Enriching Robot's Actions with Affective Movements

Julian M. Angel-Fernandez[1] and Andrea Bonarini[2]

Abstract— **Emotions are considered by many researches as a characteristic that could be beneficial in social robotics, since they enrich human-robot interaction with non-verbal clues. Although there have been works that have studied emotion expression in robots, mechanism to generate emotion are highly integrated with the rest of the system. This unable the possibility to use their approaches in different applications. This paper present a system that has been initially created for a theatrical robot to enrich with emotions its actions, but it has been designed to enable the possibility to be adapted to be used in other fields. The emotional enrichment system has been envisioned to be used with a action decision system. A formal description of the system is provided to be used as a formal reference for further extensions and possible modifications. The system has been adapted to two different platforms with different grades of freedom: Keepon and Triskarino.**

I. INTRODUCTION

The development of fast, cheap, and reliable electronics has enabled the creation of new devices and versatile robotic platforms. These new platforms' capabilities have expanded the frontiers of the robots applications to new environments where robot are expected to interact with humans, such as health care, and house cleaning, among others. However, bringing robots in these environment raises the challenge to increase robots' acceptance. Although this could be seen as an easy task that just would need improvements in robots' appearances and capabilities, it is possible that people would expect to treat robots as humans has as they do with computers. [1], which makes necessary the creation of robots that fulfil this expectations.

Some researchers have suggested that embedding emotion expression capabilities to robots could improve their acceptance in social environments [2]. As consequence researchers [3], [4] have added specific emotional poses and expression to their robots. Others have studied how to convey emotions with specific platforms [5], [6]. Nevertheless, these works have created modules to show emotions that are strongly integrated to their solutions, which eliminate the possibility to re-use or adapt their systems into other projects.

In theory, the projection of emotion with humanoid embodiments could be simplified to mimic the same movements that humans. However, this idea is not possible due robots physical limitations. Therefore, an exact matching of humans movements to convey emotions could not be used in robots [7], [8]. Therefore diverse researchers are studying diverse features and values to express emotions with different platforms. As a consequence, these results could not be widely used due to the differences of the platforms.

This paper presents an Emotional Enrichment System (EES), which modify actions' parameters and add additional actions to create the illusion of emotion expression in a robot. Although the EES was originally conceived to be used in an autonomous performance robot [9] to enrich actions with emotions, its design was devised to make it extendable to other platforms and adaptable to new tasks. To achieve this goal, the system relies on an Emotional Execution Tree (EXT), which is based on simple actions, sequential and parallel nodes. Additionally, it is used the concept of compound actions to group a bunch of nodes, which reduces the tree dimension and allows the reuse of recurrent actions generated by specific combinations of simple actions and other nodes. This EXT has been formalized to give a guideline to further implementations and extensions.

The rest of the paper is organized as follows. Section **??** provides a brief overview of particularly relevant work related to our system. Section **??** gives a brief explanation on TheatreBot architecture. Section **??** gives a general introduction to the system ideas. Section **??** gives the basic formalization of our system and principal components terms used on it. Section **??** describes the implementation of the system and shows two demonstrations done with the system using platforms with different capabilities.

II. RELATED WORK

The use of emotion enrichment to improve human robot interaction is not new trend. There have been several works that have enhance their social robots with emotions or study how to convey them in robotic platforms [5], [6]. One of the most well-known expressive robots is Kismet [3], a robotic face able to interact with people and show emotions. This platform uses a specific set of movements based on the Ekman's studies on human emotion expression [10]. Other approaches have tried to use anthropomorphic [4] and human-like platforms to convey emotions to study the response of people towards the robot. However, the emotions portrayed were hand-coded, hard-wired to the respective platforms, and their parametrization was not available.

