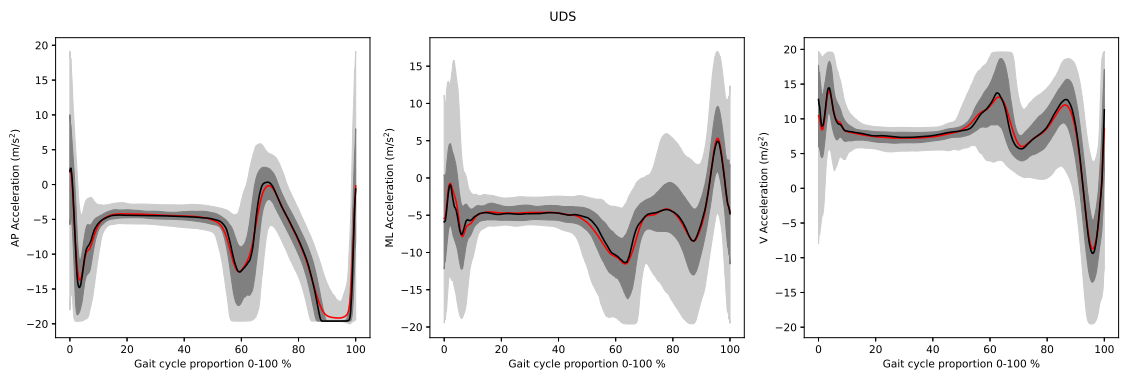
Le contenu fréquentiel des signaux d'accélération se situe entre 0 et 15 Hz principalement, comme on peut le voir dans figure 1.8, avec un premier pic d'accélération est de l'ordre de 0.9 Hz. et correspond à la cadence moyenne de marche des participants. Le second pic fréquentiel est proche du double du premier (environ 2Hz) et correspond à la fluctuation entre la phase d'appui et la phase de swing [39]. Le reste du contenu fréquentiel des signaux d'accélération se trouve sur des pics correspondant à des harmoniques de la fréquence fondamentale du cycle de marche (0.9 Hz). On peut clairement l'observer ces harmoniques pour la base de donnée UDS. On voit aussi que le contenu fréquentiel de UDS possède moins de contenu fréquentiel au-dessus de 10 Hz, il est probable que cela provienne des spécifications du capteur, dont le niveau de saturation était limité à  $\pm 2$  g, contrairement aux autres capteurs limités à  $\pm 8$  g.



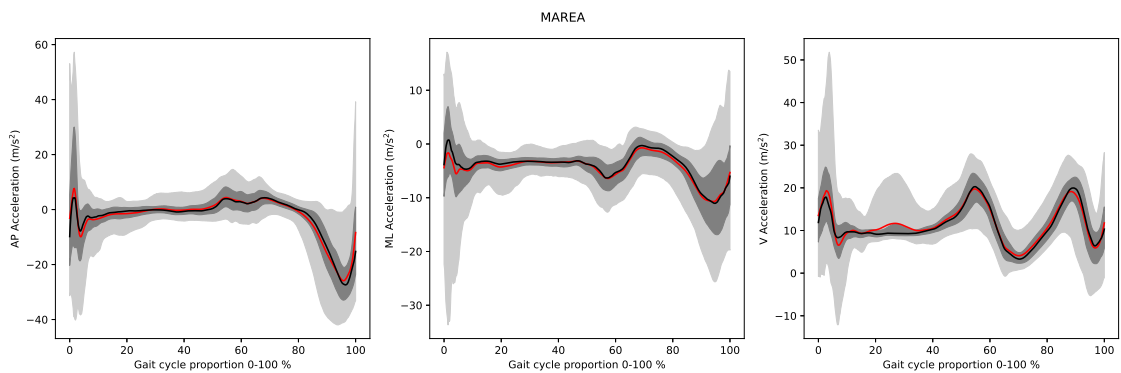
(a) CIRRS20 (Jeune adultes sains)



(b) CIRRS22 (Inclue jeunes adultes sains et adultes agés avec arthrose)



(c) UDS (Jeune adultes sains)



(d) MAREA (Jeune adultes sains)

Figure 1.7 – Allure des accélérations dans le domaine temporel sur un cycle de marche.

### 1.4.3 Caractérisation des évènements de marche

La figure 1.9 permet de visualiser à quels endroits se distribuent les évènements de marche dans un cycle. Les évènements de marche sont détectés à l'aide des mesures des forces au sol pour les bases CIRRIS et avec des FRS placés sous le pied pour les bases UDS et MAREA. Chaque évènement correspond à des minima, maxima ou à des fronts montant dans les signaux de force ou FSR.

- L'évènement du HS est défini comme l'évènement de référence, détecté par le premier front montant de force du pied au sol.
- L'évènement du HP correspond au premier pic de force maximal sur la force au sol pour le CIRRIS.
- L'évènement du FF correspond au minimum de la force au sol durant la phase d'appui.
- L'évènement du HO correspond au début du soulèvement du talon durant le début du pre-swing.
- L'évènement du TP correspond au deuxième maximum de la force verticale au sol durant la phase d'appui.
- L'évènement du TO correspond au front descendant de la force au sol totale ou de la force mesurée au bout de l'orteil avec les FSR dans les bases UDS et MAREA.

Chaque évènement est détecté selon un pré-traitement du signal expliqué dans l'annexe B. La figure 1.9 permet d'identifier la position relative de chaque évènement de marche par rapport au HS (positionné à 0% du cycle de marche). Pour chaque base de données et pour chaque évènement de marche, on montre un histogramme de l'occurrence des évènements de marche par rapport au HS au début de chaque cycle de marche.

On voit que les évènements HP, FF, HO, TP et TO apparaissent respectivement approximativement à 18%, 30%, 40%, 48% et 60 %. Le positionnement des évènements est moins variable (distribution plus étroite) pour les évènements du CIRRIS20 et CIRRI22 et pour le TP et TO de UDS. L'évènement du HO dans le UDS et du TO dans le MAREA est beaucoup plus variable (distribution très large). Selon les bases de données, la position dans le cycle de marche du TO varie beaucoup. Pour les bases de données du CIRRIS20, CIRRIS22 et UDS, on voit que le TO (incluant uniquement des données de marche) est bien autour de 60 %, similairement au modèle de Perry. Pour des données dans la base de MAREA, incluant de la marche et de la course, le TO peut survenir beaucoup plus rapidement, jusqu'à 25 % du cycle de marche.

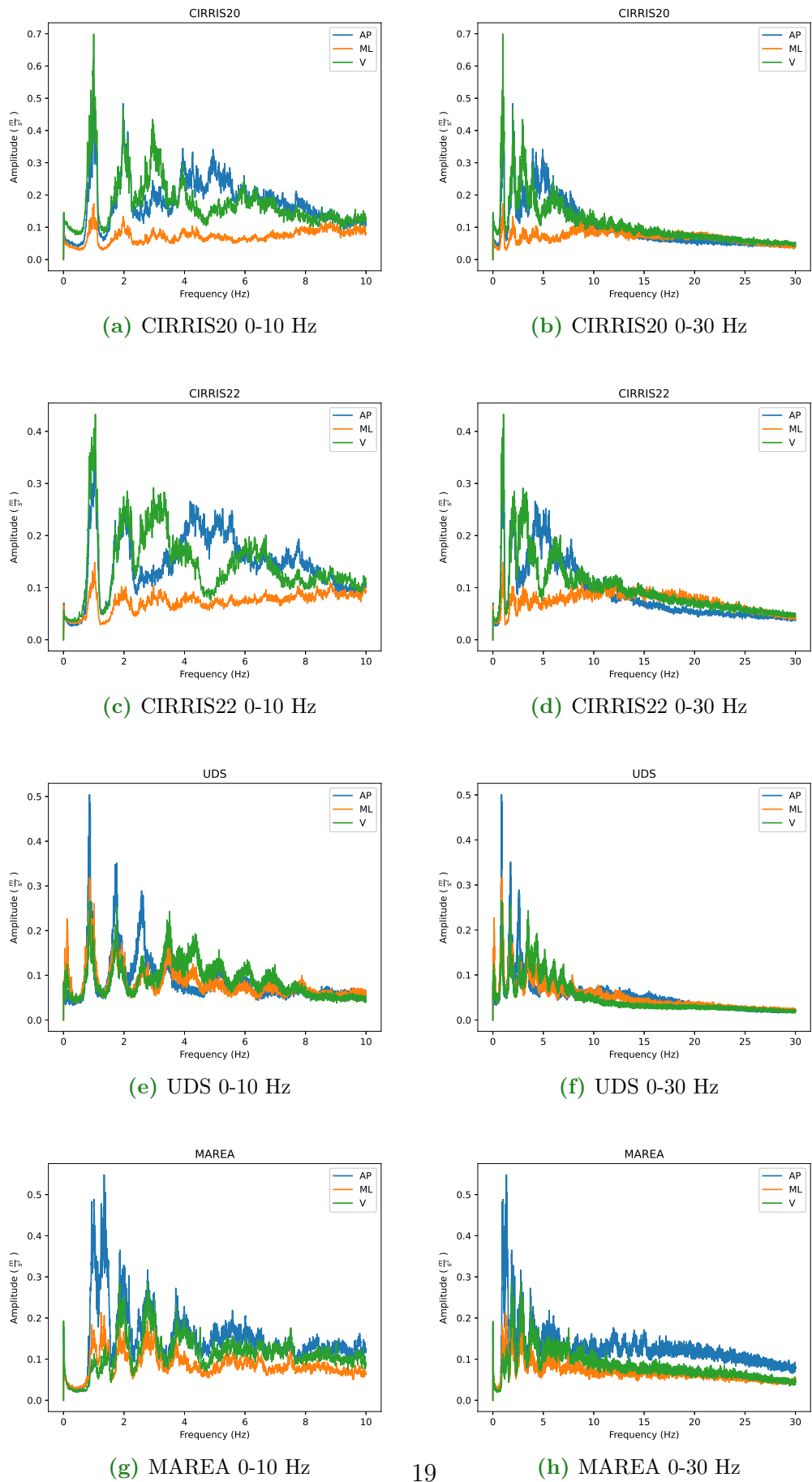
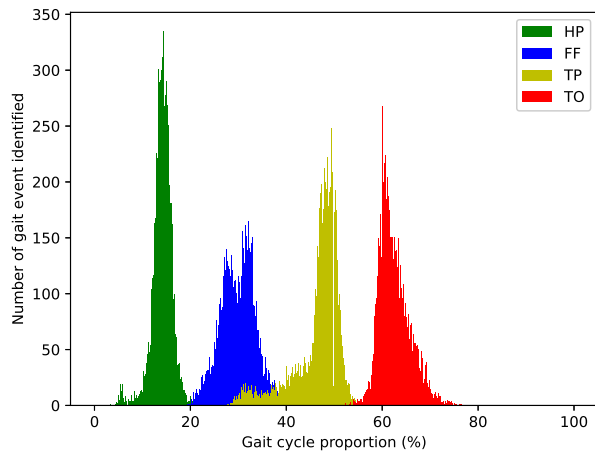
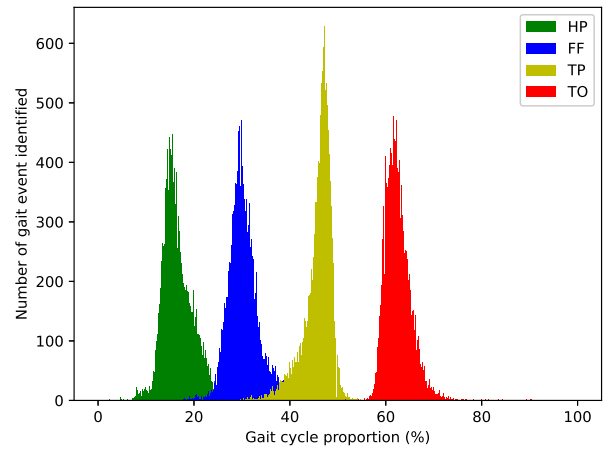


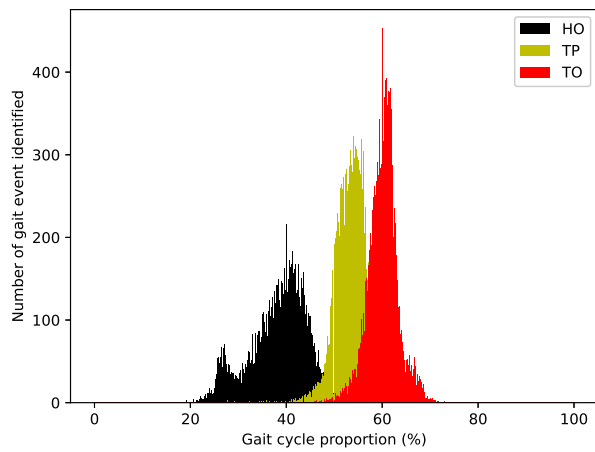
Figure 1.8 – Contenu fréquentiel des signaux d'accélération dans les bases de données.



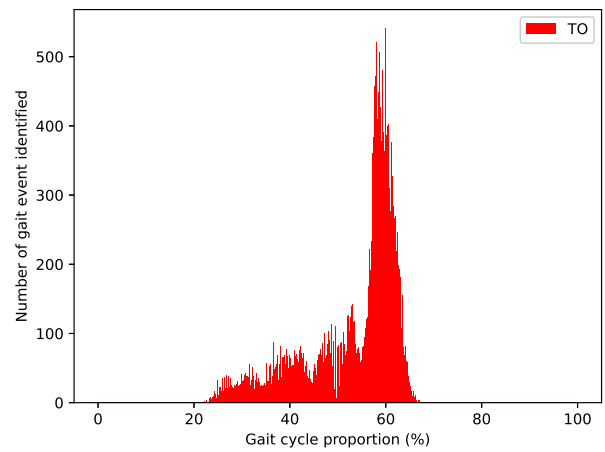
(a) CIRRIS2020



(b) CIRRIS2022



(c) UDS



(d) MAREA

Figure 1.9 – Distribution des occurrences des évènements de marche.



# Chapitre 2

## Introduction au reservoir computing

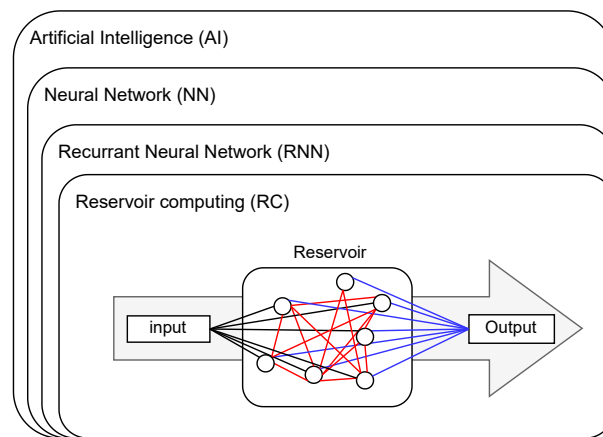
### Éléments clé

Ce chapitre présente le concept du reservoir computing (RC), définit comme une sous catégorie d'algorithme de réseaux de neurones, puis introduit la technologie du RC-MEMS conçu dans le groupe de recherche de Julien Sylvestre.

- Les réseaux de neurones artificiels (ANN) peuvent être considérés comme des classificateurs de données qui transforment l'information de manière non linéaire et dans une dimension supérieure à celle des données initiales.
- Les RNN sont une catégorie d'ANN adaptée pour traiter des données séquentielles ou temporelles comme les accélérations de la marche.
- Le nombre très élevé de connexions temporelles rend l'utilisation de la rétropropagation du gradient, méthode conventionnellement utilisé pour l'entraînement des ANN, coûteuse en temps de calcul et difficilement exploitable dans le cas des RNN.
- Le RC est une forme de RNN dont les connexions dans les couches cachées sont fixes durant l'entraînement. Cela permet d'éviter l'utilisation des méthodes d'entraînements par rétropropagation du gradient.
- Un algorithme de RC numérique largement répandu dans la littérature est le Echo State Network (ESN). Il est déjà utilisé dans de nombreuses applications de traitement de signal de classification et de génération de signal et de commande.
- Le concept du RC est applicable dans des systèmes physiques. Le principe du RC-MEMS est de générer la fonction non linéaire du neurone en utilisant la réponse vibratoire d'une micro poutre bi-encastree.
- La version neuro-accéléromètre du RC-MEMS est un dispositif couplé à une masse inertielle en suspension sur des ressorts. Cela le rend sensible aux accélérations externes unidimensionnelles.
- Les RC physiques de type RC-MEMS ont démontré une performance de classification sur des tâches de la fonction de parité et de la reconnaissance vocale. Il s'avère donc pertinent d'évaluer la performance d'utilisation d'un RC pour résoudre des tâches de biomécanique.
- Il est nécessaire de respecter les contraintes de l'utilisation d'une seule accélération en entrée, de limiter la taille du réseau à 100 noeuds et de conserver une fréquence de 200 Hz au maximum afin que la démonstration avec le ESN soit compatible avec le neuro-accéléromètre RC-MEMS.

## 2.1 Mise en contexte sur les NN, RNN et RC

Le reservoir computing (RC) est une branche spécifique de l'AI, faisant partie de la catégorie des réseaux de neurones, plus spécifiquement des réseaux de neurones récurrents, comme illustré dans la figure 2.1. Le RC permet de générer des prédictions basées sur des signaux temporels comme les accélérations biomécaniques. Dans ce chapitre, on introduit le concept des RC et on le situe dans le contexte de l'AI. On explique le fonctionnement général d'application d'un RC, puis on explique le fonctionnement de deux types d'implémentations de RC, l'un numérique et l'autre physique.



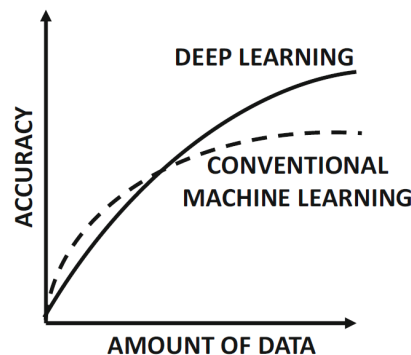
**Figure 2.1** – Situation du concept de reservoir computing (RC) en AI.

Dans cette section, trois concepts sont introduits : les réseaux de neurones (NN), les réseaux de neurones récurrents (RNN) et le reservoir computing (RC). Ces trois concepts sont imbriqués, car le RNN est une sous-branche des NN, spécialisés dans le traitement de données temporelles, alors que le RC est une sous-branche des RNN, dont la structure permet de faciliter en pratique la phase d'apprentissage d'un RNN.

### 2.1.1 Deep learning et réseaux de neurones (NN)

Le concept de calcul par réseaux de neurones (NN) est une branche de l'intelligence artificielle qui existe depuis les années 1940 [40], et qui émerge de l'étude de la physiologie du cerveau humain. Notre compréhension du traitement biologique de l'information permet d'identifier le fonctionnement de l'unité élémentaire du cerveau : le neurone. Celui-ci est représenté par une fonction mathématique binaire prenant un état actif ou inactif selon les stimuli reçus par les neurones adjacents. On comprend que le cerveau contient un réseau d'une grande quantité ( $\approx 10^{11}$  [40]) de neurones interconnectés, qui permet le traitement de l'information entre les systèmes perceptifs et les systèmes moteurs de l'humain. Un stimulus extérieur (ex : sensation de brûlure) transmis par des signaux électriques au cerveau influence l'activité neuronale (les états des neurones) et permet de modifier d'autres parties du réseau associées aux signaux moteurs (ex : activation des muscles de la main). En s'inspirant de ce processus dans le cerveau, le domaine de l'intelligence artificielle a fait émerger des algorithmes de traitement de données basés sur une architecture en réseaux de neurones.

La première démonstration purement mécanique d'un réseau de neurones fut mise en oeuvre dès 1950 par des étudiants de l'université d'Harvard [40]. Ce concept était toutefois prématuré par rapport aux outils technologiques de l'époque, laissant peu d'applications apparaître avant la fin du 20e siècle. Au début des années 2000, l'apparition de bases de données suffisamment grandes, des capacités substantielles de stockage d'information et de puissance de calculs, et le développement des méthodes d'optimisation par rétropropagation [2], ont permis de démontrer les performances compétitives des NN face aux autres méthodes de traitement exploitées en apprentissage automatique et basées sur des connaissances d'experts (ex : logique de 1<sup>er</sup> ordre, réseaux bayésiens [40]). Comme illustré dans la figure 2.2, la performance des réseaux de neurones se distingue par la nature et la quantité de données disponibles pour l'entraînement par opposition aux méthodes traditionnelles centrées sur la complexité de l'algorithme de traitement et de la nécessité d'experts dans le domaine d'application.



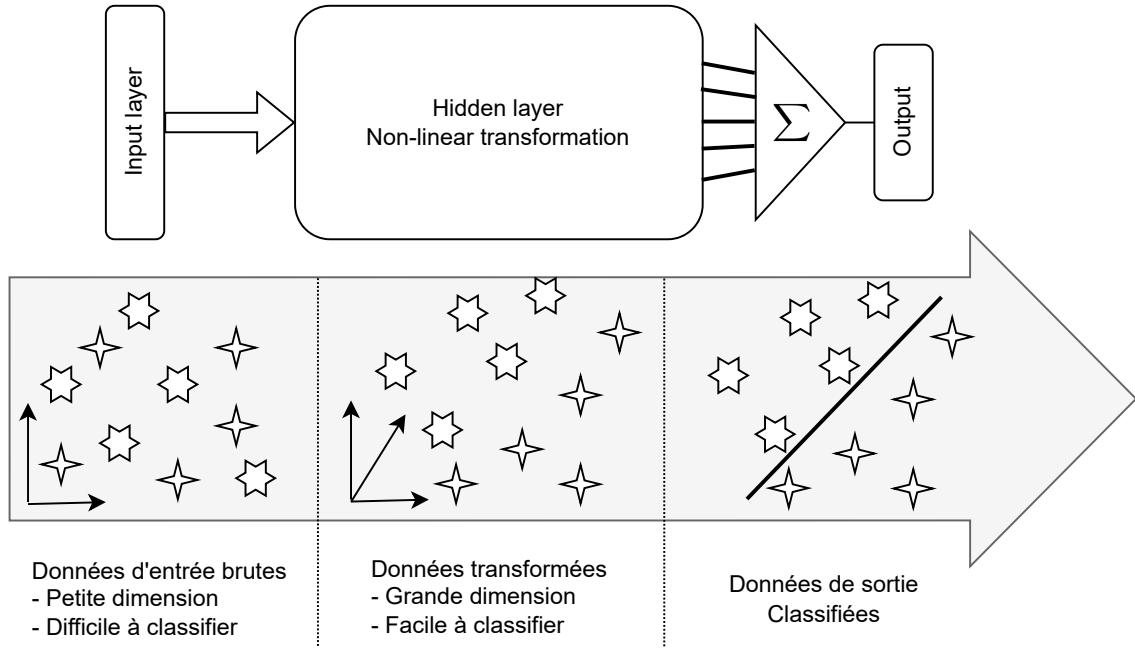
**Figure 2.2** – Comparaison de la performance du DL et des autres méthodes de Machine Learning en fonction de la disponibilité des données d'apprentissage. Le DL est compétitif lorsque la quantité de données disponibles est importante. [2]

### 2.1.1.1 Concept général des NN

De nombreux problèmes en ingénierie nécessitent de classifier des données difficilement séparables linéairement. Le but général des NN est d'appliquer une transformation non linéaire à des données afin de permettre une séparation d'information non triviale [2]. Une représentation simple du type d'analyse que l'on peut faire avec les NN est celle de la figure 2.3. Les données de base ne sont pas séparables dans l'espace 2D. Toutefois, la couche cachée applique une transformation non linéaire et une augmentation de la dimension dans les données. Cela permet de faire une séparation linéaire par la suite à la couche de sortie.

### 2.1.1.2 Opérateur élémentaire d'un NN : le perceptron

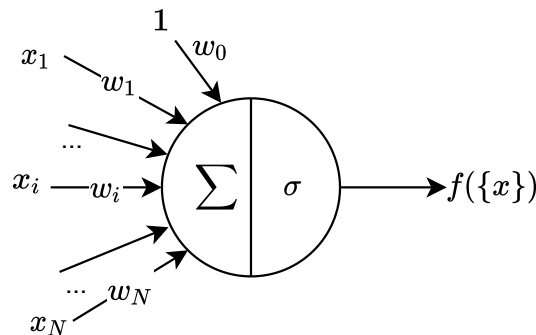
Pour appliquer cette transformation des données, un NN est basé sur un opérateur élémentaire non linéaire appelé le perceptron. Cet opérateur est le calcul de base utilisé dans un NN, il peut se définir comme



**Figure 2.3** – Principe général d'utilisation du NN : transformation non linéaire et augmentation de la dimension des données pour faciliter la séparation de l'information. [2]

$$f(\mathbf{x}) = \sigma\left(w_0 + \sum_{i=1}^N x_i w_i\right), \quad (2.1)$$

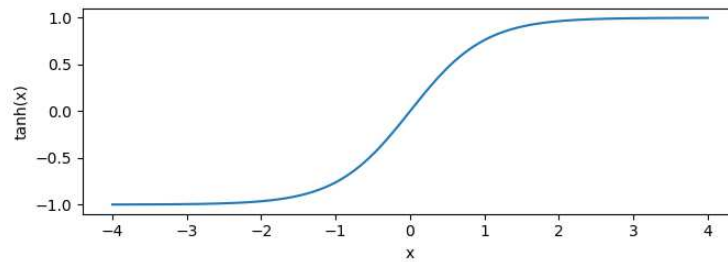
dont les variables d'entrées sont notées  $\mathbf{x} = [x_1, x_2, \dots, x_N]$  et dont les paramètres, notés  $\mathbf{w}$ , correspondent à des facteurs de pondération des entrées, aussi nommés poids synaptiques [41]. La fonction  $f$  est le résultat d'une somme pondérée des  $N$  entrées  $x_i$ , des poids  $w_i$  et d'un biais  $w_0$ , passée dans une sigmoïde non linéaire appelée la fonction d'activation  $\sigma$ . Le perceptron est représenté tel que montré dans la figure 2.4.



**Figure 2.4** – Représentation d'un perceptron

La fonction  $\sigma(x) = \tanh(x)$ , montrée sur la figure 2.5, est l'une des fonctions très utilisées en pratique [41]. D'autres fonction d'activation sont toutefois aussi utilisées, comme la Sigmoid, la Sign, la ReLU ou la Hard tanh [2]. Comme nous le mentionnons dans la section 2.1.1.5, les fonctions d'activation se doivent d'être continues, différentiables et afficher

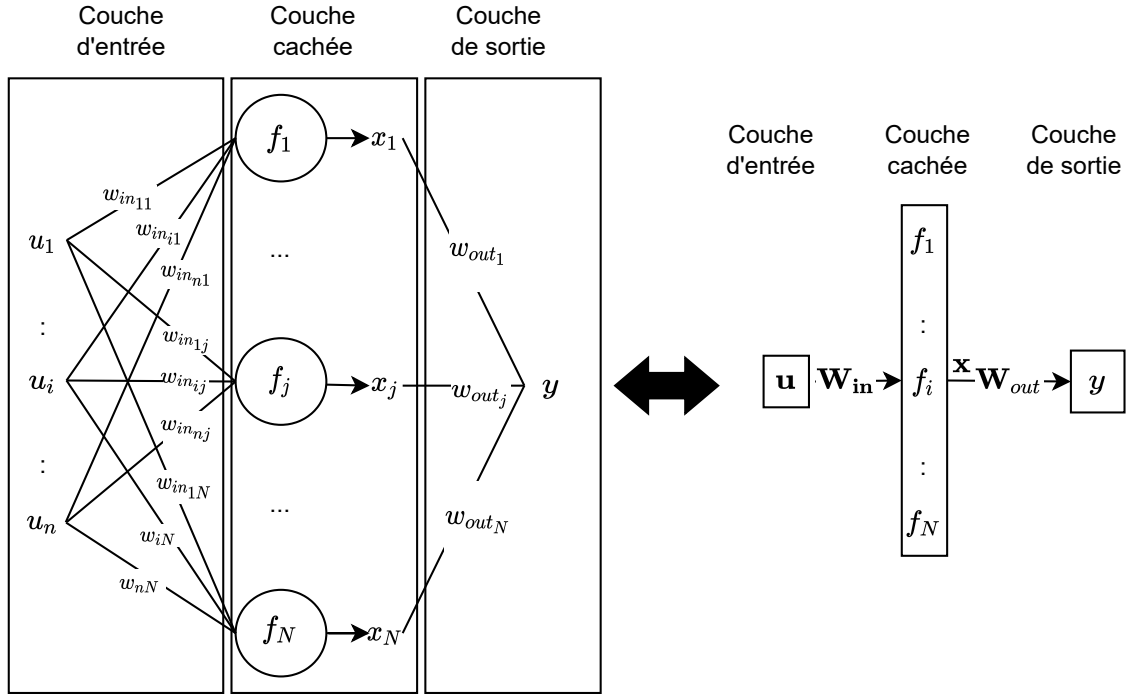
une non-linéarité pour être entraînées par la méthode de rétropropagation couramment utilisée avec les ANN.



**Figure 2.5** – Fonction d'activation  $\sigma(x) = \tanh(x)$  couramment utilisée pour un ANN.

### 2.1.1.3 Construction d'un NN avec plusieurs perceptrons

Les réseaux de neurones représentent un regroupement de plusieurs perceptrons, comme illustré dans la figure 2.6. Les entrées  $\mathbf{u} = \{1, u_1, \dots, u_i, \dots, u_{N_{in}}\}$  du NN représentent les données externes mesurées, composées de  $N_{in}$  valeurs numériques. Les opérateurs  $\{f_1, \dots, f_j, \dots, f_N\}$  représentent les différents perceptrons du NN. Chaque perceptron prend en argument les  $N_{in}$  données d'entrées. Le poids de pondération de l'entrée  $u_i$  dans le perceptron  $f_j$  est défini comme  $w_{inij}$ ,  $y$  représente la sortie du NN, aussi appelé la prédiction, qui peut être composée de plusieurs dimensions, notées alors  $\mathbf{y} = \{y_1, \dots, y_i, \dots, y_{N_{out}}\}$ .  $N_{out}$  représente le nombre de valeurs numériques dans la prédiction  $\mathbf{y}$ . Les valeurs de sortie des perceptrons  $x_j = f_j(\{u\})$  sont appelées comme les états de sortie du NN,  $\mathbf{x} = \{x_1, \dots, x_j, \dots, x_N\}$ .



**Figure 2.6** – Représentation développée (à gauche) et compacte (à droite) d'un NN, composé de  $N_{in}$  entrées et d'une couche cachée de  $N$  perceptrons.

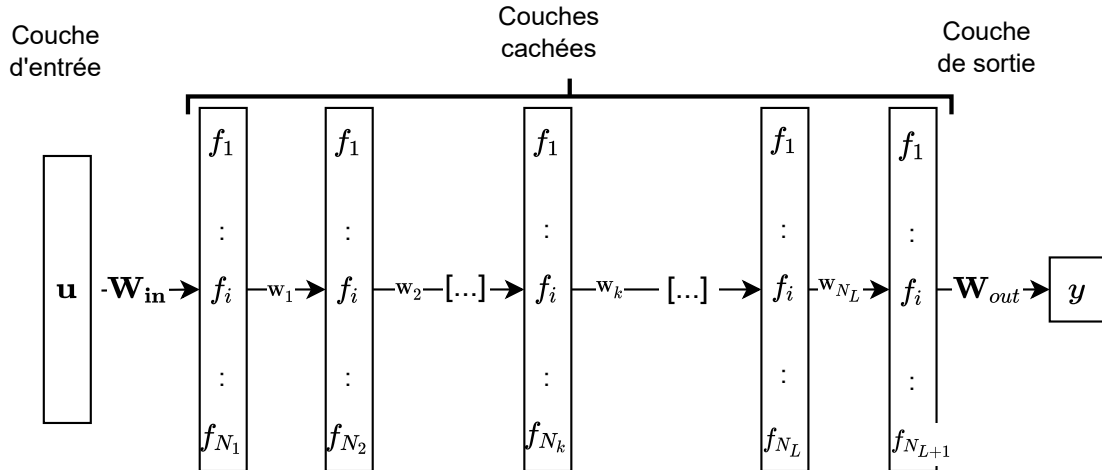
Les matrices  $\mathbf{W}_{in}$  et  $\mathbf{W}_{out}$  contiennent l'ensemble des poids de connexions, respectivement de la couche d'entrée vers la couche cachée et de la couche cachée vers la couche de sortie. Par exemple,

$$\mathbf{W}_{in} = \begin{bmatrix} w_{in01} & w_{in11} & w_{in21} & \dots & w_{in(N_{in}-1)1} & w_{inN_{in}1} \\ w_{in02} & w_{in12} & w_{in22} & \dots & w_{in(N_{in}-1)2} & w_{inN_{in}2} \\ \vdots & \vdots & \vdots & \dots & \vdots & \vdots \\ w_{in0(N-1)} & w_{in1(N-1)} & w_{in2(N-1)} & \dots & w_{in(N_{in}-1)(N-1)} & w_{inN_{in}(N-1)} \\ w_{in0N} & w_{in1N} & w_{in2N} & \dots & w_{in(N_{in}-1)N} & w_{inN_{in}N} \end{bmatrix}. \quad (2.2)$$

La matrice  $\mathbf{W}_{in}$  est de taille  $N_{in} \times N$  et la matrice  $\mathbf{W}_{out}$  est de taille  $N \times N_{out}$ . On note donc les opérations d'un NN avec une couche cachée comme

$$y = \mathbf{W}_{out}\mathbf{x} = \mathbf{W}_{out}f(\mathbf{W}_{in}\mathbf{u}). \quad (2.3)$$

L'utilisation d'un nombre élevé de perceptrons permet de faire des séparations complexes dans les données. En général, la complexité de cette séparation est proportionnelle à la quantité de perceptrons contenue dans un NN. Théoriquement, un NN possédant suffisamment de neurones pourrait aller jusqu'à estimer n'importe quelle fonction [2].



**Figure 2.7** – Représentation d'un NN à plusieurs couches cachées (DNN), composé de  $N_{L+1}$  couches cachées, composées de  $N_L$  matrices de connections des couches cachées ( $\mathbf{W}_k$ ) plus deux matrices de connections pour la couche d'entrée ( $\mathbf{W}_{in}$ ) et la couche de sortie ( $\mathbf{W}_{out}$ ).

Les NN profonds (DNN) regroupent plusieurs étapes de calculs en cascade de plusieurs neurones. Cela revient à exécuter le calcul de l'équation (2.1) à plusieurs reprises en utilisant les valeurs de sortie de l'étape de calcul précédentes comme suit,

$$y = \mathbf{W}_{out} f(\mathbf{W}_{N_L} f(\dots f(\mathbf{W}_2 f(\mathbf{W}_1 f(\mathbf{W}_{in} \mathbf{u}))))). \quad (2.4)$$

Chaque étape de calcul d'un DNN est appelée une couche du NN. Sa structure est montrée dans la figure 2.7. Pour chaque couche cachée, une matrice  $\mathbf{W}_k$  de taille  $N_k \times N_{k+1}$  définit les poids de pondérations entre les valeurs de sorties des perceptrons de la couche  $k$  vers les perceptrons de la couche  $k+1$ . En général, les DNN possèdent un nombre décroissant de noeuds vers les couches de sortie afin d'améliorer la stabilité de ses prédictions [2]. Chaque couche ne considère comme valeurs d'entrée que les valeurs obtenues dans la couche antérieure. On distingue trois types de couches :

1. La **couche d'entrée** du réseau de neurones représente les données d'entrées initiales du NN, noté par  $\mathbf{u}$ .
2. Les couches suivantes sont nommées les **couches cachées** composées de  $N_k$  perceptrons, de valeur pouvant être différente pour chaque couche. La première couche cachée prend en entrée les valeurs de  $\mathbf{u}$ , puis les couches ultérieures prennent en entrée les valeurs de sorties des perceptrons de la couche précédente.
3. La **couche de sortie** prend en argument les valeurs de sortie de la dernière couche cachée, puis en fait une combinaison linéaire afin de fournir une prédiction représentée par la variable  $\mathbf{y}$ . Dans certains cas, une non-linéarité  $\sigma(x)$  peut aussi être appliquée sur la couche de sortie, surtout dans le cas où le NN est appliqué pour faire une classification [2].

Les DNN sont principalement utilisés dans des applications de données non temporelles, comme la reconnaissance d'image. Dans le cadre de ce mémoire, focalisant sur l'utilisation des réseaux de neurones récurrents, les réseaux seront composés d'une seule

couche cachée, comme représentée dans la figure 2.6. Toutefois, le concept de DNN est introduit ici afin de faire un parallèle avec le concept de RNN et d'illustrer une problématique liée à l'entraînement par la méthode de rétropropagation introduite dans la section 2.1.1.5.

#### 2.1.1.4 Structure des NN selon le type de données à traiter

Différentes structures de NN ont été formulées pour s'adapter à des données et des applications de différentes natures. Les réseaux à propagation avant (FNN) sont des modèles utilisés pour traiter des données formées de vecteurs 1D indépendants [2, 42]. Nous représentons cette structure dans les figures 2.6 (simple couche) et 2.7 (multicouches). Les réseaux de neurones convolutifs (CNN) sont utilisés pour le traitement de données matricielles comme des images. Ils utilisent la convolution 2D afin de transformer une image en un vecteur numérique 1D, traité ensuite avec des réseaux FNN [2, 42]. Les réseaux de neurones récurrents (RNN) sont des réseaux spécialisés pour des données séquentielles, comme du texte ou des signaux temporels. Une introduction plus générale sur le concept de RNN sera faite dans la section 2.1.2. Le concept du RC, une branche émergente des RNN, est le cœur du sujet de ce mémoire et sera abordé dans la section 2.1.3.

#### 2.1.1.5 Méthode d'entraînement d'un NN par rétropropagation

L'entraînement d'un NN (à une ou plusieurs couches) consiste à ajuster des poids  $w_{k,ij}$  de manière à ce que la sortie  $y$  obtenue soit la meilleure prédiction possible. Le concept d'entraînement est l'approche où la meilleure prédiction possible est définie pour minimiser une erreur entre la prédiction  $y$  du NN et une valeur objective notée  $y^t$ . La principale méthode d'ajustement des poids pour minimiser cette erreur dans les NN est la rétropropagation du gradient, popularisée en 1986 [43]. Cette méthode permet de calculer un gradient de l'erreur  $e = y - y^t$  sur chaque poids  $w_{ij}$  du modèle en utilisant la règle de dérivation en chaîne,

$$\frac{\partial e}{\partial w_{ij}} = \frac{\partial e}{\partial x_j} \frac{\partial x_j}{\partial w_{ij}}. \quad (2.5)$$

où  $x_j$  représentant l'état de sortie du neurone  $j$  du NN. Si le NN se compose de plusieurs couches cachées, le calcul du gradient pour un poids  $w_{k,ij}$  connectant le  $i^{\text{eme}}$  perceptron d'une couche cachée  $k$  vers le  $j^{\text{eme}}$  perceptron d'une couche cachée  $k + 1$  se définit par,

$$\frac{\partial e}{\partial w_{k,ij}} = \frac{\partial e}{\partial x_{k+1,j}} \frac{\partial x_{k+1,j}}{\partial w_{k,ij}}, \quad (2.6)$$

où  $x_{k+1,j}$  est l'état de sorti du perceptron  $j$  de la couche  $k + 1$  et  $f_{k+1,j}$  représente la fonction du perceptron  $j$  de la couche  $k + 1$  calculée avec les sorties  $\mathbf{x}_k$  de tous les perceptrons de la couche  $k$ , et donc on définit la sortie du perceptron dans la couche suivante du réseau comme  $f_{k+1,j}(\mathbf{x}_k) = x_{k+1,j}$ . Le gradient de l'erreur sur la sortie du perceptron  $x_{k+1,j}$  peut ainsi se développer avec la règle de dérivation en chaîne comme



$$\begin{aligned} \frac{\partial e}{\partial x_{k+1_j}} &= \frac{\partial e}{\partial x_{k+2_l}} \frac{\partial x_{k+2_l}}{\partial x_{k+1_j}} = \frac{\partial e}{\partial x_{out_l}} \frac{\partial x_{out_l}}{\partial x_{N_L l}} \frac{\partial x_{N_L l}}{\partial x_{N_L-1_l}} \cdots \frac{\partial x_{k+3_l}}{\partial x_{k+2_l}} \frac{\partial x_{k+2_l}}{\partial x_{k+1_j}} \\ &= \frac{\partial e}{\partial y} \frac{\partial y}{\partial x_{out_j}} \prod_{m=k+1}^{N_L} \frac{\partial x_{m+1_l}}{\partial x_{m_l}}. \end{aligned} \quad (2.7)$$

Cela revient à exprimer le gradient de l'erreur sur le poids  $w_{k,ij}$  comme suit,

$$\frac{\partial e}{\partial w_{k,ij}} = \frac{\partial x_{k+1_j}}{\partial w_{k,ij}} \left[ \frac{\partial y}{\partial x_{out_j}} \prod_{m=k+1}^{N_L} \frac{\partial x_{m+1_l}}{\partial x_{m_l}} \right] \frac{\partial e}{\partial y}. \quad (2.8)$$

Le calcul du gradient sous forme de dérivation en chaine est montré ici en fonction d'un seul chemin de perceptrons reliant le poids  $w_{k,ij}$  et la sortie  $y$ . Ce chemin est noté par une liste d'indices  $l_k$  reliant les perceptrons de chaque couche. Ici, pour la couche  $k$ ,  $l_k = j$ , puis ensuite pour la couche  $k + 1$ , on passe par le perceptron  $l_k + 1$ , pouvant avoir une valeur arbitraire entre 1 et  $N_{k+1}$ , puis  $l_{k+2}$  qui peut avoir une valeur entre 1 et  $N_{k+1}$  pour la couche  $k + 2$ , etc. La liste représentant les indices choisis sur chaque couche peut se définir comme un vecteur d'indice comme suit,  $[j, l_{k+2}, \dots, l_{N_L}, l_{out}]$ . On peut calculer le gradient total en considérant tout les vecteurs d'indices (chemins) possibles reliant le perceptrons au poids  $w_{k,ij}$  avec la sortie  $y$  [ $j \rightarrow (l)_{out}$ ], regroupé dans l'ensemble  $\mathcal{P}$  [2],

$$\frac{\partial e}{\partial w_{k,ij}} = \frac{\partial x_{k+1_j}}{\partial w_{k,ij}} \left( \frac{\partial y}{\partial e} \left[ \sum_{\mathcal{P}} \frac{\partial y}{\partial x_{out_l}} \prod_{m=k+1}^{N_L} \frac{\partial x_{m+1,l_m}}{\partial x_{m,l_m}} \right] \right). \quad (2.9)$$

Ce calcul correspond au calcul du gradient total de l'erreur d'un poids  $w_{k,ij}$  quelconque dans un NN à plusieurs couches cachées [2]. Ce développement permet d'observer des propriétés du calcul de BP imposant trois contraintes importantes de son utilisation dans le cas des RNN et seront reprises ensuite dans la section 2.1.2 portant sur les RNN afin d'expliquer la motivation du concept de RC :

1. **Dérivabilité des perceptrons** : Le calcul de la fonction d'activation dans chaque perceptron se doit d'être différentiable afin d'établir les différentes valeurs de gradient entre les perceptrons, soit

$$\frac{\partial x_{m+1_l}}{\partial x_{m_l}} = \partial[f'(x_{m_l})]/\partial x_{m_l} = \partial[\sigma(w_{m0} + \sum_{i=0}^{N_k} x_{m_i} w_i)]/\partial x_{m_l} = \sigma'(w_{m0} + \sum_{i=0}^{N_k} x_{m_i} w_i) w_{m_l}.$$

La dérivée de la fonction d'activation  $\sigma'$  doit donc être connue.

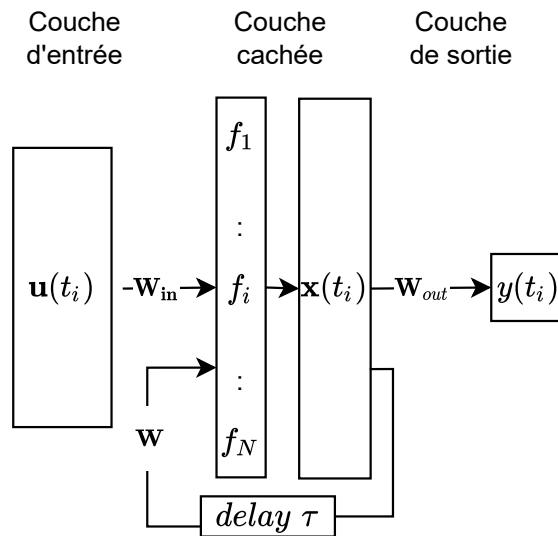
2. **Complexité** : Le nombre théorique de calculs dans BP est exponentiel en fonction du nombre de couches cachées. En effet, le nombre de chemins possibles entre le poids de la première couche  $w_{1,ij}$  et  $y$  dans P dans un NN de  $N_{L+1}$  couches cachées, chacune composée de  $N$  perceptrons, est de  $N^{N_L}$ . Chaque chemin demande de faire le produit de  $N_L$  gradients. Le nombre de calculs bruts demandé pour faire le calcul de BP sur  $w_{1,ij}$  est de  $N_L(N+1)^{N_L}$  multiplications et de  $N^{N_L}$  additions. Cela fait un total théorique de  $\sum_{s=1}^{N_L} s(N+1)^s$  multiplications et de  $\sum_{s=1}^{N_L} N^s$  additions pour l'ensemble des poids du NN. En pratique, ce nombre de calculs est moindre grâce aux méthodes de programmation dynamique développé afin de minimiser le nombre

de calculs à effectuer [2]. La complexité du calcul reste toutefois exponentielle en fonction du nombre de couches cachées  $N_{L+1}$ .

3. **Stabilité** : La stabilité de la méthode BP sur un DNN est inversement proportionnelle au nombre de couches cachées. Le gradient de la méthode BP est un calcul en chaîne qui se fait de manière itérative sur chacune des couches cachées du NN. Elle implique une multiplication en chaîne des gradients des couches antérieures. Si le réseau possède beaucoup de couches cachées, cela peut faire tendre le gradient de manière exponentielle vers des nombres très grands (si les gradients antérieurs sont tous plus grand que 1) ou très petits (si les gradients antérieurs sont tous plus petits que 1). La méthode de BP se confronte donc souvent à des problèmes de divergence du gradient (aussi appelé le *exploding gradient* [2]) ou de stagnation du gradient (aussi appelé le *vanishing gradient* [2]), et qui défini une instabilité de la méthode BP.

### 2.1.2 Les réseaux de neurones récurrents (RNN)

Les réseaux de neurones récurrents (RNN) sont conçus pour traiter des données séquentielles. La structure du RNN, représentée dans la figure 2.8, regroupe des connexions avec les entrées ainsi qu'avec les états de sortie du réseau au dernier calcul.



**Figure 2.8** – Représentation d'un RNN (à une couche cachée).

Les séquences de texte, les signaux temporels, ou les séquences ADN sont des exemples typiques de données séquentielles [2]. Ces données sont distribuées en segments discrets pour plusieurs pas de temps  $t_i$  dans un ordre chronologique :  $\mathbf{u}(t_1)$ ,  $\mathbf{u}(t_2)$ , ...,  $\mathbf{u}(t_i)$ , ...,  $\mathbf{u}(t_f)$ . Chaque segment de données  $\mathbf{u}(t_i)$  est un vecteur colonne de dimension  $N_{in}$  représentant des valeurs numériques acquises au pas de temps  $t_i$  de la séquence. La dimension de  $\mathbf{u}(t_i)$  est fixe pour tous les indices  $t_i$ . Pour chaque pas de temps,  $\mathbf{u}(t_i)$  est utilisé comme vecteur d'entrée du RNN, pondéré par la matrice de poids  $\mathbf{W}_{in}$ .

Par exemple, un texte est un ensemble de données séquentielles qui peut être représenté par un vecteur binaire  $\mathbf{u}(t_i)$  encodant l'information textuelle en pas de temps discret  $t_i$

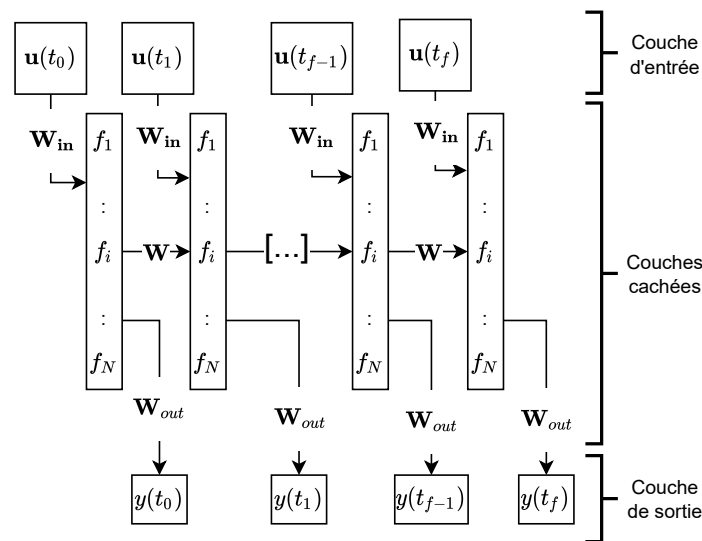
pour chaque caractère. La dimension de  $\mathbf{u}(t_i)$  serait égale au nombre de caractère possible dans le langage. Pour chaque vecteur  $\mathbf{u}(t_i)$ , une valeur 1 serait attribuée uniquement à l'indice correspondant au caractère [2], les autres indices du vecteur ayant des valeurs de 0. Les signaux numériques temporels d'un IMU sont aussi des données séquentielles réparties par segment pour chaque point de mesure à des instants  $t_i$ . Pour un IMU comprenant un accéléromètre et un gyroscope 3 axes,  $\mathbf{u}(t_i)$  serait formulé comme un vecteur de 6 dimensions dont chaque élément représente l'accélération et la vitesse angulaire de chaque axe. Les données séquentielles peuvent être considérées comme une catégorie distincte de données en raison de deux aspects principaux par rapport aux autres types de données (vecteur/matrice) [2] :

1. **La dimension variable des séquences** : Les données séquentielles ont une dimension variable dans le temps. Chaque séquence de données peut avoir un nombre d'indices différents. Dans plusieurs applications de contrôle ou de systèmes de détection, l'analyse en temps réel des données est importante. On cherche à obtenir une valeur de sortie en continu (à chaque pas de temps), comme le signal de contrôle d'un moteur ou de détection d'une problématique. Le traitement des séquences doit se faire pour chaque pas de temps, sans avoir à attendre l'acquisition de l'ensemble de la séquence de donnée. Dans les réseaux comme les FNN et les CNN [42], la dimension des données doit être fixe, ce qui les rend moins compatibles avec ce type d'application. Le RNN fait donc un traitement des données de manière discrète et chronologique de chaque segment  $\mathbf{u}(t_i)$  pour des séquences de taille arbitraire dans le temps.
2. **L'interdépendance des segments  $\mathbf{u}(t_i)$  dans les données séquentielles** : Chaque segment de la séquence est interdépendant. Autrement dit, l'information dans les données séquentielles est reliée à la fois aux valeurs et à l'ordonnancement des segments [2]. Cette information est modifiée ou perdue si l'on modifie l'ordre des pas de temps. Par exemple, l'information contenue dans la séquence de texte "madam Curie", est transformée si les indices  $t_i$  sont permutés pour constituer "radium came" ou est perdue, si l'on change l'ordre des indices pour "mu iamceadr". Il en va de même avec les signaux d'accélération du pied, dont l'information est aussi contenue dans la dimension temporelle. Pour extraire efficacement l'information dans ces données, la sortie de prédiction du RNN,  $y(t_i)$ , se doit de tenir compte de l'ordonnancement des segments de donnée d'entrée. Autrement dit, les états de sortie des neurones pour un segment d'indice  $t_i$ , doivent dépendre de l'entrée  $\mathbf{u}(t_i)$ , mais aussi de toutes les autres entrées précédentes de  $\mathbf{u}(t_{i-1})$  à  $\mathbf{u}(t_0)$ . Pour un FNN ou un CNN, le calcul d'une prédiction se fait de manière indépendante des précédents calculs effectués et ne permet pas d'obtenir cette propriété. Dans un RNN, les états des nœuds de la couche cachée du réseau sont conservés en mémoire à chaque itération et réutilisés de manière récursive. À chaque pas de temps, les perceptrons prennent comme argument l'entrée au pas de temps actuel,  $\mathbf{u}(t_i)$ , et l'état des neurones au pas de temps précédent  $\mathbf{x}(t_{i-1})$ . De manière récursive,  $\mathbf{x}(t_{i-1})$  est calculé à partir de  $\mathbf{u}(t_{i-1})$  et  $\mathbf{x}(t_{i-2})$ , et ainsi de suite. Ainsi, pour chaque itération, les états de sorties d'un RNN dépendent des données d'entrée pour l'ensemble des indices actuels et précédents de la séquence d'entrées. Une prédiction en sortie  $y(t_i)$  du RNN tiendra donc en compte l'ordonnancement des données dans une séquence. Si l'on traite la même séquence en modifiant l'ordre de ces indices, la sortie obtenue

par le RNN sera alors complètement indépendante à la première prédiction. Cette propriété est souvent décrite comme la mémoire d'un RNN.

### 2.1.2.1 Complexité d'entraînement d'un RNN avec la méthode BP

La représentation d'un RNN est aussi faite sous une forme déployée dans la littérature [2], c'est-à-dire en développant chaque itération d'un RNN comme un FNN composé de  $t_f + 1$  couches cachées, comme illustrée dans la figure 2.9. Dans cette représentation, le calcul des états des neurones fait à chaque indice de la séquence pourrait correspondre à une couche cachée d'un DNN. Chaque couche cachée prend en entrée une combinaison des états des noeuds de la couche précédente ainsi et des entrées à l'itération  $t_i$ . D'un point de vue théorique la représentation en boucle et la représentation développée est équivalente [2].



**Figure 2.9** – Représentation développée d'un RNN à une couche cachée.

Lors de l'entraînement, des données séquentielles avec un grand nombre de pas de temps sont utilisées afin de faire l'ajustement des poids. Cela pose plusieurs problèmes dans le cas d'un entraînement par rétropropagation. Pour une séquence de  $t_f$  indices et un réseau de  $N$  neurones, on peut voir, en représentant le réseau sous sa forme développée, que le nombre de calculs à faire dans la méthode de rétropropagation est de l'ordre de  $(t_f(N + N_{in})^{t_f})$  multiplications et  $((N + N_{in})^{t_f})$  additions. La complexité de calcul et le manque de stabilité de la méthode BP, vues dans la section 2.1.1 avec la méthode de BP, augmentent rapidement en fonction de  $t_f$ . Cette problématique fait en sorte que la méthode de BP peut prendre un grand temps de calcul et peut diverger. Plusieurs concepts ont été développés afin de s'affranchir de cette complexité, notamment avec les concepts de *backpropagation-decorrelation rule* (BPDC)[44], de *long short term memory* (LSTM) [45] et du *reservoir computing* (RC). Dans ce projet, on s'intéresse à la formulation du RC, qui sera présentée dans la section 2.1.3.

### 2.1.3 Principe émergent de RNN : les reservoirs computers (RC)

Les reservoir computers sont une forme de RNN définis par Jaeger dans les années 2000. La motivation du RC est de s'affranchir de la méthode de BP qui aurait servi à ajuster les poids dans les trois couches du reservoir. Dans le concept du RC, on fixe les poids des nœuds de l'ensemble de la couche d'entrée et cachée du réseau. Les poids de ces couches ne seront pas modifiés lors de la phase d'entraînement. Ces deux couches deviennent alors une boîte noire dont le rôle est de transformer les données pour plus facilement la classifier. Cette boîte noire est appelée le reservoir, comme représenté à la figure 2.10.

Évidemment, pour qu'un RC soit efficace, les connexions du reservoir ne doivent toutefois pas être laissées au hasard. Ce choix préliminaire de la distribution des poids de connexions du reservoir est le défi principal d'exploitation d'un RC. Comme proposé par Jaeger [46], le choix des poids se fait de manière aléatoire, selon une certaine distribution stochastique établie selon plusieurs hyperparamètres. Ces hyperparamètres permettent de modifier la nature de cette distribution des poids des couches d'entrée et cachée (amplitude moyenne, densité de connexions). L'ajustement des hyperparamètres pour optimiser la performance d'un RC devient un problème d'optimisation assez complexe qui est abordé avec plusieurs méthodes dans la littérature.

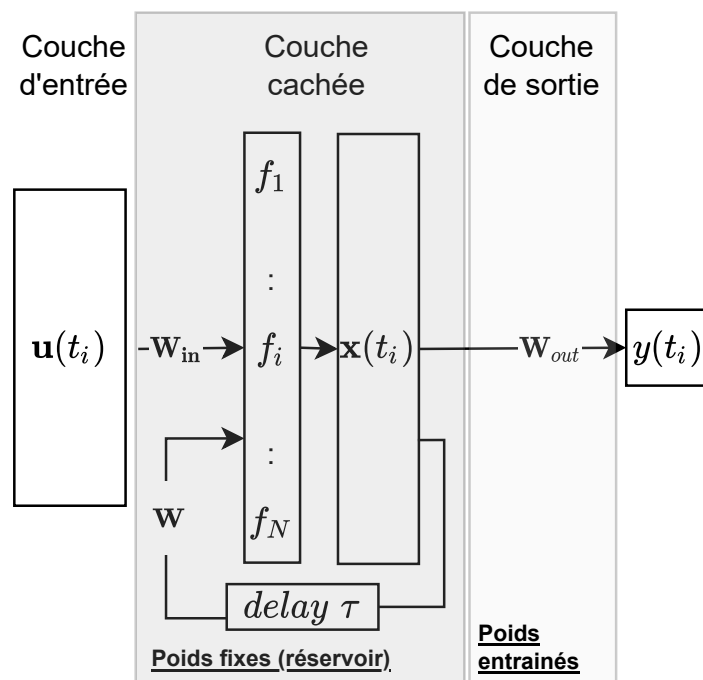


Figure 2.10 – Représentation d'un RC.

#### 2.1.3.1 Avantage des reservoir computers

En pratique, un RC peut être entraîné sur de longues séquences de données en n'ajustant que les poids de la couche de sortie durant l'entraînement. La complexité de cet entraînement est indépendante du nombre de pas de temps présents dans les données

séquentielles utilisées. En effet, même si le nombre de pas de temps dans les données augmente, l'entraînement correspond à la résolution d'un système linéaire  $\mathbf{x}(t)\mathbf{W}_{out} = \mathbf{y}(t)$ , de taille  $N \times N$  afin de trouver le meilleur  $\mathbf{W}_{out}$  pour corrélérer  $\mathbf{x}$  à une valeur objective  $\mathbf{y}_t(t)$ . Cet entraînement est très rapide et facilite énormément la mise en application d'un RNN.

## 2.2 Implémentation numérique de RC : echo state network

### 2.2.1 Formulation du ESN

Le ESN est une implémentation numérique du concept de RC formalisée par Jaeger [46]. Comme le reste des NN, le but du ESN est de faire une transformation non linéaire entre des entrées temporelles  $\mathbf{u}(t)$  vers un signal de prédiction  $\mathbf{y}(t)$  avec une architecture schématisée dans la figure 2.11.

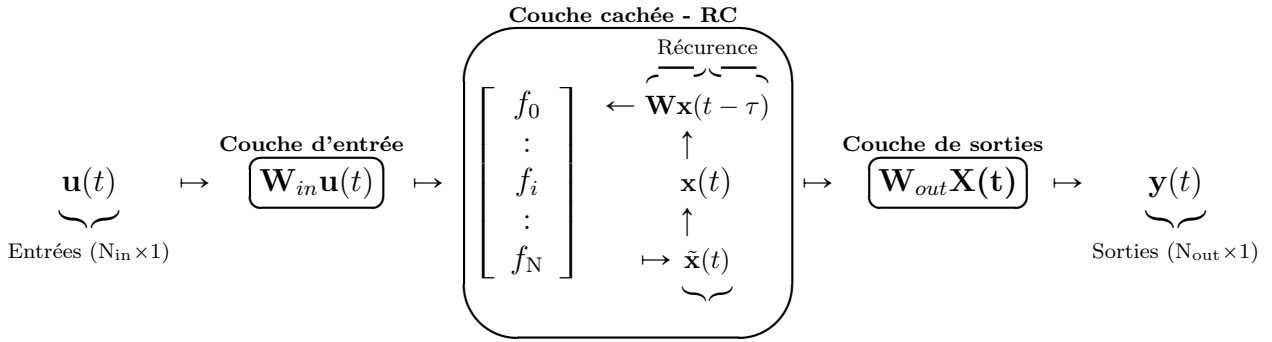


Figure 2.11 – Architecture du traitement d'un ESN.

La matrice de poids d'entrées  $\mathbf{W}_{in}$  ( $N \times N_{in}$ ) connecte le signal d'entrée  $\mathbf{u}(t)$  ( $N_{in} \times 1$ ) aux  $N$  neurones d'un réservoir (où  $N_{in}$  correspondent à la dimension des signaux d'entrées). Le résultat des fonctions d'activation correspond aux valeurs d'états du réservoir  $\mathbf{x}(t)$  ( $N \times 1$ ). La matrice de poids de sortie  $\mathbf{W}_{out}$  permet de faire la combinaison linéaire des états pour former un signal de prédiction  $\mathbf{y}(t)$  ( $N_{out} \times 1$ ) (où  $N_{out}$  correspond au nombre de signaux de prédiction). Dans le cas d'un réservoir récurrent, une matrice de poids de rétroaction  $\mathbf{W}$  ( $N \times N$ ) connecte les noeuds du réservoir avec ses états de sorties passés. Le paramètre  $\alpha$  est un facteur de pondération entre l'état présent et l'état du passé qui permet de contrôler le taux d'oubli des états passés  $\mathbf{u}(t)$  qui ajuste la dynamique de la réponse du réservoir. Le calcul itératif du ESN est :

$$\tilde{\mathbf{x}}(t) = \tanh(\mathbf{W}_{in} \mathbf{u}(t) + \mathbf{W} \mathbf{x}(t - \tau)) \quad (2.10a)$$

$$\mathbf{x}(t) = \alpha \tilde{\mathbf{x}}(t) + (\alpha - 1) \mathbf{x}(t - \tau) \quad (2.10b)$$

$$\mathbf{y}(t) = \mathbf{W}_{out} \mathbf{x}(t) \quad (2.10c)$$

où  $\tau$  représente un délai de 1 indice dans le passé des états  $\mathbf{x}$ .

### 2.2.1.1 Principe d'entraînement supervisé du RC

Dans un processus d'entraînement supervisé, les poids de sortie ( $\mathbf{W}_{out}$ ) d'un reservoir sont ajustés avec des données préliminaires observées par des séquences de données d'un signal d'entrée  $\mathbf{U}$  et d'un signal objectif  $\mathbf{Y}$  composés de  $T$  segments  $\mathbf{u}(t_i)$  et  $\mathbf{y}(t_i)$ . Ce processus d'entraînement se fait en deux phases : l'*entraînement* et le *test*. Une partie des séquences est utilisée uniquement pour l'*entraînement* ( $\mathbf{U}$  et  $\mathbf{Y}$ ) et le reste est utilisé uniquement pour le *test* ( $\mathbf{U}^{test}$  et  $\mathbf{Y}^{test}$ ).

Lors de l'*entraînement*, les données d'entrée  $\mathbf{U}$  ( $N_{in} \times T$ ) sont utilisées pour générer un ensemble d'états de sorties  $\mathbf{X} = [\mathbf{x}(t_1), \mathbf{x}(t_2), \mathbf{x}(t_3), \dots, \mathbf{x}(t_T)]$  ( $N \times T$ ) par le reservoir. Avec la sortie  $\mathbf{Y} = [\mathbf{y}(t_1), \mathbf{y}(t_2), \mathbf{y}(t_3), \dots, \mathbf{y}(t_T)]$  ( $N_{out} \times T$ ), on établit ensuite la matrice de poids de sortie  $\mathbf{W}_{out}$  ( $N \times N$ ) par une régression linéaire régularisée, soit la régularisation de Tikhonov [47] :

$$\mathbf{W}_{out} = \mathbf{Y}\mathbf{X}^T(\mathbf{X}\mathbf{X}^T + \gamma\mathbf{I})^{-1}, \quad (2.11)$$

Avec  $\mathbf{I}$  la matrice identité ( $N \times N$ ) et  $\gamma$  le coefficient de Tikhonov, un paramètre de régularisation prévenant la sur-contrainte du reservoir (over-fitting).

