Physical Inertial Poser (PIP): Physics-aware Real-time Human Motion Tracking from Sparse Inertial Sensors
- Supplementary Material -

Xinyu Yi[1] Yuxiao Zhou[1] Marc Habermann[2] Soshi Shimada[2]
Vladislav Golyanik[2] Christian Theobalt[2] Feng Xu[1]
[1]School of Software and BNRist, Tsinghua University [2]Max Planck Institute for Informatics, Saarland Informatics Campus

A. Implementation Details

Network Structure. We schematically visualize the network structures in our kinematics module in Fig. 7. The recurrent neural network (RNN) P_{L}, P_{A}, R_{A}, V_{A}, and C_{F} share the same structure. Each network includes a linear input layer with a ReLU activation, two Long Short-term Memory (LSTM) [92] layers with the width of 256, and a linear output layer. A 40% dropout is applied to prevent over-fitting. The RNN C_{F} is finally activated by a Sigmoid function to obtain probability values. The initial states of P_{L} and V_{A} are regressed from the starting leaf joint positions $\boldsymbol{p}_{\mathrm{leaf}}^{(0)}$ and joint velocities $\boldsymbol{v}^{(0)}$ using the fully-connected network (FCN) I_{PL} and I_{VA}, respectively. Each FCN consists of 3 fully-connected (FC) layers with the width of 256, 512, and 1024 using the ReLU activation. The output of the FCN is used to initialize the hidden/cell states of the two LSTM layers of the RNN at the beginning.
Rotation Representation. The inertia input vector $\boldsymbol{x}$ consists of accelerations and *rotation matrices*, which are obtained after the calibration. The output of R_{A} is the non-root joint rotations *w.r.t* the root parameterized in the *6D representation* [107]. Combining the estimated non-root joint rotations with the root orientation measured by the IMU placed on the pelvis, we obtain the vector $\boldsymbol{\varphi}$. The character pose in the physics module is described by local joint rotations (*i.e.,* each joint relative to its parent) in *Euler angles*, which is denoted as $\boldsymbol{\theta}$. The configuration vector $\boldsymbol{q} = [\boldsymbol{r}_{\mathrm{root}} \quad \boldsymbol{\theta}]$ is then composed of the root translation and the pose in Euler angles.
Datasets. Following [105], we use the AMASS [96] dataset and the train split of the DIP-IMU [93] dataset for the network training, and use the TotalCapture [102] dataset and the test split of the DIP-IMU dataset for evaluation. For AMASS, we synthesize the IMU measurements and foot-ground contact labels as proposed by Yi et al. [105], and synthesize the ground-truth joint velocities using:

$$\boldsymbol{v}^{\mathrm{GT}}(t) = (\boldsymbol{R}_{\mathrm{root}}^{\mathrm{GT}}(t))^{-1}(\boldsymbol{r}^{\mathrm{GT}}(t) - \boldsymbol{r}^{\mathrm{GT}}(t-1))/\Delta t, \quad (14)$$

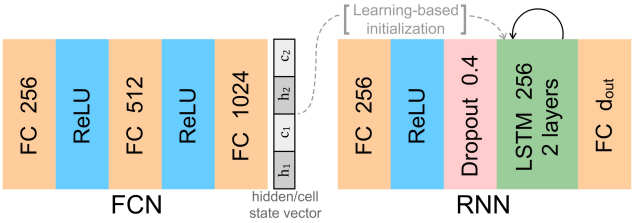

Figure 7. Detailed structures of the recurrent neural network (RNN) and the fully-connected network (FCN) in our kinematics module. "FC" represents a fully-connected layer. The output dimension and other hyper-parameters are marked in each block.