On the other hand, studies focused on entertainment robotics have tried to introduce emotional actions to improve the audience's experience. Breazeal and collaborators [11] used one robot on the stage. This anemone-like robot had few behaviours, which included getting scared when a person comes too close; the robot was able to show some basic emotions (i.e., fear and interest). Knight [12], [13] used the

[1]Julian M. Angel-Fernandez is a Research at Automation and Control Institute, Vienna University of Technology, Vienna, Austria `julian.angel.fernandez@tuwien.ac.at`

[2]Andrea Bonarini is a Professor at the Department of Electronics, Information, and Bioengineering, Politecnico di Milano, Milan, Italy, `andrea.bonarini@polimi.it`

platform NAO to produce a sort of stand-up comedy. The robot performs basic actions to add some expressiveness to the joke, but it is not intended to project any emotion. Trying to add some theatrical realism, Breazeal and collaborators [14] designed and implemented a system to control a lamp. The main characteristic of this lamp is that it could be controlled by just one person, which selected pre-coded emotions.

Other works in performance robotics have developed systems that do not convey any kind of emotion as *Roboscopie* [15], [16], Fan and collaborators [17], [18], and adaptation of Shakespeare's Midsummer Night's Dream [19] uses robots in performances with real actors but without any emotion expression.
Although these works use emotions, their main focus was the interpretation of postures or just the use of emotions to increase their robot appealing, but none of them considered the importance of emotional system that could be used by others. As a consequence, most of these works have created emotional systems that could just work in their specific system.

III. Basic Concepts

The system is based on six main concepts: *simple actions*, *compound actions*, *action message*, *emotional descriptors*, *character description* and *emotional execution tree*. *Action message* establishes the structure of the message to describe any kind of action (i.e., simple and compound). This message also specifies how the actions are executed (i.e., in parallel or in sequence) and which action is predominant (i.e., primary or secondary). *Emotional parameters* describe how the emotional enrichment should be done to convey a specific emotion in a specific simple action. This description could also include addition of other simple actions and vary over time. *Character description* enables the possibility to establish how to modify emotional expressions to generate diverse treats. Finally, *Emotional Execution Tree* is a computational representation of desired actions that should be executed. This tree is first created from the action message description and then modified using the emotional parameters and character description.

A. Simple and Compound Actions

In order to generate a system that could be used in diverse platforms require an abstraction level in which all of them could fall. Therefore simple and compound actions are used to achieve this goal. Simple actions are actions that are considered as primitives: they are used as building blocks. Therefore, these actions are described in the system and are the ones in which the emotional enrichment takes place. Their description specifies mandatory and optional parameters that are required to execute an action. Compound actions are actions that are created from simple actions. These actions are not implemented in the system, but, if it is needed, they can be described in it (e.g. compound actions that are used often).

TABLE I: Description of the seven simple actions implemented, and their respective parameters. Where P is 2D position, V is 2D velocity vector and angular velocity, and T is time

Action Name	Description	Parameter(s)
Do nothing	It waits for a time t before it is terminated. It could be seen as a delay.	T
Move body	It moves the platform from its current position a to a desired position b.	P, and V
Oscillate body	It generates an oscillation in the whole platform by an angle θ.	θ and V
Move shoulder	It moves the shoulders to a desired angle θ. It is considered as angular movement.	θ and V
Oscillate shoulder	It oscillates the shoulders by a given angle θ	θ and V
Move torso	It moves the torso to a desired angle in yaw, $pitch$ and $roll$	yaw, $pithc$, $roll$ and V
Oscillate torso	It oscillates the shoulders by a given angle θ	θ and V

The simple actions to be implemented to test the system were selected by considering platforms' capabilities and the requirements. The eight actions selected are: move body, oscillate body, move shoulder, oscillate shoulder, move torso, oscillate torso, and do nothing. Description for each action, its mandatory parameters and optional parameters are shown in Table I.

