BALANCE CONTROL STRATEGY FOR YU-BIBOT

by

Can Polat Çıgay

Submitted to Graduate School of Natural and Applied Sciences
in Partial Fulfillment of the Requirements
for the Degree of Master of Science in
Mechanical Engineering

Yeditepe University
2018

BALANCE CONTROL STRATEGY FOR YU-BIBOT

APPROVED BY:

Assoc. Prof. Dr. Koray Kadir Şafak ....
(Thesis Supervisor)

Assoc. Prof. Dr. Zeki Yağız Bayraktaroğlu ....

Assist. Prof. Dr. Namık Çıblak ....

DATE OF APPROVAL: ..../..../2018

ACKNOWLEDGEMENTS

I would like to express my very great appreciation to my advisor Assoc. Prof. Dr. Koray Kadir Şafak for his guidance.

Also, I would like to thank to Yeditepe University and Department of Mechanical Engineering for education and opportunity to complete my master degree.

Lastly, I would like to thank to my family with their support.

ABSTRACT

BALANCE CONTROL STRATEGY FOR YU-BIBOT

A method of controlling dynamic balance for the planar robot platform YU-Bibot as well as its simulation and experimental results are presented in this study. In this method force sensitive resistors (FSR) were placed under the feet, linearized in the circuit on top of the feet, and connected to the controller area network (CAN) of YU-Bibot. Linearized sensor data is used to measure the reaction forces under the feet and the location of the zero moment point. Difference between the measured ZMP location and the reference ZMP location is compensated by adjusting robot’s joints. In order to replicate real robot behavior, the same method was simulated using SimMechanics toolbox of MATLAB®. Three test procedures were conducted in both the real and virtual robot by sending them ZMP trajectories as references and their responses were inspected. In the first and the second procedures a square wave and a triangular wave were used as reference and in the third procedure a staircase wave was used as reference. It was observed that both YU-Bibot and simulated virtual robot were able to track a square wave reference ZMP trajectory.

ÖZET

YU-BİBOT İÇİN DENGE DENETİM STRATEJİSİ

Bu yüksek lisans çalışmasında iki ayaklı düzlemsel YU-Bibot robotunun statik ve dinamik dengesinin denetimi için geliştirilmiş bir yöntem anlatılmaktadır. Bu yöntemde robotun ayaklarının altına yerleştirilmiş kuvvete duyarlı direnç sensörleri ayağın üzerinde bulunan devrede doğrusallaştırılarak robotun denetleyici alan ağına bağlanmıştır. Doğrusallaştırılan sensör bilgileri, ayaklardaki tepki kuvvetlerinin ve yerel sıfır moment noktalarının (SMN) konumlarının hesaplanmasında kullanılmıştır. Yerele SMN değerleri Kalman filtersinden geçip, filtrelenmiş SMN değerleri sayesinde robotun esas SMN konumu hesaplanmıştır. Hesaplanan SMN konumu ile referans SMN konumu arasındaki fark hata olarak kabul edilip bu hata sırasıyla PID denetleyicisinde ve birinci derece bir filtreden geçirilir. Filtre çıkışı kalça konumu için referans kabul edilerek bu konum için gereken eklem pozisyonları hesaplanır. Robotun eklemlerinin yeniden konumlandırılması ile SMN hatası giderilmektedir. Yapılan testlerde bir kare dalga referans SMN yörüngesini takip edebildiği gözlemlenmiştir.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS/ABBREVIATIONS

A	State matrix
C	Measurement matrix
d_y	Distance between each feet
d_{zmp}	Location of absolute ZMP
d_{zmp_L}	Location of left ZMP
d_{zmp_R}	Location of right ZMP
e_{zmp}	ZMP error
F	Total force
F_{A0}	Force at sensor A0
F_{A1}	Force at sensor A1
F_{B0}	Force at sensor B0
F_{B1}	Force at sensor B1
K	First order filter gain
q_i	Position of node I
$\dot{q}_i$	Velocity of node I
$\ddot{q}_i$	Acceleration of node I
R	Noise covariance matrix
r_{zmp}	ZMP reference
T	Time step
Q_k	Process noise covariance
V_k	Measurement noise
W_k	Process noise
X_k	Current State Vector
X_{k+1}	Next State Vector
X_{k_Left}	Current State Vector of the Left Leg
X_{k_Right}	Current State Vector of the Right Leg
Δx	Horizontal hip displacement
τ	First order filter time constant

CAN	Controller area network
CoM	Center of mass
CoP	Center of pressure
PID	Proportional–Integral–Derivative controller
ZMP	Zero moment point

1. INTRODUCTION

A biped robot is a type of robot which employs bipedal locomotion to have mobility. In this method of locomotion robot moves itself by using its two legs. This method of locomotion has a mobility advantage over wheeled or tracked robot locomotion methods. It can transverse rough terrains and climb steps or move over high obstacles where before mentioned methods fail to do so. However bipedal walking method also brings some challenges of its own about maintaining the stability of the robot. This problem can be treated under two main headings, namely static stability and dynamic stability. Static stability can be maintained by controlling the location of center of gravity (CoG) of the robot, but this is not sufficient by itself for maintaining the dynamic stability. Some researchers such as Stephens [1] use center of pressure (CoP) and tries to keep its location under the foot, away from the edges of the foot. Data from the joints are sent to a linear quadratic regulator and a center of mass (CoM) controller. These controllers give torque references which are then used to calculate the desired CoP. This desired CoP is then combined with the posture reference to keep robot's balance while maintaining a standing upright posture. Winter et al. [2] also work with CoM and CoP. In their work they use human subjects to stand in different postures while measuring their anterior/posterior and medial/lateral CoP. They suggest that CoM and CoP are virtually in phase. Another approach is to use zero moment point (ZMP). During the bipedal locomotion, there are forces and moments acting on the foot of a biped robot. These forces and moments can be balanced with an appropriate single ground reaction force and location of this force is called zero moment point (ZMP) [3]. However location of the ZMP changes during the bipedal locomotion and it might move to a position outside the foot. Since the reaction force can only exist in the region between the foot and the ground, this reaction force will not be able to balance the loads acting on the foot and robot will start to roll around the edge of its foot. Therefore, to achieve and maintain the dynamic stability, location of ZMP has to remain under the robot’s foot. Hence the gait trajectory for bipedal locomotion must be created by taking the location of ZMP into consideration.

ZMP can be controlled either directly or by indirectly; depending on how well the dynamics of the robot is known [4]. Some researchers depend on their model and control ZMP indirectly like by controlling another aspect and relating its effect to ZMP. Sugihara et al. [5] used inverted pendulum model for simple calculation of the ZMP location. Sugihara in his research developed a method in which ZMP location is indirectly controlled by controlling the CoG location. In this method the robot is given a reference ZMP trajectory and during the gait, CoG location is altered to match the actual ZMP location with the reference. Park et al. [6] used a robot with 7 degrees of fredom (DoF) and created a gait trajectory by implementing fuzzy logic algorithms. In gait creation Runge-Kutta method of 4th order is used to calculate upper body location of the robot. Torques in the joints are measured during the gait to track the robot movement and used as feedback in the control system.

