Remote Control of AAU^3 Cubli

Bjørn Kitz, Mikael Sander, Florian Franco and Ignacio Trojaola

***Abstract*—In this paper we investigate the possibility to control the AAU^3 Cubli over a wireless network and how much packet loss and induced delay the controller can handle. First, we introduce the dynamics of the system using the Lagrange equation. Following that, we derive a state-space model and compute a linear state feedback to control the AAU^3's angle. Then, we simulate our controller over a WLAN network using the TrueTime Matlab toolbox in order to prove that the Cubli can be controlled over a physical network. The obtained linear feedback controller is able to balance the Cubli over a WLAN network.**

***Index Terms*—Inverted Pendulum, Reaction Wheel, Linear Feedback, Network Control**

I. Introduction

The AAU^3 Cubli project is based on "The Cubli" project made by the ETH Zurich and here at AAU the one-dimensional test setup, that was used to test control algorithms, has been replicated [1]. The Cubli is a three dimensional cube that can jump up and balance on its corner. This ability to remain in an equilibrium position is possible by controlling the torque of the reaction wheels mounted on three of its faces, one for each axis. Control laws, tilt estimation and other algorithms are computed locally in the on-board microcomputer.

The inverted pendulum is a common system to work with in control engineering [2]. The AAU^3 Cubli is a form of inverted pendulum, but instead of the classical version where the pendulum is actuated at the end of the stick, the Cubli has a reaction wheel mounted that is placed in the middle of it.

The local control of the Cubli has already been done in a previous project using a state-space controller [3], that runs on a microcomputer (BeagleBone Black) on Cubli. The control objective of this paper will focus on controlling one or more Cubli's over a wireless connection, determining how much degradation of the wireless connection can occur, resulting in data packages lost or delayed, before the control loop is no longer stable.

The Cubli itself is just a learning setup, but the idea behind it can be used for other applications. A small robot could use the reaction wheels to move around in an environment with lower gravity than here on earth. An idea would be having a swarm of cubes that move. Controlling all of these over a network would make it possible to adjust the controllers during operations and have the cubes clear obstacles or solve problems. A single Cube controlled over a network can function as a extension of a main unit. A possible advantage of the Cubli being a cube is that a robot could be constructed without having any external moving part and thus not be affected by particles getting into its propulsion system.

In an educational or demonstration context, it could be interesting to control multiple Cubli with the same or different controllers.

This article is structured as follows : section II introduces the dynamics of the one-dimensional Cubli using the Lagrangian mechanics. In section III, the state-space controller will be derive. In section IV, it is explained how data is send over the network. Section V presents the TrueTime model of the network control. In Section VI the test of the system are shown.

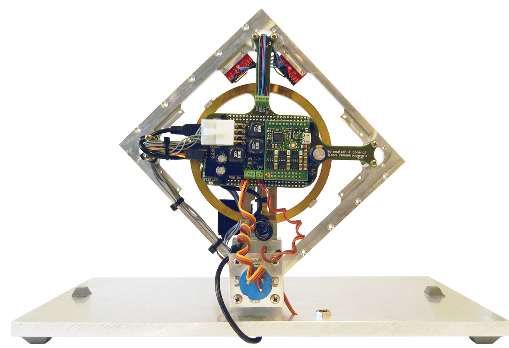

Fig. 1. Picture of the AAU^3 setup with the frame placed in the equilibrium position.

II. Model of the one dimensional Cubli

Fig.1 shows a picture of the Cubli, which consists of one dimensional aluminium frame attached from one of its corner to a base plate, that gives it a single degree of freedom. A reaction wheel, controlled by a brushless DC-motor, is also fixed to the center of the frame. On the opposite side are the embedded electronic components, Inertial Measurement Unit (IMU) and the motor control-board (connected to the microcomputer) attached to the frame.

The operation of the system is that the controller determines the torque needed and converts it to a current signal, that gets applied to the motor through the motor control board. The wheel then starts to accelerate or decelerate. The angle of the frame is determined by measuring the speed of the frame and the gravitational acceleration acting on the frame. This is done with the IMU.

