A Framework for Recognizing Industrial Actions via Joint Angles

Avinash Kumar Singh, Mohamed Adjel, Vincent Bonnet, Robin Passama, Andrea Cherubini

A Framework for Recognizing Industrial Actions via Joint Angles

Avinash Kumar Singh*[1] Mohamed Adjel*[1,2] Vincent Bonnet[3] Robin Passama[1] and Andrea Cherubini[1]

***Abstract*— This paper proposes a novel framework for recognizing industrial actions, in the perspective of human-robot collaboration. Given a one second long measure of the human's motion, the framework can determine his/her action. The originality lies in the use of joint angles, instead of Cartesian coordinates. This design choice makes the framework sensor agnostic and invariant to affine transformations and to anthropometric differences. On AnDy dataset, we outperform the state of art classifier. Furthermore, we show that our framework is effective with limited training data, that it is subject independent, and that it is compatible with robotic real-time constraints. In terms of methodology, the framework is an original synergy of two antithetical schools of thought: model-based and data-based algorithms. Indeed, it is the cascade of an inverse kinematics estimator compliant with the International Society of Biomechanics recommendations, followed by a deep learning architecture based on Bidirectional Long Short Term Memory. We believe our work may pave the way to successful and fast action recognition with standard depth cameras, embedded on moving collaborative robots.**

I. INTRODUCTION

In recent years, industry is adopting more and more collaborative robots (*cobots*). This opens the door to new scenarios, where human and robot share the same workspace [1]. For a robot to be effective in this context, there must be a tight choreography between the two (human and robot). Typically, for successful human-robot collaboration, it is crucial that the robot understands human actions. This will be helpful in numerous scenarios and tasks including perilous environments, heavy object lifting, tool pick and place, etc. Human-robot collaboration should not only increase productivity; it should also reduce musculoskeletal disorders [2] induced by repetitive tasks performed in awkward postures. Therefore, online recognition of the human actions will also provide feedback on ergonomic risks [3].

Human action recognition has been an active research domain since more than two decades [4], [5]. Some researchers focus on recognizing human actions from RGB and RGB-D videos [6][7], while others utilize the position of the human joints (skeleton) [8][9], obtained from motion capture systems [10][11]. There are many labeled datasets online [4][5][9], to support or verify methodologies, particularly those based on machine learning. Yet, most of these datasets are limited to daily life actions, such as making coffee, reading the newspaper, using the telephone etc. Few focus on tasks which take place in industrial scenarios, such as pick/place a tool, screw, etc. As per our knowledge, there are only two industry-oriented motion capture datasets: InHard [12] and AnDY [13]. In InHard, humans perform all actions while standing, although this is not always the case in industrial setups. AnDY instead, covers a broader variety of postures (e.g., walking, sitting). Therefore, we decided to use AnDY dataset to test our approach.

In terms of methods, most recent works on human action recognition rely on deep learning. Researchers have used a variety of neural networks such as Convolutional Neural Networks [14][15][16], Recurrent Neural Networks [17][18][19], Graph Neural Networks [20][21] and combinations of the above [22]. All the cited studies have addressed the problem in the Cartesian space (XYZ coordinates of the human joint centers). In our work, we also opt for deep learning, but propose the use of joint angle space rather than Cartesian space.

In the context of Human-Robot Collaboration, two important characteristics are the low quality of data (since the sensor is often on-board the robot) and the need for fast processing. Specifically:

- sensor data can be noisy, incomplete or blurred (due to occlusions, limited field of view or robot motion),
- data must be processed and classified (e.g., actions must be recognized) at a framerate that is high enough to control a robot, at least at 5-10 Hz.

These issues make the action recognition problem much more challenging in Robotics than it is in the field of Computer Vision, where the papers [5][23][24] present solutions, which do not account for low quality data nor for framerate requisites. Besides, they do not easily generalize to:

- viewpoint (orientation and translation),
- subject-specific anthropometry,
- sensors other than cameras.

To address these aspects, one should train the classifier on datasets covering many cases, at a huge cost in terms of acquisition, labeling and training.

Our solution consists in mapping the 3D Cartesian coordinates of the human skeleton joint centers, which can be derived from diverse sources (RGB-D camera, stereophotogrammetric device, inertial suits, ...) to his/her joint angles. The joint angles representation has many advantages:

- it is sensor agnostic, i.e., one can apply it to any *motion capture device* measuring the human joint Cartesian coordinates; such device may be a dedicated motion capture sensor (e.g., optical, inertial, etc) or the combination of a sensor and processing algorithm (e.g., RGB-

*A. K. Singh and M. Adjel contributied equally.

[1]LIRMM, Université de Montpellier, CNRS, Montpellier, France firstname.secondname@lirmm.fr

[2]LISSI, Université de Paris-Est Créteil, Créteil, France adjelmohamed@gmail.com

[3]LAAS-CNRS, Université de Toulouse, CNRS, Toulouse, France vincent.bonnet@laas.fr

D image stream processed by openpose[1]),
- it is invariant to affine transformations (scaling, translations, rotations),
- it generalizes well across the anthropometry of different subjects,
- it is less sensitive to outliers, which can be processed via standard joint angle limits (from biomechanical tables).

