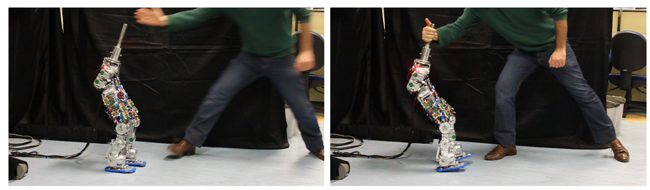
The Anatomy of a Fall: Automated Real-time Analysis of Raw Force Sensor Data from Bipedal Walking Robots and Humans

Petar Kormushev[1], Barkan Ugurlu[2], Luca Colasanto[1], Nikolaos G. Tsagarakis[1], Darwin G. Caldwell[1]

***Abstract*— An automated approach is proposed which can analyze ground reaction force data from bipedal walking robots and humans. The input of the automated analysis is the raw data from force sensors mounted in the feet of a robot. The output is detailed information, such as detected single support, double support, and swing phases, their durations, timings of events like heel strikes, properties of the phase transitions and of the robot itself. The proposed approach is generic, parameter-free, model-free, robust, computationally efficient, and applicable for real-time use during walking. It can detect early indications of instability that could lead to a fall of the robot. Three real-world experiments are presented: with a compliant bipedal robot, with a stiff humanoid robot, and with a human subject.**

(a) The robot falls forward, caused by mismatch of the reference and response phase timings. It can be predicted by the earlier-than-expected heel strike events preceding the fall, which indicate robot instability.



(b) The robot falls backward, caused by smaller response velocity of the pelvis than the reference one. It can be predicted by the changed pressure distribution on the foot sole and the later-than-expected heel strike events.

Fig. 1: Two examples of falling of a bipedal walking robot. In both cases, the humans around the robot could anticipate very early the imminent fall, and reacted quickly to catch the robot before it hits the ground.

I. INTRODUCTION

Humanoid robots have been growing progressively more and more complex both in hardware and software. Currently, they possess on the order of 50-100 degrees of freedom, and have multiple sensor modalities: position/velocity, force/torque, orientation/acceleration, tactile perception, computer vision, etc. As a result, they require increasingly more sophisticated methods at many levels for planning, control, perception, sensor fusion, etc.

Despite the abundance of sensory feedback, the state-of-the-art humanoid robots unfortunately do not yet make full use of the available data. This is largely caused by the difficulty of *semantic parsing* of the sensory data, which is the process of making sense out of the noisy and inconsistent raw data. The goal of semantic parsing is to transform the input sensory data into meaningful information, in a form readily useful for the robot controllers, planners, and other modules. This is an extremely difficult cognitive task, and solving it is almost as difficult as solving the whole artificial intelligence problem.

In this paper, we have a more modest goal. Instead of trying to come up with an extremely complicated system for solving some contrived problem, here we try to propose a simple and efficient system for solving the particular real-world problem of semantic parsing of raw force sensor data from bipedal walking robots and humans.

We embrace the philosophy that it is better to spend efforts extracting as much as possible useful information from a few simple sensory modalities, rather than greedily fusing multiple expensive sensors together and ending up producing relatively little useful information. There are many arguments to support this philosophy. For example, imagine the following experiment: a person walking inside a dark room on a ship. The person's main sensory modality (vision) is useless. The vestibular system is strongly challenged, because of the movement of the ship. The only reliable sensory modality left is the sense of force[1] from the feet. And, this little source of information is still enough for a person to walk robustly even in such situation. This demonstrates that our brain can extract a large amount of useful information even from a low-quantity and low-quality sensory source such as the force perception.

Going back to bipedal robots, there is a more relevant example. An expert roboticist, presented with the raw force sensor data from the robot's feet, is able to extract an amazing amount of useful information, such as: stance and swing phases, their durations, timings of events like heel strikes, the mass of the robot, even the rigidity of the floor, as well as any changes in these parameters, e.g. starting to walk uphill, changes in the mass of the robot, etc.

In this paper, we propose an algorithm that can perform

[1]P. Kormushev, L. Colasanto, N. G. Tsagarakis, and D. G. Caldwell are with the Department of Advanced Robotics, Istituto Italiano di Tecnologia, via Morego 30, 16163 Genova, Italy. {petar.kormushev, luca.colasanto, nikos.tsagarakis, darwin.caldwell}@iit.it

[2]B. Ugurlu is with Control Systems Laboratory, Toyota Technological Institute, 468-8511 Nagoya, Japan. barkanu@toyota-ti.ac.jp

[1]Throughout this paper, by 'force' we mean the ground reaction force.

such an analysis of raw force sensor data from bipedal walking robots, and is able to produce most of the same information which an expert roboticist can, given the same data. Our focus is on producing a generic solution that can work for any bipedal robot without requiring any *a priori* information about the robot itself, and without any parameter tuning.

II. ANALYSIS OF ROBOT FALL

A robot fall does not happen instantaneously. Instead, there is a short period of time leading into the fall. It is exactly this period of time immediately preceding a fall that we investigate here.

For example, Fig. 1 shows two cases in which a bipedal walking robot lost its balance. In both cases, the experimenters around the robot were able to predict much in advance the imminent fall, and reacted quickly to catch the robot before it fell down completely. Also for both cases, after careful *post-hoc* examination of the force sensor data, it seemed possible to predict the instabilities well ahead in time before the actual tipping over, just by looking at the force data signal from the foot sensors. Although the falls themselves happen very quickly, within half a second, the indications of instability leading to the fall take a relatively longer time, around 1-2 seconds. In this paper, we investigate automated means for analyzing the force data in order to detect these early indications prior to a fall.