Une seconde phase de *test* permet d'établir la performance prédictive du reservoir en comparant la corrélation entre  $\mathbf{Y}^{predict}$  générée par une nouvelle séquence de données d'entrées et les nouvelles observations expérimentales  $\mathbf{Y}^{test}$ . La métrique de corrélation entre les deux signaux est la moyenne normalisée de la somme des erreurs au carré (NRMSE) :

$$\text{NRMSE} = \frac{1}{N_{out}} \sum_{i=1:N_{out}} \sqrt{\frac{1}{T} \frac{\sum_{j=1:T} (\mathbf{Y}_{ij}^{predict} - \mathbf{Y}_{ij}^{test})^2}{\sigma^2}}, \quad (2.12)$$

Avec  $N_{out}$  le nombre de signaux de prédictions différents,  $\sigma$  l'écart-type entre données de prédiction, et  $T$  le nombre de points dans la série de données de  $Y_{target}$ . Les matrices  $\mathbf{W}_{in}$ ,  $\mathbf{W}$  et la variable  $\alpha$  sont établies par des hyperparamètres qui restent fixes durant les phases *entraînement* et *test*. On peut toutefois les optimiser de manière itérative afin de minimiser le NRMSE. Certains algorithmes adaptés pour l'optimisation de la performance d'un RC sont disponibles en open source, comme OPTUNA [48]. D'un point de vue physique, cette optimisation permet d'ajuster la nature du RC selon une dynamique non linéaire (riche en information) et reproductible (non chaotique) pour maximiser ses performances prédictives [49].

## 2.2.2 Applications du ESN

Une fois correctement optimisé, un ESN est une fonction de traitement de données de type "boite noire",  $\mathbf{Y} = \mathbf{W}_{out} \text{ESN}(\mathbf{U}) = \mathbf{W}_{out} \mathbf{X}$ , qui se calcule par des opérations élémentaires (multiplications matricielles et fonctions tanh). Leurs applications sont versatiles, car elles ne requièrent pas de compréhension pointue de la physique des signaux traités, mais uniquement d'une quantité suffisante d'exemples observés du traitement souhaité. Les RC peuvent notamment être utilisés comme classificateurs, filtres, contrôleurs ou comme modèles de prédiction de systèmes dynamiques [41].

## 2.3 Implémentation physique d'un RC par MEMS-RC

Il est bien établi dans la littérature que le concept de RC est un principe intrinsèquement adapté à l'analogique, mais utilisé plus fortement par calcul logiciel en raison de la technologie relativement immature utilisée pour l'analogique [49]. Cette idée d'implémentation physique (hardware) des RC est au coeur du travail de développement fait par le présent groupe de recherche. On peut voir se développer plusieurs travaux sur des systèmes physiques offrant des propriétés non linéaires capables d'être exploitées comme un RC, par des méthodes optiques [50, 51] et électroniques [52]. Ces méthodes exploitant des RC analogiques montrant des temps de réponse très courts permettent de réduire les délais de traitement et la taille du dispositif. L'implémentation d'un RC dans un MEMS (appelé le RC-MEMS) fait par le groupe de recherche de M. Sylvestre est présentée dans cette section. La technologie développée au sein du groupe est un **reservoir à délais couplé (DCR)**.

### 2.3.1 Similarité entre un oscillateur de duffing et une fonction d'activation numérique

Ce reservoir est une forme particulière de RC dont la fonction d'activation  $f(x)$  d'un neurone correspond à la réponse dynamique en déplacement ( $x(t)$ ) d'une poutre physique bi encastree micro fabriquée. Les effets mécaniques des encastremets de la poutre produisent un comportement vibratoire non linéaire à grand déplacement que l'on peut représenter par l'**oscillateur non linéaire de Duffing** [53, 54],

$$m_{\text{eff}}\ddot{x}(t) + \omega_0^2 x(t) + \frac{\omega_0}{Q}\dot{x}(t) + \beta^3(t) = F(t), \quad (2.13)$$

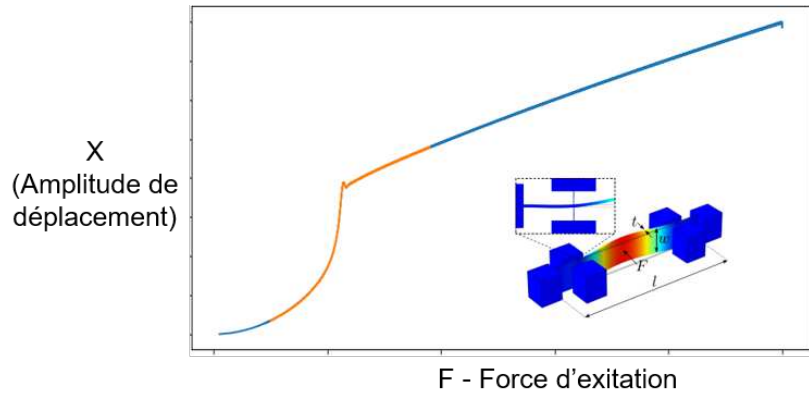
où  $m_{\text{eff}}$ ,  $\omega_0$ ,  $Q$  et  $\beta$  sont respectivement la masse effective, la fréquence naturelle, le facteur de qualité et le coefficient de Duffing de l'oscillateur. Le système de poutre bi encastree développé par le groupe se caractérise par  $m_{\text{eff}} \approx 1.14610^{-11}$  kg,  $f_0 = \frac{\omega_0}{2\pi} \approx 500$  kHz,  $Q \approx 145 \pm 10$  et  $\beta \approx 10^{24} \frac{\text{Hz}^2}{\text{m}^2}$  [55]. Pour une fréquence proche de sa résonance, le modèle de Duffing crée une réponse vibratoire en fonction de l'excitation impliquant un saut non linéaire en amplitude, comme illustré dans la figure 2.12. La réponse dynamique de l'oscillateur de Duffing dans une zone orange de la courbe a une forme d'un sigmoïde comme celle d'une  $\tanh(x)$  de la figure 2.5. Cette propriété non linéaire du système peut être exploitée afin de faire du calcul par RC de signaux d'accélération externe.

On illustre clairement la ressemblance entre la fonction non linéaire  $\tanh$ , utilisée dans les réseaux ESN par rapport à la fonction de réponse vibratoire de la poutre dans la figure 2.13.

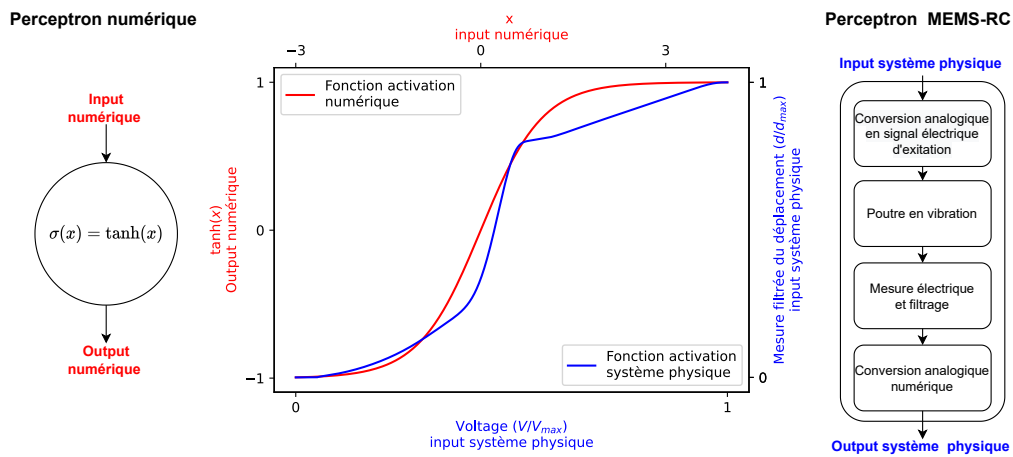
### 2.3.2 Méthode d'excitation et de lecture du reservoir MEMS

En raison de sa haute fréquence naturelle, la micro poutre de Duffing n'est pas directement sensible aux accélérations externes. Ce couplage se fait donc par un second système vibrant, beaucoup plus massif (une masse inertielle mobile) et beaucoup plus sensible aux accélérations  $\mathbf{U}(t)$  basses fréquences (comme l'accélération d'un pied durant la marche).





**Figure 2.12** – Réponse de déplacement pour une micro poutre bi encastree dans un régime de grande amplitude non linéaire correspondant au modèle de Duffing [3].



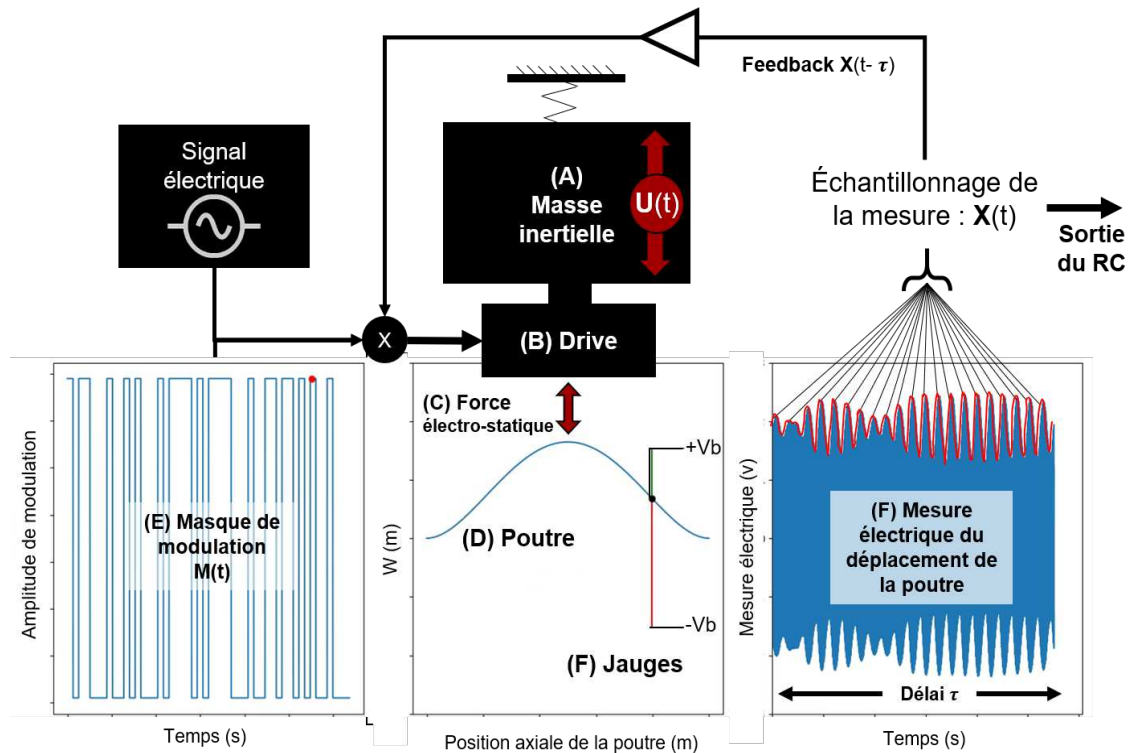
**Figure 2.13** – Similarité de la réponse de la fonction d'activation  $\tanh(x)$  par rapport à celle de la réponse en vibration de la poutre (RC-MEMS) développée [3].

On peut voir ce système illustré dans la figure 2.14. L'extrémité de la masse inertielle à proximité de la poutre correspond à un actionneur (drive) mis sous tension par un signal alternatif à fréquence fixe et génère une force électrostatique  $F_{es}$  qui fait vibrer la poutre. Cette force électrostatique est inversement proportionnelle au carré de la distance  $d$  entre la drive et la poutre,

$$F_{es} = \frac{\epsilon_r \epsilon_0 A V^2}{2d^2}, \tag{2.14}$$

où  $A$  est la surface d'exposition entre la poutre et la drive et  $V$  est le tension électrique. Le mouvement causé par l'accélération externe de la masse inertielle entraîne une variation de la distance entre la drive et la poutre fixe, influençant l'amplitude de vibration de la poutre. Ensuite, on peut lire le déplacement en flexion de la poutre à l'aide d'un signal électrique via des jauges piézo-résistives. Le processus d'excitation et de lecture de la micro poutre en fonction de l'accélération par l'entremise d'une masse inertielle est schématisé dans la figure 2.14 [55].

Pour un délai  $\tau$ , on segmente la réponse en déplacement de l'oscillateur dans le temps



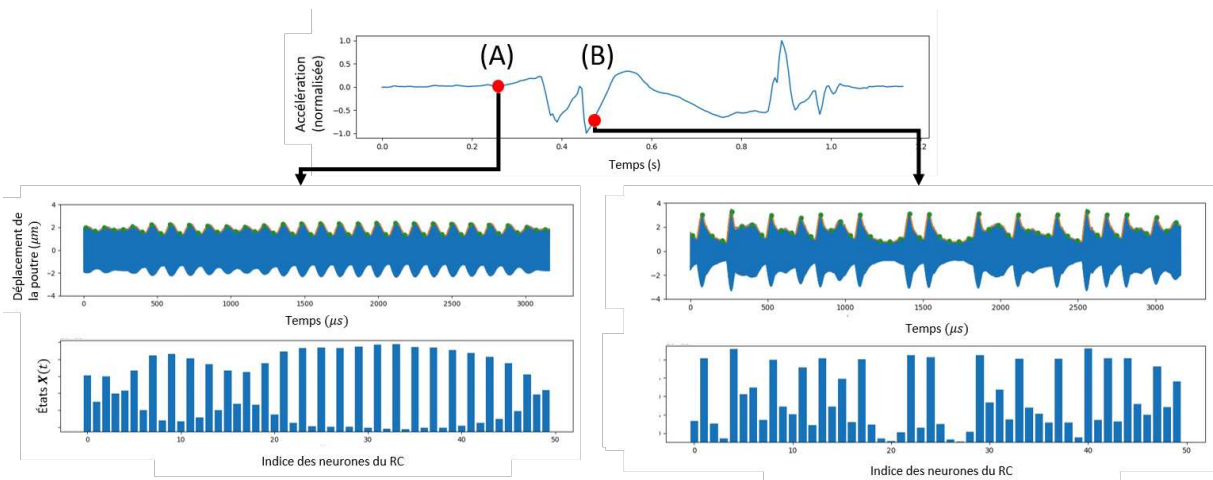
**Figure 2.14** – Traitement du signal par le système physique. La masse inertielle (A), sensible à l'accélération externe  $U(t)$ , se déplace et fait varier la position relative entre la drive (B) et la poutre (D). La force électrostatique (C) générée par un courant alternatif modulé en amplitude par un masque d'échantillonnage (E) provoque le déplacement en flexion de la poutre  $x(t)$  mesuré en différents points par la jauge en (F). Les échantillons de mesure forment les états du réservoir pour un intervalle de temps  $\tau$  :  $\mathbf{X}(t)$ . Une valeur de rétroaction  $\mathbf{X}(t - \tau)$  est renvoyée au signal d'excitation pour créer un réservoir récurrent. [55]

en  $K$  échantillons différents, formant les  $K$  états du réservoir  $\mathbf{X}(t)$ . Afin d'obtenir des états non monotones amplifiant la richesse de la réponse du réservoir, on impose une fluctuation périodique du voltage nominal du signal électrique avec un masque de modulation  $M(t)$ . Ce masque module le voltage entre deux valeurs binaires de manière cyclique à chaque mesure échantillonnée sur la poutre. Afin d'ajouter une capacité de mémoire au RC-MEMS, une récurrence est aussi créée par une rétroaction modulée par les états du réservoir sur le voltage du signal électrique de la drive.

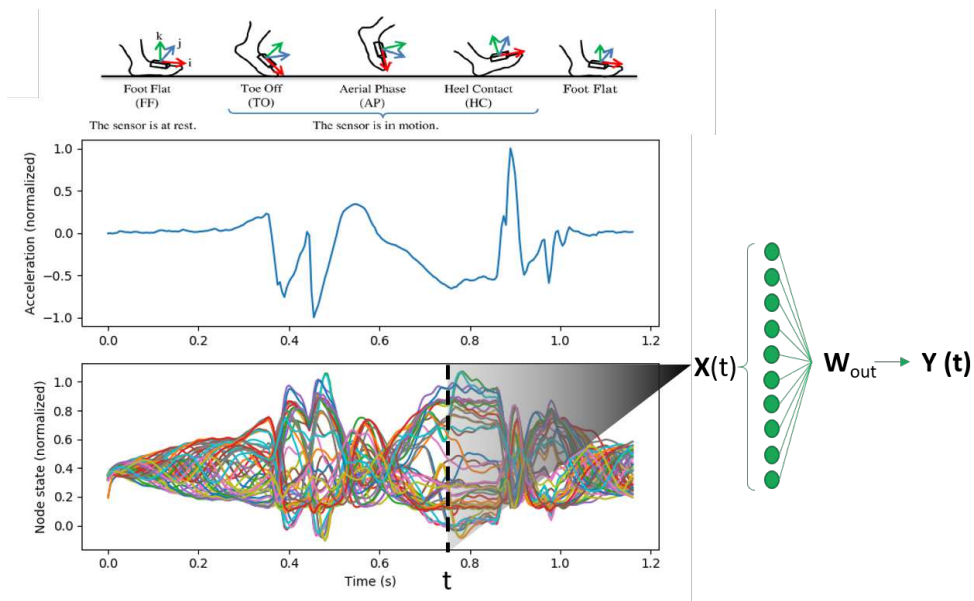
### 2.3.3 Extraction et exploitation des données de sorties du RC-MEMS

Il est donc possible avec le RC-MEMS de générer un signal de sortie  $\mathbf{X}(t)$  correspondant aux états du réservoir lorsqu'il est soumis à un signal d'accélération, tout comme un ESN. La figure 2.15 montre les résultats d'une simulation numérique de l'oscillateur de Duffing, où le niveau d'accélération influence les valeurs d'états du réservoir. Les états peuvent être ensuite combinés comme un réservoir logiciel pour former une sortie de

prédiction  $\mathbf{Y} = \mathbf{W}_{out}\mathbf{X}$  corrélée à diverses observations expérimentales, comme illustré sur la figure 2.16.



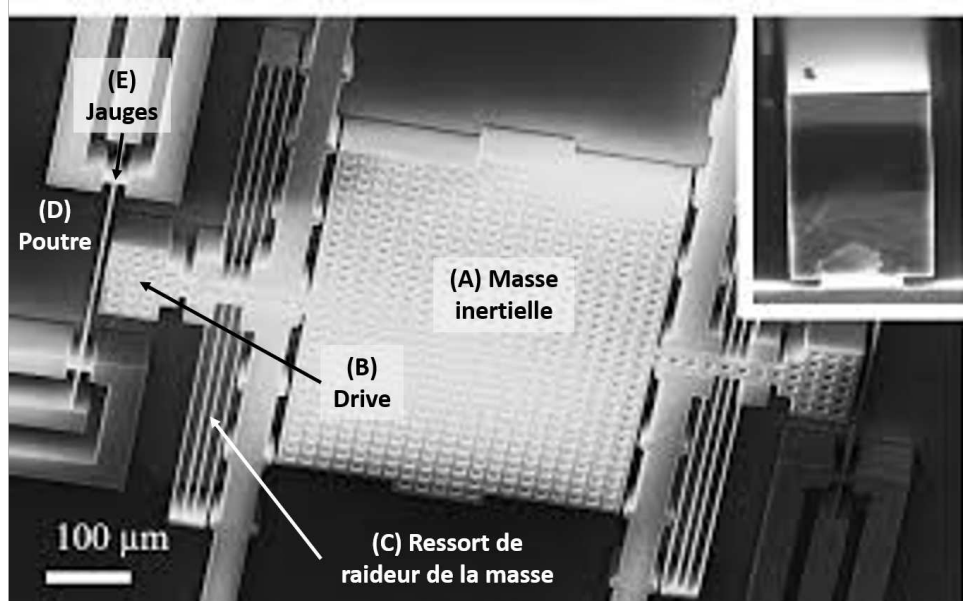
**Figure 2.15** – Illustrations de l'influence de l'accélération (graphique du haut) sur le signal de réponse de la poutre (graphiques du centre) et donc sur les 50 états  $\mathbf{X}$  du réservoir (graphiques du bas). La faible accélération en (A) génère des états réguliers proportionnels au masque  $\mathbf{M}$ , alors que l'accélération non nulle en (B) génère des états irréguliers et non linéaires.



**Figure 2.16** – États du réservoir  $\mathbf{X}(t)$  (en bas) pour l'accélération d'un cycle de marche complet (en haut). La combinaison linéaire de  $\mathbf{X}(t)$  par la matrice  $\mathbf{W}_{out}$  donne le signal de sortie  $\mathbf{Y}(t)$  représentant une information utile par la suite en application.

### 2.3.4 Avancements actuels de la technologie

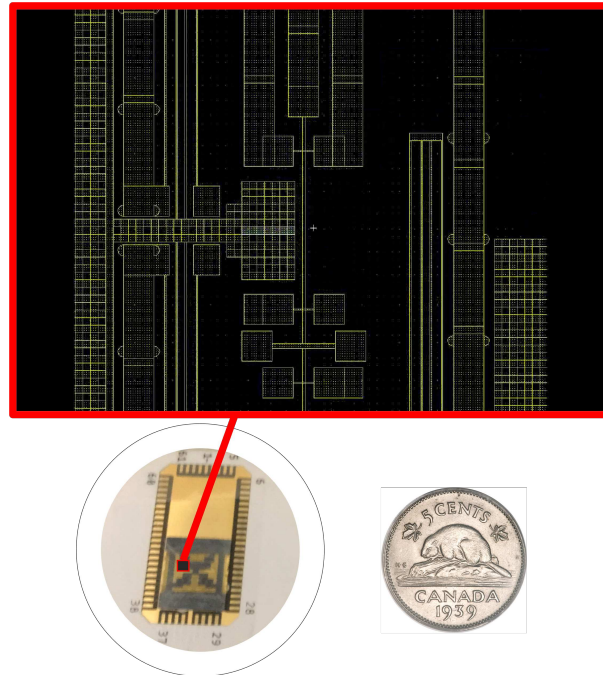
Le processus de conception et la microfabrication d'oscillateurs MEMS physiques, présenté à la figure 2.17, sont la première partie du développement technologique du groupe de recherche. Le présent groupe est le seul actuellement qui étudie le potentiel d'intégration et de miniaturisation des RC-MEMS. La taille actuelle du MEMS est représenté dans la figure 2.18 Le groupe a démontré expérimentalement et numériquement que le comportement vibratoire non linéaire de la micro poutre bi encastree correspond fidèlement au modèle de Duffing [56]. On observe une concordance entre un modèle numérique par éléments finis, le comportement expérimental de la poutre et le modèle analytique de Duffing [3]. Le groupe a aussi élaboré un circuit électronique permettant d'exciter le MEMS, de faire la lecture du déplacement et de la rétroaction [4, 56–58].



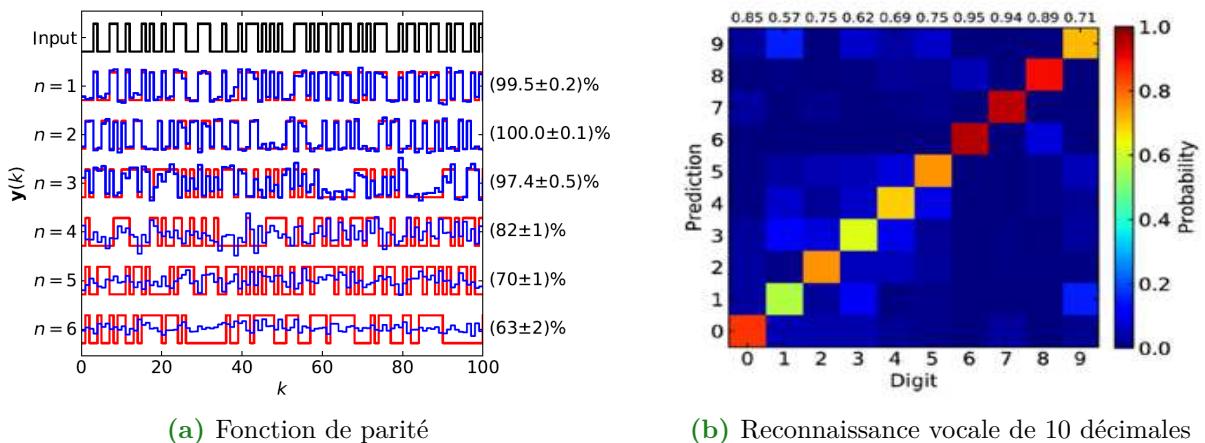
**Figure 2.17** – Image au microscope du dispositif. La masse inertielle (A) est suspendue sur un ressort (C), sensible aux accélérations extérieures. La drive (B) est solidaire à la masse inertielle et excite la poutre (D) électriquement. Les jauges piézoélectriques (E) permettent la lecture de la déflexion de la poutre. [4]

Plusieurs démonstrations de la performance du reservoir à délais MEMS ont été publiées [3, 55, 57]. La fonction de parité et la reconnaissance vocale montrées à la figure 2.19 sont des tâches élémentaires qui ont permis de confirmer la capacité de calcul du reservoir ainsi que d'offrir une compréhension de la configuration des paramètres du réseau permettant d'optimiser les performances.

Le RC-MEMS constitue donc un système intégré offrant les fonctions d'un capteur et d'un processeur qui peut remplacer un appareil intelligent conventionnel. Les performances testées jusqu'à maintenant ont été validées avec un système électronique externe en laboratoire. Le travail actuel (2021-2022) du groupe est orienté vers l'interfaçage électronique du RC-MEMS pour avoir une version portable du système électronique pour tester des applications concrètes comme en biomécanique. Le groupe de recherche souhaite intégrer le MEMS sur une carte de quelques  $cm^2$  et d'une consommation d'environ 25-50  $mW$ . La capacité de calcul du RC-MEMS ainsi que le temps de réponse se limite



**Figure 2.18** – Aperçu de la taille du RC-MEMS actuelle (version 2020-2021)



**Figure 2.19** – Performance du dispositif physique dans des tâches testées dans le groupe de recherche [3], [55]. Le taux de réussite jusqu'à 70% pour la fonction de parité d'ordre 5 et le taux de prédiction entre 57% et 90% du classement vocal des nombres démontrent ses capacités de calcul.

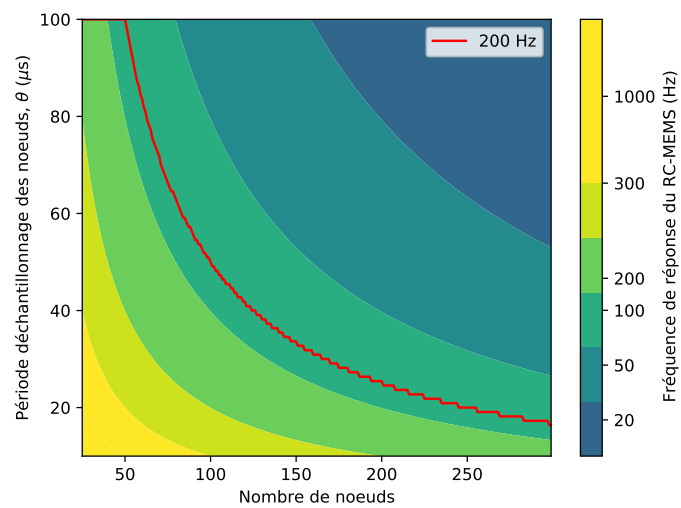
pour l'instant à la dynamique d'un seul résonateur physique pouvant aussi être amélioré par des concepts d'architecture de réservoirs MEMS couplant plusieurs résonateurs physiques, tel que proposé dans [55].

### 2.3.5 Retour sur les contraintes des RC-MEMS dans l'application des RC dans les tâches biomécaniques

Dans le cadre de ce mémoire, le ESN est utilisé afin de démontrer les performances d'un RC pour traiter des données reliées au domaine de l'analyse de la marche. Toutefois, l'objectif global du projet est de réussir à exploiter le MEMS-RC afin d'accomplir ce traitement. Une considération importante de ce mémoire est donc d'évaluer les performances du ESN dans des conditions compatibles avec ce qui peut se faire avec le RC-MEMS. Cela implique que certaines contraintes sur les données d'entrées et sur le design du reservoir doivent être considérées :

- Le RC-MEMS est un dispositif sensible aux accélérations unidimensionnelles, sensible dans l'axe de mouvement de la masse inertielle (l'orientation du capteur peut donc être optimisé selon l'axe fournissant les meilleures prédictions). Les données d'entrées se définissent donc comme un signal unidimensionnel mesuré sur le pied.
- Aucun filtrage ni prétraitement du signal d'entrée n'est possible.
- Le nombre de noeuds du reservoir se doit d'être raisonnable, de l'ordre de 100-200 noeuds, car le RC-MEMS impose un délai d'échantillonnage, qui devient trop grand lorsque le nombre de noeuds dépasse une valeur de 100 noeuds. Comme montré dans la figure 2.20, la fréquence de traitement du système RC-MEMS est inversement proportionnel au nombre de noeuds ainsi qu'au délai d'échantillonnage  $\theta$ . Dans une application biomécanique, la fréquence de traitement de l'ordre de 200 Hz (ligne rouge dans la figure 2.20) serait largement suffisant pour capturer la dynamique des signaux d'accélération, dont le contenu fréquentiel est en dessous de 15 Hz.
- De même, la fréquence des données d'entrée se doit d'être compatible avec le délai de traitement fait par un MEMS-RC. En effet, un MEMS-RC prend minimalement environ  $50 \mu\text{s}$  par noeuds, ce qui donne pour un RC de 100 noeuds un temps total de traitement de l'ordre 5 ms, ce qui donne une fréquence de 200 Hz.

Ainsi, selon ces critères, les démonstrations d'application du RC pour des tâches biomécaniques seront avec des signaux de l'ordre de 200 Hz et avec un reservoir de 100 noeuds. Les données d'entrées qui sont compatibles avec le RC-MEMS seraient des signaux unidimensionnels d'accélération, directement mesurés sur la personne (pieds, bras, taille) et les signaux seront exploités par le RC sans prétraitement (aucun filtrage ou modifications du signal ...).



**Figure 2.20** – Délai d'utilisation d'un RC-MEMS en fonction du nombre de noeuds et de la période d'échantillonnage. La ligne rouge représente la fréquence de 200 Hz des données biomécaniques utilisées dans le reste de ce mémoire.

# Chapitre 3

---



## Détection des évènements de marche

### Éléments clé

Ce chapitre démontre l'application de la détection des évènements de marche (GED) avec le reservoir computing (article publié : [Detecting Gait Events from Accelerations Using Reservoir Computing](#))

- La détection des évènements de marche se fait en entraînant le ESN sur des signaux binaires dont les valeurs sont de 1 uniquement aux indices temporels des évènements de marche de chaque cycle de marche.
- Une grande variabilité de données ont été utilisées pour l'entraînement du RC, soit 4 bases de données, incluant un total de 28 participants, 15 conditions de marche différentes, 4 positions du IMU sur le corps. Un capteur positionné sur le pied performe mieux pour le GED avec le ESN que ceux sur le haut du corps.
- Le ESN peut prédire les évènements de marche dans de multiple conditions (sur tapis roulant ; sur plancher intérieur plat et avec obstacle ; sur plancher extérieur) et de course (sur plancher intérieur et extérieur). La prédiction en erreurs absolues moyennes (MAE) reste en dessous de 100 ms (95<sup>e</sup> percentile de l'ensemble des signaux d'accélération testés) dans l'ensemble des conditions testées.
- L'utilisation d'un ESN sur des données de nature différentes de celles utilisées durant l'entraînement occasionne de grandes erreurs (au dessus de 250 ms). Toutefois, le ESN peut être entraîné sur les trois bases de données et obtenir des résultats presque aussi bons qu'un ESN entraîné individuellement sur l'une de ces bases de données.
- Comparativement à quatre autres algorithmes de la littérature, le ESN est en mesure de performer avec un MAE d'au plus 10 ms au dessus de ceux-ci avec des conditions de marche intérieur (subset  $MAR_{IW}$ ) et extérieur (subset  $MAR_{OW}$ ). Il est plus robuste que deux d'entre eux (meilleur taux de vrai positif (TPR) et spécificité (SPF)) dans des conditions incluant de la marche et de la course extérieur (subset  $MAR_{OWR}$ ).
- Six évènements de marche (GE) différents ont été prédits par le ESN. Les résultats montrent que les GE qui surviennent lors de fortes accélérations sont plus faciles à prédire, comme les évènements au début de la phase d'appui (HS et HP). Les évènements survenant à la fin de la phase d'appui du cycle de marche (TP et TO), ont une accélération moins grande et sont prédits avec une précision variable. Les évènements au centre de la phase d'appui (FF et HO) se trouvent dans une zone du cycle de marche sans aucune accélération et sont les moins bien prédits.

### 3.1 Abstract

---

Segmenting the gait cycle into multiple phases using gait event detection (GED) is a well-researched subject with many accurate algorithms. However, the algorithms that are able to perform accurate and robust GED for real life environments and physical diseases tend to be too complex for their implementations on simple hardware systems limited in computing power and memory, such as those used in wearable devices. This study focuses on a numerical implementation of a Reservoir computing (RC) algorithm called the Echo state Network (ESN), that is based on simple computational steps that are easy to implement on portable hardware systems for real-time detection. RC is a neural network method that is widely used for signal processing applications, and uses a fast-training method based on a ridge regression adapted to the large quantity and variety of IMU data needed to use RC in various real-life environment GED. An ESN was used to perform offline GED with gait data from IMU and ground force sensors retrieved from three databases, for a total of 28 healthy adults and 15 walking conditions of various walking patterns and ground conditions. Our results show the ESN is robust enough to obtain good detection results in all conditions if the algorithm is trained with variable data that match those conditions. The distribution of the mean absolute errors (MAE) between the detection times from the ESN and the force sensors were between 40 and 120 ms for 6 defined gait events (95th percentile). We compared our ESN with four different state-of-the-art algorithms from the literature. The ESN obtained a MAE not more than 10 ms above three other reference algorithms for normal walking indoor and outdoor conditions and yielded the 2nd lowest MAE and the 2nd highest true positive rate and specificity when applied on outdoor walking and running conditions. Our work opens the door to using the ESN as a GED for applications in wearable sensors for long-term patient monitoring.

### 3.2 Introduction

---

The functional analysis of the human locomotion allows defining the gait cycle in time intervals [1] called the gait phases, they are distinguished by specific gait events. The gait event detection (GED) can thus be considered as a fundamental processing step needed for gait analysis. The two main gait events are the heel strike (HS) and the toe off (TO) that respectively define the beginning of the stance phase and of the swing phase. Multiple clinical analysis and applications use gait events, such as foot clearance estimation [33] and monitoring the physical behavior of participants with neurological [17, 59] or osteoarthritis [59] disease. Some of the clinical applications requiring real time GED include functional electrical stimulation [16, 27] and gait retraining [60, 61]. Among several technologies used for GED, user-friendly, light, wireless and compact inertial measurement unit (IMU) sensors are crucial to provide follow-ups in the daily movements of the participants. IMU sensors have thus become an essential tool for research and diagnosis. However, the accelerations captured by an IMU sensor are exposed to variable movements (nominal cadence, regularity, symmetry) inherent to the ecological environment and pathological walking conditions. Pathologies, disease progression and participant fatigue contribute to the movement variation and the speed of the gait. External conditions, such as the type

of ground and the walking trajectory, can also change the gait dynamics. Several complex IMU-based GED processing methods have therefore been widely explored over the past decade to increase the precision and the robustness of GED under various walking conditions. GED methods use different strategies, such as biomechanical models [62], empirical filtering and signal processing methods [5, 23, 27, 63–65], time-frequency analysis [39, 66], symbols- or statistics-based methods [67, 68] and hidden Markov models [69]. Most algorithms need to be designed for acquisition protocols that use only one sensor position on the body to reduce complexity and cost [5, 27, 38, 39, 63, 69]. The previous studies are limited by most algorithms being designed to only distinguish between two gait events : the HS and the TO [5, 39, 63, 64, 66]. They distinguish the phase when the foot is in contact with the ground (stance phase) from when it moves (swing phase). Biomechanical models or empirical filtering methods are also limited because they often require parameter modifications when new walking conditions are considered [27, 70]. This paper implies a new approach for GED by using a class of artificial neural network (ANN) algorithms called reservoir computing (RC) [71]. Recently, ANN algorithms have become prominent due to highly accessible digital data [2]. Affordable IMU sensors and force sensitive resistors (FSR) make easier large-scale gait data collection to apply ANN for GED. RC specializes in processing temporal signals such as accelerations. It can also be implemented quickly with a fast and systematic training methodology to adjust coefficients using a linear regression. In this study, we use the numerical implementation of RC called the Echo States Network (ESN) [72]. Many applications use the ESN algorithm [73], but only a few are applied to biomechanics, such as gesture recognition [74], muscle drive-in actuation [75] and exoskeleton control [76]. The ESN model is much simpler to implement on hardware devices compared to most of the advanced GED models recently developed. It can be applied to many walking conditions without increasing the algorithm complexity. To the best of our knowledge, the current literature neither addresses the application of ESN for GED.

As other ANN algorithms, RC models can be seen as black box models that learn how to process an input to give a useful output prediction. The RC model is applied using a supervised training method, where the predictions are adjusted to be as close as possible to a useful target, that is hard to compute otherwise. In this study, we exploited the RC to learn the best relation between example datasets of input and output target pairs. The inputs being the acceleration signals and the output being a binary indicators signals indicating the gait events. This method is separated in two steps called the training and the testing. The training step adjust the trainable parameters of the RC model to establish the optimal relationship between the targets and the RC model predictions. We used many examples of input-target pairs based on experimental measurements. The parameters of the RC model are adjusted to minimize the error between the prediction computed by the RC and the target. The testing step evaluates the capability of the RC to correctly predict the target from new and unseen data (testing dataset). This indicates how well the RC will perform in operation. We measured the RC performance by the error between the model predictions and the targets of this new independent dataset.

The novelty of our study is to demonstrate the benefits of the ESN approach for GED by adjusting the model for the detection of multiple events (in addition to HS and TO) over multiple walking conditions using a fast training methodology. No GED algorithm in the literature had the advantages to offers similar adaptability with the low

complexity needed for hardware implementation in simple wearable devices. Our objective is to characterize the ESN performance for GED for six types of event in the gait cycle. We considered different aspects impacting its performance : the characteristics of the training data used, the nature of the sensor data (orientations and positions on the body and the ground walking conditions), as well as the detection capability on the different types of events. Multiple training and testing datasets from three different databases are used to understand the impact of each of these aspects. We also compared the performance of the ESN to other algorithms in the literature for the HS and the TO gait events.

The methodology presents how to collect, format and process the gait data using a reservoir computer algorithm. Section 3.3.1 introduces the three databases of experimental gait data used for our experiments. Section 3.3.2 explains the process to prepare the gait data in a usable format for the ESN training and testing. Section 3.3.3 presents the ESN algorithm in details and section 3.3.4 presents a comparative GED algorithm tested in this work. Section 3.3.5 presents the standard metrics used to evaluate the GED algorithm performance. Section 3.4 presents the results for the ESN and the comparative algorithms. Finally section 3.5 is a discussion about the advantages and limitations of the ESN and it establishes a comparison of the ESN performance with the other algorithms from the literature.

## 3.3 Methods

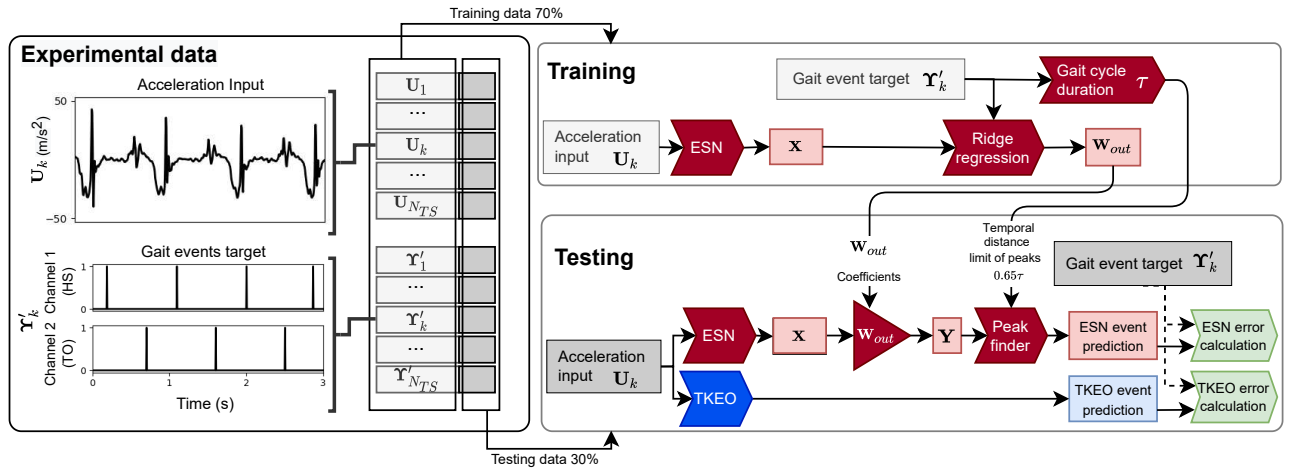
---

The evaluation process of GED performance of the ESN and the Teaser Keaser energy operator (TKEO) algorithms is presented in the flowchart of figure 3.1. In this process, we used experimental data of the acceleration and ground reaction force measured on participants during variable walking tests, as detailed in section 3.3.1. We apply preprocessing steps including identifying gait events from force sensors signals, structuring the data in timeseries and partitioning the data in a training set and a testing set, as explained in section 3.3.2. We use a supervised training methodology, which implies that the evaluation process of the ESN was separated in two main processing steps : the training and the testing. As explained in the section 3.3.3, the training step established the mean gait cycle duration of the gait data ( $\tau$ ) and the output weight matrix ( $\mathbf{W}_{\text{out}}$ ) of the ESN. The testing step of the ESN predicted the gait events using the peaks in its output signal ( $\mathbf{Y}$ ). We used the TKEO GED algorithm [5], presented in section 3.3.4, for comparison purpose in the testing step. We established the ESN and the TKEO performance with the error between the prediction events and the target events, expressed as the positive prediction value (PPV), the true positive rate (TPR), the T1 score and the mean absolute error (MAE) between event times, as explained in section 3.3.5.

### 3.3.1 Data Collection

---

We use three database to test the ESN under a large variability of the walking conditions, namely the MAREA, CIRRIIS and UDS database, all including gait acceleration and ground force signal timeseries recorded on healthy adults. The MAREA gait data-



**Figure 3.1** – Procedure used to evaluate the performance of the ESN and of the TKEO algorithms. Arrow boxes represent calculation steps and square boxes represent data. The experimental data were collected during walking tests as explained in section 3.3.1 and preprocessed as explained in section 3.3.2. The input time series  $\mathbf{U}_k$  are acceleration signals recorded on the participant left foot. The output timeseries  $\mathbf{Y}'_k$  are the gait event targets identified with ground force sensors and formatted as a multi-channel binary signal. Each channel indicated the event time indices of one gait event class. The timeseries lists of the inputs  $\{\mathbf{U}_1, \dots, \mathbf{U}_k, \dots, \mathbf{U}_{N_{TS}}\}$ , and the outputs  $\{\mathbf{Y}'_1, \dots, \mathbf{Y}'_k, \dots, \mathbf{Y}'_{N_{TS}}\}$ , represent a subset of multiple input and output timeseries for each participant and walking condition considered in the procedure. The timeseries are separated in two groups, one for the training step of the ESN algorithm and one for the testing step of both ESN and TKEO algorithms. The ESN algorithm steps are represented by the red boxes and explained in section 3.3.3. The TKEO algorithm steps are represented by the blue boxes and explained in section 3.3.4. The green boxes evaluate the error between predicted and target events and establish performance criteria as presented in section 3.3.5.

base<sup>1</sup> is an open source database of accelerations and prelabelled gait events recorded under various walking and running conditions that has been already used to test various GED algorithms [5, 38, 39]. We also assembled the CIRRIS and the UDS databases for the purpose of this study respectively at the CIRRIS research center<sup>2</sup> (Université Laval, Canada) and at the Université de Sherbrooke (Canada). All participants in the CIRRIS and UDS databases gave written informed consent to participate in the study, according to experimental procedures approved by the IRDPQ-CIUSSS-CP research ethical committee (2021-2269)<sup>3</sup>. Because of the different acquisition frequencies in each database, the data were resampled at 200 Hz, using polyphase filtering [77]. Each database was established under independent conditions (treadmill, indoor, outdoor, pathology simulation, obstacles). Table 3.1 gives an overview of all conditions recorded in each database,

1. MAREA : Movement Analysis in Real-world Environments using Accelerometers [https://wiki.hh.se/caisr/index.php/Gait\\_database](https://wiki.hh.se/caisr/index.php/Gait_database)

2. CIRRIS : Centre Interdisciplinaire de Recherche en Réadaptation et Intégration Sociale <https://www.cirris.ulaval.ca/en/>

3. IRDPQ-CIUSSS-CP : Institut de Réadaptation en Déficience Physique de Québec et Centre Intégré Universitaire de Santé et de Services Sociaux de la Capital Nationale

**Table 3.1** – Databases characteristics

Database	Walking condition	# participants	Duration (min)	Speed (m/s)	Type and number of time series selected in each subset							
MAREA	Treadmill walking TW	11	$\approx 5$	1,1 to 2,2	MAR <sub>TW</sub> 11	MAR <sub>G1</sub> 11	MAR <sub>G2</sub> 11	MAR <sub>all</sub> 11				
	Treadmill run TR		$\approx 5$									
	Treadmill incline walk TI	11	12	SS*								
	Indoor flat walk IW		3									
	Indoor flat run IR	9	3	SS								
	Outdoor walk OW		3									
	Outdoor run OR	3	3	SS								
	Total ( $N_{TS}^{**}$ )					11	11	11	9	18	33	62
CIRRIS	Normal walk N	2	2	for each speed	[0.8-1.2 -1.4-1.6]	CIR <sub>N</sub> 10	CIR <sub>TO</sub> 10	CIR <sub>TL</sub> 10	CIR <sub>FF</sub> 10	CIR <sub>all</sub> 10		
	Right toe out TO											
	Right trunk leaning TL											
	Right foot flat FF											
	Total ( $N_{TS}^{**}$ )					10	10	10	10	40		
UDS	Normal walk T1	15	8	SS	UDS <sub>T1</sub> 15	UDS <sub>T12</sub> 15	UDS <sub>T13</sub> 15	UDS <sub>all</sub> 15				
	walk with obstacles T2											
	walk around obstacles T3											
	Total ( $N_{TS}^{**}$ )					15	15	14	30	29	29	44

\*SS : self-selected speed \*\*  $N_{TS}$  : the number of time series in each subset, is equal to the cumulative number of participants selected in each walking condition. This gives an indication of the amount of data variability in each subset. The larger  $N_{TS}$  is, the greater is the variability of the gait data (more patients and more different gait conditions).

including ground conditions, number of participants, duration and walking speed. Details of the data collection process for each database are given below.

### 3.3.1.1 MAREA Database

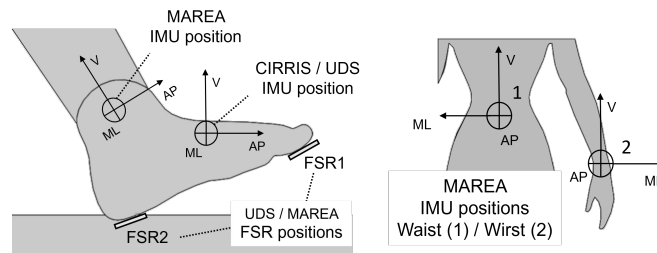
The MAREA database was collected on 11 subjects under 5 indoor conditions : flat treadmill walking (TW) and running (TR), inclined treadmill (TI), indoor flat walking (IW) and running (IR). The database also covers 9 subjects under 2 outdoor conditions : outdoor walking (OW) and running (OR). Each subject had a 3-axes Shimmer3 accelerometer ( $\pm 8g$ ) attached to their waist, left wrist and left and right ankles (figure 3.2). Two force-sensitive resistors were also placed under the foot, at the heel and the toes, to acquire foot pressure measurements on the ground.

### 3.3.1.2 CIRRIS Database

The CIRRIS database was recorded on two healthy participants, during normal walking (N) and while subjects were simulating 3 different gait strategies related to knee osteoarthritis [34, 35, 78] : right toe out (TO). flat strike of the right foot (rFF) and right trunk leaning (rTL). For each randomized condition, both participants were walking for 2 minutes at 5 different speeds (0.8, 1, 1.2, 1.4, 1.6 m/s). A 3-axis Physilog 4 accelerometer ( $\pm 8g$ ) was placed on the top of both shoes, localized as shown in figure 3.2. Ground reaction forces for each foot were recorded separately using a treadmill with two embedded force plates (Bertec, Columbus, OH, USA, 1000 Hz). All signals were synchronized using a heel impact on the treadmill.

### 3.3.1.3 UDS Database

The UDS database was recorded on 15 participants in three walking circuits in an indoor gym. The circuits were a 12 m traverse on a gym track. Three different tracks were used in the tests, as shown in figure 3.3 : the normal (T1), the obstacle (T2) and the walk around (T3) tracks. Each participant was equipped with an Adafruit MMA8451 Triple-Axis Accelerometer [ $\pm 2$  g] placed on the top of the left foot inside the shoe (figure 3.2). Force resistive sensors were placed under the left shoe, under the shoe sole. Records of 2 minutes of continuous round trip walking without stopping at the end were done for each track and each participant. The data of T3 for one participant was not taken into account due to incorrect positioning of the FSR.



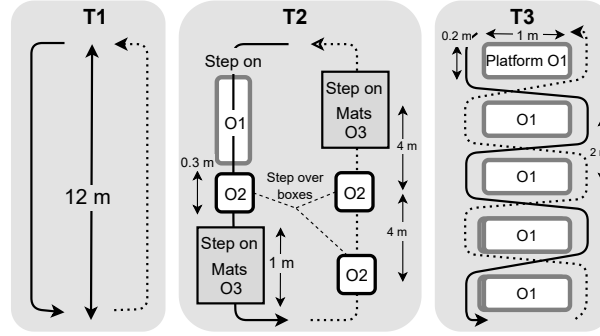
**Figure 3.2** – IMU and FSR positioning for the three databases. In CIRRIS tests the IMU was placed outside the shoes using straps. For MAREA tests, the three IMU were attached with straps at the three positions shown and the FSR were placed into the shoe soles. In UDS tests, the IMU and the FSR were placed on a sock inside the shoes. The orientations are defined as the antero-posterior (AP), the medio-lateral (ML) and the vertical (V) axis

## 3.3.2 Data Formatting and Preprocessing

### 3.3.2.1 Target Gait Events Identification

The target gait events (GE) were defined by time indices of different peaks or drops of the foot ground reaction force. These indices were identified with FSR measurements for the MAREA and UDS databases and instrumented treadmill measurements for the CIRRIS database. Six GE classes, the Heel Strike (HS), Heel Push (HP), Foot Flat (FF), Heel OFF (HO), Toe Push (TP) and Toe Off (TO) were defined in the gait cycle, as shown in figure 3.4. All GE classes were located in the stance phase, and one GE index was defined in each gait cycle.

Only some of the six gait event classes were identified for each database. For the MAREA database, only HS and TO were identified directly by the authors of the data source [39]. The HS, HP, FF, TP and TO event classes were determined for the CIRRIS database and the HS, HO, TP and TO event classes were determined for the UDS database. Processing details of the ground reaction force signal to obtain the targets gait event are given in appendix B.



**Figure 3.3** – Walking tracks of UDS walking tests. The subjects were walking continuously on the full arrow and coming back following the dotted arrow. T1 is a normal straight walk without obstacles. T2 is a walking circuit over small obstacles, stepping on a 0.3 wide by 1 m long by 0.2 m high hard platform (O1), stepping over a 0.1 m high by 0.3 m square side platforms (O2) and stepping on a 0.5 m wide by 1 m long by 4 cm tick exercise mat (O3). The participant is stepping on O1 and O3 by putting their feet on it, but has to pass over O2 without touching it. T3 is a walking circuit around 0.3 m wide by 1 m long platforms (O1) aligned in a row with 2 m distance between platforms.

### 3.3.2.2 Input and Target Timeseries Definition

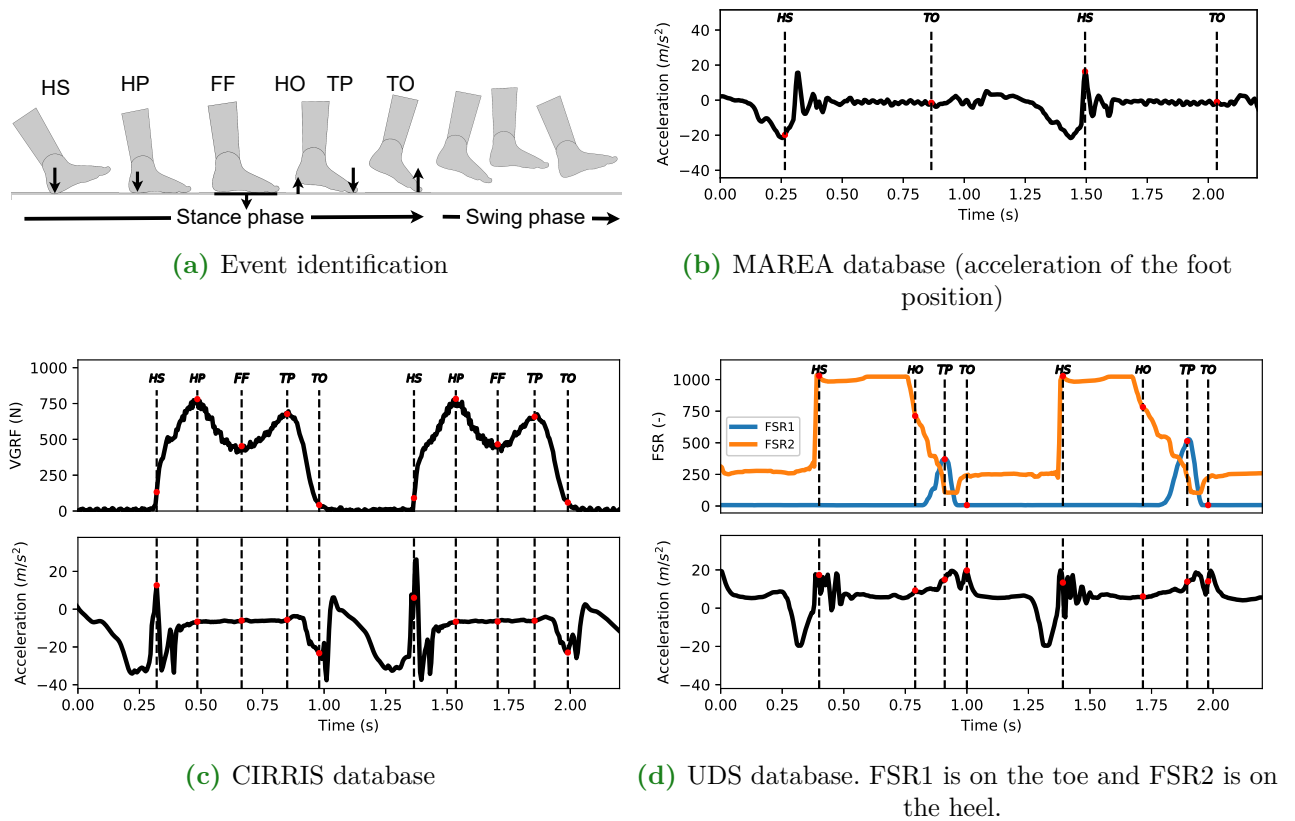
The acceleration and gait event targets were recorded separately for each subject and each walking conditions. Different data subsets grouping all the records for one or more types of walking conditions were formed as presented in table 3.1. All records in each subset were identified by the index  $k$ . For example, in the subset  $\text{MAR}_{\text{all}}$ ,  $k=1$  corresponds to the record of the first participant tested in the treadmill walking condition,  $k=2$  corresponds to the data from the second participant on the same condition, and  $k=12$  corresponds again to the first participant, but now tested in the treadmill running condition.  $\mathbf{U}_k$  corresponds to the acceleration signal of length  $T_k$  (with  $T_k$  the number of time points in the recording).  $\mathbf{U}_k$  may have only one row corresponding to one of the accelerometers. When various axes are considered,  $\mathbf{U}_k$  has multiple rows, each corresponding to an individual axis.

$\mathbf{Y}'_k$  corresponds to the gait event targets signal of length  $T_k$  and is a binary signal, as shown in figure 3.1. A value of 1 is inserted in  $\mathbf{Y}'_k$  at the time index of each target gait event identified from ground force measurements, and otherwise,  $\mathbf{Y}'_k$  is set to 0.  $\mathbf{Y}'_k$  has multiple rows, each corresponding to one of the GE classes (HS, HP, FF, HO, TP, TO). For example, in  $\text{MAR}_{\text{all}}$ , since only the HS and the TO were detected,  $\mathbf{Y}'_k$  had two rows. The first row had a value of 1 only at each HS event index identified, and the second row had a value of 1 only at each TO event index identified; other entries were 0.

### 3.3.2.3 Subset Partitioning of the Timeseries

The training and testing process in figure 3.1, was performed several times with multiple subset selections of walking conditions, as presented in the rightmost column of table 3.1. These subsets defined the two lists of input timeseries  $\{\mathbf{U}_1, \dots, \mathbf{U}_k, \dots, \mathbf{U}_{N_{\text{TS}}}\}$  and target timeseries  $\{\mathbf{Y}'_1, \dots, \mathbf{Y}'_k, \dots, \mathbf{Y}'_{N_{\text{TS}}}\}$ .  $N_{\text{TS}}$  is the total number of timeseries and varies for each subset selection. The results presented in sections 3.4.1, 3.4.3 and 3.4.5.1 are based on the subsets  $\text{MAR}_{\text{all}}$ ,  $\text{CIR}_{\text{all}}$  and  $\text{UDS}_{\text{all}}$ , where all timeseries of each database were used.





**Figure 3.4** – Event targets identification process from foot pressure measurements for the three databases. Examples of foot pressure measurements, accelerations measurements, and target gait events identified in two gait cycles for each database are shown in subfigures (a-c). The heel strike (HS) is the first contact of the foot with the ground. The heel push (HP) is the maximum load response of the foot after the HS. The foot flat (FF) is the minimum load response of the foot in the middle of the stance phase. The heel off (HO) is the end of the heel contact on the ground in the second part of the stance phase. The toe push (TP) is the second maximum load response of the foot in the second part of the support phase. The toe off (TO) is the final contact of the foot on the ground.

In section 3.4.2, we use subsets selecting few specific walking conditions to evaluate the influence of the walking condition variability on the performance of the ESN algorithm. We used also specific conditions subsets in section 3.4.5.2 to compare the ESN with the TKEO and three other algorithms from the literature.

### 3.3.2.4 Training and Testing Sets Partitioning

We partitioned the timeseries of accelerations and gait event targets in two sets by taking 70% of the total number of gait cycles for the training set, while the other were assigned to the testing set. We defined the separation index in the middle of the stance phase (MSP) using the gait event classes in each database. For the MAREA database, the MSP was defined by the mean indices between HS and TO. For the UDS and CIRRIIS databases, the MSP was defined with the HO and the FF respectively. We select the separation index at the closest MSP to 70% of the length of the whole timeseries.

We have used a training method where all records are used partially for the training and the testing. We assume that a training method where one record would be used only for the testing might lead to lower accuracy of the ESN prediction. Moreover, the 70-30% separation method could have been chosen differently in order to include a cross validation or between the duration of each signal, but was not judged essential because each record contains signal quite stationary. We eliminate the data before the first MSP in training signal to ensure that both training and testing signals started at the same place in the gait cycle (the MSP). This preprocessing ensure to removed a uniform gait cycle in each timeseries, considered as transient data in the ESN method explain the section 3.3.3.1.

## 3.3.3 ESN Procedure

### 3.3.3.1 Main ESN Structure

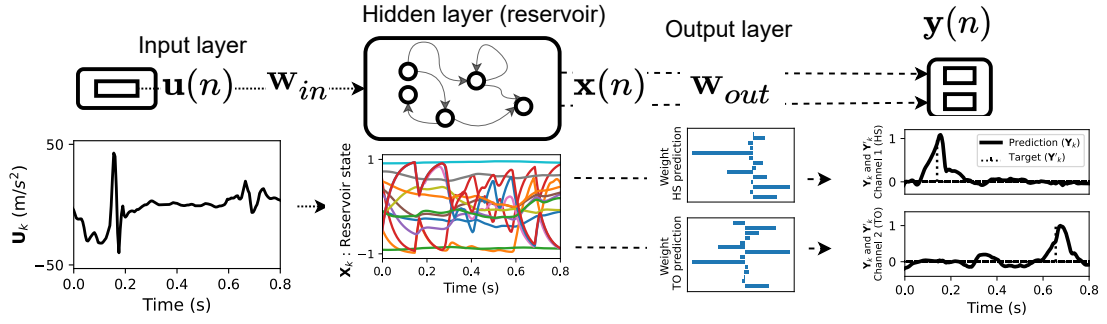
We present here the ESN algorithm presented by Lukoševičius [72] applied for GED using recorded body accelerations. The ESN is a neural network build from three layer : the input layer, the hidden layer and the output layer, as shown in figure 3.5. The hidden layer is commonly called the reservoir and is composed of  $N$  artificial neurons. The ESN processes the input accelerations  $\mathbf{u}(n)$  and returns predictions  $\mathbf{y}(n)$  for the GE according to

$$\tilde{\mathbf{x}}(n) = \tanh(\mathbf{W}_{in}\mathbf{u}(n) + \mathbf{W}\mathbf{x}(n-1)) \quad (3.1a)$$

$$\mathbf{x}(n) = \alpha\tilde{\mathbf{x}}(n) + (\alpha-1)\mathbf{x}(n-1) \quad (3.1b)$$

$$\mathbf{y}(n) = \mathbf{W}_{out}\mathbf{x}(n), \quad (3.1c)$$

where  $\mathbf{u}(n)$  is a column vector of size  $N_{in}$  and where  $N_{in}$  corresponds to the number of acceleration channels in the input augmented by 1 to add a bias value.  $n$  is the time step index.  $\mathbf{y}(n)$  is a column vector of size  $N_{out}$ .  $\mathbf{W}_{out}$  is the output matrix of size  $N_{out} \times N$ , where  $N$  is the number of neurons in the reservoir, and  $N_{out}$  is the number of gait event classes predicted. When the model is used only for HS and TO events,  $N_{out} = 2$ .  $N_{out}$



**Figure 3.5** – ESN algorithm representation considering the HS and TO gait event classes for the output prediction.

= 2, 4 and 5 respectively for the events for the MAREA, UDS and CIRRIIS databases when all event are used.  $\mathbf{x}(n)$  is a column vector of size  $N$  corresponding to the states of the individual artificial neurons in the reservoir. The variable  $\alpha$  is the leaking rate of the reservoir. It can be set between 0 and 1 and is used to adjust the response time of the ESN. The nodes connection matrix  $\mathbf{W}$  of size  $N \times N$  and the input layer matrix  $\mathbf{W}_{in}$  of size  $N \times N_{in}$  are fixed and are initialized from random uniform distributions. A distribution between  $[-1, 1]$  is used to set the elements of the matrices  $\mathbf{W}$  and  $\mathbf{W}_{in}$ . The sparsity of  $\mathbf{W}$  is adjusted by setting to zero a fraction  $1-P$  of randomly chosen elements, where  $P$  is the sparsity. The spectral radius,  $\rho$  is set by dividing all elements of  $\mathbf{W}$  by its absolute maximum eigenvalue and multiplying it by  $\rho$ . For the input layer matrix  $\mathbf{W}_{in}$ , its sparsity is adjusted by setting to zero a fraction of  $1-P_i$  of randomly chosen elements. The input bias scaling of the matrix  $\mathbf{W}_{in}$  is adjusted by multiplying its first row by  $S_b$ . The input scaling is adjusted by multiplying all rows but the first of  $\mathbf{W}_{in}$  by  $S_i$ . The parameters used to set the ESN are shown in table C.1. We set these parameters fixed for all the results. They were chosen based on an exploration of the CHARC properties [79] as presented in appendix C.

The equations 3.1a - 3.1c are iterated for  $n$  running over all the timesteps of each input timeseries  $\mathbf{U}_k = [\mathbf{u}(1), \mathbf{u}(2), \dots, \mathbf{u}(T_k)]$  of size  $N_{in} \times T_k$ , where  $T_k$  is the total number of time steps for the input timeseries  $\mathbf{U}_k$ . We forced the states to zeros at the first time step,  $\mathbf{x}(0) = \mathbf{0}$ . For the indices of the first gait cycle, the reservoir states  $\mathbf{x}(n)$  were different compared to other gait cycles since the initial value was 0. We considered the time indices of the first 1.25 seconds ( $n \leq 250$ ) as a transient phase and we eliminated them before further processing. We concatenated of the states  $\mathbf{x}(n)$  of the reservoir over all the time steps of one timeseries to form the states matrix  $\chi_k = [\mathbf{x}(251), \mathbf{x}(252), \dots, \mathbf{x}(T_k)]$  of size  $N \times (T_k - 250)$ .

