# Neural Network-Based External Impact Force Estimation for Compliant Mobile Robots

Marina Kollmitz Tobias Schubert Wolfram Burgard

*Abstract*— **Most of the conventional approaches to mobile robot navigation avoid any kind of contact with the environment or with humans. However, most distance sensors have a limited field of view, so that collisions cannot be fully avoided in practical mobile robot applications. At the same time, direct physical contact also can be used as a means of intuitive communication between a robot and humans. We present a whole-body sensory concept based on a 6-DoF force-torque sensor to detect interaction forces between the robot body and humans. To distinguish between external contact forces and forces that result from the accelerations of the mobile platform, we employ a novel neural network-based filtering approach. The network fuses information from an inertial sensor, odometry data and the 6-DoF force-torque sensor. Our approach allows the robot to detect and react to physical contact during autonomous motion. Extensive experiments with our robot Canny demonstrate the effectiveness of our approach.**

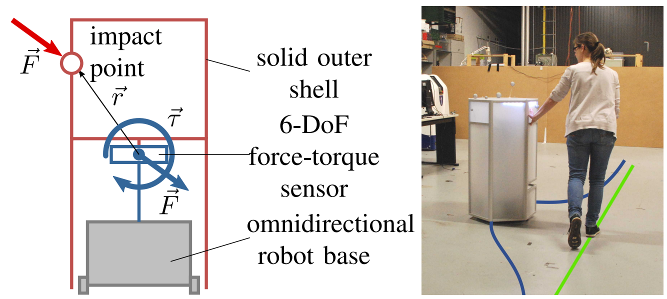


Fig. 1. Whole-body tactile sensory concept and robot prototype reacting to an external interaction force.

## I. Introduction

Robots increasingly share their workspaces with humans, which requires new concepts for human robot interaction. Unwanted contact with the environment or with people must be handled appropriately to ensure human safety. The close proximity also allows for intuitive and efficient deliberate physical interaction between humans and robots. One possible interaction scenario is shown in Fig. 1, where a person pushes the robot out of her desired path.

Our approach enables the robot to feel interaction forces on the entire robot body, in rest and during autonomous motion. Our whole-body sensory concept is based on one six degrees of freedom (DoF) force-torque sensor mounted stiffly on an omnidirectional mobile base. We employ an end-to-end learning approach with a multi-task output to distinguish between external forces on the robot shell and disturbance forces, e.g., caused by the motion of the robot. The sensory concept enables us to precisely estimate the magnitude, direction and location of an external force on the entire robot body by using just one sensor.

## II. Related Work

Work on force-controlled robots dates back to the 1990's when Khatib [1] introduced the robotic assistant for co-operative object manipulation between humans and mobile manipulators. Walking helper systems for the elderly [2, 3, 4] or helpers for cooperative object transport [5] have been presented that react to a force exerted by the user, measured e.g., by strain gauges or 6-DoF force-torque sensors. The approaches are similar to ours in that a force sensor adds sensitivity to a mounted stiff structure. However, the motion of the presented devices is always a reaction to the user's intended force. Instead, we strive to combine autonomous motion with a compliant reaction to interaction forces.

Kim et al. [6] estimated the external forces based on the joint torques of their mobile base with torque sensors in the drive trains of the three omnidirectional wheels. Their robot can react to interaction forces and detect collisions with the entire robot body, also when the robot is moving. However, they can only estimate the position and direction of the external force in the horizontal plane due to the limited degrees of freedom of the robot. Furthermore, they do not evaluate the location, direction and magnitude of the external force when the robot is in motion.

## III. Whole-Body Sensory Concept

Our robot Canny is based on the omnidirectional research platform Robotino. The Robotino is shipped with a mounting tower that attaches tightly to the base. We attach a high-precision 6-DoF force-torque sensor to the mounting tower as depicted in Fig. 1. The other side of the force-torque sensor is attached to a solid robot shell, made of aluminum profiles and semi-transparent acrylic glass.

For known convex shell geometries it is possible to calculate the impact locations of external forces on the shell from the forces and torques measured by the force-torque sensor, assuming that the force results from pushing the robot instead of pulling. Thus, Canny can perceive external forces using just one sensor, instead of using an expensive sensor skin or a sensor array. Furthermore, the sensor is not directly exposed to the environment but protected by the solid shell.

