Funded by the European Union's
Seventh Framework Programme

BALANCE

Balance Augmentation in Locomotion, through Anticipative, Natural and Cooperative control of Exoskeletons

Deliverable 5.7

Evaluation of the Balance Supporting Controller

Due Date of Deliverable: **Month 48**
Actual Submission Date: **Month 55**
Partner Responsible**: CEA-L**

Classification:	[Confidential]
Grant Agreement Number:	601003
Contract Start Date:	01-01-2013
Duration:	55 Months
Project Coordinator:	TECNALIA
Partners:	TECNALIA, ICSTM, CEA-L, TUDLL, UTWEN, ETH, XSENS, URI
Project website:	BALANCE-FP7.EU

Table of Contents

1 Summary

The high-level control schemes were implemented in the platform-independent SL (Simulation Laboratory): this concerns the hip torque feedforward strategy and the VPP and FMCH scheme. This document reports the work done for implementing the SL inverse dynamics controller and the above mentioned high-level balance control strategies. The inverse dynamics controller was tested on the EMY exoskeleton but did not gave expected results due to several issues. These issues are discussed and the we draw lessons learned from the experience concerning the exoskeleton control and prototype development.

2 Derogation from Description of Work

As the described in the final report, the development of the EMY hardware platform, that was a required upgrade of the initial Hercule hardware platform, was severely delayed, and the type of validations foreseen when drafting the Description of Work of the BALANCE project have not been feasible. When this became clear, part of the work has been replanned, and validations of high level control concepts have also been carried out on other hardware platforms, especially the LOPES II at UTWEN facility, a treadmill-mounted robot that has similar capacities as an autonomous lower extremity exoskeleton, and the BAR-TM at URI facility, a treadmill mounted robot for support and training of balance, externally at the pelvis, providing assistance that can in principle also be provided by an autonomous lower extremity exoskeleton. The LOPES II was primarily used to validate control with healthy subjects, although validation has also been carried out on a few stroke subjects. The BAR-TM was used to implement concepts of balance training for stroke patients, applying BALANCE concepts to improve their balance function.

These validations carried out with other, ’secondary’ hardware, are reported in other Deliverables, especially:

- Deliverable D4.4 reports implementations of balance support in LOPES II on healthy subjects
- Deliverable D7.6 reports implementations of balance support in LOPES II on stroke subjects
- Deliverable D7.7 reports BAR-TM based training to stroke subjects

In principle, all these implementations of balance controllers on robot systems, should be considered as validations of BALANCE concepts, and therefore they are referred here. This specific Deliverable D5.7 solely reports what has been implemented on the EMY platform, which was in principle the primary validation platform in BALANCE; moreover, it reports the lessons learned in this process.

3 Controller integration in SL and EMY

The high-level control was implemented in the platform independent SL (Simulation Laboratory): this concerns the hip torque feedforward strategy and the VPP and FMCH scheme.

3.1 Integration SL-EMY

The integration of SL with EMY is based on the API developed in WP5 and described in deliverable 5.2.The SL software and interface was ported in Linux Preempt-RT patch, to be installed on the onboard computer of the EMY exoskeleton and run in “soft” real-time.
The API and low level control code were adapted to provide an acceleration estimation taking advantage of the low sampling (200µs) of internal control loop in the low level controller.

3.2 Integration EMY model in SL

The multibody dynamics model of EMY was implemented in SL for simulation purpose. In the exoskeleton model only the serial kinematic chains of the leg was considered (i.e. internal parallel mechanisms were not implemented). Also, the sole flexion was not considered and the contact at the foot was modeled as a flat four contact points. Note also that the SL inverse dynamics is considering the actuators as pure torque source, and that consequently the actuator inertia, due mainly to reflected rotor inertia, was not included in the model.
The inverse dynamics control was implemented in C++, using inbuilt functions of SL, RobCoGen and SVD from eigen library. Implementation details can be found in the Master thesis report enclosed in Annex in deliverable 5.4.
The underactuation and possibly unknown ground reaction forces issues in the inverse dynamics algorithm are handled by a pseudo-inverse formulation:

$$\boldsymbol{\tau} = \left(\boldsymbol{P}\boldsymbol{S}^T\right)^+ \boldsymbol{P}\left(\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}}_{des} + \boldsymbol{h}(\boldsymbol{q}, \dot{\boldsymbol{q}})\right)$$

where P is the orthogonal projection of the constraint Jacobian and S is the selection matrix of the actuated joints. Note that the constraint Jacobian is changing with the contact state.