Zhang et al. [7] used D'Alembert principle to relate the reaction forces to joint movements and created a ZMP control method which uses fuzzy logic algorithm to decrease the upper body motions. This method keeps its validity in both single and double support phases by adjusting the boundary conditions. Lingyun et al. [8] used Takagi-Suzeno type fuzzy logic to find gain parameters which are used to adjust hip movements. This hip trajectory is then related to the ZMP trajectory. Although this method has its advantages, the gain parameters are found by learning and therefore its quality is dependent on the initial candidate sets. Er et al. [9] used Dynamic Fuzzy Q-Learning (DFQL) controller to make their robot follow a reference ZMP trajectory. The algorithm generates criteria for robot to ensure and create fuzzy rules and parameters to try to match those criteria during its operation. Tests showed that the method is able to create structure and parameters of the robot behavior by itself. Ding et al. use [10] ZMP trajectory reference and transforms this into CoM trajectory reference. Afterwards different approaches for single support and double support phase are employed. In double support phase linear inverted pendulum model is used to create governing equation. Then an offline optimization algorithm is used to compute the CoM boundaries. In single support phase a quadratic programming method is used to create CoM trajectory.

Other researchers control ZMP by employing sensors to calculate the ZMP location, based on sensor measurements. One of the sensors that can be used is a force sensitive resistor (FSR). FSR sensors are thin multi-layer sensors and when a force is applied to them their resistance decreases non-linearly. However FSR sensors suffer from resistance drift and hysteresis [11] and therefore care must be taken before selecting an FSR sensor. Erbatur et al [12] used FSR to measure the reaction forces under the robot's foot and used these forces to calculate the ZMP location. The sensor setup was used in Mari-1, Mari-2, and human subjects with good results. Kim et al. used [13] FSRs to measure ZMP and neuro-fuzzy algorithms to model ZMP trajectory. The ZMP trajectory generated by their model is sent to their robot and actual ZMP is measured. Measured and generated ZMP trajectories are matched closely hence hinting that their model can also be used to control the trajectory as well as modeling it. Choil et al. [14] used FSRs to measure reaction forces under their robot's feet and used these force readings to calculate the ZMP location. This calculated ZMP value was used in a fuzzy algorithm to control their robot's posture. The method was tested and proved under single support phase on a slope, by disturbing the robot while it tried to remain in a fixed ZMP value. Sheng et al. [15] proposed a control method for bipedal walking on uneven slopes. The method employs different controllers to achieve this task. A landing time controller is employed to prevent large impact forces, a landing direction controller is employed to make a full foot to ground contact, a ZMP controller is employed to maintain stability and attitude controller is employed to maintain robot in upright orientation. This method enables robot to walk on uneven slopes according to the tests on SCUT-I robot. Lai et al. [16] use an inverted pendulum to model their robot. They use ZMP as reference and distribute this ZMP reference to reference force at feet and reference torque at the joints. These references are then sent to the admittance controller which adjusts the inverse kinematic constrains. The ZMP measured by the sensors are used in linear inverted pendulum model equation to calculate CoG which is used to map the CoG error. The CoG is then compensated by the stabilizer to maintain stability and to have smooth response.

As mentioned before using an FSR sensor to calculate ZMP has its difficulties because of the resistance drift, hysteresis, noise, etc. If these effects are not eliminated they can cause instability in the system. One of the methods for eliminating sensor errors is Kalman filter. Li et al. [17] use Kalman filter and fuzzy motion controller to have a dynamic balance control. In their method force sensors and an accelerometer is employed in a humanoid robot and their measurements are sent to a Kalman filter. Kalman filter estimates the current inclination angle and filters the sensor data which is used to calculate current ZMP. Then estimated current ZMP is compared with designed ZMP trajectory while estimated current inclination angle is compared with the designed inclination angle to find the errors. These errors are then sent to the fuzzy motion controller. FMC output and inverse kinematic outputs are then sent to the motion control card which then controls the actuators.

Some researchers employ simulation tools in order to verify their model or test some of the model's parameters. Models are usually simplified for various reasons, using various assumptions. Therefore it is useful to validate these methods. Rodriguez et al. [18] proposed an underactuated walking model and created a virtual model to compare biomechanical data with their method. They used SimMechanics to create a virtual human body and simulated a bipedal gait. Although SimMechanics is useful for modeling such systems, original program lack collision detection therefore contacts like a foot contacting the ground must be modeled by different approaches. Rodriguez et al. used two different approaches of hard and soft contact. Their results were consistent with the literature data albeit some discrepancies due to their approach and difficulties in replicating the conditions in their literature.

Although controlling the balance of a robot by using ZMP has benefits, it also has disadvantages as well. ZMP based approaches require each foot to make complete contact with the ground. It must be noted that this kind of gait is different from a human's gait. While humans walk, their swinging feet's orientation to the ground changes. The first ground contact is made with the heel, then the whole sole contacts the ground, and finally heel is elevated and the contact is finished at the toes. This gait helps with the efficiency and absorption of the contact forces. Some researchers forego ZMP based control methods and achieve the balance through other means. Song et al. [19] proposed an active disturbance rejection controller for balance. In their research they approach the system as an angular momentum system. The system was divided into four subsystems and each subsystem separately modeled. Then limit cycle method was used to generate an efficient gait. Finally an active disturbance rejection controller was designed to eliminate the instabilities of the limit cycle. In the present study ankle rotation during the gait was handled by the springs at the ankles. The springs allow a margin of passive rotation at the heels to increase efficiency while keeping the ground contact at full most of the time. Also effects of the contact forces was tried to be minimized with slow contact speeds and filters to eliminate disturbance at ZMP measurement.

In the current study a balance control method using ZMP feedback is proposed. Although this study mainly focuses on double support phase, swing phase tests were performed as well. In this method ZMP feedback is calculated by using FSR sensor measurements. The feasibility of using FSRs for calculation of ZMP location was inspected on YU-Bibot in a previous study [20]. Calculated ZMP positions at each feet are then filtered through a Kalman filter and the filtered values are used to calculate the global ZMP position. This global ZMP position is compared to the reference ZMP signal to find the error. The study consists of three main procedures. The first procedure describes the previous method [21] and it is used in this study to compare the results of the new and old methods. In the first procedure the error signal was filtered with a first order filter to generate a hip displacement reference. This reference was then used to calculate the reference joint positions which were then sent to YU-BiBot. In the second procedure, which is the main focus of this study, a PID controller was introduced before the first order filter. In the first two procedures a square wave and a triangular wave were used as the ZMP reference where YU-BiBot remains in double support phase. Third procedure was conducted after calibrating the control method's parameters with the results gathered from the first two procedures. In the third procedure a stair wave ZMP trajectory was given as the reference ZMP signal. In this procedure YU-BiBot starts in double support phase and moves into the swing phase by taking a step forward. After describing the procedures in detail, results are given and discussed.