where $\boldsymbol{R}_{\mathrm{root}}^{\mathrm{GT}}(t) \in \mathbb{R}^{3\times3}$ is the ground-truth root orientation at frame t; $\boldsymbol{r}^{\mathrm{GT}} \in \mathbb{R}^{3J}$ is the ground-truth joint global positions; Δt is the frame interval. We also re-calibrate the acceleration measurements in TotalCapture, as we find that they are constantly biased (see Fig. 8). Specifically, to remove the bias, we synthesize the accelerations for TotalCapture using the method of Yi et al. [105] and align the mean acceleration measurement for each sequence to the mean synthetic values by adding or subtracting a constant.
Gain Parameters for PD Controllers. The gain parameters k_{p_θ}, k_{d_θ}, k_{p_r}, and k_{d_r} of the dual PD controller introduced in Sec. 3.2.2 are derived as follows. Take the joint rotation controller (controlling $\boldsymbol{\theta}$) as an example. As we use first-order approximations in the dynamic states updater (Sec. 3.2.4), we apply first-order Taylor expansion on $\boldsymbol{\theta}$ and $\dot{\boldsymbol{\theta}}$, and rearrange the equation, which writes:

$$\ddot{\boldsymbol{\theta}}(t) = \frac{1}{\Delta t^2}(\boldsymbol{\theta}(t+2\Delta t) - \boldsymbol{\theta}(t+\Delta t)) - \frac{1}{\Delta t}\dot{\boldsymbol{\theta}}(t), \quad (15)$$

where $\Delta t = 1/60$ is the time interval between frames. By associating this equation with Eq. 3 and Eq. 5 in the main paper, the proportional gain k_{p_θ} and k_{p_r} should be 3600, and the derivative gain k_{d_θ} and k_{d_r} should be 60. For the joint rotation controller, setting the proportional gain k_{p_θ} to a lower value gives smoother angular accelerations. Thus, we set k_{p_θ} to 2400 in our experiments.
Other Details. We use a laptop with an Intel(R) Core(TM)

Method		DIP-IMU							
		SIP Error	Ang Error	Pos Error	Mesh Error	Rel Jitter	Abs Jitter	ZMP Dist	Latency
Offline	DIP [93]	16.36	14.41	6.98	8.56	2.34	-	-	-
	TransPose [105]	13.97	7.62	4.90	5.83	0.13	0.85	0.59	-
Online	DIP [93]	17.10	15.16	7.33	8.96	3.01	-	-	117
	TransPose [105]	16.68	8.85	5.95	7.09	0.61	1.46	1.67	94
	PIP (Ours)	**15.02**	**8.73**	**5.04**	**5.95**	**0.23**	**0.24**	**0.12**	**16**

Method		TotalCapture							
		SIP Error	Ang Error	Pos Error	Mesh Error	Rel Jitter	Abs Jitter	ZMP Dist	Latency
Offline	DIP [93]	18.47	17.54	9.47	11.19	2.91	-	-	-
	TransPose [105]	14.71	12.19	5.44	6.22	0.16	0.91	0.76	-
Online	DIP [93]	18.62	17.22	9.42	11.22	3.62	-	-	117
	TransPose [105]	16.58	12.89	6.55	7.42	0.95	1.87	1.40	94
	PIP (Ours)	**12.93**	**12.04**	**5.61**	**6.51**	**0.20**	**0.20**	**0.23**	**16**

Table 3. Comparison with the state-of-the-art methods on more metrics. PIP outperforms previous online methods on all metrics with much less latency, while also achieves comparable capture accuracy but higher physical correctness when compared with previous offline methods. This demonstrates the superiority of our system which runs in real-time with extremely small latency.

i7-10750H CPU and an NVIDIA RTX2080 Super graphics card to run the experiments and the live demos. We use PyTorch 1.8.1 with CUDA 11.1 to implement our kinematics estimator, and leverage the Rigid Body Dynamics Library [90] to implement our physics-based optimizer. The live demo is implemented using Unity3D. We use Noitom Perception Neuron series [95] IMU sensors in our demo. Both training and evaluation assume 60 fps sensor input. The training data is additionally clipped into short sequences in 200-frame lengths for more effective learning. Specifically, we separately train each RNN in the kinematics module using the synthetic AMASS [96] dataset with a batch size of 256 using the Adam [94] optimizer, and finetune P_{L} (together with I_{PL}), P_{A}, and R_{A} on the train split of the DIP-IMU dataset, following [105]. We do not train V_{A} and C_{F} on DIP-IMU as it does not contain global movements.