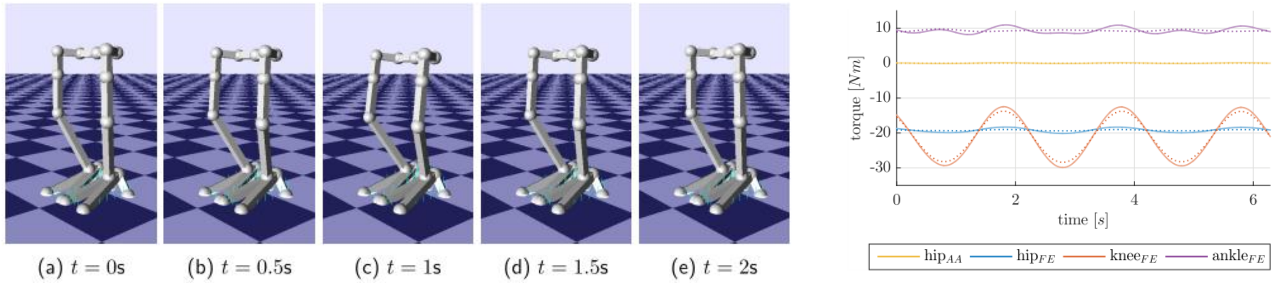
3.3 Simulation of inverse dynamics controller on EMY exoskeleton

The human movement was simulated as a reference trajectory $q_h(t)$ giving the reference acceleration to the controller. An impedance feedback controller was making the simulated robot to follow the reference trajectory. Two reference trajectories were simulated:

- a squat motion consistent with the contact constraints (4 points on each foot)
- -a leg lifting motion with hip, knee and ankle flexion/extension and the other leg staying on the floor, violating the initial constraints of contact of the foot.

In the SL simulation some artefacts e.g. increasing the feet size were used in order not to take care of balance, the reference trajectories being not generated to respect balance.

The simulation with the first trajectory allows us to check that the inverse dynamics controller was generating similar torque than the impedance controller (Figure 1).



dotted line are torques from inverse dynamics while the solid lines are the torques from the impedance controller

Figure 1 Simulation of a squat movement

The simulation with the second trajectory showed that the inverse dynamics controller was resisting to the change in floor contact configuration. In the simulation (Figure 2) the inverse dynamics controller was not able to produce the motion and the impedance controller had to fight against it. This shows that the fact the algorithm relies on the current contact configuration to apply reaction forces to compensate for gravitational and inertial forces may also prevent the user from lifting the leg. The consequence of this will be discussed in §4.1.

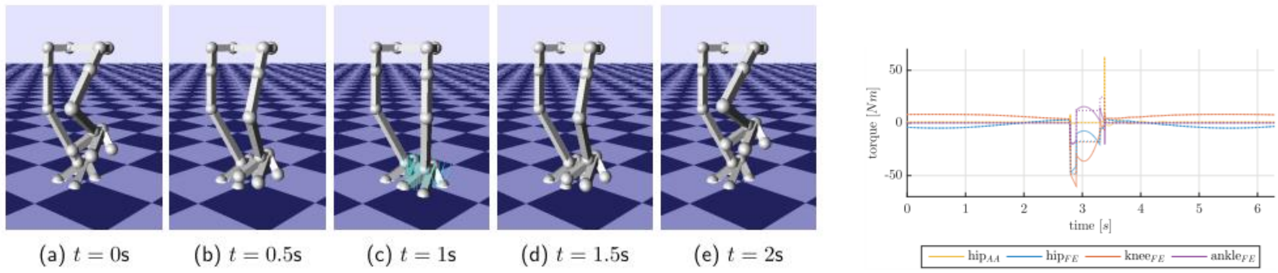


Figure 2 Simulation of a leg lifting movement

4 Lessons learned about balance control implementation issues

The inverse dynamics controller was tested on the EMY exoskeleton but did not gave expected results due to several issues. Some of these issues could be solved, such as the improvement of acceleration signal, while some could not be fixed in time, such as a calibration of the inertial parameters, some more others are fundamental issues such user intention management in switching contact condition.

We try in this part to draw lessons learned from the experience concerning the exoskeleton control while in the following part (§5) we will address the prototype development.