These advantages are crucial in the perspective of machine learning, since a joint-based classifier will require a drastically smaller amount of data than the state-of-art Cartesian-based classifiers, to achieve comparable generality. In a nutshell, the contributions of this paper are:

- on AnDY dataset, we show that the joint angle representation yields higher accuracy than the Cartesian representation and than the state of art classifier [3];
- because of the advantages cited just above, our joint angles classifier is accurate across all AnDY actions, effective with limited data and person independent;
- because of the lower dimension of the feature vector, training is much faster with joint angles than it is with Cartesian coordinates;
- the classifier computation time is compatible with real-time constraints (even considering the addition of a vision-based skeleton tracker).

Last but not least, our work is an original synergy of two schools of thought which tend to compete in the field of engineering, those of the model-based and data-based communities. Indeed, by complementing biomechanics (Sec. II) with deep learning (Sec. III), we outperform the state of art [3] in the recognition of industrial actions (Sec. IV).

II. The Pipeline for Joint Angle Estimation

A motion capture system usually measures Cartesian poses of the body segments. To estimate joint angles from these measures, we developed a *Capture System Software Library (CSSL)*, composed of an offline and an online phase. The offline phase consists in creating a biomechanical model of each subject based on a Unified Robot Description Format - URDF. At each new sample of time, the online phase updates the joint angles of the model by calculating inverse kinematics. Both phases are detailled below.

A. Biomechanical Model of the Human

The proposed 23 Degrees of freedom (DoF) biomechanical model is based on the recommendations of the International Society of Biomechanics [25]–[27]. This way, we can cope with the differences between scientific and clinical communities, and use – in future work – the model in the field of industrial ergonomics.

Figure 1 shows the model with the position of the Joint Center Positions (JCP) and the type of corresponding mechanical joints. The model base is attached to the pelvis segment. The pelvis is connected to each leg by a 3 DoF ball hip joint and the knee is represented by a hinge joint.

[1]`https://github.com/CMU-Perceptual-Computing-Lab/openpose`

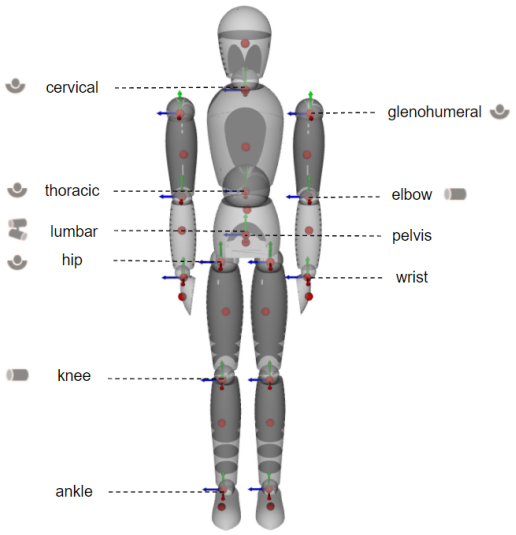

Fig. 1: Representation of the 23 DoF human biomechanical model and relevant JCP. Estimated centers of masses are represented with red spheres.

The cardan lumbar joint connects the pelvis to the abdomen that is then connected to the thorax by the 3 DoF ball thoracic joint. Each clavicle is linked to the thorax by a hinge joint. The arms are composed of a 3 DoF ball glenohumeral and by a hinge elbow joint. Estimating these 23 joint angles with the proposed model requires the measures of 16 JCP: cervical, left/right glenohumeral, left/right elbow, left/right wrist, thoracic, lumbar, pelvis, left/right hip, left/right knee and left/right ankle.

The offline phase takes as input constant parameters the subject's gender, height, and mass and it estimates – using anthropometric tables [28] – each segment's geometric and inertial parameters. With these, we generate the human model, as a Unified Robot Description Format – URDF. We do this with a C++ library (`https://gite.lirmm.fr/humar/humar-joints-estimator`) with classes representing each part of the biomechanical model (segments, joints, limbs) and its parameters (mass, length, inertia and center of mass)[2].

B. Real-time Inverse Kinematics Estimation

Once the URDF has been generated, the pipeline's online phase uses at each iteration t the JCP, to estimate the corresponding joint angles of the subject. More formally, our inverse kinematics problem consists, at each iteration t, in estimating the joint angle vector $\boldsymbol{\theta}_t \in \mathbb{R}^{23}$, corresponding to the measured 3D Cartesian coordinates of the 16 JCP, $\hat{\boldsymbol{\chi}}_t \in \mathbb{R}^{48}$. The Forward Kinematics Model (FKM), calculated from the human URDF, maps one representation to the other:

$$\boldsymbol{\chi}_t = FKM\left(\boldsymbol{\theta}_t\right). \tag{1}$$

[2]Although in this work we focus on kinematics, the body segment inertial parameters encode information about the dynamics, which could be exploited in future work.