The ultimate goal is to detect if something is going wrong as early as possible, so that there is enough time to re-plan and react. The planning and plan execution is outside the scope of this paper. Instead, we focus entirely on the preliminary phase of analysis of the live streaming sensor data. We propose a method able to analyze in real-time the force sensor input and detect early anomalies that could potentially lead to a fall.

III. RELATED WORK

The systematic study of animal locomotion, especially of human motion, is called *'gait analysis'* [1]. Gait analysis on human subjects is usually performed using motion capture systems and force sensing platforms. It can give valuable insights about the human balance and posture control during standing and walking [2].

Early fall detection is usually investigated for soccer playing robots, participating in RoboCup competitions. Renner and Benke implemented a method in which attitude sensor readings are modeled via sinusoids to detect velocity changes [3]. As another example, Ruiz *et al* processed attitude sensor information using Kalman filters to predict variations in robot configuration [4]. Both methods are efficient for small scale humanoids, but they may not be applicable to humanoids with larger sizes. Usually, postural balance of a humanoid can be recovered through contact force optimization [5], [6]. With this in mind, we designed an automated method which can analyze the force sensory data, acquired from sensors in the feet of a robot or a human.

IV. CHALLENGES FOR AUTOMATED ANALYSIS OF RAW FORCE SENSOR DATA

Manually analyzing raw force sensor data is a challenging task. It is even more challenging to create an automated system that can perform such analysis without any human intervention, and produce results as good as a robotics expert would, given the same data. This is exactly the problem we are trying to solve in this paper. Our goal is to design an algorithm for automated analysis, which takes as an input raw force signal, such as the one shown in Fig. 3a, and produces from it as much useful information as possible.

Such a task can be done relatively easily if an expert manually (often empirically) hand-crafts a set of 'magic numbers' (e.g. threshold values and other parameters). There are many such examples in existing published works, e.g. [7], [5], [8]. However, we set out to solve a more challenging problem. The goal of our proposed automated analysis approach is to meet the following challenges:

- to be *generic*, applicable to any bipedal robot and force acquisition sensor;
- to be *parameter-free*, not needing any parameter tuning;
- to be *model-free*, not requiring any prior knowledge about the robot itself;
- to be *plug-and-play*, directly applicable without any algorithm changes;
- to be *computationally efficient*, and thus applicable for real-time use;
- to be *robust*, and work even with high level of noise and missing data;
- not to require perfect *sensor calibration*, and work even if the sensor data are drifting over time.

V. PROPOSED APPROACH FOR AUTOMATED ANALYSIS OF FORCE SENSOR DATA

For illustration purpose and for consistency, throughout this section we use the same example for ground reaction force raw signal, shown in Fig. 3a.

A. Justification of the proposed approach

Performing a single all-in-one analysis of the raw signal in an automated way is rather difficult and error-prone. Instead, we employ the *'waterfall methodology'* in which the analysis is performed at distinct sequential stages. Aiming for high efficiency, this methodology removes the possibility of going back to a previous analysis stage to correct potential errors. Therefore, the order of the stages is very important. In order to minimize the probability of error, the analysis stages are sorted by their level of complexity, i.e. the easiest things to detect are done first, and the most difficult ones are left for the final stages. Another justification for this methodology is that completing the analysis of the early stages results in more information available for the analysis in the more difficult later stages.

Our second design principle, aiming to reduce the overall complexity, is the so-called *'explaining away'* principle. The main idea is that if there is a simple explanation for some observed data, then this reduces the probability that another, more complex explanation of the same observation at a later analysis stage is true. This principle shares similarity with *Occam's razor*. In combination with the waterfall methodology, it allows us to gradually narrow down the possible range of explanations as the analysis advances. This helps especially in the later stages of the analysis.

Guided by these two principles, we segment our proposed automated analysis in the following consecutive stages:

- Analysis of swing phases
- Analysis of single support phases
- Analysis of double support phases
- Analysis of phase transitions
- Analysis of robot properties
- Analysis of anomalies

B. Analysis of swing phases

This analysis stage, as well as most of the following stages, is divided in two parts: detection of the phases, and extraction of the phase properties.

1) Detection of swing phases: The detection of swing phases is trivial for a human expert, which is the reason we selected this as the first analysis stage. Still, it is rather challenging for the automated analysis because at this first stage the analyzer has zero knowledge about the signal.

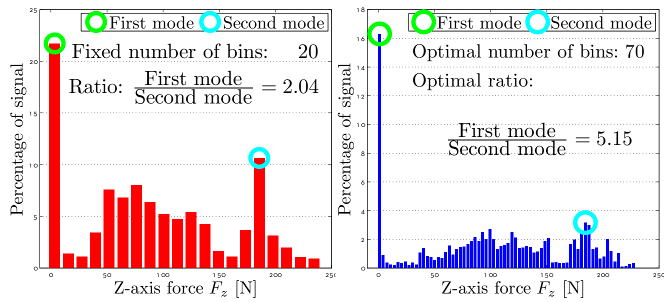
Normally, a human expert would simply look for 'relatively flat' horizontal segments within the signal and mark them as swing phases. For the automated analysis, however, it is not so simple. Since we assume that the force sensors are not perfectly calibrated, and we have no information about the mass of the foot sole, nor about the ambient force signal noise level, it is inappropriate to use a fixed pre-determined threshold value.

