A Platform for Building New Human-Computer Interface Systems that Support Online Automatic Recognition of Audio-Gestural Commands*

Nikolaos Kardaris
nick.kardaris@gmail.com

Isidoros Rodomagoulakis
irodoma@cs.ntua.gr

Vassilis Pitsikalis
vpitsik@cs.ntua.gr

Antonis Arvanitakis
antonisar@gmail.com

Petros Maragos
maragos@cs.ntua.gr

School of Electrical and Computer Engineering
National Technical University of Athens, 15773 Greece

ABSTRACT

We introduce a new framework to build human-computer interfaces that provide online automatic audio-gestural command recognition. The overall system allows the construction of a multimodal interface that recognizes user input expressed naturally as audio commands and manual gestures, captured by sensors such as Kinect. It includes a component for acquiring multimodal user data which is used as input to a module responsible for training audio-gestural models. These models are employed by the automatic recognition component, which supports online recognition of audio-visual modalities. The overall framework is exemplified by a working system use case. This demonstrates the potential of the overall software platform, which can be employed to build other new human-computer interaction systems. Moreover, users may populate libraries of models and/or data that can be shared in the network. In this way users may reuse or extend existing systems.

1. INTRODUCTION

Gestures and speech are the natural way, we humans, communicate. Inspired by this, research in human-computer interaction (HCI) and interfaces is continuously evolving to incorporate natural human communication features, which are more intuitive, perceptive and familiar [16]. Interfaces may incorporate audio/visual information via gesture, speech recognition and natural language processing, as well as physical information and haptics [13]. At the same time scientific and technological advances in speech, vision and machine learning along with sensors such as Kinect, natural interfaces such as Apple's Siri and datasets of challenging recognition tasks e.g. [23, 10, 12], signify the emerging trend of machine learning applied on natural multimodal HCI. Apart from commercial systems, we observe a steady effort since [18, 24] e.g. a crisis management system of [8], that continues to our days [15, 20, 14, 25] e.g. on natural gesture interaction [27], and multimodal recognition challenges [9, 11, 17].

Next, we present not only a software for multimodal automatic recognition. We rather introduce a complete framework for constructing new human-computer interfaces from scratch, that work online and support automatic speech/gesture recognition. To our knowledge, there is no such platform, that allows building complete task-specific multimodal recognition systems with natural interaction. Moreover we also prompt the users to upload, share and reuse the building blocks and developed models. The core software component (Sec. 2.3) performs automatic recognition of audio phrases and hand gestures from video/audio streams coming from common or depth cameras (e.g. Kinect) and microphones. It fuses recognition results from the two modalities to enhance performance. Audio-visual processing is carried out in an online fashion with real-time performance. This can be employed for multimodal audio-gestural recognition out of the box, since we provide pre-trained models for a specific task, as a use case. Particularly, our use case concerns human-robot interaction via a set of commands that are mapped to audio phrases and gestures. In addition, we provide a pipeline of components whereby users may construct new multimodal recognition systems. It includes software components for data acquisition and audio-visual training (Sec. 2.1, 2.2). The newly constructed interface system may contain new sets of concepts based on another dataset, and may employ alternative sub-components to the ones we already include; e.g. one may incorporate a different feature extraction method. Moreover, we present an actual working example. For this, we briefly show results in two alternative datasets that correspond to a specific task (Sec. 3) on which we have built our use case, evaluating the system's performance. In previous work [22, 21, 17, 19] one may find details on the datasets, acquisition and the related scientific concepts.

*For further information and source code visit `http://robotics.ntua.gr/software/building-interfaces-multimodal-interaction`.

MM '16, October 15–19, 2016, Amsterdam, Netherlands

DOI: http://dx.doi.org/10.1145/2964284.2973794

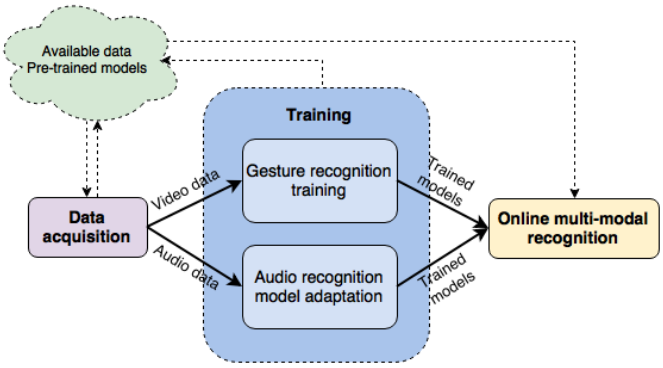

Figure 1: Conceptual block diagram of our overall framework. The three components included are: audio-visual data acquisition, training and online recognition of audio-gestural commands.