We can fit this model on the measures, by solving a constrained optimization problem, which consists in minimizing the distance between the measured and modeled JCP, while accounting for a regularization term:

$$\begin{aligned} \boldsymbol{\theta}_t^* = & \underset{\boldsymbol{\theta}_t \in \mathbb{R}^{23}}{\text{argmin}} \quad ||FKM(\boldsymbol{\theta}_t) - \hat{\boldsymbol{\chi}}_t||_2^2 + \beta ||\boldsymbol{\theta}_t - \boldsymbol{\theta}_{t-1}^*||_2^2 \\ & \text{subject to} \quad \boldsymbol{\theta}^- \leq \boldsymbol{\theta}_t \leq \boldsymbol{\theta}^+ \end{aligned} \tag{2}$$

where $\beta = 1^{-3}$ is a weight allowing to avoid discontinuities, and $\boldsymbol{\theta}^+$ and $\boldsymbol{\theta}^-$ are the upper and lower joint limits, respectively. We solve (2) efficiently using the C++ library Ipopt [29] along with CppAD library [30].

III. ACTION RECOGNITION

A. Dataset Description

To validate our method, we used AnDY dataset [13], [31]. Six industry-specific tasks (*Screw High*, *Screw Middle*, *Screw Low*, *Untie Knot*, *Carry a 5kg load* and *Carry a 10kg load*) are combined in 6 alternative sequences. Thirteen healthy adults (9 males and 4 females, of height 175.4 ± 7.9cm, weight 72.3 ± 14.4kg and age 25.7 ± 5.0years) participated in the study. Each participant repeated 5 times 3 sequences selected randomly among the 6. The dataset contains three types of human motion measurements: inertial motion capture data (Xsens MVN Link system, 240Hz, Xsens, Enschede, The Netherlands), stereophotogrammetric data obtained with a motion capture system (Qualisys, 120Hz, Goteborg, Sweden), and videos recorded by two RGB cameras [3]. Within the $13 \times 5 \times 3 = 195$ sequences of the dataset, the authors have labeled 8 actions: *Reach*, *Pick*, *Place*, *Release*, *Carry*, *Fine Manipulation*, *Screw*, *Idle*, defined in Table I.

Figure 2 shows the concatenation of actions over time, for one sequence of 107 seconds. Actions are color-coded and represented by circular sectors of radius proportional to their duration (indicated in seconds). For instance, the largest orange sector on the right indicates that the participant was idle at the end of the sequence (from 80.7 s to 107 s). We can see that some actions (*Id*, *Fm*, *Re* and *Rl*) are more frequent and on average longer than others (*Ca*, *Pl*, *Pi* and *Sc*). This is the case for all sequences, making the dataset distribution skewed, i.e., some classes have more samples than others.

Action	Definition
Reach (Re)	Move an arm towards a target, no object in hand.
Pick (Pi)	Pick up an object: starts when touching the object, ends when the arm stops moving with respect to the body.
Place (Pl)	Place an object: similar to *Re*, but with an object in hand.
Release (Rl)	Bring arm back after manipulation.
Carry (Ca)	Carry an object: starts at the end of *Pi*, ends at the beginning of *Pl*.
Fine manipulation (Fm)	Manipulate an object with dexterity.
Screw (Sc)	Rotate the hand with screwing motion (particular case of *Fm*).
Idle (Id)	Don't move the hands.

TABLE I: Definition of the actions from the AnDY dataset.

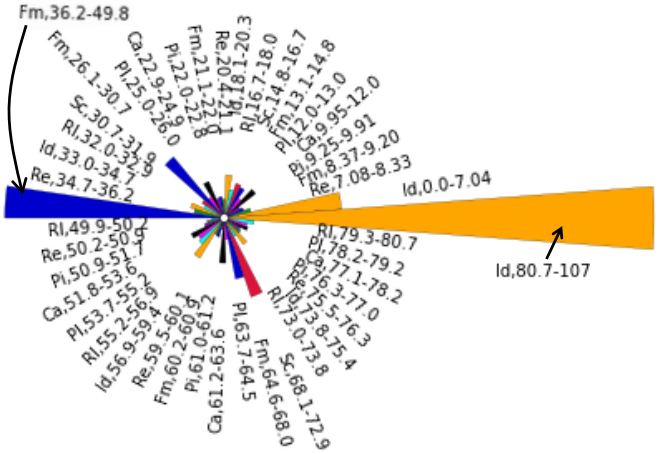

Fig. 2: Concatenation of actions for one sequence in AnDY (Participant_541_Setup_A_Seq_3_Trial_1, see [31]), starting from the mid left (*Id 0.0-7.04*). Each action is represented by a circular sector of radius proportional to its duration and labeled with the start and end time (in seconds). The color code is: orange (*Idle*), green (*Reach*), blue (*Fine Manipulation*), purple (*Pick*), black (*Carry*), magenta (*Place*), red (*Screw*), cyan (*Release*).