Instead, we base the analysis on the observation that, in relative terms, the signal variance and noise level in swing phases is much lower than in the other phases. This stems from the fact that during swing phases the force sensor has to carry only the weight of the foot sole (which is small), and that the vibrations tend to have low amplitude and high frequency.

Based on this observation, the analyzer can narrow down the range in which to look for 'relatively flat' segments. We propose to achieve this range reduction using *histogram analysis*, as shown in Fig. 2. The intuitive idea is that the parts of the signal which have relatively low noise level and are relatively flat would tend to cluster together in one or few bins of the histogram. Thus, the problem gets transformed into the trivial problem of finding the mode[2] of the histogram, and its corresponding bin interval.

Histograms have many advantages which are appropriate for our goal. They are simple and efficient to implement, with computational complexity $O(n)$, which is good for real-time use. Furthermore, histograms can be implemented in an incremental way, e.g. using a sliding window to

[2]By 'mode' (or 'first mode') we mean the highest value of the distribution defined by the histogram. By 'second mode' we mean the second highest value of the histogram.



(a) Analysis of right-leg data using an arbitrary number of bins

(b) Analysis of left-leg data using the optimal number of bins which maximizes the mode ratio

Fig. 2: Histogram analysis of the raw force sensor data

incrementally add and remove particles from the bins, thus reducing the computational effort to $O(1)$ per update.

However, histograms have an open parameter – the number of bins – which is against our goal for completely parameter-free analysis. To solve this problem, we propose a simple strategy for choosing an optimal number of bins. We introduce an indicator showing how informative a histogram is, defined as:

$$\textit{Histogram ratio} = \frac{\textit{Value of first mode}}{\textit{Value of second mode}} . \quad (1)$$

This indicator is used to choose an optimal number of bins, by selecting the number that maximizes the indicator value, as illustrated in Fig. 2. The same value is then used also as a *'confidence level'* of this analysis stage, to represent how confident the analyzer is in the correctness of the analysis results. The pseudo-code for the proposed histogram-based algorithm is given in Algorithm 1.

2) Extraction of swing phase properties: By swing phase properties we mean the baselines (the ground truth value or best estimate of it) for each leg individually, as well as the

Algorithm 1 Optimal Histogram Analyzer

Input: raw signal $S \in \mathbb{R}^N$
Output: optimal ratio $R_{opt} \in \mathbb{R}^+$, optimal number of bins $K_{opt} \in \mathbb{Z}^+$, optimal histogram $H_{opt} \in \mathbb{R}^{K_{opt}}$, optimal bin's range $B_{opt}\langle b_{min}, b_{max}\rangle$
$R_{opt} = 0$
for $K = 2$ **to** N **do**
  {calculate histogram of S using K number of bins}
  $H = calcHistogram(S, K)$
  $M_{first} = \max_i \{H(i)\}$ {the first mode}
  $I_{first} = \text{argmax}_i \{H(i)\}$
  $H(I_{first}) = 0$
  $M_{second} = \max_i \{H(i)\}$ {the second mode}
  $R = M_{first}/M_{second}$
  if $R > R_{opt}$ **then**
    $R_{opt} = R, \quad K_{opt} = K, \quad H_{opt} = H$
    $B_{opt}\langle b_{min}, b_{max}\rangle = getBin(H_{opt}, I_{first})$
  end if
end for

timings (beginning and ending time of each instance of the phase).

Although histogram analysis narrows down the range of force values, it does not produce a single best estimate for the swing phase baseline. A conventional approach for doing this would be to use least squares regression to fit a horizontal line model to the signal. However, such an approach would require a considerable computational effort, either for matrix inversion (which could be numerically unstable), or for gradient descent (which could be too slow and dependent on a learning rate open parameter). In either case, it conflicts with our goals for efficient real-time and parameter-free analysis. To solve this problem, we borrow ideas from another scientific field - computer vision.

In computer vision, there is an algorithm for detecting objects, such as lines and planes, called RANSAC (*Random Sampling Consensus Algorithm* [9]). It is essentially a Monte Carlo type approximation algorithm. RANSAC-based object detectors are very simple, very efficient, and very robust. For example, it is not unusual for RANSAC to detect an object even if as much as 50% of it is occluded. Such robustness is well appreciated for our problem, where we have unknown level of noise and strict computational complexity restrictions.

Applying RANSAC for estimation of the swing phase baseline is fairly straightforward. For completeness, we include pseudo-code of the proposed RANSAC baseline estimator in Algorithm 2. Please note that the RANSAC algorithm works well for a very wide range of values for the parameters σ and P, so they do not need tuning. Both the histogram analysis and the RANSAC detector are applied individually to each leg's signal, because the baseline is not necessary to coincide since the sensors are not perfectly calibrated. The results from applying RANSAC to a sample force signal is shown in Fig. 3b.

The next step is extraction of swing phase timings, i.e. beginning and ending time of each instance of the phase. A conventional approach for doing this would be to use a fixed pre-determined threshold value defining a distance from the phase baseline. The parts of the signal within this distance of the baseline would be marked as belonging to a swing phase, as shown in Fig. 3c. The problem with such an approach, apart from relying on a hand-crafted 'magic number', is that it is affected badly by noise. The example in Fig. 3c shows that there is bad undesired fragmentation of the detected swing phase segments.