2. YU-BIBOT

YU-Bibot is a 6 DoF planar robot with lower extremities designed at anthropomorphic dimensions. As it can be seen in Figure 2.1, the robot consists of two feet two shanks, two thighs, a lower body which is connected to thighs, upper body which holds the battery and the computer, and joints which connect the previously mentioned parts together.

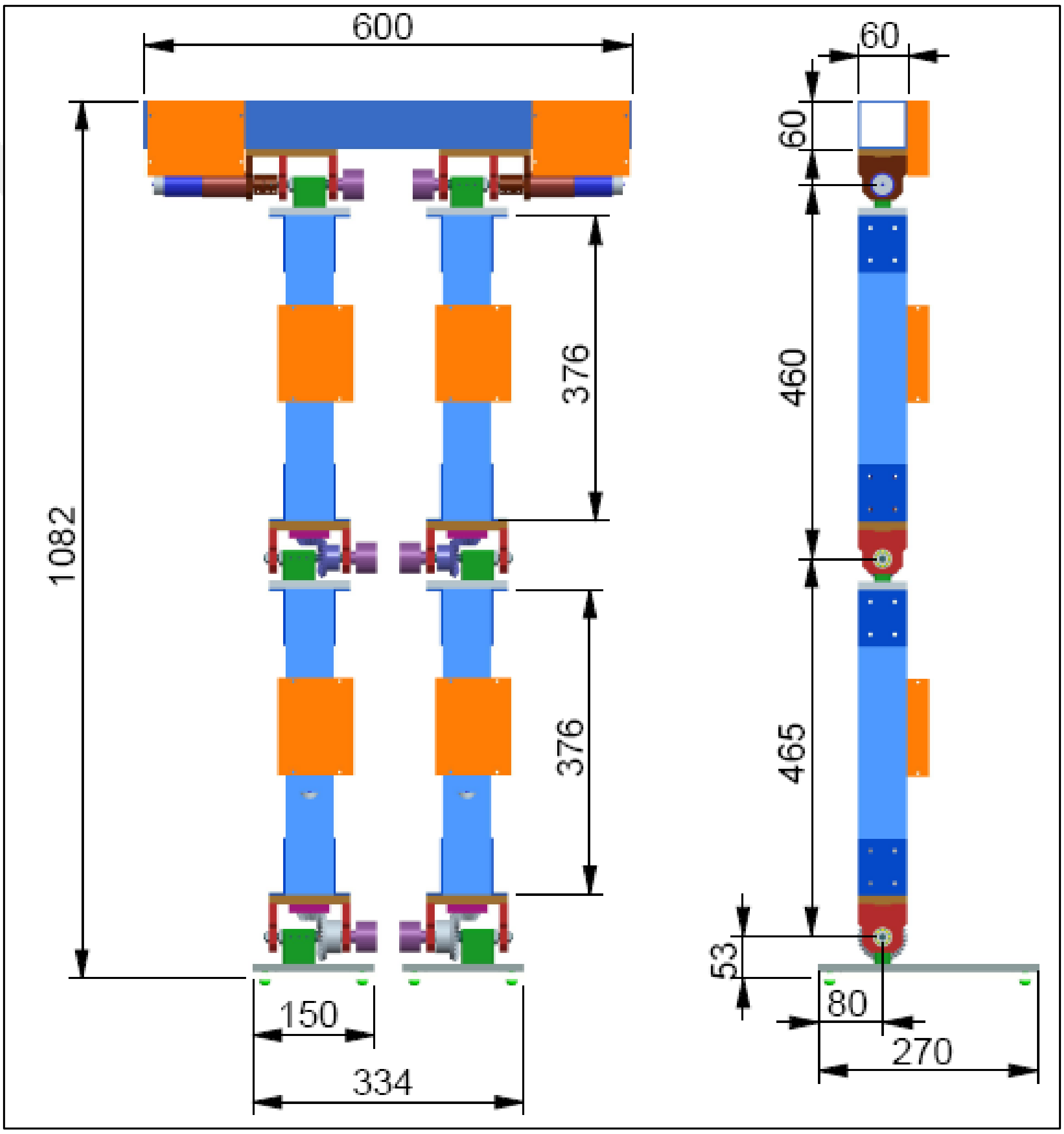

Figure 2.1. YU-Bibot

2.1. ROBOT SPECIFICATIONS

YU-Bibot’s shanks, thighs and lower body are made from extruded aluminum. All joints in YU-Bibot are driven by three-pole brushless Maxon DC motors which are controlled by EPOS2 motion controllers. All joints have incremental encoders to measure angular position. Each encoder sends position data to its respective controller that controls the joint’s motion. The controllers are connected to each other by Controller Area Network (CAN). As can be seen from Figure 2.2, the controllers are connected to each other in 1-3-5-6-4-2 order, as defined by node ID’s of controllers.