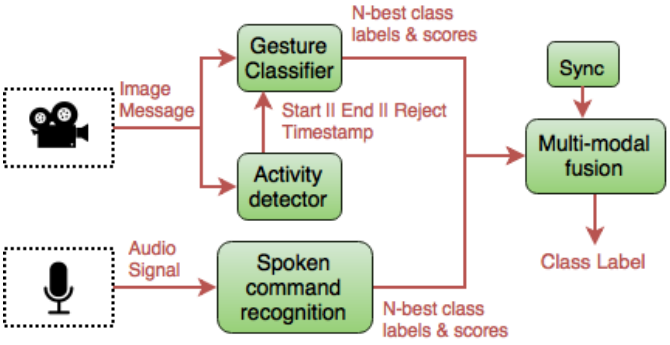

Figure 2: Multimodal command recognition system overview.

2. OVERALL SYSTEM DESCRIPTION

The overall framework is based on the *Robot Operating System* (ROS) and comprises three individual components depicted in Fig. 1, that allow users to implement their own audio-gestural interface. The data acquisition system (Sec. 2.1) assists and guides task-specific audio and video recordings. The acquired data may be employed as input to the training component (Sec. 2.2) to build audio and visual gesture models. Trained models can be used for online multi-modal recognition (Sec. 2.3). Recognition output may be directly used or further processed in multiple ways according to the developer's needs. The overall pipeline provides the software tools to develop custom user interfaces that enable them to interact with a system in new natural ways.

2.1 Multimodal data acquisition component

The data acquisition component provides the functionality of acquiring audio/video data for training and developing task-specific models and recognizers. The component has a remotely controlled GUI that assists a) the supervisor to guide the subject and annotate on-the-fly the temporal boundaries of the recorded audio-gestural commands and b) the subject to perform the audio/gestural commands by appropriate video prompts shown on the screen. The web interface allows parameterisation of the recordings and data organisation. The recorded data are streamed from the sensors to specific ROS data streams (topics) and finally stored as ROS-bag files along with the accompanying annotations. Both the annotation file format and the input data streams are customizable.

2.2 Multimodal training components

2.2.1 *Training for audio commands' recognition*

This component implements acoustic model adaptation leveraging off-the-shelf tools such as HTK [28, 2] allowing the developer to use other speech processing and modeling tools as well. The basic function is to adapt a set of pre-trained subword HMMs to the conditions of the targeted environment and potentially to the voice of a specific user when building systems with user profiles. The developer can find publicly available training recipes and resources [7] for a variety of languages and recognition engines. Adaptation can be realized on a small dataset of transcribed utterances recorded in the targeted environment which can be easily acquired by using the proposed interface. The recorded ROS-bags and annotations are converted to HTK format and then provided to the HTK binary "HERest" for MLLR adaptation.

2.2.2 *Training for gesture commands' recognition*

The system for training on the visual gesture data follows a pipeline often employed in the field of action recognition. The main steps include feature extraction, feature encoding and Support Vector Machine (SVM) training. We employ the popular Dense Trajectory features with the MBH descriptor [26]. Features are encoded with Bag-of-Visual-Words using a visual codebook of $K = 4000$ centres constructed with K-means. Encoded features are used to train one-versus-all SVM classifiers with the χ^2 kernel. The output of the training phase includes the codebook, encoded features, a kernel normalization factor and SVM models and is used at runtime for testing.

2.3 Online recognition component

2.3.1 *Interprocess communication*

The on-line recognition system relies on ROS to provide the main software layer for interprocess communication. Components are implemented as *ROS nodes*; communication, synchronization and data exchange is accomplished by *message passing*. In addition, input data from sensors are acquired at runtime as ROS message streams.