A. Dynamics of the Cubli

In order to obtain the dynamics of the model, different procedures have been followed such as the Newton's Law of motion [3] and rotation in or Kanes equation [4]. In this work, the Lagrange's equation is used to obtain an energy model of the Cubli. The Lagrangian function describes the motion in a mechanical system by means of its potential and kinetic energies.

$$L = T - V \tag{1}$$

L	Lagrangian	[J]
T	Total expression for kinetic energy	[J]
V	Total expression for potential energy	[J]

Finding the expression for the total kinetic and potential energy.

$$T = \frac{1}{2} \cdot J_F \cdot \dot{\theta}_F^2 + \frac{1}{2} \cdot m_\omega \cdot {l_\omega}^2 \cdot \dot{\theta}_F^2 + \frac{1}{2} \cdot J_\omega \cdot (\dot{\theta}_\omega + \dot{\theta}_F)^2 \quad (2)$$

$$V = (m_\omega \cdot {l_\omega}^2 + m_F \cdot {l_F}^2) \cdot g \cdot \cos\theta_F \quad (3)$$

where m_F, m_ω are the mass of the frame and the wheel, J_F is the moment of inertia of the frame around its center of mass, J_ω is the moment of inertia of the wheel around the point where it is fixed to the frame, l_F is the length between the corner of the base plate to the mass center of the frame, l_ω the length between the corner of the base plate and the rotation point, g is the gravitational acceleration, B_F and B_ω are the frictions acting on the frame and the wheel and τ_m is the torque produced by the motor which is opposite to the direction of the frame.

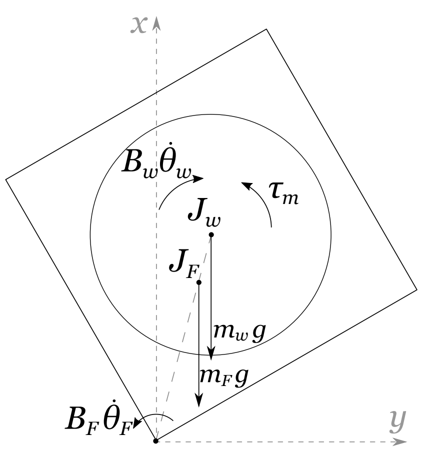

Fig. 2. Sketch of the AAU^3 Cubli setup with parameters.

The general expression of the Lagrangian model is:

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\theta}_i}\right) - \frac{dL}{d\theta_i} = F \quad (4)$$

By differentiating equation (4) with respect to the frame and the wheel and assuming θ_F and θ_W represent the angular position, we find the non-linear dynamics of the setup:

$$\ddot{\theta}_F = \frac{(m_\omega \cdot {l_\omega}^2 + m_F \cdot {l_F}^2) \cdot g \cdot \sin\theta_F - B_F \cdot \dot{\theta}_F + B_\omega \cdot \dot{\theta}_\omega - \tau_m}{(J_F + m_\omega \cdot l_\omega^2)} \quad (5)$$

$$\ddot{\theta}_\omega = \frac{(J_F + J_\omega + m_\omega \cdot l_\omega^2)(-B_\omega \cdot \dot{\theta}_\omega + \tau_m)}{J_\omega(J_F + m_\omega \cdot l_\omega^2)} - \frac{(m_\omega \cdot {l_\omega}^2 + m_F \cdot {l_F}^2) \cdot g \cdot \sin\theta_F - B_F \cdot \dot{\theta}_F}{J_F + m_\omega \cdot l_\omega^2} \quad (6)$$

All the paramaters are obtained by measurements, while the inertia and the friction of the wheel are determined using the Matlab toolkit SENSTOOL. The table in appendix presents all the measured and identified parameters of the model.

B. Linearization of model

Since the system is non-linear, the linearization of the equation (5) is made using small-angle approximation. The operating point is defined as the frame being in an upright position, with the wheel not turning. That yields that $\theta_F = 0$, $\dot{\theta}_F = 0$ and $\dot{\theta}_\omega = 0$. With no change in position and no velocity or acceleration, the torque acting on the wheel can be assumed to be zero also.