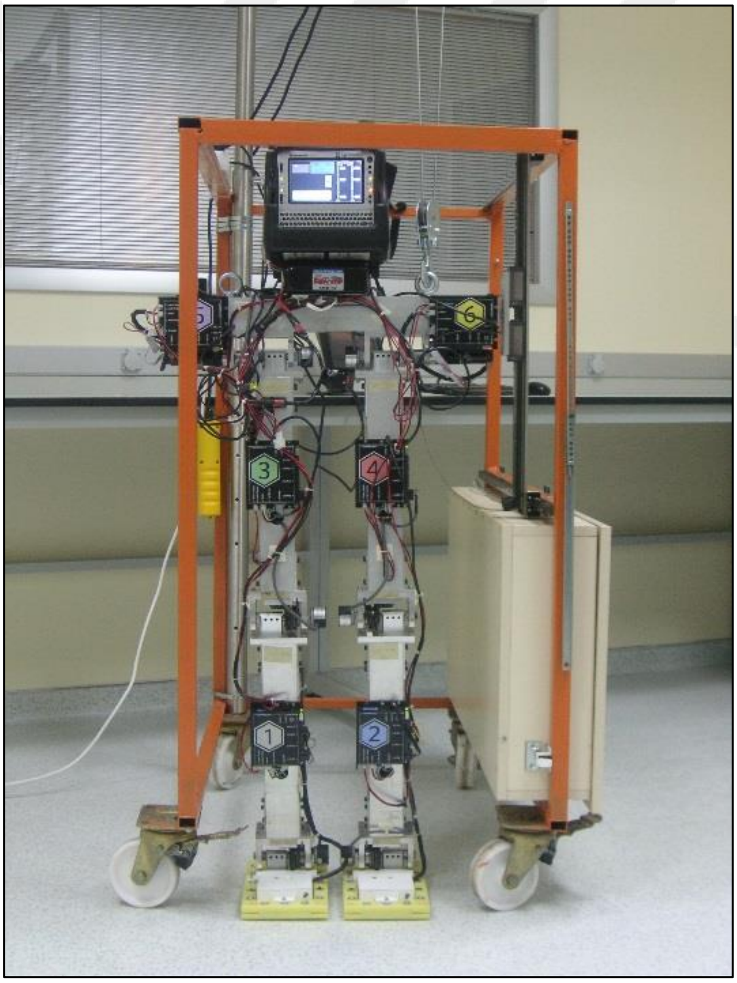

Figure 2.2. Actual YU-Bibot

All messages sent in YU-Bibot CAN have the information to be sent and also the address of the controller to be sent to. In this serial network messages are sent to each controller, the controller which matches the address in the message accepts this message while other controllers ignore it. Also one of the controllers is connected to the computer in the upper body by a USB cable. As it is shown in Figure 2.3, the computer actuates the joints over the controllers by YU-Bibot's CAN bus.

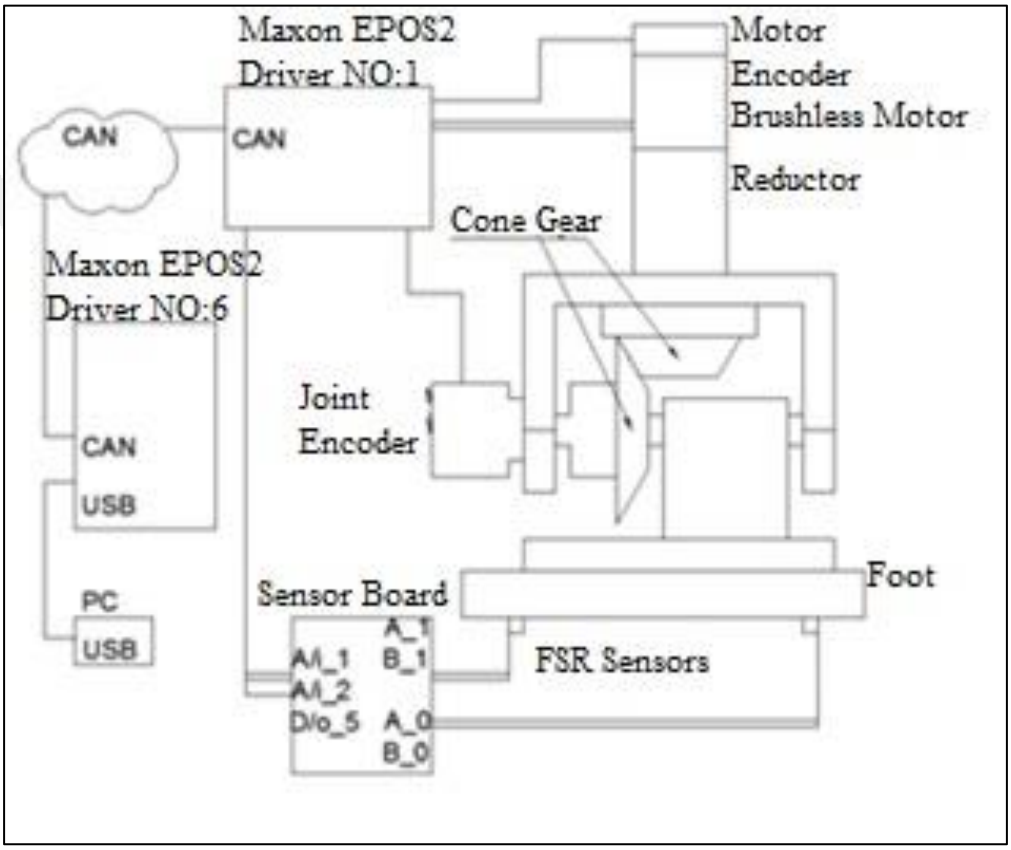

Figure 2.3. Scheme of foot sensors and actuator

The lower body and thighs are connected to each other by a shaft, which is connected to a motor and an encoder at its ends. In other joints, parts are connected to each other by a gearbox and these joints also have torsion springs. With the help of these springs the robot uses less energy while it is standing still.

Feet of YU-Bibot are made of cast polymide and each of them holds 4 FSRs and the circuitry which is connected to the sensors. Other parts are made of aluminum and each of them holds the motor and the controller which drives that motor. There is an accelerometer and gyroscope in the upper body to measure the acceleration and the orientation. The battery is placed securely in upper body and connected a power cable network to supply power to each motor, encoder, and controller. Also in the upper body, the computer is placed in hard plastic case to make it impact resistant.

2.1.1. Sensors

In order to maintain dynamic stability, ZMP location must be monitored and this location must be maintained in a certain range. This is achieved by four FSR sensors in each foot. FSR sensor is a sensor which couples force to electrical resistance; therefore a FSR sensor changes its electrical resistance when a force is applied to it. Each foot of YU-Bibot has four FSR sensors, which are in contact with four plastic pins as seen in Figure 2.4. These pins transmit the normal ground reaction forces to the FSR sensors in the foot.