B. From inertial suit measures to joint angles

This section describes how we utilized the AnDY dataset to build the training and testing subsets for our human action classifier. To this end, we exploited the data recorded with the inertial suit at 240Hz. At each iteration t, the inertial suit estimates the Cartesian Coordinates of 23 joints of the human body, denoted $\boldsymbol{\psi}_t \in \mathbb{R}^{69}$ and listed in Table II. However, to calculate the corresponding joint angles with the CSSL, we require the 16 JCP indicated in Sec. II-B. To this end, we map $\boldsymbol{\psi}_t$ to $\boldsymbol{\chi}_t$, as shown in Table II. The head motions were not considered in our study, since they are prone to variations reflecting the subject's cognitive state. The so-called shoulder joint provided by the inertial suit corresponds to the center of the sterno-clavicular joints, and can be estimated only using regression methods. Therefore, we only use the cervical joint as center of rotation of the clavicules, as recommended in biomechanics [27]. We do not consider the positions of the toes, and thus the ankle joint angles, and the Cartesian positions of vertebrae T12 and T3, since they displayed small amplitude in the investigated tasks and are therefore error prone. At each sample time, we set the model base at the pelvis, with the transformation matrix calculated from the upper legs, pelvis and L5 Joint Coordinates, measured by the inertial suit [32]. Figure 3 shows the results of the CSSL for four representative postures of the investigated tasks.

C. Architecture of the Action Classifier

In this section, we describe the architecture of the classifier, which we have designed to map a sequence $\boldsymbol{X}$ of T consecutive feature vectors:

$$\begin{aligned} \boldsymbol{X}_t &= \{\boldsymbol{x}_{t-T}, \ldots, \boldsymbol{x}_{t-1}\} \\ \boldsymbol{x}_k &= \{\boldsymbol{\psi}_k, \boldsymbol{\theta}_k\} \in \mathbb{R}^n \end{aligned} \tag{3}$$

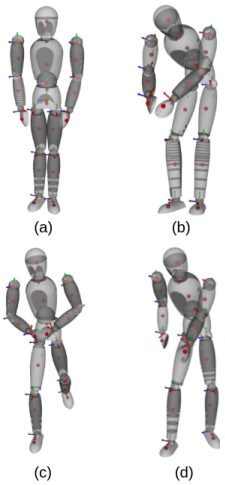

Fig. 3: Representative postures showing the results of the CSSL while: (a) standing in a neutral posture *(Idle)*, (b) bending to pick-up an object with both hands *(Pick)*, (c) walking while carrying an object with both hands *(Carry)*, (d) bending to place an object with one hand *(Place)*.

to one of the 8 action classes in Table I. The size n of feature vector can be either 69 – when using all the JCP provided by the inertial suit, $\boldsymbol{\psi}$ – or 23 – when using the joint angles $\boldsymbol{\theta}$ estimated using the CSSL. At each iteration k, our classifier assigns an action label to $\boldsymbol{x}_k$; after T iterations, the classifier labels the sequence $\boldsymbol{X}_t$ with the class which has obtained the majority of labels.

Our approach is inspired by [17], [19]. The authors used stacked LSTM for action recognition, but in this work, we tried [34], a bidirectional variant of LSTM (Bi-LSTM). Bi-LSTM adds one more LSTM layer, which reverses the direction of information flow. With the help of this additional input layer the input flows in backward direction. Further the the output of both the layers are combined to better understand the input sequences. This way the network considers both forward and backward time directions for classification. We believe Bi-LSTM is particularly useful when dealing with action sequences, since the classifier can refer to future actions to label the current one. Figure 4 illustrates our Bi-LSTM architecture. We stacked three Bi-LSTM layers, to make the network preserve the spatial relation between joints. Each of the 3 hidden layers is composed of 32 cells ($h = 32$). Before training and validation, input $\boldsymbol{x}_t$ is normalized (shifting it by its mean, and then dividing by its standard deviation). Then, the first layer applies to this normalized input. The output of the LSTM layer is then passed to the following layers. To avoid overfitting, we apply a dropout of 0.5 to the third layer. The stacked layers are followed by the final fully connected layer, with softmax activation to map $\boldsymbol{x}_t$ to one of the 8 classes. We used categorical cross entropy as the loss function, Adam as optimizer, and we set the learning rate to 0.001.

Inertial suit joints ψ [33]	**CSSL Joints Center Positions χ**
Head	N/A
T8	Thoracic
T12	N/A
Left upper arm	Left glenohumeral
Right upper arm	Right glenohumeral
Left shoulder	N/A
Right shoulder	N/A
Left lower arm	Left elbow
Right lower arm	Right elbow
Left hand	Left wrist
Right hand	Right wrist
Neck	Cervical
L5	Lumbar
T3	N/A
Pelvis	Pelvis
Left upper leg	Left hip
Right upper leg	Right hip
Left lower leg	Left knee
Right lower leg	Right knee
Left foot	Left ankle
Right foot	Right ankle
Left toe	N/A
Right toe	N/A

TABLE II: Mapping from the 23 joints (ψ) measured by the inertial suit to the 16 JCP (χ) needed by the CSSL for inverse kinematics.