*This work has been partially supported by the German Federal Ministry of Education and Research (BMBF), contract number 01IS15044B-NaRKo.
The authors are with the Faculty of Computer Science, University of Freiburg, Freiburg, Germany. {kollmitz, tobschub, burgard}@informatik.uni-freiburg.de

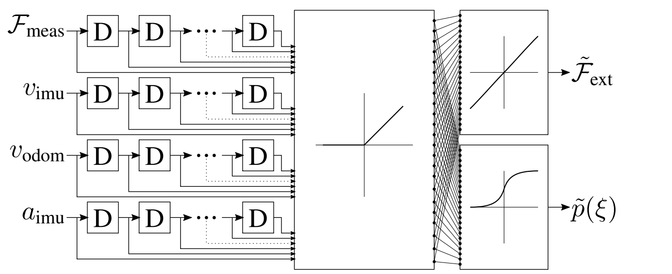


Fig. 2. Employed time-delay neural network structure. The inputs are delayed with time delays $\boxed{\text{D}}$ before entering the neural network. The network is split up into one regression and one classification part.

Fig. 3. Force stick used to exert and measure external forces on the robot shell. The optical markers are used to determine the 6D pose of the stick with respect to the robot.

## IV. Neural Network for External Force Estimation

The external forces acting on the robot shell are in practice overlaid by disturbance forces introduced by the robot motion or oscillations of the structure. We employ a model free filtering approach based on a time-delay neural network (TDNN) [7] to separate the external forces from the disturbance forces. TDNNs capture temporal information by showing the network multiple consecutive data points of a time series at one instance. To this end, the (discrete) input time series is delayed and buffered before entering the network. The TDNN has a feed-forward structure and can be trained using standard back-propagation.

In the following, we denote the true external wrench acting on the robot shell as $\mathcal{F}_{\text{ext}} = [F_{\text{ext}}, \tau_{\text{ext}}] \in \mathbb{R}^6$, the measured wrench is denoted as $\mathcal{F}_{\text{meas}} = [F_{\text{meas}}, \tau_{\text{meas}}] \in \mathbb{R}^6$. Furthermore, we introduce a random variable $X$ that takes the values $\{\xi, \bar{\xi}\} = \{\text{force impact}, \text{no force impact}\}$ and captures whether an external force currently acts on the robot shell. Our network generates estimations $\hat{\mathcal{F}}_{\text{ext}}$ for $\mathcal{F}_{\text{ext}}$.

Our TDNN has two output modalities, one regression output $\tilde{\mathcal{F}}_{\text{ext}}$ and one classification output $\tilde{p}(\xi)$. $\tilde{\mathcal{F}}_{\text{ext}}$ estimates the values of the external wrench components, $\tilde{p}(\xi)$ estimates whether an external force is present on the robot shell. Finally, we fuse both output modalities and calculate the expectancy of $\tilde{\mathcal{F}}_{\text{ext}}$, given $\tilde{p}(\xi)$ as our final filter result

$$
\begin{aligned}
\hat{\mathcal{F}}_{\text{ext}} &= E_{\tilde{p}(X)}(\tilde{\mathcal{F}}_{\text{ext}}) && (1)\\
&= \tilde{\mathcal{F}}_{\text{ext}}\tilde{p}(X = \xi) + \bar{\tilde{\mathcal{F}}}_{\text{ext}}\tilde{p}(X = \bar{\xi}), && (2)
\end{aligned}
$$

where the external wrench without force impact is $\bar{\tilde{\mathcal{F}}}_{\text{ext}} \equiv 0$.

The input to our network is the wrench $\mathcal{F}_{\text{mess}}$ measured by the force-torque sensor. We further input velocity $v$ and acceleration $a$ measurements from an inertial measurement unit (imu) on the robot base and the robot's odometry to capture the base excitation caused by the motion of the robot. Our imu measures linear accelerations $a_{\text{imu}} \in \mathbb{R}^3$ and angular velocities $v_{\text{imu}} \in \mathbb{R}^3$. The odometry gives the robot's omnidirectional velocity in the plane, $v_{\text{odom}} \in \mathbb{R}^3$.

We delay and buffer the six imu measurements, three odometry measurements and the six wrench measurements by $n$ time steps. The input to our network is given by $x^i = [\mathcal{F}_{\text{meas},i-n:i}, v_{\text{imu},i-n:i}, v_{\text{odom},i-n:i}, a_{\text{imu},i-n:i}]$, where $i$ is the current time step in the sequence and $i - n : i := i - n, \ldots, i$.