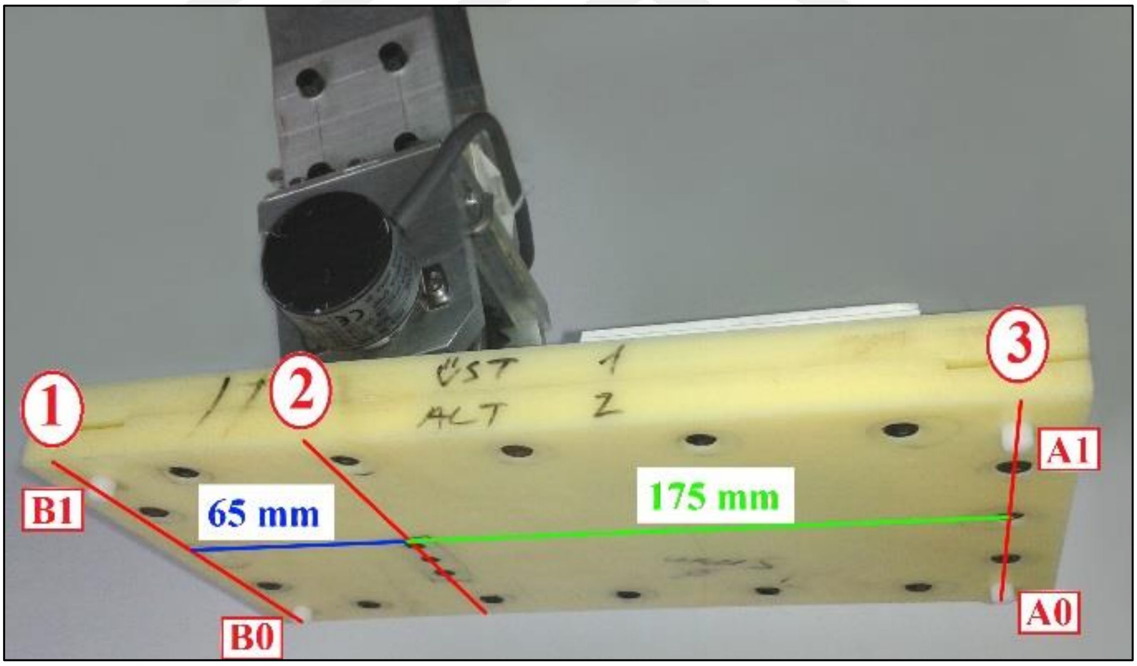

Figure 2.4. Robot foot and sensors

FSR sensors are named as A0, A1, B0, B1, the line 2 is used as reference for ZMP, hence line 1 stays in negative ZMP region where line 3 stays in positive ZMP region. Also each foot houses a circuitry to process the FSR signals.

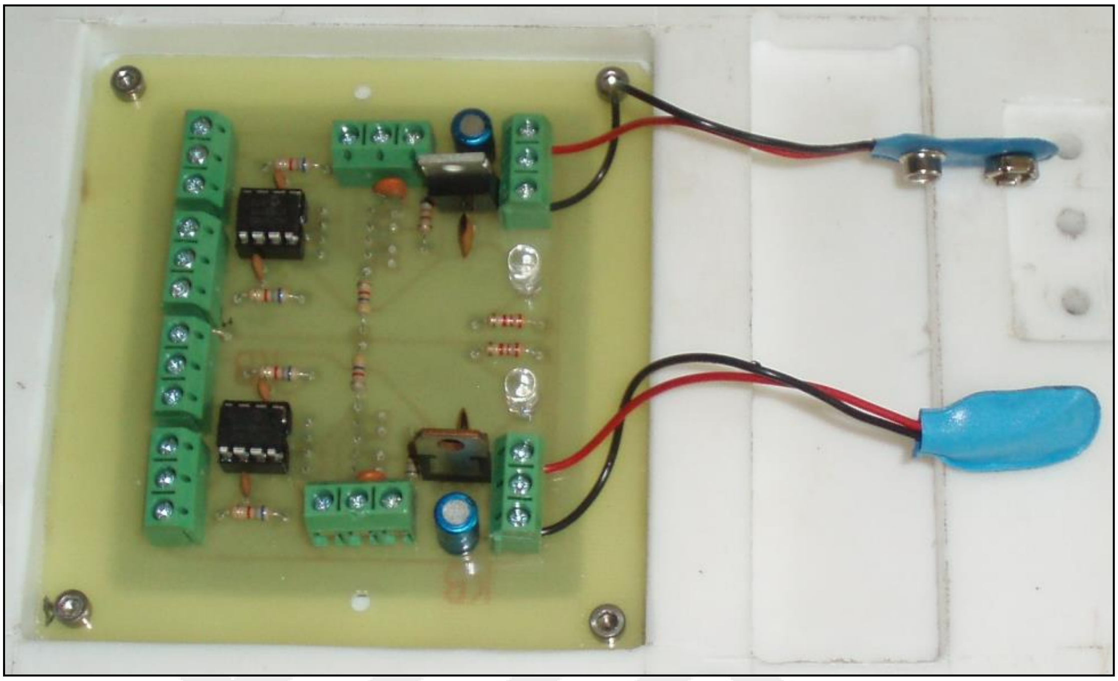

Figure 2.5. FSR circuitry

The circuitry shown in Figure 2.5 is powered with two 9 V batteries and it has two MCP6002 chips in this circuitry. Each MCP6002 is connected to two FSR sensors and amplify their signal. These signals are then sent to the CAN bus by two analog input ports of a controller. However, two analog input ports for each foot are not enough to read 4 sensors. Therefore a switching mechanism is used in the foot sensor circuitry. Switches are operated with a digital output coming out of each foot controller. Since the force measured by an FSR changes during the operation, reading only two sensors might cause an error when calculating the ZMP. However if a high sampling rate is used, then the force measured in switched on case and switched off case will be the same. Therefore in YU-Bibot, using high sampling rate is mandatory for better ZMP position calculation.

2.1.2. CAN Network

Controller Area Network is a type of a message passing multi master communication protocol developed by Uwe Kiencke, Siegfried Dais, Martin Litschel for Bosch in 1986 [22]. CAN bus protocol requires two essential wires to transmit the data; CANH and

CANL. These two wires can have a resistance value of 120 Ω and connect two nodes together or they can be implemented in high speed CAN bus and connect more than two nodes in a linear fashion as seen below in Figure 2.6. In this type of CAN bus connection two nodes are selected as starting and ending nodes where 120 Ω resistance is used at both nodes to terminate the bus.

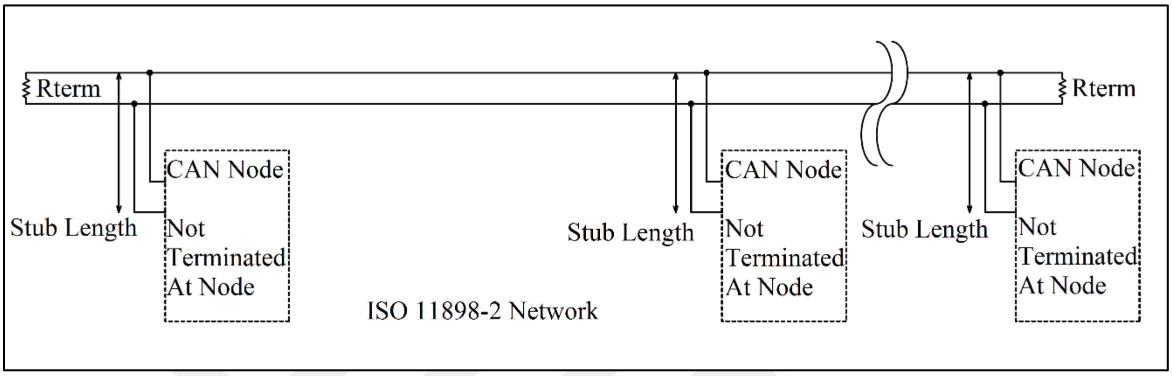

Figure 2.6. CAN network [23]

CAN bus employs two distinct states by using differentiation method between CANH and CANL wires. In recessive state both CANH and CANL are at 2.5 V, therefore making their voltage difference 0 V. This state corresponds as the logical value of 1. In dominant state CANH is driven to 3.5 V while CANL is lowered to 1.5 V, therefore making their voltage difference 2 V. This state corresponds as the logical value of 0.