### 3.3.3.2 ESN Training

The training of the ESN fixes the weights of the output matrix  $\mathbf{W}_{out}$  and determines the gait cycle duration  $\tau$ , defined as the mean duration between each HS gait events. As explained in section 3.3.2.3, the ESN processed input accelerations from multiple timeseries,  $\{\mathbf{U}_1, \dots, \mathbf{U}_k, \dots, \mathbf{U}_{N_{TS}}\}$ , each of size  $N_{in} \times T_k$ . Equations 3.1a and 3.1b are first

**Table 3.2** – Echo State Network Hyperparameters

Number of nodes	N	100
Leaking rate	$\alpha$	0.1053
Hidden layer spectral radius	$\rho$	0.7471
Hidden layer sparsity	P	.21
Input layer scaling	$S_i$	2.300
Input layer Sparsity	$P_i$	0.41
Input layer bias scaling	$S_b$	2.911
Regularization parameter	$\gamma$	1.e-6

used to compute the reservoir state matrices  $\chi_k$  for each input timeseries of the training dataset.

We defined the same number of target matrices (output signal excluding the first 1.25 seconds),  $\{\mathbf{Y}'_1, \dots, \mathbf{Y}'_k, \dots, \mathbf{Y}'_{N_{TS}}\}$ , each of size  $N_{out} \times (T_k - 250)$ . We then computed the  $\mathbf{X}\mathbf{X}^T$  and  $\mathbf{Y}'\mathbf{X}^T$  using the summation of premultiplied states matrices for each individual timeseries, as  $\mathbf{X}\mathbf{X}^T = \sum_{k=1}^{N_{TS}} \chi_k \chi_k^T$  and  $\mathbf{Y}'\mathbf{X}^T = \sum_{k=1}^{N_{TS}} \mathbf{Y}'_k \chi_k^T$ , where  $\chi_k^T$  is the transposed of the matrix  $\chi_k$ .<sup>4</sup>  $\mathbf{W}_{out}$  is computed using a ridge regression,

$$\mathbf{W}_{out} = \mathbf{Y}'\mathbf{X}^T(\mathbf{X}\mathbf{X}^T + \gamma\mathbf{I})^{-1}, \quad (3.2)$$

to minimize the mean square error between the output target events and the predicted event timeseries. The matrix  $\mathbf{I}$  ( $N \times N$ ) is the identity matrix and  $\gamma$  is a regularization parameter used to prevent overfitting. The parameter  $\gamma$  in equation 3.2 was adjusted to  $10^{-6}$  (table C.1) by trying multiple values on a logarithmic scale and keeping the one giving the best NRMSE according to equation 3.3, using the  $MAR_{all}$ ,  $CIR_{all}$  and  $UDS_{all}$  as a validation set.

Finally, the mean gait cycle duration,  $\tau$ , was established from the HS gait event target of all the timeseries in the training set. A lower range gait cycle duration for each timeseries,  $\tau_k$ , was computed using the 5th percentile of the time distances separating each HS events indicated in  $\mathbf{Y}'_k$ . The value  $\tau$  is computed as the mean of each timeseries gait cycle duration,  $\tau = \frac{1}{N_{TS}} \sum_{k=1}^{N_{TS}} \tau_k$ . The value  $\tau$  was equal to 0.96, 1.17 and 1.19 seconds per step respectively for the MAREA, CIRRIIS and UDS databases considering subsets with all the conditions selected ( $MAR_{all}$ ,  $CIR_{all}$  and  $UDS_{all}$ ). The MAREA database includes more running conditions than CIRRIIS and UDS, which explains the lower value of  $\tau$ .

### 3.3.3.3 ESN Testing

During the testing, we use the equations 3.1a, 3.1b and 3.1c with the  $\mathbf{W}_{out}$  learned in the training phase to compute a prediction output  $\mathbf{Y}_k$  from the ESN for each input timeseries of the testing dataset.  $\mathbf{Y}_k$  was a multichannels matrix of continuous signals with the same size as the gait events target matrix  $\mathbf{Y}'_k$ . Each channel in  $\mathbf{Y}_k$  was the prediction for one gait event class. We evaluate the error of the ESN was evaluated by the normalized root mean squared error (NRMSE), defined as

4. The matrix  $\mathbf{X}$  of size  $N \times T_{tot}$ , is the matrix of all  $\chi_k$  concatenated together, and  $\mathbf{Y}$ , of size  $N_{out} \times T_{tot}$ , is the matrix of all  $\mathbf{Y}'_k$  concatenated together, where  $T_{tot}$  is the total number of time steps over all the timeseries,  $T_{tot} = \sum_{k=1}^{N_{TS}} T_k$ . In practice, we are not computing these matrices  $\mathbf{X}$  and  $\mathbf{Y}$  explicitly, because if all states matrices  $\chi_k$  and  $\mathbf{Y}'_k$  are kept in memory in order to build the matrix  $\mathbf{X}$ , this quickly induces memory saturation since many timeseries are considered.

$$\text{NRMSE}_k = \sum_{i=1}^{N_{\text{out}}} \sqrt{\frac{1}{T} \frac{\sum_{n=1}^T (\mathbf{Y}_{k,i,n} - \mathbf{Y}'_{k,i,n})^2}{\sigma_{k,i}^2}}, \quad (3.3a)$$

$$\text{NRMSE} = \frac{1}{N_{\text{TS}}} \sum_{k=1}^{N_{\text{TS}}} \text{NRMSE}_k, \quad (3.3b)$$

where  $k$  is the index of the input timeseries, and  $i, n$  are the row and column indices respectively related to the event type and the time step in  $\mathbf{Y}'_{k,i,n}$  and  $\mathbf{Y}_{k,i,n}$ .  $\sigma_{k,i}$  is the standard deviation of the  $i^{\text{th}}$  column target signal  $\mathbf{Y}'_{k,i}$ .

We finally used a peak finder algorithm [80] to get the time indices of the peaks of each prediction channel signal,  $\mathbf{Y}_{k,i}$ . The time indices of the peaks in each channel corresponded to the predicted time of the gait event according to each gait event class. We used a threshold of  $0.65\tau$  to limit the minimum allowed time between two peaks from a given event class in order to minimize the misdetection of events as explained in appendix C.1.

### 3.3.4 Comparative Procedure

One of the objectives of this work was to establish a performance comparison between an ESN and other IMU GED algorithms. The Teaser Keaser Energy Operator (TKEO) method proposed by Flood et al. [5] uses an energy operator to determine the two energy peaks associated with the initial contact and the final contact of the foot on the ground, referred here as the HS and the TO gait events. This method has demonstrated good performance on the MAREA walking database for the mean absolute error metric (MAE) [5].

Like the ESN, the TKEO method has the advantage of minimizing the number of assumptions about gait (e.g. foot symmetry, regularity) to process events. In this paper, the TKEO method was implemented following the indications of [5].

The TKEO method consists in a two steps non linear filtering process executed individually on each timeseries  $\mathbf{U}_k$  of the test set. The first step computes a time serie  $\Phi_k$  of size  $T_k$  representing the amount of energy at each time points of the acceleration  $\mathbf{U}_k$ , expressed in terms of the function  $F_{TKEO1}$  [5],

$$\Phi_k = F_{TKEO1}(\mathbf{U}_k), \quad (3.4)$$

where  $\mathbf{U}_k$  is a single channel of acceleration. The HS events are detected from the indices of the peaks in  $\Phi_k$  using a peak finder algorithm [80] with a time detection threshold defined as 0.7 of the first autocorrelation peak of  $\Phi_k$  [5]. As a second step, a function  $F_{TKEO2}$  [5] computes a modified energy function  $\Psi_k$ , also of size  $T_k$ ,

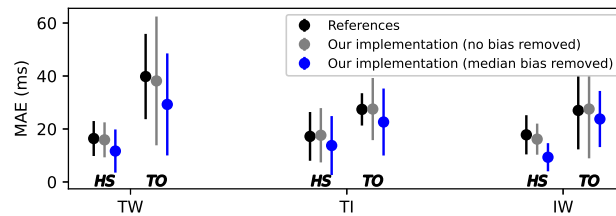
$$\Psi_k = F_{TKEO2}(\Phi_k, \{i_{HS}\}_k), \quad (3.5)$$

based on the energy  $\Phi_k$  and the HS event indices  $\{i_{HS}\}_k$ , in order to eliminate the peaks-previously detected for the HS in  $\Phi_k$  and to smooth the signal. The TO events are detected from the indices of peaks in  $\Psi_k$  using the same peak finder algorithm used for the HS events and with a time detection threshold equal to 0.7 of the first autocorrelation peak

of  $\Psi_k$ . We tested our implementation of this algorithm with the same reference datasets as in [5]. As shown in figure 3.6, there is a good agreement.

In order to compare the precision of the ESN, TKEO and other models from the literature, we removed the median bias from the errors in all results for all methods.<sup>5</sup> This reduced the MAE obtained by the TKEO compared to the value reported in the literature [5]. This hypothesis is done specifically for comparison purpose. In real world, this bias might have bad influence if it is not recentered for estimation like cadence and gait phase duration. For each algorithms, a recenter bias of each event individually should be applied on the prediction based on the median or the mean of the error of a test set used before real world application.

Small variations in the standard deviation were observed compare the values reported by Flood [5]. This could be related to our implementation in the Python programming language where the peak finder algorithm [80] was different compared to the one used in the original Matlab implementation [81]. Since the values of the MAE obtained were mostly the same compared to the reference, our implementation was considered as a valid comparisons algorithm of the TKEO reported by [5] tested here on all our databases.



**Figure 3.6** – Average (dots) and  $\pm 1$  standard deviation (error bars) of the MAE for the values reported in reference [5] and our own implementation of the TKEO algorithm, under conditions of treadmill walking (TW), treadmill incline (TI) and indoor walking (IW).

### 3.3.5 Performance Criteria

The metrics of GED performance are based on the error between the time positions of the target and the predicted events. For each timeseries  $k$  and class of event  $i$ , the vector of the error values  $\mathbf{E}_{ik}$  (each of size  $1 \times N_{TE}(i, k)$ ) was calculated by taking the time of each target event, minus the time of the closest prediction event.  $N_{TE}(i, k)$  is the number of target events in the channel of the target vector  $\mathbf{\Upsilon}'_k$ , and  $N_{PE}(i, k)$  is the number of predicted gait events detected from the GED. The two can be different for a gait event class  $i$  and timeseries  $k$ .

As proposed in other works on GED [38, 39], we removed the constant bias in the error before performance metrics assessment for each class of events, by subtracting the median of the error over all timeseries<sup>5</sup>. Using the absolute values of the vector  $\mathbf{E}_{ik}$ , we classified the target and prediction events into three detection outcomes following the rules presented in table 3.3. To consider the target events as correctly detected, a threshold of 65 ms was chosen exactly like [5]. For each class of event  $i$ , the number of True Positive

<sup>5</sup>. A median bias in the overall prediction can appear for any type of GED algorithm, can be easily removed in applications without changing the GED algorithm performance.

**Table 3.3** – Classification of the target events (TE) and the prediction events (PE) for FN, TP and FP.

<b>1. False negative (FN)</b> Target event not detected	<b>2. True positive (TP)</b> Target event correctly detected	<b>3. False Positive (FP)</b> Prediction event incorrect
TE with a time distance higher than 65 ms from the closest PE.  $N_i^{FN} = \# \{  \mathbf{E}_{ikl}  > 65 \text{ ms} \mid 1 \leq k \leq N_{TS}, 1 \leq l \leq N_{TE}(i, k) \}$	TE with a time distance equal or lower than 65 ms from the closest PE.  $N_i^{TP} = \# \{  \mathbf{E}_{ikl}  < 65 \text{ ms} \mid 1 \leq k \leq N_{TS}, 1 \leq l \leq N_{TE}(i, k) \}$	PE with a time distance with the closest TE higher than 65 ms or higher than any time distance of other PE and the TE.  $N_i^{FP} = \# \{ N_{PE}(i, k) \mid 1 \leq k \leq N_{TS} \} - N_i^{TP}$
<p><b>4. True Negative (TN)</b> All remaining points The number of remaining time points in the time series unselected as a TP, FP or FN event <math display="block">N_i^{TN} = \# \{ N_T(k) \mid 1 \leq k \leq N_{TS} \} - (N_i^{TP} + N_i^{FP} + N_i^{FN})</math> <math>N_T(k)</math> is the total number of time points in each time series</p>		

(TP), False Positive (FP) and False Negative (FN) were cumulated for all timeseries  $k$ . We express results with standard metrics of the detection rate used in GED, the true positive rate or sensitivity (TPR),

$$TPR_i = \frac{N_i^{TP}}{N_i^{TP} + N_i^{FP}} \quad (3.6)$$

and the positive prediction value or precision (PPV),

$$PPV_i = \frac{N_i^{TP}}{N_i^{TP} + N_i^{FN}}. \quad (3.7)$$

These metrics are generally anti-correlated : oversensitive algorithms will have a low TPR but a high PPV, and vice versa for a undersensitive algorithms. The  $T_1$  score represents the harmonic mean of the TPR and PPV,

$$T_1 = \frac{2 \text{TPR} \text{PPV}}{\text{TPR} + \text{PPV}}, \quad (3.8)$$

where a  $T_1$  result of 1 represents perfect detection without any FP and FN. Another metric that is used in the litterature is called the specificity (SPF),

$$SPF = \frac{N_i^{TN}}{N_i^{TN} + N_i^{FP}}, \quad (3.9)$$

and measures the oversensitivity of an algorithm. Like the TPR, the value of the SPF will be low if the algorithm is oversensitive. This metric is used to compared the ESN with other algorithms from the litterature in figure 3.14. However, the SPF is not used to express the results in the other parts of the paper because its value is influenced by the ratio between the number of gait cycles and the total number of points in the timeseries. since The walking speed and the sampling frequency will then influence this proportion and the SPF result, even if the algorithms have the same range of errors. Comparing different walking conditions or other works in the literature is then less robust.

GED algorithms performance is also measured using the mean absolute error (MAE). For each timeseries  $k$  and event class  $i$ , the mean absolute error  $MAE_{ik}$ , in seconds, is

calculated as the mean of the absolute of the error vector  $\mathbf{E}_{ik}$  (already recentered for each event by the median error over all timeseries). The  $\text{MAE}_{ik}$  results are grouped by gait event class and the distribution over all timeseries are displayed in whisker plots. The mean value is expressed as a triangle and the median as a central line, while boxes are the 25th and 75th percentiles and the whiskers are the extensions of the last value of  $\text{MAE}_{ik}$  of the distribution over and under 1.5 times the interquartile range. The values of TPR, PPV and  $T_1$  are plotted with three lines and are relative to the right-side axis.

## 3.4 Results

---

We present results in five sections. In section 3.4.1 we focus on the influence of the training data (training length, choice of the database in training versus testing) used to evaluate the performance of the ESN. In section 3.4.2, we evaluate the influence of the walking conditions on HS and TO events using the antero-posterior (AP) acceleration of the foot. In section 3.4.3, we focus on the choice of axis : AP, Medio-lateral (ML) and vertical (V), and various sensor positions (foot, wrist and waist) as the input signal of the model. In section 3.4.4, we compare the performance between all the event classes. Finally, in section 3.4.5, we compare the prediction performance of the ESN tested with other GED algorithms in the literature on the MAREA database, based on the prediction of the HS and TO events.

### 3.4.1 Training Data

---

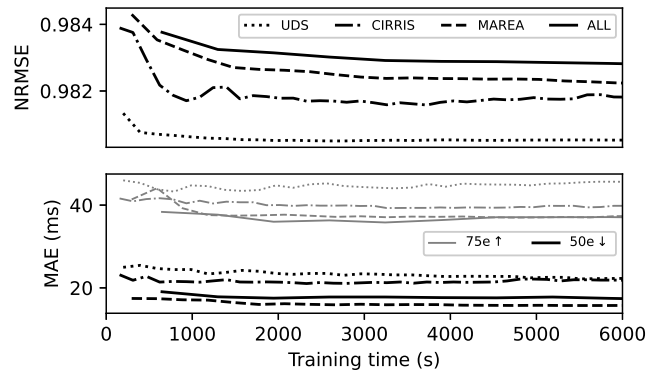
This section evaluates the influence of two aspects of the training data on the ESN performance. The NRMSE and the MAE results are presented as a function of the amount of training data in section 3.4.1.1. The MAE (mean and standard deviation) for each dataset is presented for various training and testing sets in section 3.4.1.2.

#### 3.4.1.1 Training Data Length

Figure 3.7 shows the NRMSE and MAE as a function of the length of the training data. All time series in the testing set were kept with a fixed length equal to the last 30% part of the full recorded data, as explained in section 3.3.2.4. The MAE is presented with the 50th and 75th percentiles of all the  $\text{MAE}_{ik}$  of all gait event classes  $i$  and time series  $k$ . Looking at the 75th percentile of the MAE, the performance of the ESN converges for a total training time of approximately 2000 seconds. The training for the three databases combined (ALL) did not need more time to converge than each individual database. The NRMSE for the three databases combined is  $0.980 \pm 0.002$  after convergence. Considering that the average cadence in the different databases is about 0.9 steps/s, 2000 gait cycles is a good standard to train the ESN to obtain the best precision for GED. We observed both the NRMSE and the MAE as a convergence criterion to establish the minimum training size, but the MAE metric is more representative of the ESN performance for GED.<sup>6</sup>

6. We see in figure 3.7 that the NRMSE keeps decreasing with longer training even if performance in terms of MAE stays approximately the same. In many machine learning applications, the NRMSE is a gold standard to evaluate the training convergence. Here, the NRMSE continues to decrease with training time much longer than 2000 cycles before to obtain stable value. However, the NRMSE expressed the





**Figure 3.7** – NRMSE and MAE as a function of the length of the training set (with the AP axis acceleration input) considering all events for each database. The errors were evaluated using the  $MAR_{all}$ ,  $CIR_{all}$  and  $UDS_{all}$  subsets. The three subsets were considered together in the set ALL, with the HS and TO gait event types only. The MAE and NRMSE are different for all databases, mostly because of the harder and higher number of gait event types detected in the CIRRIS and UDS databases than in the MAREA.

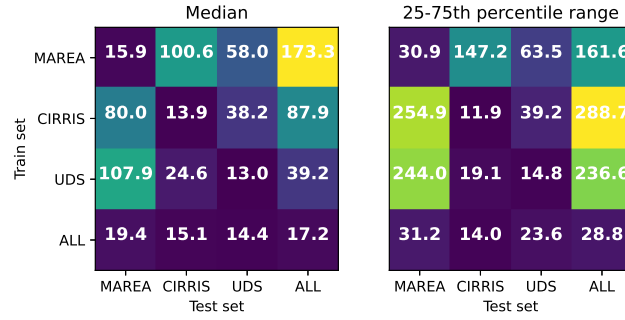
### 3.4.1.2 Training Between Datasets

We show the median and interquartile range of the MAE for different training and testing sets in figure 3.8. To allow the use of the ESN with training data time series from a different database, the predictions considered only two classes of events identified for all three databases, the HS and TO. All time series in each database were considered. The results on the diagonals of the matrices in fig. 3.8 correspond to the case where the training set and the testing sets were coming from the same database. The off-diagonal results correspond to the cases where the training and the testing sets came from different databases (the first 70% of the time series from one database was used for the training set and the last 30% of the time series from another database was used for the testing set). The first three rows correspond to the cases where only one of the three databases was used in the training and the last row corresponds to the case where all databases were used for the training.

As expected, the best results were obtained when the training and testing data came from the same population. We obtain intermediate results when the training was done using all databases. The ESN performance was then not quite as good as when the training and testing database matched exactly, but its error remained under a value of MAE of 20 ms. High errors were obtained when the training was done using a different database than the one used for testing. Datasets vary in different ways (participant, ground conditions, sensors type and position, gait velocity). The higher error found was while mixing the training and testing of the MAREA and the others two database. The MAREA was the

---

error of all points between the prediction signal and the target binary signal. The positions in time and the sharp shape of the peaks in the prediction signal can influence the NRMSE. Results show that even if the NRMSE decreased with higher training length over 2000 gait cycles of training data, the ESN did not perform better for GED in terms of MAE. The MAE calculation considers only the time position of each detected peak. Even if the noise of the prediction signal might decrease with more training data, the MAE would stay the same if the positions of these peaks stayed the same. Thus, the MAE is a better indicator of the performance than the NRMSE for GED.



**Figure 3.8** – MAE (ms) of the prediction with the ESN on HS and TO events as a function of the training / testing databases combinaison, considering all walking conditions subsets  $MAR_{all}$ ,  $CIR_{all}$  and  $UDS_{all}$ . For all group of training and testing set, the training was done with the first 70% of the records and the testing was done with the last 30 % of the records.

only database with a sensor positionned at the different place compared to the other database (ankle vs top of shoes). This implies that the change in the position of the sensor is an important factors on the drop in performance.

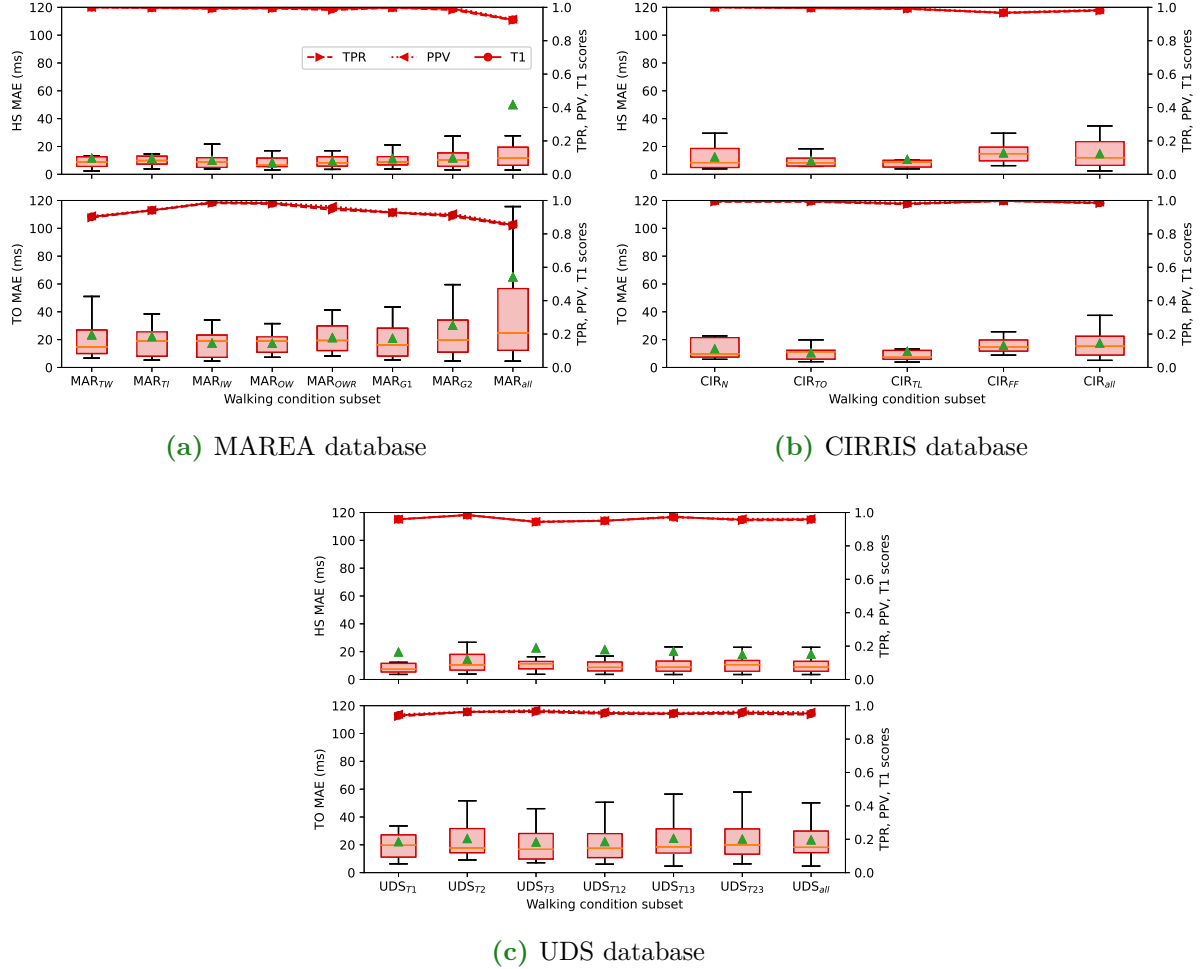
### 3.4.2 Performance Under Various Walking Conditions

Figure 3.9 shows that the ESN algorithm is robust enough to perform similarly under all of the walking conditions tested in this work. The prediction results for variable walking conditions are shown for the HS and TO gait events, common to the three databases. Each category along the horizontal axis represents the result for a specific subset of walking condition considered in the train and test sets, as presented in table 3.1.

These categories showed that the MAE distributions increased slightly when more walking conditions were included in the subsets. In the MAREA database, the prediction performance was mostly equivalent between indoor conditions ( $MAR_{TW}$ ,  $MAR_{TR}$ ,  $MAR_{IW}$ ,  $MAR_{G1}$ ) and outdoor conditions ( $MAR_{OW}$ ,  $MAR_{OR}$ ). Mixed conditions ( $MAR_{G2}$ ,  $MAR_{all}$ ) obtained higher MAE, due to the larger variability of the walking speed of the participants. The MAE distribution of the TO event class in the subsets  $CIR_{all}$  was also slightly higher compared to the MAE of other subsets in the CIRRIS database. The results in the UDS database were stable over all the subsets. Overall, the detection performance of the ESN kept a MAE under 30 ms for the HS and under 60 ms for the TO with the variability of all the walking conditions on each database.

Only the condition of indoor flat run (IR) increased the MAE distribution slightly over 110 ms, as shown in the leftmost boxplot of fig. 3.9a. The average of the MAE distribution was also significantly above its median. This was because the bias prediction errors of the majority of the events on few time series under the IR condition was much higher than for the others. Recentering the error of all the time series with a uniform bias, equal to the median error of all the time series (see section 3.3.5), causes an high error for these few time series that are badly centered according to their respective median. The resulting MAE for these few time series were much larger, increasing the mean MEA, but did not affect the rest distributions over all time series. Similar cases of higher mean value of the MAE distribution were also observed in the UDS database. In all cases we see on the

right-side axis that the values of the TPV and the PPV were very close from each other, which means that the number of FP and FN was similar. This confirms that the choice of the time threshold of  $0.65 \tau$  was adequate.



**Figure 3.9** – Mean absolute error (MAE), False positive rate (FPR), Positive predictive value (PPV) and T1 score of the ESN in function of the subsets of walking condition. TPR, PPV and T1 scores are put on the same scale (0 to 1) for all subfigures in order to compare performance between event and between databases.

### 3.4.3 Performance Between Orientations and Positions

This section presents the MAE, TPR, PPV and T1 results for all the event classes. Figures 3.10, 3.11 and 3.12 are boxplots of the MAE grouped individually by event class. Each category on the horizontal axis represents a different choice of axes for the accelerations used as the input, between the AP, ML and V orientations (as shown in figure 3.2). The three first categories were defined using one orientation of the acceleration ( $N_{in} = 2$ ). The next three categories denoted with two axes were defined using two orientations of acceleration ( $N_{in} = 3$ ), and the last category was defined with the three orientations

( $N_{in} = 4$ ). The selected acceleration axes were the same for both the training part and the testing part.

We observed variable performance of the ESN according to the sensor localisation. The results in figure 3.10a were obtained with an input defined by the acceleration recorded on the foot. For the results in figure 3.10b and 3.10c, the input was defined with the accelerations respectively recorded on the waist and the wrist, and gave significantly poorer results than the foot positions. The median MAE when the foot acceleration was used as input was between 17 and 54 ms, and the T1 score was mostly over 0.75, compared to approximately 170 and 230 ms and 0.3 when the waist or the wrist accelerations were used as the input. These results show that the ESN GED has much higher performance with foot acceleration signals than upper body accelerations.

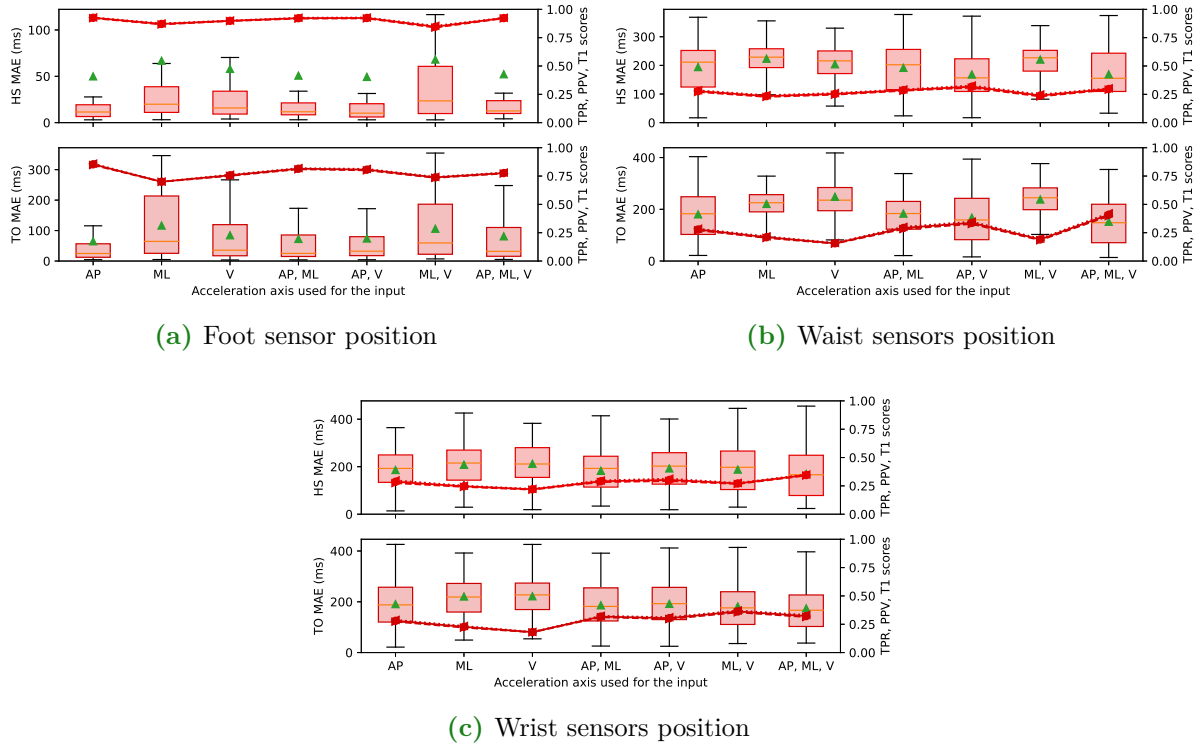
We also observed small differences of GED performance according to the choice of axes for the accelerations used as the input. The categories using two or three axes did not seem to offer a significant improvement in performance compared to the categories using a single axis. The best result was obtained using only one axis was the AP, and the input categories with several channels which included the AP axis were also more efficient.

### 3.4.4 Performance According to Various Gait Event Classes

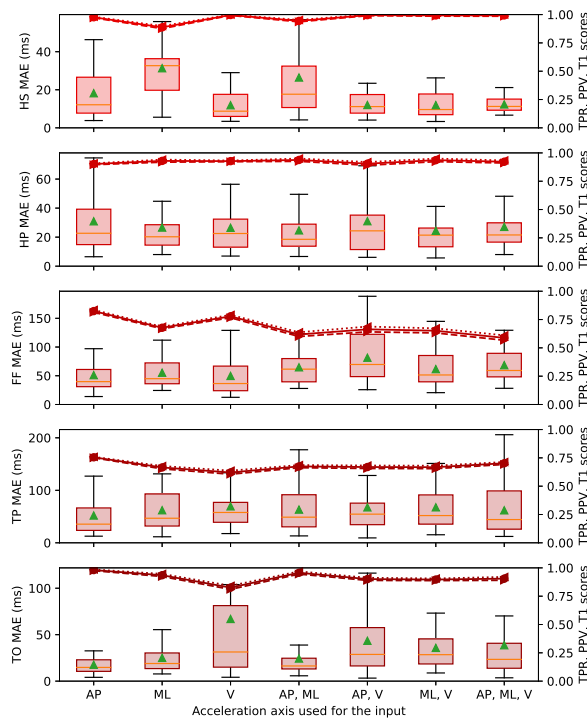
The performance of the ESN varies significantly over the different gait event classes. Observations for each gait event class are listed below : (1) The **HS event** class obtained the best predictions. The upper whisker of the box plot remained under 30 to 50 ms for the three databases and almost all acceleration axes, and the T1 scores were over 0.92. (2) The **HP event** class (in the CIRRIS database) obtain also really good prediction, i.e. with higher whisker of the MAE values of 50 ms, and T1 scores of 0.94. (3) The **FF and HO event** classes in the CIRRIS database and in the UDS database were predicted with lower accuracy. All accelerations axes for these event classes had an upper whisker of 100 ms (except for the V axis of the HO in the UDS database, which exceeded 200 ms) and a T1 score between 0.6 and 0.8. (4) The **TP event** classes was predicted with variable performance depending on the database. The MAE distributions in the CIRRIS database were quite stable with respect to the different acceleration axes and had an upper whisker of 100 ms and a T1 score of 0.9. In the UDS database, the error results were more variable, with an upper whisker ranging between 70 and 200 ms for various acceleration axes, and T1 scores of 0.75. (5) The **TO event** class prediction performance were also variable according to the database. Results for the MAREA were worst, with MAE distributions up to 120 ms and T1 scores of 0.85. The distributions were clearly better in the CIRRIS database, where the MAE distribution was under 50 ms and the T1 score over 0.95. The results in the UDS database were almost as good as for CIRRIS with a upper whisker of the distribution of 60 ms and a T1 score of 0.93. The higher variability of walking condition and timeseries duration in the MAREA. This might induce training with overfitting where longer time series, having much higher amount of data in the trainingset then small timeseries, get more importance in the ridge regression and get unbalanced prediction performance between timeseries.

In summary, the event classes at the start of the stance phase, the HS and the HP, were better predicted, with a MAE below 50 ms. The event classes in the middle of the stance phase, the FF and the HO, were predicted with less precision, with a MAE of up

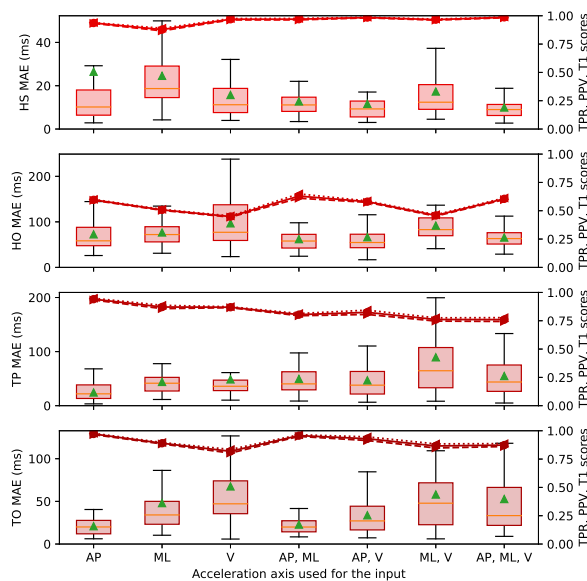
to 100 ms. The gait events in the end of the stance phase (i.e. TP and TO) were predicted with high variability, the MAE varied from values just below 50 ms to values up to 120 ms.



**Figure 3.10** – MAREA database results of MAE, TPV, PPV and T1 performance of the ESN for GED on subset  $MAR_{all}$ . The boxplots are related to the left axis and represent the MAE distribution over all the time series of the testing set. The lines, relative to the right axis, represent the value of the TPV, PPV and T1 score. The horizontal axis categories are relative to different input axis combinations. Each vertically aligned subplot represents the results for one event class. The three subfigures represent the result from different IMU positions.



**Figure 3.11** – Same as figure 3.10 for the CIRRIS database. The ESN for GED on subset  $CIR_{all}$  with foot IMU position.



**Figure 3.12** – Same as figure 3.10 for the UDS database. The ESN for GED on subset  $UDS_{all}$  with foot IMU position.

### 3.4.5 Algorithms Comparison

We compared the ESN with four algorithms from the litterature. We first compare it with our implementation of the TKEO over the three different databases. The TKEO algorithm had similar characteristics as ESN : It required a low number of assumptions and

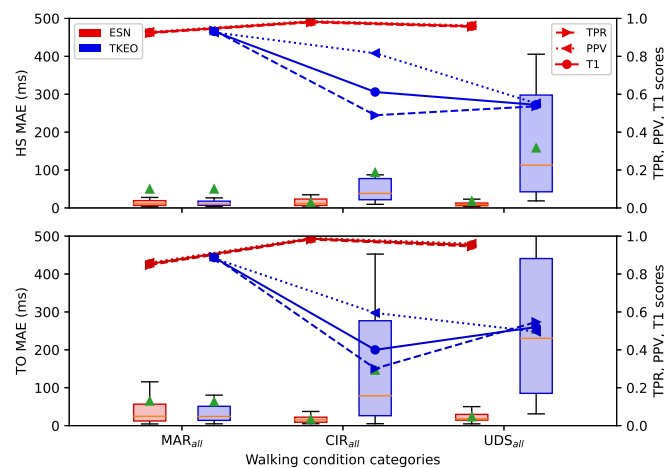
only a few calculation steps were needed to compute the HS and TO event predictions. A second comparison was done with three other algorithms that required complex calculation steps and gave higher level of precision compared to most of the results seen in litterature. The results of these three algorithms were taken from [39] obtain with three walking conditions of the MAREA database .

### 3.4.5.1 Comparison with TKEO

Figure 3.13 shows the distribution of MAE for the three databases, using input accelerations from the AP axis of the foot, as it was proposed in the original TKEO method [5]. This allows to compare how the two algorithms performed in each database and with all the walking conditions considered, with the ESN trained specifically on each database.

In the MAREA database, the ESN and TKEO algorithms performed with in a similar way. For both algorithms, the MAE distributions was under 30 ms for the HS events and mostly under 100 ms for the TO events. A T1 score over 0.92 was obtained for the HS and 0.85 for the TO. On the two other databases, the TKEO algorithm was highly inefficient compared to the ESN. The TKEO had a MAE of up to 500 ms and a T1 score around 0.6, compared to 50 ms and around 0.94 for the ESN. The MAE of the TKEO in the UDS and CIRRIS databases represented an error equivalent to half of the walking cycle.

Figure D.1 provides more comparatives results of the two algorithms with the other walking conditions subsets presented in table 3.1. The ESN algorithm thus showed robust prediction capabilities over all databases compared to TKEO. Figure D.1 shows similar comparative results for the two algorithms when specific walking conditions are considered in each of the databases.



**Figure 3.13** – Comparison of MEA, FPR, PPV and T1 score on the three databases for the ESN and the TKEO algorithms.

### 3.4.5.2 Comparison with other Algorithms from the Literature

Figure 3.14 shows results taken from the literature [39] for three other algorithms and their comparison with the performance of the TKEO and ESN algorithms. The same walking conditions of the MAREA database were used (the subsets  $MAR_{TW}$ ,  $MAR_{OW}$ ,

MAR<sub>OWR</sub> of table 3.1). The results for the event classes HS and TO were presented with different walking conditions with the MAE<sub>*i*</sub> distribution and by the TPR and SPF values. The MAE<sub>*ik*</sub> distributions were described by their mean and their standard deviation.

In order to have a fair comparison, the results of the MAE given for the TKEO and ESN were evaluated with the same way as other results from the literature [39], using the error of only correctly detected events with a threshold of 40 ms. Threshold is an arbitrary choice to estimate the performance and is not standardized in the literature. The rest of the article considered a chosen threshold of 65 ms to get result uniformly as defined by the TKEO algorithm [5], this choice is, however, arbitrary and could have been changed to another threshold value.

The Kh and Ru algorithms were more precise and less variable than the other algorithms. In the Kh algorithm, the detection was almost perfect with a TP threshold of 40 ms and a TPR and SPF always around 0.99. For the Ru algorithm, the detection was also almost perfect in the IW and OW, but a lower accuracy was obtained in the OWR, where running conditions were considered. The Au algorithm was oversensitive for TO events, considering a low SPF value in the three subsets, meaning that a lot of false positive TO events were detected.

The TKEO and the ESN implementations obtained equivalent results. The MAE distributions of the TKEO and the ESN were mostly higher than the three other algorithms (here on the correctly detected events, with error under 40 ms only). The maximal difference of the mean MAE was about 10 ms larger in the case of the TO in IW subset. They obtained poorer detection performance than the Kh and Ru algorithms in the IW and OW, but they kept approximately the same results in the OWR, contrarily to Kh and Ru, showing that they were less influenced by the walking speed variations. The detection rate of the ESN and the TKEO was always acceptable (0.75 of TPR in the worst case) for all conditions, contrary to the Au and the Ru algorithms that show oversensitivity for TO event in the subset MAR<sub>OWR</sub>.

## 3.5 Discussion

---

We discuss here about five important aspects of the application of the ESN for GED : A) Training characteristics of the ESN have been found to be critical; B) The ESN robustness over various aspects of input data; C) The ESN robustness over different types of gait event predictions; D) Simplicity of the ESN compared to other state-of-the-art algorithms; E) The clinical perspectives and some further investigations needed of the ESN algorithm.

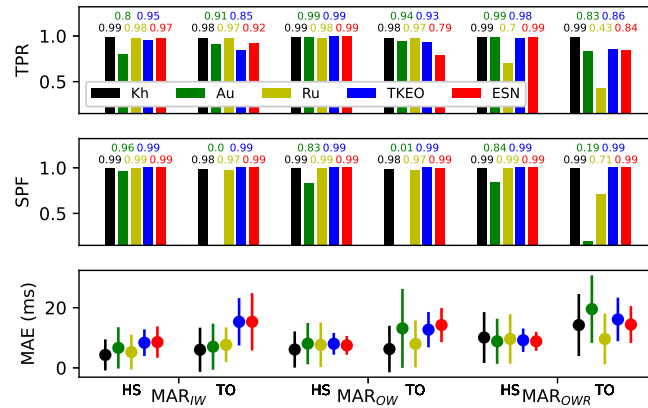
### 3.5.1 Training Characteristics to ensure good performance

---

#### 3.5.1.1 Minimum Training Duration

A total of 2000 gait cycles was a good standard to obtain the smallest MAE over the three different databases containing at least 40 (for the case of the CIR<sub>all</sub>) different walking conditions and participants, as shown in figure 3.7. This means that each participant needs to perform about 50 steps (less than 1 min) of continuous walking in each expected conditions to ensure stable performance. If more variabilities was considered in the training





**Figure 3.14** – Comparison of the TKEO and ESN algorithms to other algorithms in terms of the TPR, the SPF and the mean (central point) and  $\pm 1$  standard deviation (errorbar) of the MAE. The subsets  $MAR_{IW}$ ,  $MAR_{OW}$  and  $MAR_{OWR}$  of the the MAREA database were used, as for the results of the Kh, Au and Ru algorithms from [39]. The performance criteria used here for TKEO and ESN were adjusted to match the criteria used for Kh, Au and Ru. A threshold of 40 ms was used to consider the events as TP. The MAE was evaluated with the error from the TP events only. The number of points in each time series was considered for a sampling rate of 128 Hz.

data, from other pathologic gaits or with new walking environments, this minimal training duration might need to be increased.

### 3.5.1.2 Training for Specific vs General Applications

Multiple walking conditions, sensor locations and saturation characteristics induce many differences in the amplitude and the signal shapes over the three databases. These differences can significantly increase the prediction error for the ESN. Changing gait data conditions between the training and test sets was the main issue leading to decreased performance. With specific training data, the ESN precision is maximized for similar testing data. However, we observed in figure 3.8, that the mean absolute error can increase by over 10 times if a condition unseen in the training is used later in testing. Such results allow to conclude that a training made with healthy population will not be efficient after on population with different speed or with OA or any pathology not considered in the training population.

We also observed, in the last row in the matrices of figure 3.8, that the ESN was able to generalize event predictions over all datasets when the training set considered all databases. Generalization, in this application, indicates the possibility to maintain the ESN accuracy for GED even with datasets with high variabilities and different sources. We can see that high variability in the training data, also increase a little bit the MAE. In other words the generalization capability of the ESN is obtained with a small tradeoff in terms of precision, but help to get stable prediction on a higher variability of type of population.

This work showed that both general and specific training data could be advantageous to ensure good GED performance, depending on the application and the training data

availability. The general training data need to be sufficiently variable, under the multiple conditions that can appear in applications, and could ensure the robustness of the ESN under all of these conditions. The specific training data should be sufficiently close to one condition of use (same user, device, body position and walking conditions), and could ensure higher precision for this specific condition.

## 3.5.2 Robustness over Various Types of Input Data

### 3.5.2.1 Walking Conditions

Figure 3.9 showed that ESN obtained good predictions over all the walking conditions tested for the HS and TO events. The HS event was predicted with equivalent MAE over all conditions and database. For the TO event, the performance obtained was more variable, as presented in section 3.3.1, and appears to be related to the variability of acceleration profiles during TO events. A peak in the acceleration signal is visible in each gait cycle happening at the TO event, but it takes various forms depending on the database, patient and walking condition. We can see such differences in figure 3.4, where the shape of the acceleration signal at the end of the stance phase is different between the UDS, CIRRIIS and the MAREA databases. Increasing these variabilities in the shapes seemed to affect the consistency of the prediction performance, because, as shown in figure 3.9, the performance of the TO event detection clearly degrades with the amount of variability in the walking conditions in the three databases. In the MAREA database, the time series had 11 subjects with 5 walking conditions and 9 subjects with 2 outdoor walking conditions. Moreover, large variability of walking speeds are considered, from slow walking to running, that affect variability of the walking. Most of the time, slower (than normal) walk increase variability of each gait duration. In the CIRRIIS database, only two subjects performed 4 simulated walking patterns at five continuous and regular speeds on a treadmill. In the UDS database, all 14 subjects walked on 3 different tracks with a self-selected speed level. We can therefore qualitatively see that the variability of walking conditions seems to be smaller in the CIRRIIS database, larger for the UDS database and largest for the MAREA database. The performance of the TO predictions (Fig. 3.10a, 3.11, 3.12) followed this level of variability of gait conditions and speed (CIRRIIS performs best, followed by UDS, then MAREA).

### 3.5.2.2 Sensor Position

The input defined by the acceleration of the foot is more relevant for GED than upper body accelerations, as shown in figures 3.10b-3.10c. As discussed in the literature [36], the acceleration of the upper body is more disturbed by parasitic movements uncorrelated with the gait cycle. The prediction of a walking event with a wrist or a waist sensor represents a more complex task. We have seen similar observation in the literature [5] that highlighted the difference in prediction performance between algorithms exploiting upper body accelerations and those exploiting lower body accelerations.

### 3.5.2.3 Sensor Orientation

The orientation of accelerometer also influenced the ESN performance. A uniaxial acceleration along the AP axis located on the foot was the best choice to optimize the precision over all types of events and for the three databases. Defining input with additional axes obtain equivalent error, and did not seem to add significant information extractible for the RC. The AP acceleration is usually the primary acceleration considered in other types of GED algorithms [5, 23] and can be considered as highly relevant signal for the gait cycle. Nevertheless, the ESN is robust enough to work for GED with multiple axis orientations and combinations because the MAE results remained the order for all the different axes configurations and combinations of inputs tested. The norm would be another possibility that offer good performance, but was not considered in this work because it is not possible to do so for the RC-MEMS system.

### 3.5.2.4 ESN Adaptability Compared to TKEO

Figure 3.13 shows that the ESN was much more robust than the TKEO algorithm over all the walking conditions included in the three databases. We observed equivalent predictions from the ESN and the TKEO algorithms on the MAREA database, but on the UDS and the CIRRIIS databases, only the ESN was able to make reliable predictions. The lack of robustness of the TKEO is explained by the way the algorithm distinguishes the HS from the TO events. As explain in detail in the appendix D.1, the prediction of the TKEO depends on the assumption that the acceleration signal has a higher peak at the heel strike than at the toe-out. This assumption is true for the MAREA database (the sensor was placed at the ankle), but not in the two other databases (the sensors were placed on the middle top of the foot). In comparison, the ESN predicts each type of event independently without being affected by the shape of the signal or by misdetection dependencies between events. It is thus easier to adapt for various types of acceleration signals.

## 3.5.3 Robustness for Multi-events Detection

---

### 3.5.3.1 Capability Comparison with other Algorithms

The ESN also offers the distinct advantage of being trainable for a high number of events which is not the case for most of the other methods. The Kh algorithm [39] and the TKEO [5] are limited to the detection of the HS and TO only. These event type are the most important in practice, used to identify the swing phase and the stand phase, but other clinical purpose can benefit from the detection of higher number of event types.

The Ru algorithm [27] was created for real time fonctionnal electrical stimulation (FES) applications, where the distinction of four gait phases is needed. This method uses a rule-based state machine to detect a higher number of events inside the gait cycle. Transition criteria between events that follow the interrelated logic of the gait cycle are used to establish four different events and avoiding misdetection between the events. The results of [27] showed that the HS is precisely detected, but midstance, pre-swing and swing transistion phases, similar to the FF, HO and TO events are harder to detect, with errors extending respectively around 50, 100 and 150 ms of interquartile range. Results

obtained with the ESN were as good as Ru or better, as shown in figure 3.10, 3.11 and 3.12, where the distribution range of all error remaining between 50 and 120 ms for all the events. This range of error (5-10 % of the gait cycle) is less than the error (13 % of the gait cycle) tolerated for FES as proposed in previous study [82]. This shows the capability of the ESN to detect a high number of events with the required precision for applications such as FES.

An other limitation mentioned in [27], is related to the thresholds and transition delay criteria needed to detect preswing events (similar to the HO). These are set empirically with offline processing, and could become invalid if variations occurred in the speed of walking, which is the particularity in real world conditions. Additional work to establish rules for adaptation of the model could be avoided by the systematic training-based method of the ESN.

### 3.5.3.2 Expected Performance According to Event Classes

We demonstrated that the ESN can predict up to six gait events class (HS, HP, FF, HO, TP and TO), but with different performance for each of them, as discussed in Section 3.4.3. The maximal value of the MAE over for all walking conditions is between 30 ms for the easiest event to detect (HS) and 120 ms for the hardest events (FF, HO). We looked at the shape of the walking acceleration signals, as illustrated in figure 3.4 to interpret this dispersion in the predictive performance. Gait events that happen at the beginning of the stance phase (HS and HP), are much more precisely detected. These events occur when the foot encounters the ground. The resulting impact creates the largest peak in acceleration through the walking cycle. These events are then easily distinguishable by selecting the highest acceleration peak of the gait cycle. Events happening in the middle of the stand phase (FF and HO) were less precisely detected. They appears at the center of the stance phase when the foot is static on the ground and no features in the acceleration can be captured. Their prediction are therefore a more complex challenge for an ESN, as we observed in our results.

We found that TO event, occurring at the end of the gait cycle, obtain variable performance. As discussed in section 3.5.2.1 , the foot movement is important during this part of the gait cycle, creating a peak in the acceleration, like the HS. However, this peak is more variable in shape and in amplitude depending of the walking condition and database inducing variability in the performance.

Finally, we observed two different level of precision in the result of the event TP event for the to database tested. A lower error was obtain for the CIRRIIS database (60 ms MAE) and a higher one was obtain for the UDS database (120 ms MAE). We interpreted that this difference might be causeby the difference of the ground true defined y for each database. The ground true of TP events in the CIRRIIS database was defined by the second peak of ground reaction force sensed by the treadmill. The ground true of TP in the UDS database was defined by the peaks of FSR1 measurements positioned under the toe surface, as shown in figure 3.2. These two peaks did not occur exactly in the same part of the stance phase, as shown in figure 3.4d. In the CIRRIIS database, the TP occured in the middle of the stance phase, when the foot was still immobile on the ground and the acceleration was still zero, like the FF event class. This event was predicted with similar performance as the FF event class. For the UDS database, the TP occured later in the stand phase, at the start of the swing movement, when the acceleration

signal was non-zero, and its prediction was much better rather like the TO. According to these interpretations, we can conclude that the events closest to prominent features of the accelerations are easier to predict. The events located in the gait cycle where the acceleration signal is low or more variable from one cycle to another are more difficult to detect.

### 3.5.4 Simplicity Compared to other Methods

---

One of the main advantages of the ESN is its simplicity without significant compromise on accuracy. The three other algorithms presented in figure 3.14 were based on multi-steps filtering and data processing methods. The Kh algorithm from Khandewal [39] uses wavelet transforms to individually detect the nominal cadences, the HS and the TO walking events within separate parts of the time-frequency domain. The search space of event peaks is continuously adjusted according to the nominal cadence, giving robustness and precision for high cadence variability. Small delays are necessary to apply a Gaussian fit over a complete gait cycle to identify peaks without noise perturbations. The Au algorithm [66] exploits a process based on a continuous wavelet transform followed by a locality preserving projection [83] to reduce data dimensionality. The detection of the events is made by a training methodology using a probabilistic classification based on a gaussian mixture model. The Ru algorithm [27] is based on a multi-step filtering process to locate peaks and a rule-based state machine to maintain the logical order of detection between events. The Ru algorithm is developed by iterative empirical process trials.

For the ESN model, most of the computation complexity is done preliminarily to optimize the hyperparameters (annexe C) and to formatting the training data (annexe B) of the ESN. The calculation step of the ESN is much simpler in operation mode, contrary to the other methods. Once the training is done, only the equations 3.1a, 3.1b and 3.1c and a peak finder algorithm is needed. The integration of this algorithm into portable hardware systems is therefore easier.

Under normal walking conditions indoor and outdoor, the ESN is less performant in mean and standard deviation of the MAE compared to these 3 other algorithms. However, the ESN seems to be more robust over all conditions since it does not have any oversensitivity problem for the TO detection as the Au and Ru algorithms does. The ESN also give similar errors even for the conditions with more variability (OWR) contrary to three others that show higher mean result

### 3.5.5 Clinical Perspectives and Further investigations

---

The ESN algorithm has several practical advantages for its use in clinical applications. The ESN can process GED in real time, with low temporal delays and the number event classes predicted does not increase this delays. The ESN also has the advantage to adapt to each participant. Gait variability is a complex topic that can be related to multiple independent metrics [84]. Even for healthy walkers, many people have a unique gait pattern [85]. Gait variability and stride regularity can be affected by age and walking speed of the subject [26]. Since the training of the ESN is a quick and systematic procedure, it could therefore be useful to individualize the training by calculating the output weights  $\mathbf{W}_{\text{out}}$  specifically to each participant, using data obtained during a clinical phase prior to

the use of the ESN GED system. This allows to improve the ESN prediction efficiency for participants who have a very specific walking pattern.

This leads to further investigations needed to test such algorithm in more various type of gait. The results presented here were obtained from healthy participants only. GED performances are yet to be evaluated on patients with various conditions affecting their gait. We plan to apply the ESN for GED in a pathological population. As an example, individuals with knee osteoarthritis present heterogeneous and altered gait patterns that could influence the performance of the ESN model [37, 86, 87]. The comparison of the performance between healthy and pathologic groups of participants will allow to further evaluate the ESN capability and robustness.

Lastly, better positioning of the sensors could also improve the ESN capability for GED. The positioning of the sensor has been tested for only two positions on the foot and two positions on the upper body, but prediction performance elsewhere on the body remains unknown. For usability concern, position at the ankle should also be tested since it is easier to position than at the top of shoes.

### 3.6 Conclusion

---

To conclude, this work shows that the ESN is an efficient and robust algorithm to detect gait events using the acceleration signal measured by a single-axis IMU system. The ESN can be used for various types of gait events simultaneously and get detection with acceptable errors over various walking conditions. The ESN had a MAE at most 10 ms higher compared to four state-of-the-art algorithms on indoor and outdoor walking conditions. Unlike these algorithms, the application of the ESN detector, after training, requires limited computing resources and embedded memory. The ESN is thus easily adaptable for hardware implementations on small microcontrollers, to be used in real time clinical applications, such as FES or cadence monitoring systems. This confirms the applicability of the ESN for GED and even other gait analysis computations that need to be done in real time in wearable (embedded) systems. This also opens the door to the applications of new relevant technology such as the neuromorphic MEMS accelerometers [3, 55] designed to capture one axis acceleration signal and process it using a physical RC micro-mechanical system. Compared to ESN, the neuromorphic MEMS have the advantages of significantly increased autonomy and decreased size in wearable gait analysis devices. Like ESN, the neuromorphic MEMS are further expected to provide more robust, trainable GED capabilities compared to other popular algorithms.

# Chapitre 4

## Classification de patrons de marche

### Éléments clé

Ce chapitre démontre la classification de patrons de marche (CPDM) avec un RC.

- On tente de faire la classification entre cinq patrons de marche considérés mutuellement exclusifs, enregistrés de manière indépendante. Des séquences de données continues d'entraînement et de test ont été régénérées en post traitement avec les données expérimentales et sont utilisées avec le RC.
- Les méthodes de classification de la littérature sont principalement basées sur des corrélations statistiques ou des modèles abstraits de type AI. Les modèles AI sont plus performants que les modèles physiques pour une grande quantité de données brutes disponibles, et font maintenant partie intégrante du domaine de la reconnaissance des patrons de marche.
- Différentes applications peuvent profiter d'une CPDM, notamment la détection de pathologies, l'établissement d'une biométrie non-invasive et l'assistance au processus de réentraînement à la marche.
- Le nombre de points utilisés en post-traitement pour moyenniser la sortie de RC est de l'ordre de 375 (1.875 secondes) lorsque la modification des patrons de marche se fait à des intervalles de chaque 10 cycles. Le nombre de points est proportionnel à la longueur des intervalles entre les changements, plus de points d'intégration sont nécessaires pour obtenir des performances optimales lorsque les intervalles sont plus longs.
- En considérant le cas où le RC est entraîné et appliqué sur un seul patient à la fois (réaliste dans un contexte clinique), les résultats donnent une précision de classification de l'ordre de 82 % pour toutes les classes avec une vitesse uniforme et un taux de détection de 62 % lorsque toutes les vitesses sont mélangées.
- L'augmentation de la durée des intervalles de cycles avec un patron de marche de même classe permet d'améliorer les performances de classification, passant d'un AUC de 0.8 pour des intervalles de 5 cycles à 0.86 pour des intervalles de 50 cycles.
- La performance du RC pour la CPDM dans des conditions spécifiques (un seul patient et une seule vitesse), de l'ordre de 0.936 d'AUC, est suffisante pour envisager une application de rééducation de la marche.

## 4.1 Introduction

---

### 4.1.1 Contexte d'application clinique

---

La classification de patrons de marche (CPDM) consiste à détecter et classer plusieurs postures et mouvements reliés à la marche. La grande variabilité des patrons de marche (PDM) de chaque humain apporte toutefois un défi important à cette tâche. Les mouvements de la marche peuvent grandement varier notamment à travers l'âge des individus, la vitesse, le milieu écologique ainsi qu'avec l'apparition d'une blessure et d'une maladie [1, 26]. Plusieurs métriques pour caractériser cette variabilité [84, 88] existent et sont associées à différentes parties du corps. Mais le mouvement de la marche des humains possède un caractère si singulier pour chacun qu'il a même orienté certains travaux de recherche [89] sur l'utilisation des PDM comme biométriques d'identité, au même titre que les empreintes digitales ou l'empreinte cornéenne.

Malgré cette complexité, une CPDM adéquate peut permettre d'évaluer une condition physique spécifique d'une personne (fatigue, blessure, inflammation) et d'améliorer les traitements préventifs de maladies dégénératives. Tel que discuté dans le chapitre 1, plusieurs stratégies sont établies dans la littérature afin de limiter les contraintes dans les articulations du genou, comme la modification de l'angle de progression du pied (pointe de pied vers l'extérieur ou vers l'intérieur) [34, 35, 90] et l'inclinaison du tronc [78]. Dans ce processus clinique pour réduire la progression d'une condition comme l'arthrose, on souhaite corriger la démarche d'un patient lorsque des mouvements indésirables apparaissent hors de la démarche prescrite par le clinicien.

L'utilisation d'un algorithme RC pour détecter ce type de variation du mouvement de marche serait très utile. Elle permet une détection et une rétroaction en temps réel (avec un certain délai associé au moyennage de la prédiction, de l'ordre de 2 secondes), intégrable dans un capteur portable. Bien que certaines technologies de rétroaction sont actuellement appliquées pour améliorer la technique de course [60, 91] ou pour contrôler l'angle de progression du pied [92–94], aucune semble pouvoir s'adapter sur plusieurs PDM différents.

La section 4.1.2 introduit quelques méthodes fréquemment utilisées dans la littérature permettant de faire de la CPDM, et la section 4.1.3 introduit les perceptives les objectifs de l'utilisation du RC pour la CPDM.

### 4.1.2 Revue de la littérature sur la CPDM

---

Parmi les différentes méthodes identifiées dans la littérature afin de classifier des données de marche, deux approches sont généralement utilisées.

La première approche tente d'identifier différentes conditions du patient avec des corrélations (généralement linéaires) entre des biométriques de la marche. La seconde approche tente de faire une classification sur de grands ensembles de données en utilisant des méthodes abstraites provenant de différentes méthodes issues de l'AI.

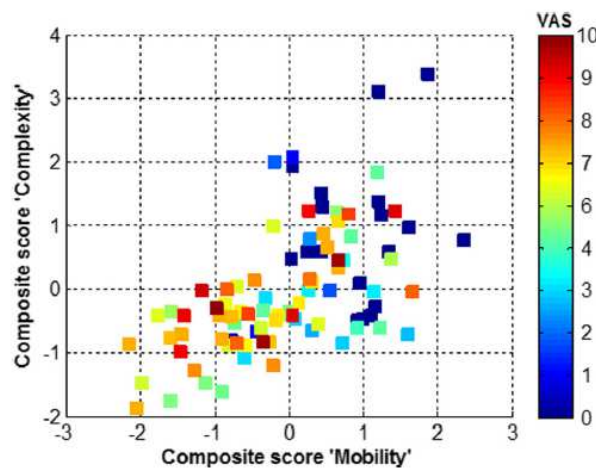
#### 4.1.2.1 Méthode d'analyse par corrélation de biométriques

Plusieurs travaux tentent d'identifier des corrélations entre une pathologie et des biométriques associées au mouvement de la marche. Les biométriques peuvent par exemple



être associées à des paramètres temporels (cadence et durée des phases du cycle de marche [17, 95]), spatiaux (hauteur et longueur de foulée, angle du pied [33, 59]) ou à la variabilité du mouvement de la marche (stabilité dynamique [96], régularité, coefficient de variation [26] de signaux inertiels de marche).

Une recherche [96] a notamment démontré qu'une mesure de la divergence cinématique du pied sur de multiples cycles est un indicateur du risque de chute d'une personne âgée. Leurs résultats montrent une sensibilité (taux d'identification de personne à risque) de 72% et une précision (taux d'identification de personne saine) de 90%, ce qui laisse place à amélioration. Une autre étude [6] faite avec des capteurs inertiels installés durant 5 x 8h sur des patients victimes de douleur chronique a démontré la dépendance entre le faible niveau de complexité (entropie du signal [97]) et de mobilité (ratio entre le temps actif et passif du patient) d'une personne souffrant de douleur chronique par rapport à des personnes en bonnes conditions (figure 4.1). Malgré la quantité de données cliniques importantes, la corrélation reste visiblement peu robuste.



**Figure 4.1** – Corrélation entre les deux métriques (*Complexity* et *Mobility*) établies avec l'accélération enregistrée avec un capteur inertielle sur l'humain et qui sont corrélées au niveau de douleur (VAS) établie pour chaque patient sur une échelle de 0 à 10 [6]

Une étude [95] a observé qu'une faible augmentation de la phase d'appui dans le cycle de marche (figure 1.1) est un indicateur de la présence d'une condition d'arthrite de la cheville, passant de  $0.6 \pm 0.06$  seconde pour des patients normaux à  $0.68 \pm 0.04$  seconde pour des patients avec la condition d'arthrose, ce qui est une variation relativement petite, mais néanmoins significative.

Une autre étude [59] s'est intéressée aux paramètres cinématiques (vitesse, largeur et distance de foulé du pied) pour étudier des patients avec le syndrome de Parkinson (PD). Cette recherche a identifié une diminution dans la vitesse de foulée, soit  $1.38 \pm 0.1$ ,  $1.32 \pm 0.08$  et  $1.05 \pm 0.3$  m/s respectivement pour des participants sains, PD avec médication et PD sans médication. Elle a aussi indiqué une diminution de la largeur de foulée, soit  $5 \pm 2$  cm pour des PD avec médication versus  $3.5 \pm 2$  cm pour des PD sans médication. Les corrélations restent relativement faibles pour la majorité des autres métriques cinématiques.

