Development of a Semi-Autonomous Robotic System to Assist Children with Autism in Developing Visual Perspective Taking Skills

Abolfazl Zaraki, Luke Wood, Ben Robins and Kerstin Dautenhahn

***Abstract*— Robot-assisted therapy has been successfully used to help children with Autism Spectrum Condition (ASC) develop their social skills, but very often with the robot being fully controlled *remotely* by an adult operator. Although this method is reliable and allows the operator to conduct a therapy session in a customised child-centred manner, it increases the cognitive workload on the human operator since it requires them to divide their attention between the robot and the child to ensure that the robot is responding appropriately to the child's behaviour. In addition, a remote-controlled robot is not aware of the information regarding the interaction with children (e.g., body gesture and head pose, proximity etc) and consequently it does not have the ability to shape live HRIs. Further to this, a remote-controlled robot typically does not have the capacity to record this information and additional effort is required to analyse the interaction data. For these reasons, using a remote-controlled robot in robot-assisted therapy may be unsustainable for long-term interactions. To lighten the cognitive burden on the human operator and to provide a consistent therapeutic experience, it is essential to create some degrees of autonomy and enable the robot to perform some autonomous behaviours during interactions with children. Our previous research with the Kaspar robot either implemented a fully autonomous scenario involving pairs of children, which then lacked the often important input of the supervising adult, or, in most of our research, has used a remote control in the hand of the adult or the children to operate the robot. Alternatively, this paper provides an overview of the design and implementation of a robotic system called *Sense-Think-Act* which converts the remote-controlled scenarios of our humanoid robot into a semi-autonomous social agent with the capacity to play games autonomously (under human supervision) with children in the real-world school settings. The developed system has been implemented on the humanoid robot Kaspar and evaluated in a trial with four children with ASC at a local specialist secondary school in the UK where the data of 11 Child-Robot Interactions (CRIs) was collected. The results from this trial demonstrated that the system was successful in providing the robot with appropriate control signals to operate in a semi-autonomous manner without any latency, which supports autonomous CRIs, suggesting that the proposed architecture appears to have promising potential in supporting CRIs for real-world applications.**

I. MOVING BEYOND WOZ IN ROBOT-ASSISTED THERAPY

Many social robots have already been used in robot-assisted therapy [1]–[13], which facilitates delivering a standard and effective treatment to children with ASC using a remote controlled or Wizard of Oz (WoZ) method where the robot is used as a therapeutic tool in the hand of therapist or researcher [5]. In WoZ situations the robot is being fully controlled remotely by a therapist or additional person, hidden from the child and removed from the therapy session. This method has been used successfully but requires an additional operator, different from the therapist who engages with the child [14] [15]. Other approaches, e.g. our own previous work with the Kaspar robot in schools [16] and in family homes have been using a keypad to remotely control some our robot Kaspar's behaviour, not hidden, but in plain sight of the child. Either the adult or the child can be in control of the remote. Although this method has been reliable and successful in robot-assisted therapy, it imposes a cognitive load on the operator during the intervention which may affect the performance of the therapist over time. In addition to this, in WoZ, or remote-controlled scenarios, the robot typically doesn't record the information of the child regarding the child's task performance data, body gesture and head pose, proximity, etc. during the interaction, which means that such systems are typically not suitable to shape live HRIs. To overcome these two issues, it is essential to create some degrees of autonomy and enable the robot to perform some autonomous behaviours during CRIs whilst keeping track of the interaction data.

The position that we need to move towards autonomous robots in robot-assisted therapy is confirmed by Esteban et

Fig. 1. Robot-assisted therapy has been successfully used to help children with Autism Spectrum Condition develop their social skills. As shown Kaspar humanoid robot is interacting with an autistic child to teach him different social skills.

*This work has been partially funded by the BabyRobot project supported by the EU Horizon 2020 Programme under grant 687831. The authors of the paper are with Computer Science Department, University of Hertfordshire, AL10 9AB Hatfield, United Kingdom a.zaraki@herts.ac.uk

al. [15] who developed an autonomous robot and introduced a clinical framework where their robotic development can be tested (Figure 2.(B)). In fact, in this framework the robot is capable of performing tasks autonomously but with the supervision of human who remains in the robot's control loop. Although the presented work [15] seems interesting, the application is limited to the clinical framework and associated equipment (intervention table, sensors and cameras with fixed setup, etc.) and thus this type of autonomous system would likely not work in a real-world setting (e.g. schools) where the environmental factors are not fully within our control. Since our robot needs to function in different schools with unstructured and noisy environments (Figure 2.(C)), we need to design an autonomous robot that supports reliable CRIs in such environments.

