Group Members:
Dogan Yazar (leader)
Muneeb Anwar Khan
Erdem Aksu
Marcell Alzate
Li Yang
Hao Zhang

This week, we have read all the previous report about our project and finished reviewing them. You can find all the reviews done by the group members in the following pages (in their pure form such that they include offers about our methodology, personal ideas, etc).

We have analyzed the working of the “SpyBO” project from the year 2007 and discussed it’s working in detail. According to our critical analysis, this project group has come up with a really good solution and apparently has optimized the solution by using time varying joint-stiffness technique. According to our thinking, it’s most probably not possible to optimize the technique for a normal gait beyond this. In short, the work done on AIBO’s body in this project is optimal and possibly the best amongst all the possible solutions to the “seeing while walking” problem, while trying to overcome the problem without working much with the head. Since the work done by the “SpyBO” team is pretty good, our group has decided to put up a proposal (also our rough sketch), to use the gait of the AIBO from the “SpyBO” project, which will help us with a much stable gait for the robot, while we as a group try to extend this project further by developing better techniques for managing the movement of the head. We intend to use the effort made by the “SpyBO” team, and while using their “very stable” gait, develop a better technique for stabilizing the head along the different axis, to further stabilize the “SpyBO” vision. According to our analysis having an optimized gait will help us in maintaining a much more stable vision for AIBO, while working with its head, and having optimized its head movement we can expect some really good results. This idea seems productive to us, but we certainly are need of a critical review or possible limitations we can come across while doing so.

Also, as we stated earlier, we are not comfortable with the control design part and we agreed to apply to following approach: We will try to run and make some experiments using the code of “SpyBO” project and at the same time, we will try to understand all the
specific details especially about the control design part and ask for your help when we stuck. :))

Other than that, we worked on Tekkotsu and as usual had some troubles. First of all, only the stable version of Tekkotsu currently works with 64 operating systems (we installed it on a notebook having 64 bit OS), so latest release has compile problems. Next, even the sample project of Tekkotsu requires more space then it is offered by our memory sticks. We are really surprised why this is not mentioned in the previous years reports and asked it to Tekkotsu mail group and we are told that we need to remove some sample behaviours and/or remove sound files to make it fit on 8 MB sticks. Or get a bigger memory stick :))). They also advise to remove some of the sample behaviors since ERS 210 model is a little bit old and has a tight memory. In addition, we tried to run the “Head Leveler” example mentioned in the Report 1 of 2004 (Our aim is to get used to Tekkotsu framework and have some insight). But again, since it is an old example, it is not compatible with the framework now, so we are trying to find a work around to solve the problem.

One of our members had listed the pros and cons of Tekkotsu and OPEN-R frameworks (mostly based on the information given in the previous reports) which can also be found in the following pages. Sadly, we don't have any experience about their comparison yet. Moreover, we could not have much chance to work on Urbi. We only loaded a precompiled Urbi stick and got familiar with commands. For the next reports, we are hoping to have more experience and write more stuff about it.

By the way, there is one point that we could not really understand in report 9 of 2007. In the Acquiring Data section, it is written: “Since the robot tipped over when trying to walk straight using the pre-defined gaits, the standard crawl was chosen. Thus for the rest of the report, the walk used is a crawl on the front paws. Though this doesn’t solve the problem of seeing while walking it is not unreasonable to believe that the models generated are not specific to the walking style chosen.”

Can you please explain it?
Comparison between OPEN-R and Tekkotsu

1. Compare OPEN-R SDK and Tekkotsu Framework  
(Mainly based on previous groups reports)

1.1 OPEN-R SDK
1.1.1 OPEN-R SDK Pros
1) lowest level programming interface, its C++ library provides lower level of 
programming.
2) the sample programs are an excellent source for learning OPEN-R and getting ideas on 
how to solve problems. (although need time to understand them)
3) provides the maximum freedom known until now to control AIBO.

1.1.2 OPEN-R SDK Cons
1) very poorly documented.
2) only some basic tutorials available, that's all.
3) one of the sample programs include any documentation about what they do or how, 
need to spend a lot of time on figuring out how things work.
4) complex installation and coding.