III. State-Space control

With the two equations that describe the acceleration of the wheel and the frame a state-space model of the system is made.

$$\dot{x} = Ax + Bu \quad (7)$$

where $x = (\theta_F,\, \dot{\theta}_F,\, \dot{\theta}_\omega)$.

This state-space model is compared to measurement data taken from the Cubli in order to confirm that the model describes the behavior of the Cubli and can be used as a basis for a controller. The results are found in section VI.

To discretize the continuous space-state model, a sampling time of 20 ms and the Zero Order Hold discretization method are used, resulting in the following state-space representation:

$$x(k+1) = A_d x(k) + B_d u(k) \quad (8)$$

This yields the following A_d and B_d matrix.

$$A_d = \begin{bmatrix} 1.015 & 0.02 & 3.904 \cdot 10^{-7} \\ 1.514 & 0.9997 & 3.904 \cdot 10^{-5} \\ -1.513 & 0.0002645 & 0.9994 \end{bmatrix} \quad (9)$$

$$B_d = \begin{bmatrix} -0.02293 \\ -2.293 \\ 35.56 \end{bmatrix} \quad (10)$$

where $u(k)$ is the input and $x(k)$ is the state vector.

To guarantee the stability of the discrete closed loop system the poles have been placed inside the unit circle of the z domain, what makes the chosen poles as:

$$p = [\,0.8869 \;\; 0.8694 \;\; 0.7408\,] \quad (11)$$

The networked controller is designed as $u = Fx$, where the state feedback gain matrix F being as follows:

$$F = [\,2.3190 \;\; 0.2616 \;\; 0.00038\,] \quad (12)$$

The controller requires the three states of the Cubli which are measured directly by the sensors.

IV. Network

Networked control systems involves that information (plant output, control input, etc.) being exchanged over a network between systems components: sensors, controller and actuators [5]. The structure of a such system using a Ethernet network is shown in Fig.3.

In order to control the Cubli through the network, control functions are executed by another computer connected to the local network.

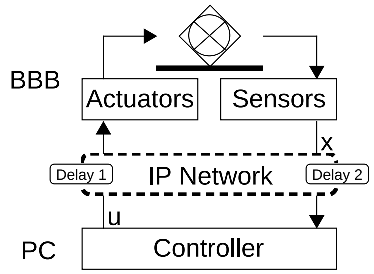

Fig. 3. Chart of the Networked control system between BeagleBone and PC

Based on the data measured by the sensors, the states of our model can be calculated. Assuming we send the three states and the packet number, we wrap all this information in a 32-bytes packet since each variable is coded on 8 bytes. This packet grows up to 72 bytes when headers are included. Once the control function is computed, the resulting torque is multiplied by the motor constant to get the current. Then, the controller sends back the current, in a 8-bytes message, without header. We apply the current value to the motor in order to remain stable at the equilibrium point.

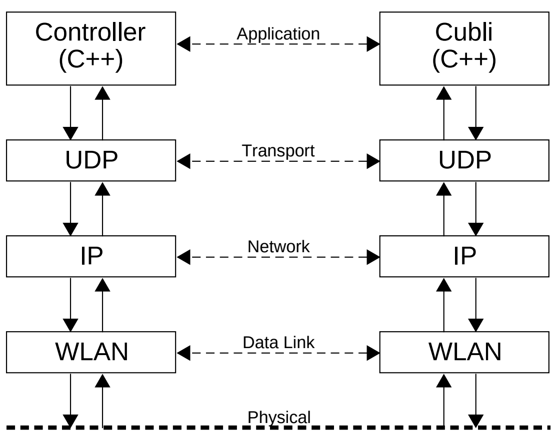

Fig. 4. Sketch of the protocol used when transmitting data between BeagleBone (Cubli) and Computer (Controller)

Fig.4 shows the protocols used to implement the experimental system. Physical and link layers use the IEEE 802.11 wireless LAN protocol. The network layer uses the IP protocol. Transport layer uses the UDP protocol, instead of the most common TCP protocol. Whereas the TCP protocol is connection oriented and reliable, the UDP protocol provides a faster transfer speed because it doesn't guarantee the message to reach its destination and doesn't provide any data flow control service. Moreover, in a digital control system, sensors are typically sampled at a constant rate [6], which means that sensor readings are transmitted at the same rate. For our application, a sampling time of 20 ms (50 Hz) is used.