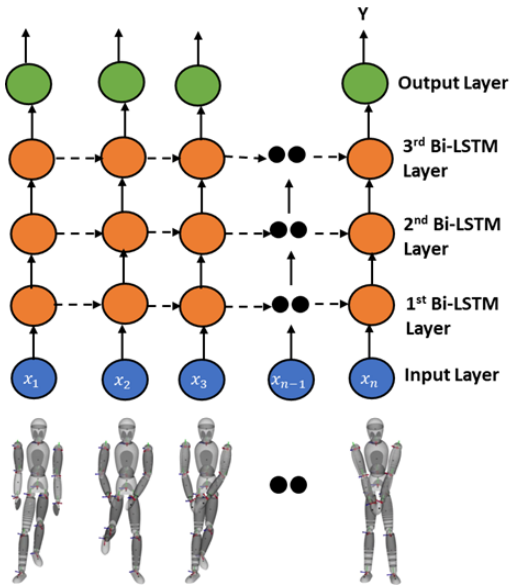

Fig. 4: Architecture of our action classifier. The input (blue) is n-dimensional vector $\boldsymbol{x} = \{\boldsymbol{\psi}, \boldsymbol{\theta}\}$, with $n = 23$ or 69. This is normalized, passed through 3 Bi-LSTM layers (orange), followed by a fully connected layer (green) with softmax activation to obtain the action class label (Y).

IV. VALIDATING THE ACTION CLASSIFIER

In this Section, we present the tests that we ran to validate the classifier, developed with TensorFlow 2.7, on a computer with Intel© Core i7-1165G7@2.80GHz, 16 GB RAM and 4 GB GPU. The python code implementing the classifier can be found at `https://gite.lirmm.fr/humar/research-projects/humar_dnn`. A video presenting the results is attached to this paper and available at `https://youtu.be/2GNWYsOfDYw`.

As in [3], we use a time window of 1 second with 50% overlapping, to define each sample of an action. Since data is measured at 240 Hz, this corresponds to $T = 240$ in (3). At this stage, some classes have many more samples than others, due to their frequency and execution length (Sec. III-A). Hence, to balance the data, we randomly discard some samples from the most represented classes (*Id*, *Fm*, *Re*, *Rl*).

In a first experiment, we randomly split the dataset into 70% training and 30% validation subsets, and validate it with the action classifier presented in Sec. III-C. We use the same architecture and hyperparameters for both representations of the input data, $\boldsymbol{\psi} \in \mathbb{R}^{69}$ and $\boldsymbol{\theta} \in \mathbb{R}^{23}$. Table III compares the F1 scores of our classifier with the two inputs $\boldsymbol{\psi}$ and $\boldsymbol{\theta}$, with the state of art Hidden Markov Model from [3]. Both the figure and table show that the joint angle representation ($\boldsymbol{\theta}$) slightly prevails over the Cartesian one ($\boldsymbol{\psi}$). Despite the small difference in accuracy, it should be noted that training is much faster with $\boldsymbol{\theta}$ than it is with $\boldsymbol{\psi}$. On average, a training epoch takes 130.13 s with $\boldsymbol{\theta}$ and 230.38 s with $\boldsymbol{\psi}$. This is because of their different dimensions (23 vs 69). In terms of accuracy, both $\boldsymbol{\theta}$ and $\boldsymbol{\psi}$ outperform [3], where the F1 score was 86.63% (see Table III). We believe that the reason is that Recurrent Neural Networks can handle long term data dependencies better than Hidden Markov Models, where the state at time t is related only to the state at $t-1$. Another advantage of our approach over [3], is that it requires solely the joint positions, which can be measured with a variety of human motion capture devices. Instead, classifier [3] needs 11 features, including higher-order kinematic variables (e.g., angular velocities and accelerations), which cannot be measured by most devices (e.g., depth cameras).

Input	n	**F1 Score**
23 JCP measured by inertial suit, $\boldsymbol{\psi}$.	69	98.99%
Joint Angles derived via inverse kinematcis, $\boldsymbol{\theta}$.	23	99.42%
Various kinematic features [3].	11	83.36%

TABLE III: F1 scores and input dimensions (n) of: our Bi-LSTM with input $\boldsymbol{\psi}$, our Bi-LSTM with input $\boldsymbol{\theta}$ and the state of art Hidden Markov Model [3].

Figure 5 is the confusion matrix of our Bi-LSTM with joint angles $\boldsymbol{\theta}$ as input. For each action the values on the diagonal are the percentage of true positives, while the others are the percentage of misclassifications (false negatives or false positives). The very high values on the diagonal confirm the quality of our classifier across all eight actions.