1.2 Tekkotsu Framework
1.2.1 Tekkotsu Framework Pros
1) less steep learning curve.
2) include a handy Java graphical interface with many useful capabilities.
3) higher level framework based on OPEN-R, extended with several ready made libraries 
and packages for performing common tasks easily, is a kind of wrapper of OPEN-R 
platform.
4) more user friendly platform for AIBO programming.
5) very well encapsulate the AIBO side codes.
6) lots of classical C++ idea, like templates, smart pointers, are presented in its design.
7) the codes are elegant.
8) provides a well designed remote control user interface in JAVA which gives an 
illuminating idea of how AIBO can walk, shout, hear, and see in the easiest and fastest 
way.
9) pre-coded monitors and controllers makes it a good example to learn how to work with 
AIBO.
10) easier to start with, also with enough freedom.
11) well documented.
12) designed with portability in mind, currently runs on the ERS-210, ERS-220 and 
ERS-7.
13) Events and class interfaces make it easy to add new functionality.

1.2.2 Tekkotsu Framework Cons
1) less freedom in programming to control AIBO.
2) stdout in program running on AIBO's side is redirected to telnet port 59000 by default, unlike outputting characters to console, redirecting stdout via network is not a synchronized operation, which means if program crashes while running, it is not guaranteed that all output data buffered in network cache will be sent out.

1.3 Reasons to choose OPEN-R SDK
1) all the OPEN-R SDK pros listed in 2.1.1.
2) some of the project reports from earlier years indicate that implementing the controlling software in Tekkotsu might not provide good enough performance to achieve satisfactory results.
3) most of the previous projects have relied on Tekkotsu, it would be interesting to take a different approach in this regard, to see how it works out.

1.4 Reasons to choose Tekkotsu Framework
1) all the Tekkotsu framework pros listed in 2.2.1.
2) From the interface, it is possible to execute so called behaviours, both pre-installed and behaviours written by the user. The behaviours are C++ classes with certain methods defined by Tekkotsu.
3) it is an application framework which handles the low level or routines tasks for you so that one can concentrate on whatever is unique to the application.
4) Tekkotsu monitoring tools (TekkotsuMon) are important tools for controlling movements, start behaviors and capturing images. Its main interfaces include:
   - Walk Remote Control
   - Head Pointer Remote Control
   - Sensor Observer
   - Raw Cam Window

1.5 Conclusion

OPEN-R SDK enables lower level programming, better controlling AIBO, but poor documented and has few learning material, hard and time-consuming to get along with it.

Tekkotsu framework enables higher level programming based on OPEN-R, well encapsulated, well documented and supported, offers user-friendly GUI, easy to start with and learn, but has limitation on controlling AIBO and less freedom in programming.

2. My personal suggestion for our project

After comparison between OPEN-R SDK and Tekkotsu framework, I find both have their strong and weak points. However, it seems that Tekkotsu framework is more preferred than OPEN-R by most previous project groups. And it turned out that Tekkotsu could meet computational and performance requests in the "Seeing while Walking" project although it contains certain limitations.

So I suggest two alternatives:
1) Totally base our project on Tekkotsu framework
2) Use OPEN-R SDK for programming together with Tekkotsu monitoring tools
   TekkotsuMon (Like the Seeing While Walking project group in 2006)

3. **Future work**
   1) Try sample codes and read more documentations on OPEN-R and Tekkotsu
   2) Observe Urbi framework and try to find out the differences
Reviews

The previous year experiments have mostly worked with either stabilizing the head or softening the gait and then further stabilizing the head in the vertical direction and hence trying to dampen the frequency of the shocks received by AIBO’s head. Such experiments have shown much improvement in the way AIBO sees while it is walking. Previous groups working with stabilizing the head, have gone through a whole procedure of taking test data from different test runs performed on the AIBO, and then used these test results to generate information regarding the affects of the shocks received by the AIBO’s head, as to how do the shocks displace the head with respect to the point of focus along the X and the Y-axis. Then according to the findings, the head is controlled such that it manages its position according to the data it receives from the controllers operating the legs. Stiffening the head while walking may also help in reducing the movements due to shocks. The test data is then used to generate a suitable function (this is done by interpolating the function received through actual runs and then adjusting it). This entire procedure of working with the test data and coming up with a suitable function can be termed as an “offline procedure”. Once we have come up with a suitable equation as a result of the offline procedure, we can convert it to an algorithm (an “online algorithm”), which we can run on the AIBO and hope that it behaves correctly under the exact conditions as the test runs. However, playing around with the gait may produce incorrect behaviour. This solution probably requires further control design to improve its robustness.