IV. Describing Components

The description files allow the parametrization of the system and its adaptation to different circumstances. The system has the following parametric data:

- *Emotion description* gives information of the parameters that should be changed in all the simple actions to express the desired emotion. Therefore, for each combination emotion-action should be given a description of how the parameters are defined and how they change in time. The first version of this parameters is done for movement. Therefore, it was considered four descriptors for this type of parameter. The first is the reference, which inform the system the velocity used to calculate the parameter. This reference is used to let the modulation of the parameters depending on the character. The second descriptor is the tuple space and time. with this tuple is possible to calculate the velocity in which the movement must be done in order to convey the desire emotion. in order to enable the possibility to have different velocities during the execution, it could be described a sequence of this tuple. in this case, the space descriptor plays an important role, which is to inform the length in which the desired velocity must be used. This bring the last description which is the repetition that is used if there is more than one tuple and it should be repeat during the duration of the movement. The EBFN to described these descriptors in the system is the following:

⟨emotion description⟩⊨ **'{''emotion:'**⟨string⟩ **',''observation:'**⟨string⟩ **','**⟨action⟩**'(,'**⟨action⟩)* **'}'**
⟨action⟩⊨⟨string⟩**':{'**⟨description⟩ **','**⟨actions affected⟩ **'}'**
⟨description⟩⊨**'description:{'**⟨action name⟩**','**⟨emotion⟩**','**
⟨parameter's type⟩**'}'**
⟨action name⟩⊨**'emotionProfileAction:'**⟨string⟩
⟨emotion⟩⊨**'emotionProfileEmotion:'**⟨string⟩
⟨parameter's type⟩⊨ **'movement_parameter'**
⟨actions affected⟩⊨**'actions:{'**⟨action parameter⟩ (**','**⟨action parameter⟩)***'}'**
⟨action parameter⟩⊨ ⟨string⟩**':{'**⟨reference⟩**','** ⟨repetition⟩**','**⟨parameters⟩**'}'**
⟨reference⟩⊨ **'reference:'**⟨number⟩
⟨repetition⟩⊨ **'repetition:''yes'**|**'no'**
⟨parameters⟩⊨ **'parameters:['** ⟨parameter description⟩
(**','**⟨parameter description⟩)* **']'**
⟨parameter description⟩⊨ ⟨movement parameter description⟩|⟨new parameters⟩
⟨movement parameter description⟩⊨**'{time:'**⟨number⟩**',space:'** ⟨number⟩**'}'**

- *Character's emotions* parameters give the system information about character's "rhythm" for each pair action-emotion. To describe this changes, it was decided to used bias, amplitude and long as descriptors. Bias refers to a constant value that is added to all the sequence of emotional parameters. Amplitude amplify the velocity respect the bias if the number is bigger than one, or attenuate if it is between zero and one.
- *Action message* contains all the necessary information for to execute an action. The following is the EBNF of an action:

⟨action⟩⊨ ⟨simple action⟩|⟨compound action⟩|⟨context⟩
⟨simple action⟩⊨ **'{'** ⟨action header⟩ **',''parameters:['** ⟨simple action parameters⟩ **']}'**
⟨compound action⟩⊨ **'{'** ⟨action header⟩ **',''parameters:['** ⟨compound action parameters⟩ **']}'**
⟨action header⟩⊨ **'type:'** ⟨action type⟩ **',''name: '** ⟨string⟩ **','** ⟨is primary⟩
⟨simple action parameters⟩⊨ **'{'**⟨parameter header⟩ **','** ⟨parameter description⟩ **'}'**
⟨compound action parameters⟩⊨ ⟨simple action parameters⟩ (**','** ⟨simple action parameters⟩)*
⟨context⟩⊨**'{'** ⟨action type⟩ **','** ⟨emotion sync⟩ **','** ⟨action sync⟩ **','** ⟨is primary⟩ **','**
'information:' ⟨string⟩ **','** **'actions: '** ⟨action⟩ **'}'**
⟨parameter header⟩⊨ **'type:'** ⟨parameter type⟩ **','** **'name:'**⟨string⟩
⟨parameter description⟩⊨ ⟨parameter amplitude⟩|⟨parameter circle⟩|⟨parameter landmark⟩|
⟨parameter point⟩|⟨parameter speech⟩|
⟨parameter square⟩|⟨parameter time⟩
⟨parameter type⟩⊨ **'mandatory_parameter'**|**'optional_parameter'**
⟨is primary⟩⊨ **'yes'**|**'no'**
⟨emotion sync⟩⊨ **'yes'**|**'no'**
⟨action type⟩⊨ **'parallel_context'**|**'serial_context'**|**'simple_action'**|
'composite_action'