One of the key challenges in the development of any autonomous robot is creating the ability for the robot to reason, or understand the sensory information it is receiving, and plan the most appropriate action based on this input. This challenge becomes even more complex when operating in human-centred applications since the robot's reasoning relies on understanding human intention and social behaviours. In such situations, the robot should not only act as a *reactive* agent that simply displays a predefined set of behaviours without maintaining any internal state and without being aware of the status of the interaction. Instead, the robot should have the capacity to act as a *deliberative* agent [17], [18] that explores its behaviour space and predicts the effect of its reaction and displays an acceptable behaviour while taking into account the interaction scenario. The latter case is well suited for the development of autonomous robots and as such may enable an acceptable CRI however, due to the performance requirements to facilitate such scenarios the implementation of a single module to deal with both the robot's planning and motor control is computationally expensive. Taking this into consideration, it seems that a *hybrid deliberative-reactive* control architecture is most likely to provide the best solution when we develop an autonomous robot for real-world settings.

This paper summarizes the design, implementation and experimental evaluation of a deliberative-reactive control architecture called *Sense-Think-Act* that gives some degrees of autonomy to the Kaspar robot [19], [20] for CRIs in real-world settings. The proposed architecture has three subsystems that are fully interconnected via a TCP/IP network which deals with the robot's features from the perception system to the real-time action control system. The 'brain'of the architecture, the Think layer, has been fully developed using the IrisTK [21] which is a powerful state chart-based toolkit for multiparty HRI designed for defining the interaction flow and developing autonomous systems.

Although the architecture is technically capable of *fully autonomous* control over the robot's behaviour in a multi-party CRI, due to the technical issues and the fact that a fully-autonomous robot is not desirable when working with vulnerable children [22], [23], the robot must have approval from the operator before displaying any behaviour to the child. As such, the final framework in a school setting is designed to perform tasks in a *semi-autonomous* manner. For this reason, we have integrated a *permission key* (a Bluetooth key in the operators's hand) into the architecture in order to keep the adult operator in the robot's control loop which eliminates the possible technical and ethical issues during CRI. Alongside the permission key that allows the human operator to take the control of the robot at any stage of interaction (e.g., to re-engage the child to the interaction with the robot), there is also an override button on the key to correct the robot's behaviour if necessary, e.g. if the robot's perception system has made an incorrect classification of an object or other perceptual information that may lead to making a wrong decision by the robot. Regarding the ethical and practical issues concerning full robot autonomy, our experience shows that in order to provide the child with autism with a playful and enjoyable experience with the robot, it is important to take the child's lead, one cannot strictly follow an automated, pre-defined procedure, it is important at certain times to suspend the procedure and move into free play, e.g. when the child gets tired or distracted, and re-engage them by providing play scenarios that the child is known to enjoy, e.g. drumming, the robot singing one of their favourite songs etc. Thus, a careful balance has to be struck between robot autonomy, and preserving the important input of the supervising adult.

In order to evaluate the performance of the autonomous system, we devised nine therapeutic games in which children individually play with the Kaspar robot (dyadic CRI) as well as two joint games in which children would play with the robot in pairs (triadic CRI). The games have been designed to encourage the development of Visual Perspective Taking (VPT) skills in children with ASC (the details of the games are presented in [24] and [16]). For the purpose of this evaluation, we tested the semi-autonomous implementation of Kaspar that we developed to play the same games that we had previously devised and tested. The semi-autonomous implementation of Kaspar played four individual games with two children and also played a joint game with a pair of children while a researcher was sitting next to the robot to facilitate the interaction by evaluating the robot's behaviour and giving the final permission for the robot to display the behaviours recommended by the system (see Figure 9). The preliminary results demonstrated that the architecture has promising capabilities with regards to generating appropriate autonomous behaviours for the Kaspar robot to display which resulted in successful dyadic and triadic CRIs.

Very often autonomous robots are designed and tested in a laboratory environment that is carefully controlled with an array of sensors and recording equipment, clean neutral backgrounds, precisely considered lighting and the exact positioning of equipment (see Figure 2.(A and B)). Taking this approach minimises noise for the robot's perception system to contend with and gives the robot the best possible chance of performing well and functioning reliably. However, laboratory environments are not always suitable for children with ASC as they often find change stressful and will take

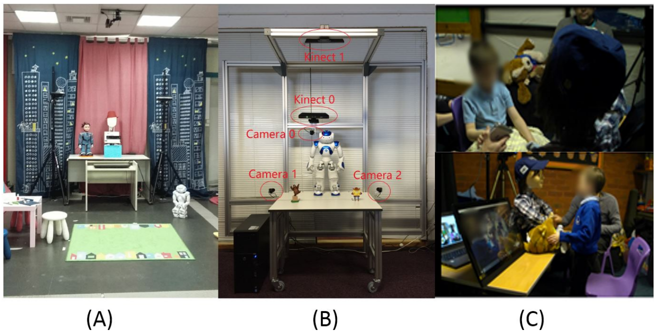

Fig. 2. (A) (AthenaRC Team - babyRobot Project [25]) and (B) (DREAM project [15]) show two examples of the experimental setup in the lab environment, where the location of different sensors, robots and the intervention table are fixed and known while (C) shows the real-world settings with unstructured environments.

a long time to adapt to a new environment. Further to this, operating in a laboratory environment presents a scalability problem because of the logistical constraints of getting the children to and from the laboratory for many interaction sessions. Ultimately, if research is aimed at potentially developing products, robots have to be tested and developed under real-world conditions. Starting with laboratory-only studies and assuming that they will scale up into the real world has often proven to be difficult if not impossible.