In part §4.1 we discuss the user intention issue, in part §4.2 we discuss the limitation of inverse dynamics in EMY hardware.

The balance controllers proposed by some partners and tested on the LOPES exoskeleton were also implemented in a preliminary version of EMY exoskeleton and this work is reported in parts §4.3 and §4.4 with the discussion about implementation issues for these control strategies.

4.1 User intention against inverse dynamic control

The simulation of inverse dynamics control put in evidence that any strategy trying to get an equilibrium either static or dynamic tend to prevent the user from moving freely when contact situations need to be switched e.g. lifting the feet (§3.3).

This problem not encountered in master arm or arm exoskeleton, is specific to walking exoskeleton, and more specifically to walking exoskeleton, when carrying its own load and allowing the user to be the pilot through transparency control.

When the exoskeleton supports itself, it must exert extension torques on the ankle and knee flexion, which would have to be overcame by the user in order to take a step. Thus, to allow the user to take a step freely the exoskeleton controller has to change its contact state before the leg is actually lifted. This would require a user intention detection and an adaptation of the controller.

Surprisingly this is to our knowledge, not discuss in the scientific and technological literature. For instance, BLEEX exoskeleton existing publications do not describe how they manage this issue. With the HERCULE exoskeleton, CEA-L indeed encountered the same issue and solved it using force sensing between the user sole and the exoskeleton together with a control law adapting to this sensing. No publication was done at the time of its implementation, being kept confidential. A patent is actually pending (FR1656377 - EXOSQUELETTE AMBULATOIRE PROCEDE DE COMMANDE D'UN EXOSQUELETTE AMBULATOIRE, submitted 4/7/2016).

Solution to this issue could be based on algorithm for gait initiation detection based on IMU or sensors fusion. Balance observers as developed in WP3 of the project could also be adapted and used in some case to predict a step.

4.2 Inverse dynamics control

The balance control was chosen to rely on two levels of controllers:

- A low-level torque control of actuators[1]
- A transparency model-based controller, based essentially on a floating base inverse dynamics control formulation.

To test the efficiency of inverse dynamics, the SL- high level control was first put in position control, then torques from inverse dynamics were introduced and a decrease in the position control feedback was observed. Although this was the expected behavior, the magnitude of this torque reduction was small and considered to be insufficient.

The inverse dynamics can be affected by different non-ideal behavior of the system. The main sources of errors are: model uncertainties, torque tracking performance, state estimate.

Considering <u>torque tracking</u>, EMY actuators have good bandwidth, but even with back-drivability some static friction is present. The static friction was compensated as a feedforward term function of the velocity sign, the transition during velocity sign changes being filtered by a nonlinear function. The inverse dynamics while feeding back position signal tends to reduce the stability of this friction compensation whose value must be reduced.

[1] Joint position control was also available but was used only for identification purpose.

The static friction observed on the hip actuators was within the range we can expect from this type of actuators with 2% of the static torque capability. In the lower leg we could observe up to 9% in the worst case configuration. Also as the friction varies depending on the configuration and on the combined movement of the two actuators in the four-bar linkage, the constant feedforward compensation of the static friction was limited to the smallest value of friction.

The attempt to design a torque observer which could be feedback was not successful in designing a stable controller. With more resolution on the joint position sensors and a more adapted design of the controller we can expect still to improve the torque control resolution. However, it is needed also to adapt the inverse dynamics control to the performance of the actuators. Recent work by Del Prete and Mansard is dealing with this issue in the context of task-space inverse dynamics .

A. D. Prete and N. Mansard, "Robustness to Joint-Torque-Tracking Errors in Task-Space Inverse Dynamics," in *IEEE Transactions on Robotics*, vol. 32, no. 5, pp. 1091-1105, Oct. 2016

Considering state estimate, the inverse dynamic controller was getting the joint position and velocities from the low-level controller. In this version of the low-level controller the joint position was given from the motor encoder. The advantage was to have a high resolution and to take advantage of a 200µs sampling to derive the velocity. A numerical calculation of the acceleration was added in the low-level control and included in the API to improve the inverse dynamics.