V. EMOTIONAL TREE

Emotional Execution Tree is a connected acyclic graph $G(V, E)$ with $|V|$ vertexes and $|E|$ edges. The root and non-leaf nodes could be of either *parallel* or *sequential* type. Sequential nodes execute one branch after the other. Once all the branches have been executed, it notifies to its predecessor to continue with the execution of other nodes. The parallel node offers two type of synchronization. one is at action and the other at emotion. This distinction between action and emotion synchronization enables the possibility to change emotion parameter without affecting the action. done could be one out of four different sub-types: (i) action and emotion synchronous, (ii) action synchronous and emotion asynchronous, (iii) action asynchronous and emotion synchronous, or (iv) action and emotion asynchronous. Action synchronous means that each time that a parallel node receives a "finish" notification (i.e., success or failure), it will broadcast the message to all nodes that derived it and to its predecessor, which will stop their execution.

If a sequence node receives a finish message, it will execute the next branch. When all branches have been executed, it communicates the end of the action. On the other hand, emotion synchronous means that each time that a node (either sequence or parallel) receives an emotion synchronization message, it will propagate the message to all branches to move to the consecutive emotional expression. If a node is principal and it has finished to execute all actions, it will notify its predecessor.

This distinction creates the possibility to synchronize emotional changes without affecting the normal execution of an action and it also enables synchronization among parallel actions. Finally, leaf nodes could only be simple action nodes that have been implemented in the system. Any node can belong to one of two levels: principal or secondary. If a node is principal, it will notify its predecessor about the messages that it has received, while the secondary node cannot propagate any message to its predecessor. This enables the possibility to have actions that are the ones to command over the others. Compound actions are implemented combining all type of nodes.

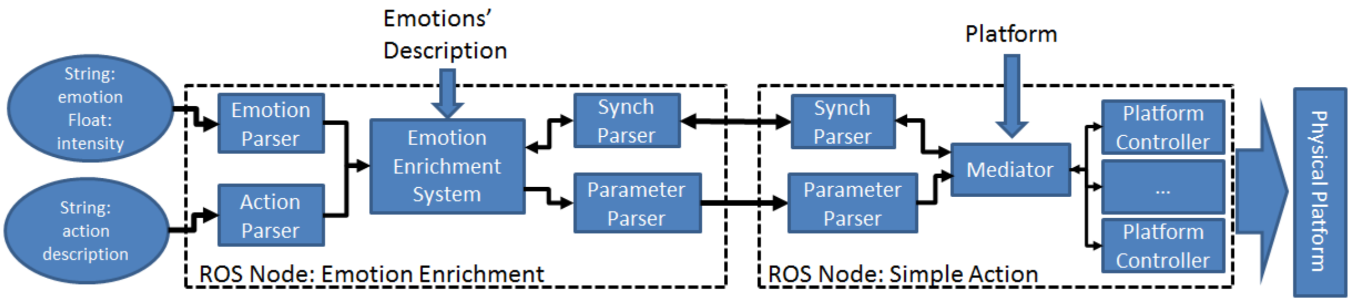


Fig. 1: General system design. Each simple action corresponds to one ROS node, and there is just one node for the emotion enrichment system. The ovals represent the ROS topic parameters, rectangles represent *black boxes*, and texts outside containers represent input files that contain the system parametrization.