II. Challenges Encountered Translating Developments from the Laboratory to Real-World Settings

Taking all of these factors into consideration an inherent requirement of autonomous CRIs is the capacity for the system we develop to operate in child-friendly environments such as schools, similar to where our study took place (Figure 2.(C)). Clearly, the development of an autonomous robot that can perform reliably in noisy, less constrained real-world environment is a much more demanding task that presents new challenges. There are multiple aspects that can have an impact on the performance of the system which vary in different schools: lighting conditions, background noise, dimensions of the room and the desk (see Figure 2.(C)). This is a clear demonstration that the system we have been developing is required to be very robust and resistant to the challenges presented in these types of environments. The main question that has to be answered is: how can we develop a reliable system to function in such environments? What are the best technologies to use for developing an autonomous robot to operate robustly in the real-world environments for real-world applications? To briefly answer these questions, we can say that 2D vision-based systems alone may not be the best solution to functioning reliably in these environments, while sensor-based embedded systems (passive/active) together with 3D vision are likely to provide a more robust option for this task. The rational for this conclusion is discussed in the following sections.

```
{
"class" : "iristk.system.Event",
"event_name" : "action.speech",
"event_id" : "my_unique_id_123",
"text" : "Hello there"
}
```

Fig. 3. The events are serialized into JSON format and distributed to the network via broker. For example, as shown an action event asks the robot to speak with users (for further details please refer to *http://www.iristk.net*).

III. The Hybrid Control Architecture: Sense-Think-Act

As shown in Figure 4, the architecture includes three standalone layers interconnected via a TCP/IP network. Each layer has a number of modules that process either the sensory data captured by sensors/hardware or the high-level information that is distributed to the network as *events* as standard JSON data packets (see Figure 3). The layers and modules are fully interconnected and have the capacity to send and receive high-level information to the network. Thanks to the architecture's modularity and network structure of the system, it is capable of running on multiple devices which facilitates the overall processing cycles for real-time applications, if required. Since the architecture is network-based it is platform independent which means that it supports any new module which is connected to the network, irrespective of programming language which the module has been developed in and also irrespective of the operation system that the module is functioning on, this subsequently means they can easily be integrated into the architecture. One of the primary benefits to this architecture is the potential for scalability allowing us to easily extend the architecture by adding new sensors/hardware devices and also new modules to the system.

In short, the architecture collects the sensory data and extracts high-level information and then streams the corresponding *sense events* as JSON packets to the network (Sense Layer). The central layer receives the JSON packets and evaluates which reactive behaviour is the most appropriate for the current situation taking into account the interaction status and high-level information, and then streams an *action event* (behaviour name) to the network and asks the robot to display that behaviour (Thinks Layer). The Act layer receives the action event from the network and moves the robot servos to display the behaviour on the permission of operator and returns the *feedback/monitor event* to the network to show that performing the action has been completed.

Since the architecture communicates the high-level information in the JSON packets there are two main benefits. It facilitates a real-time CRI since the data communication is so fast, and there is also the potential to create an interaction log-file which includes all the distributed events during the interaction. As discussed the Sense-Think-Act is a fully interconnected architecture which means that the modules