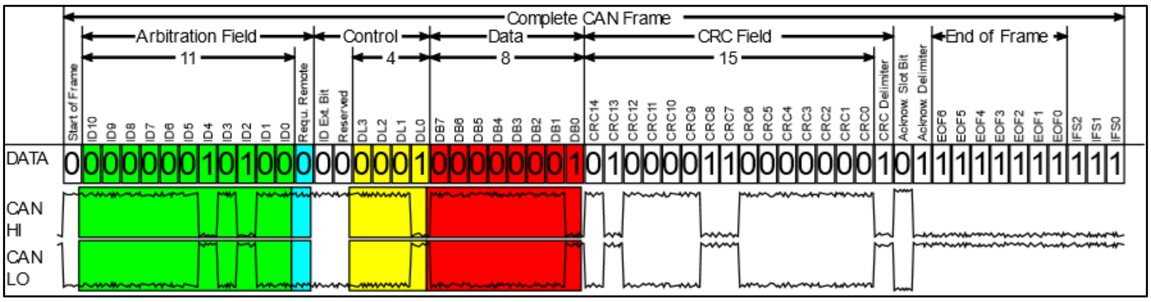

Figure 2.7. CAN frame [23]

When a data is transmitted by CAN bus in a message, a single bit called start of frame (SOF) is sent to the other nodes in the system. This bit helps synchronization of other nodes with the sender note and used as a marker where the message starts as seen in Figure

2.7. After the SOF, message includes Arbitration Field also known as 11-bit identifier in base frame format or 29-bit identifier in extended frame format. In this field, a numerical value is sent to both identify the message and its importance. When multiple messages start their transmission at the same time, arbitration field determines which message has the higher priority. In this process, each message’s most significant bit is compared with each other and if any message has a recessive state, then those messages lose the comparison and their transmission is stopped. Then this comparison is conducted in the next bits towards the least significant bit until remote transmission request (RTR) bits are compared. During the comparison process, only a single message wins the comparison process and continues with its transmission as seen in Figure 2.8. Because of this process a message with the lowest value will have dominant states in their higher significant bits and therefore will be broadcasted.

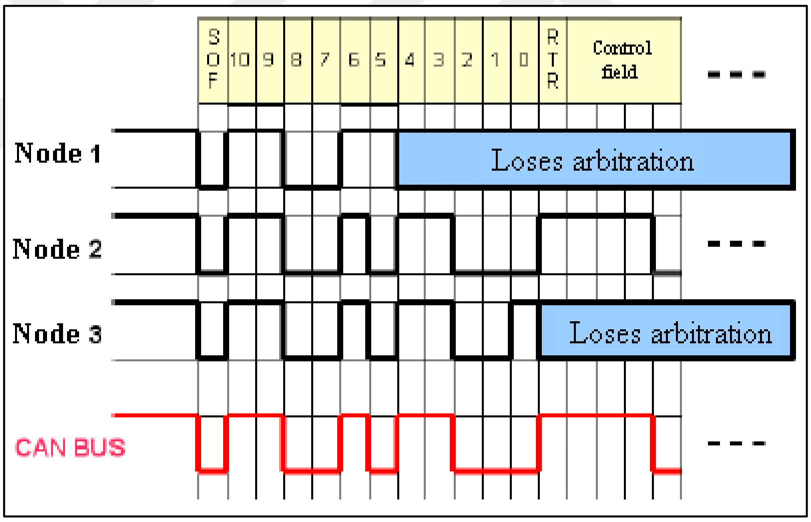

Figure 2.8. Arbitration [24]

RTR bit is used to state if an information is requested from another node, if it is requested, then RTR is in dominant state. RTR bit only indicates data request but does not specify which data is requested or from which node the data is requested; these are specified in the identifier. After this bit identifier extension (IDE) bit follows in dominant state, to specify that the identifier does not have any extension. Later a bit with dominant state is transmitted. This bit is called R0 and does not have any function, but it is reserved for future use. Then a section called data length code (DLC) with a size of 4-bits is transmitted. This section is used to specify the number of bytes in the following data section.

There are some differences between extended frame format and standard frame format. Extended frame format has different sections between 11-bit identifier section and DLC section and order of some sections differs from standard frame format. In extended frame format, a section called substitute remote request follows the 11-bit identifier instead of RTR bit. SRR is used to give higher priority to the messages with standard frame format because SRR bit is always recessive while RTR is always dominant. Therefore, a standard frame message would have a dominant bit against an extended frame message's recessive bit, and because of this it always wins the comparison. Afterwards IDE bit is transmitted. This comes after SRR and its recessive. IDE bit is followed by 18-bit identifier which contains the rest of the identifier. Then RTR section is transmitted which has the same function as the RTR in standard frame format. Two 1-bit sections which are called R1 and R0 follow RTR section. These bits are in dominant state and they are reserved for future use like the R0 in standard frame format. Reserved bits are then followed by DLC section which specifies the length of the following data as described before.

After this point both standard frame and extended frame format follow the same message structure by transmitting data section, cyclic redundancy check (CRC), acknowledge section (ACK), end of frame section (EOF), and interframe space section (IFS). In data section, 0 to 8 bytes of data is transmitted. Since data size can change between previously mentioned range, data size must be the same with the size specified in DLC section. Data section is then followed by CRC section, which includes 15 bits of checksum and CRC delimiter. CRC delimiter is used to denote the end of the CRC section. In the checksum section, total number of previously transmitted bits is multiplied by a generator polynomial and transmitted to the network. Each node in the network, which receives the transmission, calculates their own checksum value and compares this with the checksum value in the received message to inspect if an error has occurred. This section is followed by ACK section. ACK section consists of two bits, which are acknowledgement bit and acknowledgement delimiter. Both ACK bits in the transmitted message are in recessive state. In this section, an arbitration occurs like the process in arbitration field. Nodes which received the message send dominant bits if they did not encounter any errors. If ACK section is not overridden with a dominant state, then transmitting node resends the message.