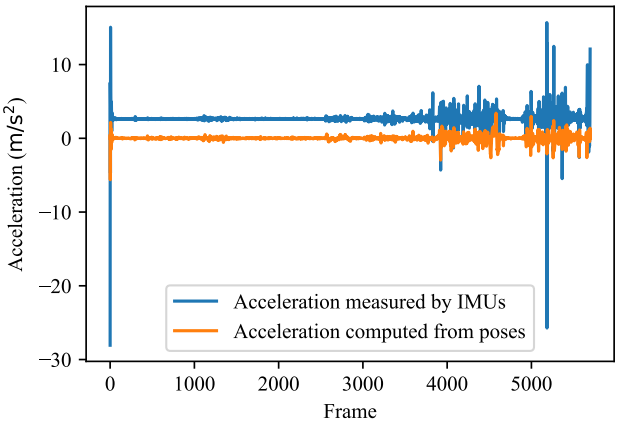

Figure 8. The acceleration measurements in the TotalCapture [102] dataset is constantly biased. We visualize the accelerations (x-axis component) measured by IMUs in blue and the one computed from the subject motions based on Vicon [103] by a finite-difference method in orange. We can see an obvious constant bias in the IMU acceleration measurements (blue) based on the fact that real accelerations should be approximately zero-centered.

B. Comparisons on More Metrics

In this section, we show the comparison results with the previous state-of-the-art methods [93, 105] on more metrics. In addition to the metrics used in the main paper, we also evaluate *1) Angular Error*: the mean rotation error of all body joints in the global space in degrees; *2) Positional Error*: the mean position error of all body joints in the global space with the root position and orientation aligned in cm; *3) Relative Jitter*: the jitter calculated in the local (root-relative) frame in $\mathrm{km/s^3}$, where the root translation is not considered. Notice that due to the length limit of the main paper, we only showed the mesh error as it incorporates both angular and positional error, and the SIP error as it is directly related to motion ambiguities in the main text. Here, we report the results on more metrics for a fair comparison. We also evaluate previous offline methods for references, which need to pre-record the inertia measurements during the whole motion and estimate the motion with the help of the complete inertia sequence. The results on TotalCapture [102] and the test split of DIP-IMU [93] dataset are shown in Tab. 3. We outperform previous online methods on all metrics with largely reduced latency, which demonstrates the accuracy and effectiveness of our approach. Moreover, compared with the offline methods, PIP achieves comparable motion accuracy (reflected in the first 5 metrics) but higher physical plausibility (reflected in Absolute Jitter and ZMP Distance). We attribute this to the physics-based motion optimizer proposed in the main paper. Most importantly, our system runs *in real-time*, while

the offline approaches require the access to the complete inertia sequence. Thus, our approach significantly closes the gap between online and offline methods, and enables a wide variety of real-time applications such as gaming.

C. Discussions and Future Works

Quantitative Evaluations of Physics. A direct quantitative evaluation of physics (*e.g.,* joint torques and ground reaction forces) would be advantageous. However, to the best of our knowledge, there is no public dataset containing both IMU measurements and ground-truth forces (either joint torques or ground reaction forces). We believe that creating such a dataset requires research on its own, and would have great value for the community. For now, we can only provide qualitative visualization of torques/GRFs in our supplemental video and Fig. 5, which is intuitively plausible and in line with the references [99, 106]. Besides, as the output motion is *entirely* driven by the estimated forces, the quantitative evaluation of the motion can also implicitly demonstrate the quality of our force estimation. Furthermore, we use jitter (jerk) and ZMP distance as indirect quantitative evaluations of the physics estimation, which reflect the naturalness [91] and equilibrium [104] of the motion, respectively. Since we do not adopt any explicit penalty on these two metrics, nor do we use any temporal filter or balancing technique on the motion, the better results on these two metrics actually suggest the improved physical correctness achieved by our motion optimizer.