To solve this problem, we propose a probabilistic approach. We introduce a probability of belonging to a swing phase, defined for each point of the signal as follows:

$$P^E_{swing}\left(S^E(i)\right) = \left(1 - \frac{|S^E(i) - B^E_{swing}|}{max(S^E) - min(S^E)}\right)^{\rho}, \quad (2)$$

where E is the leg ('right' or 'left'), S^E is the raw signal for leg E, and B^E_{swing} is the baseline value for this leg computed by the RANSAC estimator. The power ρ allows us to increase the distinction between swing- and non-swing-phase points, which helps to avoid undesired fragmentation successfully, as illustrated in Fig. 3d and 3e. Although we use a fixed probability threshold of 90% to segment the swing phase, this probabilistic approach is not as sensitive to the threshold value as the conventional approach, and works well for a wide range of threshold values. For example, any value for $\rho \in (2, 10)$ seems to work well in practice.

C. Analysis of single support phases

As before, we divide the analysis in two parts.

1) Detection of single support phases: Here we apply the 'explaining away' principle, which implies that for the signal parts where we have high probability of swing phase for one leg, the probability of single support phase for the other is automatically increased. Formally:

$$P^E_{s.s.}\left(S^E(i)\right) = 1 - P^{\neg E}_{swing}\left(S^{\neg E}(i)\right), \quad (3)$$

where $\neg E$ is the opposite leg.

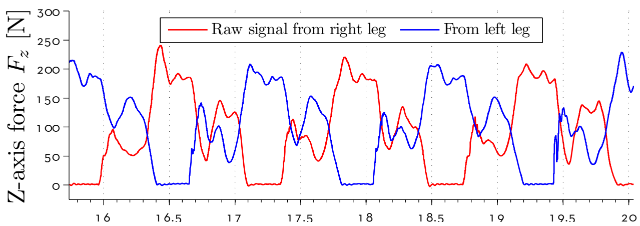
2) Extraction of single support phase properties: Histogram analysis is not useful in this case, because of the high variance in the single support phases. Instead, we exploit the robustness of the RANSAC detector and apply it directly on the 'explained away' signal (not the whole signal) to detect the baselines, one for each leg. The results are shown in Fig. 3f.

D. Analysis of double support phases

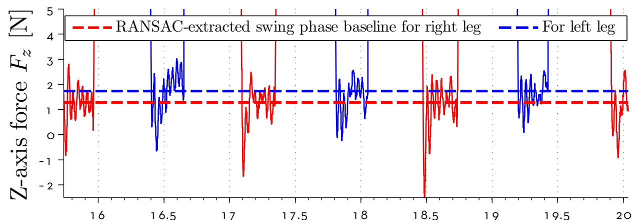
1) Detection of double support phases: Detection of the double support phases is, in general, extremely difficult, because of the huge variance of the signal in these phases. To bypass this difficulty, the proposed automated analysis utilizes fully the two adopted methodologies. Following the

Algorithm 2 RANSAC baseline estimator

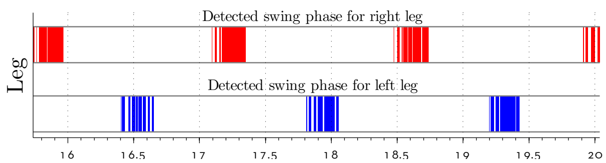
Input: raw signal $S \in \mathbb{R}^N$
Output: baseline $B \in \mathbb{R}$, confidence level $L \in \mathbb{R}^+$
$\sigma = 0.05$ {5% inlier selectivity of RANSAC}
$\epsilon = \sigma * (\max(S) - \min(S))$
$K_{max} = -\infty$ {max. number of inliers so far}
$P = \lceil(0.2 * N)\rceil$ {20% number of points to sample}
for $i = 1$ **to** P **do**
  Draw $u \sim U(0, 1)$
  $j = 1 + \lfloor(u * (N - 1))\rfloor$
  $K = 0$
  for $l = 1$ **to** N **do**
    if $|S(l) - S(j)| < \epsilon$ **then**
      $K = K + 1$
    end if
  end for
  if $K > K_{max}$ **then**
    $K_{max} = K$
    $B = S(j)$
  end if
end for
$L = K_{max}/N$



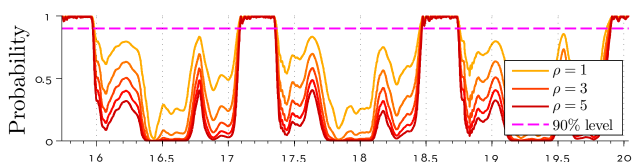
(a) Raw data signal from foot force sensors of humanoid robot COMAN



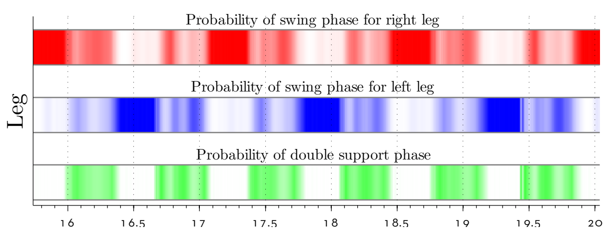
(b) Detection of swing phase baselines using RANSAC-based algorithm



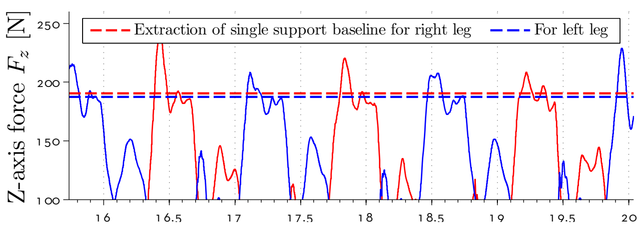
(c) Detection of swing phases based on thresholding - bad segmentation