Our overall system comprises two separate sub-systems: (a) The spoken command recognizer, is a single node (*SPK node*) that performs always-listening speech recognition of specific user-defined command phrases that accompany the gestures. (b) The gesture recognizer consists of two nodes: the activity detector (*AD node*) that performs temporal localization of segments that contain visual activity and the gesture classifier (*GC node*) that assigns each detected activity segment to one of the pre-defined classes. The output from both sub-systems are combined at the *Fusion node* producing a single result. These interconnections between the ROS nodes are illustrated in Fig. 2. Synchronization issues are handled by an auxiliary node, which receives the final recognized command. All nodes are automatically launched by a single ROS launch file (Listing 1).

```
<launch>
   <!-- launch Kinect -->
  <include file="$(find openni_launch)/launch/
       openni.launch" />
  <!-- load configuration -->
  <rosparam command="load" file="$(find cvsp_hri)/
       config/gestures.yaml" />
  <rosparam command="load" file="$(find cvsp_hri)/
       config/fusion.yaml" />
  <!-- launch ROS nodes-->
  <node name="GC_node" pkg="cvsp_hri" type="
       DenseTrack" output="screen" required="true"/
       >
  <node name="AC_node" pkg="cvsp_hri" type="
       activity_detection.py" output="screen"
       required="true"/>
  <node name="audio_feedback" pkg="cvsp_hri" type=
       "soundplay.py" output="screen" required="
       true"/>
  <node name="SPK_node" pkg="cvsp_hri" type="scr.
       py" output="screen" required="true"/>
  <node name="sync" pkg="cvsp_hri" type="clock.py"
        output="screen" required="true"/>
  <node name="fusion_node" pkg="cvsp_hri" type="
       audioGestCmdRec.py" output="screen" required
       ="true"/>
</launch>
```

Listing 1: ROS launch file used to start separate nodes of the online multimodal recognition system.

2.3.2 *Audio commands' recognition*

The SPK node is designed to detect and recognize the commands provided by the user freely, at any time, among other speech and non-speech events possibly affected by environmental noise and reverberation. We enhance robustness through a) denoising of the far-field signals, b) adaptation of the acoustic models, and c) combined command detection/recognition by performing traditional ASR in overlapping windows with rejection mechanisms of non-speech and generic speech segments.

As depicted in Fig. 3, a sliding window of 2.5 *sec* duration slides every 0.6 *sec* in which we enforce recognition of one of the pre-defined command sentences against other phrases of various lengths included in the employed finite state grammar to catch any speech or non-speech cases different than the expected commands. Moreover, rejection of background events is feasible by thresholding their low-likelihoods. For robustness, the node outputs a recognition result only in case the same result is found in two successive overlapping windows.

The node is implemented as a multithreaded Python class with separate threads dedicated to recording and recognition. The recording thread streams multichannel audio from the audio card to an allocated buffer in memory where the signals are accessed,segmented and processed by the recognition thread. Processing includes delay-and-sum beamforming for denoising with the implemented *dsb* class and extraction of standard MFCCs-plus-derivatives features utilizing HTK's "HCopy". Finally, N-Best, grammar-based recognition is realized with "HVite".

2.3.3 *Gesture commands' recognition*

The AD node processes the RGB stream in a frame-by-frame basis and determines whether there is any visual activity in the scene, based on an activity "score", whose value is thresholded. When a transition from non-activity (activity) to activity (non-activity) is detected, a `START` (`END`) signal is sent to the GC node, along with the timestamp

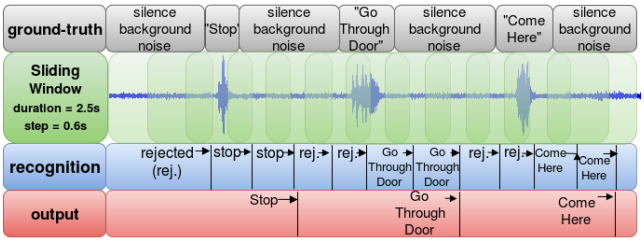

Figure 3: SPK node timeline: Sliding window based spoken command processing and recognition.