Ces quelques exemples permettent de voir qu'il est relativement difficile et non gé-

nérialisable de faire une bonne identification d'une pathologie particulière basée sur des biométriques de la marche. Ces classifications sont basées sur des corrélations empiriques permettant de faire des classifications acceptables. Ces travaux sont menés pour plusieurs conditions physiques différentes et avec un nombre restreint de biométriques pertinentes et un modèle de corrélation spécifique doit être établi pour chaque type de pathologie.

#### 4.1.2.2 Méthode d'analyse abstraite / AI

Les méthodes abstraites de type AI prennent maintenant une place prépondérante dans la reconnaissance des PDM par capteur inertiel, notamment parce qu'une grande quantité de données est accessible avec les capteurs IMU intégrés dans l'ensemble de nos téléphones [89]. Plusieurs travaux ont aussi permis l'établissement de bases de données de marche déjà labélisées selon différentes conditions, notamment pour établir un diagnostic de l'arthrose [98]. L'une des méthodes très utilisées dans la littérature est le *dynamic time wrapping* (DTW), basé sur le niveau de similarité entre différents signaux temporels, généralement de longueurs uniformes [89], mais dont la vitesse est variable. En Machine Learning sans apprentissage, le DTW peut notamment être utilisé comme métrique de distance dans les algorithmes des K plus proches voisins [99–101].

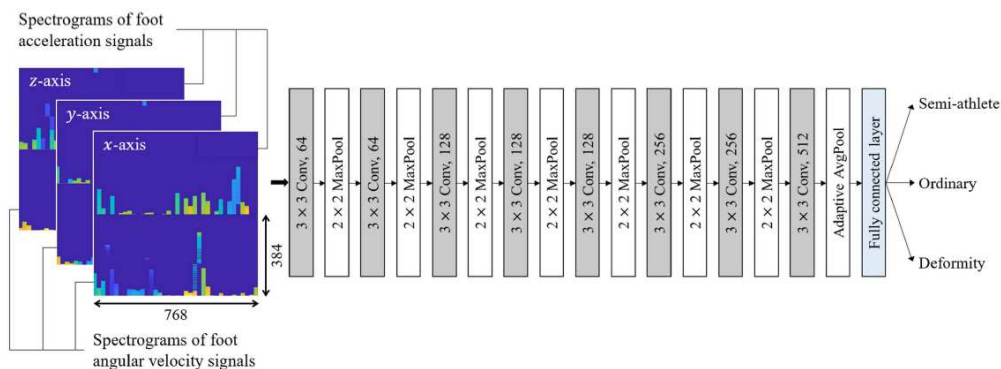
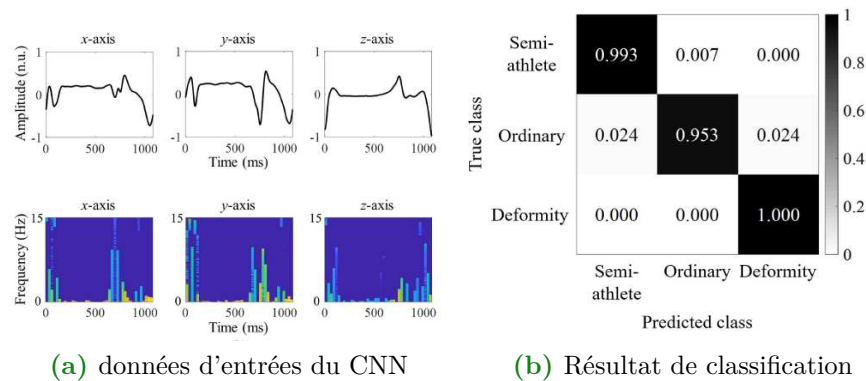
Une application de cette méthode [98] permet de visuellement distinguer un nouveau signal indiquant la présence d'inflammation articulaire des membres inférieurs en le comparant à une banque de signaux de référence préalablement identifiés en les comparant avec le DTW.

Un autre exemple du DTW est le travail de [102] afin de faire la reconnaissance de vidéos de gestes de la main indiquant les 25 lettres et les 10 chiffres en langage des signes. La méthode permet de classifier les gestes avec une précision de 90%.

D'autres modèles d'apprentissage automatique sont aussi utilisés pour établir une CPDM [89], comme les modèles statistiques de mélange gaussien (GMM) [103], le support vector machine (SVM) [104] ou le modèle de Markov caché (HMM) [105].

Les approches par ANN sont aussi utilisées pour détecter différentes conditions de marche. Les modèles sont principalement des ANN de type statique [89, 106]. Par exemple, une étude [107] utilise un FNN pour classifier des participants normaux par rapport à des participants avec des troubles neurobiologiques basés sur des paramètres d'angles des hanches, des genoux et de chevilles, avec une précision de 85 %. Les travaux de [7] utilisent un CNN afin de discriminer des sujets de trois différentes conditions, comme montré dans la figure 4.2. Les résultats de classification sont excellents, mais la tâche reste tout de même assez facile.

Une approche [108] par reconnaissance d'image d'une représentation 2D des signaux des capteurs inertiels avec un CNN a aussi été utilisée pour éviter une perte de précision liée à la présence de bruit dans les données brutes, qui est généralement un problème dans le cas des capteurs inertiels [106]. Cette approche réussit à faire la reconnaissance d'identité selon les signaux IMU de marche avec 95 % de précision sur deux bases de données indépendantes, ce qui est une nette amélioration par rapport à la méthode de référence de l'ordre de 85 %. On voit toutefois que si l'on tente de faire la reconnaissance avec des signaux pris sur des jours différents entre les données d'entraînement et de tests, la précision du modèle chute alors à environ 65 %. Plusieurs autres exemples de reconnaissance d'image pour l'analyse de la marche en exploitant des CNN sont répertoriés dans [109].



**Figure 4.2** – Application d'un CNN logiciel pour la classification de patients selon trois conditions (sportif, normale et pathologique). Les données d'entrées du CNN correspondent aux pixels du spectrogramme obtenu par transformées de Fourier par fenêtrage du signal d'accélération. [7]

Finalement, d'autres travaux utilisent aussi des réseaux récurrents (RNN) afin de faire une classification avec des signaux temporels [110]. Une méthode par analyse d'image a été investiguée afin de faire l'identification du syndrome de Parkinson avec un RNN en exploitant des coordonnées de position 2D tirés d'une séquence d'images [110]. Cette étude réussit à classifier plus de 150 séquences d'images de 45 personnes et démontre une capacité de classification d'une sensibilité (taux de distinction de personnes pathologiques) de 83.8 % et d'une précision (taux de distinction de personne saine) de 82.3 %. Ce résultat est encourageant, mais indique aussi que la classification reste complexe et qu'une amélioration des performances est encore possible.

### 4.1.3 Perceptive de la méthode RC pour la classification

Les méthodes de classification de la marche mentionnées dans cette revue ont été faites dans des conditions variables (nombre de patients, durée et types de mesures, lieux de test, nature de la classification), ce qui rend chaque application unique et difficilement comparable aux autres. Toutefois, il est clair que les corrélations à établir entre la condition d'un patient et les signaux de mouvement de la marche sont non-linéaires et non-

généralisables. Les approches de corrélation linéaire entre des métriques biomécaniques et une pathologie spécifique sont intéressantes pour adopter une compréhension physique d'une condition, mais offrent un faible potentiel de classification, surtout en début d'une condition dégénérative.

Les modèles AI, comme les ANN semblent adaptés dans ce domaine afin de traiter plusieurs conditions et de grands ensembles de données expérimentales. Les exemples de classification de données biomécaniques fait par ANN montrent clairement leur bon potentiel pour la CPDM.

L'utilisation des ANN dans les exemples de la revue cible généralement des informations hauts niveaux de la marche (conditions, états physiques). Chaque condition peut être reliée à des PDM ou à des biométries assez spécifiques selon les patients et les conditions qui sont très peu généralisables. Une seconde approche serait d'utiliser un ANN capable de s'adapter selon la situation sur divers PDM jugées plus pertinentes. Nous allons donc utilisé le RC pour une nouvelle application de CPDM dont l'objectif est de détecter des mouvements spécifiques bas niveaux. La méthode est démontrée avec 5 types de PDM différents, mais se veut adaptable à d'autre éventuellement.

Le potentiel computationnel du RC est comparable à certain ANN [111], mais l'utilisation du ESN afin de faire de la CPDM en temps réel n'est pas encore investigué pour des applications de l'analyse de marche. Le RC-MEMS démontre déjà des capacités encourageantes (figure 2.19) pour une tâche de classification de signaux vocaux.

#### 4.1.4 Objectifs du chapitre

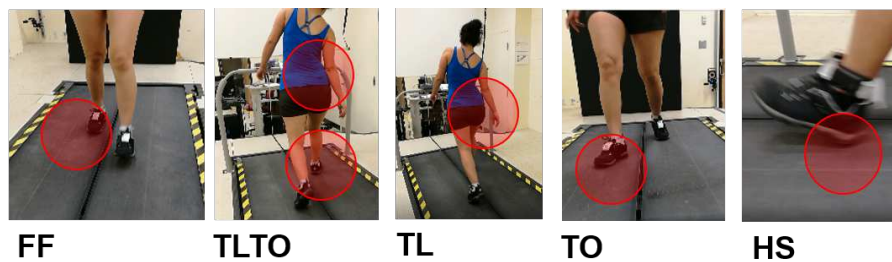
---

L'objectif du chapitre est d'évaluer la performance du ESN pour la CPDM de 5 types de PDM avec l'accélération du pied. La démonstration est faite sur des PDM pris selon 5 différentes vitesses et deux patients, un homme (H) et une femme (F). Il est aussi difficile d'établir quelles sont les limites de performances de classification d'un ESN en fonction de différents facteurs de variabilité. On évalue ici le résultat de performance du RC en utilisant plusieurs niveaux de variabilité sur les données (nombre de patients, vitesses). On souhaite aussi évaluer la performance du ESN dans un cas ou les données d'accélération d'entraînement et de test vont changer en continu, tout comme dans un cas réel de classification.

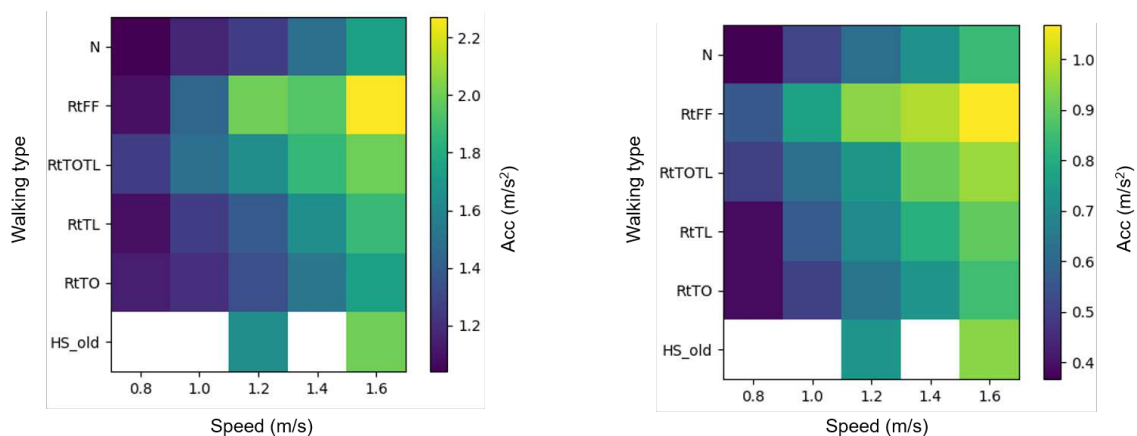
## 4.2 Méthodologie

### 4.2.1 Données expérimentales

La CPDM dans ce projet se concentre sur quelques PDM typiques observés en clinique pour les patients ayant une condition d'arthrose. Des kinésithérapeutes du laboratoire CIRRIS à Québec ont fait des essais expérimentaux avec 5 types de patrons de marche comme illustré dans la figure 4.3. Trois de ces PDM, soit l'ouverture du pied droit vers l'extérieur RtTO, l'inclinaison du tronc vers la droite rtTL et les deux à la fois rtTOTL, sont des stratégies utilisées pour le traitement préventif de l'arthrose, [34, 35, 78, 90]. Le PDM RtHS est une mauvaise technique de marche qui inclut un coup prononcé lors du contact initial du talon, pouvant être la cause du début ou de la progression de l'arthrose au genou [90]. Le PDM RtFF se distingue par une marche sans dépôt initial du talon altérant la phase de loading response (figure 1.1) au début du cycle et peut être représentative d'une boiterie. Le préfixe Rt signifie que la démarche avait été modifiée uniquement sur le pied ou le côté droit du corps. Chaque PDM ont été répété pour différentes vitesses (5 pour tous les types et 2 pour HS). Le signal d'accélération était mesuré à une fréquence de 200 Hz sur le dessus du pied.



(a) Types de pathologies simulées par les chercheurs du CIRRIS, soit le normal (N), le Foot-flat (FF), le Toe out (TO), le Trunk lean (TL), les deux, soit Trunk-lean et Toe out (TLTO), et le vendredi, Heel strike (HS).



(b) 95<sup>e</sup> percentiles de la valeur maximale des signaux d'accélération.

(c) Écart-type de l'amplitude des signaux d'accélération.

**Figure 4.3** – Représentation des accélérations pour les différents PDM inclus dans la base de données CIRRIS20.

Les figures 4.3b et 4.3c montrent l'amplitude et l'écart-type des signaux d'accélération selon les différents PDM. Selon ces deux métriques, certains signaux peuvent être à la fois très similaires ou très différents. Par exemple, le PDM RtTO et N à une vitesse de 1 m/s montrent approximativement la même amplitude aux 95<sup>e</sup> percentiles et le même écart-type. Ces deux patrons sont beaucoup plus similaires comparativement par exemple au RtFF ayant une amplitude et écart-type nettement supérieur.

La similitude augmente généralement la difficulté à discriminer des signaux. Les PDM RtTO et N seront donc probablement plus difficiles à classer que le PDM RtFF.

## 4.2.2 Définition des données d'entrées et objectives

### 4.2.2.1 Données d'entrée

Les données d'entrées du RC sont les accélérations unidimensionnelles dans l'axe AP sur le dessus du pied. Le vecteur d'entrée  $\mathbf{u}(t_i)$  se définit donc comme un vecteur colonne de deux dimensions (avec l'ajout d'un élément constant de 1) :

$$\mathbf{u}(t_i) = \begin{bmatrix} 1 \\ a_{AP}(t_i) \end{bmatrix}. \quad (4.1)$$

Les autres axes (ML et V) ont aussi été utilisés comme données d'entrées lors de tests préliminaires. De moins bonnes performances par rapport à l'accélération dans l'axe AP avaient été obtenues. Les résultats présentés sont donc uniquement ceux de l'axe AP.

### 4.2.2.2 Données objectives

Les données objectives  $\mathbf{y}^t(t_i)$  à chaque pas de temps  $t_i$  représentent un vecteur colonne binaire de 5 dimensions. Chaque dimension représente une valeur binaire associée à l'un des types de patrons de marche. La valeur du vecteur sera de 1 uniquement pour l'indice du vecteur associé au type de PDM effectué pour cette donnée. Les autres valeurs de  $\mathbf{y}^t(t_i)$  sont de 0 pour toutes les autres valeurs du vecteur.

$$\mathbf{y}^t(t_i) = \begin{array}{c} \text{for N} \quad \text{for RtTO} \quad \text{for RtTL} \quad \text{for RtTOTL} \quad \text{for RtFF} \\ \hline \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} \quad \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} \quad \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \end{array} \quad (4.2)$$

Les PDM HS n'ont pas été enregistré dans toutes les vitesses et pour les deux participants dans cette expérience. Ce PDM a été fait uniquement pour le participant masculin et pour les vitesses 1.2 et 1.6 m/s. Nous avons donc conservé uniquement les données des autres PDM ayant été testés à toutes les vitesses afin d'avoir un mélange uniforme des différents PDM dans les résultats.

### 4.2.3 Configuration des données d'entraînement

Les données de marche du CIRRIS20 ont été enregistrées dans des essais individuels pour chaque patron de marche, chaque patient (homme et femme) et chaque vitesse. Plusieurs entraînements et tests ont été faits en ne considérant qu'une seule vitesse (ISpeed) ou participant (IPart), alors que d'autres ont inclus toutes les vitesses (MSpeed) et les deux participants (MPart). De plus, les données étaient combinées selon deux types de configuration, l'une sous forme de signaux individuels pour chaque PDM (blocs individuels), et l'autre sous forme de signaux continus incluant tous les PDM (bloc continu).

#### 4.2.3.1 Sélection du participant et de la vitesse

Les données ont été enregistrées avec deux patients (une femme et un homme), ainsi qu'avec 5 vitesses différentes (0.8, 1, 1.2, 1.4, 1.6) m/s pour chaque patron de marche.

La configuration ISpeed ne considère qu'une seule vitesse de marche à la fois dans les données d'entraînement et de test. Les résultats pour la configuration ISpeed sont représentés par la moyenne des résultats de classification de 5 procédures indépendantes d'entraînement et de test font sur chacune des vitesses. MSpeed considère toutes les vitesses de marche à la fois dans les données d'entraînement et de test. Les résultats sont la moyenne de 5 procédures d'entraînement et de test, dont chacun regroupait les données des 5 vitesses à la fois.

La configuration IPart ne considère qu'un seul participant (soit un des deux participants) à la fois. De même, les résultats de IPart représentent la moyenne d'un entraînement/test fait uniquement sur les données F et d'un autre fait uniquement sur les données M. MPart considère les deux participants (les deux participants) dans les données d'entraînement et de test, mais un seul test a été fait pour les deux participants.

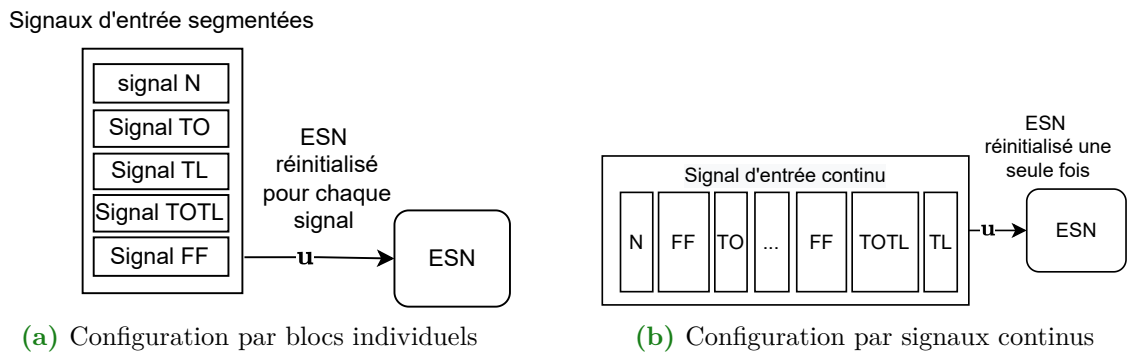
#### 4.2.3.2 Configuration des données par bloc individuel ou continu

Des configurations par blocs individuels et par blocs continus des signaux d'entraînement et de tests ont été utilisés pour évaluer les performances du ESN, représentées dans la section 4.2.4 .

La première configuration, par blocs individuels, permet d'entraîner le ESN sur un format brut des données expérimentales, c'est à dire séparés en signaux individuels pour chaque type de patron de marche. Dans cette configuration, le ESN est réinitialisé entre chaque signal et la phase transitoire au début de chaque signal est éliminée. La classification est donc faite sans transition entre les patrons de marche, ce qui permet d'évaluer la performance de classification uniquement pour une réponse en régime permanent (après plusieurs cycles dans le même du même patron de marche). Cela est donc moins réaliste par rapport à une application réelle, ou une personne en rééducation risque de souvent alterner entre une marche correcte et incorrecte.

Pour la seconde configuration, par bloc continu, deux signaux sont formés (un pour l'entraînement et un pour le test), comprenant chacun des données de tous les patrons de marche, vitesse et participant dans le même signal continu. Cela a été fait dans le but d'avoir des données plus représentatives d'une condition réelle de détection. Elle permet d'évaluer la performance du ESN en incluant des phases transitoires (changement de type

de patron de marche) sans réinitialiser le ESN. Les signaux continus sont construits selon la procédure expliquée dans la section 4.2.4.



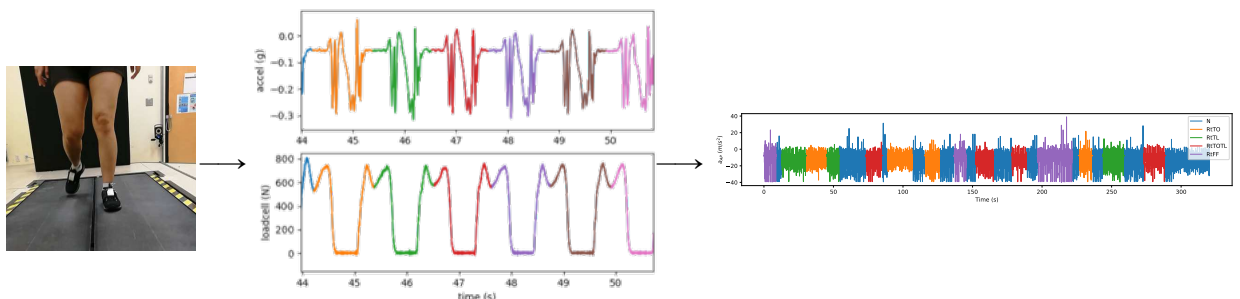
**Figure 4.4** – Deux configurations pour la construction des signaux utilisées dans la phase d'entraînement et la phase de test.

#### 4.2.4 Construction des données par bloc continu

On explique ici la construction des séquences continues de données d'accélération regroupant plusieurs types de PDM et avec plusieurs vitesses de marche. Elle consiste en un prétraitement des données brutes en deux étapes. La segmentation et la reconstruction de signaux continus, illustrées dans la figure 4.5, nécessaire afin de construire les séquences d'accélérations mixtes entre des données des différents PDM.

##### 4.2.4.1 Segmentation des données brutes

La première étape de prétraitement est la segmentation par cycle de marche des données continues d'accélération. On utilise des données de force normale au sol, synchronisées avec les signaux d'accélérations selon la procédure de l'annexe A, afin de séparer chaque cycle de marche dans une zone où l'accélération est approximativement nulle (au foot-flat FF), comme définies dans le chapitre 3. Cela permet d'éviter que la transition ne se fasse lorsque l'amplitude des signaux est non nulle, ce qui peut ajouter un contenu haute fréquence indésirable.



Segmentation des données brutes d'accélération au foot flat (FF)

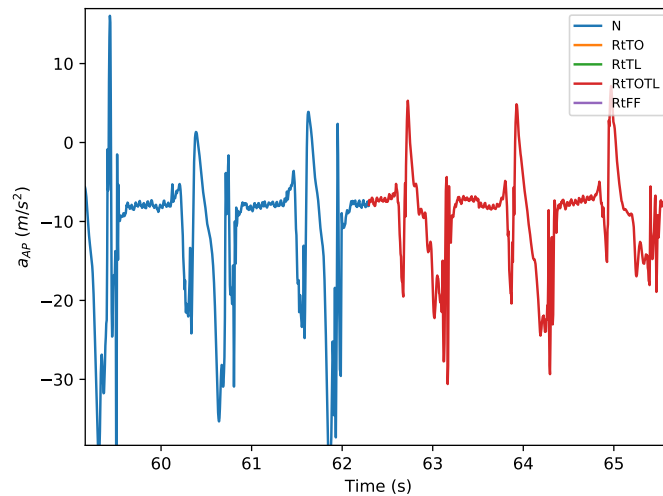
Recombinaison des séquences d'accélération avec plusieurs patrons de marche

**Figure 4.5** – Traitement des données de marche pour la tâche AD.



#### 4.2.4.2 Combinaison des données segmentées

La deuxième étape est la combinaison des signaux segmentés afin de construire différentes séquences continues de signaux d'accélération intégrant différents PDM. Les cycles d'accélération individuelle sont recentrés à une médiane nulle et sont connectés ensemble au point du foot-flat, tous bout à bout formant le signal continu utilisé dans l'entraînement et le test. Comme l'amplitude est quasiment nulle au point du foot-flat, il n'est pas nécessaire de connecter les cycles avec une transition linéaire. Les connexions des différents signaux sont faites avec une transition instantanée, comme montré dans la figure 4.6.



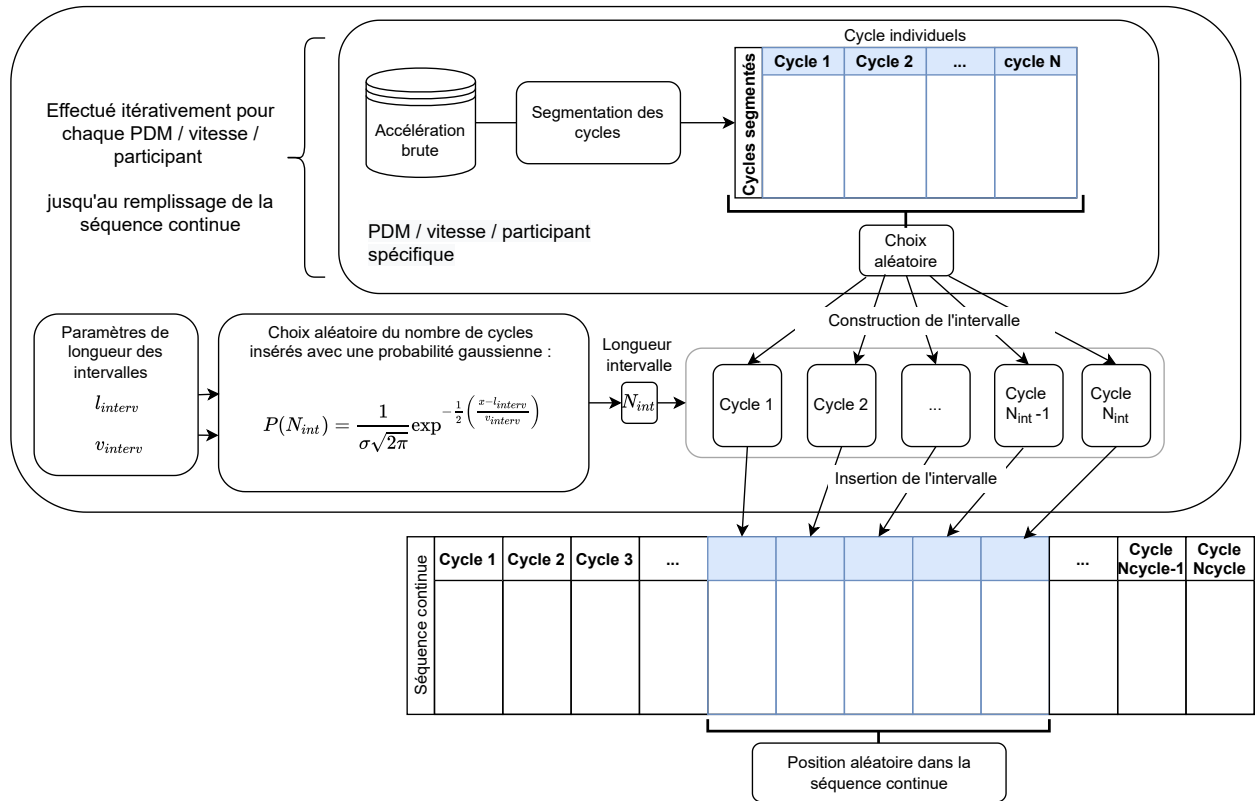
**Figure 4.6** – Transition entre un PDM N vers un PDM RtTOTL

Pour ce faire, on établit d'abord un nombre de cycles de marche  $N_{int}$  totaux de la séquence finale mélangée. Ensuite, pour chaque cycle, on sélectionne un PDM parmi les possibilités de PDM, de vitesse et de participant. La sélection est faite de manière aléatoire dans la séquence, mais en respectant des intervalles de plusieurs cycles ayant le même type de données. La position de ces intervalles est aléatoire, et leur longueur est choisie selon une probabilité gaussienne, comme montré dans la figure 4.7.

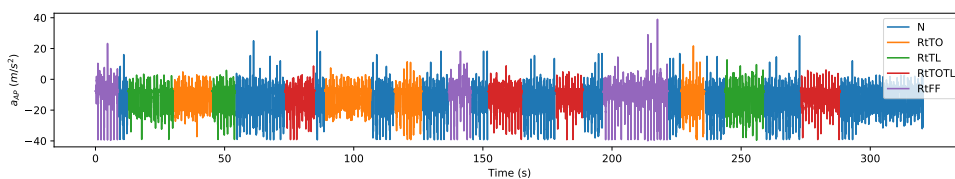
La longueur des intervalles est déterminée de manière aléatoire suivant une probabilité gaussienne. La gaussienne est définie avec un centre  $\mu = l_{interv}$  ou d'écart-type  $\sigma = lv_{interv}$ . Dans le résultat des sections 4.3.1, 4.3.3 et 4.3.4,  $l_{interv}$  est fixé à 10 cycles et l'écart type ( $v_{interv}$ ) est fixé à 1 cycle de variabilité. Cela revient à inclure les patrons de marche par morceaux en moyenne de 10 cycles, avec une très faible variabilité d'environ  $\pm 2$  cycles pour 95% des intervalles incluses dans la séquence continue, comme illustré dans la figure 4.8. Considérant que le remplissage final de la séquence continue se fait par des cycles N, les intervalles de présence de cycle N sont généralement de longueur plus variable par rapport aux autres cycles.

Ces paramètres sont toutefois modifiés dans la section section 4.3.5 afin d'évaluer l'influence de la longueur des intervalles et leur variabilité sur la performance du RC ainsi que les paramètres de traitements. Dans la section 4.3.5.1 des résultats, on évalue l'influence de la longueur des intervalles sur la performance de prédiction du RC. Le

Insertion de cycle dans la séquence continue à partir de données brutes

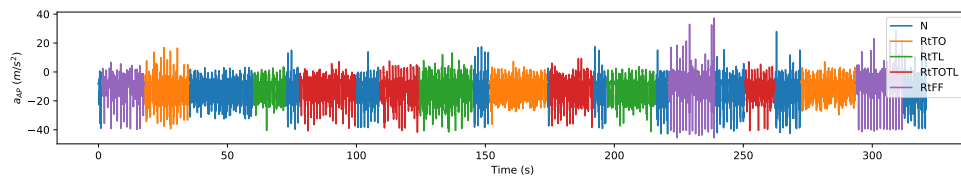
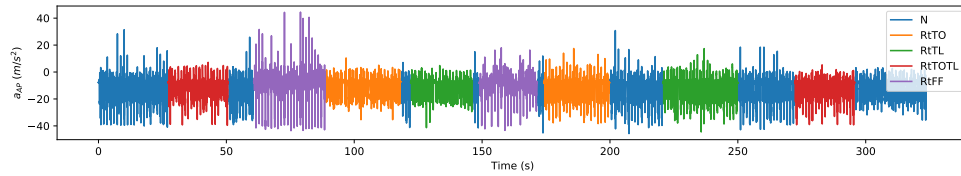
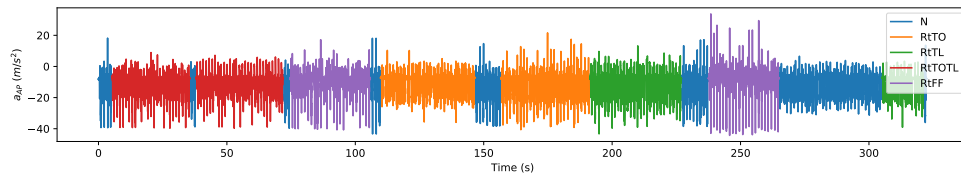
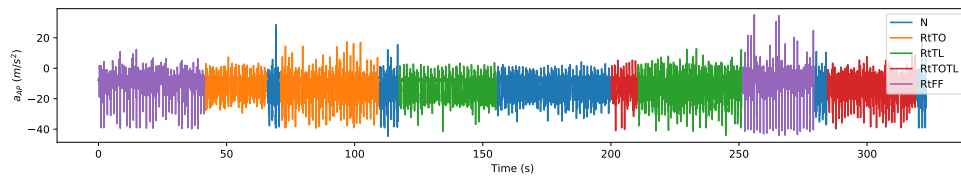


**Figure 4.7** – Protocole de construction des séquences continues en sélectionnant de manière aléatoire parmi les cycles segmentés de chaque signal expérimental acquis (pour différentes vitesses, PDM et participant). Le nombre de cycles pour chaque enregistrement spécifique est déterminé selon la loi gaussienne centrée selon une moyenne  $l_{interv}$  et un écart-type de  $v_{interv}$ . Les cycles  $N_{int}$  sont insérés à une position aléatoire de la séquence continue. L'espace résiduel se remplit par des cycles de marche N.



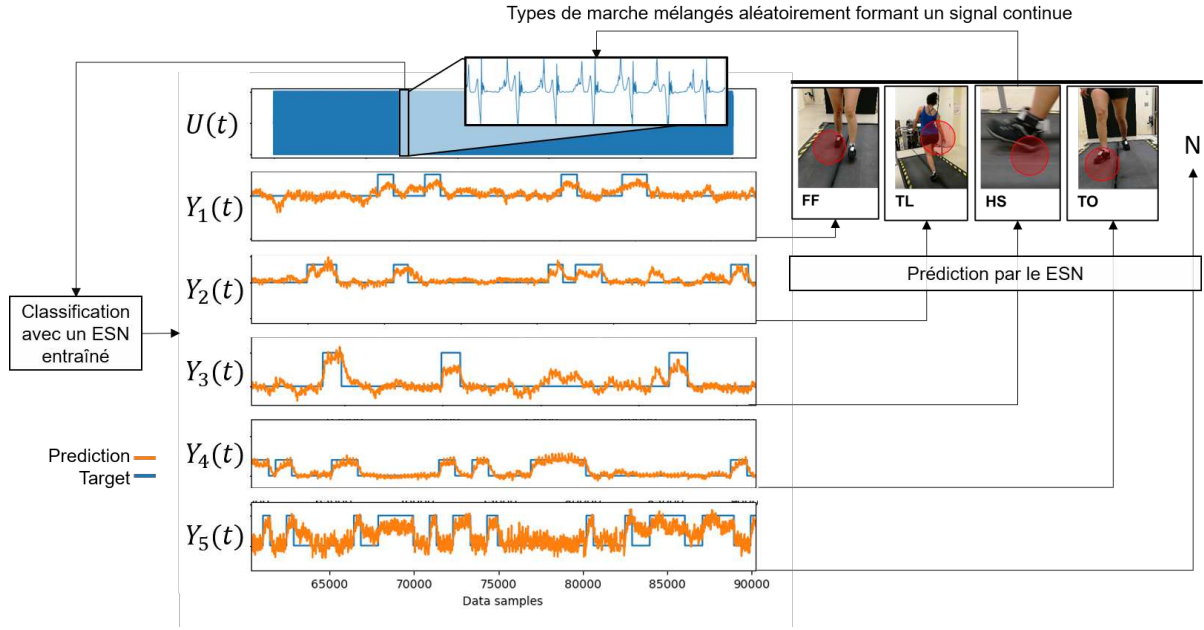
**Figure 4.8** – Construction d'une séquence continue de données.

paramètre  $l_{interv}$  sera variable entre 5 et 50 cycles. On peut voir différentes séquences continues incluant plusieurs choix de valeur moyenne  $l_{interv}$  dans la figure 4.9.

(a) Pour  $l_{interv} = 15$ (b) Pour  $l_{interv} = 20$ (c) Pour  $l_{interv} = 25$ (d) Pour  $l_{interv} = 30$ 

**Figure 4.9** – Construction de séquences de données avec différentes longueurs d'intervalle  $l_{interv}$  de PDM.

Dans la section 4.3.5.2, on considère cette fois une situation où on conserve la longueur moyenne constante à  $l_{interv} = 20$  cycles, mais où on augmente progressivement la variabilité des intervalles sélectionnés entre  $v_{interv}=1$  à  $v_{interv}=8$  cycles d'écart-type. Cela permet d'évaluer un cas plus complexe où les séquences d'entraînement ou de test sont introduites avec des longueurs variables. Cela nous permet donc d'avoir des signaux d'accélération ( $\mathbf{u}(t_i)$ ) en entrée et un vecteur d'identification objectif ( $\mathbf{y}^t(t_i)$ ) continue en sortie avec différents PDM mélangés. La figure 4.10 montre une tâche de classification de 5 types de PDM mélangés dans un signal continu résolue avec un ESN.



**Figure 4.10** – Détection de divers types de marche normale et pathologique pour un test de classification fait sur 500 cycles de marche en continu. Les lignes bleues sont les sorties objectives définies par l'identification expérimentale et les courbes oranges sont les prédictions d'un RC pré-entraînés.

## 4.2.5 Post-traitement des données de prédiction

### 4.2.5.1 Moyennage des valeurs de prédiction

Afin de réduire le bruit haute fréquence dans les données de prédiction  $\mathbf{y}(t_i)$ , permettant aussi d'éliminer une partie des fausses détections, on fait un moyennage des valeurs de sortie du RC avant de faire la classification. Selon le nombre de points d'intégration dans le passé,  $N_{int}$ , la valeur moyennée de la sortie se définit comme

$$\tilde{\mathbf{y}}(t_i) = \sum_{\tau=0}^{N_{int}} \mathbf{y}(t_i - \tau). \quad (4.3)$$

Le paramètre  $N_{int}$  est optimisé dans le posttraitement des données de test de manière à maximiser la valeur de l'AUC. Une première section des résultats (section 4.3.1) tente d'évaluer la performance de classification selon différentes valeurs de  $N_{int}$ .

### 4.2.5.2 Construction de la courbe AUC

La classification pour chaque PDM à l'instant  $t_i$  se fait en comparant la valeur prédite  $\tilde{y}_i$  par rapport à un certain seuil  $S_i$ . Si la valeur de prédiction est en dessous du seuil, on considère que le signal n'est pas celui PDM  $i$ . Si la prédiction dépasse le seuil, on considère que le signal est celui du PDM, et donc,

$$y^p = \begin{cases} 0 & \text{si } \tilde{\mathbf{y}}(t_i) > S \\ 1 & \text{si } \tilde{\mathbf{y}}(t_i) < S \end{cases} \quad \begin{array}{l} \text{: PDM non détecté} \\ \text{: PDM détecté} \end{array} \quad (4.4)$$

La valeur objective  $y_i^t$  indique si réellement le signal provenait ( $y_i^t = 1$ ) ou non ( $y_i^t = 0$ ) de ce PDM. Selon la valeur objective  $y^t$  et la valeur prédite  $y^p$ , 4 possibilités de résultat sont possibles, soit un vrai positif (TP), un vrai négatif (TN), un faux positif (FP) ou un faux négatif (FN), comme montré dans la tableau 4.1.

Objectifprédiction	$y^p=1$	$y^p=0$
$y^t =1$	TP	FN
$y^t =0$	FP	TN

**Table 4.1** – Attribution des cas de classification pour un des PDM

Le paramètre de seuil  $S$  peut être ajusté. Plus le seuil  $S$  est bas, plus le nombre de FP est grand, alors que plus le seuil est haut, plus le nombre de FN est grand. On choisit le paramètre  $S$  de manière à ce que le nombre de cas FP et de cas FN soit le plus balancé possible. Pour ce faire, on peut donc définir la sensibilité et la spécificité de la classification sur la totalité des pas de temps de la séquence de données en prenant les différents cas TP, FP, TN et FN. Les métriques de sensibilité et de spécificité de la classification sont définies selon les métriques ROC du *True positive rate* (TPR) et du *False positive rate* (FPR) [112], comme suit :

$$\text{Sensibilité : TPR} = \frac{\# \text{ cycles pathologiques estimés}}{\# \text{ cycles pathologiques}} = \frac{\# \text{ TP}}{\# \text{ TP} + \# \text{ FN}}, \quad (4.5)$$

$$\text{Spécificité : FPR} = \frac{\# \text{ cycles non-pathologiques estimés}}{\# \text{ cycles non-pathologiques}} = \frac{\# \text{ TN}}{\# \text{ TN} + \# \text{ FP}}. \quad (4.6)$$

Ces deux métriques permettent de traduire la qualité de la détection pour une certaine valeur de seuil  $S$ . Le seuil peut varier afin de trancher les données de prédictions à une certaine valeur entre le minimum et le maximum de  $\tilde{y}$ . Le choix du seuil  $S$  influence directement ces deux métriques de performance de la classification. Un seuil très faible sera très sensible à la présence du PDM, mais sera peu spécifique (grand TPR et faible FPR). Un seuil très grand sera moins sensible, mais sera plus spécifique et évitera l'identification erronée du PDM (grands FPR et faible TPR). Le choix de  $S$  peut être laissé à la discrétion du clinicien souhaitant éventuellement exploiter ce système. On peut évaluer une performance globale de discrimination sous la forme de la courbe ROC en représentant  $\text{TPR}(S_i)$  en fonction de  $(1-\text{FPR}(S_i))$  dans un graphique pour de multiples valeurs de seuil  $S_i$ , représenté dans la figure 4.11a. On peut ainsi évaluer la performance de classification en considérant tous les seuils  $S$  possibles entre  $\min(\tilde{y}_j)$  et  $\max(\tilde{y}_j)$ , en calculant l'aire sous la courbe (AUC), représentée dans la figure 4.11b. L'indice  $j$  indique l'indice de un des 5 PDM (1 = N, 2 = rtTO, 3 = RtTL, 4 = RtTOTL, 5 = RtFF). Ainsi,

$$\text{AUC} = \int_{S=\min(\tilde{y}_j)}^{S=\max(\tilde{y}_j)} \text{TPR}(S)(1 - \text{FPR}(S))dS, \quad (4.7)$$

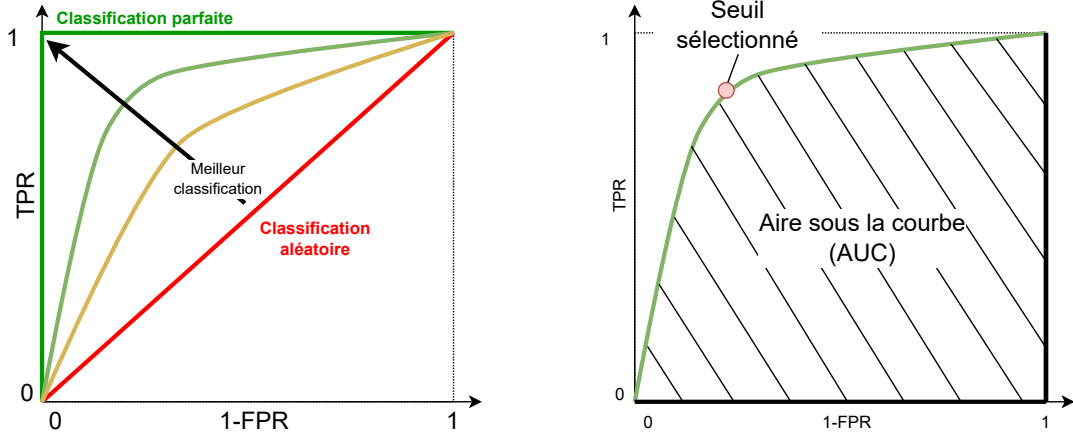
que l'on peut approximer avec une intégration numérique par trapèze sur plusieurs valeurs discrètes de sensibilité  $\text{TPR}(S_i)$  et de spécificité  $\text{FPR}(S_i)$  (associées à une valeur de seuil  $S_i$ ), comme suit,

$$\text{AUC} \approx \frac{1}{4} \sum_{i=0}^{N_S} [\text{TPR}(S_{i+1}) + \text{TPR}(S_i)] [\text{FPR}(S_i) - \text{FPR}(S_{i+1})] + [\text{TPR}(S_i) + \text{TPR}(S_{i-1})] [\text{FPR}(S_{i-1}) - \text{FPR}(S_i)], \quad (4.8)$$

avec  $S_i$  défini comme un seuil dans l'intervalle de possibilité dans  $\tilde{\mathbf{y}}_j$ ,

$$S_i = \min(\tilde{\mathbf{y}}_j) + i \frac{\max(\tilde{\mathbf{y}}_j) - \min(\tilde{\mathbf{y}}_j)}{N_S}, \quad (4.9)$$

et  $N_S$  représente le nombre de seuils testés pour construire la courbe AUC. Cette métrique traduit la qualité de classification d'un test selon une valeur entre 0.5 et 1. Une valeur de 0.5 représente une classification faite de manière aléatoire (mauvaise classification), alors que 1 est une classification parfaite. Le jugement de la qualité d'un test est variable selon l'application. De manière générale, un AUC se situant entre 0.9 et 1 est jugé comme une classification très performante, entre 0.9 et 0.7 est jugé comme correcte, et en dessous de 0.7 comme une classification peu performante.



(a) Courbes ROC pour différentes performances de classification : aléatoire, mauvaise, correcte, parfaite

(b) Représentation du AUC de la courbe ROC. Le point rouge représente le seuil sélectionné choisi dans la section 4.2.5.3

**Figure 4.11** – Représentation de la courbe ROC et du calcul de l'aire sous la courbe (AUC)

#### 4.2.5.3 Choix du seuil optimal par classe individuelle

Afin de sélectionner un seuil optimal, permettant d'équilibrer le nombre de FP et de FN dans le tableau 4.1, on peut définir le seuil associé au point supérieur gauche de la courbe ROC (montré dans la figure 4.11b), défini comme

$$S_{\text{best}}(\tilde{\mathbf{y}}_j) = \operatorname{argmax}_{S_i} ((\text{TPR}(S_i))^2 + (\text{FPR}(S_i))^2) \quad (4.10)$$

où  $j$  indique l'indice de l'un des 5 PDM (1 = N, 2 = rtTO, 3 = RtTL, 4 = RtTOTL, 5 = RtFF), et  $i$  est l'indice de la position du seuil dans l'intervalle de possibilité entre  $\min(\tilde{\mathbf{y}}_j)$  et  $\max(\tilde{\mathbf{y}}_j)$ . La sélection du seuil  $S_{\text{best}}$  se fait de manière individuelle pour chaque PDM, telque,

$$\mathbf{S}_{\text{best}} = [S_{\text{best}}(\tilde{\mathbf{y}}_1) S_{\text{best}}(\tilde{\mathbf{y}}_2) S_{\text{best}}(\tilde{\mathbf{y}}_3) S_{\text{best}}(\tilde{\mathbf{y}}_4) S_{\text{best}}(\tilde{\mathbf{y}}_5)] \quad (4.11)$$

#### 4.2.5.4 Choix du PDM parmi toutes les classes

Les signaux continus mélangeaient les différents types de PDM dans la séquence continue. La dernière étape consiste donc à sélectionner parmi les 5 types de patrons de marche, lequel sera sélectionné comme le PDM choisi par la sélection. Le choix est fait en sélectionnant la valeur qui est la plus haute par rapport à son seuil,

$$PDM_{\text{selected}}(t_i) = \operatorname{argmax}_j (\tilde{\mathbf{y}}(t_i) - \mathbf{S}_{\text{best}}), \quad (4.12)$$

où  $PDM_{\text{selected}}$  représentant un indice allant de 1 à 5 indiquant le PDM sélectionné.

### 4.2.6 Métrique de performance

Les performances du RC sont exprimées selon deux métriques et deux représentations graphiques. La première métrique est l'AUC moyenne de chaque classe de PDM. On représente graphiquement les résultats selon la courbe ROC de chaque type de PDM. Dans la section 4.3.1, on montre les résultats du nombre de points d'intégration en fonction de la valeur moyenne de l'AUC sur l'ensemble des classes. La deuxième métrique est la précision de classification pour chaque patron de marche  $\mathbf{S}_{\text{best}}$ , soit :

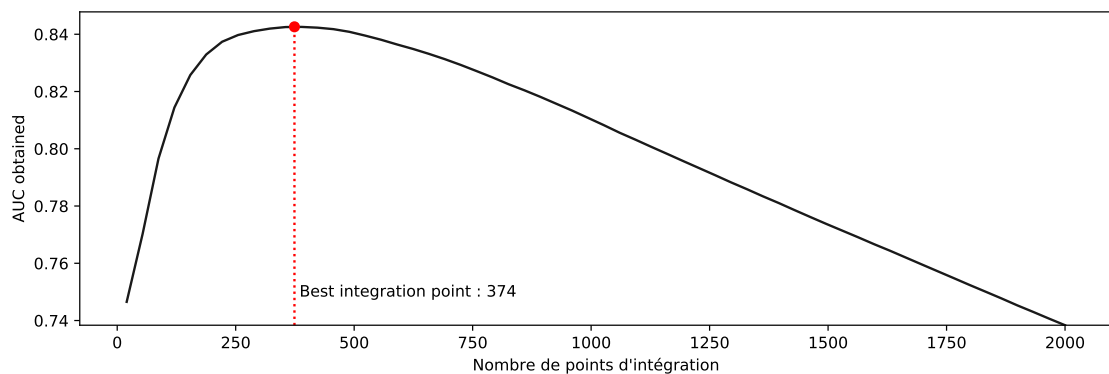
$$\text{Acc} = \frac{\sum_{j=1}^5 \#TP_j}{\#\{\tilde{\mathbf{y}}\}} \quad (4.13)$$

avec  $j$  l'indice du patron de marche ( $1 = N$ ,  $2 = \text{rtTO}$ ,  $3 = \text{RtTL}$ ,  $4 = \text{RtTOTL}$ ,  $5 = \text{RtFF}$ ) et  $\#\{\tilde{\mathbf{y}}\}$  est le nombre d'indices de temps dans la série temporelle. Cette métrique correspond à la proportion de points correctement identifiés de chaque classe par rapport au nombre de points total. Une classification aléatoire aurait une précision égale à l'inverse du nombre de classes indépendantes, soit de 20% pour le cas des 5 différents PDM. Les résultats sont aussi montrés sous la forme d'une matrice de confusion [113] entre les 5 classes, montrant la proportion de chaque classe prédite et objective possible obtenue avec les seuils sélectionnés pour chaque patron de marche  $\mathbf{S}_{\text{best}}$ .

## 4.3 Résultats

### 4.3.1 Choix du nombre de points d'intégration

La figure 4.12 montre la valeur de la moyenne des AUC obtenues (pour tous les PDM) en fonction du nombre de points d'intégration  $N_{\text{int}}$ . Ce résultat est montré avec des signaux en blocs continus dont l'intervalle de changement de patron était de  $l_{\text{interv}} = 10$  et  $v_{\text{interv}} = 1$  et avec une vitesse et les deux participants (MSpeed et MPart). On voit que le point optimal est de l'ordre de 374 points, soit environ 1.9 secondes de temps d'intégration, ou environ deux cycles de marche. On peut voir qu'il est mieux en général de surestimer le paramètre du seuil (faire un moyennage trop grand que nécessaire) au lieu de le diminuer, car la perte de performance selon la courbe de la figure 4.12 est plus rapide avant le maximum qu'après.



**Figure 4.12** – AUC moyen en fonction du nombre de points d’intégration utilisé en post traitement.

### 4.3.2 Résultats globaux pour les différentes configurations

Dans cette section, on présente les résultats pour les différentes configurations de construction des données d’entraînement et de test. Dans cette comparaison, les données par bloc continu ont été générées en utilisant une longueur moyenne de  $l_{interv}=10$  et  $v_{interv}=1$ . De plus, pour tous les résultats, le nombre de points d’intégration a été optimisé de manière à maximiser la métrique AUC, de manière identique à la section 4.3.1.

	Blocs individuels	Bloc continu		Blocs individuels	Bloc continu
ISpeed, IPart	0,980747	0,936316	ISpeed, IPart	0,972257	0,822084
ISpeed, MPart	0,920383	0,900063	ISpeed, MPart	0,775982	0,720658
MSpeed, IPart	0,880206	0,859548	MSpeed, IPart	0,6649	0,624057
MSpeed, MPart	0,882136	0,82102	MSpeed, MPart	0,665947	0,572018

(a) AUC moyen

(b) Précision de classification (Acc)

**Table 4.2** – Comparaison des résultats de classification pour tous les types de configurations des données d’entraînement : blocs individuels vs continu, ISpeed vs MSpeed et IPart vs MPart. Le AUC et la précision Acc sont des résultats moyens de toutes les PDM.

On voit de manière générale que plus les configurations incluent de la variabilité (participant et vitesse), plus la performance de classification diminue, la tâche est donc plus difficile. La configuration des données en blocs continus rend aussi la classification plus difficile par rapport aux blocs individuels. Cela fait diminuer le AUC moyen entre 2 et 6% et la précision entre 4 et 15 %. Cela est normal, car suite à une transition en continu des types de signaux dans la séquence, la réponse du RC doit prendre un certain temps pour obtenir une prédiction significative du nouveau PDM. Une erreur plus grande peut donc être associée à ces phases de transitions.

On voit selon la métrique de précision que 57% des points parmi l’ensemble, dans le pire des cas, seraient correctement classés, alors qu’une classification aléatoire serait de 20% considérant qu’il y a 5 types de PDM.



### 4.3.3 Classification PDM avec des vitesses uniformes

La figure 4.13 montre la matrice de confusion pour une classification à vitesse unique (ISpeed). Une procédure d'entraînement et de test pour chaque vitesse de marche (0.8, 1.0, 1.2, 1.4 et 1.6 m/s) a été effectuée. Les résultats illustrés dans la figure 4.13 représentent le cumulatif des proportions de prédictions faites sur chacune des vitesses.

#### 4.3.3.1 Difficulté de la classification selon la variabilité de participant

Les résultats sont presque parfaits lorsque la classification ne considère qu'un seul participant (IPart) et par bloc individuel. On voit que l'ajout des deux patients diminue significativement les performances. Les erreurs se retrouvent principalement dans les classes RtTL, RtTO et N. Cela laisse indiquer que les signaux des deux patients sont de natures assez différentes pour ces PDM. Généraliser la classification pour différents patients peut donc beaucoup augmenter la complexité de la tâche.

#### 4.3.3.2 Difficulté de la classification selon des blocs individuels ou continus

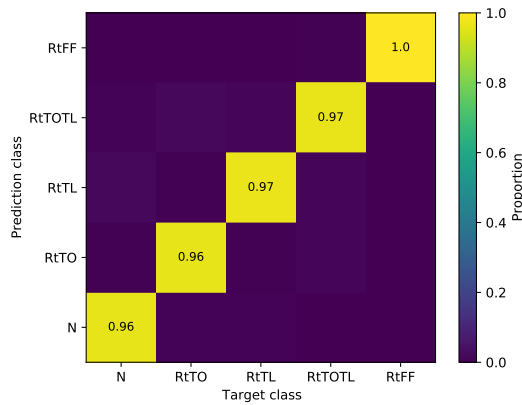
On voit que l'utilisation de signaux continus versus discrets fait aussi diminuer les performances de classification. Les cas d'erreur sont répartis de manière plus uniforme parmi les 25 possibilités de classification, mais que la prédiction N est tout de même celle avec la plus grande proportion d'erreurs.

#### 4.3.3.3 Difficulté de la classification selon les différents PDM

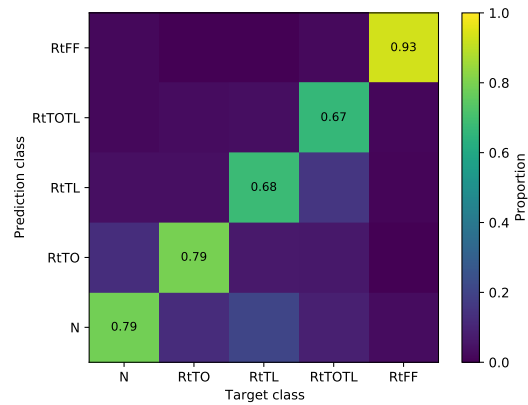
On voit que le PDM RtFF est nettement plus facile à détecter par rapport aux autres type de marche. Comme on peut l'observer dans la figure 4.3, il est facile d'identifier le RtFF uniquement en se basant sur l'amplitude et l'écart type du signal d'accélération. Il ne nécessite donc pas un algorithme complexe car le profil différent est visible à l'oeil nu dans les données. Il est donc normal que le PDM de type RtFF soit plus facile à classifier. Les PDM RtTL et RtTOTL sont ceux qui ont été les moins bien classifiés. Le RtTL semble être principalement confondu avec la marche normale et le RtTOTL est surtout confondu avec le RtFF. Dans les deux cas, on peut voir dans la figure 4.3 que les signaux présentent une grande similarité. On peut voir aussi que les signaux N, RtTL et RtTO sont assez similaires, ce qui explique aussi pourquoi la proportion d'erreurs est grande pour les PDM RtTO et RtTL détecté comme des PDM N. Toutefois, la proportion d'erreurs observable sur les valeurs des cases à l'extérieur de la diagonale est assez équilibrée entre la ligne et la colonne de chaque PDM. Cela confirme que le choix des seuils de prédiction  $S_{\text{best}}$  pour chaque PDM semble être adéquat, car le nombre de FP et de FN est alors assez équilibré.

#### 4.3.3.4 Répétabilité pour différentes vitesses

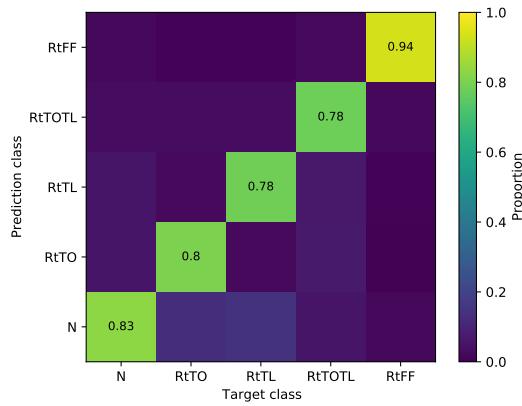
La figure 4.14, montre les courbes AUC de chaque PDM pour les différentes vitesses. La courbe du RtFF (en mauve) est nettement meilleure que les autres courbes. Les résultats de classification sont généralement assez similaires pour toutes les vitesses, sauf pour les vitesses de 1.2 m/s et de 1.4 m/s qui semblent légèrement meilleurs. Il est possible qu'il



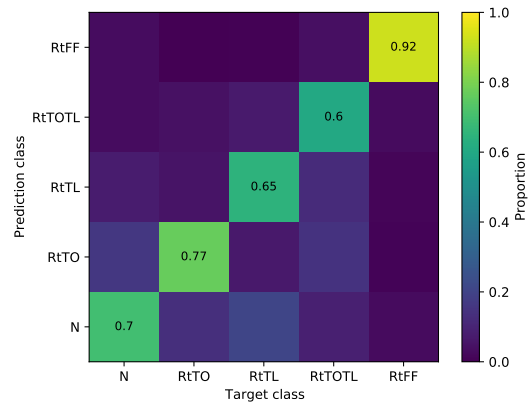
(a) IPart - blocs individuels



(b) MPart - blocs individuels



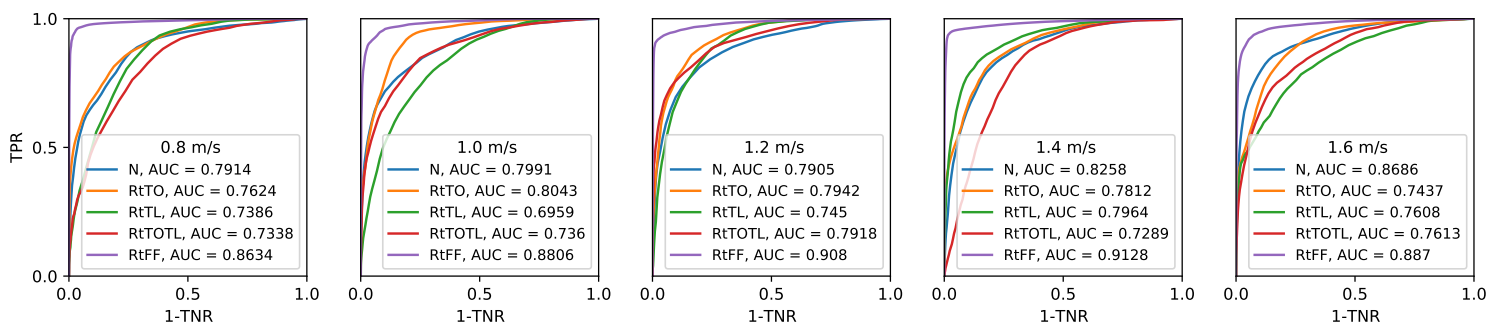
(c) IPart - bloc continu



(d) MPart - bloc continu

**Figure 4.13** – Résultats de classification pour des données de marche avec vitesse fixe (ISpeed).

était plus facile pour les deux participants de bien effectuer les PDM à ces vitesses qui sont plus proches d'une marche naturelle que les autres vitesses.



**Figure 4.14** – Courbes ROC obtenues pour les tests de classification avec RC selon les différentes vitesses de marche faites de manière individuelle, avec les deux participants considérés (MPart) et par blocs continus.

### 4.3.4 Classification PDM avec toutes vitesses confondue

La figure 4.15 montre les résultats de classification pour des entraînements et tests avec des signaux continus regroupant l'ensemble des vitesses et des PDM. Les résultats sont moins bons que dans les essais avec une seule vitesse à la fois. Dans ces blocs les amplitudes et les écarts-types des signaux se confondent pour tous les PDM ce qui rend le problème de classification beaucoup plus complexe. Une discrimination linéaire basée sur des métriques comme l'amplitude ou l'écart type n'est plus possible.

#### 4.3.4.1 Difficulté selon les participants et la continuité des données

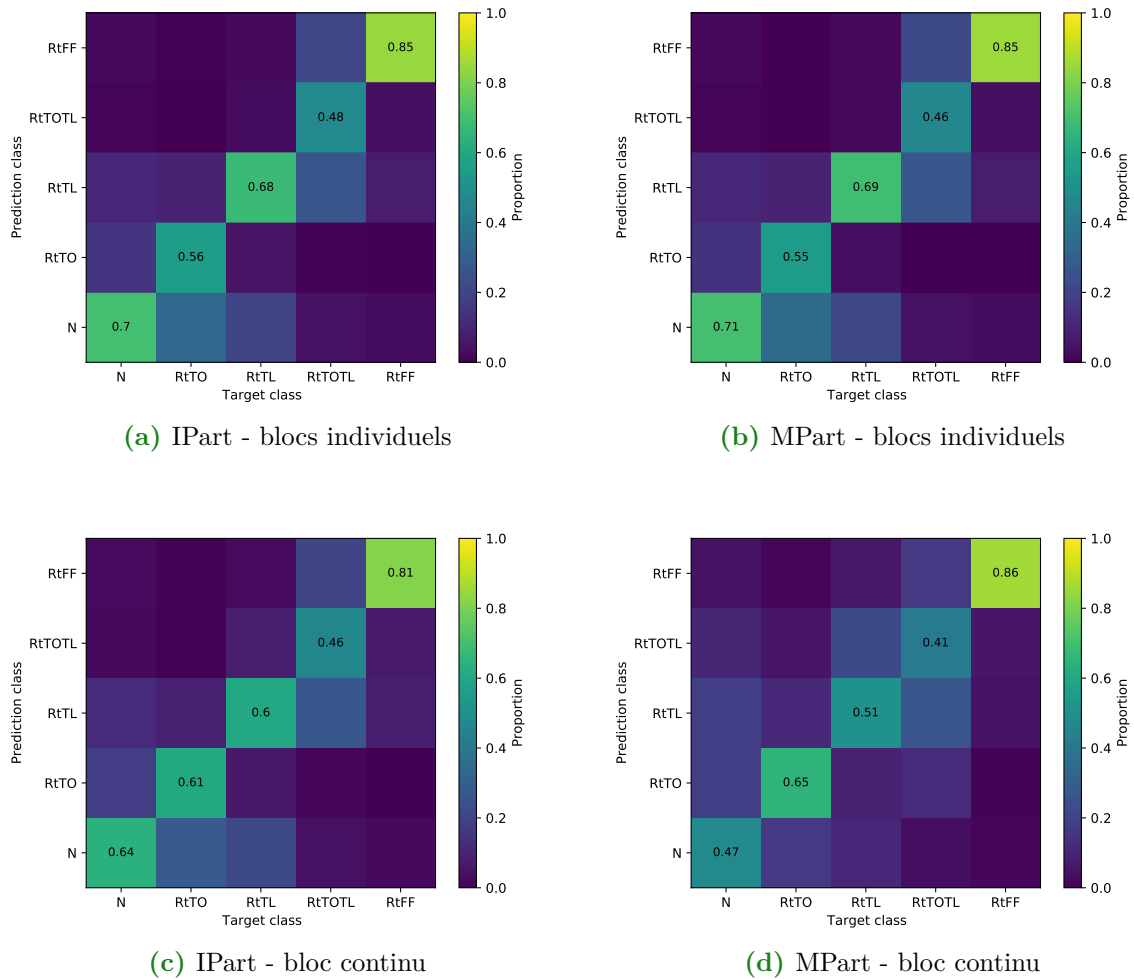
Dans ce cas-ci, l'utilisation de données provenant d'un (IPart) ou des deux participants (MPart) semblent donner environ les mêmes résultats. Lorsque les données sont utilisées selon des blocs continus, la performance chute légèrement.