Regarding the ground contact evaluation, previous works [100, 101] use mean penetration error to evaluate the non-physical foot penetration. As we explicitly model the contacts as hard constraints, both sliding and ground penetration are *strictly* avoided with any contacting part of the body. Thus, these errors would be zero.

Zero Moment Point vs. Center of Pressure. Previous works [97, 98] use Center of Pressure (CoP) accuracy to quantify the force estimation, which is related to our Zero Moment Point (ZMP) distance. Here we point out the difference between these two notations and the reason why we choose to use ZMP distance. The pressure between the human body and the ground can be represented by a force exerted at the CoP. If such a force can balance all active forces acting on the human body during the motion, the human body is in dynamic equilibrium, and ZMP coincides with CoP (*i.e.,* within the support polygon). However, when the force acting on the CoP cannot balance other forces, the human will fall down about the foot edge, and the ZMP (more precisely, the fictitious ZMP) will be outside the support polygon, whose distance to the polygon is proportional to the intensity of the unbalanced force. In such cases, CoP is on the border of the support polygon as the ground reaction forces cannot escape the polygon. Thus, the reason to use ZMP distance in our physics evaluation becomes clear:

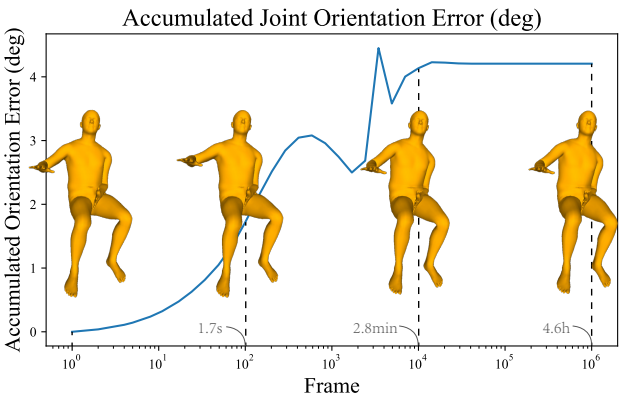

Figure 9. Pose drifts in a perfectly-still sitting pose. We evaluate PIP on 4.6-hour artificial inertia measurements with zero accelerations and fixed orientations of a sitting pose. We plot the accumulated orientation error of all body joints over time and pick four frames for visualization. Our system stably estimates a sitting pose during the entire sequence with a total drift of 4.2 degrees.

since the estimated motion cannot be perfectly physically correct and contains unbalanced movements, the ZMP distance can better reflect the disequilibrium in the captured motion. On the other hand, evaluating CoP accuracy needs a more sophisticated modeling of human feet (rather than a simplified square facet contact) and ground-truth pressure annotations, which we leave as a future work. For more detailed introductions of ZMP, readers are referred to [104].

Drifts in Long-term Tracking. As a purely inertial sensor based approach, PIP inevitably suffers from drifts in long-term tracking. As measured in Fig. 3, the translation drift of our system depends on how far the subject moves, and is about 4.6% in our experiments. Regarding the subject's pose, we do not see an evident drift in our experiments. This may be because the subject is always moving, and the orientation and acceleration measurements effectively confine the possible human pose. Therefore, it is interesting to examine the *pose drift in still poses*, especially for the ambiguous ones like sitting. However, as the IMUs always have small noises and humans cannot keep perfectly still for a long time, it is difficult to quantify the pose drifts in real settings. Thus, we conduct a toy experiment where we artificially set all acceleration measurements to zero and orientations unchanged at the point after the sit-down motion in Fig. 6, *i.e.,* to simulate a perfectly-still sitting pose. As shown in Fig. 9, our system can keep estimating sitting poses stably with a total drift of 4.2 degrees for all body joints at 1 million (4.6 hours) frames. This demonstrates the robustness of our system in long-term tracking, which is ensured by the RNNs and the learning-based RNN initialization scheme. We also conduct a live experiment where our method can track long-period sitting for half an hour stably and is not getting worse as time goes by. Please refer to our supplementary video for more results.