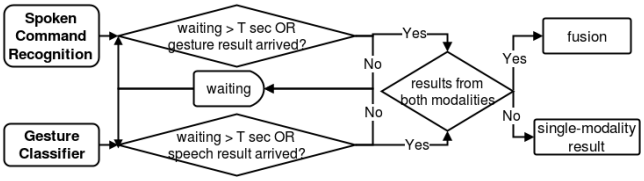

Figure 4: Fusion node: Handling and synchronizing outputs from the individual recognizers. If outputs from both modalities are available they are fused to form a single recognition result.

of the corresponding frame. Small segments of activity are rejected (a `REJECT` signal instead of an `END` signal is sent to the GC node) to ensure that small spontaneous movements of the user are not processed.

The Gesture Classifier node processes video segments and assigns them to one of the pre-defined categories. The classification pipeline includes feature extraction, feature encoding and SVM classification. The segments' temporal boundaries are indicated by the AD node as described previously. In particular, the GC node caches input frames from the camera. When a `START` signal is received, feature extraction begins immediately, starting from the indicated timestamp, until an `END` signal is received. Subsequently, features are extracted from the remaining frames of the activity segment, the classification pipeline continues and the result is broadcast to the Fusion node.

2.3.4 *Multimodal post-processing and fusion*

We combine the individual multi-class results using a late fusion scheme that encodes inter-modality agreement described simply with the following ranking rule: "if the best speech recognition hypothesis is among the 2-best gesture hypotheses, then it is announced as the multi-modal result". We found it quite effective and fast among other approaches like the N-best hypothesis rescoring we have proposed in [22]. Nevertheless, the developer is allowed to incorporate any other rules and fusion schemes.

The implemented Fusion node of Fig. 4 receives the N-best results announced by the individual recognizers. It checks for available results every 0.5 *secs* by receiving periodically messages from the clock node Sync in order to synchronize the recognizers. A waiting period T is also defined, during which the node waits to combine the incoming messages. If this period expires, it is assumed that one modality either was not activated by the user, or failed to detect the given input. In such cases, the node announces single-modality results.

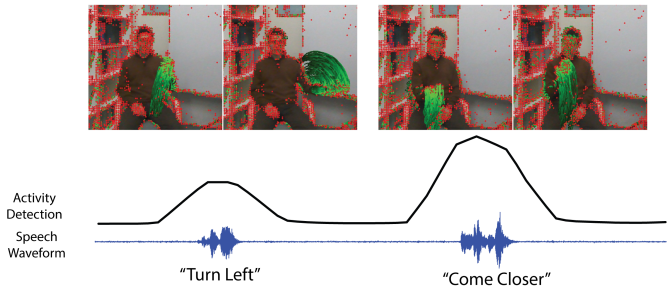

Figure 5: Multimodal command recognition example. The user gestures and utters audio commands. Activity segments are detected by the AD node based on an "activity" score (second row). These are processed by the GC node (top row; dense trajectories superimposed to RGB frames). The third row depicts the respective audio waveforms.

3. CASE STUDY: AN HRI SYSTEM

Based on the above described software platform, we provide a specific audio-gestural interface. This is a working human-robot interaction (HRI) system developed with the presented platform in the context of the MOBOT project [4]. This application concerns elderly people communicating with a robotic platform in order to get assistance. Our audio-gestural vocabulary includes 19 verbal commands along with the corresponding gestures for concepts like "Come here", "Help me", "I want to stand up" etc. The robot reacts either providing audio responses or moving in order to assist the elderly user. The system was trained on the data acquired in [21] using the acquisition system described briefly in Sec. 2.1. To reduce the total processing time we downsample RGB frames both in space and time by a factor of 2. This results to an average processing rate of 20 fps. The SPK node supports close to real time spoken command recognition in German, English, and Greek. Figure 5 illustrates visual processing aspects of online recognition.

Our online recognition system is tested on MOBOT-6a task [22] and is currently being evaluated on the multimodal gesture dataset acquired in [21]. All experiments are carried out with the leave-one-out scheme, i.e. testing on one subject, while the rest of the subjects are used for training; this is repeated for all subjects. We report average recognition accuracy across all subjects.

The MOBOT-6a includes 8 different gestural and German verbal commands performed by 8 elderly subjects. These are developed to accommodate their communication with the robotic platform. Patients are sitting in front of the rollator, placed at a distance of 2.5 meters. Each command is performed $3-6$ times by each patient. We obtained 84.13%, 56.98%, and 90.15% accuracy for the audio, visual modality, and their fusion respectively. The dataset described in [21] includes 13 subjects performing the 19 gestural and Greek audio commands $4-5$ times under different settings, e.g. sitting, standing, with viewpoint change. Preliminary evaluation yields 66.15% and 82.46% accuracy for the audio and visual modality respectively. Note that the designed Greek verbal commands were not as discriminative as the German ones, yielding lower results.