A. Description of communication protocol

The Controller is set to wait for a initial message from the Cubli in order to start a control thread. It is possible to service multiple Cubli's since the controller is multi-threaded and keeps accepting new clients.

Once the Cubli has a response from the assigned controller thread, it starts periodically sending data packets containing its states. After sending a packet the Cubli has a listen phase (7 ms) where it receives the feedback from the controller. The controller is event triggered and waits continuously for a data packet. Upon recipe of a packet the control signal is computed and send back. An overview of the Cubli loop can be found in the appendix as Fig.10

B. Packet delay on the Network

Packet that are delayed on the network and arrive after the listening period are discarded.

V. TrueTime

To test if the control works properly when moving it away from the Cubli and over Wi-Fi network, a Simulink model of the system is implemented using the TrueTime Toolbox [7]. In the model, a bidirectional network transmission delay is added.

The network model consists of two units on a wireless network represented by the third block in Fig.5. The network chosen to simulate is 802.11b (WLAN) and its own parameters are changed to emulate a connection using the UDP protocol. This is done by shutting down all retransmission and acknowledge transmission. The TrueTime model is compared

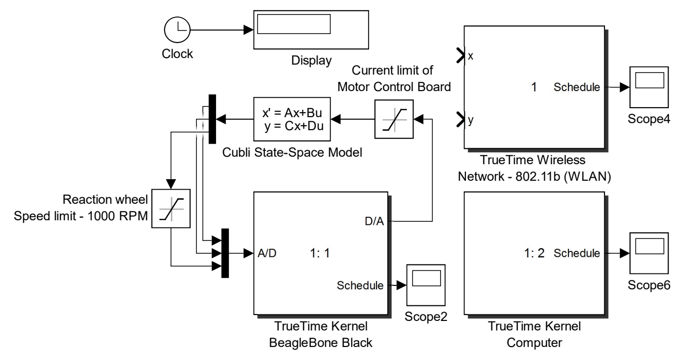

Fig. 5. The TrueTime block system of the network simulation. BeagleBone unit is the one connected to the state-space block.

to measurement data to determine how reliable the simulation is. Details of the test are in section VI-B. Looking at the result from the comparison the TrueTime model can be used to predict the behavior of the Cubli being controlled over the WLAN.

Then, to ensure that the TrueTime simulation matches approximately with the running code and the network behavior, we include measured parameters like mean delay, mean execution time and packet size.

VI. Test / measurement

A. Validation of state-space model

To validate the state-space model, the fall test is used. The model Cubli's behavior is simulated without a controller acting. This is compared to a measurement of the Cubli under similar conditions. The behavior chosen to compare is the one of the falling frame, where the frame is placed a few degrees away from the equilibrium point, and then allowed to fall over without any controller active.

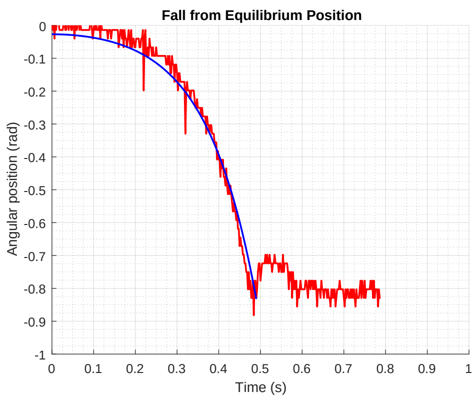

Fig. 6. Comparison of the falling simulation of the model (blue curve) against the measured behavior (red curve).

Looking at Fig.6 shows that the simulated fall response and the measured fall response match. The conclusion of this is, that the model can be used to describe the behavior of the Cubli when it falls.