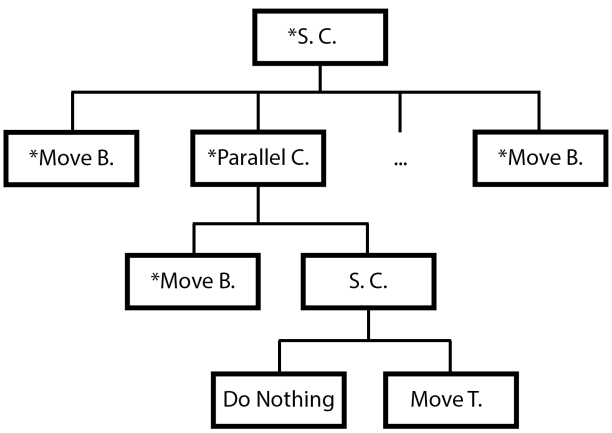


Fig. 2: Emotional Execution Tree for the example shown in the Figure **??**. The context node colored in purple represents a sequential context, while the other represents a parallel.

VI. Implementation

The system was implemented in *C++* with interface with ROS. The design (Fig. 1) was created following the description done in previous sections. The emotion enrichment core is divided in three different modules. Each module is responsible for one of the following phases:

1) *Generation of emotional execution tree:* this phase starts every time that a new action message is received. The process begins by parsing the format, verifying that the actions described on it exist in the system, and that the parameters correspond to the ones expected by each action described on the message. This parameters' verification is done on the implemented description for each action, which describes the parameters that are mandatory and those that are optional. When the verification is done, and all the action exists and the parameters correspond, an ext is created such as the one presented in the Figure 2 that corresponds to the example of walk and speak.
2) *Emotion addition:* uses the ext created in the previous phase. In this phase new sa are added to the ext and the sa's parameters are modified following the emotion description, which is loaded from files. This process is broken down in two steps. First, all the actions that are required to convey the desired emotion, and that are not yet present are added. Second, the emotional parameters are modulated based on the emotion's intensity and character traits.
3) *Execution:* this is the last phase and it is done after the ext is "coloured" with emotional characteristics (actions additions and emotional parameters). The decision to have two different communication channels, one for action parameters and another for the action emotional parameters, was taken to enable the possibility to update the emotional parameters without interfering with the current execution. In this phase is maintain a reference to the mandatory and emotional action, thus when a new emotion is received the system stops the previous emotional action and start executing the new ones without affecting the general execution of the actions.

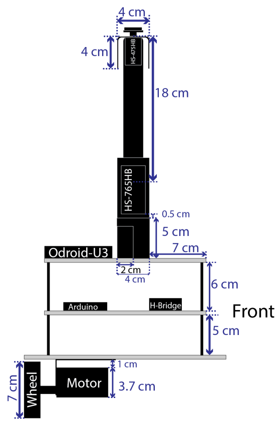
All the text message broadcast among the nodes are written using JSON format, which is human understandable, is light and there are diverse available parser.
To test the system was used to different platforms: Keepon Pro and an own made platform called Triskarino. Keepon was used to test the interoperability of the system to different platforms, while Triskarino has been used as a part of the robot actor project [9]. The whole TheatreBot architecture has not finished just, thus the change of emotion and selection of action to execute were done manually.

A. Keepon Test

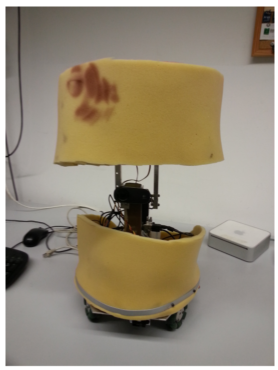
To test the system with Keepon (Fig. 3) was just necessary to implement the platform's controllers in each one of the simple action ROS nodes. Given that this platform does not have the capability to displace its body, the action move body was not implemented. Once added these controllers to the system, we proceeded to modify the configuration files to use this platform instead of Triskarino. With this small modification, we were able to change from one platform to other. In the video[1] is shown the test where Keepon has

[1] YouTube video name Emotional Enrichment System, url: https://youtu.be/bRSXQ0rzkO8

Fig. 3: Keepon platform.