The actual joint and motor positions can differ due to the joint stiffness: observed deformation in static configuration was reaching 0.03 rad for the knee joint, 0.02 rad for hip flexion/extension, 0.01 rad for hip abduction/adduction. Another source of error in the position was the offset initialization: the low-level controller was reading the joint potentiometers at the initialization of motor control and using them to define an offset in the motor position (reflected at the joint output). Thus depending on the configuration especially on the loading of the joints we would have some systematic error in the position (in the limit of deformation). These limitations of the current prototype will be discussed in part §5.

Another important issue on the state estimation is the measurement of the torso position (in our case the back segment). The height of the torso was indirectly obtained by the legs kinematics, while its absolute orientation was derived from the measurement of an IMU present in the exoskeleton back. Data fusion (encoders + IMU) could be used to improve the estimation of the torso state, including its velocity and acceleration.

Considering the uncertainties in the model, the mass properties of segment are the main concern. Indeed, the actuators are stiff and the mass properties can differ significantly for the parameters derived from CAD data. The identification of these parameters from the joint torques (derived from current measurement) has been treated in deliverable D5.3. In order to reach first a gravity compensation, we focus on the first momentum of mass parameters ($m_i\ \underline{O_i G_i}^{(i)}$). When the base is fixed and the leg are moving we identified some specific grouping of the moving segment parameters. The friction is part of the joint torque model and we include the load dependency of the static friction to prevent a bias on the mass parameters identified. The identification of the mass parameters acting in the hip joints was successful. Concerning the lower leg, the friction model was extended to include a static friction torque on the unactuated knee joint. The estimate of the mass parameters of the lower segment is probably bias by the friction variation, the internal friction and loading inside the kinematic chain and interim movements inside the kinematic chain. However, these parameters are low and this bias should not impaired inverse dynamics.

More problematic was the identification of parameters in double (or single) support. We imagine a procedure based on double support standing but isostatic. However, we underestimate the practical issue of generating calibration trajectories without falling. This problem is indeed encountered in humanoid robotics for which most of the work reported were dealing with a robot having a balance controller already implemented but still comment on the difficulty to generate identification trajectories. Some solutions are envisaged that should be taken into account in the design phase of the exoskeleton (see §5.1); solutions with external force measurement could improve the precision of the identified parameters. Finally, the practicability of hybrid testing configurations based on introducing constraints to ensure the stability of the exoskeleton while eliminating the associated constraints forces from the observed system equation.

4.3 Feedforward assistive torque control

The feedforward assistive torque was implemented for hip flexion/extension. The tests were not relevant for evaluation of the control as the exoskeleton was not self-supporting with an inverse dynamics control, which was more a perturbation on the operator. Inverse dynamics control was not performing well as explained above, but in addition, the supporting rope was introducing an important perturbation.

For future development of exoskeleton and balance control, the feedforward hip torque assistance could be used in different type of exoskeleton or hip active orthosis, provided that hip movement are torque controlled. With combining hip flexion/extension and abduction/adduction this hip assistance should be based on a foot step prediction. A critical point in this method is the detection of the unbalance and need for stepping. However, if the level of assistance is moderate, a detection error would not cause the user to step but would make more resistive the hip extension and mode easy the flexion. Analyzing and testing the effect on user in case of false detection is an open issue for which we could be easier to test with an active hip orthosis than with a full exoskeleton.

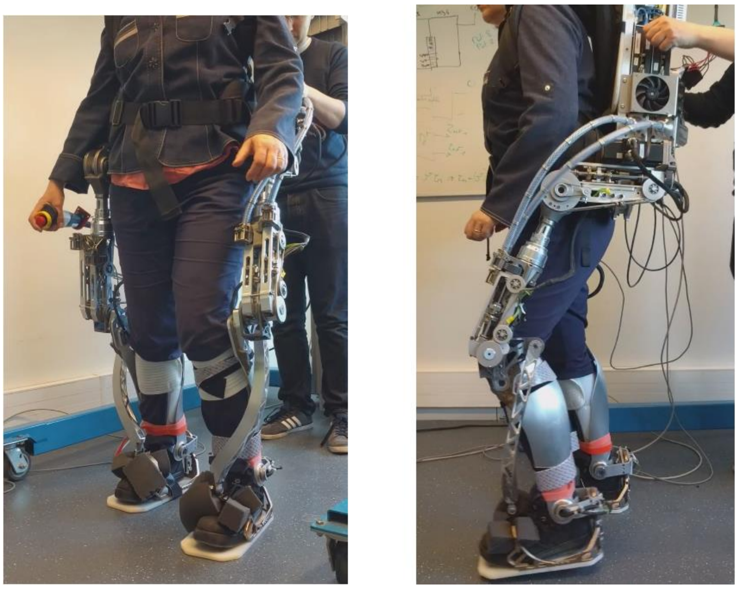


Figure 3 Preliminary tests in the EMY exoskeleton with the hip assistive torque

4.4 VPP based control

The control derived from the VPP is applying hip (and possibly knee) flexion/extension torque during the stance phase based on an estimation of the two legs forces. In the implementation tested on LOPES exoskeleton this estimation was derived from force plate measurement and the leg positions.