(d) Probability of swing phase for right leg - improved segmentation

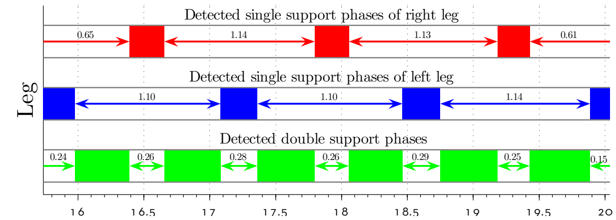


(e) Detected swing phases and double support phases based on their probabilities: P_{swing}^{right}, P_{swing}^{left}, and $P_{d.s.}$

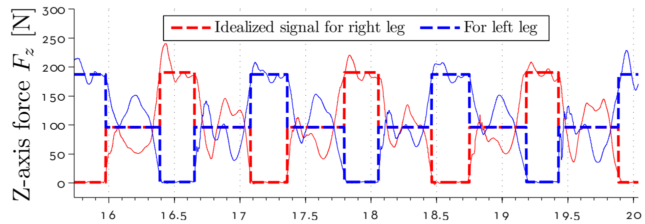


(f) Extraction of single support baselines using 'explaining away'

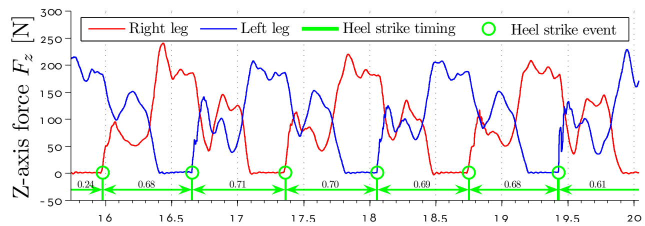
Fig. 3: Automated analysis of raw force sensor data from experiment (A) - bipedal walking robot COMAN (described in Section VI-A). All subfigures are derived from the same source signal, and have the same x-axis (Time [s]), omitted to save space.



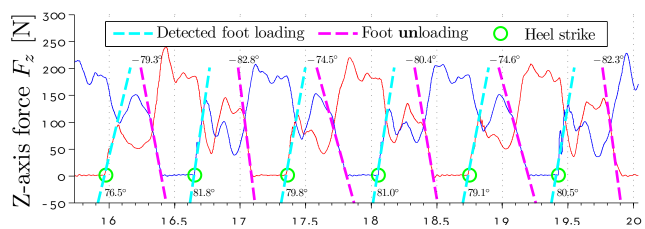
(a) Detection of all single- and double-support phases, and their timings



(b) Synthesizing an 'idealized' signal based on the extracted baselines



(c) Detection of heel strike events and their timing



(d) Analysis of phase transitions, foot loading and unloading patterns

Fig. 4: (Continued from Fig. 3)

'waterfall methodology', it assumes that all the information acquired from the previous analysis stages is true, and applies the 'explaining away' principle to narrow down the search scope. More formally, the detection of double support phases is performed using the following probability:

$$P_{d.s.}\left(S(i)\right) = 1 - max \begin{Bmatrix} P_{swing}^{E}\left(S^{E}(i)\right) \\ P_{swing}^{\neg E}\left(S^{\neg E}(i)\right) \end{Bmatrix}. \quad (4)$$

At this point in the analysis, all phases have been detected, and it is possible to calculate their exact timings, as shown in Fig. 4a.

2) Extraction of double support phase properties: RANSAC analysis is not useful in this case, because of the extremely high variance in the double support phases. Instead, we use simple averaging of the 'explained away' signal to detect the (just one) baseline for double support phase. Using all the five extracted baselines so far, it is possible to reconstruct a synthetic 'idealized' signal, as shown in Fig. 4b.

E. Analysis of phase transitions

1) Detection of heel strikes: Heel strikes mark the transitions from single support to double support phases and are very important events. They are easily detected at the endings of the swing phases, as shown in Fig. 4c.

2) Extraction of phase transition properties: The phase transitions are mostly characterized by the slope (rate of change) of the force. The slope reveals the foot weight loading and unloading patterns. In order to detect this slope, we use a modified version of the RANSAC algorithm, which is able to detect arbitrarily oriented lines in 2D. The difference with Algorithm 2 is that this time two points are sampled at each iteration, and a candidate line is fitted through them. The results from this phase transition analysis is shown in Fig. 4d.

F. Analysis of robot properties

From the collected information it is possible to calculate some global properties of the robot. For example, using the extracted baselines it is straightforward to estimate the entire robot's mass, which for the given example equals 19.673 kg.

G. Analysis of anomalies

The ability to distinguish between what looks normal and what seems abnormal requires a certain high level of cognition and accumulated experience. At this final stage of the automated analysis, there is plenty of the latter, i.e. accumulated knowledge about the sensory signal.

An in-depth anomaly detection analysis is beyond the scope of this paper. However, using the collected information about timings, baselines, and slopes by the analyzer, it is trivial to detect the following anomalies:

- too early/late heel strike events;
- too fast/slow foot loading/unloading;
- too high/low single support phase level (e.g. if extra weight is added to the robot);
- too small/big ground reaction force (e.g. if the robot is being lifted up in the air).