Our network structure is depicted in Fig. 2. The input vector is passed to the shared part of the network with non-linear rectified-linear units (ReLUs). The network then splits up into separated regression and classification parts. The regression part consists of neurons with linear activations and outputs $\tilde{\mathcal{F}}_{\text{ext}}$. The classification layer outputs $\tilde{p}(\xi)$ with non-linear sigmoid activations.

The network parameters $\phi$ are optimized for all $N$ training examples using stochastic gradient descent with momentum according to

$$
\phi^* = \underset{\phi}{\operatorname{argmin}} \sum_{i=1}^{N} \mathcal{L}\left(\tilde{y}^i, y^i\right), \tag{3}
$$

where $y^i$ denotes the ground truth values of the estimated outputs, or labels. The estimated output $\tilde{y}^i$ is given by a forward pass of the input $x^i$ through the network. Our loss function $\mathcal{L}\left(\tilde{y}^i, y^i\right)$ is a multi-task loss that comprises of the squared Euclidean norm of the regression output $\tilde{F}_{\text{ext}}$ and the cross-entropy loss of the classification output $\tilde{p}(\xi)$:

$$
\begin{aligned}
&\mathcal{L}\left(\tilde{y}^i, y^i\right) = w_r \|\mathcal{F}_{\text{ext}} - \tilde{\mathcal{F}}_{\text{ext}}\|^2 \\
&\quad - w_c \left[p(\xi)\log(\tilde{p}(\xi)) + (1 - p(\xi))\log(1 - \tilde{p}(\xi))\right]
\end{aligned} \tag{4}
$$

The weight factors $w_r$ and $w_c$ are hyperparameters that can be adjusted for the desired regression and classification performance. We found that $w_r = 10w_c$ yields good overall results. Another important hyperparameter of the network is the number of time steps $n$ delayed and buffered at the input. We observed that a value of $n = 10$, which corresponds to a time window of $0.2\,\text{s}$ at our sampling frequency of $50\,\text{Hz}$, sufficiently captures the temporal information of the data.

## V. Experiments

We assembled a force stick depicted in Fig. 3 to generate the ground truth external force magnitudes. It is based on a flexible 1D pressure sensor attached to a wooden stick. The direction of the force is obtained by the pose of the force stick with respect to the robot shell, which we obtain from a motion capture system. The force stick is used to exert forces as shown in Fig. 3 in order to generate training data for the neural network and testing data for the presented experiments. We conducted experiments where the robot

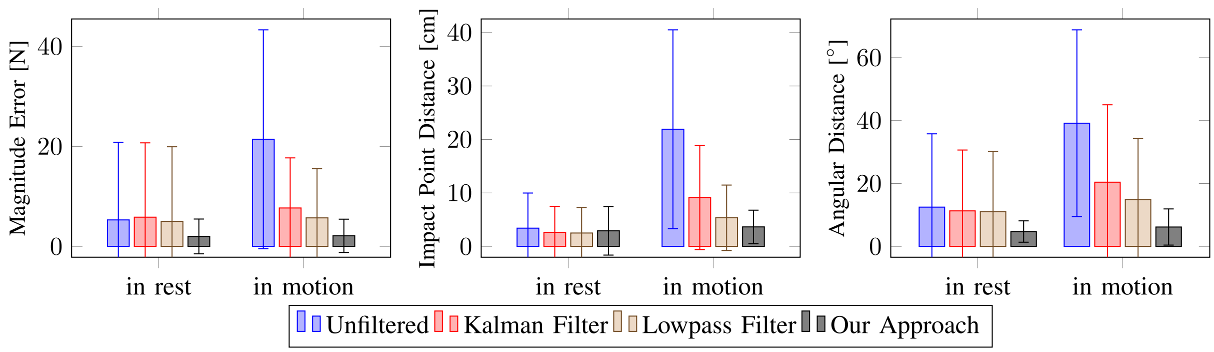


Fig. 4. Magnitude error, impact point distance and angular distance for the external force estimation, in rest and in motion.

is standing still and experiments where the robot is tele-operated at varying rotational and translational speeds.

The first experiment evaluates the ability of our approach to predict the magnitude, direction and location of an external force on the robot shell. To this end, we collected two data sets, one where the robot is in rest and one where the robot is tele-operated. We compare our approach to the unfiltered case, a simple Kalman filter and a Finite Impulse Response (FIR) lowpass filter. The Kalman filter uses a constant motion model and its covariances are obtained from the training data. Note that we compensate for the filter delay of the lowpass filter when calculating the comparison metrics.