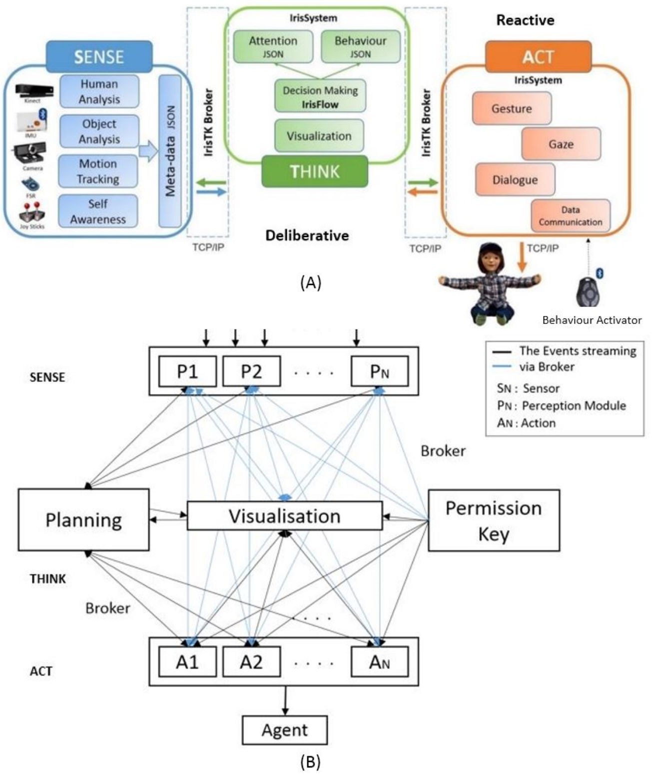

Fig. 4. (A) Kaspar's autonomous hybrid deliberative-reactive control architecture. As shown a Bluetooth key is integrated to the architecture that allows the operator to override the robot's bahaviour and replace it with other behaviour (by pressing the left button on the key) or ask the robot (give permission) to display the behaviour that is planned by the autonomous system (by pressing the right button on the key). (B) The Sense-Think-Act architecture has a fully interconnected structure.

are connected to the same broker, they will receive all the distributed events over the network, however to reduce the computational costs, in each layer there is the possibility to subscribe only to the events that are necessary for that layer and dismiss all the other events.

A. The Integration of the IrisTK Toolkit

The Intelligent Real-time Interactive Systems Toolkit (IrisTk) [21] is an event-based toolkit that supports real-time multiparty HRI and plays an essential role in the Kaspar's autonomous software architecture. It consists of an event passing system, a set of modules for multimodal input and output, and a dialog authoring language system. There are three important components that have to be initialized in any IrisTK-based system: the Broker, IrisFlow, and IrisSystem.

1) - The Broker: is a TCP/IP network that allows sending and receiving the events communicated by different modules over the network. In fact, the Broker creates a fully-connected network of the modules in which the modules of the framework can easily communicate to each other (Figure 4.B). Although the framework is fully connected, to reduce the total computational costs there is the possibility for each module to subscribe to receive only the number of events which are required for that module. For example, as shown in Figure 4.B, the modules of the Act layer (Gesture, Gaze,

Fig. 5. The sense layer senses the user voice sends *sense.speech* to the Broker. The think layer evaluates the event and based on the rules defined in IrisFlow chooses and sends *action.speech* to Broker (robot side). The act layer sends *monitor.speech.end* when the robot has completed the action and finally the think layer ask for the new sense event by sending *action.listen* to the Broker.

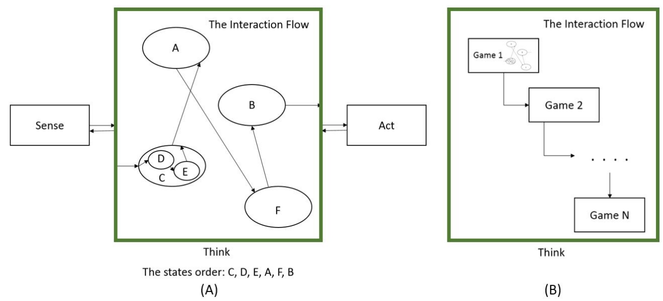

Fig. 6. (A) The implementation of interaction scenario of a game as different states in the IrisFlow. (B) The full implementation of the games in the IrisFlow (think layer).

Dialogue, Data communication) receives the events from the modules of the Sense and Think layers, however this layer has subscribed only to some of the events (see the black arrows). As discussed, the IrisTK is an event-based system which means that it handles the events (instead of low/high level sensory information) to control the behaviour of the autonomous robot. Each event has a name and a set of parameters. By convention, the name of events start with one of the following types: *Sense*, *Monitor*, *Action*. Figure 5 shows an example of how the system communicates these events.

2) - IrisFlow: IrisTK provides a state chart-based framework called IrisFlow for defining the flow of the interaction. IrisFlow is the 'brain'of the architecture and in fact does the autonomous decision making for the system. We have defined all of the game scenarios (the games rule, the meaning of the sense events, and the action that Kaspar should perform in reaction to the children's action, in the autonomous manner) in the IrisFlow.

3) - IrisSystem: The IrisSystem is the main component of IrisTK that manages the IrisTK modules (the Broker and IrisFlow) and events.

B. The Think Layer

The Think layer is the 'brain'of the architecture which receives via the Broker, all the events streamed from the Sense and Act layer and decides how to handle this information in order to make an appropriate decision for the robot's action considering the number of the game, the status of the game, and the previous action shown by the Kaspar robot. We have fully implemented the Think layer using the IrisTK which we have defined in the games [24] as different states in the IrisFlow module. As shown in Figure 6, the interaction flow has an initial state (starting point) for