To verify the performance across different subjects, we tried leave-one-out cross validation for joint angles input $\boldsymbol{\theta}$. We excluded all 15 trials of one participant from the training set, leaving data of the other participants. Then, we validated only with the excluded participant. We show the results for each action in Fig. 6, with the excluded participants in abscissa, and the F1 score in ordinate. For example, the first column indicates the F1 score of participant $p5124$ with the classifier trained on the rest of the population. Validation accuracy is similar (and high) across all actions for all participants, except $p3327$ (column 9). The reason

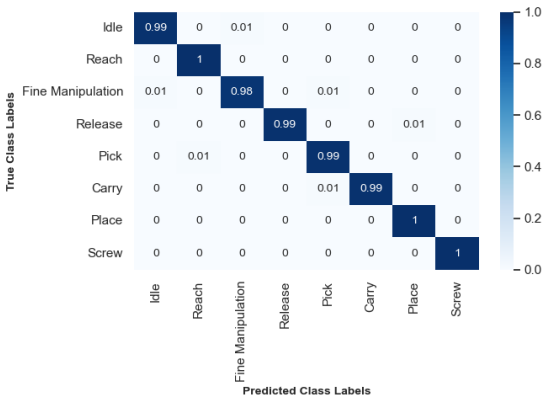

Fig. 5: Confusion Matrix of our Bi-LSTM with input $\boldsymbol{\theta}$.

could be that participant $p3327$ is left handed, while most participants (11 out of 13) are right handed. Ideally, the dataset should be equally distributed among left and right handed participants. Table IV shows the F1 scores when validating with all participants, the classifier was trained with and without participant $p3327$. For all actions, the F1 score is greater than 89.8%, and when excluding $p3327$ its decreases by 2.0% (average across the 8 actions). These results show the robustness of our classifier, with regards to new data.

Training set	***Id***	***Re***	***Fm***	***Pi***	***Ca***	***Pl***	***Sc***	***Rl***
With *p3327*	98.3	91.6	98.0	96.9	91.5	95.9	96.2	94.9
Without *p3327*	97.5	90.6	97.3	92.7	89.8	94.0	92.7	92.6

TABLE IV: F1 scores (in %) of the $\boldsymbol{\theta}$ classifier across all participants, trained with/without participant *p3327*.

In a more challenging experiment, we excluded almost half (6/13) of the population and we trained the classifier with the rest. We excluded all samples of 4 male participants ($p909$,

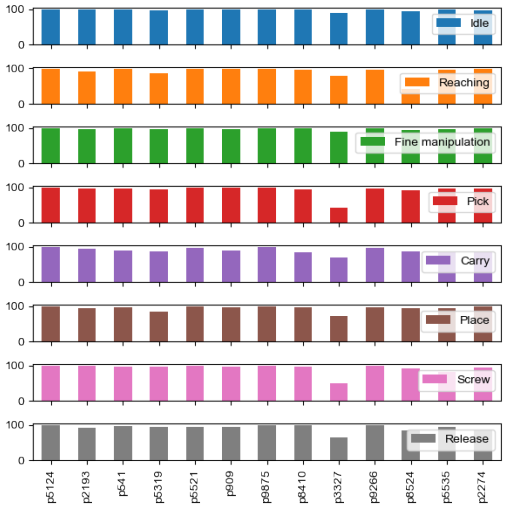

Fig. 6: Leave-one-out F1 score of our Bi-LSTM with input $\boldsymbol{\theta}$. Abscissa: excluded participants, ordinate: F1 score with that participant.

$p3327$, $p5124$ and $p9266$) and 2 female participants ($p8524$ and $p5535$) from the training, while we used only these 6 participants for validation. We repeated the experiment with both inputs: raw inertial suit JCP $\boldsymbol{\psi}$ and joint angles $\boldsymbol{\theta}$. We report the F1 scores for all eight actions in Table V. Obviously, with less examples for training, all values have decreased with regards to Tables III. Yet, there is a clear gain in using joint angles $\boldsymbol{\theta}$ compared to $\boldsymbol{\psi}$. The average F1 scores were 85.48% and 68.19% for $\boldsymbol{\theta}$ and for $\boldsymbol{\psi}$, respectively. This shows that joint angles classification is robust to the use of limited data, and that it is subject independent. It also means that when using a biomechanical model and calculating joint angles, we make the data agnostic to the subject's biometrics.

Input	***Id***	***Re***	***Fm***	***Pi***	***Ca***	***Pl***	***Sc***	***Rl***
Inertial suit JCP, $\boldsymbol{\psi}$	76.2	34.8	70.7	18.8	82.0	58.0	67.2	47.8
Joint angles, $\boldsymbol{\theta}$	90.3	68.3	88.4	65.6	85.3	74.5	78.9	70.1

TABLE V: F1 scores (in %) of the classifier, trained with 7 of the 13 participants, with inputs $\boldsymbol{\psi}$ and $\boldsymbol{\theta}$.

Performances	**$\boldsymbol{\psi}$@240Hz**	**$\boldsymbol{\theta}$@240Hz**	**$\boldsymbol{\psi}$@10Hz**	**$\boldsymbol{\theta}$@10Hz**
F1 score	98.99	99.42	95.44	96.38
Classification time	1.70	1.44	0.18	0.17
IKE time	N.A.	5.25	N.A.	5.25

TABLE VI: F1 scores (in %) and average times (ms) needed for Classification and Inverse Kinematics Estimation (IKE), with four versions of the classifier.