(a) Triskarino design.



(b) Triskarino with its cover.

Fig. 4: Triskarino platform.

to move its torso forward and backward with a "happy" emotion. The action is given by a console telling the robot to bend the torso to a desire angle in x. The torso oscillation in y is added automatically by the system following the description given to happiness.

B. Triskarino Test

The system has been widely used with Triskarino (Fig. 4) during our studies on projecting emotions with a non human like platform. During these studies the system was just used to move the robot in straight line with different emotions, each one selected from our command console. But to show the whole capabilities of the emotional enrichment system and the approach used by us, it was described a little scene that we are preparing. The scene is a modification of the first part of the balcony scene of Shakespeare's Romeo and Juliet play [20]. The simplified sketch of our version could be seen in the Fig. 5. As it can be seen the stage was divided in 81 squares and the positions were given to the system in term on the desire square, which allow the robot to adjust to the stage's dimension. The whole sequence of actions were specified as unique action, having several sequential context and just move body action. The other actions, such oscillate body or blend upper body were added online accordingly the desired emotion.

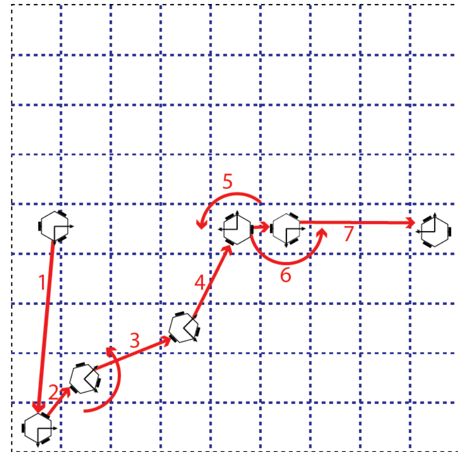


Fig. 5: Sketch of Romeo's movements for the balcony scene in Romeo and Juliet play. The arrows show the direction of the movements, and the numbers their sequence.

VII. Conclusions and Further work

An Emotional Enrichment System has been designed and implemented to enrich robots' movements with emotions. To achieve this, it was used an Emotional Execution Tree created from three different types of nodes: simple actions, parallel, and sequential. Simple actions are functions that map a set of parameters to specific movements. Sequential executes in order the sequence of actions associated to this type of node, while parallel executes them all at the same time. To enable synchronization among simple actions, parallel nodes could be one of four different subtypes, and sequential just one of two different subtypes. A formalization of the Emotional Execution Tree and the principal consideration during the implementation of the system have also been provided. To show the system's versatility, it was used with quite different platforms such as a Keepon Pro and Triskarino. Keepon was used to perform simple actions, while Triskarino was used to test complex actions with different parametrizations of emotions.

The results obtained show that the system could enrich the robotic actions with emotions, which could be parametrized from configuration files. Additionally, the design of the system makes it possible to adopt it for different platforms using the same action description.

References

[1] B. Reeves and C. Nass, *The Media Equation: How People Treat Computers, Television, and New Media Like Real People and Places*. New York, NY, USA: Cambridge University Press, 1996.

[2] A. Paiva, I. Leite, and T. Ribeiro", ""emotion modeling for social robots"." [Online]. Available: "//www.oxfordhandbooks.com/10.1093/oxfordhb/9780199942237.001.0001/oxfordhb-9780199942237-e-029"

[3] C. Breazeal, *Designing Sociable Robots*. Cambridge, MA, USA: MIT Press, 2002.