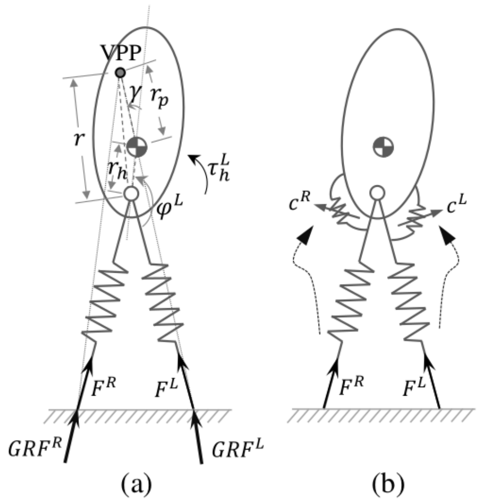
Figure 4 Structure of the VPP based controller implementation on exoskeleton

In an ideal implementation the VP based controller would use ground reaction force under each foot. In the EMY implementation the VPP controller, the center of pressure was assumed to be at (under) the ankle joint center.

Leg force F was estimated based on leg angle (assuming vertical component of each leg force is half body weight):

$$F = |\mathbf{p}_h - \mathbf{p}_a| \frac{W/2}{z_h - z_a}$$

where $\mathbf{p}_h$ and $\mathbf{p}_a$ denote the position of hip $(x_h, y_h, z_h)^T$ and ankle joint in the global Cartesian space, $\mathbf{z}$ denotes the vertical direction, W denotes the body weight.



$$(x_a, y_a, z_a)^T$$

Figure 5 the VPP (a) and FMCH (b) model used for the controller

The VPP equation is

$$\tau_h = Fl \frac{r_h \sin\varphi + r_p \sin(\varphi + \gamma)}{l - r_h \cos\varphi - r_p \cos(\varphi + \gamma)}$$

where τ_h, F, l, φ, r_p and r_h denote individual leg hip torque, leg force, leg length, virtual hip angle (the angle between trunk axis and the vector from CoP to hip joint), and the distance from CoM to

VPP and hip joint respectively. γ denotes the angle between trunk axis and the vector from CoM to VPP.

The parameters r_p, r_h and γ were set with constant values based on previous studies.

The FMCH equation is

$$\tau_h = cF(\varphi_0 - \varphi)$$

where c and φ_0 denote hip spring stiffness (constant, normalized to body weight) and rest angle, respectively. Leg force F was estimated in the same way as the VPP controller.

For the EMY test, c and φ_0 were set with constant values based on previous studies.

The desired exoskeleton hip torque is then the torque control input $\tau_{hExo} = \eta\tau_h$

where η is the gain for adjusting hip assisting level.

A quick trial realized on EMY exoskeleton was not successful as instabilities were observed. This may have to do with the absence of damping in the controller and possibly other hardware issues.

The VPP based balance controlled could be implemented with an active hip orthosis together with some monitoring of the leg positions and measurement of the foot ground reaction force. The measurement of foot-ground full reaction force is however expensive and possibly bulky. It is why could be explored the feasibility of a partial measurement with two envisaged options:

- measuring the 3D or vertical force with assumption of passive ankle
- measuring the cop from plantar pressure measurement with an extrapolation on the body weight repartition between the two feet.

5 Lessons learned about Prototype development

The prototype development was ambitious in term of specification. Several new development of the CEA LIST laboratory were included: type B screw cable system actuators, parallel linkage for ankle actuation, remote motorization with flexible shafts, low level control board.

5.1 Mastering the mass of the exoskeleton

It appears that to have an exoskeleton carrying its own load is a strong specification. The total mass is 50kg with about 20 kg for the back. The increase in mass is reducing the effective torque available and can lead to saturation in the control. However, this was not encountered in the preliminary tests realized.