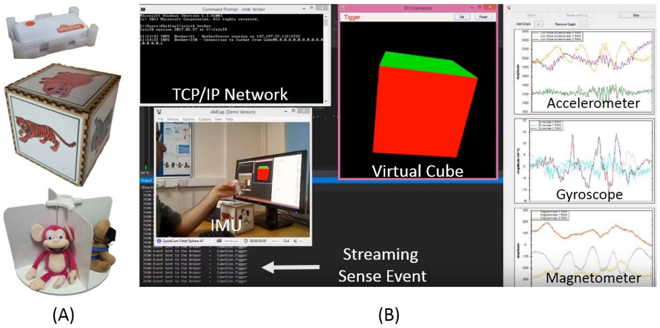

Fig. 7. The Inertial Measurement Unit (IMU) used to develop and wireless sensorised cube and turn table, (B) The IMU's tracking system estimates the 3D orientation of the IMU analysing the signals of its embedded sensors (accelerometer, gyroscope, and magnetometer). The module streams the *sense event* (3D orientation vector) to the TCP/IP network to drive a virtual cube as well as to control the Kaspar autonomous behaviour in playing with children.

example state C. Using two main commands of IrisTK (*goto* and *return*), the interaction flow goes back and forth in the states based on the events streaming from other modules. Following the arrows in the Figure 6.(A), the states of the interaction flow that will be triggered are ordered as follows: C(D,E), A, F, B. The output of the final state (B) is the name of the action that system drives Kaspar to display. Since the architecture has a semi-autonomous process, firstly, the Think layer communicates the name of the selected action with the operator, and the robot must have the approval from the operator before displaying that action to the child. We have implemented the games in a single interaction Flow (Figure 6.B) in the Think layer. Before starting the game with children the human operator specifies the game number by scanning an RFID card to the system and afterward the architectures will activate the rules of that game.

C. The Sense Layer

The sense layer includes a number of sensors which sense the environment, and the associated perception modules that we have developed to interpret the sensory information in order to extract *sense events* occurring in the CRIs. We have chosen the sensors and developed the associated perception modules base on the requirement that the Kaspar robot needs to play the dyadic and triadic games [24] in a semi-autonomous manner. Therefore, based on the requirements of these games, Kaspar should possess the following perception capabilities: 3D orientation of cube and turntable tracking, object recognition, and human gesture analysis. The following section describes the implementation method of the above perception modules.

1) - 3D Orientation of Cube and Turn Table Tracking: This module has been design to estimate the 3D orientation of a cube and the rotation angle of a turn table (Figure 7) with the final aim of recognizing the animal pictures on the cube sides as well as the animal toy that is placed on a section of the turn table between the dividers. Since doing this perception task using a vision-based system is relatively difficult, we developed a wireless sensorized cube and turn table by placing a Bluetooth compatible IMU into the cube and on the top of the turn table (Figure 7.(A)). The IMU's 3D orientation has been achieved by integrating a software API (provided by Shimmer company [26]) and analysing the accelerometer, gyroscope and magnetometer signals of the IMU using an algorithm presented in [27] (Figure 7.(B)). Knowing the 3D orientation of the cube/turn table is very important since that enables the system to understand which side of the cube/turn table is currently being observed by the child and which side is being presented to Kaspar robot which is a part of the game.

The main factor affecting the precision of the IMU's data as well as the performance of the tracking algorithm is the calibration parameters. The three sensors of IMU need to be well calibrated before the using it. The calibration process will result in three matrixes called the calibration parameters which can be stored on the IMU's either in the IMU's built-in memory or as an external file on the PC. As observed in our trials, due to the very fast or unexpected motion of IMU, sometimes the system loses tracking of the orientation of the IMU. In this case, we need to perform a set/reset by pressing a button on the graphical interface on the screen which changes the current values of the IMU's sensors to the standard values and will correct the errors of the system. Finally, the system estimates in real-time the 3D orientation of the IMU and via the IrisTK broker sends the orientation vector as a JSON data packets to the Think layer.

2) - Object Recognition: This perception module has been implemented using an image processing library which receives the image of the embedded camera from the Kaspar robot's eyes and tracks and recognizes multiple toys based on their colour and the size of colour region. In order to analyse multiple toys, image processing techniques such as blob detection and colour filtering have been employed to detect and extract an object from the background and determine the pixel address in the 2D frame. For this reason, the object analysis module, firstly, acquires the image constructed by the RGB camera embedded in Kaspar's eye, and processes the image in order to convert its specifications (dimensions and pixel ratios) into the one required for the filtering step. The module then applies different filters to extract the specific colours in order to identify the colour regions in the image. Finally, it returns as the output, the pixel address (x,y) of each object in the camera's FOV. In order to facilitate and improve the above aforementioned image processing tasks, the open source image processing/vision library called AForge .Net has been integrated into the toy analysis module. We have produced an automatic filter in which system will recognise colours without applying any manual filter adjustment. All we need to do is clicking on the region of the interest on the image captured from Kaspar's eye camera through the interface. The module will evaluate the size and the colour of the object and store its corresponding ID. The module detects and tracks multiple objects simultaneously and streams the information of the objects to the Think layer via IrisTK Broker.