3.1 Practical issues and dependencies

Other employed software: Our training and recognition components have been developed mainly in Python and C. In addidion, we use publicly available 3rd party software. Specifically, spoken command recognition is based on the HTK 3.4 toolkit [2] which can be downloaded upon registration. Regarding the gesture recognizer: a) dense trajectories [26] are extracted using an custom version of the publicly available code [1], b) optical flow for activity detection is estimated by using the openCV library [5], c) the VLFeat library [6] is utilized for encoding and other operations and finally d) SVM training and classification is based on LIBSVM [3].

Sensors: We have successfully tested our system with a linear 8-channel MEMS array and a Kinect camera, though conventional microphones and cameras supported by Linux have been tested and can be used as well. Note that distant speech recognition performance depends on the quality of the audio signals and our system takes advantage of the configuration of a microphone array's channels internally. Nevertheless, the user is free to use any external or built-in microphone.

3.2 Building new interfaces and models

Apart from reusing the provided models to perform online audio-gestural command recognition and mapping the recognition results to specific user or system defined actions, the overall framework allows the design and implementation of various multimodal audio-gestural automatic recognition interfaces. This is accomplished by setting basic parameters such as the task, the concepts, and other configuration variables or metadata information. Guided by the acquisition software one may record data with multiple subjects performing gestures and uttering audio prompts according to the task definition. The training pipeline is responsible to produce the appropriate models using the acquired data, or reusing models which can be found on the web. Finally, given the available or new models built from scratch, the online system is ready to be used.

4. CONCLUSIONS

We presented a complete software platform for the development of multimodal interface systems that offer audio-gestural commands' automatic recognition. We also briefly describe a specific system that has been constructed as a use case, tested and employed for human-robot interaction. Moreover, we have provided experimental evidence on the performance of the employed methods. These are only instances related to the current use case and users may employ other algorithms for the included components. Considering the emergent fields of human-computer interaction and multimodal human action, gesture, speech recognition we think that the introduced software platform, as well as the sharing and reusability of data/models opens new perspectives for human-computer natural interaction.

5. ACKNOWLEDGMENTS

This research work was supported by the EU under the projects MOBOT with grant FP7-ICT-2011.2.1-600796 and I-SUPPORT with grant H2020-643666.

6. REFERENCES

[1] Dense trajectories. https://lear.inrialpes.fr/people/wang/dense_trajectories. Accessed: 20/5/2016.

[2] Htk. http://htk.eng.cam.ac.uk/. Accessed: 20/5/2016.

[3] libsvm. https://www.csie.ntu.edu.tw/ cjlin/libsvm/. Accessed: 20/5/2016.

[4] MOBOT project. http://www.mobot-project.eu/.

[5] Open cv. http://opencv.org/. Accessed: 20/5/2016.

[6] Vlfeat. http://www.vlfeat.org/. Accessed: 20/5/2016.

[7] Voxforge. http://www.voxforge.org/. Accessed: 20/5/2016.

[8] P. Agrawal, I. Rauschert, K. Inochanon, L. Bolelli, S. Fuhrmann, I. Brewer, G. Cai, A. MacEachren, and R. Sharma. Multimodal interface platform for geographical information systems (geomip) in crisis management. In *Proceedings of the 6th international conference on Multimodal interfaces*, pages 339–340. ACM, 2004.

[9] I. Bayer and T. Silbermann. A multi modal approach to gesture recognition from audio and video data. In *Proceedings of the 15th ACM on International conference on multimodal interaction*, pages 461–466. ACM, 2013.

[10] C. Chen, R. Jafari, and N. Kehtarnavaz. Utd-mhad: a multimodal dataset for human action recognition utilizing a depth camera and a wearable inertial sensor. In *Image Processing (ICIP), 2015 IEEE International Conference on*, pages 168–172. IEEE, 2015.