Let us conclude on the possibility of using our classifier for real time robot control in the context of human-robot interaction. This paper introduces the foundations of our future work, which aims at fast action recognition with low cost sensors embedded on a – possibly moving – robot. Ultimately, we believe these will be RGB-D cameras with the inclusion, in the perception pipeline, of a skeleton tracking algorithm such as Openpose. Such algorithms nowadays require long computing time (e.g. 60 to 80ms from our experience) even with the GPU embedded on our robot (NVIDIA GeForce GTX 1080). This will substantially reduce the data frame rate, from the current 240Hz, to approximately 10Hz. To verify the feasibility at such framerate, we subsampled the inertial suit data from the original 240Hz to the targeted 10Hz. This corresponds to taking a measure every 24 and setting $T = 10$ in (3). Under such conditions, we train and validate the classifier with both $\boldsymbol{\psi}$ and $\boldsymbol{\theta}$, to obtain the F1 scores shown in Table VI, along with the ones at 240Hz (copied from Table III). The results are very encouraging, with a loss of only 3% when subsampling the joint angles $\boldsymbol{\theta}$. Along with the accuracy, one should verify the feasibility in terms of computation time. Fulfilling the 10Hz constraint in the worse case scenario – camera acquisition and skeleton tracking taking 80ms – leaves 20ms for the other operations. These are: classification (i.e., the time required by the Bi-LSTM to label a sequence) and (when using joint angles $\boldsymbol{\theta}$) inverse kinematics estimation, described in Sec. II-B. Table VI shows the computation times of these two operations for the four versions of the classifier. These are averages measured on the mentioned computer, for one subject. The results show that even with the addition of inverse kinematics, the pipeline should not violate the 100ms constraint.

V. CONCLUSIONS

We have introduced and validated a framework for recognizing industrial actions from human motion data. We argue that relying on joint angles rather than on the Cartesian coordinates – commonly used in the literature – enhances the classification performance at many levels. The classifier is less sensitive to subject variance, faster to train, and compatible with robotic real-time constraints. Our claim is confirmed by the results obtained on inertial suit measures from AnDy dataset, to classify 8 different actions.

In the future, we will apply the classifier to data from low-cost depth cameras, combined with a visual skeleton tracker (e.g., openpose). Currently, to generate the biomechanic model of each subject, we must know his/her gender, height and weight. We will try to avoid this, by relying solely on sensor data. We also believe that the joint angles representation will help address challenges present in vision systems, such as occlusions and undetected joints.

ACKNOWLEDGMENT

This research was carried out in the context of the SOPHIA project, which received funding from the EU Horizon 2020 research and innovation program under Grant Agreement No. 871237. We would like to thank the LARSEN team at INRIA for providing the AnDy dataset and Raphael Dumas from IFFSTAR-University of Lyon for his help on biomechanical modeling.

REFERENCES

[1] M. Hvilshøj, S. Bøgh, O. Rosenlund, and O. Madsen, "Autonomous industrial mobile manipulation (aimm): Past, present and future," *Industrial Robot*, pp. 120–135, 2012.

[2] E. Schneider, *OSH in figures: Occupational safety and health in the transport sector — an overview*. 2012.

[3] A. Malaisé, P. Maurice, F. Colas, and S. Ivaldi, "Activity recognition for ergonomics assessment of industrial tasks with automatic feature selection," *IEEE Robotics and Automation Letters*, pp. 1132–1139, 2019.

[4] I. Rodríguez-Moreno, J. M. Martinez-Otzeta, B. Sierra, I. Rodriguez Rodriguez, and E. Jauregi Iztueta, "Video activity recognition: State-of-the-art," *Sensors*, p. 3160, 2019.

[5] J. Zhang, W. Li, P. O. Ogunbona, P. Wang, and C. Tang, "Rgb-d-based action recognition datasets: A survey," *Pattern Recognition*, pp. 86–105, 2016.

[6] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, "Large-scale video classification with convolutional neural networks," *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[7] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatiotemporal features with 3d convolutional networks," *Computing Research Repository*, pp. 4489–4497, 2015.

[8] Y. Du, Y. Fu, and L. Wang, "Skeleton based action recognition with convolutional neural network," *IAPR Asian Conf. on Pattern Recognition (ACPR)*, pp. 579–583, 2015.

[9] J. Liu, A. Shahroudy, M. Perez, G. Wang, L.-Y. Duan, and A. C. Kot, "NTU RGB+D 120: A large-scale benchmark for 3d human activity understanding," *Computing Research Repository*, 2019.

[10] M. Barnachon, S. Bouakaz, B. Boufama, and E. Guillou, "Ongoing human action recognition with motion capture," *Pattern Recognition*, pp. 238–247, 2014.

[11] C. Ionescu, D. Papava, V. Olaru, and C. Sminchisescu, "Human3.6m: Large scale datasets and predictive methods for 3d human sensing in natural environments," *IEEE Tran. on Pattern Analysis and Machine Intelligence*, pp. 1325–1339, 2014.