#### 4.3.4.2 Difficulté selon les PDM

La répartition des prédictions pour les différents PDM est similaire aux résultats obtenus avec les vitesses uniques. Le RtTOTL et RtTL sont encore les deux conditions de marche les moins bien prédites et le PDM RtFF reste celui qui est le mieux prédit. On voit que le patron RtTOTL est souvent prédit comme du RtFF. Les PDM , et RtTL et RtTO sont fréquemment prédits comme N.

#### 4.3.4.3 Répétabilité pour différents essais

Afin d'évaluer la répétabilité des résultats, 5 procédures d'entraînements et de test, basées sur des séquences continues indépendantes, ont été faites. La figure 4.16 montre les courbes AUC pour les 5 séquences indépendantes entraînées et testées. Elles obtiennent une bonne répétabilité selon les différents essais. Le RtFF possède la plus grande aire sous la courbe, suivi par le RtTO, puis par le N, RtTOTL et puis RtTL. Le RtTOTL et RtTL, sont les cas les plus difficile à classifier, et leur courbe ROC sont plus variables entre les différents essais.

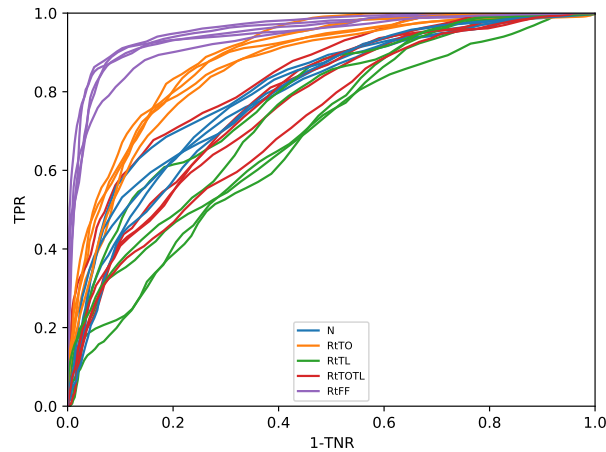


**Figure 4.15** – Résultats de classification pour des données de marche mélangeant toutes les vitesses pour tous les patients (MSpeed).

### 4.3.5 Classification selon la fréquence et la variabilité de la modification du PDM de la séquence continue

#### 4.3.5.1 Influence de la longueur moyenne $l_{interv}$ des intervalles de changement du PDM

La figure 4.17 permet d'évaluer l'influence de la durée des intervalles dans des données par blocs continus sur la performance de classification. La longueur des séquences est modifiée en faisant varier le paramètre  $l_{interv}$  utilisé lors de la construction d'une séquence continue, comme représenté dans la figure 4.7. La figure 4.17a montre la variation de la valeur du AUC optimal ainsi que le nombre de points d'intégration optimal en fonction du paramètre  $l_{interv}$ . On voit dans le graphique supérieur de la figure 4.17a que l'AUC augmente légèrement en augmentant la longueur des intervalles, soit environ de 5% (AUC moyen) en augmentant l'intervalle de 5 à 50 cycles de marche. Le graphique inférieur de la figure 4.17a montre aussi que le nombre de points d'intégration pour avoir un AUC optimal augmente. On passe d'un nombre de points optimal de 250 points d'intégration pour des



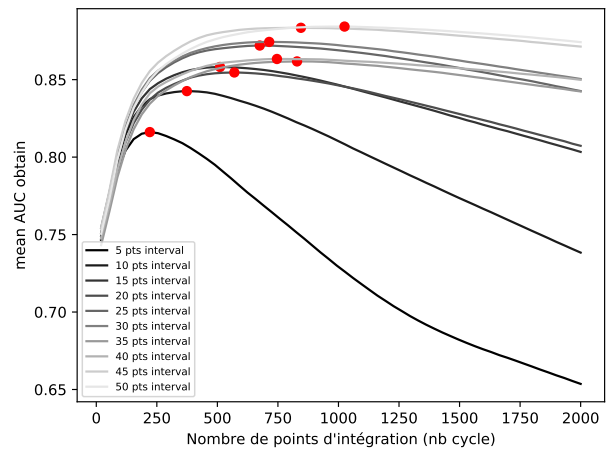
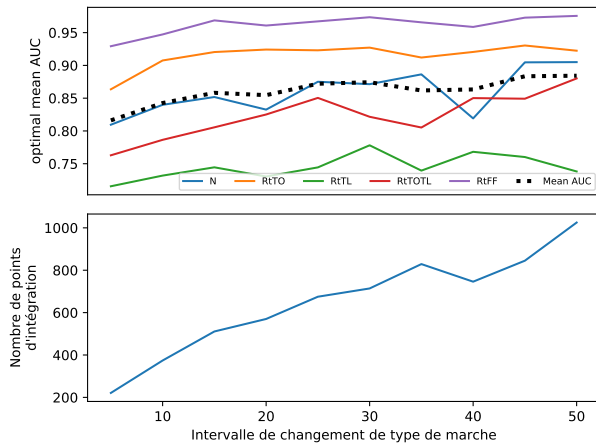
**Figure 4.16** – Courbe ROC sur les 5 classifications de séquences continues avec toutes vitesses confondues et en considérant les deux participants (MPart)

intervalles  $l_{interv} = 5$  cycles de marche jusqu'à 1000 pour  $l_{interv} = 50$  cycles. La classification est donc meilleure en augmentant le nombre de points d'intégration lorsque la durée dans laquelle un PDM est conservé par le marcheur est grande. La figure 4.17b montre les courbes d'AUC moyen en fonction du nombre de points d'intégration sélectionné pour l'ensemble des longueurs moyennes  $l_{interv}$  des intervalles. Dans les cas où  $l_{interv}$  est petit (courbe noire foncée), le choix du nombre de points d'intégration devient plus critique. Pour  $l_{interv} = 5$  cycles, le AUC optimal est obtenu pour un nombre de points de 250 environ. Si l'on choisit un nombre de points qui s'éloigne un peu de cet optimal (environ  $\pm 25$  points), le résultat de l'AUC chute rapidement. Toutefois, plus  $l_{interv}$  est grand, moins le choix du nombre de points d'intégration est critique. Dans le cas de  $l_{interv} = 50$  cycles, on voit que le résultat du AUC reste pratiquement constant pour toutes les valeurs de points d'intégration sur une plage d'environ  $1000 \pm 200$  points.

#### 4.3.5.2 Influence du paramètre de variabilité de la longueur des intervalles $v_{interv}$ sur la modification des PDM

La figure 4.18 permet d'analyser l'influence de la variabilité  $v_{interv}$  de la longueur sur la performance de classification. Ce paramètre permet d'ajuster la largeur de la gaussienne utilisée pour choisir la longueur des intervalles de chaque PDM insérés dans les signaux continus, comme illustré dans la procédure de construction de la figure 4.7.

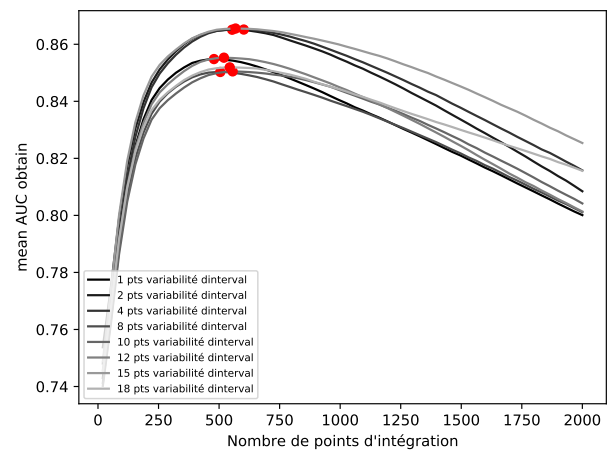
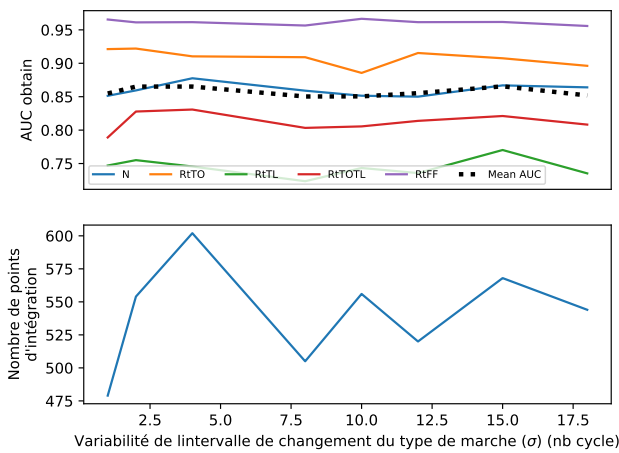
Les résultats présentés ici sont faits avec un paramètre  $l_{interv}$  fixé à 20 cycles. Le paramètre de variabilité  $v_{interv}$  de la longueur augmente graduellement, de 1 à 18 cycles. Cela implique que les cycles ont des longueurs de plus en plus variables. On voit que l'augmentation de la variabilité sur la longueur des intervalles de changement de PDM n'a pas d'influence sur la valeur de l'AUC, ni sur la valeur optimale du seuil de détection.



(a) AUC moyen (graphique supérieur) et nombre de points d'intégration optimal (graphique inférieur) en fonction de la longueur moyenne  $l_{interv}$  des intervalles de changement de donnée lors de la construction des séquences continues.

(b) AUC moyen en fonction du nombre de points d'intégration  $N_{int}$  pour les différentes longueurs d'intervalle  $l_{interv}$  testées lors de la construction des séquences de données.

**Figure 4.17** – Évaluation de la performance de classification en fonction de la longueur des intervalles de changement des PDM (paramètres  $l_{interv}$ ).



(a) AUC moyen (graphique supérieur) et nombre de points d'intégration optimal (graphique inférieur) en fonction de la variabilité des intervalles de changement des PDM.

(b) AUC moyen en fonction du nombre de points d'intégration pour les différentes valeurs de variabilité des intervalles de valeur entre 1 et 8 cycles d'écart-type de la fonction de choix gaussienne.

**Figure 4.18** – Évaluation de la performance de classification en fonction de la variabilité des intervalles de changement des PDM (paramètres  $v_{interv}$ ).

## 4.4 Discussion

## 4.4.1 Retour sur les performances globales du RC pour la CDPM

### 4.4.1.1 Performance selon les différents PDM

La classification des 5 PDM avec le RC est possible avec un seul capteur IMU sur le pied. La classification des PDM est très bonne lorsque la vitesse de marche est constante et qu'elle considère les signaux d'un seul participant à la fois. Ce serait généralement le cas dans une utilisation clinique. Il toutefois est réaliste de considéré aussi qu'un changement de patron de marche soit aussi relié avec une variation de vitesse de la marche, car la personne qui se force à adopté un patron de marche spécifique risque de ralentir légèrement au début. Il est donc pertinent si le RC est en mesure de faire de telle classification, à tout le moins, une petite marge de vitesse de marche.

Les mauvaises détections entre les PDM semblent arriver lorsque la nature des signaux est relativement similaire et qu'une classification linéaire n'est plus possible. Par exemple, les signaux RtTL et RtTO sont fréquemment prédits comme N. En effet ces trois PDM ont des signaux dont les amplitudes et les écart-types sont particulièrement similaires selon la figure 4.3 et la modification du mouvement de marche entre ces patrons est subtile. On remarque aussi que le PDM RtTOTL se confond fortement avec le RtFF. Les signaux de marche RtFF et RtTOTL sont les deux types de marche dont les signaux ont une amplitude et d'un écart-type plus grands que tous les autres PDM, et ce sur toutes les vitesses confondues.

Le PDM le plus facile à identifier est le RtFF, car les signaux d'accélération sont faciles à distinguer par rapport aux autres signaux en raison de leur amplitudes constamment plus grandes. Le PDM RtTL est le plus difficile à discriminer. Ce patron implique surtout des variations du mouvement dans le haut du corps, ce qui le rend moins distinguable d'une marche normale selon l'accélération du pied comparativement aux autres PDM. Un entraînement du RC avec des signaux d'accélération à une position plus proche du tronc pourrait probablement optimiser les performances du RC pour la détection du RtTL.

### 4.4.1.2 Performance selon les trois facteurs de variabilité considérés

La performance du RC pour chaque PDM est très variable, entre de 57 et 97 %, selon les différents facteurs considérés (les vitesses, les individus et la continuité des données). La considération de plusieurs vitesses est le facteur le plus important identifié ici, occasionnant une perte de précision entre 15 et 30 %. La considération des deux individus au lieu d'un seul à la fois fait aussi chuter les performances, entre 3 et 20 % lorsque les tests sont faits dans des vitesses uniformes. Toutefois, la base de donnée comptant uniquement deux participants dans ce chapitre n'était clairement pas suffisant pour tester de manière réaliste l'impact de la variabilité du profil de marche du grande population de patients. Une plus grande quantité de participants aurait probablement impliqué des AUC plus faibles en utilisant une approche d'entraînement où tous les participants sont considérés dans la même régression.

Le fait d'utiliser des signaux continus plus représentatifs d'un contexte réel d'utilisation fait aussi diminuer la précision du modèle, entre 6 et 15 % pour les cas de vitesse fixe et entre 4 à 9 % pour les cas de vitesses variables.

Chaque facteur de variabilité a un impact néfaste sur la performance de classification, mais leur impact cumulé est généralement moins grand que la somme de leurs impacts

individuels en termes de chute de précision.

#### 4.4.1.3 Avantages de la méthode RC par rapport aux autres modèles abstraits pour la CPDM

Comme le RC possède l'avantage d'être entraîné rapidement, il est possible de faire des entraînements spécifiques du modèle pour chaque patient lors d'une acquisition clinique. Cela permettrait de rendre le modèle plus spécifique à chaque patient et d'éliminer la perte de précision reliée à une variabilité de patients, comme observé entre IPart et MPart. Ceci requiert toutefois de demander à la personne de simuler des PDM spécifique à sa situation (corrigé et non-corrigé), ce qui implique qu'un clinicien bien formé soit présent lors de l'entraînement d'un tel dispositif.

De plus, il est possible d'utiliser le RC en temps réel. Un délai de classification est nécessaire afin de faire un moyennage suffisant pour optimiser la classification, de l'ordre de 350 points lorsque la fréquence de changement est grande dans la marche et autour de 1000 ou plus si la fréquence est plus faible. Cela représente un délai entre 2 cycles et 5 cycles de marche environ. Ce délai ne semble pas un problème dans le contexte d'application de rééducation à la marche, car un système se doit de donner une rétroaction avec une fréquence qui est raisonnable (répétition seulement après plusieurs cycles) afin d'éviter de donner une rétroaction trop fréquente au patient en rééducation.

De plus, le RC semble bien fonctionner même si l'on augmente la variabilité de la durée de la présence des PDM. Cela indique que même si la personne varie de manière irrégulière son patron de marche, le système restera performant.

Finalement, l'avantage principal des RC, tout comme les méthodes ANN, est leur faible quantité de calcul nécessaire en mode opération. Ces algorithmes sont donc facilement implémentables sur un microcontrôleur, sans besoins de connexion internet à une base de données ou d'un ordinateur externe.

#### 4.4.2 Comparaison avec des travaux antérieurs de modèle AI

Les tâches de classification par ANN trouvées dans la littérature sont assez différentes de la nature de la tâche faite dans notre application. Aucun exemple dans la littérature ne s'intéresse à classer 5 PDM en temps réel. La classification se limite généralement à une discrimination de 2 [107, 110] ou de 3 [7] pour faire un diagnostic.

La classification binaire faite par [107] obtenait une précision de l'ordre de 95 % sur la détection d'une condition neuro-psychologique sur une population de 44 individus selon une seule vitesse de marche par participant. La tâche de classification faite par [7] (exemple dans la figure 4.2) a l'objectif de classer trois conditions parmi une grande variabilité de patients (69) et pour trois vitesses de marche différentes, et réussit avec une précision de plus de 95 %. Un niveau de précision similaire est atteint pour le classement des PDM avec le RC lorsque les signaux sont traités par blocs individuels et pour une seule vitesse. L'utilisation des autres facteurs de variabilité dans les données (participant, vitesse) diminue significativement le niveau de performance du RC par rapport à tous ces résultats. Il n'est toutefois pas possible de comparer quantitativement la difficulté entre les tâches.

La variabilité dans les données peut aussi provenir d'un changement de la condition physique du patient selon différents jours. C'est ce qu'on remarque dans les travaux de [106] où la précision sur une tâche d'identification d'identité chute de 95 à 85 % lorsque les



données de la période d'entraînement et de test ont été enregistrées dans des journées différentes. Il est donc possible qu'un impact similaire puisse être observé sur la performance du RC si celui-ci est utilisé sur plusieurs jours après l'entraînement.

### 4.4.3 Limitations et perspectives futures d'étude

---

#### 4.4.3.1 Enjeux d'entraînement du RC pour la CPDM

Un défi à considérer est d'établir des données d'entraînement correctement labélisées sur la présence et l'absence de chacun des PDM. Cela requiert l'avis d'un expert qui doit regarder la marche en direct ou en post traitement avec une séquence vidéo. Le verdict sur la présence d'un mouvement de marche par un expert reste une décision subjective. Une approche plus robuste serait d'utiliser des mesures par marqueurs et caméras infrarouges en laboratoire.

Les signaux d'accélération peuvent alors être labélisés sur d'autres mesures expérimentales comme sur un seuil des forces au sol ou de l'angle de progression du pied. Cette seconde alternative restreint toutefois l'acquisition des données d'entraînement uniquement au laboratoire.

Les données auraient aussi avantage à être enregistrées et labellisé à partir d'environnements écologiques, dans des situations similaires à ce que le patient aura comme environnement. Les données d'entraînement étant plus proche de la réalité auront clairement le potentiel de faire de meilleur classification.

Éventuellement, il serait aussi possible de combiner les deux méthodes d'entraînement, en utilisant à la fois des données labélisées avec des mesures quantitatives précises avec marqueur IR en laboratoire (dans un contexte moins écologique), puis de faire une autre phase d'entraînement en contexte plus écologique, avec un clinicien labélisant les données selon des choix simples de type "mouvement correct" ou "mouvement incorrect". Les données des deux contextes pourraient être utilisées dans une régression RR selon des pondérations ajustables, comme proposée dans la méthode d'entraînement par kernel abordée dans le chapitre 5.

#### 4.4.4 Fréquence de variation du PDM

---

Le RC obtient une baisse de performance pour détecter des changements trop fréquents de PDM. En effet, si la variation du PDM se fait de manière trop rapide, la prédiction du RC est moins bonne selon les résultats de la figure 4.17. Nous avons néanmoins pu montrer que la classification est bonne au-dessus d'intervalles de modification du PDM de l'ordre de 5-10 cycles de marche. Si la longueur des intervalles de la présence d'un PDM est plus longue, le nombre de points d'intégration optimal est plus grand et moins critique pour assurer une bonne classification. Dans un contexte où la personne suit un traitement clinique pour modifier sa marche, il est plus réaliste que la personne oublie de conserver le PDM prescrit en clinique après une période au minimum de 5 cycles. Cette limitation ne devrait donc pas empêcher l'utilisation du RC dans un contexte clinique.

Plusieurs autres aspects de variabilité pourraient être pris en compte dans le cadre de futures études. Plusieurs conditions variées (extérieurs, milieu avec obstacles) pourraient être considérées de manière à évaluer les performances dans des contextes d'application plus réalistes que sur un tapis roulant instrumenté. D'autres types de conditions comme

les virages ainsi que la montée des pentes abruptes peuvent faire varier la nature des données et la performance de classification.

## 4.5 Conclusion

---

En conclusion, ce chapitre a démontré que la CDFM avec le RC est tout à fait possible. Dans un contexte d'utilisation où l'on veut détecter les PDM d'une seule personne marchant à sa vitesse naturelle en continu, la prédiction du RC sera exact 82 % du temps. Cela est suffisant pour établir une classification fiable des PDM et donner une rétroaction après un minimum de 5 cycles de marche. Des facteurs de variabilité comme la vitesse et la fréquence de changement du PDM peuvent diminuer la précision, mais sont moins probables d'arriver en application clinique.

# Chapitre 5

## Prédiction des forces au sol

### Éléments clé

Ce chapitre décrit le processus d'utilisation du RC pour prédire des signaux de force (PFS) au sol dans trois orientations. L'entraînement de cette tâche est basé sur des mesures de force de réaction au sol d'un tapis roulant instrumenté.

- La PFS au sol ou de tout autre signal biomécanique basé sur l'accélération permet d'obtenir des mesures pertinentes pour le traitement de patients qui sont difficiles à obtenir dans un contexte extérieur à un laboratoire.
- Les modèles utilisés dans la littérature pour la PFS sont généralement basés sur des modèles physiques, statistiques ou abstraits (AI) et ont une précision de l'ordre de 5-30% d'erreur par rapport au poids (BW) du participant pour la prédiction des forces verticales.
- Le modèle RC permet de générer une PFS dans l'axe vertical (V), médio-latéral (ML) et antéropostérieur (AP) avec une erreur respectivement de 14.5 %, 1.7 % et 3.7% du BW avec une méthode d'entraînement standard (uniforme sur toutes les données) avec une accélération unidimensionnelle sur le dessus du pied, dans l'axe antéro-postérieur.
- L'entraînement par Kernel a été développé pour améliorer la performance du modèle RC. Cela consiste à faire un entraînement où l'importance des données est non-uniforme (importance pondérée selon la durée du cycle de marche, la durée de phase d'appui ou la proportion de la phase d'appui sur le cycle de marche).
- L'entraînement par Kernel donne des résultats de prédiction des GRF dans les axes V, ML et AP de 13.3%, 1.6% et 3.3% d'erreur par rapport au BW, soit une petite diminution sur l'erreur moyenne globale. Cette amélioration est toutefois très significative pour les prédictions de signaux dans les extrêmes de la variabilité des mesures. En d'autres mots, le kernel training permet de mieux estimer les signaux qui ont une amplitude loin de la moyenne par rapport à l'ensemble des mesures.
- La précision du modèle RC pour la prédiction reste toutefois insuffisante pour détecter des variations dans les données de forces au sol observable lors d'un traitement clinique. Les variations attribuables à un changement de technique de marche induisent une modification relativement subtile sur l'amplitude des forces au sol, d'au plus 100 à 150 N (soit 13-20 % du BW pour le poids moyen des patients de 75 kg). Différentes possibilités d'amélioration du modèle sont considérées.

## 5.1 Contexte

---

### 5.1.1 Contexte d'application clinique

---

L'AM exploite une grande variété de signaux de mesure, reliés aux forces de réactions au sol, à l'activité musculaire et aux mouvements linéaires et angulaires des différents membres du corps [114]. La plupart de ces signaux ne sont mesurables que dans un environnement contrôlé en laboratoire, avec des équipements standardisés comprenant une plateforme de force instrumentée, des marqueurs de position infrarouges et des capteurs électromyographiques. Pour des raisons d'encombrement, il n'est pas possible d'utiliser ces équipements dans un contexte réel, hors du laboratoire. L'estimation indirecte de signaux biomécaniques non mesurables en contexte écologique représente donc une application courante de l'utilisation des capteurs inertiels en analyse de la marche et de la course.

Les forces de réactions au sol sont essentielles pour l'évaluation des conditions musculaires et du risque de blessures d'un patient [1]. Ces mesures sont aussi utilisées pour la conception de joints artificiels ou de prothèses, dont le design mécanique est ajusté pour chaque patient selon les forces impliquées [115]. L'allure de la courbe du signal des forces de réaction verticale au sol est aussi un indicateur de l'état physique d'une personne. En effet, le balancement naturel du centre de masse pour une personne ayant une bonne stabilité et une bonne mobilité produit un transfert de poids qui améliore l'efficacité de la marche et qui s'observe via la force de réaction du pied normale au sol [1]. La valeur maximale de force au sol permet aussi d'évaluer la performance sportive d'un coureur au sprint [116] voulant optimiser son accélération linéaire lors du départ de la course. Le risque de chute d'une personne âgée peut aussi être anticipé avec des signaux de forces de réaction au sol [117] ou une mesure du dégagement au sol du pied [33, 118].

Des équipements encombrants et onéreux, comme un tapis roulant ou une piste instrumentée, étant nécessaires pour évaluer ces mesures, plusieurs références de la littérature sont dédiées à l'estimation indirecte des signaux de forces au sol par des capteurs IMU. Ce chapitre aborde donc utilisation du RC pour d'évaluer les forces de réaction au sol à partir de signaux d'accélération mesurés sur la jambe.

Toutefois, d'autres signaux biomécaniques mesurables en laboratoire uniquement, sont pertinentes en clinique, comme les forces et les moments intersegmentaires [115] ou les forces de contact dans les articulations [28], tel que mentionné dans la section 1.2.2. Ceux-ci sont calculés à partir de modèles biomécaniques comme OpenSim [119] ou AnyBody [120], basés sur les mesures des forces au sol et de la cinématique du corps. Les forces de contact dans les articulations sont calculées en exploitant aussi une mesure de l'activité musculaire, mesurée avec des électromyogrammes de surface [28]. Ce travail a donc été fait dans l'optique de pouvoir transférer la méthodologie RC et la tâche de PFS pour la prédiction de ces autres mesures, comme les signaux du moment abducteur du genou (KAM) et des contraintes internes (introduits dans la section 1.2.2), étant relativement similaires à celle des forces au sol en termes de contenu fréquentiel. Toutefois, ces signaux ne sont pas disponibles en quantité suffisante pour l'instant pour établir une base de données d'entraînement avec le RC.

### 5.1.2 Revue de la littérature sur l'estimation des forces au sol

L'extraction des signaux de force au sol basée sur les signaux IMU durant la marche et la course est une question de recherche en biomécanique fréquemment abordé dans la littérature. Cette section présente plusieurs modèles déjà développés pour la prédiction des GRF. Ceux-ci sont présentés selon trois familles : les modèles biomécaniques, les modèles statistiques, et les modèles par apprentissage. Un sommaire de ces méthodes et de leur performances est présenté dans la tableau 5.1. Les métriques de l'erreur de prédiction de ces différents modèles prennent différentes formes. Elles sont principalement données sous forme d'erreur absolue moyennée (MAE), mais dont les unités sont variables selon les publications, notamment exprimées en unités de force (en newton),

$$MAE_F = \frac{1}{N} \sum_{i=1}^N |y_i^t - y_i^p| (N), \quad (5.1)$$

où  $i$  est l'indice de temps,  $N$  est le nombre de points dans le signal prédit  $y_i^p$  et de référence  $y_i^t$ . L'erreur est aussi exprimée en valeur relative à l'amplitude des signaux de références mesurés (target-relative,  $tr$ ),

$$MAE_{tr} = \frac{MAE_F}{\max(y^t(i)) - \min(y^t(i))} * 100 (\%), \quad (5.2)$$

et dans [117], on utilise une erreur relative à la mesure de référence et à la mesure prédite (target-prediction-relative,  $tpr$ ) :

$$MAE_{tpr} = \frac{MAE_F}{\frac{\max(y^t(i)) + \max(y^p(i))}{2} - \frac{\min(y^t(i)) + \min(y^p(i))}{2}} * 100 (\%). \quad (5.3)$$

Dans certains cas, les modèles sont conçus pour prédire spécifiquement le maximum (peak) du VGRF. L'erreur est estimée en comparant uniquement les maxima des valeurs de référence et prédites :

$$MAE_{pk} = \frac{1}{N_c} \sum_{c=1}^{N_c} |\text{peak}_c^t - \text{peak}_c^p|(N), \quad (5.4)$$

où  $N_c$  représente le nombre de cycles comparés et  $\text{peak}_c$  représente la valeur peak trouvée pour le cycle d'indice  $c$ . La métrique peut finalement être exprimée en proportion de la masse de la personne (BW),

$$MAE_{BW} = \frac{MAE_F}{BW} (N/N), \quad (5.5)$$

où  $BW$  représente le poids en Newton du participant.

#### 5.1.2.1 Modèles physiques

Les modèles physiques permettent d'approcher le problème avec une intuition de la physique du mouvement. Les modèles analytiques les plus simplistes utilisent la loi de Newton idéalisée, en faisant l'hypothèse que les forces de réaction au sol sont directement proportionnelles aux accélérations du centre de masse (CM). Dans [122], on détermine

Réf.	Tâche	Type IMU	Position	Participant	Méthode	Reported RMSE
<b>Modèle physique</b>						
[121]	Course	3D Acc	Medial tibia	10 Ath	Loi de newton 1D	VGRF : 7 % ( $MAE_{pk}$ )
[122]	Départ, changement de direction en marche	3D Acc, 3D Gyro,	Sacrum	15 HA	Loi de newton 1D	VGRF : 37.70 à 182.92 N ( $MAE_F$ )
[123]	Course,	3D Acc.	Upper back, T2	17 HA	Loi de newton 1D	VGRF : 11 à 23% ( $MAE_{pk}$ )
[124]	Marche	1D Acc	C7	6 HA (men)	Loi de newton 1D avec correction	VGRF 4-8 % ( $MAE_{tr}$ )
[115]	Marche	3D Acc, 3D Gyro	Trunk, thigh, shank, foot	2 HA	Modèle analytique de newton 3D	VGRF : $\approx$ 66 N ( $MAE_F$ )
[125]	Marche	3D Acc, 3D Gyro, 3D Mag.	Full Body	11 HA (men)	Modèle biomécanique	VGRF : 0.063 à 0.09 ( $MAE_{BW}$ )
[114]	Marche	3D Acc, 3D Gyro, 3D Mag.	Full body	9 HA (women)	Modèle musculosquelettique	VGRF : $0.156 \pm 0.125$ ( $MAE_{BW}$ )
<b>Modèle statistique</b>						
[116]	Sprint, course	3D Acc., 3D Gyro., 3D Mag.	Shank	3 Ath	Model de regression linéaire empirique	VGRF : 33.32% ( $MAE_{pk}$ )
[117]	Marche	3D Acc.	L5, C7, Forehead	9 HA	Orthogonal forward regression algorithm	VGRF : 5.0% ( $MAE_{tr}$ )
[126]	Course	3D Acc.	Medial tibia	3 HA	Corrélation linéaire et logarithmiques	VGRF : 6% ( $MAE_{pk}$ )
[127]	Marche, course	3D Acc.	Iliac crest of the right hip	39 HA	Modèle de corrélation linéaire	APGRF : 17.8% VGRF : 8.3 % ( $MAE_{pk}$ )
[128]	Marche, course	2D Acc.	Iliac crest of the right hip	35 HT	Modèle de corrélation linéaire	VGRF $9.0 \pm 4.2\%$ ( $MAE_{pk}$ )
<b>Modèle par apprentissage</b>						
[129]	Course	3D Acc., 3D Gyro., 3D Mag.	Pelvis, shank.	8 Ath	2 FNN combinée	VGRF : 0.27 ( $MAE_{BW}$ )
[130]	Marche	3D Acc.	Shank	17 HA	1 FNN multicouche	VGRF : 5.1 MLGRF : 1.4 APGRF : 1.9 ( $MAE_{BW}$ )
[131]	Marche, course	1d AP acc. (TS)	TS	7 HA (men)	FNN simple couche	VGRF : 0,1-0,18 ( $MAE_{BW}$ )

Abbréviations : Ath = Athlètes, HA = Healty adult, HT = Healty teenager

**Table 5.1** – Résumé des différents modèles de prédictions des forces au sol lors de la marche et la course.

préliminairement l'orientation du corps à l'aide des données du magnétomètre et du gyroscope d'un IMU attaché près du sacrum. Les forces de réactions au sol sont ensuite estimées via l'accélération près du CM,  $a_{cm}$ ,

$$VGRF = ma_{cm}, \quad (5.6)$$

où  $m$  est la masse du participant, en kg. Les résultats de ce modèle physique idéalisé présentent des erreurs pouvant aller jusqu'à 160 N, soit une erreur de 23% du BW pour une personne de 70 kg, ce qui reste important. En général, cette estimation des forces de réaction basée sur un multiple de l'accélération du CM a tendance à surestimer les forces de réactions au sol [124]. Un autre modèle appliquant la 2<sup>e</sup> loi de Newton basée sur l'accélération sur une seule position sur le corps montre des résultats plus intéressants [124], en ajoutant un paramètre empirique associé à la répartition du corps,

$$VGRF(t) = m\gamma(t)a_{v-C7}(t) + mg \quad (5.7)$$

où  $VGRF(t)$  représente la force de réaction au sol verticale,  $a_{v-C7}(t)$  est l'accélération verticale mesurée près du centre de masse (à la position C7 de la colonne vertébrale),  $m$  est la masse du participant, en kg, et  $g$  est l'accélération gravitationnelle. Le paramètre  $\gamma(t)$  représente une pondération sur la masse en lien avec la répartition de la masse du corps. Cette variable  $\gamma(t)$  dépend de la phase temporelle du mouvement dans le cycle de marche, appelé le *scaled accélération model*, permettant de considérer que le corps fait un mouvement périodique continu pour chaque cycle de marche qui change la répartition de la masse du corps. La valeur du paramètre  $\gamma$  variait environ entre 0.9 et 1.1 durant l'ensemble du cycle de marche. Cette méthode d'ajustement de pondération avec un paramètre  $\gamma(t)$  permettait de réduire l'estimation des forces de réaction au sol durant la marche d'environ 3% de  $MAE_{tr}$  (4-8 % avec ajustement, vs 5 - 11 % sans ajustement), ce qui est une amélioration très significative.

Un modèle de la dynamique 3D du corps [115] permet d'estimer plusieurs mesures indirectes, non seulement les forces de réaction au sol, mais aussi l'estimation des angles et des forces intersegmentaires. Celui-ci donne une erreur assez faible, de 45-56 N d'erreur pour l'estimation de la force sous le talon, soit environ 6-8 % du BW (pour une personne de 70kg). Toutefois, cette méthode nécessite une grande quantité (7) de capteurs positionnés sur les cuisses, les tibias, les pieds et le sternum, combinés à des mesures anthropométriques.

Les autres modèles biomécaniques [125] et musculosquelettiques [114] nécessitent aussi une grande quantité d'IMU, ainsi qu'un temps de calcul généralement assez long. Les résultats obtenus dans ces publications ne semblent pas apporter de précision additionnelle significative comparativement aux autres méthodes.

### 5.1.2.2 Modèles statistiques ou de régression

Les modèles statistiques retrouvés dans la littérature [116, 126–128] ont principalement été conçus pour faire une régression entre l'amplitude des accélérations et les maxima (pics) des forces de réaction au sol dans chaque cycle, sans faire une estimation continue. En effet, plusieurs applications dans le sport ou en clinique ne s'intéressent qu'aux valeurs maximales des forces au sol durant le cycle de marche. Par exemple, dans [116], l'estimation des pics de force au sol lors d'un sprint d'un coureur est faite avec,

$$\text{peak}_c^p(n) = c_1(n)a_x + c_2(n)a_y + c_3(n)a_z \quad (5.8)$$

où chacun des paramètres  $c_1(n)$ ,  $c_2(n)$  et  $c_3(n)$  sont des fonctions linéaires par rapport au nombre de cycles de course effectués ( $c_i(n) = m_i n + b_i$ ), ou  $n$  est l'indice du cycle de pas effectués durant le sprint et  $m_i$  et  $b_i$  sont des paramètres empiriques. Les variables  $a_x$ ,  $a_y$  et  $a_z$  représente l'amplitude de l'accélération selon les trois orientations. Les résultats montrent une très bonne corrélation pour l'un des coureurs (environ 5% d'erreur sur l'amplitude maximale de la force au sol durant le cycle), mais une mauvaise corrélation dans l'ensemble des autres coureurs, soit de l'ordre de 30% d'erreur. Ce modèle de corrélation linéaire présente donc une robustesse assez faible.

Une autre approche exploitant des fonctions polynomiales avec des coefficients empiriques [117] permet de reconstruire le signal de force en continu. En utilisant une grande quantité de fonctions orthogonales, les forces a sol peuvent s'estimer comme

$$y(t) = \mathcal{F} [\mathbf{u}(t-1), \mathbf{u}(t-1), \dots, \mathbf{u}(t-n_u)], \quad (5.9)$$

où  $\mathbf{u}(t)$  représente un vecteur d'entrée au temps  $t$  comprenant 6 mesures d'accélération provenant de 2 IMU 3 axes,  $\mathbf{u}(t) = [a_x^1(t), a_y^1(t), a_z^1(t), a_x^2(t), a_y^2(t), a_z^2(t)]$ .  $\mathcal{F}$  représente une fonction multivariable de type polynomiale, et  $n_u$  est le délai maximal considéré, égal à 18 dans les résultats présentés dans [117]. Le nombre de coefficients de régression de ce modèle était très grand, soit de 5778 [117]. La prédiction de ce modèle est très bonne, avec une erreur relative à l'amplitude totale du signal de l'ordre de 5%, et ce pour tous les 9 participants adultes en santé inclus dans l'étude. Toutefois, le modèle étant testé sur un échantillon relativement petit et le nombre de coefficients de régression étant très élevé, il est probable que ce modèle soit très spécifique à ces données et qu'il ne puisse pas se généraliser sur de nouveaux participants (risque d'overfitting). Le manque d'information et de test pour évaluer la performance de la méthode dans un contexte écologique et sur une plus grande variété de participants est en effet l'une des limitations indiquées par [117].

### 5.1.2.3 Modèles abstraits / AI

Les méthodes par apprentissage provenant de la littérature étaient de type FNN, sans capacité de mémoire, comme expliqué dans la section 2.1.1. Ces modèles sont entraînés par rétropropagation du gradient (voir section 2.1.1.5). La représentation des signaux d'accélération est un vecteur uniforme basé sur les données du IMU

Chaque prédiction du FNN est basée sur un vecteur de données d'entrées qui inclut des mesures d'un IMU, incluant l'accélération et la vitesse angulaire (gyroscope). Trois méthodes de FNN sont abordées ici et possèdent une grande différence de performance (entre 5% et 30% du BW).

L'approche utilisée par [129] est de combiner les données brutes du capteur IMU dans un vecteur en incluant les données des différentes orientations et de deux positions sur le corps, soit, un vecteur d'entrée du FNN comme suit,

$$\mathbf{u}(t) = [a_x^1, a_y^1, a_z^1(t), \omega_x^1(t), \omega_y^1(t), \omega_z^1(t), a_x^2, a_y^2, a_z^2(t), \omega_x^2(t), \omega_y^2(t), \omega_z^2(t)] \quad (5.10)$$

où  $a_j^i$  et  $\omega_j^i(t)$  représentent l'accélération et la vitesse angulaire dans l'axe  $j$  et à la position  $i$ . Deux FNN étaient utilisés pour traiter ces données. Un premier (FNN<sub>1</sub>) permettant d'estimer les angles des articulations du corps, et un second (FNN<sub>2</sub>) permettant de faire l'estimation des forces au sol en exploitant à la fois les données  $\mathbf{u}(t)$  de l'IMU, ainsi que les données d'angles prédites par le FNN<sub>1</sub>. Les deux FNN étaient constitués de deux couches cachées, respectivement de 250 et de 100 neurones. Le FNN est donc entraîné en utilisant la rétropropagation du gradient point par point dans les données. En mode opération, le FNN calcule une prédiction pour chaque indice de temps de manière indépendante,

$$y^p(t) = \text{FNN}_2(\mathbf{u}(t), \text{FNN}_1(\mathbf{u}(t))), \quad (5.11)$$

où  $y^p(t)$  représente les forces au sol prédites au pas de temps  $t$ . Cette méthode avait toutefois une mauvaise performance, avec un RMSE de 27 % du BW.

Le modèle proposé par [130] utilisait une approche très similaire, mais obtenait de bien meilleurs résultats de prédiction. La méthode incluait dans le vecteur d'entrée d'autres



composantes associées aux dérivées et aux intégrales des mesures IMU en fonction du temps,

$$\mathbf{u}(t) = [a_x(t), \dots, \omega_x(t), \dots, \dot{a}_x(t), \dots, \dot{\omega}_x(t), \int a_x(t)dt, \dots \int \int a_x(t)dt^2], \quad (5.12)$$

ce qui revient à inclure une information temporelle des mesures IMU dans les données d'entrées. Pour chaque pas de temps de mesure des données IMU, le vecteur d'entrée possède donc une dimension égale six (mesures de base de l'IMU) multipliée par 3 (le nombre de dérivés et intégrales considérées dans cette étude, incluant une intégrale simple, une intégrale double et une dérivée simple), soit 18 entrées. Le vecteur d'entrée comprenait aussi une mesure de la durée de la phase d'appui et une position relative (%) des données du vecteur dans la phase d'appui. Une approche similaire fut notamment utilisée par [132] afin de prédire le moment abducteur du genou (KAM). Ce travail obtenait une bien meilleure corrélation entre les maxima de la prédiction et les valeurs maximales du KAM de référence.

L'approche proposée par [131] utilise un FNN avec une manière totalement différente de définir les données d'entrées. Les accélérations dans l'axe antéropostérieur (AP) du pied sont segmentées avec l'indice du début et de la fin de chaque cycle, correspondant à la détection du HS. Les accélérations AP sont ensuite interpolées selon une longueur standard de 100 points, correspondant aux données d'entrée du FNN,

$$\mathbf{u} = [a(t_1), a(t_2), \dots, a(t_{99}), a(t_{100})], \quad (5.13a)$$

$$\mathbf{y} = \mathbf{FNN}(\mathbf{u}) = [y(t_1), y(t_2), \dots, y(t_{99}), y(t_{100})]. \quad (5.13b)$$

La sortie  $\mathbf{y}$  du FNN est donc un vecteur de 100 points représentant le signal de force sur l'ensemble du cycle de marche. Cette méthode établit la corrélation de la force de réaction verticale au sol uniquement avec l'accélération du même cycle de marche. Les résultats de cette méthode possèdent une erreur moyenne d'environ 0.15 BW, ce qui est moins précis que les résultats obtenus avec la méthode [132], avec un FNN utilisant le vecteur d'entrée de l'équation (5.12). D'un point de vue applicatif, cette méthode possède le désavantage d'un délai minimal de un cycle de marche avant de pouvoir évaluer les forces au sol considérant que le vecteur d'entrée du FNN est construit avec un signal d'accélération mesuré sur l'ensemble du cycle. Cela limite légèrement son application en temps réel par rapport aux deux autres modèles FNN.

La méthode de FNN combinant en entrée les valeurs directes mesurées avec les valeurs intégrées et dérivées des signaux IMU [130] semblent donc obtenir les meilleurs résultats des modèles abstraits de la littérature, avec une erreur de l'ordre de 5% de la force de réaction verticale au sol.

### 5.1.3 Perspective du RC pour la PFS

En somme, parmi ces différentes méthodes de modèle physique, statistique ou d'apprentissage, les meilleures performances de prédiction possèdent une erreur de l'ordre de 5 % du BW (soit environ 35 N pour une personne de 70 kg) et les approches les moins performantes possèdent une erreur de l'ordre de 30 % du BW.

Aucune étude dans la littérature n'a utilisé la méthode RC pour la PFS. Le RC se distingue des autres méthodes trouvées notamment parce qu'elle est conçue pour faire une prédiction continue et en temps en temps réel et qu'il peut prédire plusieurs signaux en parallèle sans augmenter la complexité du modèle ou de l'entraînement. Le RC possède une faible quantité de calcul lors de son utilisation.

Comme cette tâche est directement liée à l'application du suivi de la progression de l'arthrose, il est pertinent d'évaluer la capacité de prédiction du RC sur les deux populations (patients avec arthrose (OA) et participants sains) dans cette étude.

#### 5.1.4 Objectifs du chapitre

---

Dans ce chapitre, nous évaluons les performances de prédictions du RC pour faire une prédiction des forces de réaction au sol verticales (V), antéropostérieures (AP) et médio-latérales (ML) sur une grande variété de données (27 participants sains et 18 participants avec de l'arthrose au genou), dans le but de :

- Comparer la performance de prédiction du RC avec les autres méthodes trouvées dans la littérature.
- Évaluer quelle est la meilleure position du capteur sur la jambe parmi celles testées.
- Trouver une piste d'amélioration de la méthode standard d'entraînement du RC basée sur l'utilisation d'information temporelle détectable avec les événements de marche et comparer les résultats avec la méthode standard.

De plus, on souhaite trouver une piste d'amélioration de la méthode standard d'entraînement du RC basé sur l'utilisation combinée de la méthode de détection des événements de marche.

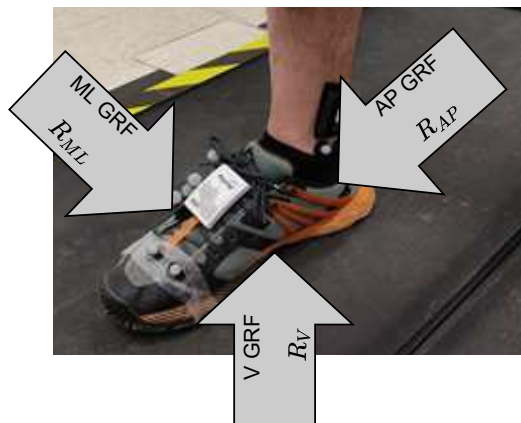
## 5.2 Méthodologie

### 5.2.1 Définition des données expérimentales

Les données expérimentales utilisées pour tester la tâche de PFS proviennent de la base de données CIRRI22. Comme les autres tâches de ce mémoire, les données d'entrées du réservoir sont les signaux d'accélération IMU. Les données d'entraînement étaient des mesures de forces au sol.

#### 5.2.1.1 Données de forces au sol

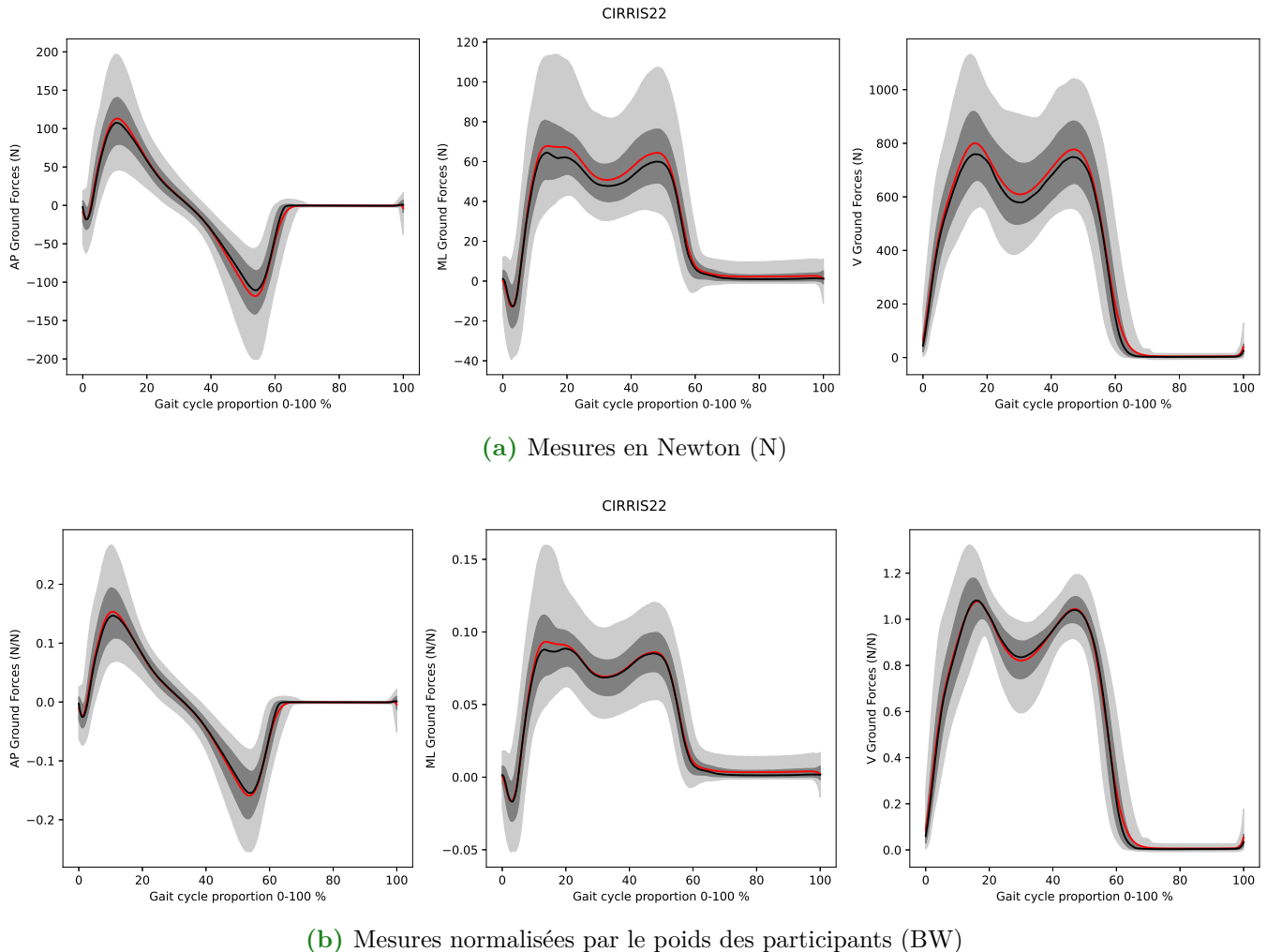
Les essais de marche ont été enregistrés sur un tapis roulant instrumenté mesurant les forces au sol. Les données de force et d'IMU ont été enregistrés en même temps, sur des systèmes d'acquisition indépendants, et resynchronisés ensuite avec une procédure de traitement expliquée dans l'annexe A. Le tapis roulant instrumenté permet d'acquérir une mesure des forces au sol selon les 3 axes : normale au sol (V), tangentiel au sol et perpendiculaire à la vitesse de déplacement du marcheur (ML), puis tangentielle au sol et parallèle à la direction de la vitesse du marcheur (AP), tel qu'illustré dans la figure 5.1.



**Figure 5.1** – Orientation des forces de réaction au sol mesurées par le tapis roulant instrumenté (Bertec, 1000 Hz).

Les courbes de force de réaction au sol selon les trois orientations sont illustrées dans la figure 5.2. On peut voir qu'il y a deux pics en amplitude des forces au sol verticales survenant durant la phase d'appui. Le premier pic survient lors du loading response (voir figure 1.1), entre 0 et 15 % du cycle de marche. Le maximum de force au sol verticales ressenti est généralement à la fin du loading response, selon le modèle de Perry (section 1.2.1). Le second pic est durant la phase du pre-swing, entre environ 50-60% du cycle de marche. Les signaux des forces verticales ont une amplitude maximale de l'ordre de 800 N lors du loading response, environ à 15% du cycle de marche, soit environ 1.2 fois le BW, et de l'ordre de 600-800 N pour la pre-swing, soit de l'ordre de 1.0-1.2 fois le BW. Dans l'axe AP, la force de réaction possède un pic maximal de l'ordre de 100 N lors du loading response et du pre-swing, soit environ 0.18 fois le BW. La force est positive lors du loading response, car le pied pousse vers l'avant (absorbition de l'impact), et négative lors du pre-swing, car le pied pousse vers l'arrière (propulsion). Les forces ML ont des

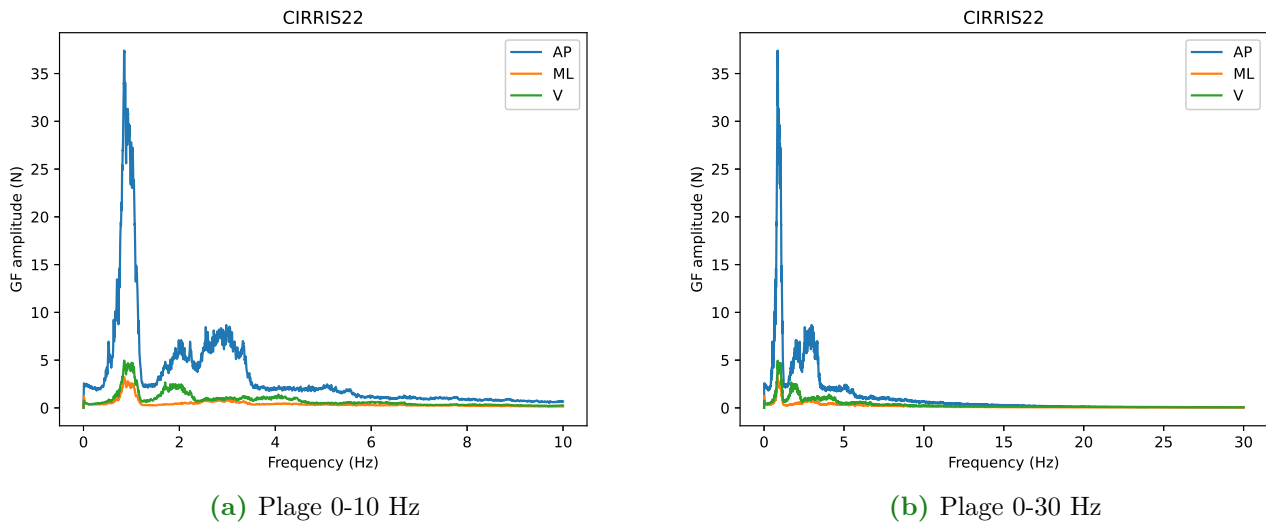
amplitudes maximales de l'ordre de 60-90 N (soit 0.09-0.15 fois le BW) lors de la phase de loading response, et de 60-75 N (soit 0.09-1.2 fois le BW) lors du pre-swing. Elles sont positives sur presque toute la phase d'appui, ce qui veut dire que le pied pousse vers les cotés extérieurs du corps dans l'axe perpendiculaire à la direction de la marche.



**Figure 5.2** – Allure des forces de réaction au sol pour l'ensemble des données de la base de données CIRRIS22. De gauche à droite, les forces au sol dans l'axe antéro-postérieur, médio-latéral et verticale. Ligne noire : la moyenne, Ligne rouge : la médiane, surface gris foncés [25-75e] percentile, surface gris pâle [5-95e] percentile.

Le contenu fréquentiel des signaux de forces est principalement entre 0 et 4Hz, comme illustré dans la figure 5.3. Contrairement aux signaux d'accélération, il n'y a pas de contenu au dessus de 6 Hz (voir figure 1.8) dans les signaux de force. Comme pour les accélérations, les deux premiers pics fréquentiels correspondent à la cadence de marche, soit à environ 1 Hz, et le second pic, qui se retrouve à environ 2 Hz, correspond à l'alternance entre la phase d'appui et la phase de swing [39].

On peut aussi voir une troisième bande fréquentielle assez large à environ 3 Hz pour les forces de l'axe verticale. Cela s'explique par la présence d'un creux dans les forces au sol verticales au tiers de la période du cycle de marche. Ce minimum local des forces au sol apparaissant au milieu de la phase d'appui est expliqué selon Perry [1] par le balancement



**Figure 5.3** – Contenu fréquentiel dans les signaux de force au sol dans la base de donnée CIRRIS22.

naturel du bassin lors de la marche afin d'obtenir une meilleure stabilité du corps. Ce mouvement de va-et-vient change la position du centre de masse et cause cette variation des réactions au sol dans l'axe V durant la phase d'appui [1]. Selon Perry, la présence du creux dans les forces au sol verticales est généralement un indicateur d'une bonne gestion de la fonction de stabilité. Les gens avec de la douleur ou des conditions physiques détériorant la stabilité ont un profil de force au sol plus plat durant la phase d'appui.

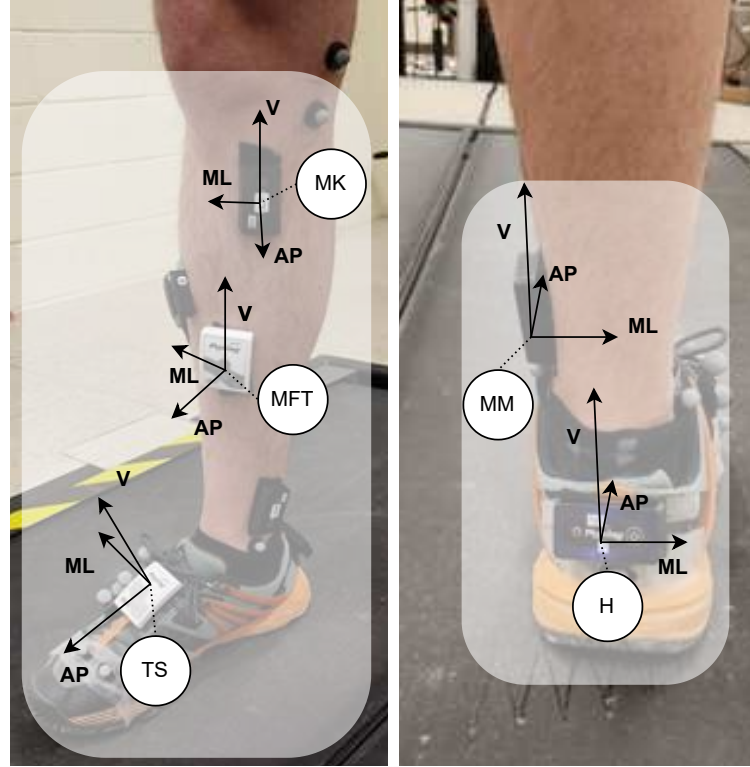
### 5.2.1.2 Données d'accélération

Dans la base de données CIRRIS22, l'accélération a été enregistrée à 5 positions différentes sur le membre inférieur dominant de chaque participant. Cela avait pour but de tester la robustesse du système à prédire les forces au sol selon différentes positions et d'évaluer la meilleure position possible. La position ainsi que les systèmes d'axes pour chacune des 5 positions sont représentés dans la figure 5.4. La performance dans la tâche PFS sera évaluée en utilisant les accélérations selon chacune des positions individuellement, afin de respecter les contraintes associées au MEMS-RC (voir la section 2.3).

## 5.2.2 Définition des données d'entrées et objectives du RC

### 5.2.2.1 Définition des données d'entrée

Le RC a été entraîné sur chacune des 5 positions individuellement. De plus, 7 combinaisons d'axes d'accélération ont été testées dans chacune des positions. Selon le choix des axes, l'entrée du RC est définie comme un vecteur de 2, 3 ou 4 entrées (nombre d'axes + 1 pour la valeur du biais). Les différentes définitions du vecteur d'entrée  $\mathbf{u}(t)$  sont représentée dans la tableau 5.2. L'objectif à long terme du projet étant de concevoir un capteur minimaliste se limitant à un seul capteur sur le corps, nous avons exclu la possibilité de combiner des données de deux capteurs différents. Ainsi, avec ces 5 positions et ces 7 axes



**Figure 5.4** – Positions et orientations des IMU dans la base de donnée CIRRS22.

différents, la tâche de PFS est donc évaluée selon 35 configurations différentes des données d'entrées.

	AP	ML	V	AP-ML	ML-V	AP-V	AP-ML-V
$\mathbf{u}(t_i)$	$\begin{bmatrix} 1 \\ a_{AP}(t_i) \end{bmatrix}$	$\begin{bmatrix} 1 \\ a_{ML}(t_i) \end{bmatrix}$	$\begin{bmatrix} 1 \\ a_V(t_i) \end{bmatrix}$	$\begin{bmatrix} 1 \\ a_{AP}(t_i) \\ a_{ML}(t_i) \end{bmatrix}$	$\begin{bmatrix} 1 \\ a_{ML}(t_i) \\ a_V(t_i) \end{bmatrix}$	$\begin{bmatrix} 1 \\ a_{AP}(t_i) \\ a_V(t_i) \end{bmatrix}$	$\begin{bmatrix} 1 \\ a_{AP}(t_i) \\ a_{ML}(t_i) \\ a_V(t_i) \end{bmatrix}$

**Table 5.2** – Définition des axes utilisés dans le vecteur d'entrée  $\mathbf{u}(t)$  du RC

### 5.2.2.2 Définition des données objectives

Les données objectives correspondent aux signaux de réaction au sol  $R_{AP}$ ,  $R_{ML}$ ,  $R_V$  (N) enregistrés par le tapis roulant.  $\mathbf{y}_p^t(t_i)$  est donc défini comme un vecteur colonne de 3 éléments. Afin de conserver une uniformité des amplitudes des signaux selon tous les participants, les signaux ont été normalisés par le poids du participant  $W_p$ (N), les signaux ont donc des unités de N/N, représentant la proportion des réactions au sol par rapport au poids de la personne,

$$\mathbf{y}_p^t(t_i) = \frac{1}{W_p} \begin{bmatrix} R_{AP}(t_i) \\ R_{ML}(t_i) \\ R_V(t_i) \end{bmatrix}. \quad (5.14)$$

### 5.2.3 Rappel de la méthode d'entraînement RR standard

Les premiers tests de la tâche de PFS ont été faits avec une régression linéaire standard, identique à celle utilisée pour le GED dans le chapitre 3 et le AD dans le chapitre 4. Une matrice de poids de sortie  $\mathbf{W}_{\text{out}}$  corrélant la valeur de sortie du réservoir aux signaux de force au sol est établie uniformément sur les données de l'ensemble des patients. La régression RR est appliquée sur l'ensemble des données (participants sains et OA) de la base de données CIRRI22. La base de données contenait 135 séries de données continues ( $N_{\text{TS}} = 135$ ), faites selon des enregistrements de marche d'environ 2 min, pour trois vitesses différentes par participant. Pour chaque enregistrement, on utilise le premier 70% de la durée de l'enregistrement pour faire l'entraînement. Les données d'entrées utilisées sont donc définies par l'ensemble  $\mathcal{U}$ ,

$$\mathcal{U} : \{\mathbf{U}_p = [\mathbf{u}_p(t_1), \mathbf{u}_p(t_2), \dots, \mathbf{u}_p(t_i), \dots, \mathbf{u}_p(t_{T_p})] \mid \forall 1 < p < N_{\text{TS}}\}, \quad (5.15)$$

où le nombre de pas de temps total d'entraînement dans une série de données  $p$  est noté  $T_p$ , où  $p$  représente l'indice de l'enregistrement. Les états  $\mathbf{X}_p$  du ESN sont calculés à partir de chaque série de données dans  $\mathcal{U}$ . Chaque série  $\mathbf{X}_p$  représente une matrice de taille  $N \times T_p$  dont chaque colonne représente un vecteur d'états de sortie du RC et chaque ligne est pour l'un des indices de pas de temps  $t_i$  calculé avec la série de données d'entrées  $U_p$ . L'ensemble  $\mathcal{X}$  regroupe toutes séries continues d'états  $\mathbf{X}_p$ ,

$$\mathcal{X} = \{\mathbf{X}_p = [\mathbf{x}_p(t_1), \mathbf{x}_p(t_2), \dots, \mathbf{x}_p(t_i), \dots, \mathbf{x}_p(t_{T_p})] \mid \forall 1 < p < N_{\text{TS}}\}. \quad (5.16)$$

Un ensemble  $\mathcal{Y}$  de  $N_{\text{TS}}$  séries temporelles de donnée objectives est construit avec les mesures de référence du tapis roulant,

$$\mathcal{Y} = \{\mathbf{Y}_p^t = [\mathbf{y}_p^t(t_1), \mathbf{y}_p^t(t_2), \dots, \mathbf{y}_p^t(t_i), \dots, \mathbf{y}_p^t(t_{T_p})] \mid \forall 1 < p < N_{\text{TS}}\}, \quad (5.17)$$

où  $\mathbf{Y}_p^t$  est une matrice de taille  $3 \times T_p$  cumulant les vecteurs de données objectives sur tous les indices de pas de temps des signaux objectifs. On utilise la méthode RR afin d'établir l'erreur minimale entre les états  $\mathbf{X}$  et les valeurs objectives  $\mathbf{Y}^t$ , tel que,

$$\mathbf{W}_{\text{out}} = \mathbf{Y}^t \mathbf{X}^T (\mathbf{X} \mathbf{X}^T + \gamma \mathbf{I})^{-1}, \quad (5.18)$$

où  $\gamma$  est le paramètre de régularisation permettant d'éviter l'overfitting, et  $\mathbf{I}$  est une matrice identité de taille  $N \times N$ . Ici les matrices  $\mathbf{X} \mathbf{X}^T$  représentant la sommation des matrices des états de sorties pré-multipliés pour chacun des enregistrements individuellement, tel que,

$$\mathbf{X} \mathbf{X}^T = \sum_{p=1}^{N_{\text{TS}}} \mathbf{X}_p \mathbf{X}_p^T, \quad (5.19)$$

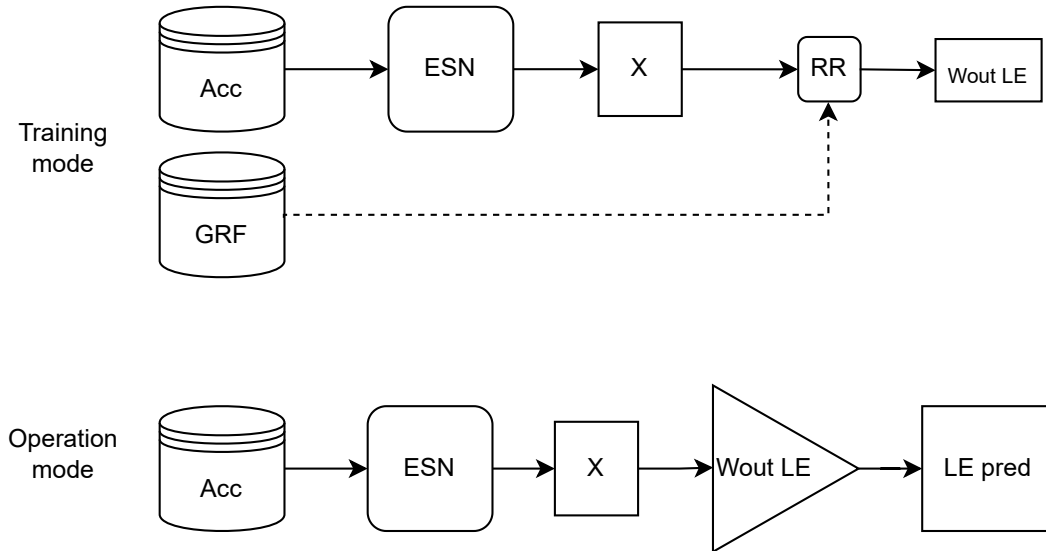
et la matrice  $\mathbf{Y}^t \mathbf{X}^T$  correspond à la somme des matrices pré-multipliées  $X_p$  et  $Y_p$  sur l'ensemble des séries de données,

$$\mathbf{Y}^t \mathbf{X}^T = \sum_{p=1}^{N_{\text{TS}}} X_p \mathbf{Y}_p^t. \quad (5.20)$$

Cela nous donne une matrice de sortie  $\mathbf{W}_{\text{out}}$  applicable en mode opération de manière uniforme sur l'ensemble des données d'entraînement. Par la suite, en mode opération, la prédiction se fait en appliquant la fonction ESN de manière uniforme sur de nouvelles données,

$$\mathbf{y}(t) = \mathbf{W}_{\text{out}} \text{ESN}(\mathbf{u}(t)) = \mathbf{W}_{\text{out}} \mathbf{x}(t) \quad (5.21)$$

La figure 5.5 illustre cette méthode en mode entraînement et en mode opération.