B. Simulation of Network Control with TrueTime

This test is done with the state-space model of the Cubli integrated with the TrueTime network model. For this test the Cubli is placed at an angle different than the equilibrium point and then let go and balanced by the controller. The initial angle from the measurement (0.0873 rads) is set as the initial condition for the TrueTime model, and a simulation of the Cubli's controller over the network is done. The results of these two tests are compared in Fig.7 to see how close the Cubli network behavior is to the model implemented TrueTime.

C. Impact of network delay

To determine the impact of the time delay of the network we observe the Cubli behavior in a TrueTime simulation. The simulated delay is assumed to be equal on both directions. The Fig.8 shows the velocity of the wheel of the test with different time delay. Incrementing the delay no remarkable changes can be seen in the behavior of the wheel, however, when the delay reaches 18 ms the control is not sufficient to maintain the Cubli in an upright position and the velocity of the wheel increases over the time.

Fig. 7. Comparison of the angle of the frame obtained in TrueTime simulation of the model (blue curve) against the measured angle in Cubli (red curve).

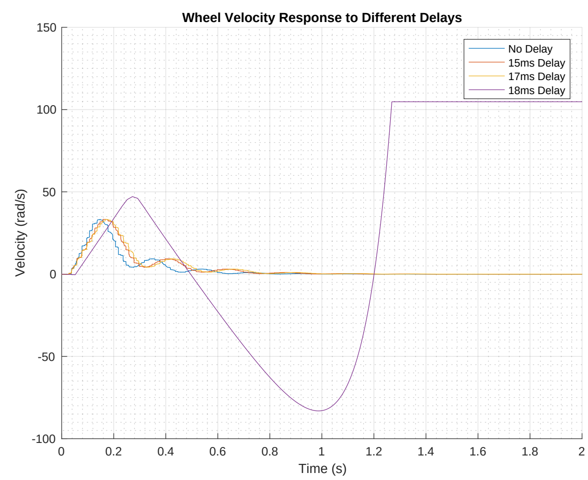

Fig. 8. Different Cubli behavior shown on the speed of the wheel with varying network delay. The listed delays are the total delay of the network sending and receiving data.

D. Round-trip time and packet discarded measurement

The round-trip time is the time it takes to send a data packet to the computer until the control signal is received on the BeagleBone. The experiment is done by saving a time-stamp before the data is send and then comparing it with a times-tamp taken when the control signal is received from the computer.

The distribution of the packet RTT can be seen in Fig.9. The packets arriving before 2 ms are suspected to be delayed more than one period, and are not counted in the statistical results presented right after. It shows that the average RRT of a packet is 3.3 ms whereas the median RTT is 3.04 ms. This shows that the data transfer takes less than half of the sampling time of the Cubli. During the test the rate of discarded packets was also measured. The average packet discarded over 10 tests (40 s each) was 2.9 %.

VII. Discussion

The results of the test with the state-space controller show that the controller is capable of controlling the Cubli over a

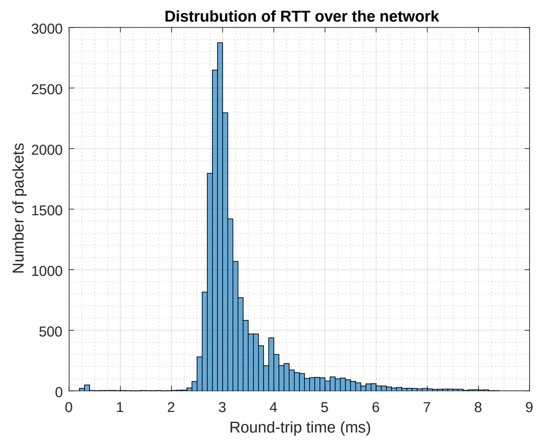

Fig. 9. Histogram over the round-trip time of the packets going from the Cubli to the Controller and back to the Cubli again

network, as long as the delay of packet between Cubli and Controller do not go over a certain threshold.