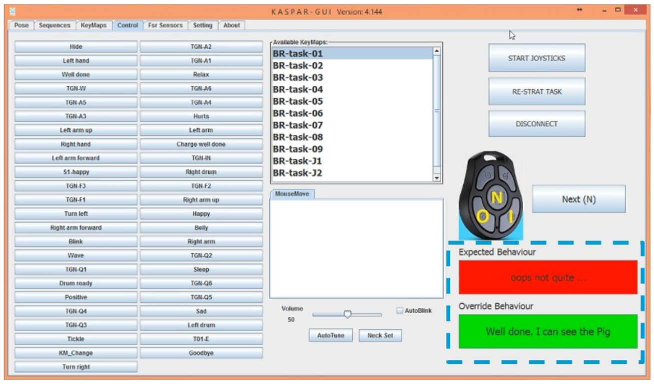

Fig. 8. The reactive system of the architecture that has been developed to control and display different behaviour on Kaspar robot. The red box displays the expected behaviour the behaviour that is estimated by the deliberative system in an autonomous manner, and the green box displays the name of the override behaviour that system estimates based on the interaction status and the Interaction Flow. These boxes will be shown on the GUI and the human operator who gives the *override or permission signals* to the robot by the pressing a button of the Blutooth key.

3) - Human gesture and voice analysis: This perception task has been achieved by integrating the Microsoft Kinect Software Development Kit (SDK) v2.0 into the sense layer. It is able to simultaneously detect up to five humans in the robot's field of view (FOV) and track the 3D positions of 25 body joints for each person, in real-world coordinates. Using this information, the body gesture and head posture of the user can be recognized, which is important in CRI, due to their social communicative roles in human social interaction. The speaker tracking capability of the layer has been realised using the beam formation functions of the SDK. By comparing the beam angle in the environment with the 3D angles of the children seen in the FOV, the individual who speaks can be tracked in real-time. In order to recognize the voice of the speaker, we have integrated the Windows speech recognition platform into the sense layer. This module receives the audio signals through the Kinect's microphone array and recognizes the utterance of users based on a pre-defined list of the word/sentences that we have defined in Kaspar's database. Using this module, the users (mostly adults) will be able to verbally interact with the robot.

D. The Act Layer

The final layer of the architecture is a reactive (act) system which has been developed to provide Kaspar's control signals in order to display different behaviours on the robot. Similar to other layers, the reactive layer is also connected to the IrisTK Broker and receives all the events, however the system has subscribed to receive only the events of the Think layer. The Kaspar reactive system has several pre-programmed behaviours stored as different external files that are typically used for generic play sessions and include various postures, hand waving, drumming on a tambourine that is placed on its legs and singing children nursery rhymes. Each behaviour file includes the name of the sequences that are required to generate the behaviour. Instead, each sequence file includes 22 motor position values to control the Kaspar's servos, and also the name of the voice file that has to be played by Kaspar. With the previous Kaspar control architecture we were able to activate these behaviours by pressing buttons either from a keypad or from the software interface. However in the current version (semi-autonomous), the Think layer will decide and activate a behaviour by sending an action event to the Act layer via the Broker. The Act layer has a sequence-player method that receives the name of the behaviour and plays the corresponding behaviour sequence. Figure 8 illustrates the Kaspar GUI for the reactive system which is connected to the architecture via the Broker. As shown there are two boxes (red, green) on the bottom-right corner of the GUI. The red box displays the behaviour that is estimated by the deliberative system according to the perceptual information provided by the Sense layer, and the green box displays the name of the correct behaviour that system estimates based on the interaction status and the Interaction Flow. These boxes are displayed in the GUI and the human operator has to make the final decision for the robot's behaviour. He/she can give the final permission for the robot to display the behaviour presented in the red box or can override the robot's behaviour and ask robot to display the behaviour presented in the green box.

Fig. 9. Trial with children with ASC in school: children are playing dyadic and triadic games with the semi-autonomous Kaspar robot.

IV. Experimental Evaluation of the System with Children at the School

We implemented the proposed architecture on the Kaspar robot and tested it at a school for two main reasons: to evaluate the real-time performance of the architecture in controlling the Kaspar robot in a real-world setting, and to find out if the architecture is capable of controlling the robot's behaviour in an autonomous and acceptable way in dyadic and triadic interactions with children. We installed the three layers of the architecture on a single laptop (Toshiba Tecra, Intel Core i7, 2.60 GHz, 16GB RAM) for the compatibility test as well as to check the overall real-time performance.

Four children (two males and two females) with the mean age of 13 and with different levels of ability that had been diagnosed with ASC took part in the study. Two children individually played two games with the Kaspar robot (dyadic CRI) and two children played two joint games with the robot in pairs (triadic CRI) (see Figure 9). The games were designed to encourage the development of Visual Perspective Taking (VPT) skills in children with ASC (the details of the game can be found in [24]). Each game had a progression criterion with the children needing to successfully complete the game three times before they could move to the next game. Subsequently each child took part in a different number of sessions as the rate at which they progressed through the games varied. In total the data of 11 child-robot interactions was collected.