**Figure 5.5** – Méthode d'entraînement standard pour la PFS.

### 5.2.4 Méthode d'entraînement par kernel

Les résultats préliminaires ont montré que la prédiction des forces au sol avec le RC comportait une erreur moyenne relativement haute, soit de au-dessus de 0.151 (BW) pour les forces de réaction verticale comme montrée avec les résultats en noir dans la figure 5.19. Une seule matrice  $\mathbf{W}_{\text{out}}$  ne semble donc pas suffisante pour reconstituer des signaux de force au sol qui considèrent l'ensemble de la variabilité de paires de signaux de forces et d'accélération contenue dans les signaux expérimentaux dans CIRRI22.

Nous voyons dans la section 5.2.4.1 que les enregistrements utilisés durant l'entraînement comportent une variabilité importante de valeurs sur les différents paramètres temporels de la marche (cadence, durée et proportion de la phase d'appui), possédant une corrélation marquée avec les forces de réaction au sol. Les données dans une seule fourchette de ces paramètres sont donc moins "variables" en termes de signaux de force au sol que sur l'ensemble des données. Le chapitre 3 a montré qu'il est possible d'identifier ces paramètres temporels de la marche avec un RC en mode GED.

L'objectif ici est donc d'utiliser la tâche de GED pour évaluer ces paramètres temporels et ensuite de déterminer plusieurs matrices  $\mathbf{W}_{\text{out}}$  spécifiques à seulement une seule fourchette de ces paramètres (une seule cadence de marche, par exemple). En réduisant cette fourchette, on souhaite donc augmenter la spécificité du RC et réduire son erreur de prédiction. En d'autres termes, on permet au RC de se concentrer sur un seul type



marche, selon les métriques temporelles établies avec le GED, définies comme des variables de pondération. L'évaluation des forces au sol selon cette stratégie impose donc un premier traitement (GED) qui peut se faire en temps réel afin d'établir la valeur de la cadence (chapitre 3). Ensuite la section ci dessous explique comment exploiter une telle information pour modifier en temps réel la matrice de sortie du modèle  $\mathbf{W}_{out}$ , afin d'améliorer la prédiction. La méthode de prédiction par kernel peut donc aussi être exploitée en temps réel.

#### 5.2.4.1 Choix des variables de pondération

Plusieurs sources de variabilité de la marche font varier significativement les forces au sol. Certains paramètres temporels pouvant être identifiables avec la tâche de GED développée au chapitre 3 sont reliés aux amplitudes maximales de la force au sol dans les différentes directions. La durée du cycle de marche est un des paramètres ayant une corrélation (non linéaire) marquée avec les amplitudes de forces au sol, comme illustré dans la figure 5.6. La durée de la période de la phase d'appui (figure 5.7), ou la proportion de temps de la phase d'appui par rapport au temps total du cycle de marche (figure 5.8), sont d'autres paramètres qui possèdent une corrélation claire avec les forces au sol. On remarque que la population avec arthrose possède une durée de la phase d'appui plus courte sur leur membre affectée par rapport aux participants sains, cela est cohérent avec ce qui est observé dans la littérature sur les patients OA [95]. Le calcul de ces variables est fait avec les indices temporels des événements de marche des heel strike ( $i^{HS}$ ) et des toe-off ( $i^{TO}$ ) détectables avec la méthode de GED développée dans le chapitre 3. Les calculs de ces variables sont présentés dans le tableau 5.3.

Durée du cycle $\overline{\Delta T_{GCp}}$	Durée de la phase d'appui $\overline{\Delta T_{PAp}}$	Prop. de la phase d'appui $\overline{Prop_{PAp}}$
$\frac{1}{N_{GC}} \sum_{n=1}^{N_{GC}} (i_n^{HS} - i_{n+1}^{HS})$	$\frac{1}{N_{GC}} \sum_{n=1}^{N_{GC}} (i_n^{HS} - i_n^{TO})$	$\frac{1}{N_{GC}} \sum_{n=1}^{N_{GC}} \frac{(i_n^{HS} - i_n^{TO})}{(i_n^{HS} - i_{n+1}^{HS})}$

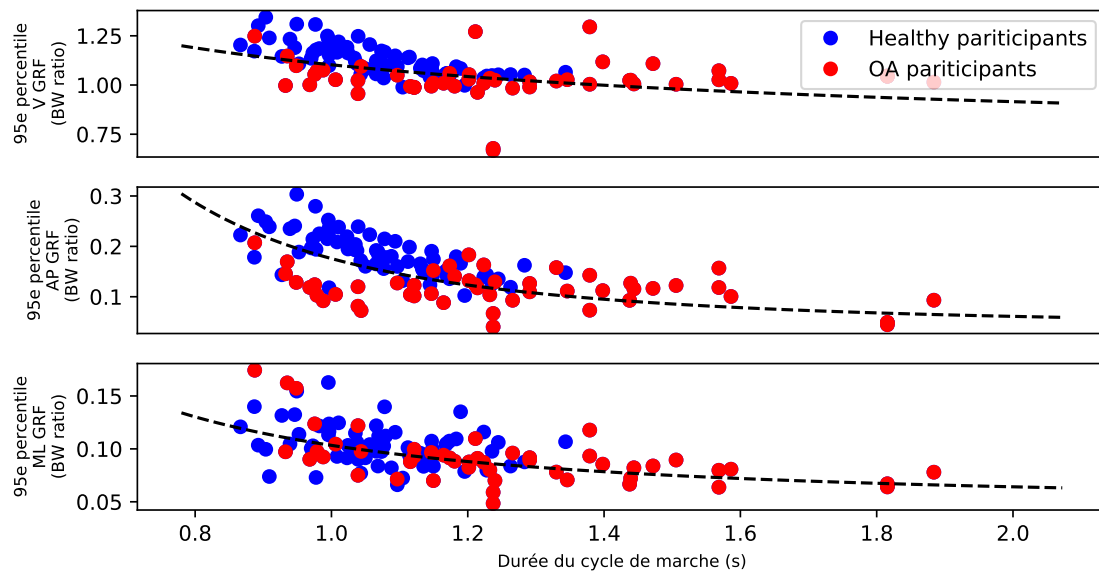
$N_{GC}$  = nombre de cycle de marche dans un enregistrement  
 $i_n^{HS}$  = indice de temps d'un événement de marche heel-strike au cycle n  
 $i_n^{TO}$  = indice de temps d'un événement de marche toe-off au cycle n

**Table 5.3** – Calcul des différents paramètres de pondération déterminés avec la tâche de GED pour la méthode Kernel training. Les variables sont moyennées sur l'ensemble de l'enregistrement de marche (environ 2 minutes de marche).

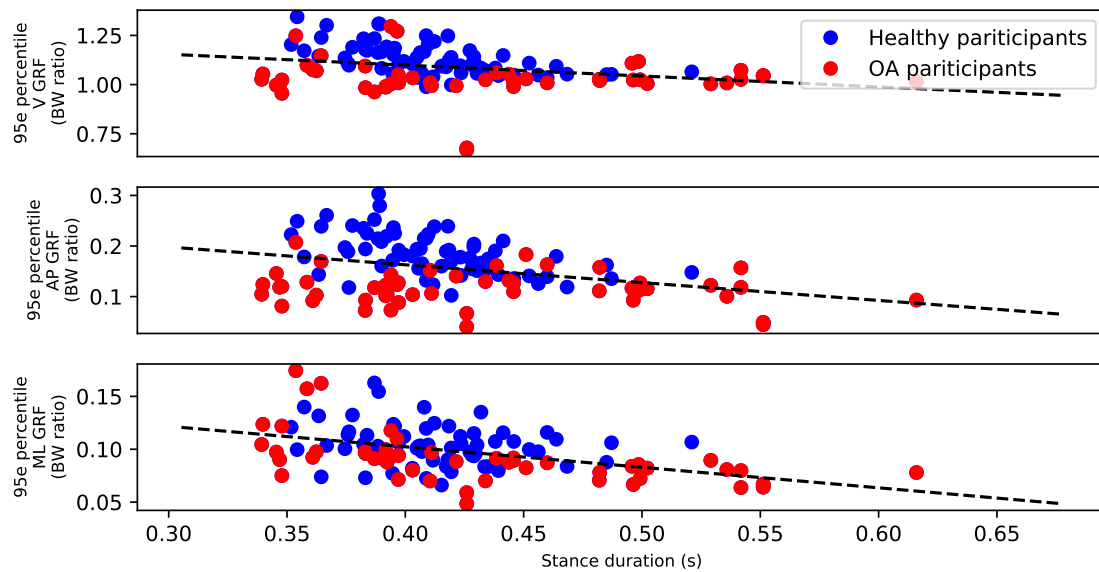
On peut noter ici que le paramètre de la cadence moyenne durant un cycle correspond au nombre de cycles par seconde effectués par un participant et se calcule avec l'inverse de la durée du cycle de marche :  $1/ \Delta T_{GCp}$ .

On voit donc que les paramètres temporels sont corrélés à l'amplitude des forces au sol, ce qui indique la pertinence de créer des matrices  $\mathbf{W}_{out_i}$  entraînées de manière spécifique sur une fourchette réduite de ces paramètres pour améliorer la prédiction.

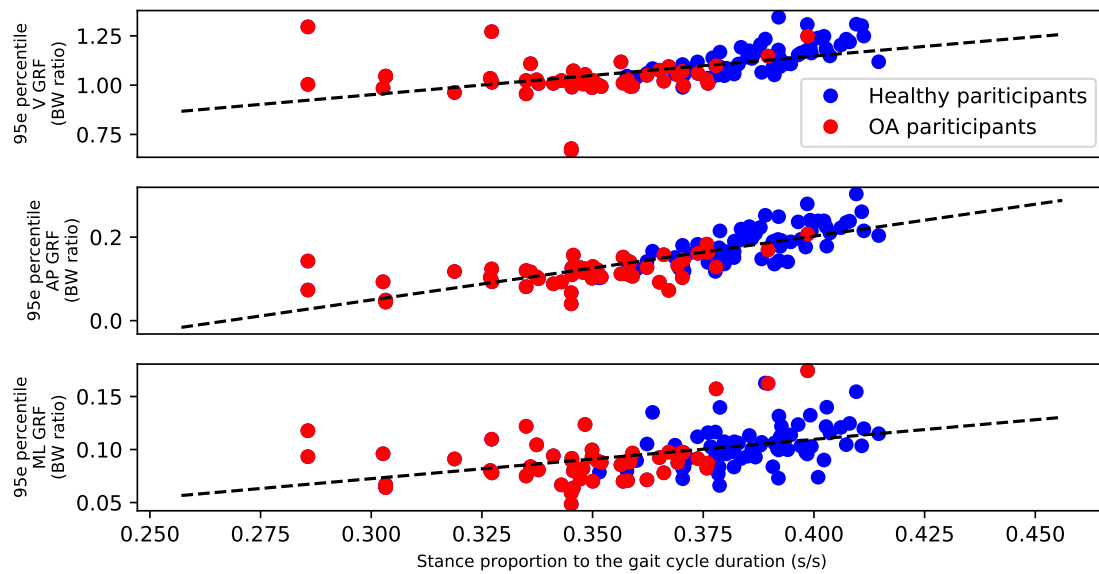
D'autres mesures de la marche pourraient éventuellement être utilisées comme paramètres de pondération, comme des métriques reliées à l'amplitude, le contenu fréquentiel, ou l'écart-type des signaux brutes d'accélération. Cela n'a toutefois pas été testé dans le cadre de ce travail, où nous nous sommes limité aux paramètres temporels mesurables avec la tâche de GED mentionné ci-dessus.



**Figure 5.6** – Amplitude au 95<sup>e</sup> percentile des signaux de forces au sol dans les 3 axes (V, AP et ML) en fonction de la durée du cycle de marche ( $\overline{\Delta T_{GCp}}$ ) pour les différents enregistrements du CIRRI22.



**Figure 5.7** – Amplitude au 95<sup>e</sup> percentile des signaux de forces au sol dans les 3 axes (V, AP et ML) en fonction de la durée de la phase d'appui ( $\overline{\Delta T_{PAp}}$ ) pour les différents enregistrements du CIRRI22.



**Figure 5.8** – Amplitude au 95<sup>e</sup> percentile des signaux de forces au sol dans les 3 axes (V, AP et ML) en fonction de la proportion de la phase d'appui sur la durée total du cycle de marche ( $\overline{Prop_{PAp}}$ ) pour les différents enregistrements du CIRRI22.

### 5.2.4.2 Méthode de pondération durant l'entraînement

La méthode d'entraînement par kernel consiste à faire plusieurs entraînements RR dont chacun nous permet d'obtenir plusieurs matrices de poids de sortie ( $\mathbf{W}_{\text{out}1}, \mathbf{W}_{\text{out}2}, \dots, \mathbf{W}_{\text{out}k}, \dots$ ) en utilisant une pondération des données d'entraînement. Chaque matrice serait donc "spécialisée" pour prédire une seule portion des données parmi l'ensemble. Cette pondération de poids s'applique en ajoutant une matrice de pondération de poids  $\mathbf{P}_{\text{pk}}$  dans le calcul de RR sur les ensemble de données d'entraînement  $\mathcal{X}$  et  $\mathcal{Y}$ , comme proposé par [72] et qui correspond à

$$\mathbf{W}_{\text{out}k} = \text{RR}(\mathcal{X}, \mathcal{Y}, \mathbf{P}_{\text{pk}}) = \left[ \sum_{p=1}^N \mathbf{Y}_p^t \mathbf{P}_{\text{pk}} \mathbf{X}_p^T \right] \left( \left[ \sum_{p=1}^N \mathbf{X}_p \mathbf{P}_{\text{pk}} \mathbf{X}_p^T \right] + \gamma \mathbf{I} \right)^{-1}, \quad (5.22)$$

où l'indice  $k$  représente l'indice des paramètres utilisés pour une certaine matrice  $\mathbf{W}_{\text{out}k}$ . La matrice strictement diagonale  $\mathbf{P}_{\text{pk}}$ , de taille  $T_p \times T_p$  correspond à une matrice de pondération des poids, tel que,

$$\mathbf{P}_{\text{pk}} = \begin{bmatrix} p_{\text{pk}}(t_1) & & & & \\ & p_{\text{pk}}(t_2) & & & 0 \\ & & \ddots & & \\ & & & p_{\text{pk}}(t_{T_p-1}) & \\ & 0 & & & p_{\text{pk}}(t_{T_p}) \end{bmatrix}, \quad (5.23)$$

et permet de multiplier la matrice  $\mathbf{X}_p$  et la matrice  $\mathbf{Y}_p^t$  de manière individuelle sur chaque indice de temps  $t_i$  (chaque colonne) par un facteur de pondération  $p_{\text{pk}}(t_i)$ . Si la valeur  $p_{\text{pk}}(t_i)$  à un pas de temps  $t_i$  est de 0, cela veut dire que ce pas de temps  $t_i$  sera ignoré dans le calcul des poids de sortie. Si le poids de pondération est de  $p_{\text{pk}}(t_i) = 1$  pour un pas de temps, et de  $p_{\text{pk}}(t_{i+1}) = 2$  pour le prochain pas de temps, cela signifie que le second pas de temps  $t_{i+1}$  sera pris en considération avec le double d'importance par rapport au premier pas de temps  $t_i$  [72]. Dans le cas d'un entraînement standard, l'importance est uniforme (les poids sont de 1 sur l'ensemble des pas de temps). Dans cette méthode, on accorde une importance plus grande à une portion spécifique des données en modifiant les valeurs de  $p_{\text{pk}}(t_i)$  pour chaque matrice  $\mathbf{W}_{\text{out}i}$ . La structure des données d'entraînement  $\mathbf{X}_p$  et  $\mathbf{Y}_p^t$  reste identique à celle utilisée dans la méthode d'entraînement standard.

### 5.2.4.3 Méthodes d'attribution des poids de pondération par kernel

Cette section explique comment définir les poids de pondération  $p_{\text{pk}}(t_i)$  dans l'entraînement de chacune des matrices  $\mathbf{W}_{\text{out}k}$ . Premièrement, les données pour chaque enregistrement  $p$  doivent être labélisées avec les paramètres de pondération  $V_p$  sélectionnés. Comme mentionné dans la section 5.2.4.1, les variables de pondération définies dans ce projet sont la durée du cycle  $\Delta T_{\text{GC}p}$ , la durée de la phase d'appui  $\Delta T_{\text{PA}p}$ , et la proportion de temps en phase d'appui  $Prop_{\text{PA}p}$ . Choisissons ici à titre de démonstration, la durée du cycle,  $\Delta T_{\text{GC}p}$ . Pour chaque enregistrement nous pouvons établir la valeur de  $\Delta T_{\text{GC}p}$  comme une valeur moyenne sur l'ensemble de la durée de l'enregistrement, soit  $\overline{\Delta T_{\text{GC}p}}$ . Bien qu'il est possible d'établir des poids de corrélation de manière discrète sur chaque

pas de temps, nous avons choisi dans ce cas-ci d'utiliser des poids de pondération uniforme sur l'ensemble de la durée de chaque enregistrement, soit

$$p_{pk}(t_1) = p_{pk}(t_2) = p_{pk}(t_i) = p_p, \quad (5.24)$$

Ce choix permet de simplifier le traitement des données, en sachant que dans le cas des essais cliniques du CIRRI22, la valeur de la vitesse de marche est constante pour l'ensemble d'un enregistrement, car le tapis roulant avançait à une vitesse fixe, et les paramètres temporels pour un individu était aussi pratiquement constant pour un enregistrement donnée. Le moyennage des paramètres temporels permettait aussi d'éliminer des valeurs de cadence aberrantes, pouvant survenir si certains événements de marche sont détectés de manière incorrecte. Cela revient donc à déterminer  $N_{TS}$  (135 pour CIRRI22) poids de pondération différents, soit un pour chaque enregistrement. Nous devons ensuite définir l'intervalle de variabilité  $I_v(\overline{\Delta T_{GCp}})$  du paramètres de pondération, défini comme,

$$\begin{aligned} \overline{\Delta T_{GCpLB}} &= 1\%_0(\{\overline{\Delta T_{GCp}} \mid \forall 1 < P < N_{TS}\}) \\ \overline{\Delta T_{GCpUB}} &= 99\%_0(\{\overline{\Delta T_{GCp}} \mid \forall 1 < P < N_{TS}\}) \\ I_v(\overline{\Delta T_{GCp}}) &= \overline{\Delta T_{GCpUB}} - \overline{\Delta T_{GCpLB}}, \end{aligned} \quad (5.25)$$

où le symbole  $\%_0$  représente un percentile, et donc  $1\%_0$  et  $99\%_0$  représente respectivement le 1<sup>er</sup> et le 99<sup>e</sup> percentile de l'ensemble des valeurs dans la liste des 135 valeurs  $\overline{\Delta T_{GCp}}$ . Si certaines valeurs aberrantes sont présentes dans les données, il est aussi possible d'utiliser des valeurs de percentile plus serrées, afin d'éviter d'avoir un intervalle inutilement grand. Dans ce cas-ci, les valeurs  $\overline{\Delta T_{GCp}}$  de la variable de pondération sont déjà moyennées et il n'y a donc pas de valeurs aberrantes, comme on peut le voir dans les figures 5.6 à 5.8.

La prochaine étape est d'établir les valeurs de pondération  $p_p$  pour un enregistrement  $p$  donné en fonction de sa valeur  $\overline{\Delta T_{GCp}}$  de la variable de pondération pour les diverses matrices  $\mathbf{W}_{outk}$  avec une fonction gaussienne  $f_{pondk}(\overline{\Delta T_{GCp}})$ ,

$$p_p = f_{pondk}(\overline{\Delta T_{GCp}}) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(POS(\overline{\Delta T_{GCp}}, I_v) - \mu_k)^2}{2\sigma^2}}, \quad (5.26)$$

où la fonction de normalisation  $POS(\overline{\Delta T_{GCp}}, I_v)$  représente la position de la valeur  $\overline{\Delta T_{GCp}}$  dans l'intervalle total,

$$POS(\overline{\Delta T_{GCp}}, I_v) = \frac{\overline{\Delta T_{GCp}} - \overline{\Delta T_{GCpLB}}}{I_v(\overline{\Delta T_{GCp}})}. \quad (5.27)$$

Dans l'équation (5.26), le paramètre  $\sigma$  représente la largeur de la gaussienne sur l'ensemble de l'intervalle  $I_v(\overline{\Delta T_{GCp}})$  et le paramètre  $\mu_k$  représente le centre de la gaussienne. L'indice  $k$  représente l'indice du choix du centre de la fonction gaussienne permettant de calculer plusieurs matrices  $\mathbf{W}_{outk}$ . Pour chaque indice  $k$ , on définit donc une gaussiennes  $f_{pondk}$  avec une valeur spécifique de centre  $\mu_k$ . Pour que ces fonctions couvrent l'ensemble de l'intervalle de la variable de pondération, on définit les différentes valeurs centrales  $\mu_k$  comme

$$\{\mu_1, \mu_2, \mu_3, \dots, \mu_k, \dots\} = \{(k/N_{sep}) \mid \forall 0 < k < N_{sep}\}, \quad (5.28)$$

où le nombre  $N_{sep}$  est le nombre de  $\mathbf{W}_{outk}$  que l'on souhaite générer. En utilisant l'une des  $N_{sep}$  fonctions gaussiennes centrées à  $\mu_k$ , on peut attribuer des poids de pondération  $p_p$  pour tous les enregistrements  $p$ , soit

$$\{p_{1k}, p_{2k}, \dots, p_{pk}, \dots\} = \{f_{pondk}(\overline{\Delta T_{GCp_p}}) \forall 1 < p < N_{TS}\}, \quad (5.29)$$

nous donnant une liste de  $N_{TS}$  poids de pondération pour chaque enregistrement. La répartition des valeurs  $\overline{\Delta T_{GCp_p}}$  enregistrées dans l'intervalle  $I_v$  n'est pas uniforme. Il y a moins de données sur les extrémités de l'intervalle qu'au centre. Ce faisant, la somme des poids  $p_p$  évalués est différente pour chacune des fonctions gaussiennes selon le centre  $\mu_k$  utilisé. Une dernière étape est donc de faire une normalisation de l'importance des poids, afin d'avoir une "quantité effective de donnée d'entraînement" équivalente à un entraînement uniforme pour toutes les matrices  $\mathbf{W}_{outk}$ . On peut donc normaliser les poids de pondération par la somme des coefficients de pondération multipliés par le nombre d'enregistrement dans la liste de données ( $N_{TS}$ ). Cela permet de conserver la matrice  $\mathbf{X}\mathbf{X}^t$  compatible avec un paramètre fixe de régularisation  $\gamma$  utilisé dans la méthode uniforme,

$$\{p_{1k}, p_{2k}, \dots, p_{pk}, \dots\} = \left\{ \frac{N_{TS}}{\sum\{p_1, p_2, \dots, p_{pk}, \dots\}} p_k \forall 1 \leq p \leq N_{TS} \right\}. \quad (5.30)$$

L'ensemble  $\mathcal{P}_k$  des matrices de pondération  $\mathbf{P}_{pk}$  est défini comme,

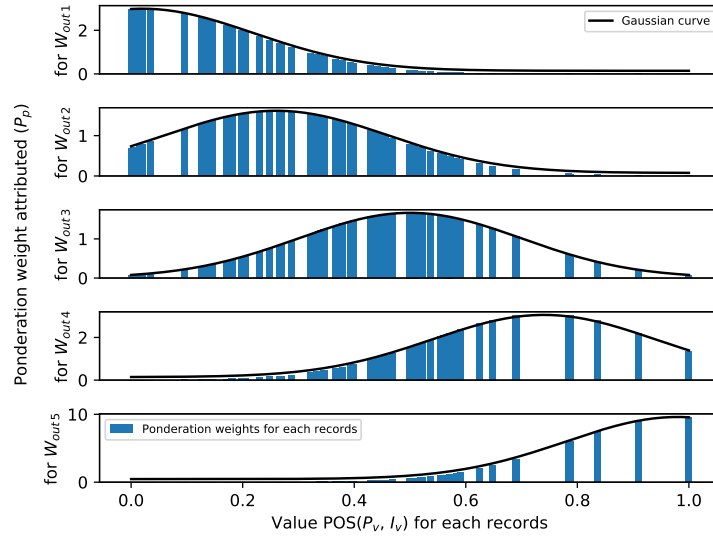
$$\mathcal{P}_k = \{\mathbf{P}_{pk} = p_{pk}\mathbf{I} \mid \forall 1 \leq p \leq N_{TS}\} \quad (5.31)$$

où  $\mathbf{I}$  représente une matrice identité de taille  $T_p \times T_p$ . On peut évaluer les matrices  $\mathbf{W}_{outk}$  avec un ensemble de poids dans l'équation (5.22), et refaire ce processus pour les  $N_{sep}$  matrices de poids de sortie.

$$\{\mathbf{W}_{out1}, \mathbf{W}_{out2}, \dots, \mathbf{W}_{outk}, \dots\} = \{\text{RR}(\mathcal{X}, \mathcal{Y}, \mathcal{P}_k) \forall 1 \leq k \leq N_{sep}\}. \quad (5.32)$$

Dans le cas actuel où les poids  $p_p$  sont uniformes sur tous les indices de temps de l'enregistrement  $p$ , la matrice est une diagonale uniforme et cela revient à multiplier les matrices  $\mathbf{X}_p$  et  $\mathbf{Y}_p^t$  par un poids scalaire  $p_{pk}$ . Afin de bien visualiser la valeur des coefficients selon les centres choisis dans l'intervalle de la variable de pondération  $I_v$ , une représentation des coefficients de pondération choisis par la méthode kernel est montrée dans la figure 5.9 avec les paramètres des fonctions gaussiennes  $N_{sep} = 5$  et  $\sigma = 0.2$  sont utilisés dans cette représentation.

En pratique, nous avons testé différentes valeurs des paramètres  $N_{sep}$  et  $\sigma$ . Nous avons établi, basé sur l'utilisation de la durée du cycle comme variable de pondération, qu'un nombre  $N_{sep}$  de 25 et un  $\sigma$  de 0.022 était un choix adéquat. La figure 5.10 montre l'allure des différentes matrices de poids de sortie  $\mathbf{W}_{outk}$  du ESN. L'augmentation de  $N_{sep}$  au-delà de 25 n'améliorait pas les résultats de prédictions. Le paramètre  $\sigma$ , d'une valeur de 0.022, était suffisamment petit pour que les entraînements soit spécifiques. Si  $\sigma$  est trop grand, cela revient à calculer un  $\mathbf{W}_{outk}$  très général, presque identique à celui obtenu selon un entraînement standard. Si  $\sigma$  est trop petit, les matrices  $\mathbf{W}_{outk}$  résultantes risquent d'être trop spécifiques (considérant une fourchette trop mince de la variable de pondération), ce qui pourrait éventuellement réduire la robustesse de la prédiction sur l'ensemble de l'intervalle du paramètre de pondération et augmenter le risque d'overfitting.



**Figure 5.9** – Exemple de coefficients de pondération calculé en fonction des valeurs du paramètre temporel (ici  $\overline{\Delta T_{GCp}}$ ). Les poids de pondération de chaque enregistrement sont représentés par la hauteur des bars, dont la position dans l'axe x représente la position de l'enregistrement dans l'intervalle de la variable de pondération  $I_v$ . Le calcul est effectué pour le cas  $N_{sep} = 5$  et  $\sigma = 0.2$ . Chaque graphique montre la pondération calculée pour les 5 différents centre  $\mu_k$ , permettant de calculer l'une des 5 matrices de poids  $\mathbf{W}_{outk}$ .

Une liste de 25 matrices  $\mathbf{W}_{outk}$  est donc générée, chacune "spécialisée" pour prédire la force au sol selon un centre  $\mu_k$  dans l'intervalle de pondération  $I_v(\overline{\Delta T_{GCp}})$ . Une représentation de la procédure globale d'entraînement par kernel est illustrée dans la figure 5.11.

#### 5.2.4.4 Mode opération d'un RC entraîné avec le Kernel training

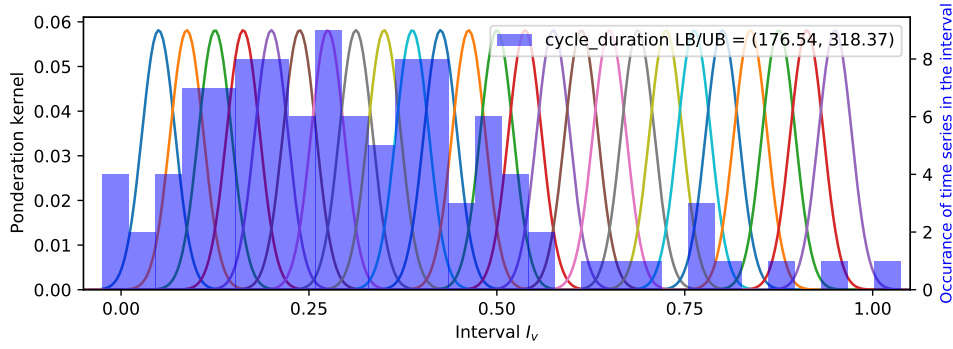
En mode opération, le RC entraîné par kernel possède plusieurs matrices  $\mathbf{W}_{outi}$  de sortie. Une fonction de choix permet au RC de sélectionner l'une de ces matrice durant la phase d'opération. Pour ce faire, une première prédiction des paramètres temporels est fait avec un RC entraîné pour le GED et avec les calculs définits au tableau 5.3. La position de  $P_v$  dans l'intervalle  $I_v(P_v)$  est ensuite définie comme,

$$POS(P_v, I_v) = \frac{P_v - C_{LB}}{I_v(P_v)}. \quad (5.33)$$

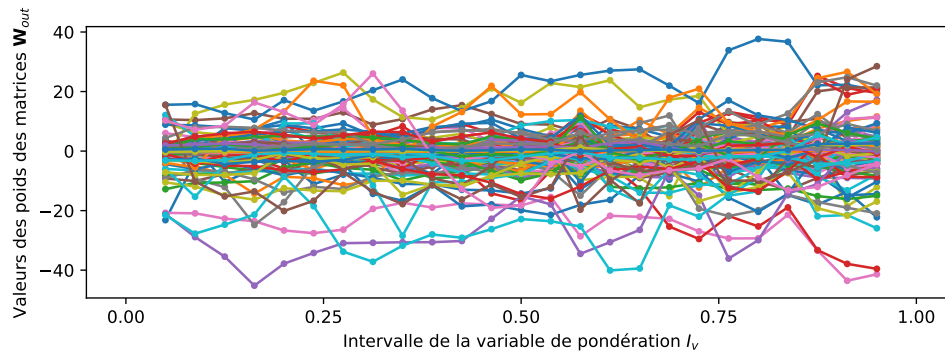
Avec cette position  $POS(P_v, I_v)$ , on peut déterminer les poids ressortant selon une interpolation à partir des matrices définies dans l'intervalle  $I_v$ . Plusieurs méthodes d'interpolation des matrices sont possible, la plus simple serait d'utiliser la matrice voisine la plus proche :

$$\mathbf{W}_{out}^{used} = \mathbf{W}_{outk} \mid k = \operatorname{argmin}_k \{ |\mu_k - POS(P_v, I_v)| \mid \forall 1 < k < N_{sep} \} \quad (5.34)$$

La seconde méthode est de faire une interpolation linéaire avec les deux matrices inférieures et supérieures adjacentes à la position  $POS(P_v, I_v)$ . La matrice utilisée dans le cas où la position  $POS(P_v, I_v)$  d'un enregistrement se trouve entre deux centre  $\mu_i$  et  $\mu_{i+1}$  sera



(a) Représentation des 25 fonctions gaussiennes utilisés pour la pondération des données d'entraînement ainsi que la distribution des données dans l'intervalle  $I_v$  (en bleu).



(b) Matrices  $\mathbf{W}_{outk}$  en fonction des des 25 kernels dans l'intervalle  $I_v$ .

**Figure 5.10** – Attribution des kernels gaussiens dans l'ensemble de l'intervalle de variabilité pour un nombre de séparation  $N_{sep} = 25$  et  $\sigma = 0.022$ .

$$\mathbf{W}_{out}^{used} = |1 - (\mu_k - POS(P_v, I_v))| \mathbf{W}_{outk} + |1 - (\mu_{k+1} - POS(P_v, I_v))| \mathbf{W}_{outk+1}. \quad (5.35)$$

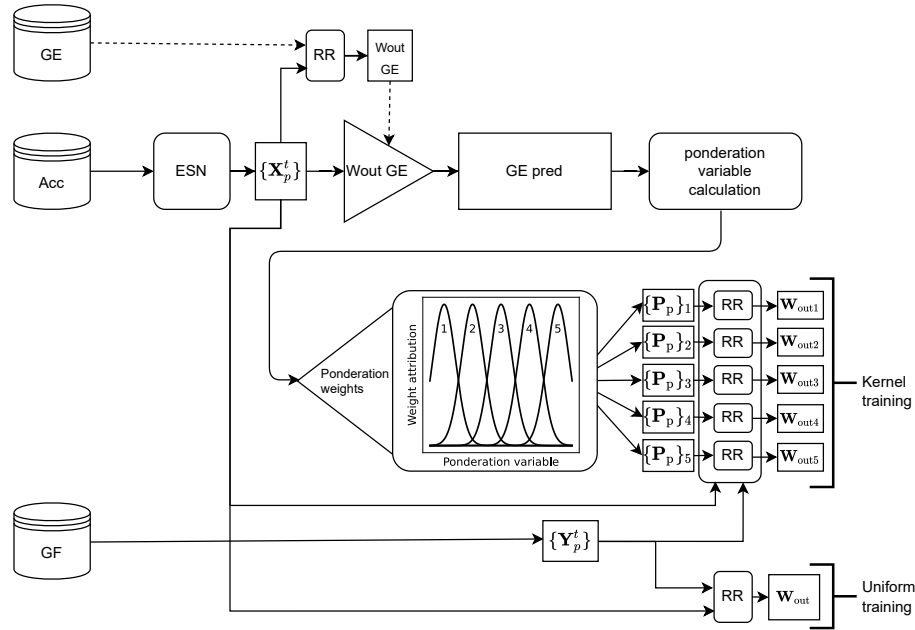
D'autres fonctions d'interpolation pourraient être utilisées, comme une interpolation cubique. Nous avons toutefois validé la méthode d'interpolation linéaire et la méthode par sélection de la matrice la plus proche et avons trouvé très peu de différence. Il est probable qu'une méthode de degré supérieur n'induisse pas d'amélioration significative ou pourrait réduire la stabilité de la prédiction. L'interpolation linéaire est donc utilisée dans nos résultats. La représentation de la mise en opération d'un RC entraîné avec le kernel training est présentée dans la figure 5.12.

## 5.2.5 Pondération multivariable

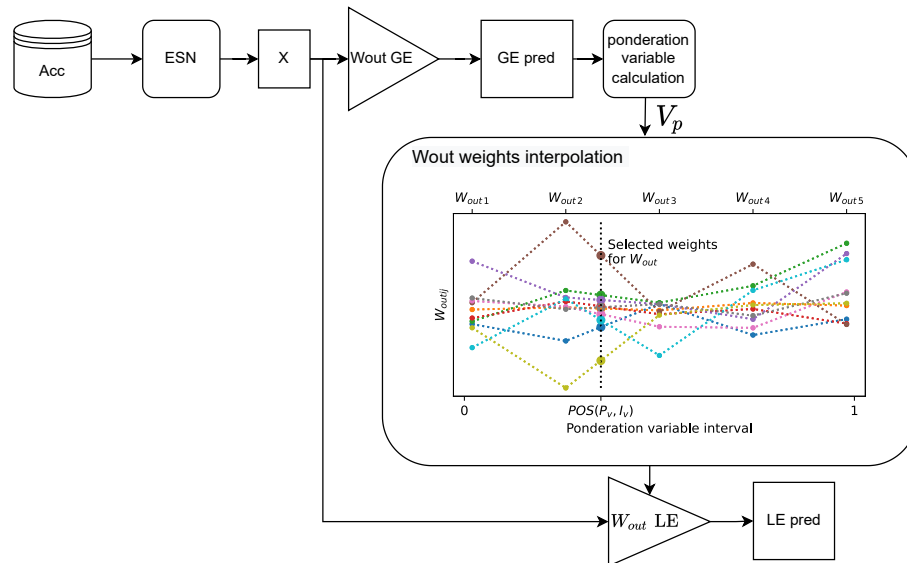
### 5.2.5.1 Extension du kernel training sur plusieurs variables de pondération

La méthode de kernel expliquée jusqu'à présent dans les sections 5.2.4.3 et 5.2.4.4 utilise une seule variable de pondération à la fois. Il est possible de généraliser le concept sur un nombre arbitraire de variables de pondération :  $\{v_p1, v_p2, v_p3, \dots, v_pi\}$ . Les matrices





**Figure 5.11** – Protocole de la méthode Kernel training. La variable  $\mathbf{P}_p$  est la variable de pondération utilisée, et elle peut représenter l'un des paramètres temporels  $\Delta T_{GCp}$ ,  $\Delta T_{PAp}$  ou  $Prop_{PAp}$ .



**Figure 5.12** – Estimation des forces au sol en mode opération avec un entraînement par kernel sur une variable de pondération.

sont alors entraînées sur des intervalles spécifiques sur chaque variable de pondération. On définit  $n_v$  comme le nombre de variables de pondération indépendantes utilisées.

Chaque enregistrement doit être pré-labelisé et réparti dans plusieurs intervalles  $I_{vpi}$ . Sur chaque enregistrement, on calcule la valeur des  $n_v$  variables de pondération  $v_{pi}$  avec le RC entraîné pour le GED et les équations du tableau 5.3. On établit ensuite pour chaque variable les valeurs extrêmes  $v_{piLB}$  et  $v_{piUB}$  et l'intervalle  $I_{vpi}$  avec l'équation (5.25). Pour simplifier la méthode, le choix des percentiles (1er et 99e) dans l'équation (5.25) est fixe

pour toutes les variables de pondération.

On établit ensuite les poids de pondération  $p_{pk}^i$  pour les différentes combinaisons de centre  $\mu_k^i$ , avec  $k$  l'indice de la position du centre de la gaussienne dans l'intervalle, l'indice  $i$  reliée aux différentes variables de pondération considérée et l'indice  $p$  relié aux différents enregistrements. Pour ce faire, les équations 5.26 à 5.32 sont calculé de manière uniformément sur chaque variable de pondération  $v_p^i$ . Les fonctions gaussiennes de l'équation (5.26) s'appliquent avec le même paramètre  $\sigma$  et le même ensemble de centres  $\{\mu_1, \mu_2, \mu_3, \dots, \mu_k, \dots\}$  pour toutes les variables  $v_p^i$ , considérant la normalisation des valeurs effectuées à l'équation (5.27).

On obtient  $n_v$  poids de pondération, notés  $p_p^i$ . On multiplie ces poids pour obtenir un poids de pondération unique pour chaque enregistrement  $p$ ,

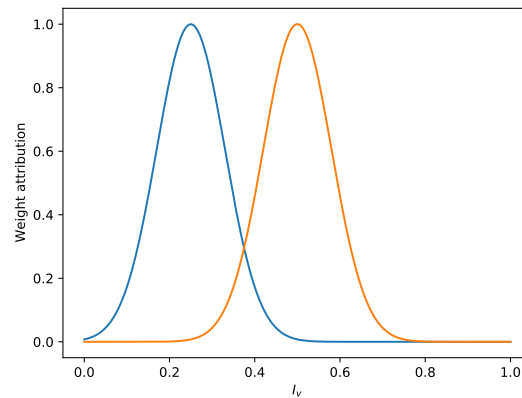
$$\{p_{1k}, p_{2k}, \dots, p_{pk}, \dots\} = \left\{ \prod_{i=1}^{n_v} p_{pk}^i \mid \forall 0 < p < N_{TS} \right\}. \quad (5.36)$$

Cela fait donc  $N_{sep}^{n_v}$  combinaisons possibles pour le calcul des poids de pondération, avec  $k$  qui représente l'indice associé à une combinaison de centre de gaussiennes  $\{\mu_k^1, \dots, \mu_k^i, \dots, \mu_k^{n_v}\}$  pour chaque variable de pondération  $v_p^i$ . Le produit des poids de pondération est ensuite normalisé comme dans la section 5.2.4.3 en utilisant l'équation (5.30). Les matrices de poids de sortie sont ensuite évaluées comme,

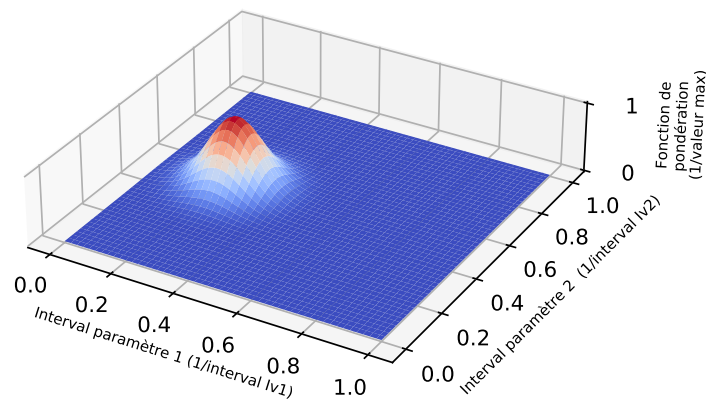
$$\{\mathbf{W}_{out1}, \mathbf{W}_{out2}, \dots, \mathbf{W}_{outk}, \dots\} = \{\mathbf{RR}(\mathcal{X}, \mathcal{Y}, \mathcal{P}_k) \mid \forall 0 < k < N_{sep}^{n_v}\}, \quad (5.37)$$

nous permettant d'obtenir  $N_{sep}^{n_v}$  matrices de poids  $\mathbf{W}_{outk}$  indépendantes et linéairement réparties dans chaque intervalle  $I_v^i$  et couvrant l'ensemble de l'espace des  $n_v$  variables de pondération. Les figures 5.13 et 5.14 représentent l'attribution des poids de pondération dans un espace de deux variables de pondération, dont les centres sont  $\{\mu_k^1 = 0.25, \mu_k^2 = 0.5\}$ .

Une limitation de la méthode est le nombre de matrices à stocker en mémoire, qui est exponentiel en fonction du nombre  $n_v$  de variables de pondération. La méthode reste donc difficilement applicable pour une grande quantité de variables. Pour réduire la quantité de matrices calculées, il est possible de réduire légèrement le nombre de séparation  $N_{sep}$ , et d'augmenter la largeur  $\sigma$  des fonctions gaussiennes, pour avoir un entraînement par kernel qui couvre encore l'ensemble de l'espace des variables de pondération, mais en faisant un compromis sur la spécificité des matrices de poids de sorties calculés. Dans notre cas, nous avons appliqué la méthode avec  $N_{sep} = 25$ ,  $\sigma = 0.022$  (donc 625 combinaisons possibles) pour tester la méthode de kernel avec deux des variables de pondération (deux parmi  $\Delta T_{GCP}$ ,  $\Delta T_{PAp}$  et  $Prop_{PAp}$ ), et nous avons utilisé  $N_{sep} = 10$  et  $\sigma = 0.05$  dans le cas où les trois variables étaient utilisées.



**Figure 5.13** – Représentation des fonctions de pondération individuelles pour chaque variable de pondération  $f_{pondk1}(v_1, I_{v1})$  et  $f_{pondk2}(v_2, I_{v2})$ , avec les centres  $\mu_{k1} = 0.25$  (bl) et  $\mu_{k2} = 0.5$  (orange). Les intervalles de pondération sont  $I_{v1} = 0.5$  et  $I_{v2} = 0.5$ .



**Figure 5.14** – Représentation de la Fonction de pondération globale  $f_{pondk1}(v_1, I_{v1}) \times f_{pondk2}(v_2, I_{v2})$  dans l'espace des variables de pondération, avec les centres  $\mu_{k1} = 0.25$  et  $\mu_{k2} = 0.5$ .

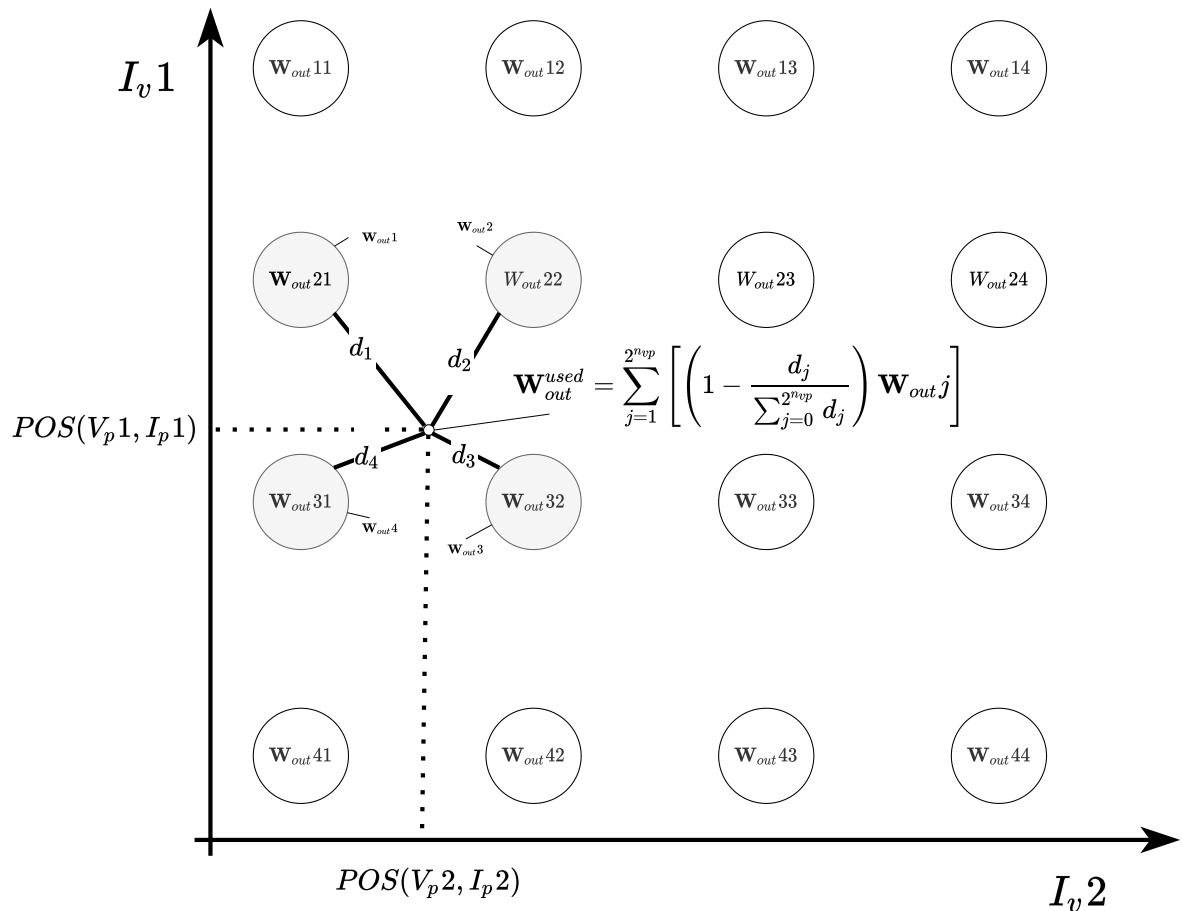
### 5.2.5.2 Interpolation sur plusieurs variables de pondération en mode opération

Une fois en mode opération, les étapes de calcul suivent encore le même protocole que dans la section 5.2.4.4. La première étape est de labeliser les enregistrements selon chacun des paramètres temporels à l'aide d'un RC appliquant une prédiction de GED et des équations du tableau 5.3. La position  $POS(v_p i, I_{v i})$  dans l'intervalle  $I_{v i}$  de chacune des variables de pondération d'indice  $i$  est calculée selon l'équation (5.33). Il est ensuite possible d'interpoler une matrice de poids de sortie dans l'espace des variables en utilisant les matrices adjacentes précalculées. Pour ce faire, on utilise une interpolation linéaire, qui se calcule selon la distance entre la position d'un enregistrement  $POS(v_p i, I_{v i})$  et la position du centre des fonctions gaussiennes des matrices adjacentes  $j$ ,

$$\mathbf{W}_{out}^{used} = \sum_{j=1}^{2^{n_v}} \left[ \left( 1 - \frac{d_j}{\sum_{j=1}^{2^{n_v}} d_j} \right) \mathbf{W}_{out,j} \right], \quad (5.38)$$

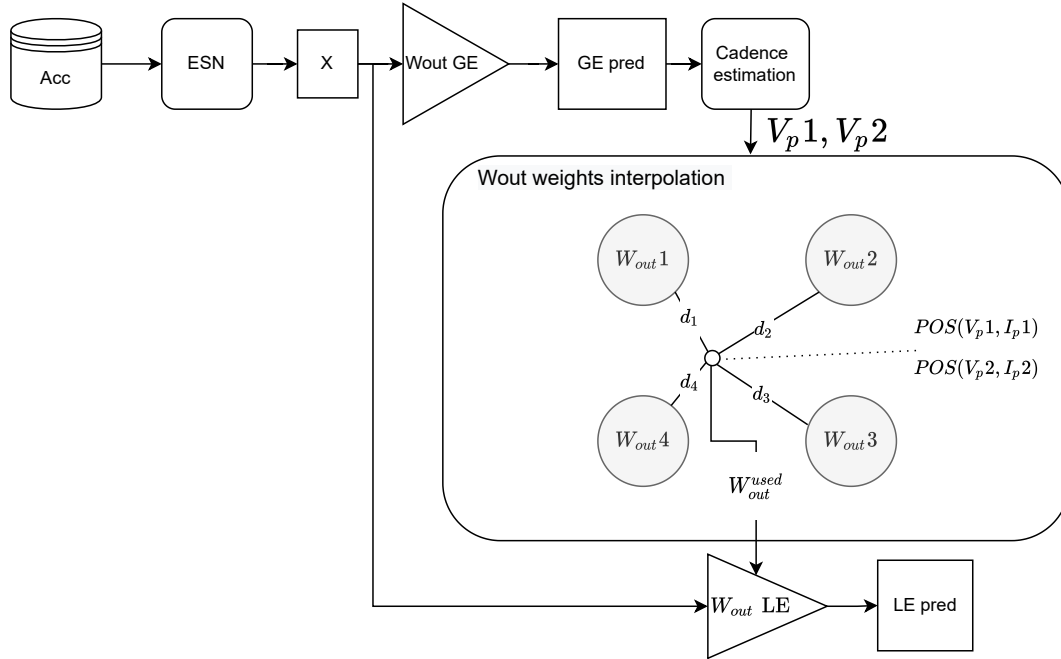
où  $d_j$  est la distance euclidienne entre les positions  $POS(v_p i, I_{v_p i})$  de l'enregistrement et les centres  $\mu_{j i}$  associés à la matrice de poids de sortie  $\mathbf{W}_{out,j}$  sont calculés comme suit :

$$d_j = \sqrt{\sum_{i=1}^{n_v} (\mu_{j i} - POS(v_p i, I_{v_p i}))^2}. \quad (5.39)$$



**Figure 5.15** – Méthode d'interpolation des noeuds.

En somme l'estimation des forces au sol en mode opération avec un entraînement par kernel multi-variable se fait de la même manière que dans le cas d'une seule variable (dans la section 5.2.4.4), mais en évaluant les  $n_v$  variables de pondération avec la tâche de GED, et en utilisant une méthode d'interpolation linéaire multidimensionnelle sur  $n_v$  variables, comme montré dans la figure 5.16.



**Figure 5.16** – Estimation des forces au sol en mode opération avec un entraînement par kernel sur plusieurs variables de pondération (la représentation est en 2D dans cette figure).

## 5.2.6 Évaluation de la performance

Pour comparer le signal de prédiction du RC avec le signal mesuré expérimentalement, on utilise une métrique d'erreur absolue MAE,

$$\text{MAE} = \frac{1}{N_{\text{TS}}} \sum_{p=1}^{N_{\text{TS}}} \left( \frac{1}{T_p} \sum_{i=1}^{T_p} |y_p(t_i) - \mathbf{y}_p^t(t_i)| \right), \quad (5.40)$$

où  $N_{\text{TS}}$  est le nombre de série de données enregistrées dans la base de données et  $T_p$  est le nombre de points dans la série de données  $p$ .

Cette métrique est utilisée en considérant l'erreur selon trois fenêtrages dans le cycle de marche. En effet, il est possible que la performance de prédiction selon les positions et les approches utilisées soit différente si l'on considère l'ensemble ou des parties spécifiques du cycle de marche.

On s'intéresse aux phases de loading response et de pré-swing comme parties spécifiques du cycle de marche. Comme discuté dans la section 1.2.2, la prédiction du signal des forces au sol lors de ces phases correspondent à des points maxima qui sont critiques dans

la progression des conditions d'arthrose [1]. Nous avons donc jugé pertinent d'évaluer la performance du modèle spécifiquement dans ces régions.

Le premier fenêtrage considère donc l'ensemble des points dans le cycle de marche. Les deux autres fenêtrages considèrent uniquement des zones du loading response et du pré-swing.

- **Cycle complet** : Les points d'erreur entre 0 et 100 % de chaque cycle de marche sont considérés.
- **Erreur sur la phase de loading response** : Les points d'erreur entre 10 et 18 % de chaque cycle de marche sont considérés. Cette région représente la zone où l'amplitude de la force verticale est la plus importante.
- **Erreur sur la phase de preswing** : Les points d'erreur entre 44 et 52 % de chaque cycle de marche sont considérés. Cette zone représente le deuxième intervalle dans le cycle où l'amplitude de la force au sol est grande.

### 5.3 Résultats

Les figures 5.17, 5.18, 5.19 montrent respectivement les résultats de  $MAE_{BW}$  pour la prédiction des forces AP, ML et V. Dans chacune des figures, la première colonne représente l'erreur sur l'ensemble du cycle de marche (0-100%), la seconde colonne représente l'erreur dans la zone loading response (intervalle de 10-18%), puis la troisième est pour la zone du pré-swing (intervalle de 44-52%). Les graphiques de chaque ligne représentent l'erreur pour une position spécifique sur le corps (MFT, TS, H, MM, MK). Chaque catégorie de l'axe des X des graphiques représente les résultats selon différentes variables d'entrées  $\mathbf{u}(t)$  du RC défini comme une combinaison d'accélération parmi les axes AP/ML/V (voir le tableau 5.2).

Pour chaque axe d'accélération, 4 mesures d'erreur sont présentées, les trois premiers points de couleur sont ceux utilisant la méthode d'entraînement par kernel. Les points en bleu utilisent la durée du cycle comme variable de pondération, ceux en rouge utilisent la proportion de la durée de la phase d'appui, et ceux en vert utilisent la durée de la phase d'appui. Les points en noir représentent le  $MAE_{BW}$  pour une méthode d'entraînement standard. Les lignes en pointillés représentent les valeurs minimales trouvées pour chacun des quatre types d'entraînement.

Les combinaisons de position/axe des IMU obtenant le plus faible  $MAE_{BW}$  sur l'ensemble des méthodes d'entraînement et d'intervalles du cycle de marche sont mis en évidence dans les tables 5.4, 5.5 et 5.6. Ces tables montrent la valeur minimal  $\min(MAE_{BW})$  parmi ces combinaisons, ainsi que la distance de cette valeur par rapport aux autres erreurs obtenues avec les autres combinaisons (Dist p/r moy). Cette distance est représentée en nombre d'écart-types ( $\sigma(MAE_{BW})$ ) de l'ensemble des erreurs  $MAE_{BW}$  obtenues pour toutes les combinaisons possibles position/axe entre le  $MAE_{BW}$  minimal,  $\min_j(MAE_{BW})$ , trouvé et la moyenne,  $\overline{MAE_{BW}}$ , de cet ensemble :

$$\text{Dist p/r moy} = \frac{\min(MAE_{BW}) - \overline{MAE_{BW}}}{\sigma(MAE_{BW})}. \quad (5.41)$$

La distance par rapport à la moyenne représente donc à quel point ce minimum est petit par rapport à l'erreur de toutes les autres combinaisons.

Une faible distance par rapport à la moyenne indique que le  $MAE_{BW}$  minimal obtenu est équivalent aux erreurs obtenues dans l'ensemble. Dans le cas contraire, si cette distance est grande, cela veut dire que l'erreur minimale avec la meilleure combinaison (position/axe) est beaucoup plus faible que le reste des erreurs obtenues avec les autres combinaisons.

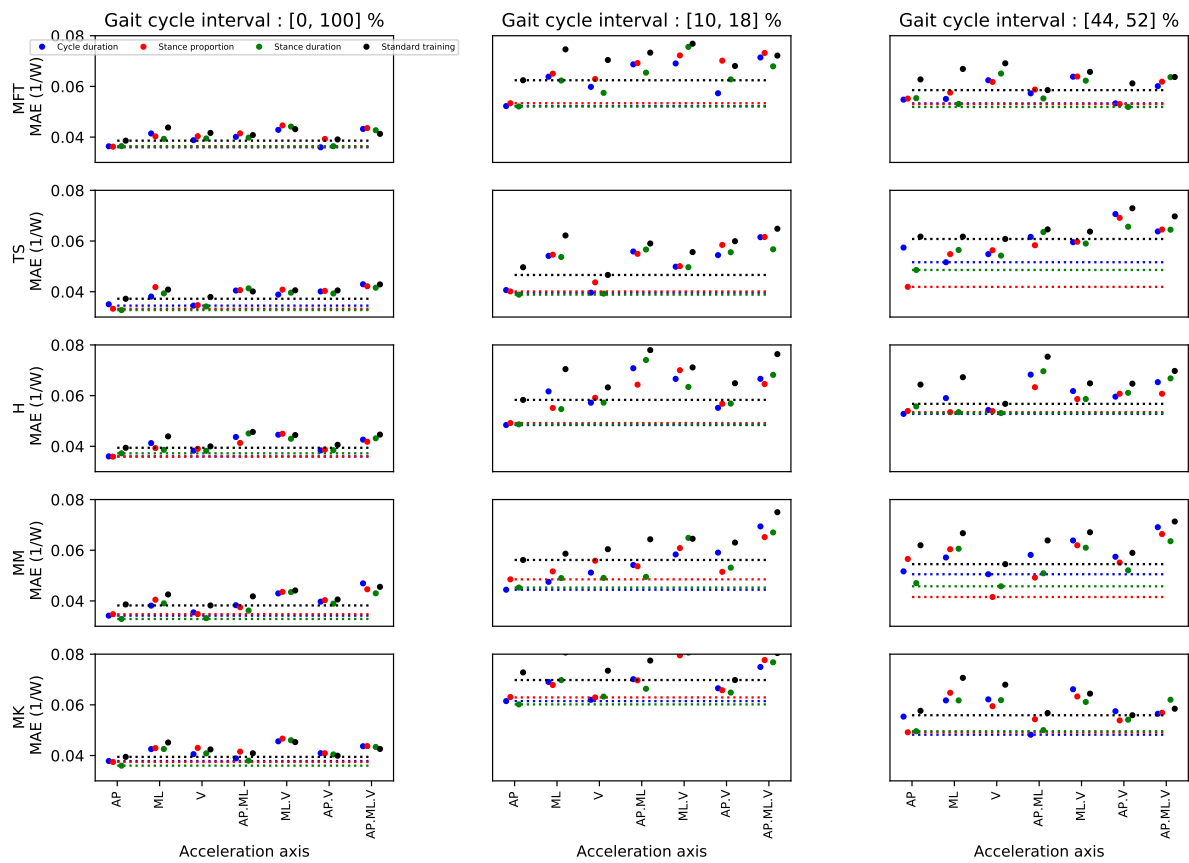


Figure 5.17 – RMSE sur l'estimation des forces AP pour les différentes positions de capteurs et combinaison d'axes d'accélération comme données d'entrées.

Méthode	Interval (%)	Meilleure valeur (MAE)	Dist p/r moy ( $1/\sigma$ )	Meilleure Position	Meilleure Axe accel
Cycle duration	[0, 100]	0.034	-1.806	MM	AP
	[10, 18]	0.040	-2.092	TS	V
	[44, 52]	0.048	-2.003	MK	AP:ML
Stance proportion	[0, 100]	0.033	-2.214	TS	AP
	[10, 18]	0.040	-2.238	TS	AP
	[44, 52]	<b>0.042</b>	<b>-2.663</b>	MM	V
Stance duration	[0, 100]	<b>0.033</b>	<b>-1.996</b>	TS	AP
	[10, 18]	<b>0.039</b>	<b>-2.055</b>	TS	AP
	[44, 52]	0.046	-1.962	MM	V
Standard training	[0, 100]	0.037	-1.855	TS	AP
	[10, 18]	0.047	-2.342	TS	V
	[44, 52]	0.054	-1.911	MM	V

Table 5.4 – Meilleurs résultats de prédiction pour l'axe AP des GRF



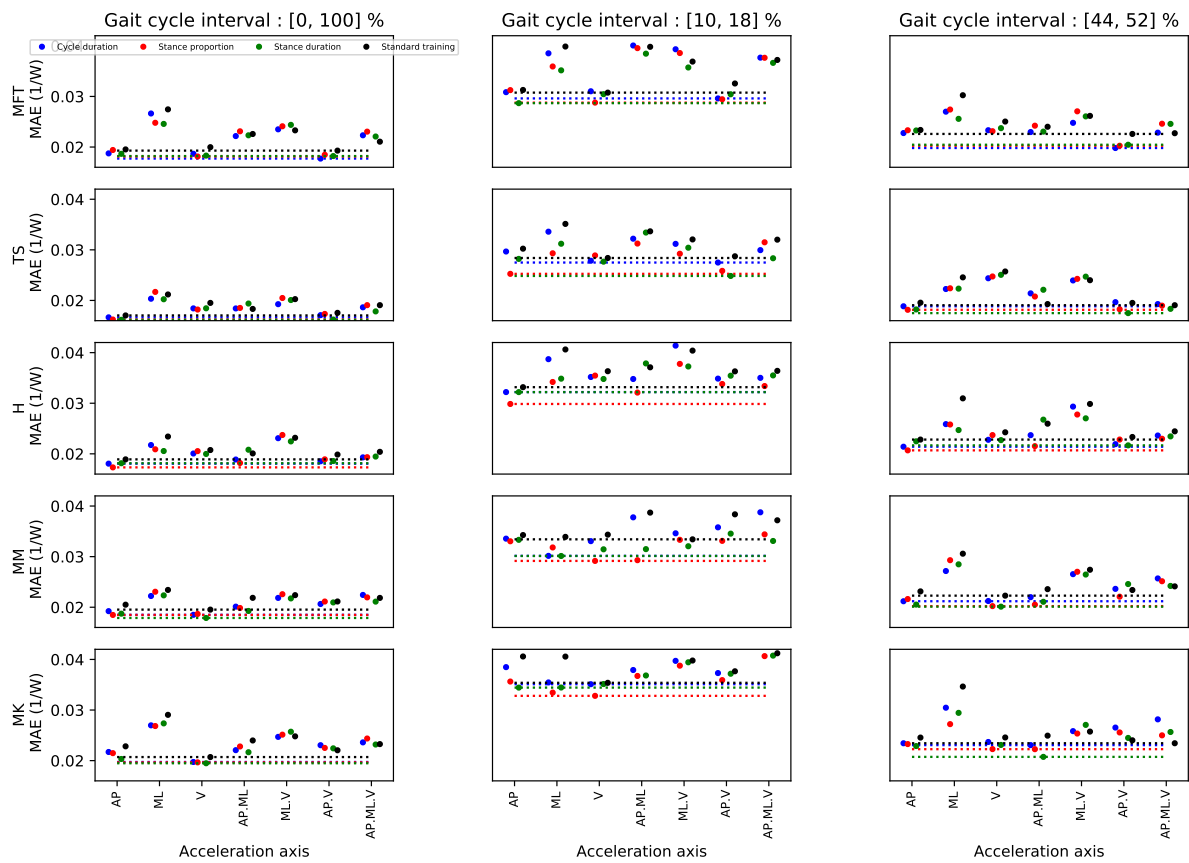


Figure 5.18 – RMSE sur l'estimation des forces ML pour les différentes positions de capteurs et combinaison d'axes d'accélération comme données d'entrées.