3. MODELING OF YU-BIBOT

3.1. ZMP PARAMETERS

FSR type sensors in the robot are used to measure local ZMP location at both feet then these local ZMP values are used to calculate the overall ZMP. Using 4 FSRs in each foot, the total forces and moments can be measured simultaneously. ZMP calculation is done as follows:

$$d_{ZMP\_R} = \frac{(F_{A0} + F_{A1})l_{ab} - (F_{B0} + F_{B1})l_{af}}{F_{A0} + F_{A1} + F_{B0} + F_{B1}} \tag{3.1}$$

In the equation A0, A1 are the sensors on the toes and B0, B1 are the sensors on the heels, l_{ab} and l_{af} are distances from reference to heel and toe, respectively, F shows the forces read from each sensor and d_{zmp_R} denotes the measured local ZMP at the right foot. Local ZMP at the left foot is calculated with the same equation. In order to calculate the absolute ZMP location the following equation is derived.

$$\left(d_{ZMP\_R} + d_y\right)F_R + d_{ZMP\_L}F_L = d_{ZMP}F \tag{3.2}$$

This equation is manipulated to leave the ZMP value alone and we get the following equation.

$$d_{ZMP} = \frac{d_{ZMP\_L}F - d_{ZMP\_L}F_R + d_{ZMP\_R}F_R + d_yF_R}{F} \tag{3.3}$$

Afterwards this equation can be simplified into the following equation.

$$d_{ZMP} = d_{ZMP\_L} + \frac{F_R}{F}\left(d_{ZMP\_R} + d_y - d_{ZMP\_L}\right) \tag{3.4}$$

However during the robot's operation, FSR readings include an amount of noise. Although effect of this noise is insignificant by itself, it will be amplified at each step and will result in instability. Therefore a Kalman filter is employed in order to filter ZMP values.

3.2. KALMAN FILTER

Kalman filter is an algorithm which uses previous state data and current measurement data to predict the future state. In order to use the Kalman filter, a state space model has to be created. First two state vectors, one for each leg, are defined by using each joint's position, velocity, and acceleration as follows.

$$x_{k\_Right} = \begin{bmatrix} q_2 \\ \dot{q}_2 \\ \ddot{q}_2 \\ q_4 \\ \dot{q}_4 \\ \ddot{q}_4 \\ q_6 \\ \dot{q}_6 \\ \ddot{q}_6 \end{bmatrix} \; and \; x_{k\_Left} = \begin{bmatrix} q_1 \\ \dot{q}_1 \\ \ddot{q}_1 \\ q_3 \\ \dot{q}_3 \\ \ddot{q}_3 \\ q_5 \\ \dot{q}_5 \\ \ddot{q}_5 \end{bmatrix} \tag{3.5}$$

Where q_i are positions, $\dot{q}_i$ are velocities, and $\ddot{q}_i$ are accelerations. By using these state vectors, the following general state equations are written.

$$\begin{gathered} \text{State equation: } \boldsymbol{X_{k+1} = AX_k + W_k} \\ \text{Measurement equation: } \boldsymbol{Y_k = CX_k + V_k} \end{gathered} \tag{3.6}$$

The state equation describes the robot's next state by using the current state. The current state of $\mathbf{X_k}$ is multiplied by the state matrix $\mathbf{A}$ and the result gives the predicted next state. A process noise $\mathbf{W_k}$ is added to the prediction to describe the difference between the prediction and the real next state. The state matrix $\mathbf{A}$ is written by using T, which is the sampling rate of the robot as shown in equation 3.7.

$$\boldsymbol{A} = \begin{bmatrix} \begin{matrix} 1 & T & {}^{T^2}/_{2} \\ 0 & 1 & T \\ 0 & 0 & 1 \end{matrix} & \cdots & 0 \\ \vdots & \begin{matrix} 1 & T & {}^{T^2}/_{2} \\ 0 & 1 & T \\ 0 & 0 & 1 \end{matrix} & \vdots \\ 0 & \cdots & \begin{matrix} 1 & T & {}^{T^2}/_{2} \\ 0 & 1 & T \\ 0 & 0 & 1 \end{matrix} \end{bmatrix} \tag{3.7}$$

Measurement matrix **C** is used to relate the robot joints' motion to the measured ZMP. Similar to the state equation there is a noise in effect. This noise is called measurement noise and is shown as $\mathbf{V_k}$. The measurement matrix **C** was identified from tests. In these tests, a square wave hip trajectory was given to the robot and ZMP values were measured. A linear model was then fitted to describe the effect of robot joints' motion to the ZMP which gave the results in (3.8).

$$\begin{gathered} C_{Right} = [0.3916 \quad 0.0569 \quad 0.7620 \quad 0.0025 \quad 0.0012 \quad 0 \quad -0.239 \quad 0 \quad 0] \\ C_{Left} = [-0.8943 \quad 0.008 \quad 0 \quad -0.9106 \quad -0.0075 \quad 0 \quad -0.922 \quad -0.0002 \quad 0] \end{gathered} \tag{3.8}$$

During the initialization Kalman filter uses the initial state values as the previous state values to calculate both the predicted new state and process covariance matrix as shown in Figure 3.1.

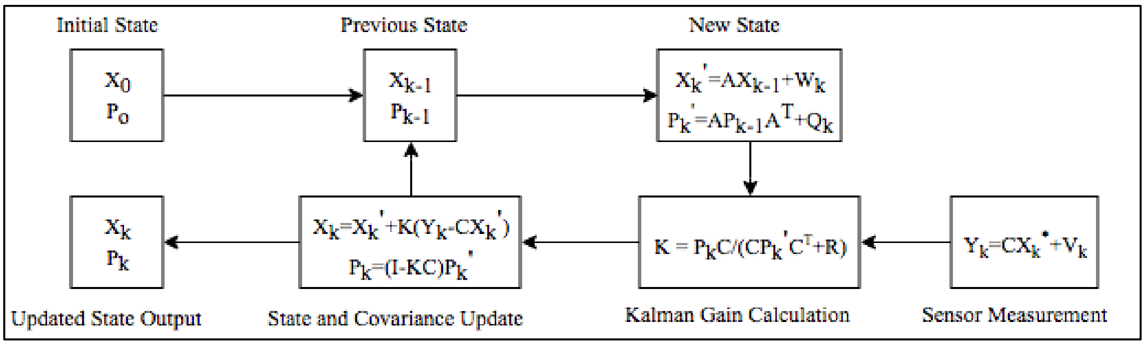

Figure 3.1. Kalman flowchart

Process noise covariance matrix of $\mathbf{Q}_k$ is used in order to calculate the process covariance matrix. Since process noise assumed to be very small, process noise covariance matrix was chosen as 0.05 multiplied by an identity matrix. In the next step sensor measurement is used to calculate the Kalman gain. During the Kalman gain calculation measurement noise covariance matrix of **R** is used. Since measurement noise is an effective element, the measurement noise covariance matrix had to be calculated. In order to calculate this matrix robot was ordered to stand still and the results in Figure 3.2 were gathered.