In fact, the analysis of the currently ongoing phase is already shown in the rightmost ends of Fig. 4a, 4b, 4c, and 4d. It can easily be used to detect deviations from the expected values. Fig. 5 illustrates some examples of unexpected events that can be detected this way.

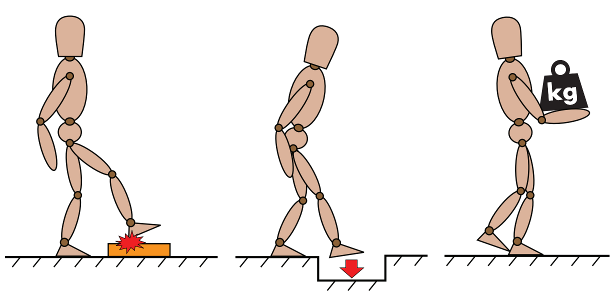


Fig. 5: Examples of unexpected events that can be detected in a fully automated way by the proposed method. From left to right, showing as follows: unexpected early heel strike, unexpected late heel strike, unexpected mass change of the robot.

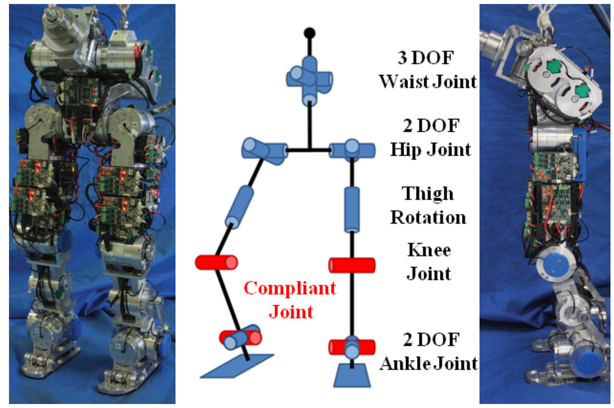


Fig. 6: Lower-body design of the compliant humanoid robot COMAN. This robot was used in experiment (A).

VI. EXPERIMENTAL EVALUATION

We used three different sources of real-world raw force sensor data, in order to evaluate experimentally the proposed automated analysis approach. These sources are:

- (A) - Compliant humanoid robot COMAN, equipped with 6-axis force/torque sensor in each foot sole;
- (B) - Stiff humanoid robot Fujitsu HOAP-2, equipped with 4 FSR pressure sensors in each foot sole;
- (C) - Human subject, wearing sensorized shoes with 4 pressure sensors in each shoe sole.

The proposed approach was able to successfully analyze, detect and extract correctly all the information about the single- and double-support phases, their durations, the timings of heel strike events, phase transition slopes, as well as the mass of the robot/human. Here we describe briefly the three systems we used as a source of force data to analyze.

A. Compliant humanoid robot COMAN

In experiment (A), we used the passively-compliant bipedal robot COMAN [10][11], shown in Fig. 6. We used only the lower body of the robot, which has a total of 15 active DoF (degrees of freedom): 6 DoF in each leg, and 3 DoF at the waist. The robot has passive compliance (via springs) in the two pitch joints (knee and ankle) of each leg, and has a force/torque sensor mounted in the sole of each foot, as shown in Fig. 7. For the compliant actuation system of COMAN, the series elastic actuator module described in [12] is used.

Making COMAN walk in a stable way is a challenging problem, because its springs can store and release energy, which violates the usual energy dissipation requirement for guaranteeing stability. The reason to use passive compliance is that it offers numerous benefits, such as increased safety, better shock absorption, adaptability to rough terrain, and improved energy efficiency [13]. However, the price paid is having a more difficult to control system. Fig. 1 shows two recent examples of falling of the COMAN robot, which were caused by two algorithmically generated reference trajectories for the center-of-mass which theoretically were within the ZMP limits. However, the interaction of the system with

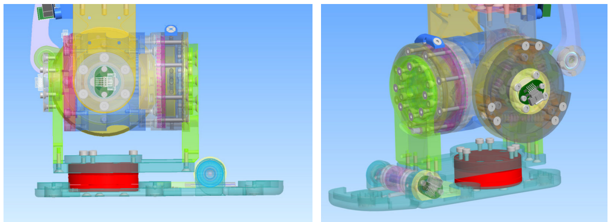


Fig. 7: A 3D model of COMAN robot's foot, indicating the location of the force sensor (in red), between the foot sole and the ankle joint. This sensor was used in experiment (A).

the springs and the floor accumulated enough undissipated energy to cause instability and make the robot fall down.

Fig. 2, 3, and 4 show results from the automated analysis of force data signal recorded from the COMAN robot walking on a hard linoleum-covered floor.

B. Stiff humanoid robot Fujitsu HOAP-2

In experiment (B), we analyzed data recorded from the 4 pressure sensors in each foot sole of the stiff[3] humanoid robot Fujitsu HOAP-2. The individual pressure sensor signals are grouped together and averaged, to produce a single z-axis force signal for each leg. Fig. 8 shows the results from the automated analysis of these data. The estimated mass of the robot is 5.59 kg.

An interesting observation for this particular signal is that more weight is put on the right leg during single support than on the left. This could be caused by inaccuracy in the robot mass distribution model or the center-of-mass position.