Report 4 – 2006

1. Methodology to measure camera movement: to build model & evaluate results (8)

2. Develop a control algorithm

3. Evaluate with methodology from 1.

Description of movement that they identified in the AIBO:

Rotation around the x axis

Rotation around the y axis

Movement along the y and z axis

Movements in the x-y plane

Backlash:

Present every time the legs touch the ground and due to the slack between the joints and motors. It is one of the most influence reasons to have this movement in the head while
the AIBO is walking. While it is physically normal; it is unpredictable and was one of the biggest problem in the implementation of their solution.

Measurements:

They focuses in 2 biggest shaking components: roll & tilt.

Solution:

Finding a solution to the problem implies to related data from the sensors (accelerator sensor, leg position derived by the joint sensors) to the camera movement.

Initial approach is to inverted/compensated camera movement; which is described as a naive solution since best approach should have prediction in consideration.

Getting the data by Telnet

Measurements sent to the PMS (Programmable Memory Stick) makes the AIBO to have changes in the walking styles; causing error in the measurements. Working with telnet make the trick, but to port 10001, not to use the port use by teksukko since it will slow down the Remote Control.

In order to be able to use a prediction method it is important to have synchronization mechanism; such mechanism let them trigger the prediction. For that matter it was important to define which was AIBO walking cycle --together with the position of the leg are the reason of the camera movement-- that would let them decide when to trigger the prediction mechanism.

Conclusion by the group:

1. Backlash is hard --impossible?-- to predict when having in consideration impulse forces when the legs hit the ground.
2. Accelerator sensors while not noisy could be really complex as a reference indicator for the same reason of the impulse forces.
3. Working with the PMS might cause some troubles, specially while doing measurement that record a lot of data into it. It might cause a short freeze in the AIBO.
4. They consider that the motor could be a factor of limitation when trying to stabilize the camera. While trying to recreate the shaking by manipulating the motors they found out that it isn't possible. Response time of the PID (motor, while do they call the PID motors?) is a limiting factor and its value is about 0.12s.

My conclusions:

PID: IMPORTANT AS A CONTROL THEORY: it will help to make the proper measurements and implement a correct control algorithms, and it was pointed out in the Introduction exercises (Intro.pdf) document that Seeing while walking group did the best
job in the 2004 course; however, they missed to have in consideration the use of the head
PID-regulator. Their model base in "use system identification algorithms with the
acceleration sensors as input and camera movement as output" clearly failed for the
person who was doing this review. We should spend some time to understand this in
order to avoid having the same problem.

I saw somewhere in other document --I don't recall which one-- that small movements
could be better than bigger movements. Anyway, we need to verify which one is better
since it is going in contradiction to 4th conclusion of this group.

I don't think that the sketch must have a definite path to follow... I would say that it can
be something like:

1. Find a proper measurement method
2. Define a model that explains the camera movement
3. Implement a control mechanism
4. Evaluate with method in 1.

During those steps, there should be small tasks like defining development tool,
understanding control theory, etc.

Report 5 – 2006

Basically, this report describes a solution to stabilize the camera on AIBO, which find an
offline module to compensate for the unwanted head movement based on data collected
in the previous step, by introducing a program written in MATLAB.
The basic idea is that: an offline model is constructed and applied to the online movement
of AIBO, by adjusting the AIBO’s head movement to stabilizing the camera, and
consequently reducing the shaking of the objects in the image stream.