Analysing the results we observed that the architecture was able to provide robot control signals consistently in real-time without any latency which supports real-world applications. Furthermore, the think and act layers functioned correctly in all interaction sessions, however due to the real-world issues (lighting conditions, etc.) we encountered some issues in the object analysis module which meant we sometimes had to override the behaviour suggested by the system. Lastly, the three layers of the architecture and their modules were successfully operated in real-time on the same laptop which facilitates running our next studies in schools.

In order to understand how the proposed system was successful in reducing the cognitive workload on the operator, in addition to children data, we collected data of the supervising adult during all of the sessions with the children. By analysing the therapist's data we can learn how the therapist shared his attention between the children and the robot and measure the reaction time in giving the override or permission signals to the robot. We expect that the therapist should dedicate more attention to the child instead of the robot as the robot is autonomous and he was just required to check and give a signal to the robot. Our initial observations show that the system was able to reduce the burden on the therapist, however detailed analysis of the data is still on going and we will present the full results in another publication once the work is completed.

V. CONCLUSION

The initial tests of the proposed robotic system sense-think-act, appear to be very promising and would suggest that the system has the potential to solve some of the problems that are experienced by a remote controlled robot in therapeutic secessions in real-world settings however, there are still some limitations and some hurdles to be overcome in order to fully utilise the potential of this robotic system.

- Increasing the system's reliability by adding new technology to the sense layer: the reliability of the system is relied on how accurately the system can sense the environment. To increase the reliability we are aiming to replace the 2D vision/perception module with RFID technology which is more robust to the changes in the environment and factors in real-world settings.
- The system's level of autonomy should be increased by adding the ability to learn on the fly and integrate it to the deliberative layer: The final aim of this work is to increase the level of system autonomy and thus the robot should be able to gradually learn from the environment and the therapist's behaviour about how to react to the children (timing and type of the behaviour) in a more autonomous manner which can result in decreasing the cognitive load on the therapist.

VI. FUTURE WORK

We are currently continuing to improve the architecture, in particular we are considering to replace some sensors with a long range RFID system which is likely to perform much more reliably under real-world conditions in schools environment.

REFERENCES

[1] Sang-Seok Yun, Hyuksoo Kim, JongSuk Choi, and Sung-Kee Park. A robot-assisted behavioral intervention system for children with autism spectrum disorders. *Robotics and Autonomous Systems*, 76:58–67, 2016.

[2] Bram Vanderborght, Ramona Simut, Jelle Saldien, Cristina Pop, Alina S Rusu, Sebastian Pintea, Dirk Lefeber, and Daniel O David. Using the social robot probo as a social story telling agent for children with asd. *Interaction Studies*, 13(3):348–372, 2012.

[3] Ramona Simut, Cristina A Costescu, Johan Vanderfaeillie, Greet van de Perre, Bram Vanderborght, and Dirk Lefeber. Can you cure me? children with autism spectrum disorders playing a doctor game with a social robot. *International Journal of School Health*, 3(3), 2016.

[4] R Simut, C Pop, J Vanderfaeillie, D Lefeber, and B Vanderborght. Trends and future of social robots for asd therapies: potential and limits in interaction. In *International Conference on Innovative Technologies for Austism Spectrum Disorders (ASD): tools, trends and testimonials*, 2012.

[5] Claire AGJ Huijnen, Monique AS Lexis, Rianne Jansens, and Luc P de Witte. Mapping robots to therapy and educational objectives for children with autism spectrum disorder. *Journal of autism and developmental disorders*, 46(6):2100–2114, 2016.

[6] Brian Scassellati, Henny Admoni, and Maja Matarić. Robots for use in autism research. *Annual review of biomedical engineering*, 14:275–294, 2012.

[7] Serge Thill, Cristina A Pop, Tony Belpaeme, Tom Ziemke, and Bram Vanderborght. Robot-assisted therapy for autism spectrum disorders with (partially) autonomous control: Challenges and outlook. *Paladyn*, 3(4):209–217, 2012.

[8] John-John Cabibihan, Hifza Javed, Marcelo Ang, and Sharifah Mariam Aljunied. Why robots? a survey on the roles and benefits of social robots in the therapy of children with autism. *International journal of social robotics*, 5(4):593–618, 2013.

[9] Ester Ferrari, Ben Robins, and Kerstin Dautenhahn. Therapeutic and educational objectives in robot assisted play for children with autism. In *Robot and Human Interactive Communication, 2009. RO-MAN 2009. The 18th IEEE International Symposium on*, pages 108–114. IEEE, 2009.

[10] Daniele Mazzei, Alberto Greco, Nicole Lazzeri, Abolfazl Zaraki, Antonio Lanata, Roberta Igliozzi, Alice Mancini, Francesca Stoppa, Enzo Pasquale Scilingo, Filippo Muratori, et al. Robotic social therapy on children with autism: preliminary evaluation through multi-parametric analysis. In *Privacy, Security, Risk and Trust (PASSAT), 2012 International Conference on and 2012 International Confernece on Social Computing (SocialCom)*, pages 766–771. IEEE, 2012.