C. Human subject wearing sensorized shoes

In experiment (C), we used a human subject wearing sensorized shoes with 4 pressure sensors in each shoe sole, shown in Fig. 9. The results from the automated analysis of the recorded data is shown in Fig. 10. The estimated combined mass of the human subject and the sensorized shoes is 70.78 kg.

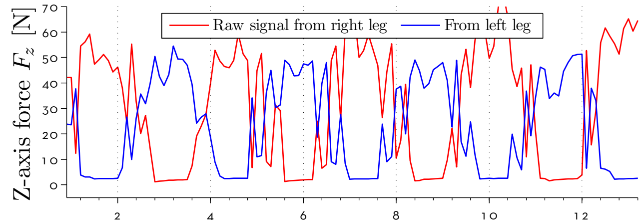
VII. DISCUSSION

It is important to highlight that the exact same algorithm was used to analyze the data from each of the three different sources, without any modification of the source code, or any parameter tuning whatsoever. This asserts the robustness of the proposed approach, and its ability to work on very different robots (in terms of hardware), or even on humans, in a completely automated way.

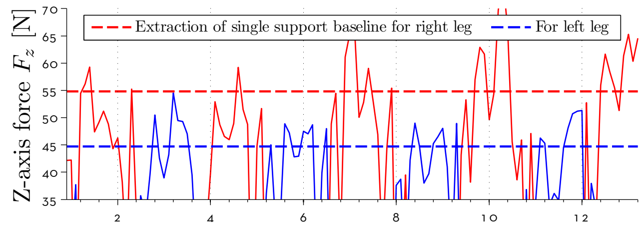
During the analysis, the left and right leg data signal are sometimes analyzed together (e.g. timings are shared and used for 'explaining away'), and sometimes separately (e.g. probabilities and baselines are measured separately for each leg).

The computational complexity of the proposed analysis is on the order of $O(n^2)$ with a very small hidden constant. It can further be reduced using incremental implementation

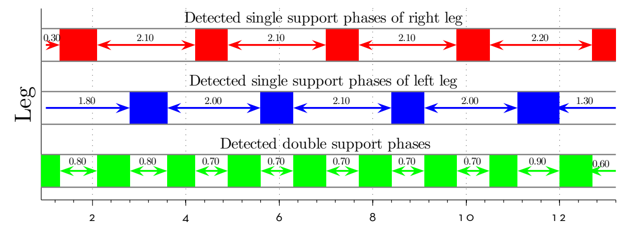
[3]By 'stiff' we mean not passively-compliant.



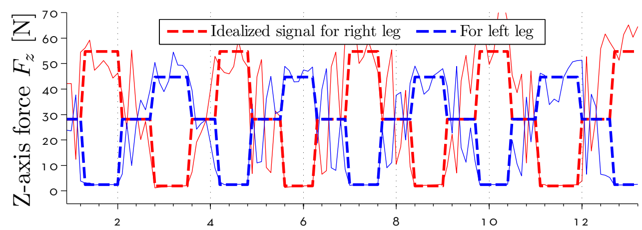
(a) Raw data signal from foot pressure sensors of humanoid robot HOAP-2



(b) Extraction of single support baselines



(c) Detection of all single- and double-support phases, and their timings



(d) Synthesizing an 'idealized' signal based on the extracted baselines

Fig. 8: Automated analysis of raw force sensor data from experiment (B) - stiff humanoid robot Fujitsu HOAP-2

with a sliding window along the signal. This makes the approach viable for embedded real-time applications.

Throughout the entire analysis, we assumed minimal knowledge of the robot properties, in order to make the approach as generalizable as possible. Due to this, the proposed method can be immediately used as a 'plug-and-play' module for any bipedal robot, regardless of the hardware differences. However, it is possible to add some *a priori* knowledge to the analyzer, which can additionally improve the robustness and fault-detection capabilities of the analysis.

One future direction, in which we are already working, is to make use of the confidence levels derived at the different analysis stages. These are a few potential uses:

- to detect if the robot is still in the air, e.g. to start walking only after the robot has been lowered down on the ground;
- to do sanity check of the input, e.g. detect if the input signal is not a proper force signal from bipedal walking;
- to detect hardware faults, e.g. detect malfunction of the

Fig. 9: The sensorized shoes used in experiment (C).

force sensor, such as too high noise level or rapid drift;
- to detect data corruption, e.g. sudden huge jumps in the signal might point to data acquisition faults.

A more challenging analysis that might be possible using supervised machine learning methods, could include:

- analysis of the impedance of the walking surface, e.g. detect floor/ground types (wood, carpet, sand, etc.);
- analysis of the terrain shape, e.g. detect up/down slopes, uneven terrain, obstacles, etc.

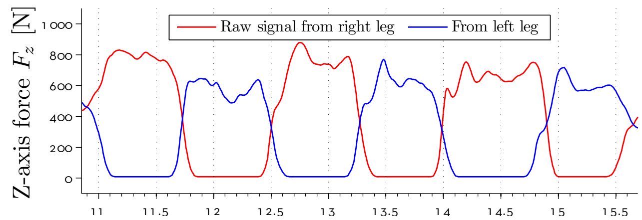
VIII. CONCLUSION

We proposed a simple and efficient approach for performing automated analysis of raw force sensor data from bipedal walkers. We proposed two different algorithms for performing the analysis: a histogram-based algorithm, and a RANSAC-based algorithm, borrowed from the computer vision domain. To improve the robustness of the analysis, we applied probabilistic methods for the detection of the phases. We evaluated the proposed approach experimentally on three different sources of real-world force sensor data. The proposed approach was able to successfully detect and extract all the desired information from these data.