They developed a program to measure the shakings of the head while AIBO is moving.
This program received pictures from the robots camera, and processed these offline to
determine the pattern of the shakings. From this, a stabilizing algorithm was constructed
for moving the head in a way that reduces the shakings. This is also done offline. This
algorithm is then transferred back to AIBO, and used while the dog is walking. This
method actually proved to be quite successful in reducing the shakings of the head, as
they managed to reduce the shakings by more than 50%. The drawback of this method is
that it's a static model, as it requires the compensation algorithm to be created offline and
not in real time on the robot. It's true that different algorithms rather quickly can be
calibrated from measurements, but it's still not really a dynamic control mechanism.
Here is a summary list for their conduction during their project
(1) Head Shaking
By using Tekkotsu’s monitor tool, “Save Image Sequence” on “Vision Raw” GUI would transfer image sequences from the AIBO to the host computer in PNG format and save them in the same folder along with an automatically generated images index file in txt format.

To produce enough scale for AIBO’S low camera resolution(176*144 pixels), a 2-D object, which is referred to as the CROSS, consists of two black decussated lines; in order to reduce the CPU load, the camera resolution is reduced to half of its maximum value; the CROSS is placed against the wall under sufficient light resource.

Camera Shaking Measurement: the movement on X and Y axis could be considered as roughly periodic. Periodic Movement of The Cross: the camera shaking amplitude is calculated by the difference between the start and the end point (mix and max). Similarly, the camera shaking amplitude of one whole walk could be measured by calculate the mean value of all the steps it contains.

(2) Walking Style and Camera Shaking
The joints of the AIBO’s leg adjust according to the angular value to achieve the position in the motion locus. To simulate a natural walk, the left-front and right-back is defined as one pair of simultaneously moving legs, and the other two legs the other pair respectively. Assume that the backlash is very small.

(3) Compensation Control Curve
Compensation function: derive two high-order finite functions for compensation in X and Y direction, in which percentage of period is the input and neck joints value is the output. A method that utilizes more memory and less CUP time is introduced. The indexed compensation curve only compensates for part of the shaking and the rest of the work is done by the backlash caused by the compensation control. Moreover, the control and the effect could not be simultaneous, because there’s a delay before neck joints are updated.

The problem is: as for translating movement on X and Y axis has a maximum amount of 5 pixels from the original center, this measurement is definitely not very precise. Since miscalculating of one pixel can leads to 10% error, therefore it’s possible that some very small fluctuations on the curve are not real camera’s motion but belongs to miscalculating.

**Report 7 – 2007**

Basically, I think this report is good. Since it analysis the precious project detailed and got nice summaries (I recommend we’d better read this analysis part: called "Background and analysis of previous work" from page 5-6), and provides good solution to improve it. Moreover, they use OPEN-R instead of Tekkostu Framework which none group had done before. At the same time, they also left some good suggestions without solving them since time restriction. So I think we can conduct these kinds of problems if we want to.
Although they have been unable to verify the performance of their model and an implementation, they gave good hints to us, such as discuss what caused the problem, and ideas about how it could be solved or avoided. So I think we can try this field further and improve the approach?

According to their analysis, their problem is the control part, so I think we can consider this, but the problem is that they conduct the project based on control theory, I do not know who is familiar control theory in our group?

They gave two ideas to solve the problems, but I'm not sure whether is ok or not, we need to test, I think. One is sending the raw data format directly or as BMP instead of converting images to JPEG. Another is that improve the control system by doing a more accurate model with more inputs and outputs. Since theirs is controlling the pan and roll separately and not as a multivariable control system.

Here I think I should explain why they choose converting images to JPEG format to taking Pictures from Robot, they did this in part to save bandwidth of either the Wifi connection or the memory stick bus, and in part because the OPEN-R sample program called W3AIBO contains a rather easy to use code package for converting pictures to JPEG.

What's more, for storing the pictures, transferring the pictures over the wireless connection, since storing in memory stick is so slow that the entire robot started shaking while walking, only to keep the client-server model for sending pictures. In a word, the system is made up of a server, written in OPEN-R and running on the robot, and a client. But I’m not sure whether it’s right if we adopt their suggestion to sending data format directly, we need to verify!!