[11] S. Escalera, J. Gonzàlez, X. Baró, M. Reyes, I. Guyon, V. Athitsos, H. Escalante, L. Sigal, A. Argyros, C. Sminchisescu, R. Bowden, and S. Sclaroff. Chalearn multi-modal gesture recognition 2013: grand challenge and workshop summary. ACM, 2013.

[12] S. Escalera, J. Gonzàlez, X. Baró, M. Reyes, O. Lopes, I. Guyon, V. Athitsos, and H. Escalante. Multi-modal gesture recognition challenge 2013: Dataset and results. In *Proc. of the 15th ACM on Int'l Conf. on Multimodal Interaction*, pages 445–452. ACM, 2013.

[13] M. A. Goodrich and A. C. Schultz. Human-robot interaction: a survey. *Foundations and trends in human-computer interaction*, 1(3):203–275, 2007.

[14] F. Grani, D. Overholt, C. Erkut, S. Gelineck, G. Triantafyllidis, R. Nordahl, and S. Serafin. Spatial sound and multimodal interaction in immersive environments. In *Proceedings of the Audio Mostly 2015 on Interaction With Sound*, page 17. ACM, 2015.

[15] J. Hare, M. Karam, P. Lewis, et al. igesture: A platform for investigating multimodal, multimedia gesture-based interactions. 2005.

[16] A. Jaimes and N. Sebe. Multimodal human–computer interaction: A survey. *Computer vision and image understanding*, 108(1):116–134, 2007.

[17] G. Pavlakos, S. Theodorakis, V. Pitsikalis, S. Katsamanis, and P. Maragos. Kinect-based multimodal gesture recognition using a two-pass fusion scheme. pages 1495–1499, 2014.

[18] A. Pirhonen, S. Brewster, and C. Holguin. Gestural and audio metaphors as a means of control for mobile devices. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 291–298. ACM, 2002.

[19] V. Pitsikalis, S. Katsamanis, S. Theodorakis, and P. Maragos. Multimodal gesture recognition via multiple hypotheses rescoring. 16:255–284, 2015.

[20] O. Portillo-Rodriguez, O. O. Sandoval-Gonzalez, E. Ruffaldi, R. Leonardi, C. A. Avizzano, and M. Bergamasco. Real-time gesture recognition, evaluation and feed-forward correction of a multimodal tai-chi platform. In *Haptic and Audio Interaction Design*, pages 30–39. Springer, 2008.

[21] I. Rodomagoulakis, N. Kardaris, V. Pitsikalis, A. Arvanitakis, and P. Maragos. A multimedia gesture dataset for human-robot communication: Acquisition, tools and recognition results. In *IEEE International Conference on Image Processing (ICIP-16), to appear*, sep 2016.

[22] I. Rodomagoulakis, N. Kardaris, V. Pitsikalis, E. Mavroudi, A. Katsamanis, A. Tsiami, and P. Maragos. Multimodal human action recognition in assistive human-robot interaction. In *Proc. IEEE Conference on Acoustics, Speech and Signal Processing (ICASSP-16)*, mar 2016.

[23] S. Ruffieux, D. Lalanne, E. Mugellini, and O. A. Khaled. A survey of datasets for human gesture recognition. In *Human-Computer Interaction. Advanced Interaction Modalities and Techniques*, pages 337–348. Springer, 2014.

[24] R. Stiefelhagen, C. Fügen, P. Gieselmann, H. Holzapfel, K. Nickel, and A. Waibel. Natural human-robot interaction using speech, head pose and gestures. In *Intelligent Robots and Systems, 2004.(IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on*, volume 3, pages 2422–2427. IEEE, 2004.

[25] H. A. Vasquez, H. S. Vargas, and L. E. Sucar. Using gestures to interact with a service robot using kinect 2. *Advances in Pattern Recognition*, page 85, 2015.

[26] H. Wang, A. Kläser, C. Schmid, and C. L. Liu. Action recognition by dense trajectories. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2011)*, pages 3169–3176, June 2011.

[27] Y. Yin and R. W. Davis. Real-time continuous gesture recognition for natural human-computer interaction. In *Visual Languages and Human-Centric Computing (VL/HCC), 2014 IEEE Symposium on*, pages 113–120. IEEE, 2014.

[28] S. J. Young et al. *The HTK Book, version 3.4*. Cambridge University Engineering Department, 2006.