Méthode	Interval (%)	Meilleure valeur (MAE)	Dist p/r moy ( $1/\sigma$ )	Meilleure Position	Meilleure Axe accel
Cycle duration	[0, 100]	0.017	-1.611	TS	AP
	[10, 18]	0.028	-1.862	TS	AP.V
	[44, 52]	0.019	-1.789	TS	AP
Stance proportion	[0, 100]	0.016	-1.803	TS	AP
	[10, 18]	<b>0.025</b>	<b>-2.078</b>	TS	AP
	[44, 52]	<b>0.018</b>	<b>-1.918</b>	TS	AP
Stance duration	[0, 100]	<b>0.016</b>	<b>-1.745</b>	TS	AP
	[10, 18]	0.025	-2.413	TS	AP.V
	[44, 52]	0.018	-2.155	TS	AP.V
Standard training	[0, 100]	0.017	-1.764	TS	AP
	[10, 18]	0.028	-2.013	TS	V
	[44, 52]	0.019	-1.689	TS	AP.ML.V

Table 5.5 – Meilleurs résultats de prédiction pour l'axe ML des GRF

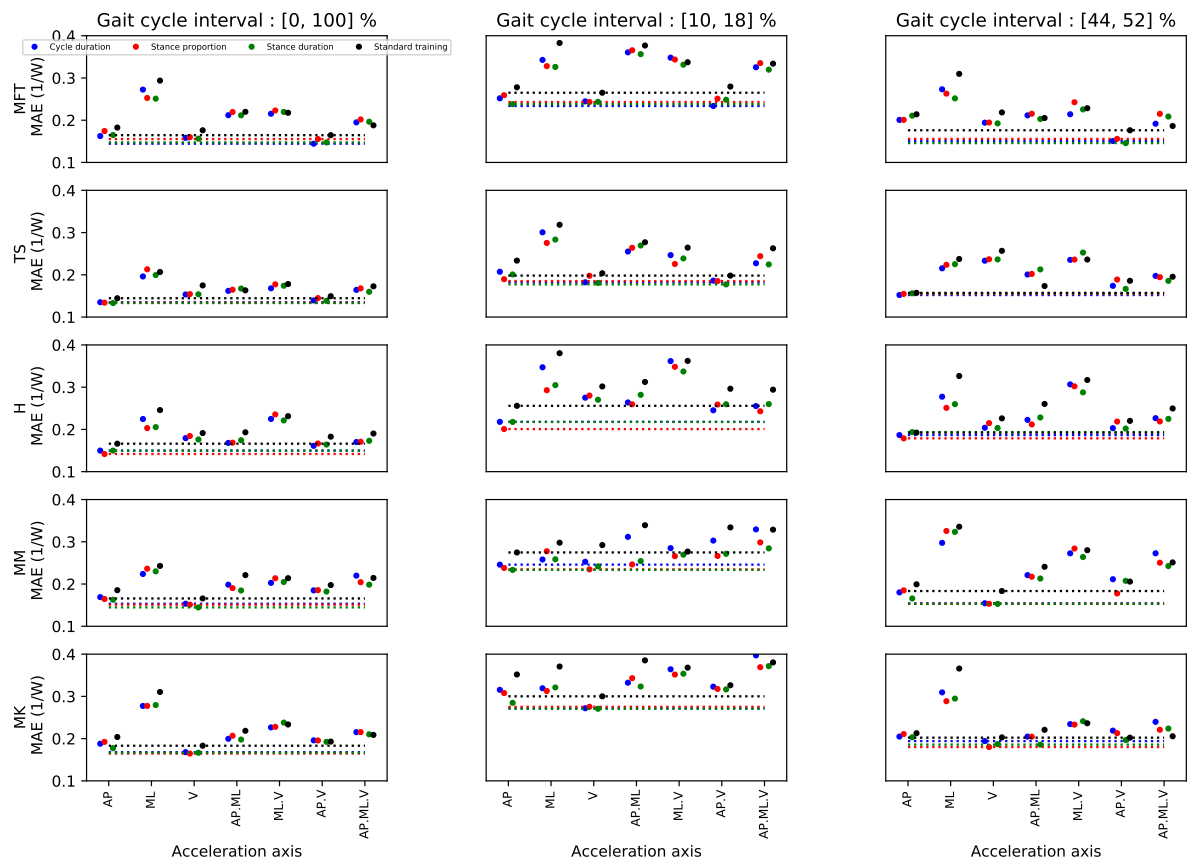


Figure 5.19 – RMSE sur l'estimation des forces V pour les différentes positions de capteurs et combinaison d'axes d'accélération comme données d'entrées.

Méthode	Interval (%)	Meilleure valeur (MAE)	Dist p/r moy ( $1/\sigma$ )	Meilleure Position	Meilleure Axe accel
Cycle duration	[0, 100]	0.135	-1.550	TS	AP
	[10, 18]	0.182	-1.921	TS	V
	[44, 52]	0.151	-1.702	MFT	AP:V
Stance proportion	[0, 100]	0.134	-1.683	TS	AP
	[10, 18]	0.185	-1.844	TS	AP:V
	[44, 52]	0.153	-1.670	MM	V
<b>Stance duration</b>	[0, 100]	<b>0.133</b>	<b>-1.611</b>	TS	AP
	[10, 18]	<b>0.177</b>	<b>-2.051</b>	TS	AP:V
	[44, 52]	<b>0.146</b>	<b>-1.793</b>	MFT	AP:V
Standard training	[0, 100]	0.145	-1.592	TS	AP
	[10, 18]	0.198	-2.256	TS	AP:V
	[44, 52]	0.157	-1.533	TS	AP

Table 5.6 – Meilleurs résultats de prédiction pour l'axe V des GRF

### 5.3.1 Comparaison globale des résultats sur les positions et les axes du corps

Le choix de la combinaison de positions et d'axes d'accélération utilisés comme données d'entrées possède une influence relativement mineure sur la performance.

Les résultats montrent que la distance (Dist p/r moy) de l'erreur minimale par rapport aux autres positions et combinaisons d'axes est d'au plus 2.6 écarts-types en dessous de la moyenne de la distribution de tous les  $MAE_{BW}$ . Cela veut dire que le meilleur résultat reste relativement proche des autres valeurs de  $MAE_{BW}$  obtenus.

En d'autres termes, certaines positions et axes sont plus adaptés pour optimiser la performance de prédiction du RC, mais les autres combinaisons devraient permettre aussi obtenir des résultats du même ordre de grandeur.

### 5.3.2 Comparaison entre les variables de pondération

On voit que le  $MAE_{BW}$  minimum de la méthode standard (ligne pointillée noire) est toujours plus grand que ceux des méthodes d'entraînement par kernel à une variable (lignes pointillées rouge, vert et bleu) dans les trois prédictions des forces au sol (AP, ML, V) et pour toutes les positions sur la jambe.

De plus, les erreurs obtenues avec la méthode de kernel training sont assez proches pour chaque variable de pondération. Cela montre que la segmentation des données par rapport à ces trois paramètres est similaire. Cela est en effet ce que l'on observe en comparant les différents paramètres temporels des différents enregistrements dans la figure 5.21, où l'on voit que ces trois variables sont fortement corrélées les unes envers les autres.

On peut néanmoins comparer la performance des différentes variables de pondération de la méthode Kernel selon les tables 5.4, 5.5 et 5.6. La performance d'une variable devrait permettre à la fois de minimiser l'erreur et de minimiser la distance (Dist p/r moy) de l'erreur minimale par rapport à la moyenne des erreurs dans la distribution des combinaisons position/axe. Une faible distance (Dist p/r moy) indique qu'une variable permet d'obtenir une erreur minimale qui est représentative de l'ensemble des combinaisons possibles. Cette variable de pondération permet donc au RC d'être robuste. Cette distance de l'erreur minimale varie entre 1.8 et 2.6 écarts-types de la moyenne.

Dans le cas de la prédiction AP, la variable de SD permet la meilleure prédiction pour l'intervalle complet (0-100%) et pre-swing (10-18%), avec la combinaison TS/AP. La variable SP permet la meilleure prédiction de l'intervalle du loading response (44-52%), avec la combinaison MM/V (résultats en gras).

Dans la prédiction des forces de l'axe ML, les erreurs minimales sont toutes obtenues avec la combinaison TS/AP. L'erreur minimale pour l'intervalle total est identique pour la variable SP versus SD, mais avec une distance légèrement plus faible pour la variable SD. L'erreur la plus faible est obtenue avec la variable SP pour les intervalles du loading response et du preswing.

Pour la prédiction des forces V, les erreurs minimales sont obtenues avec la variable de SD, avec les combinaisons TS/AP pour l'intervalle (0-100%) et (10-18%) et avec la combinaison MFT/AP.V pour l'intervalle (44-52%).

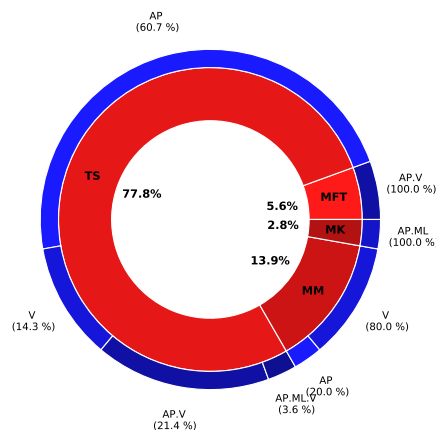
Les variables de pondération du stance duration (SD) et du stance proportion (SP) permettent donc en général d'obtenir des erreurs moyennes dans les trois intervalles (total, loading response et pre-swing) légèrement plus faibles que la variable de la durée du cycle. En somme, ces deux variables SP et SD peuvent être tous deux considérés comme des variables intéressantes à exploiter en pratique et apportant le même bénéfice.

### 5.3.3 Choix d'une position et d'un axe d'accélération optimale pour le capteur

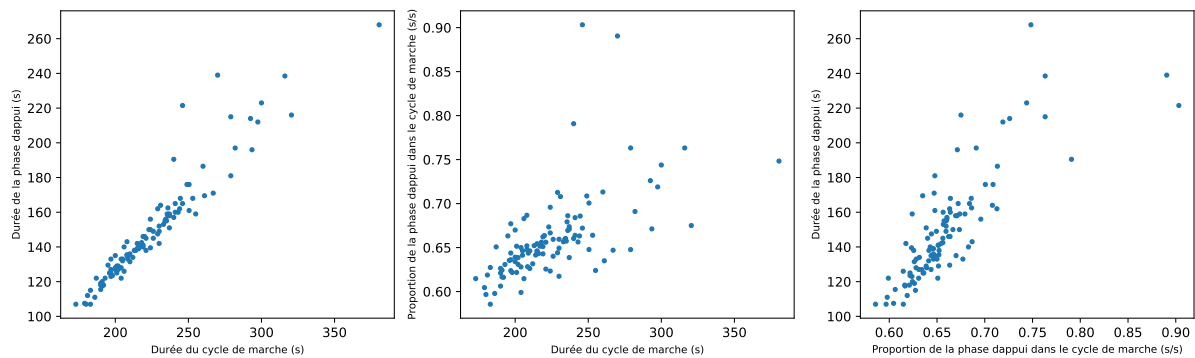
Différentes combinaisons d'axes et de positions du IMU ont permis d'obtenir des erreurs minimales : (TS/AP, MM/V, MFT/AP.V). La figure 5.20 montre un résumé de la récurrence des positions (en rouge) et des choix d'axes d'accélération (en bleu). La combinaison TS dans l'axe AP obtient le plus fréquemment les erreurs minimales pour tous types d'intervalles (cycle complet, phase de loading response et pre-swing) et de signaux de prédiction confondus.

Dans les cas où ce n'est pas la meilleure combinaison, la valeur TS/AP reste tout de même avec une petite erreur, très proche du meilleur résultat. Par exemple, on peut voir, dans la tableau 5.4, pour la prédiction des forces AP de l'intervalle 44-52% (3e colonne), que la meilleure prédiction obtenue est celle du MM/V. Toutefois, dans la figure figure 5.17 la prédiction du TS/AP est la seconde meilleure prédiction pour cet interval, avec une erreur presque identique à celle du MM/V. À l'opposé, la prédiction dans le premier intervalle (10-18%, 2e colonne), la meilleure prédiction est obtenu avec la position du TS, et les prédictions obtenues avec le MM sont significativement moins bonnes que celles du TS. De même, dans la prédiction des forces V (figure 5.19), les meilleurs résultats sont obtenus avec les combinaisons TS/AP TS/AP.V et MM/V, mais on voit que les résultats TS/AP sont toujours proches du meilleur résultat. Dans la prédiction des forces ML (figure 5.18), la combinaison TS/AP est toujours la meilleure.

En somme, le choix de la position TS avec l'axe AP permet d'obtenir la meilleure prédiction de force AP, ML et V parmi toutes les combinaisons testées avec la méthode RC sans ou avec kernel training. Il serait donc recommandé d'utiliser cette position et cette axe afin d'obtenir la meilleur des prédictions.



**Figure 5.20** – Proportion selon les positions (rouge) entre le top of shoes (TS), le medial malleolus (MM), medial front of tibia (MFT) et le medial knee (MK) et selon les choix de combinaison d'axes (en bleu) entre antéro-postérieur (AP), médio-latéral (ML) et vertical (V), ainsi donnant la prédiction optimale des forces au sol selon les trois orientations et les trois intervalles du cycle de marche. La position Heel (H) n'apparaît pas, car elle n'obtient jamais la meilleur prédiction.



**Figure 5.21** – Comparaison entre les variables de pondération de la durée du cycle de marche, de la proportion de la phase d'appui et de la durée de la phase d'appui.

### 5.3.4 Résultats avec plusieurs variables de pondération

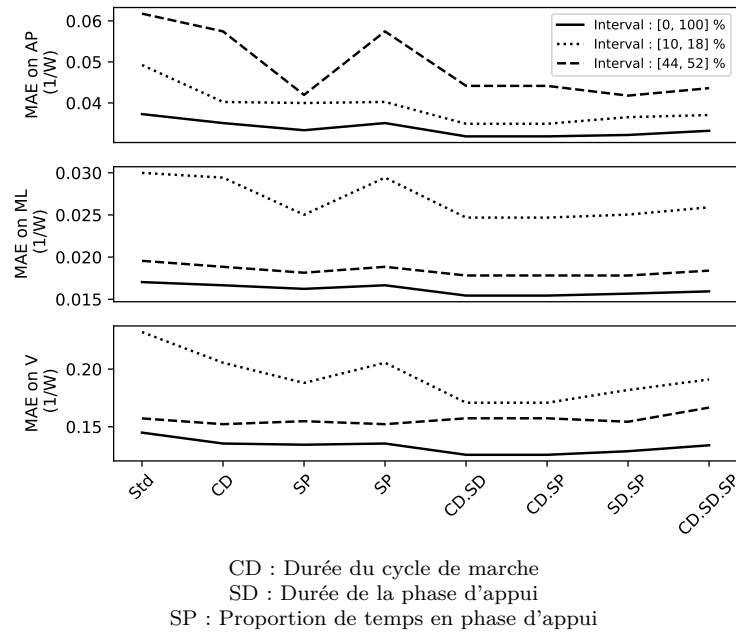
À la lumière des résultats précédemment expliqués, nous avons choisi d'évaluer plus en détail la performance de prédiction avec la position TS/AP en testant des entraînements par kernel avec 1,2 ou 3 variables de pondération. La figure 5.22 permet de visualiser la performance du RC en termes d'erreur moyenne en fonction des combinaisons de 0, 1,2 ou 3 variables de pondération différentes. Pour les entraînements avec 2 variables de pondération, le nombre de séparations était de 25, et la largeur sigma était de 0.022. Cela nous donne 625  $\mathbf{W}_{out_i}$  différentes. Pour l'entraînement à 3 variables de pondération, le nombre de séparations était de 10, avec un sigma de 0.05. Ce choix a été fait afin de limiter le nombre de matrices enregistrées dans la phase d'entraînement. Ces paramètres nous donnent 1000  $\mathbf{W}_{out_i}$  différentes.

La figure 5.22 nous montre qu'une amélioration significative des résultats est possible en utilisant la variable de pondération SP de manière unique par rapport à un entraînement standard. Le fait d'utiliser une combinaison de deux variables de pondération permet de diminuer légèrement l'erreur par rapport à l'utilisation de seulement une variable de pondération. L'amélioration des résultats entre 1 variable et 2 variables reste toutefois très faible. Autrement dit, la segmentation des données avec plusieurs variables augmente très peu la précision de la prédiction. Cela est explicable en raison de la forte corrélation entre les variables de pondération, comme montré dans la figure 5.21.

On voit que la segmentation avec 3 variables de pondération obtient aussi des résultats similaires, voire légèrement moins bons que les résultats à deux variables. La légère augmentation de l'erreur (qui s'observe notamment dans l'intervalle [44% - 52%] des forces ML et V) avec 3 variables de pondération peut s'expliquer par le nombre restreint de séparations pour chaque variable. En effet, le choix de 10 séparations avec des gaussiennes plus larges a été fait afin d'avoir un nombre raisonnable de matrices de poids de sortie à 1000. Ce nombre de séparations plus faible semble limiter les performances, indiquant qu'on limite le niveau de spécificité de chaque matrice. De plus, comme on peut voir dans la figure 5.10b, la variation des poids des différentes matrices de sortie  $\mathbf{W}_{out_i}$  semble être hautement non-linéaire à travers l'intervalle d'une variable de pondération. Il est donc possible que l'interpolation en mode opération fait dans une grille plus grossière de matrices pré-entraînées soit moins fiable.

Comme les variables sont fortement inter-reliées, ces résultats montrent qu'il est plus

efficace dans ce cas-ci d'utiliser une ou deux variables avec plus de séparation que trois variables avec moins de séparation.



**Figure 5.22** – Résultat du  $MAE_{BW}$  pour les trois prédictions (AP, ML et V) en fonction des variables de pondération utilisées.

#### 5.3.4.1 Visualisation de l'erreur dans le cycle de marche

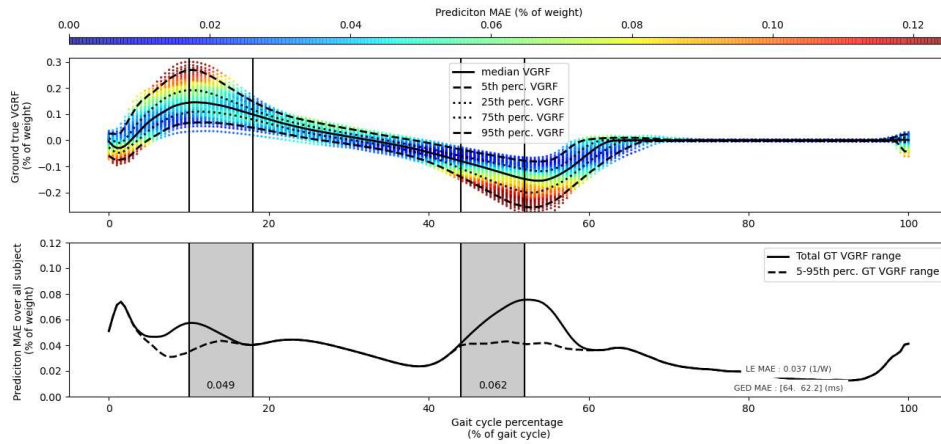
Les figures 5.23, 5.24 et 5.25 permettent de représenter l'erreur de prédiction des forces verticales de l'ensemble des cycles de marche enregistrés. Trois sous-figures sont représentées : un graphique avec un entraînement standard, un avec un entraînement kernel 1 variable de pondération SP, et un dernier avec un entraînement kernel sur 2 variables de pondération (SP et SD).

Pour chaque sous-figure, le graphique supérieur représente un nuage de points de l'erreur sur l'ensemble de la variabilité d'amplitude des signaux de force au sol mesurés (target). La couleur de ce nuage de points représente l'erreur absolue moyenne de tous les cycles de marche, interpolés en chaque point des signaux de force au sol mesurés. Les lignes pleines, en tiret et pointillés représentent respectivement la médiane, les 1er et 99e percentiles et les 25e à 75e percentiles des signaux mesurés.

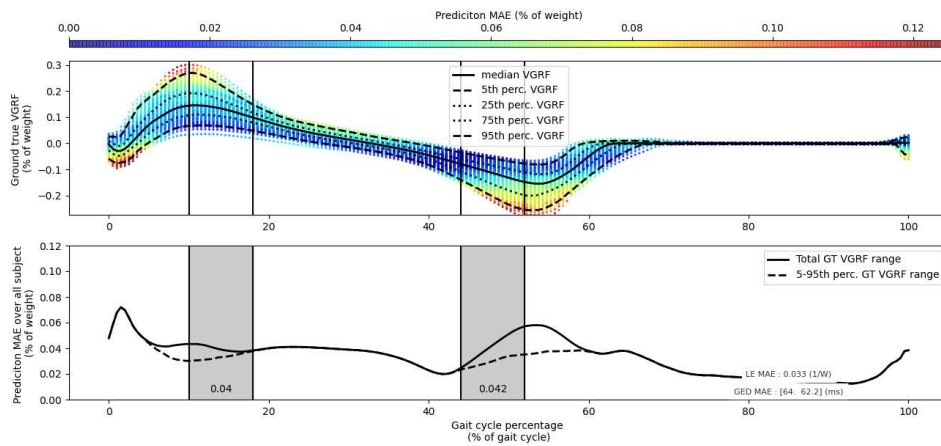
Le graphique du dessous de chaque sous-figure représente le  $MAE_{BW}$  moyen sur l'ensemble des signaux mesurés pour une position du cycle donnée entre 0 et 100 %. La ligne en tiret fait abstraction des erreurs obtenues en dessous et au-dessus du 1er et 99e percentile, alors que la ligne pleine utilise toutes les valeurs d'erreur.

On voit dans les graphiques supérieurs que l'apprentissage par kernel permet surtout de minimiser les erreurs dans les zones extrêmes de l'ensemble des courbes de force au sol. L'entraînement par kernel permet donc au modèle RC d'estimer avec plus de fidélité les signaux de force au sol dans les zones de grandes amplitudes. Ce résultat met en évidence comme l'utilisation des variables de pondération utilisées, fortement corrélées avec l'amplitude des forces de réaction verticales au sol, comme montré dans les figures 5.6

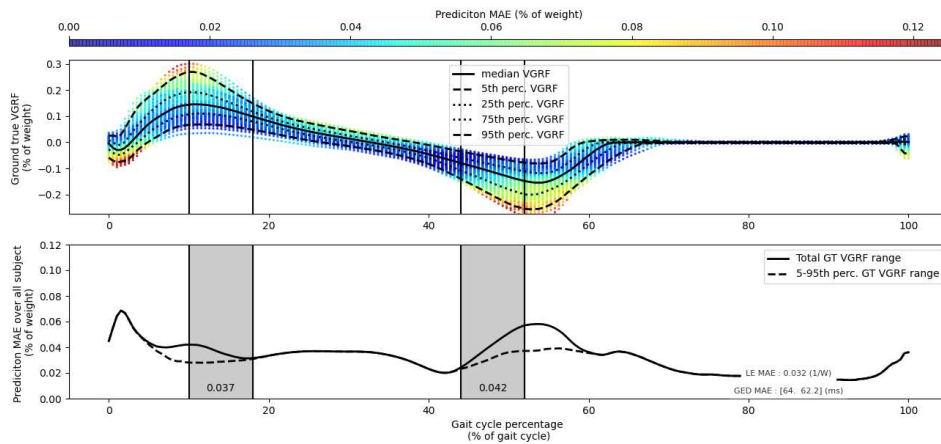
à 5.8, permettent de mieux estimer l'amplitude des signaux des forces au sol, même lorsque ceux-ci sont beaucoup plus élevées en raison d'une vitesse de marche anormale.



(a) Méthode standard



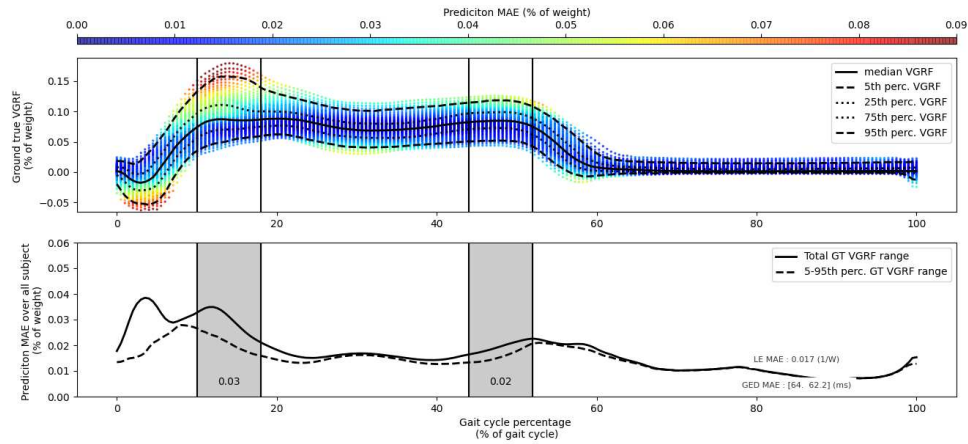
(b) Méthode kernel avec la variable SP



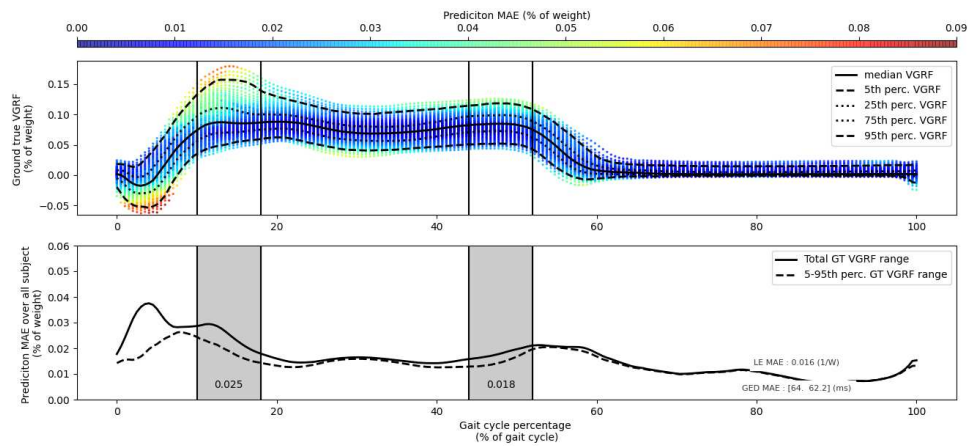
(c) Méthode kernel avec les variables SD et SP

**Figure 5.23** – Représentation de l'erreur de prédiction sur l'ensemble des signaux de force au sol antéro-postérieur avec la combinaison TS/AP comme signal d'entrée.

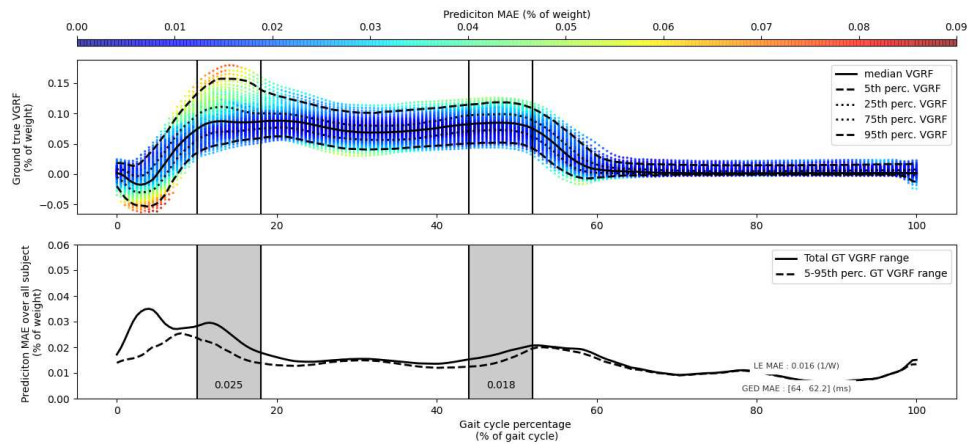




(a) Méthode standard

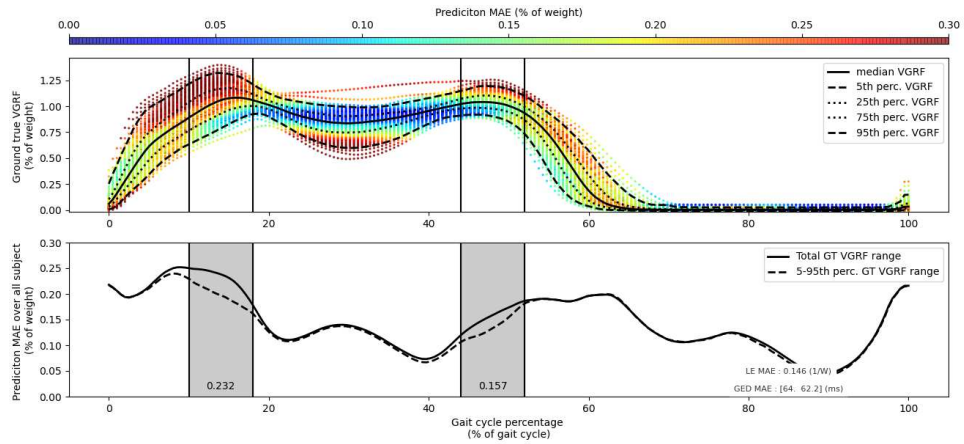


(b) Méthode kernel avec la variable SP

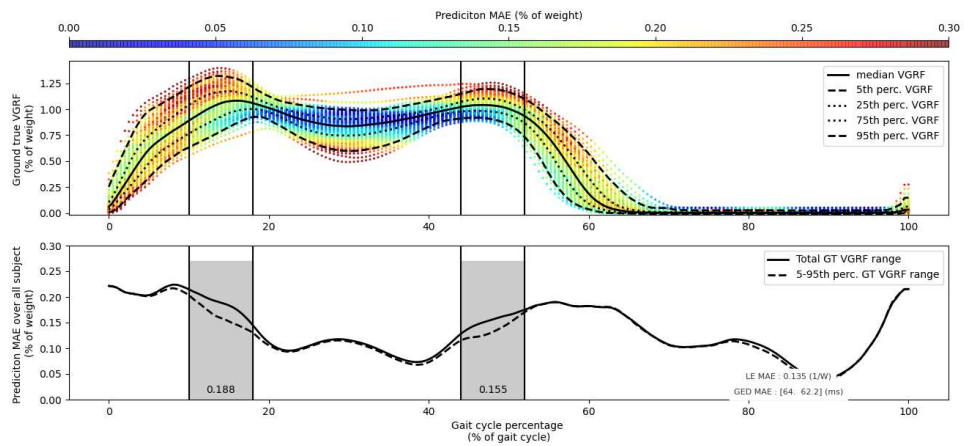


(c) Méthode kernel avec les variables SD et SP

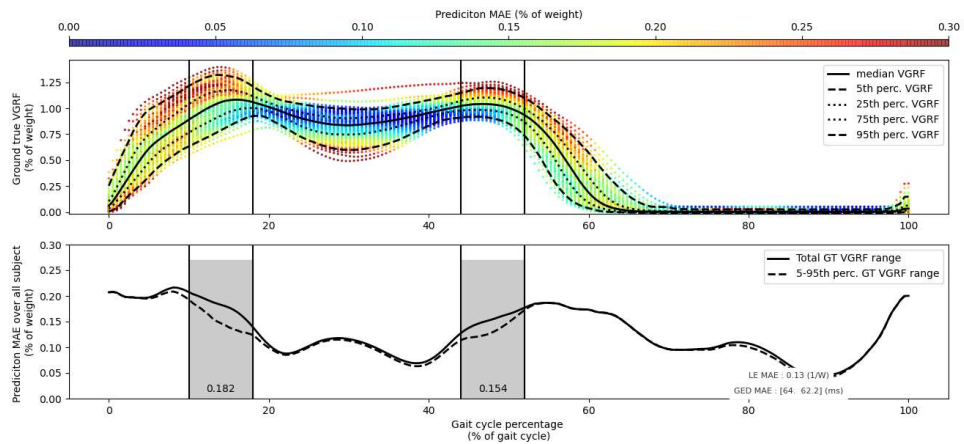
**Figure 5.24** – Représentation de l'erreur de prédiction sur l'ensemble des signaux de force au sol médio-latérale avec la combinaison TS/AP comme signal d'entrée.



(a) Méthode standard



(b) Méthode kernel avec la variable SP



(c) Méthode kernel avec les variables SD et SP

**Figure 5.25** – Représentation de l'erreur de prédiction sur l'ensemble des signaux de force au sol vertical avec la combinaison TS/AP comme signal d'entrée

## 5.4 Discussion

---

### 5.4.1 Comparaison avec la littérature

---

En somme, les résultats nous montrent que la méthode RC permet la prédiction des forces au sol selon trois orientations. La précision de la prédiction est de l'ordre de 0.12-0.13 du BW pour les forces verticales, 0.03 du BW pour les forces AP et 0.015 % du BW pour les forces ML. Cette prédiction est donc une méthode intermédiaire par rapport aux différentes méthodes trouvées dans la littérature, dont l'erreur de prédiction pouvait varier entre 0.05 et 0.35 du BW.

La variabilité des signaux des forces au sol de la base de données CIRRI22 incluait beaucoup de participants (27 normales et 18 OA) et de cycles de marche (environ 350 cycles). Cette variabilité dans les données était donc plus grande par rapport aux autres études dans la littérature. En effet, les travaux de la littérature utilisent des bases de données plus restreintes, selon le tableau 5.1, incluant moins de participants et sans intégrer à la fois des patients sains et des patients ayant une problématique physique comme l'arthrose. Parmi les méthodes AI, l'étude avec NN [130] obtenait de très bonnes prédictions (0.05 BW pour les forces verticales) et comptait un nombre raisonnable de personnes (17 patients adultes sains). Un test de prédiction a été fait avec le RC en éliminant les patients OA dans la base de données d'entraînement et de tests, et cela permettait de diminuer l'erreur à environ 0.115 BW, ce qui reste beaucoup plus grand que la méthode proposée par [130]. D'autre part, la méthode proposée par [131] était testée sur uniquement 7 adultes masculins sains. L'erreur du modèle dans ce cas-ci est légèrement moins bonne que celle du RC, soit de 0.15 BW pour [131] vs 0.13 BW pour le RC avec la méthode kernel pour la prédiction des forces V.

### 5.4.2 Performance globale et pistes d'amélioration

---

L'utilisation du kernel training avec des paramètres temporels permet d'améliorer la prédiction principalement pour des cas où la marche pourrait changer significativement de cadence et faire varier l'amplitude des signaux de GRF, alors que la prédiction standard ne permet pas d'anticiper ou de suivre ces variations. L'utilisation de cette méthode ressemble à certains modèles dans la littérature [133] permettant de faire le calibrage de la prédiction des forces en fonction de la vitesse utilisée.

D'un point de vue clinique, la performance de prédiction des forces au sol avec la méthode RC s'avère encore insuffisante pour des applications concrètes. En effet, les variations des forces entre des patients sains et ceux à risque d'avoir de l'arthrose au genou [133] sont généralement de l'ordre de 100-150 N. Cela représente environ 13 à 20 % du BW de l'amplitude totale des forces au sol verticales dans le cycle de marche en considérant le poids moyen des participants de la base de données CIRRI22, soit de 75 kg. Le RC peut avoir une erreur jusqu'à 25% dans les zones de grandes amplitudes des forces verticales au sol. L'erreur du modèle est donc trop grande pour être en mesure de capturer les variations subtiles que pourraient apporter une correction posturale. Cette erreur peut être causée aussi par la nature des données. Comme identifié dans le chapitre 3, une variation dans le positionnement du capteur entre les patients est notamment un facteur qui peut nettement influencer la performance de prédiction. La bonne calibration des accéléromètres

pourrait être investigué d'avantage afin d'établir de meilleur prédiction.

Le modèle de prédiction avec un FNN proposé par [130] semble obtenir des résultats beaucoup plus précis que ceux utilisés par le RC. Différentes pistes d'amélioration pourraient être implémentées en s'inspirant de la méthode proposée par [130]. Les entrées du FNN incluait des valeurs de dérivées et d'intégrales des accélérations depuis le début du cycle de marche. La valeur de l'intégrale dans leur méthode est réinitialisée à chaque cycle basée sur la détection d'un événement HS.

Une amélioration possible serait notamment d'ajouter un compteur d'une valeur intégrée de l'accélération d'entrée comme une autre entrée du RC et pourrait améliorer la performance de prédiction. Toutefois, il est impossible de faire ce type de modification dans la version RC-MEMS du reservoir computing.

Une seconde modification serait de définir une matrice de poids de sortie multipliée à des états de sortie du RC de plusieurs pas de temps, pour une seule prédiction  $\mathbf{y}(t)$  du RC.  $\mathbf{W}_{\text{out}}$  serait donc de taille  $\tilde{N} \times \tilde{N}$ , où  $\tilde{N}$  correspondrait au nombre de nœuds du reservoir ( $N$ ) multiplié par le nombre de pas de temps considérés pour faire une prédiction. Cette amélioration du RC, contrairement à la première, est compatible avec le MEMS-RC, car les modifications numériques s'appliquent uniquement sur les états de sorties du RC. D'un point de vue numérique, cela est l'équivalent à appliquer un filtre aux données de sortie ajustées à partir de la régression RR.

Une troisième piste d'amélioration pourrait être de rapprocher la position de l'IMU sur le corps, proche du centre de masse (CM). En effet, plusieurs algorithmes dans la littérature sont basés sur une mesure d'accélération au CM. Dans certain cas, il est possible d'obtenir une relation presque linéaire [124] entre l'accélération du CM et les réactions des forces au sol. Il est donc probable que les prédictions du RC soient s'améliorent avec des données d'entrées prises au CM. Comme présenté dans le chapitre 3, il est toutefois difficile de faire la tâche de GED avec des signaux du haut du corps. Mettre le capteur au CM pourrait donc induire d'autres difficultés pour l'application de l'entraînement par kernel pour appliquer la tâche de GED. On pourrait considérer d'exploiter deux capteurs IMU, l'un au pied, et l'autre au CM de positionner un seul IMU dans une zone intermédiaire entre le CM et le pied, sur le haut de la cuisse.

Dans l'optique minimaliste et simpliste mise de l'avant par le concept des MEMS-RC, l'utilisation de deux capteurs reste toutefois moins intéressante, impliquant deux MEMS-RC indépendants et une communication synchronisée entre ceux-ci.

Finalement, il serait aussi plus pertinent d'un point de vue clinique d'évaluer le niveau d'acceptabilité sur un patient en s'intéressant au *minimum detectable change* [134], qui permet d'évaluer quel sont les changement minimal et significatif qu'un patient peut faire varier la force de réaction au sol lors d'un processus de rééducation à la marche.

## 5.5 Conclusion

---

En conclusion, il est possible d'utiliser des RC afin de régénérer des signaux biomécaniques indirects mesurables normalement seulement dans un laboratoire avec un capteur IMU. La meilleure position et orientation du capteur pour la prédiction des forces au sol serait celle sur le dessus du pied et orienté vers l'avant, dans l'axe AP, ce qui nous donne erreur de l'ordre de 13% des forces au sol verticales.

Bien que ce résultat soit légèrement insuffisant pour une application clinique, il est possible de voir que cette méthode est d'une précision intermédiaire par rapport aux autres méthodes trouvées dans la littérature. En utilisant la méthode du kernel training, on peut exploiter de l'information reliée à la marche, comme des paramètres temporels, afin d'améliorer la précision de la prédiction. D'autres paramètres de la marche pourraient être utilisés dans la méthode d'entraînement par kernel pour tenter d'améliorer la prédiction.

# Chapitre 6

## Conclusion

### 6.1 Rappel des objectifs

---

L'objectif de ce mémoire était de définir de quelle manière le reservoir computing pouvait contribuer à l'analyse de la marche basée sur des signaux d'accélération.

- Nous voulions définir quelles étaient les tâches de prédiction biomécanique qui permettraient de supporter un traitement clinique en lien avec la marche.
- Nous voulions aussi définir dans chacune de ces tâches quelles sont les données d'entrée et les données objectives d'entraînement utilisées ainsi que les équipements nécessaires pour acquérir ces données.
- L'objectif final du projet était d'utiliser le capteur RC-MEMS pour les tâches d'analyse de la marche. Nous avons donc identifié certaines contraintes sur l'utilisation de l'algorithme de RC numérique, le ESN, pour nous assurer que les tâches définies soient compatibles avec l'utilisation éventuelle d'un reservoir computer physique comme le MEMS-RC.
- Basés sur un entraînement RR, nous voulions valider la performance de succès des diverses tâches et identifier des critères de succès permettant de se comparer à d'autres méthodes utilisées dans la littérature.
- Finalement, nous voulions comprendre quelles étaient les limitations actuelles du RC pour ces tâches biomécaniques.

### 6.2 Sommaire

---

#### 6.2.1 Définition de tâches d'analyse biomécaniques

---

Une étude bibliographique dans le chapitre 1 a permis d'identifier trois tâches en biomécanique permettant de supporter un traitement clinique de la marche. Ces trois tâches sont la détection des événements de marche, la classification de patrons de marche et la prédiction des forces au sol. Ces types de prédiction sont faciles à effectuer dans un contexte de laboratoire, mais elles deviennent plus complexes à réaliser dans un environnement écologique incluant beaucoup de variabilité et en utilisant uniquement un capteur IMU. Les algorithmes de type RC sont particulièrement intéressants dans ce contexte, car ils sont versatiles, rapides à entraîner et adaptés aux données de nature temporelle comme ceux des IMU. Ils possèdent l'avantage majeur d'être intégrables facilement dans des systèmes portatifs en raison de la faible complexité de l'algorithme et permettent de faire un traitement en temps réel. D'autres signaux biomécaniques représentatifs de l'absorption des chocs de la jambe dans les articulations, comme le moment abducteur du

genou ainsi que les forces de contact dans les articulations, seraient pertinents à prédire si suffisamment de données d'entraînement étaient disponibles. Ces données n'étaient pas accessible pour l'instant en raison de la plus grande complexité d'équipement et de calculs nécessaires pour les générer.

Les trois tâches biomécaniques sont démontrées dans les chapitres 3, 4 et 5 en utilisant des données d'IMU. Les IMU sont situés principalement sur le pied, et aussi dans certain cas à d'autres positions sur la jambe, sur le bras et au niveau de la taille. Pour construire les signaux objectifs, des appareils de mesure additionnelles peuvent être nécessaires en lien avec des tâches de prédiction. La pression ressentie par le pied en fonction du temps est nécessaire pour la détection des événements de marche, identifiable avec des FSR sous le pied ou avec un tapis roulant instrumenté. Le tapis roulant est nécessaire aussi pour la tâche d'estimation des forces aux sols. Pour la classification de patrons de marche, aucun autre appareil n'est nécessaire pour définir les données objectives. La labélisation des données est effectuée manuellement pour chaque enregistrement, selon un plan d'expérience préliminairement défini.

### 6.2.2 Définition des contraintes de conception du RC

Dans le chapitre 2, nous avons défini le concept de RC en partant du contexte général de l'AI, puis en spécifiant les avantages des RC par rapport aux autres types de méthodes de réseaux de neurones, soit la capacité de mémoire (réseaux récurrents) et leur facilité d'entraînement (poids fixes dans les couches cachée et d'entrée). Nous avons défini deux types d'implémentations du concept de RC. Le premier, le Echo States Network (ESN), est une des implémentations numériques bien connues dans la littérature, développée dès le début des années 2000, et qui a été utilisée pour tester les trois tâches biomécaniques dans les chapitres 3,4 et 5. Le second, le RC-MEMS, est un prototype de réservoir physique fabriqué dans notre groupe de recherche, basé sur la réponse non linéaire d'un oscillateur de Duffing (micropoutre en silicium bi-encastree), couplé avec un accéléromètre MEMS conventionnel (masse suspendue sur des ressorts) sensible aux accélérations externes. Le MEMS implique certaines contraintes supplémentaires à considérer dans les démonstrations avec le ESN. En raison des délais nécessaires pour l'acquisition des données de sortie du RC-MEMS de l'ordre de  $50 \mu s$ , le ESN doit se limiter à un nombre de neurones maximal de 100 nœuds avec une fréquence de traitement de 200 Hz sur les signaux biomécaniques. Le RC-MEMS est un réservoir à délai, ce qui implique que la fréquence de traitement est proportionnelle au nombre de neurones; on peut donc trouver un compromis entre ces deux paramètres (augmenter la fréquence de traitement en diminuant le nombre de neurones et vice-versa). De plus, comme le RC-MEMS est un accéléromètre uniaxial, le ESN représentatif doit utiliser une accélération unique d'entrée sans prétraitement.

### 6.2.3 Résumé des performances en biomécanique

La première tâche, abordée dans le chapitre 3, était la détection des événements de marche, utilisés pour suivre des paramètres temporels de la marche, comme la cadence ou la durée de la phase d'appui. Le RC peut détecter les événements de marche présents dans la phase d'appui du pied avec une précision entre 20 et 100 ms sur 5 événements de marche. Cette méthode est proche d'être aussi précise que les algorithmes performants retenus de la littérature. Nous avons validé la robustesse du RC pour cette tâche en testant le RC

sur un grand ensemble de participants et de conditions de marche. L'algorithme conserve pratiquement les mêmes performances dans presque toutes les conditions, si des données de marche similaires ont été utilisées lors de la phase d'entraînement. La prédiction des évènements de marche permettent ensuite d'identifier certains paramètres cliniques comme la cadence, mais ceux-ci n'ont pas été évalués dans le cadre de cette étude.

La seconde tâche, abordée dans le chapitre 4, était de faire la classification de différents patrons de marche utilisés comme stratégie pour limiter la progression de problèmes dégénératifs articulaires comme l'arthrose des genoux et des hanches. On exprime la performance d'un algorithme de classification par l'aire sous la courbe de la fonction d'efficacité du récepteur (ROC curve). Une performance de classification de l'ordre de 0,85 d'AUC est obtenue pour des conditions de marche représentatives d'une situation clinique réelle, soit avec un seul patient, avec une variabilité dans la vitesse de marche. La considération de plusieurs vitesses et plusieurs personnes à la fois augmentent le défi de généralisation du RC et peuvent réduire ses performances de classification.

La troisième tâche, abordée dans le chapitre 5, était la prédiction des forces au sol, dont la performance se mesure par l'erreur absolue moyenne entre la prédiction et le signal de référence. Le RC permettait de prédire les signaux de force au sol avec une erreur absolue moyenne d'environ 13 % par rapport au poids des patients (BW) avec un entraînement par kernel. La méthode de kernel est couplée avec la tâche de détection des évènements de marche afin de faire une évaluation préliminaire des paramètres temporels de la marche (cadence, période d'appui et proportion de la période d'appui sur celui du cycle). Les trois paramètres temporels utilisés comme variables de pondération ont obtenu une amélioration significative par rapport à la méthode d'entraînement uniforme. Une comparaison des performances de prédiction selon plusieurs localisations du capteur a montré que la position du dessus du pied permettait d'obtenir la prédiction la plus robuste possible pour l'estimation des forces au sol. Malgré l'amélioration apportée par la méthode kernel, une erreur de 13 % du BW nous montrent que la prédiction des forces au sol n'est pas encore suffisamment précise pour fournir une information fiable en clinique.

Des données d'entrée avec 1,2 ou 3 accélérations en parallèle (différents axes xyz de l'IMU) ont été testées. Nous avons observé que l'utilisation de plusieurs axes ne semblait pas améliorer la prédiction des trois tâches biomécaniques de GED et d'estimation des forces au sol et que l'axe AP du pied amenait généralement de meilleurs résultats de prédiction que les autres axes.

#### 6.2.4 Limitation de la méthode RC

Un défi important dans l'utilisation du RC dans chacune des tâches considérées est la capacité de généralisation. Les RC sont des réseaux de neurones avec un nombre de neurones relativement petit par rapport aux autres méthodes ANN numériques. Cette différence implique une difficulté de la méthode à généraliser les prédictions sur une grande quantité de patients ou de variabilité de contexte de marche. Dans le cas du GED, la tâche était relativement régulière et le ESN était en mesure de généraliser la prédiction pour plusieurs participants à la fois. Toutefois, les résultats de classification dans le chapitre 4 diminuent en fonction du nombre de patients et la prédiction du signal des forces au sol testé dans le chapitre 5 sur une large population reste trop imprécise pour l'instant pour être applicable en pratique. Dans la plupart des tâches, cela implique que la



réussite d'une prédiction en biomécanique de la marche diminue en fonction du nombre de patients ou de contextes de marche considérés. Pour l'instant, il serait plus viable en application d'accomplir des tâches de prédiction où l'entraînement est fait sur chaque patient individuellement.

## 6.3 Contributions

---

Ce travail a permis d'initier le développement d'une expertise du groupe de recherche AI-MEMS dans des applications en biomécanique du mouvement. Il a permis d'avoir une meilleure connaissance de la nature des signaux impliqués dans l'analyse de la marche (amplitude, contenu fréquentiel, variabilité) pour des signaux d'accélération et de force au sol.

La principale contribution de ce mémoire fut de développer les méthodes de traitement des données biomécaniques expérimentales et démontrer leur utilisation pour trois tâches de prédiction d'analyse de la marche. La démonstration sur une grande base de données de la tâche de GED a aussi permis d'aboutir à une première publication scientifique. Dans le cadre du GED, un système d'acquisition de données IMU et FSR pour le pied, ainsi que l'établissement d'une base de données de marche à l'université de Sherbrooke fut une autre contribution de ce projet.

Une recherche des hyperparamètres a aussi permis de déterminer une configuration du ESN qui permettait d'obtenir des résultats encourageant sur les trois tâches à la fois, ce qui a permis de mettre en valeur le fait qu'un seul réservoir est un outil exploitable sur des tâches de différentes natures. Ce choix d'hyperparamètres s'est fait en optimisant le ratio entre la métrique du *kernel rank* et de la *generalisation rank* sur des signaux, comme expliqué dans l'annexe C.

Dans le chapitre 5, la méthode d'entraînement par kernel est une contribution de ce mémoire. Cette méthode d'entraînement est un outil se voulant générique et exploitable avec des variables de pondérations arbitraires. Elle a été testée sur les paramètres temporels de la marche, mais d'autres types de paramètres pourraient éventuellement être utilisés comme variables de pondération.

Une contribution additionnelle fut effectuée avec le groupe de recherche de kinésiologie du CIRRIIS à Québec dont le but était de faire l'optimisation de la position du capteur IMU pour maximiser les performances du RC, comme abordée dans le chapitre 5. Cette contribution a aussi abouti à la rédaction d'un article scientifique.

## 6.4 Travaux futurs

---

Au terme de ce projet, deux éléments sont en cours de développement dans les activités de notre groupe de recherche en lien avec la biomécanique. Le premier élément est la mise en place d'un essai randomisé contrôlé mettant en application l'algorithme du ESN pour la détection de patrons de marche avec des patients atteints d'arthrose au genou. Le second élément est la démonstration expérimentale de résolution de tâches d'analyse de la marche avec le capteur accéléromètre MEMS.

### 6.4.1 Application du ESN dans une situation clinique

---

Plusieurs tâches ont été explorées dans ce mémoire qui sont pertinentes dans le cadre du suivi de la progression de l'arthrose. L'une des prochaines étapes du projet est d'utiliser l'algorithme numérique en application clinique. L'algorithme ESN sera implémenté dans un capteur portatif intégrant un microcontrôleur et un capteur IMU afin de faire la correction de la démarche des patients. L'objectif est de détecter en temps réel si le patient marche selon le patron prescrit par un clinicien lors d'une première rencontre clinique. L'entraînement du ESN sera fait lors de cette première rencontre sur chaque patient individuellement. Les patients utiliseront le capteur avec l'algorithme ESN en mode opération durant plusieurs semaines dans un environnement extérieur au laboratoire. Cela implique plusieurs éléments additionnels de conception matériel et algorithmique à considérer, notamment la mémoire disponible, l'autonomie du microcontrôleur et la gestion de la variabilité des données. Par exemple, comme le changement de chaussure ou le placement peut modifier l'orientation du capteur à chaque utilisation, une calibration du capteur lors de l'initialisation est nécessaire. Des cas de variabilité du mouvement doivent aussi être considérés, comme la présence d'arrêts fréquents (à l'épicerie ou dans un bureau), des mouvements du pied qui ne sont pas dans le domaine d'apprentissage du ESN, comme la voiture, le vélo, des pentes et des virages. Finalement, il est aussi nécessaire d'ajouter une fonctionnalité de rétroaction pour que l'utilisateur du capteur ESN puisse corriger efficacement sa marche. Une rétroaction par vibration sur un cellulaire est prévue dans le cadre de cette étude.

### 6.4.2 Démonstration expérimentale sur le RC-MEMS Choix de la tâche de démonstration

---

La seconde étape à venir dans ce projet est de faire une démonstration d'une tâche biomécanique avec le RC-MEMS. Nous avons choisi d'élaborer une démonstration de la tâche de classification des patrons de marche, car cette tâche est plus simple à mettre en œuvre (ne nécessite pas d'équipement de mesure de référence afin d'établir les signaux objectifs). De plus, cette tâche peut être très modulable en termes de niveau de difficulté (selon le nombre de patrons, le nombre de vitesses et de participants considérés). Une première étude effectuée sur un moteur linéaire a permis de tester cette tâche de classification, mais aucun résultat pertinent ne fut obtenu en raison de la grande distorsion des signaux biomécaniques générées expérimentalement. La prochaine étape est donc de faire la démonstration expérimentale directement sur le pied d'un humain. Nous allons fixer le RC-MEMS directement sur le pied d'un marcheur et le connecter à un appareillage de mesure et d'alimentation externe. Les signaux générés n'auront donc pas de distorsion et seront représentatifs de ceux utilisés dans les démonstrations du ESN dans le chapitre 4. Des résultats préliminaires de classification dans le MEMS-RC sur trois vitesses de marche et sur les patrons TO et TL ont montré des résultats très encourageants. Les travaux futurs se concentreront donc sur des tests sur une plus grande population. Ces tests sont confinés pour l'instant dans un laboratoire en raison des équipements de mesure externe du RC-MEMS. Le développement d'un PCB intégré des fonctionnalités du RC-MEMS développé dans notre groupe permettrait aussi d'effectuer l'application du RC-MEMS dans des environnements écologiques.

# Annexe A

## Procédure de synchronisation des données de force aux sols

### A.1 Objectif

L'objectif de ce document est d'expliquer la procédure de synchronisation entre les données de force et d'accélération provenant des prise de mesure de marche sur tapis roulant fait dans le cadre du projet FRQ-G0E4521N. Ce projet à pour but d'identifier la localisation optimal des sensors pour l'utilisation d'algorithme RC, c'est pourquoi plusieurs IMUs positionnés à cinq endroits sur la jambes sont utilisés pour chaque tests.

#### A.1.1 Nature des données

Les signaux ont été enregistré sur 21 personnes, chacun faisant trois enregistrements de marche de 2 minutes.

Les trois enregistrements ont été fait selon des vitesses différentes : une avec une vitesse normale (normal walk, NW), suggéré par le patient, la seconde (slow, S) à une vitesse plus faible de 20% que le test NW et la dernière (fast, F), à une vitesse plus grande de 20 % que le test NW.

Pour chaque tests, 5 capteurs IMU ont été positionné sur la jambe comme montré dans la Fig. A.1. Les tests étaient fait sur un tapis roulant instrumenté afin d'enregistrer les forces de réaction au sol. L'enregistrement des 6 signaux (forces + accéléros) ont été fait de manière non synchronisé pour chaque tests.

En somme, cela fait donc 6 signaux (dont 5 d'accélération et 1 de force au sol) que l'on doit synchroniser selon un vecteur temporel homogène avant de pouvoir exploiter les données.

#### A.1.2 Étape 1 : Échantillonnage homogène et filtrage des signaux

Les signaux d'accélération sont premièrement ré échantillonné à une fréquence de 1000Hz, identique à celle des signaux de forces. Les signaux de force sont ensuite filtré avec un filtre passe bas de 50 Hz de type Butterworth d'ordre 2 afin d'enlever le bruit haute fréquence. Les peaks de bruits observables dans les données dans le signal d'accélération se trouve environs à 93 Hz, 124 Hz et 185 Hz et un derniers pics plus faible à environ 372 Hz. Le Peaks de bruit à 185 Hz est le plus évelé, avec une amplitude de 3.4 N.

#### A.1.3 Principe de synchronisation

La stratégie utilisée afin de faire la synchronisation entre des signaux est de maximiser la corrélation croisé (Cross-correlation) [135], définit comme,

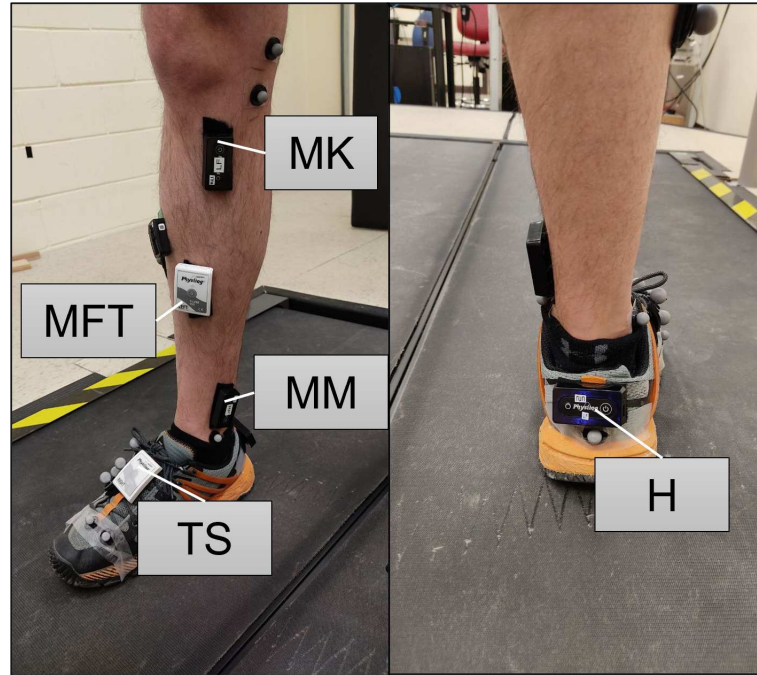


Figure A.1 – Position des 5 capteurs d'accélération

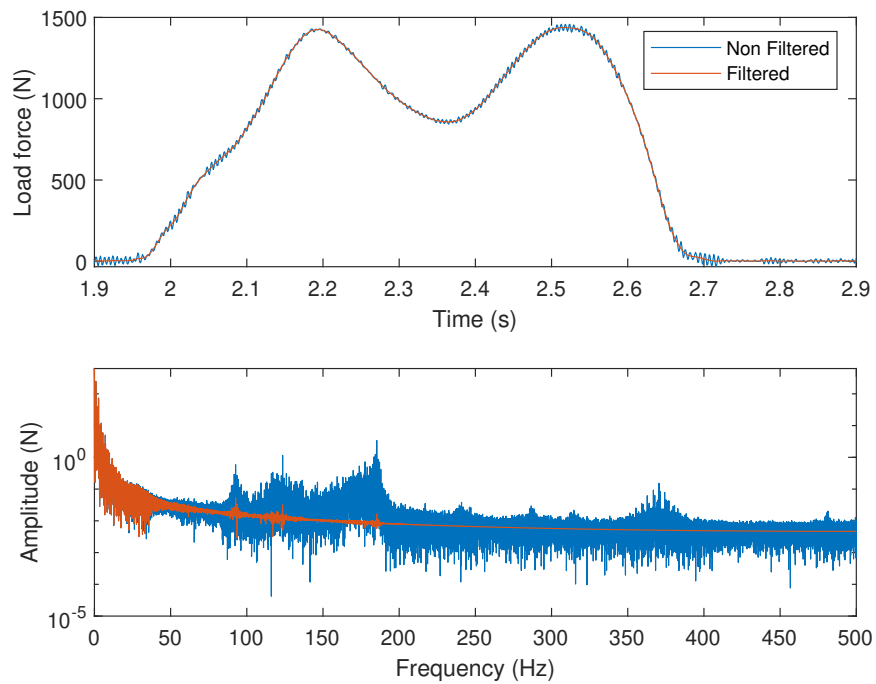


Figure A.2 – Filtrage du signal de force afin d'éliminer le bruit haute fréquence

$$[f \star g](n) = \sum_{m=-\text{inf}}^{\text{inf}} f(m) g(m + n) \tag{A.1}$$

Cette fonction est implémenté dans matlab [136] et est utilisé comme fonction de base

dans les sections A.1.4 et A.1.5. Elle représente le produit scalaire entre le signal  $f$  et un signal  $g$ , déplacé de  $n$  pas de temps par rapport au début du signal  $f$ . Deux signaux similaires auront une corrélation maximale si le décalage  $n$  choisi permet d'avoir des formes qui se chevauchent parfaitement dans le temps. Si les signaux sont positionnés de manière opposée, de sorte que les valeurs positives de  $F(m)$  sont multipliées avec les valeurs négatives de  $g(m+n)$ , la fonction d'auto-corrélation sera alors minimale.

Ainsi, la fonction de corrélation croisée est utilisée afin de déterminer le pic maximale de délai entre deux signaux, qui devrait, à priori, être le point de synchronisation optimal si ceux-ci sont suffisamment similaires.

### A.1.4 Étape 2 : Synchronisation de l'accélération avec la force de réaction au sol

On souhaite ici trouver le maximum de la fonction de corrélation croisée entre le carré du signal d'accélération  $a_{TS}$  et un signal dérivé de la force  $\phi$ ,

$$i_{syncroTS} = \mathbf{argmax}_n ([a_{TS}^2 \star \phi](n)) \quad (\text{A.2})$$

Où  $\star$  correspond à l'opération de corrélation croisée de l'équation A.1.

#### A.1.4.1 Principe de synchronisation

Le principe de synchronisation se base sur le fait que les signaux d'accélération du pied présentent deux pics majeurs durant un cycle de marche. Le premier pic survient lorsque le pied commence à toucher le sol souvent appelé le initial contact (IC) ou le heel strike (HS), et le second lorsque le pied quitte le sol, appelé le final contact (FC) ou le toe off (TO).