[11] Felippe Sartorato, Leon Przybylowski, and Diana K Sarko. Improving therapeutic outcomes in autism spectrum disorders: Enhancing social communication and sensory processing through the use of interactive robots. *Journal of psychiatric research*, 90:1–11, 2017.

[12] Takayuki Kanda, Rumi Sato, Naoki Saiwaki, and Hiroshi Ishiguro. A two-month field trial in an elementary school for long-term human–robot interaction. *IEEE Transactions on robotics*, 23(5):962–971, 2007.

[13] Hideki Kozima, Marek P Michalowski, and Cocoro Nakagawa. Keepon. *International Journal of Social Robotics*, 1(1):3–18, 2009.

[14] Elizabeth S Kim, Lauren D Berkovits, Emily P Bernier, Dan Leyzberg, Frederick Shic, Rhea Paul, and Brian Scassellati. Social robots as embedded reinforcers of social behavior in children with autism. *Journal of autism and developmental disorders*, 43(5):1038–1049, 2013.

[15] Pablo G Esteban, Paul Baxter, Tony Belpaeme, Erik Billing, Haibin Cai, Hoang-Long Cao, Mark Coeckelbergh, Cristina Costescu, Daniel David, Albert De Beir, et al. How to build a supervised autonomous system for robot-enhanced therapy for children with autism spectrum disorder. *Paladyn, Journal of Behavioral Robotics*, 8(1):18–38, 2017.

[16] Ben Robins, Kerstin Dautenhahn, Luke Wood, and Abolfazl Zaraki. Developing interaction scenarios with a humanoid robot to encourage visual perspective taking skills in children with autism–preliminary proof of concept tests. In *International Conference on Social Robotics*, pages 147–155. Springer, Cham, 2017.

[17] Daniele Mazzei, Lorenzo Cominelli, Nicole Lazzeri, Abolfazl Zaraki, and Danilo De Rossi. I-clips brain: A hybrid cognitive system for social robots. In *Conference on Biomimetic and Biohybrid Systems*, pages 213–224. Springer, 2014.

[18] Abolfazl Zaraki, Michael Pieroni, Danilo De Rossi, Daniele Mazzei, Roberto Garofalo, Lorenzo Cominelli, and Maryam Banitalebi Dehkordi. Design and evaluation of a unique social perception system for human–robot interaction. *IEEE Transactions on Cognitive and Developmental Systems*, 9(4):341–355, 2017.

[19] Kerstin Dautenhahn, Chrystopher L Nehaniv, Michael L Walters, Ben Robins, Hatice Kose-Bagci, N Assif Mirza, and Mike Blow. Kaspar–a minimally expressive humanoid robot for human–robot interaction research. *Applied Bionics and Biomechanics*, 6(3-4):369–397, 2009.

[20] Luke J Wood, Abolfazl Zaraki, Michael L Walters, Ori Novanda, Ben Robins, and Kerstin Dautenhahn. The iterative development of the humanoid robot kaspar: An assistive robot for children with autism. In *International Conference on Social Robotics*, pages 53–63. Springer, Cham, 2017.

[21] Gabriel Skantze and Samer Al Moubayed. Iristk: a statechart-based toolkit for multi-party face-to-face interaction. In *Proceedings of the 14th ACM international conference on Multimodal interaction*, pages 69–76. ACM, 2012.

[22] Mark Coeckelbergh, Cristina Pop, Ramona Simut, Andreea Peca, Sebastian Pintea, Daniel David, and Bram Vanderborght. A survey of expectations about the role of robots in robot-assisted therapy for children with asd: Ethical acceptability, trust, sociability, appearance, and attachment. *Science and engineering ethics*, 22(1):47–65, 2016.

[23] Andreea Peca. Robot enhanced therapy for children with autism disorders: Measuring ethical acceptability. *IEEE Technology and Society Magazine*, 35(2):54–66, 2016.

[24] Luke Wood, Kerstin Dautenhahn, Ben Robins, and Abolfazl Zaraki. Developing child-robot interaction scenarios with a humanoid robot to assist children with autism in developing visual respective taking skills. In *The 26th IEEE International Symposium on Robot and Human Interactive Communication, RO-MAN 2017*, pages 1–6. http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8172434, 2017.

[25] BabyRobot. Nextgen social robotics, 2018. available http://babyrobot.eu/, 2018.

[26] ShimmerSensing. Inertial measurement unit, shimmer and technology available http://www.shimmersensing.com, 2018.

[27] Sebastian OH Madgwick, Andrew JL Harrison, and Ravi Vaidyanathan. Estimation of imu and marg orientation using a gradient descent algorithm. In *Rehabilitation Robotics (ICORR), 2011 IEEE International Conference on*, pages 1–7. IEEE, 2011.