The good mastering of the mass is important for two issues: reducing the mass during the design and in the redesign and providing good estimate of mass and inertial parameters for the controller.

The complexity of the actuators drawing in the CAD make it difficult to have a CAD model structuration fitting perfectly with the kinematic segments. However, it would be possible to improve our estimate. Including all the data on material and components weight is time consuming for the designer and neglected by the designers under time constraints. We encounter problems also with some standard components library of the CAD system. From experience in other projects inside and outside the laboratory the CAD prediction of mass parameters can be improved but need anyway to be checked on the realized device.

For checking and identification of the mass parameters some procedure should be established during the mechanical assembly. If we want to get the position of the com and not just the mass of the subassemblies, good geometrical reference need to be defined.

However identification from the joint torque remains essential. Tests by module would be the more efficient, with for instance testing the back moved by the hip joints while the upper thigh is frame fixed. This issue could be included in the design requirement and the special tools for holding intermediate segment designed together.

Finally as identification in double or single support is needed, fixing the feet in order to use a wider identification movement set is an option that put important loading in the ankle and foot mechanical structure. A practical solution for the foot would be to have a special (heavy) foot for testing.

5.2 Mechanical issues

The new screw-cable system actuator reaches the design objective and was stiff enough to reach the required torque control bandwidth of 20 Hz with a simple low level controller. The increase in TRL is continued in the laboratory with especially endurance testing, stabilization of assembly procedures and adjustment, in the context of other projects.

Due to a too tight adjustment several pieces needed to be modified or remanufactured. While looking for compactness, some room must be maintained to allow for deformations and small oscillations without lost of performance.

The actuation four-bar linkage poses several problems:
Friction and internal loading in the four-bar mechanical linkage of the lower leg were detected. They need to be better characterized. The unactuated joint was realized as a plain bearing and we were planning to replace it by a ball-bearing. However, the load transferred in the ankle structure need to be reassessed and may prevent us to effectively use a ball bearing.
The four-bar linkage is also making difficult to wear the exoskeleton. This could be improved by design but will stays a drawback with respect to serial linkage.
Considering these problems and the possibility to reduce the weight and dimension of actuators, the gain in the leg inertia provided by this design is questionable. However, this question is also dependent on the performances we will achieve with the flexible shaft that allows to relocate the motors higher in the exoskeleton.

The friction in the flexible shaft appears to be too high and in collaboration with the industrial manufacturer we are working on it.

The metal 3D printing of several pieces has been successful for the more massive segment using Titanium alloy. However, in the smaller parts, located in the actuators and produced with Stainless Steel Alloy infiltrated with Bronze, we encountered several damages and the parts were redesigned for iron steel and standard manufacturing.

5.3 Software, control and sensors issues

The global control software architecture was integrated and test successfully. A practical feature is that is allows to switch controller and users easily, allowing several control projects to use the exoskeleton. However some technical issues still need to be solved to improve the efficiency during development and testing, and to achieve the best performance. These issues are listed bellows.

At the initialization of the system, two low level operation are executed:

- The alignment of motors rotors
- The initialization of motor position offset from the current joint potentiometers reading.

The alignment of rotors sometimes fails while the motor shaft is too loaded to move freely. The failure of the procedure is not detected and produce trouble as the current control is not correctly calibrated. The current profile during this operation is showing the problem and could be used for an automatic detection.

Currently only the position motor is used during the control mode and is returned as joint angle. So at each initialization, the effective calibration of joint angles is lightly changed (within the range of deformation happening between motor and joint output).

These two issues would be avoided using absolute motor encoders.

In the current control of motors, the compensation of Torque ripple was not implemented due to some issues with the memory management of the control chards. This issue being solve this should be corrected. The oscillation observed is about ±1 Nm at the joint output.

The observation of joint position is derived from the motor position encoder: this ensure a good resolution. Although for calibration and identification purpose, continuous reading of the joint potentiometers would be useful.

The low level controller was developed using Simulink C generator. Although this developing environment is complete and professional, once the four control chards are embedded in the complete system, the debugging becomes difficult, as we do not access all the variables in the algorithm. This solution appears to be convenient for limited testing of simple prototype and possibly for developing safe and stable control chard intensively tested on test bench. For the use we have with complex robot prototype, we would prefer a direct C programming environment.