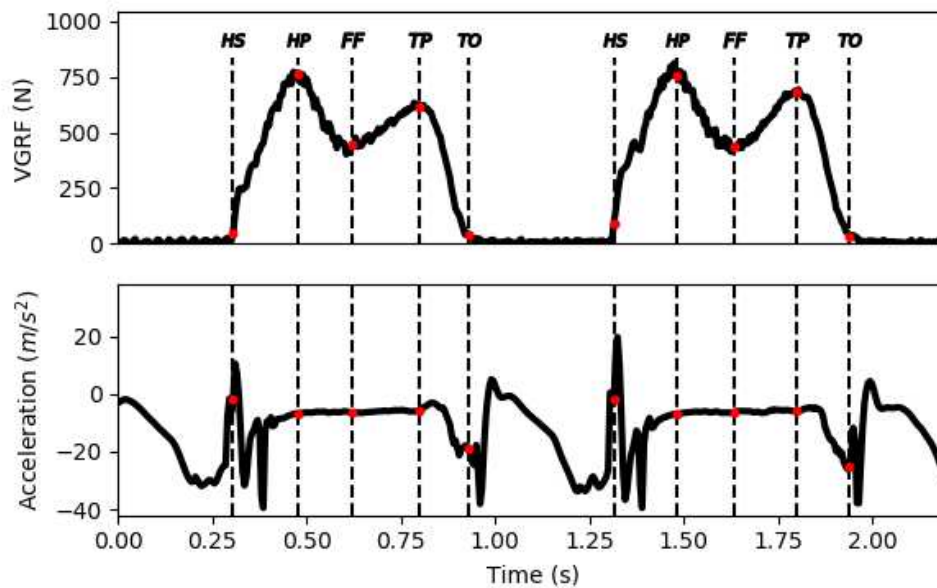
Comme on peut le voir sur la Fig. A.3, ces deux pics d'accélération sont synchronisés avec les fronts montant et descendant de la force au sol. On souhaite optimiser la corrélation entre les pics d'accélération et les fronts montant et descendant de la force de réaction. Pour ce faire, on définit deux signaux : le carré de l'accélération, puis un signal  $\phi$  comprenant des pics positionnés au front montant et descendant des forces verticales au sol (FV).

#### A.1.4.2 Choix de l'accélération

Afin de synchroniser les accélérations à ceux des signaux de force, on choisit un parmi les 5 signaux d'accélération ayant une meilleure sensibilité des pics en lien à la fois au coup de talon lors du premier contact au sol du pied et du début de la phase de swing, lorsque la pointe du pied quitte le sol. L'accélération positionnée au TS ( $a_{TS}$ ) est donc choisie comme la position ayant le plus de chance de voir apparaître clairement les deux pics dans le cycle.

#### A.1.4.3 Définition du signal $\phi$ dérivé de la force de réaction au sol

Le second signal  $\phi$  correspond à une version de la force verticale (FV) transformée afin d'avoir des valeurs négatives pour toutes les valeurs de FV au-dessus d'un certain seuil et positives au-dessous de ce seuil, puis en ayant des pics au front montant et descendant. Le



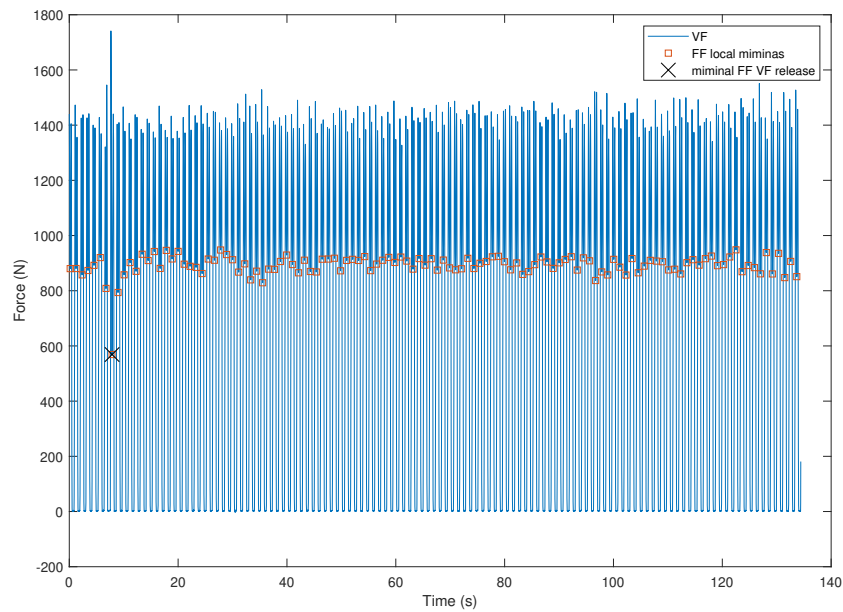
**Figure A.3** – Évènements de marche détectés avec les forces au sol et mise en évidence des évènements de marche sur le signal d'accélération, ici provenant de l'axe AP sur le dessus du pied.

fait de mettre de valeur positive pour les faibles valeur de FV et négative pour les hautes valeurs de FV permet d'ajuster la partie non nul dans le signal d'accélération (phase de swing) avec les zones de FV de valeurs nulles.

La corrélation sera maximisé si la partie non-nul du signal d'accélération est positionné dans les zones positive. Le fait d'avoir d'imposé dans  $\phi$  des valeurs positives pour les faibles valeurs de FV et négative autrement maximise la corrélation si la phase de swing (visible par les valeurs non nulles de l'accélération) se positionne dans les zones de nulle de FV. La fonction  $\phi$  est représenté en rouge dans la Fig. A.6 alors que FV est représenté en bleu.

Ce signal  $\phi$  ce construit selon les étapes suivantes :

1. Identification du redoux minimal de foot flat (FF) comme montré en exemple dans la Fig. A.4, correspondant à la hauteur minimale de la force de réaction au sol présent lors de la phase de FF dans le cycle de marche. Pour ce faire, on utilise la fonction `findpeaks` [137] de matlab sur le négatif de la force (-F). On sélectionne les minimum locaux trouvé entre une valeur de 100 N et du 99e percentile de la totalité du signal de force. On élimine aussi ceux ayant une proéminence P au dessous de 50 N. La proéminence est un paramètre donnée par la fonction `findpeaks` définissant la hauteur du pics relative au creux le plus proche dans le signaux (voir [137] pour la définition complète de la proéminence). Parmi l'ensemble des minimums locaux trouvé, on sélectionne le minimum de ces peaks pour définir le redoux minimal de FF.
2. On définit le niveaux de saturation haut comme minimum entre 500 N et le redoux minimal de FF.
3. Les points du signal de force ayant une valeur de force vertical plus grande que la limite de saturation haute sont définit à -200 N.



**Figure A.4** – Choix du redoux dans le signal de force vertical selon le minimum des tous les minimum locaux apparaissant lors du foot flat dans la phase d'appui du pied au sol

4. Les points en dessous de la limite de saturation basse de 100 N sont ensuite saturés à 100 N.
5. Les autres points entre 100 et la limite de saturation haute sont conservés à la même valeur.

La hauteur de saturation haute est ajustée en fonction du redoux minimale de FF afin d'éviter que les minimum locaux dans la phase d'appui de la force soit converti en valeur positive.

On peut ensuite calculer la fonction de corrélation croisée entre les signaux d'accélération du pied et de  $\phi$  montré dans la Fig. A.5. Le maximum de cette fonction correspond à l'index de lag choisi pour la synchronisation. L'allure finale des signaux synchronisés est montré dans la Fig. A.6.

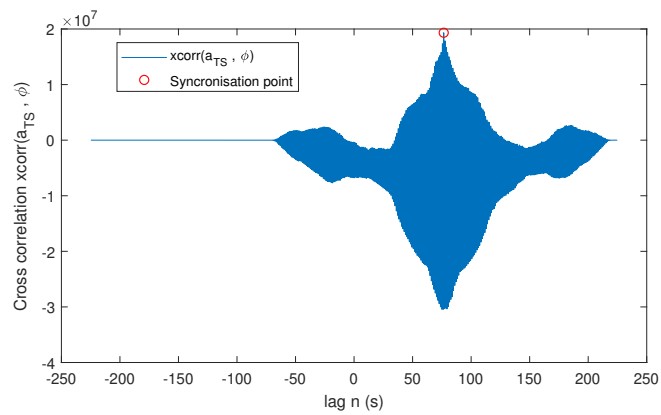


Figure A.5 – Valeur de la fonction de corrélation en fonction du lag  $n$  choisis

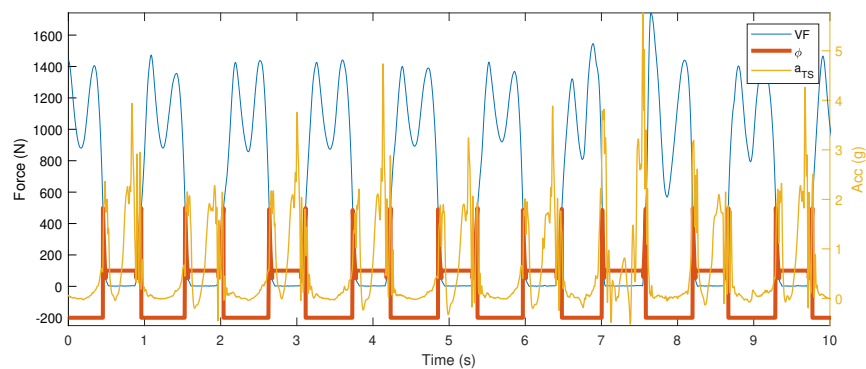


Figure A.6 – Aperçu de la synchronisation entre les signaux de force et de l'accélération TS



### A.1.5 Étape 3 : Synchronisation entre les accélérations

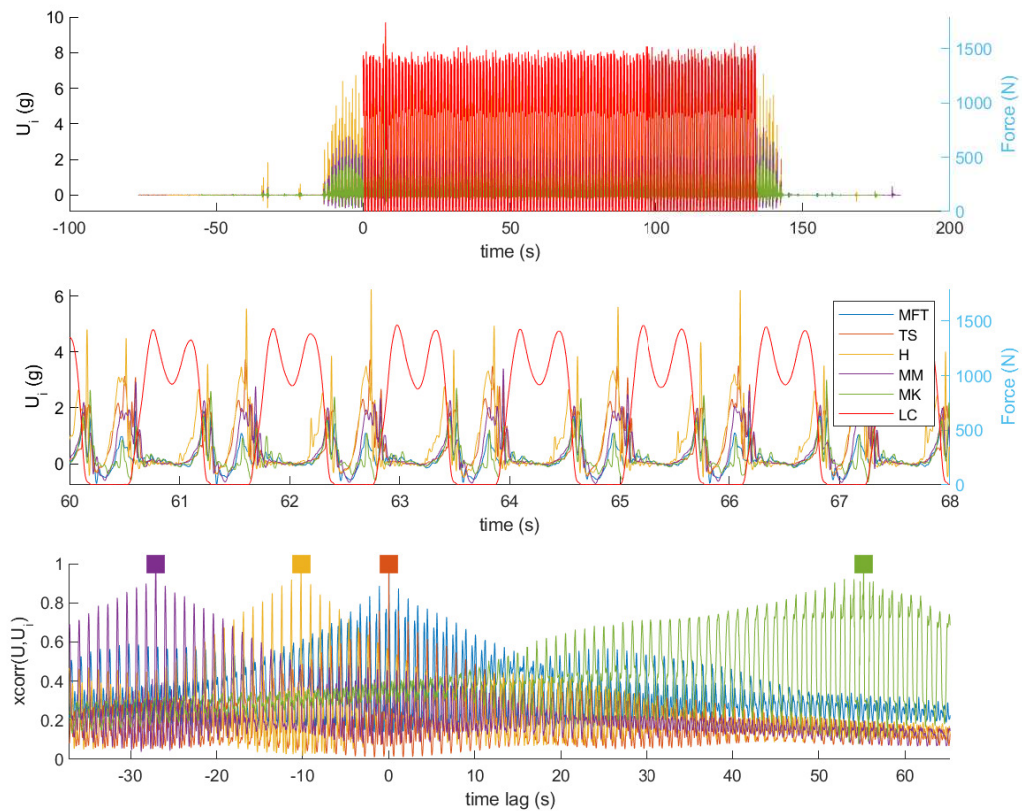
La seconde étape de synchronisation vise à maximiser la fonction de corrélation entre les accélération entre elle comme suit,

$$i_{syncro a_K} = i_{syncro TS} + \mathbf{argmax}_n ([a_{TS}^2 \star a_K^2](n)) \quad (\text{A.3})$$

On peut évaluer la corrélation entre les signaux d'accélérations avec le carré de l'accélération de base utilisé dans la sect. A.1.4 et le carré des autres accélérations ( $a_K$ ) individuellement.

Le lag correspondant au maximum de la fonction de corrélation est ajouté au lag déterminé en A.1.4 pour obtenir le même point de synchronisation. Il a été observé pour les patients 2,14,15 et 21 que les accélérations MK était mal synchronisé en phase avec l'accélération TS. Ceci peut s'expliquer par la position plus distancé entre les capteurs qui occasionne dans ces cas là une forme de signaux trop différentes pour facilement corrélés directement les signaux. La synchronisation dans ces cas là a été fait en corrélant les signaux  $a_{MK}$  dans directement avec le signal  $\phi$  comme fait en A.1.4 et donc sans ajouté par la suite la valeurs  $i_{syncro TS}$ .

La figure



**Figure A.7** – Synchronisation de tous les autres signaux d'accélération afin de maximiser la fonction de corrélation entre TS et les autres individuellement

### A.1.6 Étape 4 : Remise à l'échantillonnage d'origine

---

Comme le traitement par la suite considère une fréquence d'échantillonnage de l'accélération de 200 Hz, l'index de synchronisation est finalement converti comme suit :

$$i_{syncro a_K} = i_{syncro a_K} \frac{200}{1000} \quad (\text{A.4})$$

et ce pour toutes les accélérations ( $K \rightarrow \text{MFT, TS, H, MM, MK}$ ).

# Annexe B

## Identifying gait events with Ground reaction force preprocessing

Many preprocessing steps were used to identify the gait event targets from the ground reaction force, either from FSR signal with the UDS database or the VGRF signals of the treadmill for the CIRRIS database. An explanation of the process on each database is given here.

### B.1 MAREA

The MAREA database provides two time index vectors of correct gait events identified for each timeseries [38]. These events were detected with force sensitive resistors (FSR) placed at both extremities of the foot. The two events were defined as the initial ground contact of the heel (HS) and the final ground contact of the toe (TO).

### B.2 UDS

Five gait events were identified with the signals of the two FSR under the foot. The signal  $\mathcal{F}_T$  was the measurement of FSR1 placed under the toe and the signal  $\mathcal{F}_H$  was the measurement of FSR2 placed under the heel, as shown in figure 3.2. Both FSR signals were low pass filtered at 15 Hz. The heel off (HO) and the heel strike (HS) events corresponded to the rising and falling edges of the signal  $\mathcal{F}_H$ . The toe-push (TP) and toe-off (TO) corresponded respectively to the largest peak and the falling edge of the signal  $\mathcal{F}_T$ . The following procedure explains how each event was determined :

1. Calculate the mean period  $\tau$  of the gait cycle as the first peak in the auto-correlation function of  $\mathcal{F}_H$ .  $\tau$  had a mean value of 1.2 seconds per step over all timeseries of the UDS database.
2. Calculate  $\kappa$ , the moving 98e percentile of the  $\mathcal{F}_H$  signal at every  $0.2\tau$  indices of the signal over a  $0.4\tau$  wide window. Interpolate  $\kappa$  for all time indices using the nearest index value, then apply a moving average to  $\kappa$  with a  $0.1\tau$  wide window.
3. Compute  $\tilde{\mathcal{F}}_H$  by setting to zero values that were under  $\kappa$ .
4. Compute  $\tilde{\mathcal{F}}_H^c$  by taking the convolution between  $\tilde{\mathcal{F}}_H$  and a ramp of 100 samples between -1 and 1, and normalize the result by its maximal absolute value.
5. Estimate the time of the HS and HO events respectively as the location of the maximum and the minimum peaks of  $\tilde{\mathcal{F}}_H^c$ , using a peak finder function [80] with a temporal threshold of  $0.7\tau$  and a height threshold of 0.5.
6. To find the event TP and TO, separate the signal  $\mathcal{F}_T$  on individual cycles  $x_l$ , using the time index of the HS. Each  $x_l$  corresponded to one gait cycle of the  $\mathcal{F}_T$  signal, and  $l$  corresponded to the gait cycle index from the beginning.

7. For each  $x_l$ , calculate  $x'_l = \frac{x_l - P1(x_l)}{P99(x_l)}$ , where  $P1(x_l)$  and  $P99(x_l)$  are the 1st and the 99th percentiles of  $x_l$ . The normalized value of  $\mathcal{F}_T$ , denoted as  $\mathcal{F}_T^n$ , is then defined by the concatenation of all  $x_l$ .
8. Compute  $\mathcal{F}_T^{n,c}$ , as in step 4, using  $\mathcal{F}_T^n$  and a 50 samples ramp.
9. Estimate the time of the TP and TO events, respectively, as the maximum peaks of  $\mathcal{F}_T^n$  and  $\mathcal{F}_T^{n,c}$ , as in step 5, using a height threshold of 0.15 to find the  $\mathcal{F}_T^n$  peaks and 0.5 to find the  $\mathcal{F}_T^{n,c}$  peaks.

A verification loop was performed to ensure that the number of events for each class within all cycles was equal to one. The cycles were then separated on individual cycles between two HS events. For each class of event, the position of each event in the gait cycle  $d_{i,l}$  was determined by the distance between the event index and the first index of its cycle (HS event), where  $i$  is related to the gait event class and  $l$  is the index of the gait cycle in the signal. The mean event position in the gait cycle  $\bar{d}_i$  was determined for each event class using the mean value of all event positions  $d_{i,l}$ ,  $\bar{d}_i = \frac{1}{N_{cycle}} \sum_{l=1}^N d_{i,l}$ .  $N_{cycle}$  is the number of gait cycle in the signal evaluated by the number of HS. Then, for each gait cycle and each event classes, the following steps were taken :

1. The number of indices of target event detected was counted.
2. If more then one indices of gait event were found in the gait cycle, the indices were compared using the distance  $d_{i,l}$  from the begining of the cycle. The event with the smallest difference from  $\bar{d}$  was kept.
3. If no event was found, the true gait event was placed at a distance  $\bar{d}$  from the begining of the cycle.

A last step was performed to connect into a single record the signals of the four individual records made on the same track for each patient. Both acceleration and target gait events of the four recorded signals were joined together at the HO events index. The HO events were chosen for connection because there is almost no amplitude in the acceleration during this part of the cycle, as shown in the figure 3.4d. This was limiting jumps in the acceleration signal at connection points. To do so, the data from the start to the last index before the first HO event as well as the data from the last HO event to the end were deleted from each signal of acceleration and gait events target. Each group of signals from the same track and with the same patient were then concatenated.

## B.3 CIRRIIS

---

The gait events were detected with the vertical ground reaction force (VGRF) from the instrumented treadmill. Five types of events were identified from these data, using the following procedure :

1. HS and TO events were obtained from the position of the maximal and minimal peaks of the convolution between the VGRF and a ramp of 50 points between -1 and 1.
2. TP events were the positions of the local maximum peaks in the VGRF before the TO event in a window of 50 ms.

3. HP events were the positions of the first local maximum peaks in a window of 25 ms after the HS.
4. FF were the positions of the central local minimum peak in a 50 ms window between the HP and the TP events.

The number of gait cycles recorded experimentally in the CIRRIS database was lower, around 120 cycles per patient and per gait condition. A final step was taken to increase the number of walking cycles for training and testing. The acceleration and gait event signals were first divided into individual cycles, separated at the mid-stance indices (MS), corresponding to the FF indices for CIRRIS as explain in section 3.3.2. The acceleration and target gait event signals were then generated by a taking 200 gait cycles randomly chosen through all of the individual cycles and were then put together at the MS index. As mention in section 3.3.2, the MS event was chosen to connect acceleration because almost no amplitude is observed at this event of the cycle, limiting jumps at connection points in the acceleration.

# Annexe C

## Hyperparameters optimization

The hyperparameters (HP) for the input and the bias scaling, the spectral radius and the leaking rate were identified in the literature as those having the most influence on the computing performance of the ESN [72]. The other ESN HP were taken into account from preliminary optimization tests and did not show a significant influence on either the CHARC metrics (as explain below) or the GED performance.

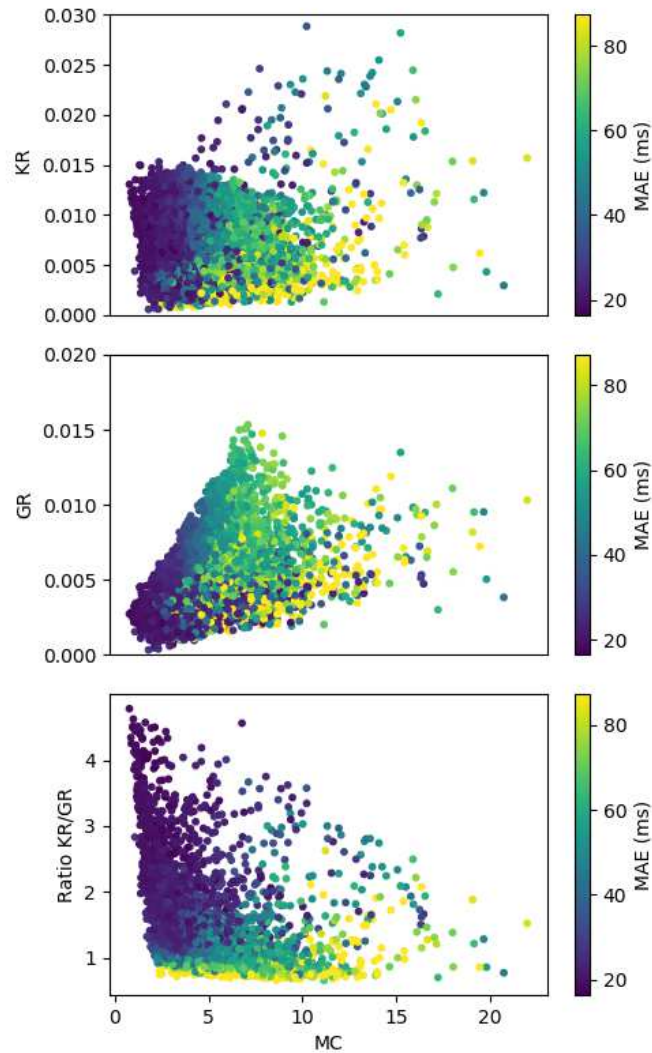
**Table C.1** – Echo State Network Hyperparameters

Number of nodes	N	100
Leaking rate	$\alpha$	0.1053
Hidden layer spectral raduis	$\rho$	0.7471
Hidden layer sparsity	P	.21
Input layer scaling	$S_i$	2.300
Input layer Sparsity	$P_i$	0.41
Input layer bias scaling	$S_b$	2.911
Regularization parameter	$\gamma$	1.e-6

These first four hyperparameters presented in table C.1 were thus optimized in order to reduce the error of the ESN for GED. A fast RC characterisation method called CHARC, proposed by [138], was used to obtain the best ESN HP. The CHARC method consists in a numerical procedure that estimates the performance of the ESN with metrics that are independent to the GED task. These metrics are much faster to compute with the ESN then the GED performance over all databases.

Three metrics are estimated using CHARC : the memory capacity (MC), the kernel rank (KR) and the generalization rank (GR). The MC is a measure of the ability to regenerate an input time series from the past states of the ESN. The KR is a metric for the ability to distinguish between different time series : a large value of KR means that the output states of the ESN are linearly independent for many different time series. The GR is a metric for measuring the ability to generalize identical inputs : a low value of GR means that the output states of the ESN computed with two different inputs remains almost linearly dependent when the input time series used are similar up to a certain noise level.

The CHARC exploration of the ESN parameters was done by comparing CHARC metrics and GED MAE using the protocol proposed by [138]. A medium size dataset (30 time series of 300 gait cycles) from the CIRRIIS database was used due to its early data availability. It can be seen in figure C.1 that the ESN having the highest ratio KR/GR are also the ones with the highest probability to produce an optimal mean absolute error. In this work, the four first HP of table C.1 were then selected in order to maximize the ratio  $\frac{KR}{GR}$ .

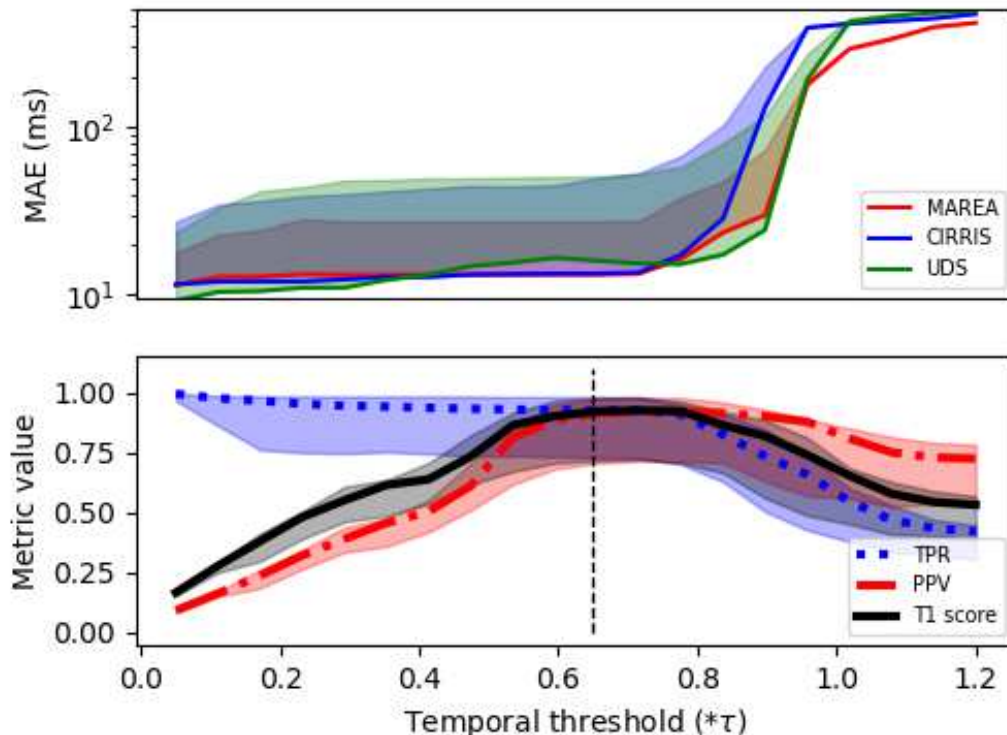


**Figure C.1** – Relations between the CHARC metrics and the mean absolute error of the GED task for all events. Exploration was done on the CIRRIIS database with time series of 3 conditions : Normal, right Toe out and right side trunk leaning. 5 different walking speeds and both subjects were selected (30 time series).

## C.1 Influence of peaks detection threshold

The temporal threshold used in the peak finder algorithm in the testing part of the ESN procedure, as presented in figure 3.1, influences the results of the metrics FPR, PPV and T1 significantly. This parameter sets the minimal distance between peaks detected in the output of the ESN. In figure C.2, the horizontal axis values represents the temporal threshold defined as a proportion of the average period of the gait cycle,  $\tau$ .

The threshold had a direct effect on the number of FP or FN obtained by the algorithm. A low threshold increased the sensitivity of the model and tended to increase the number of FP. This resulted in an over-sensitive algorithm with low PPV results. A high threshold tended to increase the number of FN. The algorithm will then missed many events and quickly increased the MAE and decreased the TPR. As shown in figure C.2, a value of 0.65-0.70  $\tau$  for the temporal threshold ensured balanced sensitivity and precision values and optimal T1 score. When the threshold value was over 0.8  $\tau$ , the MAE stabilized at a value of approximately 500 ms, close to the half period of a gait cycle, because prediction peaks were missed for half of the gait cycles. All results presented of this paper were obtained with a threshold value of 0.65  $\tau$ .



**Figure C.2** – GED performance according to the temporal threshold for peak detection (the shaded areas represent the variability over all the datasets and event classes). The lower, middle and upper line of the shaded areas correspond respectively to the 10th, 50th and 90th percentiles of the metric.



# Annexe D

## All walking conditions results of TKEO and TKEO algorithms

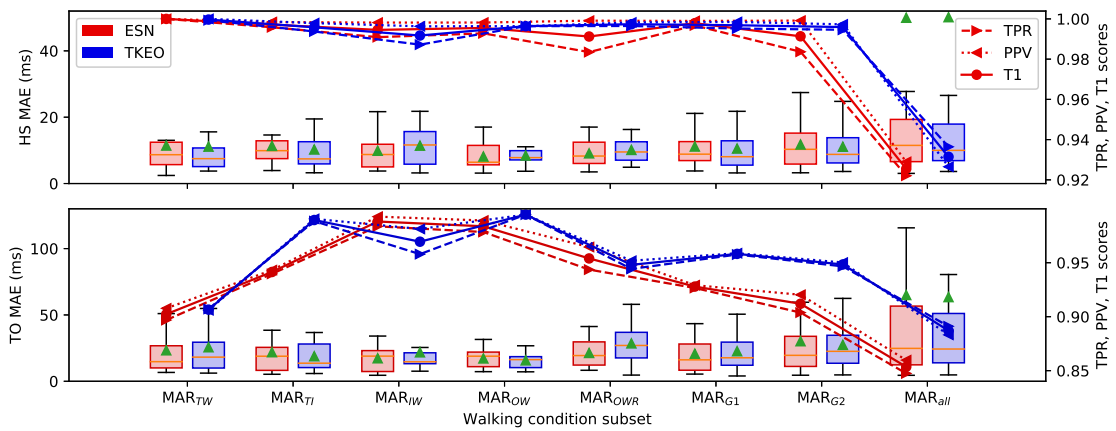
The figure D.1 shows results of the MAE error prediction of the HS and TO events under all the subsets defined in table 3.1.

### D.1 Processing comparison between ESN and TKEO

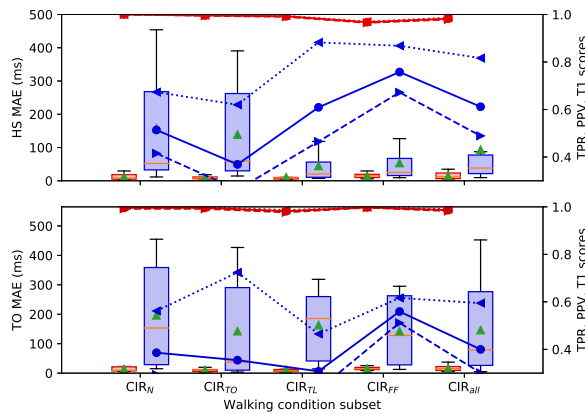
This section explains in further details the lack of robustness of the TKEO compared to ESN for other databases than the MAREA. In the first step of the TKEO process, the energy found in the acceleration signal is calculated. The HS events are then discriminated from the TO events, considering that the energy peaks are higher at the HS than the TO. This hypothesis was valid for the MAREA signals, where the sensor is closer to the heel than the toe. For the two other databases, the IMU sensors were located at the middle of the foot. The sensing saturation was also limited to  $\pm 2$  g for the UDS database. Most of the energy peaks measured at the HS and the TO in the walking cycle then had similar amplitudes. For most timeseries in the CIRRIIS and UDS databases, the TKEO mixed the HS and TO detections. We illustrate this in figure D.2, showing the resulting detection made by both algorithms for four gait cycles of one of the UDS timeseries. Both algorithms made a good detection in the two first gait cycles. In the last two gait cycles, the TKEO interchanged the HS and TO detections : the first energy signal computed by TKEO,  $\phi(n)$ , was used to estimate the HS peaks. In the last two gait cycles of  $\phi(n)$ , the HS event predictions were selected at the TO, because their peaks were slightly bigger than the HS peaks. These errors then induced a misdetection from the second energy signal  $\chi(n)$ , by changing values inside  $\phi(n)$  to zeros around HS peaks (section 3.3.4). In the second cycle, the TO event was also misdetected because the time distance with the event detected in cycle 3 was smaller than the threshold used in the peak finder algorithm. We observed such errors in the majority of the timeseries of the CIRRIIS and UDS databases.

These errors are good examples of how GED algorithms might be sensitive to data variability. To improve the TKEO algorithm, adjustments could be needed to preserve the ordered alternance of peak detections for each event, like rule based state machines [27] or alternance forcing detection [39].

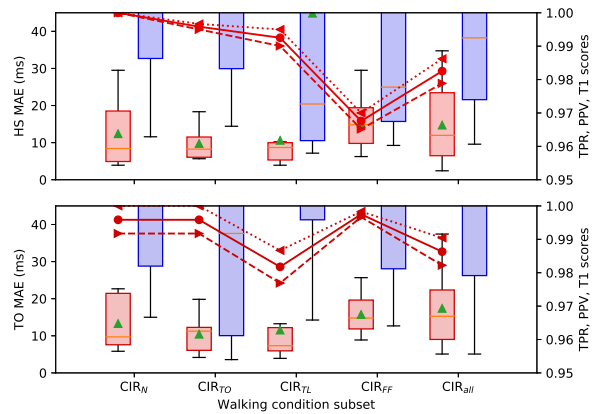
In comparison, the ESN algorithm predicts each type of event independently without being affected by the shape of the signal or by misdetection dependencies between events. We demonstrate that, without redesigning the algorithm, the ESN can correctly make GED with all of the walking conditions, even when the sensor position on the foot and the hardware system used is different.



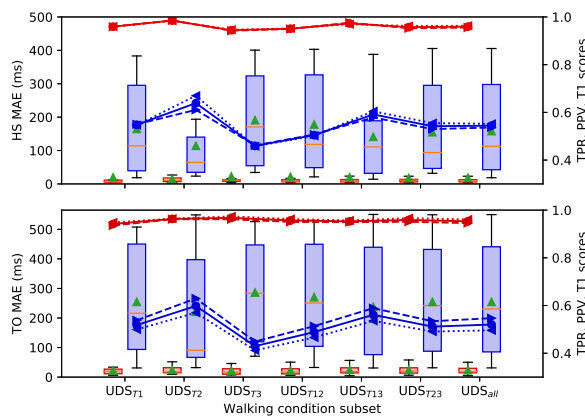
(a) MAREA database



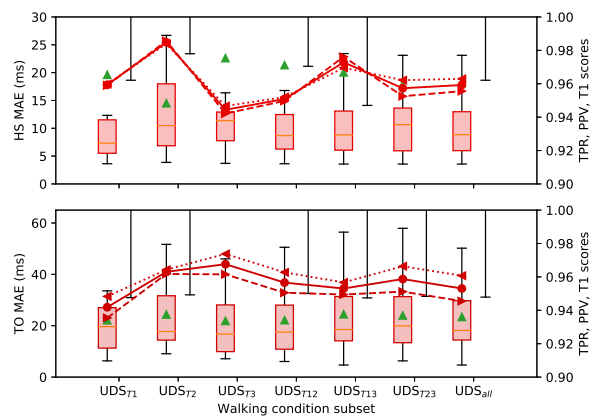
(b) CIRRIIS database



(c) CIRRIIS database (zoom on ESN results)

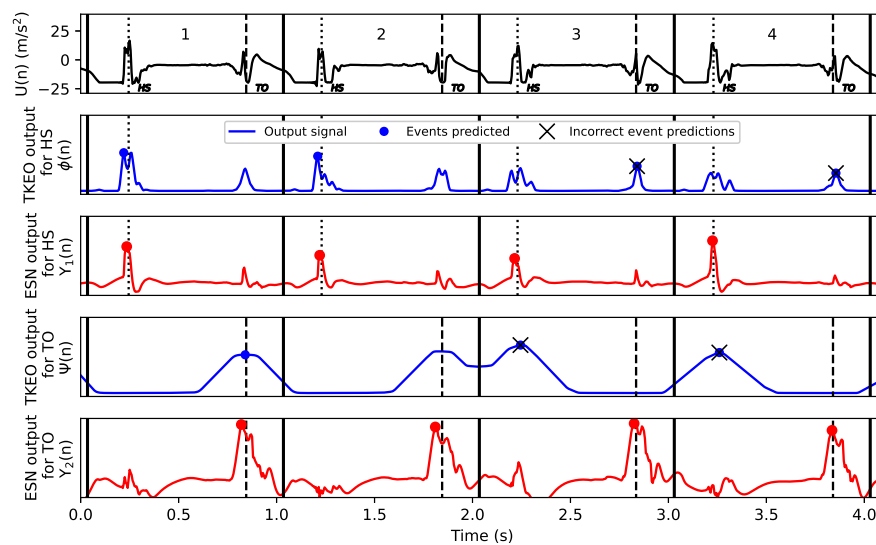


(d) UDS database



(e) UDS database (zoom on ESN results)

**Figure D.1** – MAE, TPR, PPV and T1 comparison between the ESN and TKEO algorithms for HS and TO events for multiple walking condition subsets



**Figure D.2** – Examples of predictions on a timeseries for the UDS database for the ESN and the TKEO. Four gait cycles are represented.  $U(n)$  is the AP acceleration. The other subplots show the output signals computed from the TKEO (blue lines),  $\phi(n)$  and  $\chi(n)$ , or the ESN (red lines),  $Y_1(n)$  and  $Y_2(n)$ . The events predicted (dots) were identified from the peak finder algorithm applied on each output signal. All the events were correctly predicted by the ESN. For the TKEO, the HS of cycles 1-2 were correctly predicted. HS and TO events of cycles 3-4 were misdected ( $\times$ ), and the TO of cycle 2 was not detected.

## Bibliographie

- [1] Jacquelin Perry. *Gait analysis : normal and pathological function*. SLACK. OCLC : ocm27816876.
- [2] Charu C. Aggarwal. *Neural Networks and Deep Learning : A Textbook*. Springer International Publishing.
- [3] Guillaume Dion, Salim Mejaouri, and Julien Sylvestre. Reservoir computing with a single delay-coupled non-linear mechanical oscillator. 124(15) :152132.
- [4] Bruno Barazani, Guillaume Dion, Anouar Idrissi-El Oudrhiri, Farnaz Ghaffari, and Julien Sylvestre. MICROMACHINED NEURO-PROCESSING ACCELEROMETER. page 4.
- [5] Matthew William Flood, Ben P. F. O’Callaghan, and Madeleine M. Lowery. Gait event detection from accelerometry using the teager–kaiser energy operator. 67(3) :658–666.
- [6] Anisoara Paraschiv-Ionescu, Christophe Perruchoud, Blaise Rutschmann, Eric Buchser, and Kamiar Aminian. Quantifying dimensions of physical behavior in chronic pain conditions. 13(1) :85.
- [7] Dawoon Jung, Mau Dung Nguyen, Mina Park, Jinwook Kim, and Kyung-Ryoul Mun. Multiple classification of gait using time-frequency representations and deep convolutional neural networks. 28(4) :997–1005.
- [8] Statistique Canada Gouvernement du Canada. Apparition des symptômes, diagnostic et gestion de l’arthrose. Last Modified : 2014-09-17.
- [9] J. Harlaar, E.M. Macri, and M. Wesseling. Osteoarthritis year in review 2021 : mechanics. page S1063458422000115.
- [10] Statistique Canada Gouvernement du Canada. Indice de masse corporelle, embonpoint ou obèse, autodéclaré corrigé, adulte, selon le groupe d’âge (18 ans et plus). Last Modified : 2021-09-08.
- [11] Ziyun Ding, Hannah L. Jarvis, Alexander N. Bennett, Richard Baker, and Anthony M. J. Bull. Higher knee contact forces might underlie increased osteoarthritis rates in high functioning amputees : A pilot study. 39(4) :850–860. \_eprint : <https://onlinelibrary.wiley.com/doi/pdf/10.1002/jor.24751>.
- [12] Lauren A. Sepp, Brian S. Baum, Erika Nelson-Wong, and Anne K. Silverman. Hip joint contact loading and muscle forces during running with a transtibial amputation. 143(3) :031012.
- [13] Parkinson’s disease - symptoms and causes.
- [14] FFRE | les causes de l’épilepsie.

- [15] Henry Chambers and David Sutherland. A practical guide to gait analysis. 10 :222–31.
- [16] Dayna Pool, Jane Valentine, Natasha Bear, Cyril J. Donnelly, Catherine Elliott, and Katherine Stannage. The orthotic and therapeutic effects following daily community applied functional electrical stimulation in children with unilateral spastic cerebral palsy : a randomised controlled trial. 15 :154.
- [17] Anisoara Paraschiv-Ionescu, Christopher J. Newman, Lena Carcreff, Corinna N. Gerber, Stephane Armand, and Kamiar Aminian. Locomotion and cadence detection using a single trunk-fixed accelerometer : validity for children with cerebral palsy in daily life-like conditions. 16(1) :24.
- [18] D. Patikas, S. Wolf, and L. Doderlein. Electromyographic evaluation of the sound and involved side during gait of spastic hemiplegic children with cerebral palsy. 12(9) :691–699.
- [19] Catarina Godinho, Veronique Ferret-Sena, Jose Brito, Filipe Melo, and Margarida Silva Dias. Postural behavior and parkinson's disease severity. In *2016 1st International Conference on Technology and Innovation in Sports, Health and Well-being (TISHW)*, pages 1–6. IEEE.
- [20] Michael J. Falvo and Gammon M. Earhart. Six-minute walk distance in persons with parkinson disease : A hierarchical regression model. 90(6) :1004–1008.
- [21] Gloria Vergara-Diaz, Kamila Osypiuk, Jeffrey M Hausdorff, Paolo Bonato, Brian J Gow, Jose GV Miranda, Lewis R Sudarsky, Daniel Tarsy, Michael D Fox, Paula Gardiner, Cathi A Thomas, Eric A Macklin, and Peter M Wayne. Tai chi for reducing dual-task gait variability, a potential mediator of fall risk in parkinson's disease : A pilot randomized controlled trial. 7 :2164956118775385.
- [22] Levodopa and the progression of parkinson's disease. page 11.
- [23] R.W. Selles, M.A.G. Formanoy, J.B.J. Bussmann, P.J. Janssens, and H.J. Stam. Automated estimation of initial and terminal contact timing using accelerometers ; development and validation in transtibial amputees and controls. 13(1) :81–88.
- [24] Stefan van Drongelen, Mariska Wesseling, Jana Holder, Andrea Meurer, and Felix Stief. Knee load distribution in hip osteoarthritis patients after total hip replacement. 8 :578030.
- [25] Jacquelin Perry, Judith M. Burnfield, JoAnne K. Gronley, and Sara J. Mulroy. Toe walking : Muscular demands at the ankle and knee. 84(1) :7–16.
- [26] Young-Kwan Kim, Ji-Yong Joo, Sang-Hyeok Jeong, Jean-Hong Jeon, and Dae-Young Jung. Effects of walking speed and age on the directional stride regularity and gait variability in treadmill walking. 30(6) :2899–2906.
- [27] Jan Rueterbories, Erika G. Spaich, and Ole K. Andersen. Gait event detection for use in FES rehabilitation by radial and tangential foot accelerations. 36(4) :502–508.

- [28] Susana Meireles, Mariska Wesseling, Colin R. Smith, Darryl G. Thelen, Sabine Verschueren, and Ilse Jonkers. Medial knee loading is altered in subjects with early osteoarthritis during gait but not during step-up-and-over task. 12(11) :e0187583.
- [29] Relationship between knee abduction moment with patellofemoral joint reaction force, stress and self-reported pain during stair descent in women with patellofemoral pain | elsevier enhanced reader.
- [30] Anisoara Ionescu. Foot clearance.
- [31] T.A. Gerbrands, M.F. Pisters, P.J.R. Theeven, S. Verschueren, and B. Vanwanseele. Lateral trunk lean and medializing the knee as gait strategies for knee osteoarthritis. 51 :247–253.
- [32] Jill Monson and Elizabeth Arendt. Rehabilitative protocols for select patellofemoral procedures and nonoperative management schemes. 20 :136–44.
- [33] Farzin Dadashi, Benoit Mariani, Stephane Rochat, Christophe Büla, Brigitte Santos-Eggimann, and Kamiar Aminian. Gait and foot clearance parameters obtained using shoe-worn inertial sensors in a large-population sample of older adults. 14(1) :443–457.
- [34] Rongmei Qiu, Rui Xu, Deqiang Wang, and Dong Ming. The effect of modifying foot progression angle on the knee loading parameters in healthy participants with different static foot postures. 81.
- [35] Ryan Pinto, Trevor Birmingham, Kristyn Leitch, Hayden Atkinson, Ian Jones, and J. Giffin. Reliability and validity of knee angles and moments in patients with osteoarthritis using a treadmill-based gait analysis system. 80.
- [36] Benedikt Fasel, Cyntia Duc, Farzin Dadashi, Flavien Bardyn, Martin Savary, Pierre-André Farine, and Kamiar Aminian. A wrist sensor and algorithm to determine instantaneous walking cadence and speed in daily life walking. 55(10) :1773–1785.
- [37] Dylan Kobsar, Jesse M. Charlton, and Michael A. Hunt. Individuals with knee osteoarthritis present increased gait pattern deviations as measured by a knee-specific gait deviation index. 72 :82–88.
- [38] Siddhartha Khandelwal and Nicholas Wickström. Evaluation of the performance of accelerometer-based gait event detection algorithms in different real-world scenarios using the MAREA gait database. 51 :84–90.
- [39] Siddhartha Khandelwal and Nicholas Wickstrom. Gait event detection in real-world environment for long-term applications : Incorporating domain knowledge into time-frequency analysis. 24(12) :1363–1372.
- [40] Stuart J. Russell, Peter Norvig, and Ernest Davis. *Artificial intelligence : a modern approach*. Prentice Hall series in artificial intelligence. Prentice Hall, 3rd ed edition.
- [41] G Dreyfus. 1 neural networks : An overview. page 83.

- [42] Sandro Skansi. *Introduction to Deep Learning : From Logical Calculus to Artificial Intelligence*. Undergraduate Topics in Computer Science. Springer International Publishing.
- [43] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. page 4.
- [44] J.J. Steil. Backpropagation-decorrelation : online recurrent learning with  $o(n)$  complexity. In *2004 IEEE International Joint Conference on Neural Networks (IEEE Cat. No.04CH37541)*, volume 2, pages 843–848 vol.2. ISSN : 1098-7576.
- [45] S Hochreiter and J Schmidhuber. Long short-term memory. 9(8) :1735–1780. Place : United States Publisher : MIT Press.
- [46] Herbert Jaeger. The “echo state” approach to analysing and training recurrent neural networks – with an erratum note. page 47.
- [47] Gerhard Goos, Juris Hartmanis, Jan van Leeuwen, David Hutchison, Takeo Kanade, Josef Kittler, Jon M Kleinberg, Alfred Kobsa, Friedemann Mattern, John C Mitchell, Moni Naor, Oscar Nierstrasz, C Pandu Rangan, and Bernhard Steffen. Lecture notes in computer science. page 1053.
- [48] Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. Optuna : A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2623–2631. ACM.
- [49] André Röhm. Symmetry-breaking bifurcations and reservoir computing in regular oscillator networks. page 248.
- [50] François Duport, Akram Akrout, Anteo Smerieri, Marc Haelterman, and Serge Massar. Analog input layer for optical reservoir computers. page 18.
- [51] François Duport, Anteo Smerieri, Akram Akrout, Marc Haelterman, and Serge Massar. Fully analogue photonic reservoir computer. 6(1) :22381.
- [52] M. L. Alomar, Erik S. Skibinsky-Gitlin, Christiam F. Frasser, Vincent Canals, Eugeni Isern, Miquel Roca, and Josep L. Rosselló. Efficient parallel implementation of reservoir computing systems. 32(7) :2299–2313.
- [53] Eddy Collin, Yuriy M. Bunkov, and Henri Godfrin. Addressing geometric nonlinearities with cantilever MEMS : beyond the duffing model. 82(23) :235416.
- [54] L. G. Villanueva, R. B. Karabalin, M. H. Matheny, D. Chi, J. E. Sader, and M. L. Roukes. Nonlinearity in nanomechanical cantilevers. 87(2) :024304.
- [55] Bruno Barazani, Guillaume Dion, Jean-François Morissette, Louis Beaudoin, and Julien Sylvestre. Microfabricated neuroaccelerometer : Integrating sensing and reservoir computing in MEMS. 29(3) :338–347.

- [56] Salim Mejaouri. Conception et fabrication de micro-résonateurs pour la réalisation d'une puce neuromorphique.pdf.
- [57] Jean C. Coulombe, Mark C. A. York, and Julien Sylvestre. Computing with networks of nonlinear mechanical oscillators. 12(6) :e0178663.
- [58] Guillaume Dion, Anouar Idrissi-El Oudrhiri, Bruno Barazani, Albert Tessier-Poirier, and Julien Sylvestre. Reservoir computing in MEMS. page 27.
- [59] Benoit Mariani, Mayté Castro Jiménez, François J. G. Vingerhoets, and Kamiar Aminian. On-shoe wearable sensors for gait and turning assessment of patients with parkinson's disease. 60(1) :155–158.
- [60] Kelly R. Sheerin, Duncan Reid, Denise Taylor, and Thor F. Besier. The effectiveness of real-time haptic feedback gait retraining for reducing resultant tibial acceleration with runners. 43 :173–180.
- [61] Joaquin A. Barrios, Kay M. Crossley, and Irene S. Davis. Gait retraining to reduce the knee adduction moment through real-time visual feedback of dynamic knee alignment. 43(11) :2208–2213.
- [62] Stefan Lambrecht, Anna Harutyunyan, Kevin Tanghe, Maarten Afschrift, Joris De Schutter, and Ilse Jonkers. Real-time gait event detection based on kinematic data coupled to a biomechanical model †. 17(4) :671.
- [63] Benedikt Fasel, Cyntia Duc, Farzin Dadashi, Flavien Bardyn, Martin Savary, Pierre-André Farine, and Kamiar Aminian. A wrist sensor and algorithm to determine instantaneous walking cadence and speed in daily life walking. 55(10) :1773–1785.
- [64] Michael Hanlon and Ross Anderson. Real-time gait event detection using wearable sensors. 30(4) :523–527.
- [65] J. Schuy, T. Mielke, M. Steinhausen, P. Beckerle, and S. Rinderknecht. Design & evaluation of a sensor minimal gait phase and situation detection algorithm of human walking. In *2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)*, pages 20–25. IEEE.
- [66] Min S. H. Aung, Sibylle B. Thies, Laurence P. J. Kenney, David Howard, Ruud W. Selles, Andrew H. Findlow, and John Y. Goulermas. Automated detection of instantaneous gait events using time frequency analysis and manifold embedding. 21(6) :908–916.
- [67] Anita Sant'Anna and Nicholas Wickström. A symbol-based approach to gait analysis from acceleration signals : Identification and detection of gait events and a new measure of gait symmetry. 14(5) :1180–1187.
- [68] R.R. Torrealba, J. Cappelletto, L. Fermin-León, J.C. Grieco, and G. Fernández-López. Statistics-based technique for automated detection of gait events from accelerometer signals. 46(22) :1483.



- [69] Andrea Mannini, Vincenzo Genovese, and Angelo Maria Sabatin. Online decoding of hidden markov models for gait event detection using foot-mounted gyroscopes. 18(4) :1122–1130.
- [70] Shuozhi Yang and Qingguo Li. Inertial sensor-based methods in walking speed estimation : A systematic review. 12(5) :6102–6116.
- [71] Herbert Jaeger. The “echo state” approach to analysing and training recurrent neural networks – with an erratum note. page 47.
- [72] Mantas Lukoševičius. A practical guide to applying echo state networks. In Grégoire Montavon, Geneviève B. Orr, and Klaus-Robert Müller, editors, *Neural Networks : Tricks of the Trade*, volume 7700, pages 659–686. Springer Berlin Heidelberg. Series Title : Lecture Notes in Computer Science.
- [73] Chenxi Sun, Moxian Song, Shenda Hong, and Hongyan Li. A Review of Designs and Applications of Echo State Networks. *arXiv :2012.02974 [cs]*, December 2020. arXiv : 2012.02974.
- [74] Weipeng Wang, Xiangpeng Liang, Maher Assaad, and Hadi Heidari. Wearable wrist-worn gesture recognition using echo state network. In *2019 26th IEEE International Conference on Electronics, Circuits and Systems (ICECS)*, pages 875–878. IEEE.
- [75] Yu Cao, Jian Huang, and Caihua Xiong. Single-layer learning-based predictive control with echo state network for pneumatic-muscle-actuators-driven exoskeleton. 13(1) :80–90.
- [76] Byunghun Choi, Changhoon Seo, Sanghoon Lee, Byungun Kim, and Dojong Kim. Swing control of a lower extremity exoskeleton using echo state networks. 50(1) :1328–1333.
- [77] scipy.signal.resample\_poly—SciPy v1.7.0 manual. [https://docs.scipy.org/doc/scipy/reference/generated/scipy.signal.resample\\_poly.html](https://docs.scipy.org/doc/scipy/reference/generated/scipy.signal.resample_poly.html). urldate : 2021-06-21.
- [78] T.A. Gerbrands, Martijn Pisters, P.J.R. Theeven, Sabine Verschueren, and Benedicte Vanwanseele. Lateral trunk lean and medializing the knee as gait strategies for knee osteoarthritis. 51.
- [79] Matthew Dale, Susan Stepney, and Martin Trefzer. Designing computational substrates using open-ended evolution. In *The 2020 Conference on Artificial Life*, pages 665–667. MIT Press.
- [80] scipy.signal.find\_peaks — SciPy v1.6.3 reference guide.
- [81] Find local maxima - MATLAB findpeaks. <https://www.mathworks.com/help/signal/ref/findpeaks.html>. urldate : 2021-06-21.
- [82] M.M. Skelly and H.J. Chizeck. Real-time gait event detection for paraplegic FES walking. 9(1) :59–68.

- [83] Sebastian Thrun, Lawrence K. Saul, and Bernhard Schölkopf. *Advances in Neural Information Processing Systems 16 : Proceedings of the 2003 Conference*. MIT Press.
- [84] Rolf Moe-Nilssen, Mona Kristin Aaslund, Caroline Hodt-Billington, and Jorunn L. Helbostad. Gait variability measures may represent different constructs. 32(1) :98–101.
- [85] René Schwesig, Siegfried Leuchte, David Fischer, Regina Ullmann, and Alexander Kluttig. Inertial sensor based reference gait data for healthy subjects. 33(4) :673–678.
- [86] Josefine E Naili, Anna-Clara Esbjörnsson, Maura D Iversen, Michael H Schwartz, Margareta Hedström, Charlotte K Häger, and Eva W Broström. The impact of symptomatic knee osteoarthritis on overall gait pattern deviations and its association with performance-based measures and patient-reported outcomes. 24(3) :536–546.
- [87] Juan C. Perez-Ibarra, Harry Williams, Adriano A. G. Siqueira, and Hermano I. Krebs. Real-time identification of impaired gait phases using a single foot-mounted inertial sensor : Review and feasibility study. In *2018 7th IEEE International Conference on Biomedical Robotics and Biomechatronics (Biorob)*, pages 1157–1162. IEEE.
- [88] Jeffrey M Hausdorff. Gait variability : methods, modeling and meaning. 2(1) :19.
- [89] Maria De Marsico and Alessio Mecca. A survey on gait recognition via wearable sensors. 52(4) :86 :1–86 :39.
- [90] Eric L. Radin, King H. Yang, Cheryl Riegger, Vince L. Kish, and John J. O'Connor. Relationship between lower limb dynamics and knee joint pain. 9(3) :398–405.
- [91] Cristine Agresta and Allison Brown. Gait retraining for injured and healthy runners using augmented feedback : A systematic literature review. 45(8) :576–584.
- [92] Daniel K. Y. Chen, Markus Haller, and Thor F. Besier. Wearable lower limb haptic feedback device for retraining foot progression angle and step width. 55 :177–183.
- [93] Haisheng Xia, Jesse M. Charlton, Peter B. Shull, and Michael A. Hunt. Portable, automated foot progression angle gait modification via a proof-of-concept haptic feedback-sensorized shoe. 107 :109789.
- [94] Junkai Xu, Tian Bao, Ung Hee Lee, Catherine Kinnaird, Wendy Carender, Yangjian Huang, Kathleen H. Sienko, and Peter B. Shull. Configurable, wearable sensing and vibrotactile feedback system for real-time postural balance and gait training : proof-of-concept. 14(1) :102.
- [95] Benoit Mariani, Hossein Rouhani, Xavier Crevoisier, and Kamiar Aminian. Quantitative estimation of foot-flat and stance phase of gait using foot-worn inertial sensors. 37(2) :229–234.

- [96] Espen A. F. Ihlen, Aner Weiss, Jorunn L. Helbostad, and Jeffrey M. Hausdorff. The discriminant value of phase-dependent local dynamic stability of daily life walking in older adult community-dwelling fallers and nonfallers. 2015 :1–11.
- [97] Bilal Fadlallah, Badong Chen, Andreas Keil, and Jose Principe. Weighted-permutation entropy : A complexity measure for time series incorporating amplitude information. page 7.
- [98] P. Parthasarathy and S. Vivekanandan. A typical IoT architecture-based regular monitoring of arthritis disease using time wrapping algorithm. 42(3) :222–232. Publisher : Taylor & Francis \_eprint : <https://doi.org/10.1080/1206212X.2018.1457471>.
- [99] Miroslav Kubat. *An Introduction to Machine Learning*. Springer International Publishing.
- [100] T. Charles Clancy, Awais Khawar, and Timothy R. Newman. Robust signal classification using unsupervised learning. 10(4) :1289–1299. Conference Name : IEEE Transactions on Wireless Communications.
- [101] Jonas Nordhaug Myhre, Karl Øyvind Mikalsen, Sigurd Løkse, and Robert Jenssen. Robust clustering using a kNN mode seeking ensemble. 76 :491–505.
- [102] Manisha Kowdiki and Arti Khaparde. Algorithmic analysis on deer hunting-based grey wolf for dynamic time wrapping-based hand gesture recognition. In *2022 International Conference for Advancement in Technology (ICONAT)*, pages 1–8.
- [103] G. Grégoire. Elements of statistics. 77 :13–37.
- [104] Claudia Nickel, Holger Brandt, and Christoph Busch. *Classification of Acceleration Data for Biometric Gait Recognition on Mobile Devices*. Pages : 66.
- [105] Claudia Nickel, Christoph Busch, Sathyanarayanan Rangarajan, and Manuel Möbius. Using hidden markov models for accelerometer-based biometric gait recognition. In *2011 IEEE 7th International Colloquium on Signal Processing and its Applications*, pages 58–63.
- [106] Vinay Kukreja, Deepak Kumar, and Amandeep Kaur. Deep learning in human gait recognition : An overview. In *2021 International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE)*, pages 9–13. IEEE.
- [107] Suryani Ilias, Nooritawati Md Tahir, Rozita Jailani, and Che Zawiyah Che Hasan. Classification of autism children gait patterns using neural network and support vector machine. In *2016 IEEE Symposium on Computer Applications & Industrial Electronics (ISCAIE)*, pages 52–56. IEEE.
- [108] Yongjia Zhao and Suiping Zhou. Wearable device-based gait recognition using angle embedded gait dynamic images and a convolutional neural network. 17(3) :478.

- [109] Alireza Sepas-Moghaddam and Ali Etemad. Deep gait recognition : A survey. pages 1–1. Conference Name : IEEE Transactions on Pattern Analysis and Machine Intelligence.
- [110] Kun Hu, Zhiyong Wang, Wei Wang, Kaylena A. Ehgoetz Martens, Liang Wang, Tieniu Tan, Simon J. G. Lewis, and David Dagan Feng. Graph sequence recurrent neural network for vision-based freezing of gait detection. 29 :1890–1901.
- [111] Mantas Lukoševičius and Herbert Jaeger. Reservoir computing approaches to recurrent neural network training. 3(3) :127–149.
- [112] Tom Fawcett. An introduction to ROC analysis. 27(8) :861–874.
- [113] Confusion matrix - an overview | ScienceDirect topics.
- [114] Maximilian Aurbach, Kilian Wagner, Franz Süß, and Sebastian Dendorfer. Implementation and validation of human kinematics measured using IMUs for musculoskeletal simulations by the evaluation of joint reaction forces. In Almir Badnjevic, editor, *CMBEBIH 2017*, volume 62, pages 205–211. Springer Singapore. Series Title : IFMBE Proceedings.
- [115] Eileen Chih-Ying Yang and Ming-Hsu Mao. 3d analysis system for estimating intersegmental forces and moments exerted on human lower limbs during walking motion. 73 :171–179.
- [116] David V. Thiel, Jonathan Shepherd, Hugo G. Espinosa, Megan Kenny, Katrien Fischer, Matthew Worsey, Akifumi Matsuo, and Tomohito Wada. Predicting ground reaction forces in sprint running using a shank mounted inertial measurement unit. 2(6) :199. Number : 6 Publisher : Multidisciplinary Digital Publishing Institute.
- [117] Yuzhu Guo, Fabio Storm, Yifan Zhao, Stephen A. Billings, Aleksandar Pavic, Claudia Mazzà, and Ling-Zhong Guo. A new proxy measurement algorithm with application to the estimation of vertical ground reaction forces using wearable sensors. 17(10) :2181.
- [118] B. Mariani, S. Rochat, C. J. Büla, and K. Aminian. Heel and toe clearance estimation for gait analysis using wireless inertial sensors. 59(11) :3162–3168.
- [119] SimTK : OpenSim : Accueil du projet.
- [120] Lucas Engelhardt, Maximilian Melzner, Linda Havelkova, Pavel Fiala, Patrik Christen, Sebastian Dendorfer, and Ulrich Simon. A new musculoskeletal AnyBody™ detailed hand model. 24(7) :777–787. Publisher : Taylor & Francis \_eprint : <https://doi.org/10.1080/10255842.2020.1851367>.
- [121] Damian Raper, Jeremy Witchalls, Elissa Phillips, Emma Knight, Michael Drew, and Gordon Waddington. Use of a tibial accelerometer to measure ground reaction force in running : A reliability and validity comparison with force plates. 21.

- [122] Reed D. Gurchiek, Ryan S. McGinnis, Alan R. Needle, Jeffrey M. McBride, and Herman van Werkhoven. The use of a single inertial sensor to estimate 3-dimensional ground reaction force during accelerative running tasks. 61 :263–268.
- [123] Daniel Wundersitz, Paul Gustin, Chris Richter, and Kevin Netto. *Validity of wearable technology to measure peak impact during high-intensity treadmill running*. Pages : 425.
- [124] E. Shahabpoor and A. Pavic. Estimation of vertical walking ground reaction force in real-life environments using single IMU sensor. 79 :181–190.
- [125] Angelos Karatsidis, Giovanni Bellusci, H. Schepers, Mark de Zee, Michael Andersen, and Peter Veltink. Estimation of ground reaction forces and moments during gait using only inertial motion capture. 17(12) :75.
- [126] Edgar Charry, Wenzheng Hu, Muhammad Umer, Andrew Ronchi, and Simon Taylor. Study on estimation of peak ground reaction forces using tibial accelerations in running. In *2013 IEEE Eighth International Conference on Intelligent Sensors, Sensor Networks and Information Processing*, pages 288–293.
- [127] Jennifer M. Neugebauer, Kelsey H. Collins, and David A. Hawkins. Ground reaction force estimates from ActiGraph GT3x+ hip accelerations. 9(6) :e99023.
- [128] Jennifer M. Neugebauer, David A. Hawkins, and Laurel Beckett. Estimating youth locomotion ground reaction forces using an accelerometer-based activity monitor. 7(10) :e48182.
- [129] Frank J. Wouda, Matteo Giuberti, Giovanni Bellusci, Erik Maartens, Jasper Reenalda, Bert-Jan F. van Beijnum, and Peter H. Veltink. Estimation of vertical ground reaction forces and sagittal knee kinematics during running using three inertial sensors. 9.
- [130] Gustavo Leporace, Luiz Alberto Batista, Leonardo Metsavaht, and Jurandir Nadal. Residual analysis of ground reaction forces simulation during gait using neural networks with different configurations. In *2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 2812–2815. ISSN : 1558-4615.
- [131] Kieron Jie-Han Ngoh, Darwin Gouwanda, Alpha A. Gopalai, and Yu Zheng Chong. Estimation of vertical ground reaction force during running using neural network model and uniaxial accelerometer. 76 :269–273.
- [132] Chao Wang, Peter P. K. Chan, Ben M. F. Lam, Sizhong Wang, Janet H. Zhang, Zoe Y. S. Chan, Rosa H. M. Chan, Kevin K. W. Ho, and Roy T. H. Cheung. Real-time estimation of knee adduction moment for gait retraining in patients with knee osteoarthritis. 28(4) :888–894. Conference Name : IEEE Transactions on Neural Systems and Rehabilitation Engineering.
- [133] K.A. Costello. Ground reaction force patterns in knees with and without radiographic osteoarthritis and pain : descriptive analyses of a large cohort (the multicenter osteoarthritis study) | elsevier enhanced reader.

- [134] Alan K. Bourke, Alf Scotland, Florian Lipsmeier, Christian Gossens, and Michael Lindemann. Gait characteristics harvested during a smartphone-based self-administered 2-minute walk test in people with multiple sclerosis : Test-retest reliability and minimum detectable change. 20(20) :5906.
- [135] Cross-correlation. Page Version ID : 1031522391.
- [136] Cross-correlation - MATLAB xcorr.
- [137] Find local maxima - MATLAB findpeaks.
- [138] Matthew Dale, Julian F. Miller, Susan Stepney, and Martin A. Trefzer. A substrate-independent framework to characterize reservoir computers. 475(2226) :20180723.