What we can conduct based on this report:

An obvious improvement would be to use a persistent connection between the robot and the client, instead of opening a new connection for every photo. What is needed then is some way for the client to tell where the data of one picture ends and the next begins. This could be accomplished either by the server first sending the length of the next picture and the client only receiving that number of bytes, or by deciding on some token (i.e. some sequence of bytes) that indicates the end of a picture. It is not certain what performance increases these modifications would provide though, and they have not had time to evaluate them.

Implementation of the filter and control mechanism. They just evaluate the correctness of phase shifts in simulation; they need to compare it again with the image analysis to prove it further. Moreover they did not evaluate whether the amplitude is ok or not.

Here is some conclusions from their report, might be useful for us during our project so I make summary as follows:

(1) System identification: the right hind leg knee was most likely to be correlated with the shakings in the head.

(2) It's hard to discover any periodicity at all in the Y(tile) sequence so after some tries with different models the decision was made that there was no use in trying to control the shakings in the Y direction.

(3) Since all data from the image analysis has the low sampling frequency. To make their
model work for these fast sampled signals they did an up-sampling of the signals before the modeling was made.

(4) To compensate for the phase shift, they use a formula to calculate based on storing the previous 11 samples, which is a simple algorithm, so I think we can try.

(5) The input signal that is used comes from a knee joint that is well correlated with the output signal from the image analysis that represents the movements for the head. The first part: input signal is filtered to a nice sinusoidal signal with the same period as the original knee joint signal, which depends on previous values, here they use 8 significant digits instead of the default 4 from MATLAB, and finally, the algorithm scales the signal linearly to the interval that the actuator expects.

**Report 8 – 2007**

Another group that worked on the project is SpyBO, worked with AIBO’s gait, which attributes the most to the shocks transferred to the head from the paws. The solution provided assumes that a softer gait or “stepping” can improve the image quality captured by AIBO’s camera while it is walking. In this project, the group mainly focussed on minimizing the shocks transferred to the head by softening the joints on impact with the surface. This significantly reduced the jerks (high frequencies) received by the AIBO’s head. The residual shocks transferred to the head were compensated by adjustment of the head in the Z direction.

**Report 9 – 2007**

The report 9 of year 2007 is a well written and informative report. The best thing in the report for us is their systematic way of working. It starts with the abstract of what they did, then step by step the problem is defined, general approaches are specified and most importantly a very good literature survey of what has been done by the previous groups is given; their approaches, the problems they faced and how they tried to solve them, failures, successes, lessons learned from them, etc. Obviously, the review of the previous reports is a really good source for us so that we can avoid doing the same mistakes or reinvent the wheel.

There is a comparison of Tekkotsu and OPEN-R frameworks which is valuable. It is specified that OPEN-R is much low level than Tekkotsu which makes it harder and more time consuming to code. Also Tekkotsu-Mon GUI provides an easy media to control AIBO remotely. However, OPEN-R has a certain advantage because of the low abstraction: Performance.

The group specifies that when using OPEN-R only, since no unused code is being run on AIBO, data derived from sensors are evenly sampled which is not the case with Tekkotsu because of its excessive power usage. So Tekkotsu introduces us a burden of the necessity of resampling.

The group worked on the approach that compensates the vibrations using the neck-joints and for that purpose, they have identified 2 systems, one is from accelerometer to camera
position and the other is from the neck servo targets to actual neck position. And an ARMAX model was used. Unfortunately, our group does not have any control design background so we are having hard time to understand the all details about the process (also the sampling and interpolation part) so after trying to gain a little insight about these concepts, we will ask for your assistance. If we can, we will run and examine their code and hope that may ease the process of learning for us.

The group also offers some advices/hints that if implemented, may make their solution better. Some of which are:

- Avoid writing/reading to/from memory stick too much because of the stalls. Data can be buffered in heap segment and written as a burst for example. The group believes that they could not achieve constant data sampling (so had to resample the data) partly because of that reason. The other possible reason given is the use of Tekkotsu framework.
- Don't use Tekkotsu directly, instead try to use a hybrid way (performance of OPEN-R and platform capabilities of Tekkotsu) may be the best solution (but the group could not make it because they could not find a way to get rid of event handling mechanism of Tekkotsu.)