Simulations show that the theoretical delay of the network cannot be more than 18 ms before the controller, without predictor, is unable to control the Cubli. For our setup we do have an additional restriction in the form of the communication protocol that has been implemented. The fact that the packets arriving later than the listening are discarded puts the maximum delay that can be handled continuously at this time threshold. Repeatedly delayed packet will in this case all be discarded and the Cubli is without control. Depending on the state it was in it might be able to catch itself, once the networks conditions get better or it will have fallen over.

VIII. Conclusion

A model of the AAU[3] Cubli has been derived with the Lagrangian mechanics. Based on the expressions of the frame's and the wheel's acceleration, a state-space model has been designed. The results showed that a linear feedback controller can be use to control a networked control system such as the Cubli. The induced delay by the WLAN does not prevent the control system from working. It is able to keep the Cubli around its equilibrium position.

Acknowledgment

The authors would like to thank Simon V. Johansen for discussing his experience with network on the Cubli with us.

References

[1] M. Gajamohan, M. Merz, I. Thommen and R. D'Andrea, *"The Cubli: A cube that can jump up and balance"*, 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems, Vilamoura, 2012, pp. 3722-3727. doi: 10.1109/IROS.2012.6385896

[2] Kent H Lundberg and Taylor W. Barton, *History of Inverted-Pendulum Systems*, Elsevier, 15 April 2016, http://www.sciencedirect.com/science/article/pii/S1474667015316049 (url date: 21-11-2016)

[3] J. Brehin, N. Vestergaard, N. Villarmarzo, B. Kitz and M. Sander, *Cubli:Dynamic Control of a Reaction Wheel Inverted Pendulum*, Aalborg University, Denmark, 2016.

[4] M. Gajamohan, M. Muehlebach, T. Widmer and R. DAndrea, *The Cubli: A Reaction Wheel Based 3D Inverted Pendulum*, ETH Zurich, Switzerland, 2013.

[5] W. Zhang, M. Branicky and S. Phillips, *Stability of Networked Control Systems*, 2001.

[6] G. Alt and W. Fetter, *Networked Robot Control with Delay Compensation*, Federal University of Rio Grande do Sul, Brazil.

[7] Anton Cervin, Dan Henriksson and Martin Ohlin, *TrueTime 2.0 - Reference Manual*, Department of Automatic Control, Lund University, Sweden, February 2016.

Appendix

A. Parameter identification

g	9.82 m·s^2
m_F	0.548 kg
m_ω	0.222 kg
l_F	0.085 m
l_ω	0.093 m
J_F	$6.8 \cdot 10^{-3}$ kg·m^2
J_ω	$0.601 \cdot 10^{-3}$ kg·m^2
B_F	$5.7 \cdot 10^{-3}$ N·m·s·rad^{-1}
B_ω	$17.03 \cdot 10^{-6}$ N·m·s·rad^{-1}
K_t	$3.35 \cdot 10^{-3}$ N·m·A^{-1}

B. Cubli system loop

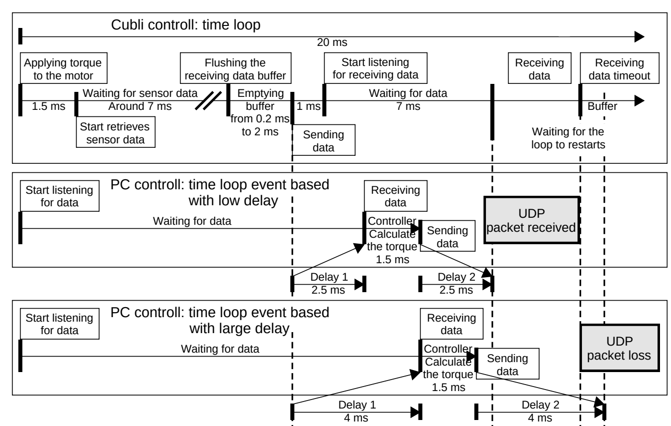

Fig. 10. Cubli System loop. This is a timeline over the different functions to execute in the loop that runs on the Cubli. The upper loop shows theoretical loop. The middle shows what happens with a low delay on the network. The lower one shows what happens with a high delay on the network.