[12] M. Dallel, V. Havard, D. Baudry, and X. Savatier, "Inhard - industrial human action recognition dataset in the context of industrial collaborative robotics," *IEEE Int. Conf. on Human-Machine Systems (ICHMS)*, pp. 1–6, 2020.

[13] P. Maurice, A. Malaisé, C. Amiot, N. Paris, G.-J. Richard, O. Rochel, and S. Ivaldi, "Human movement and ergonomics: An industry-oriented dataset for collaborative robotics," *The International Journal of Robotics Research*, pp. 1529–1537, 2019.

[14] X. Wang, A. Farhadi, and A. Gupta, "Actions ˜ transformations," *Computing Research Repository*, 2015.

[15] H. Morgan and F. Milliken, "Keys to action: Understanding differences in organizations' responsiveness to work-and-family issues," *Human Resource Management*, pp. 227–248, 2006.

[16] J. Liu, N. Akhtar, and A. Mian, "Skepxels: Spatio-temporal image representation of human skeleton joints for action recognition," *Computing Research Repository*, 2017.

[17] S. Zhang, X. Liu, and J. Xiao, "On geometric features for skeleton-based action recognition using multilayer lstm networks," *IEEE Winter Conf. on Applications of Computer Vision (WACV)*, pp. 148–157, 2017.

[18] J. Liu, A. Shahroudy, D. Xu, and G. Wang, "Spatio-temporal LSTM with trust gates for 3d human action recognition," *Computing Research Repository*, 2016.

[19] Y. Du, W. Wang, and L. Wang, "Hierarchical recurrent neural network for skeleton based action recognition," *2015 IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pp. 1110–1118, 2015.

[20] L. Shi, Y. Zhang, J. Cheng, and H. Lu, "Skeleton-based action recognition with directed graph neural networks," *IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, pp. 7904–7913, 2019.

[21] S. Yan, Y. Xiong, and D. Lin, "Spatial temporal graph convolutional networks for skeleton-based action recognition," *Computing Research Repository*, 2018.

[22] A. Ullah, J. Ahmad, K. Muhammad, M. Sajjad, and S. W. Baik, "Action recognition in video sequences using deep bi-directional lstm with cnn features," *IEEE Access*, pp. 1155–1166, 2018.

[23] J. Wang, Y. Chen, S. Hao, X. Peng, and L. Hu, "Deep learning for sensor-based activity recognition: A survey," *Computing Research Repository (CoRR)*, 2017.

[24] H. Yasin, M. Hussain, and A. Weber, "Keys for action: An efficient keyframe-based approach for 3d action recognition using a deep neural network," *Sensors*, 2020.

[25] G. Wu and P. R. Cavanagh, "Isb recommendations for standardization in the reporting of kinematic data," *Journal of Biomechanics*, pp. 1257–1261, 1995.

[26] G. Wu, S. Siegler, P. Allard, C. Kirtley, A. Leardini, D. Rosenbaum, M. Whittle, D. D. D'Lima, L. Cristofolini, H. Witte, O. Schmid, and I. Stokes, "Isb recommendation on definitions of joint coordinate system of various joints for the reporting of human joint motion—part i: Ankle, hip, and spine," *Journal of Biomechanics*, pp. 543–548, 2002.

[27] G. Wu, F. van der Helm, D. Veeger, M. Makhsous, P. Roy, C. Anglin, J. Nagels, A. Karduna, K. McQuade, X. Wang, F. Werner, and B. Buchholz, "Isb recommendation on definitions of joint coordinate systems of various joints for the reporting of human joint motion - part ii: Shoulder, elbow, wrist and hand," *Journal of biomechanics*, pp. 981–992, 2005.

[28] R. Dumas and J. Wojtusch, "Estimation of the Body Segment Inertial Parameters for the Rigid Body Biomechanical Models Used in Motion Analysis," *Handbook of Human Motion*, pp. 47–77, 2018.

[29] A. Wächter and L. Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Mathematical programming*, pp. 25–57, 2006.

[30] B. Bell and J. Burke, "Algorithmic differentiation of implicit functions and optimal values," *Lecture Notes in Computational Science and Engineering*, pp. 67–77, 2008.

[31] P. Maurice, A. Malaisé, S. Ivaldi, O. Rochel, C. Amiot, N. Paris, G.-J. Richard, and L. Fritzsche, "Andydata-lab-oneperson," *Zenodo*, 2019.

[32] A. Cappozzo, F. Catani, U. D. Croce, and A. Leardini, "Position and orientation in space of bones during movement: Anatomical frame definition and determination.," *Clinical biomechanics*, pp. 171–178, 1995.

[33] D. Roetenberg, H. Luinge, and P. Slycke, "Xsens mvn: Full 6dof human motion tracking using miniature inertial sensors," *Xsens Motion Technol. BV Tech. Rep.*, 2009.

[34] M. Schuster and K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Tran. on Signal Processing*, pp. 2673–2681, 1997.