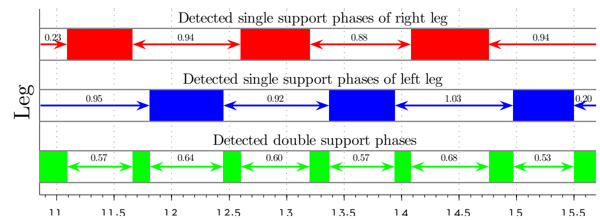
ACKNOWLEDGMENTS

The authors would like to thank Prof. Yoshikazu Kanamiya and Takumi Aikawa, from Department of Mechanical Systems Engineering at Tokyo City University, for conducting the walking experiments with Fujitsu HOAP-2 robot.

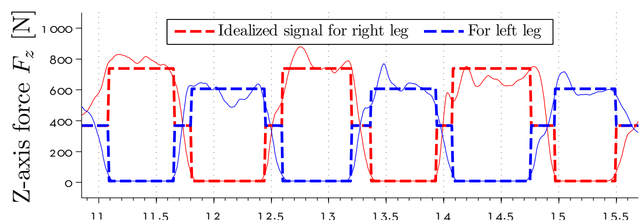
This work is partially supported by the AMARSi European project under contract FP7-ICT-248311.



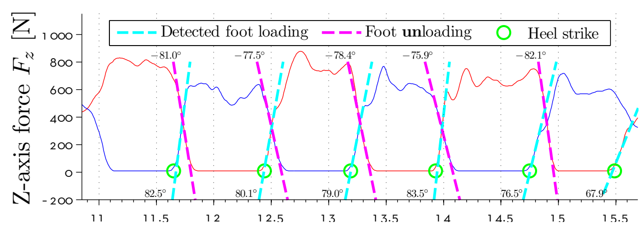
(a) Raw data signal from sensorized shoe sensors of human subject



(b) Detection of all single- and double-support phases, and their timings



(c) Synthesizing an 'idealized' signal based on the extracted baselines



(d) Analysis of phase transitions, foot loading and unloading patterns

Fig. 10: Automated analysis of raw force data from experiment (C) - human subject walking with sensorized shoes

REFERENCES

[1] M. W. Whittle, *Gait Analysis: An Introduction*. Butterworth-Heinemann Ltd, Oxford, 1991.

[2] D. Winter, "Human balance and posture control during standing and walking," *Gait & Posture*, vol. 3, no. 4, pp. 193–214, 1995.

[3] R. Renner and S. Behnke, "Instability detection and fall avoidance for a humanoid using attitude sensors and reflexes," in *Proc. IEEE/RSJ Intl Conf. on Intelligent Robots and Systems (IROS)*, Beijing, China, October 2006, pp. 2967–2973.

[4] J. Ruiz-del Solar, J. Moya, and I. Parra-Tsunekawa, "Fall detection and management in biped humanoid robots," in *Proc. IEEE/RAS Intl Conf. on Robotics and Automation (ICRA)*, Anchorage, US, May 2010, pp. 3323–3328.

[5] T. Buschmann, S. Lohmeier, and H. Ulbrich, "Biped walking control based on hybrid position/force control," in *Proc. IEEE/RSJ Intl Conf. on Intelligent Robots and Systems (IROS)*, St. Louis, US, October 2009, pp. 3019–3024.

[6] C. Ott, M. A. Roa, and G. Hirzinger, "Posture and balance control for biped robots based on contact force optimization," in *Proc. IEEE Intl Conf. on Humanoid Robots (Humanoids)*, Bled, Slovenia, October 2011, pp. 26–32.

[7] K. Van Heerden and A. Kawamura, "Maintaining floor-foot contact of a biped robot by force constraint position control," in *Proc. IEEE Intl Conf. on Mechatronics (ICM)*, Istanbul, Turkey, April 2011, pp. 857–862.

[8] Y. Fujimoto, S. Obata, and A. Kawamura, "Robust biped walking with active interaction control between foot and ground," in *Robotics and Automation, 1998. Proceedings. 1998 IEEE International Conference on*, vol. 3. Ieee, 1998, pp. 2030–2035.

[9] M. Fischler and R. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[10] N. G. Tsagarakis, Z. Li, J. A. Saglia, and D. G. Caldwell, "The design of the lower body of the compliant humanoid robot cCub," in *Proc. IEEE Intl Conf. on Robotics and Automation (ICRA)*, Shangai, China, May 2011, pp. 2035–2040.

[11] B. Ugurlu, N. G. Tsagarakis, E. Spyrakos-Papastravridis, and D. G. Caldwell, "Compiant joint modification and real-time dynamic walking implementation on bipedal robot cCub," in *IEEE Intl. Conf. on Mechatronics*, 2011.

[12] N. G. Tsagarakis, M. Laffranchi, B. Vanderborght, and D. G. Caldwell, "A compact soft actuator unit for small scale human friendly robots," in *Proc. IEEE Intl Conf. on Robotics and Automation (ICRA)*, Kobe, Japan, May 2003, pp. 4356–4362.

[13] P. Kormushev, B. Ugurlu, S. Calinon, N. G. Tsagarakis, and D. G. Caldwell, "Bipedal walking energy minimization by reinforcement learning with evolving policy parameterization," in *Proc. IEEE/RSJ Intl Conf. on Intelligent Robots and Systems (IROS)*, San Francisco, USA, September 2011, pp. 318–324.