Fig. 4 shows the mean absolute error of the estimated force magnitude and the mean impact point distance and angle between the true and estimated force vector. While all approaches give comparable results when the robot is in rest, the performance decreases drastically when the robot is in motion for the baseline approaches. Only our approach is able to accurately estimate the external forces when the robot is moving. The results show that our sensory concept combined with the neural force filtering is able to accurately predict the magnitude, direction and location of external force impacts.

We conducted a second experiment where we exert a known external force on the robot shell while the robot is driving on different floors to examine their influence on the induced disturbance forces. Here, we only compare the force magnitude, because we cannot obtain the direction of the external forces outside the motion capture area.

In addition to the PVC floor of the motion capture area for which we trained our approach and adjusted the baselines, we tested on two other floors, stone and carpet. Table I presents the mean absolute force magnitude error for our approach and the baselines. Our approach outperforms all baselines on all tested floor conditions. The results show that our approach generalizes well to previously unseen floor conditions, which indicates that our sensory concept can successfully be employed outside our robot hall.

| | PVC $\mu \pm \sigma$ | Stone $\mu \pm \sigma$ | Carpet $\mu \pm \sigma$ |
|---|---|---|---|
| Unfiltered | $21.23 \pm 21.92$ | $30.04 \pm 27.64$ | $20.00 \pm 18.68$ |
| Lowpass filter | $5.95 \pm 9.92$ | $7.16 \pm 12.02$ | $4.80 \pm 6.67$ |
| Kalman filter | $7.88 \pm 10.22$ | $9.94 \pm 12.19$ | $7.82 \pm 8.82$ |
| Our Approach | $\mathbf{2.13 \pm 3.32}$ | $\mathbf{2.82 \pm 4.59}$ | $\mathbf{3.08 \pm 4.00}$ |

TABLE I

MEAN ABSOLUTE FORCE MAGNITUDE ERROR $\mu$ AND STANDARD DEVIATION $\sigma$ FOR DIFFERENT FLOOR CONDITIONS IN N

## VI. Conclusion and Future Work

We presented a sensory concept that enables us to estimate forces and torques resulting from external force impacts onto the shell of a mobile robot. A neural network-based filtering approach distinguishes external forces from forces stemming from oscillations or force stimuli introduced by motions of the robot. Experiments with our robot Canny demonstrate that our neural filtering technique outperforms standard filtering methods. Our tactile sensor concept enables the robot to precisely estimate the magnitude, location and direction of the impact force, even when the robot is in motion. In the future, we want to use the presented framework for estimating external forces within a navigation system allowing our robot to provide force compliance.

## References


[1] O. Khatib, "Mobile manipulation: The robotic assistant," *Rob. Auton. Syst.* , vol. 26, pp. 175–183, Feb. 1999.

[2] A. Sabatini, V. Genovese, and E. Pacchierotti, "A mobility aid for the support to walking and object transportation of people with motor impairments," in *IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS)*, Lausanne, Switzerland, Oct. 2002, pp. 1349–1354.

[3] M. Spenko, H. Yu, and S. Dubowsky, "Robotic personal aids for mobility and monitoring for the elderly," *IEEE Trans. Neural Syst. Rehab. Eng.*, vol. 14, pp. 344–351, Sep. 2006.

[4] Y. Hirata, T. Baba, and K. Kosuge, "Motion control of omni-directional type walking support system "walking helper"," in *Proc. IEEE Int. Workshop Robot and Human Interactive Communication (ROMAN)*, Millbrae, CA, Oct. 2003, pp. 85–90.

[5] Y. Hirata, T. Takagi, K. Kosuge, H. Asama, H. Kaetsu, and K. Kawabata, "Map-based control of distributed robot helpers for transporting an object in cooperation with a human," in *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, Seoul, Korea, May 2001, pp. 3010–3015.

[6] K. S. Kim, T. Llado, and L. Sentis, "Full-body collision detection and reaction with omnidirectional mobile platforms: a step towards safe human-robot interaction," *Auton. Robots (AuRo)*, vol. 40, no. 2, pp. 325–341, Feb. 2016.

[7] A. Waibel, T. Hanazawa, G. Hinton, K. Shikano, and K. Lang, "Phoneme recognition using time-delay neural networks," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 37, pp. 328–339, Mar. 1989.