References

[90] Martin Felis. Rbdl: an efficient rigid-body dynamics library using recursive algorithms. *Autonomous Robots*, 41, 02 2017. 2

[91] Tamar Flash and Neville Hogan. The coordination of arm movements: An experimentally confirmed mathematical model. *The Journal of neuroscience : the official journal of the Society for Neuroscience*, 5, 08 1985. 3

[92] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9, 12 1997. 1

[93] Yinghao Huang, Manuel Kaufmann, Emre Aksan, Michael J. Black, Otmar Hilliges, and Gerard Pons-Moll. Deep inertial poser learning to reconstruct human pose from sparseinertial measurements in real time. *ACM Transactions on Graphics, (Proc. SIGGRAPH Asia)*, 37, nov 2018. 1, 2

[94] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *International Conference on Learning Representations*, 12 2014. 2

[95] Noitom Ltd. Perception neuron series. Website. https://www.noitom.com/. 2

[96] Naureen Mahmood, Nima Ghorbani, Nikolaus F. Troje, Gerard Pons-Moll, and Michael J. Black. Amass: Archive of motion capture as surface shapes. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2019. 1, 2

[97] Bharadwaj Ravichandran. Biopose-3d and pressnet-kl: A path to understanding human pose stability from video. 2020. 3

[98] Jesse Scott, Bharadwaj Ravichandran, Christopher Funk, Robert T. Collins, and Yanxi Liu. From image to stability: Learning dynamics from human pose. In *Computer Vision – ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXIII*, 2020. 3

[99] Erfan Shahabpoor and Aleksandar Pavic. Measurement of walking ground reactions in real-life environments: A systematic review of techniques and technologies. *Sensors*, 17(9), 2017. 3

[100] Soshi Shimada, Vladislav Golyanik, Weipeng Xu, Patrick Pérez, and Christian Theobalt. Neural monocular 3d human motion capture with physical awareness. *ACM Transactions on Graphics*, 40, aug 2021. 3

[101] Soshi Shimada, Vladislav Golyanik, Weipeng Xu, and Christian Theobalt. Physcap: physically plausible monocular 3d motion capture in real time. *ACM Transactions on Graphics*, 39, dec 2020. 3

[102] Matthew Trumble, Andrew Gilbert, Charles Malleson, Adrian Hilton, and John Collomosse. Total capture: 3d human pose estimation fusing video and inertial sensors. In *2017 British Machine Vision Conference (BMVC)*, 09 2017. 1, 2

[103] Vicon. Award winning motion capture systems. Website. https://www.vicon.com/. 2

[104] Miomir Vukobratovic and Branislav Borovac. Zero-moment point - thirty five years of its life. *I. J. Humanoid Robotics*, 1, 03 2004. 3

[105] Xinyu Yi, Yuxiao Zhou, and Feng Xu. Transpose: Real-time 3d human translation and pose estimation with six inertial sensors. *ACM Transactions on Graphics*, 40, 08 2021. 1, 2

[106] Petrissa Zell, Bodo Rosenhahn, and Bastian Wandt. Weakly-supervised learning of human dynamics. In *ECCV*, 07 2020. 3

[107] Yi Zhou, Connelly Barnes, Jingwan Lu, Jimei Yang, and Hao Li. On the continuity of rotation representations in neural networks. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 1