[4] S. Embgen, M. Luber, C. Becker-Asano, M. Ragni, V. Evers, and K. O. Arras, “Robot-specific social cues in emotional body language,” in *Proceedings of the IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN’12)*. USA: IEEE Computer Society, September 2012, pp. 1019–1025.
[5] J. Li and M. H. Chignell, “Communication of emotion in social robots through simple head and arm movements,” *I. J. Social Robotics*, vol. 3, no. 2, pp. 125–142, 2011.
[6] L. Brown and A. M. Howard, “Gestural behavioral implementation on a humanoid robotic platform for effective social interaction,” in *Robot and Human Interactive Communication, 2014 RO-MAN: The 23rd IEEE International Symposium on Robot and Human Interactive Communication*, Aug 2014, pp. 471–476.
[7] M. Saerbeck and A. J. N. van Breemen, “Design guidelines and tools for creating believable motion for personal robots,” in *RO-MAN*, 2007, pp. 386–391.
[8] A. Beck, A. Hiolle, A. Mazel, and L. Cañamero, “Interpretation of emotional body language displayed by robots,” in *3rd ACM Workshop on Affective Interaction in Natural Environments*, 2010, pp. 37–42.
[9] J. M. Angel F. and A. Bonarini, “Towards an autonomous theatrical robot,” in *ACII’13*, 2013, pp. 689–694.
[10] P. Ekman, *Emotions Revealed : Recognizing Faces and Feelings to Improve Communication and Emotional Life*. Owl Books, Mar. 2004.
[11] C. Breazeal, A. Brooks, J. Gray, M. Hancher, C. Kidd, J. McBean, D. Stiehl, and J. Strickon, “Interactive robot theatre,” in *Proceedings of IEEE/RSJ Int. Conf. on Intelligent Robots and Systems(IROS 2003)*, vol. 4, 2003, pp. 3648–3655.
[12] H. Knight, S. Satkin, V. Ramakrishna, and S. Divvala, “A savvy robot standup comic: Online learning through audience tracking,” TEI 2011, January 2011.
[13] H. Knight, “Heather knight: la comedia de silicio,” TED Ideas Worth Spreading, December 2010. [Online]. Available: http://www.ted.com/talks/heather_knight_silicon_based_comedy.html
[14] G. Hoffman, R. Kubat, and C. Breazeal, “A hybrid control system for puppeteering a live robotic stage actor,” in *RO-MAN 2008*, M. Buss and K. Kühnlenz, Eds. IEEE, 2008, pp. 354–359.
[15] LAAS-CNRS, “Roboscopie, the robot takes the stage!” http://www.openrobots.org/wiki/roboscopie.
[16] S. Lemaignan, M. Gharbi, J. Mainprice, M. Herrb, and R. Alami, “Roboscopie: a theatre performance for a human and a robot,” in *Proceedings of the seventh annual ACM/IEEE international conference on Human-Robot Interaction*, ser. HRI ’12. New York, NY, USA: ACM, 2012, pp. 427–428.
[17] C.-Y. Lin, C.-K. Tseng, W.-C. Teng, W. chen Lee, C.-H. Kuo, H.-Y. Gu, K.-L. Chung, and C.-S. Fahn, “The realization of robot theater: Humanoid robots and theatric performance,” in *International Conference on Advanced Robotics, 2009. ICAR 2009.*, 2009.
[18] C.-Y. Lin, L.-C. Cheng, C.-C. Huang, L.-W. Chuang, W.-C. Teng, C.-H. Kuo, H.-Y. Gu, K.-L. Chung, and C.-S. Fahn, “Versatile humanoid robots for theatrical performances,” *International Journal of Advanced Robotic Systems*, 2013.
[19] R. Murphy, D. Shell, A. Guerin, B. Duncan, B. Fine, K. Pratt, and T. Zourntos, “A midsummer night’s dream (with flying robots),” *Autonomous Robots*, vol. 30, no. 2, pp. 143–156, 2011. [Online]. Available: http://dx.doi.org/10.1007/s10514-010-9210-3
[20] R. S. Company, “Royal shakespeare company - romeo & juliet, on stage footage - ny,” http://www.youtube.com/watch?v=FHoaPLO6Zd8.