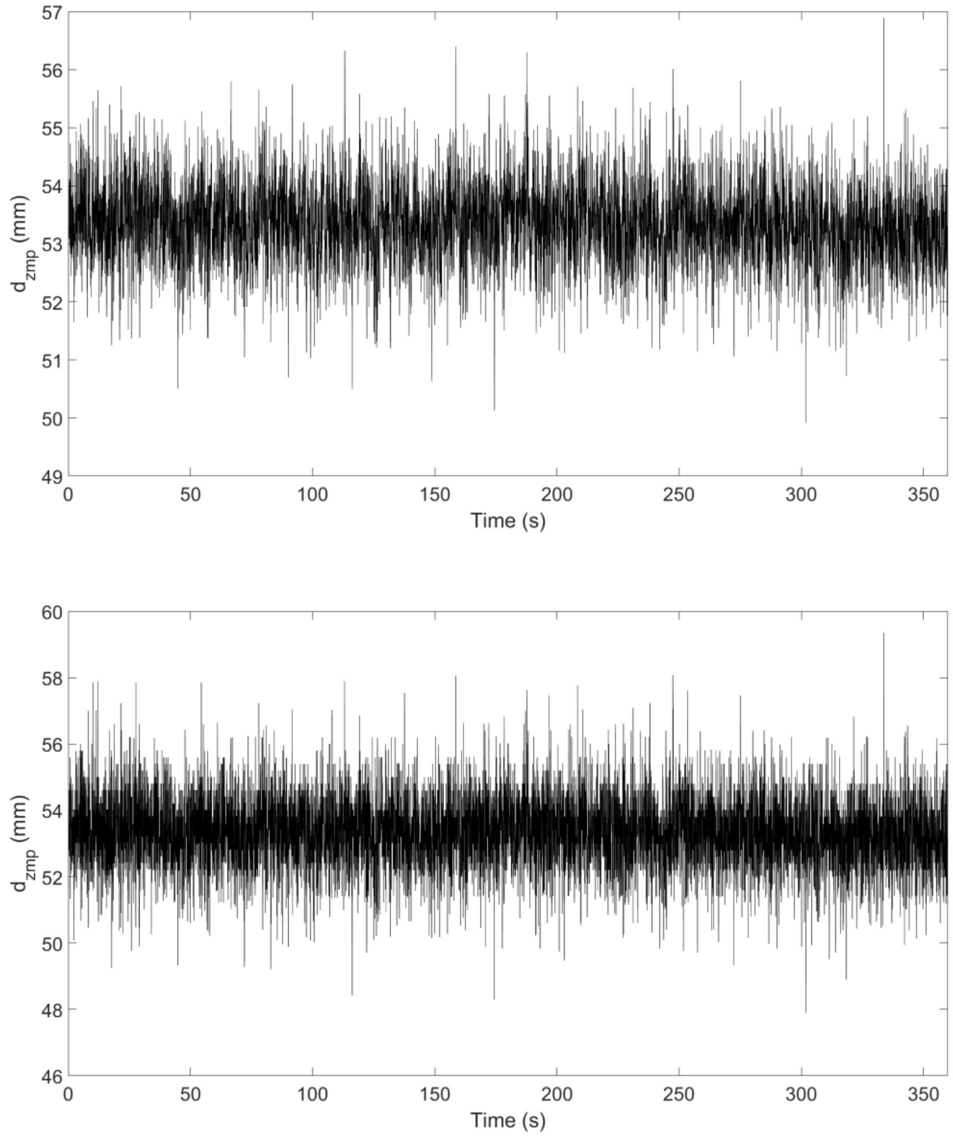

Figure 3.2. ZMP recordings of high (a) and low (b) variances at stand still.

In this state ZMP value had to be a constant value and any divergence from this value was considered as the noise. Therefore average of the measured ZMP data was subtracted from the measured ZMP data and remaining data was assumed as the noise. Then the variance of this noise was calculated. This variance differs due to the outside noise and battery level at the circuit. Therefore these variances are calculated each time during the robot's initialization. Approximately variances range between 1.2355 mm^2 (Figure 3.2(.a)) and 0.4738 mm^2 (3.2(.b)). These variances were then multiplied by an identity matrix in order to determine the measurement noise covariance matrices for each foot.

3.3. CONTROL ALGORITHM

The control algorithm works as a single-input and multiple-output (SIMO) system in a closed loop, where a reference ZMP signal is sent to the system as the input and actual ZMP measured from FSR sensors is treated as the output, as seen in Figure 3.3.

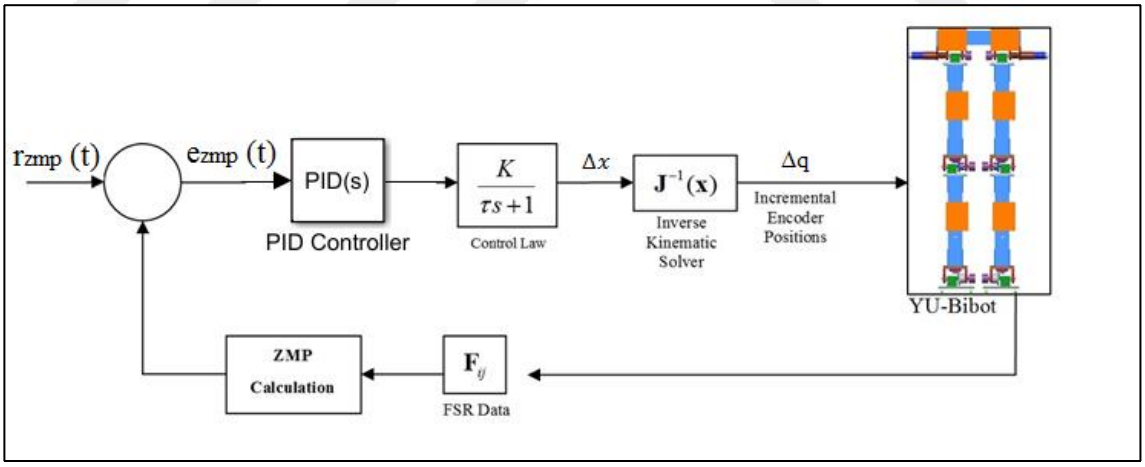

Figure 3.3. Balance control scheme

The control scheme for maintaining stability is shown in figure (3.3). Deviation from the desired ZMP is calculated as shown in equation (3.9).

$$e_{zmp}(s) = r_{zmp}(s) - d_{zmp}(s) \tag{3.9}$$

Deviation from the desired ZMP reference passes through a PID controller then this signal is filtered through a first order proportional controller.

$$\Delta x(s) = \left(P + \frac{I}{s} + D\frac{Ns}{s+N}\right)\frac{K}{\tau s + 1} e_{ZMP}(s) \tag{3.10}$$

In these equations r$_{zmp}$ denotes ZMP reference, e$_{zmp}$ denotes ZMP error. **K** and $\boldsymbol{\tau}$ denotes proportional constant and filter time constant. The control signal, which is the calculated horizontal hip displacement, is denoted as Δx.

Acquired horizontal hip displacement is then sent through a backward kinematic solver to calculate the compensated changes in joints. The gait of the robot is created through an inverse kinematic solver which uses the model seen in Figure 3.4.

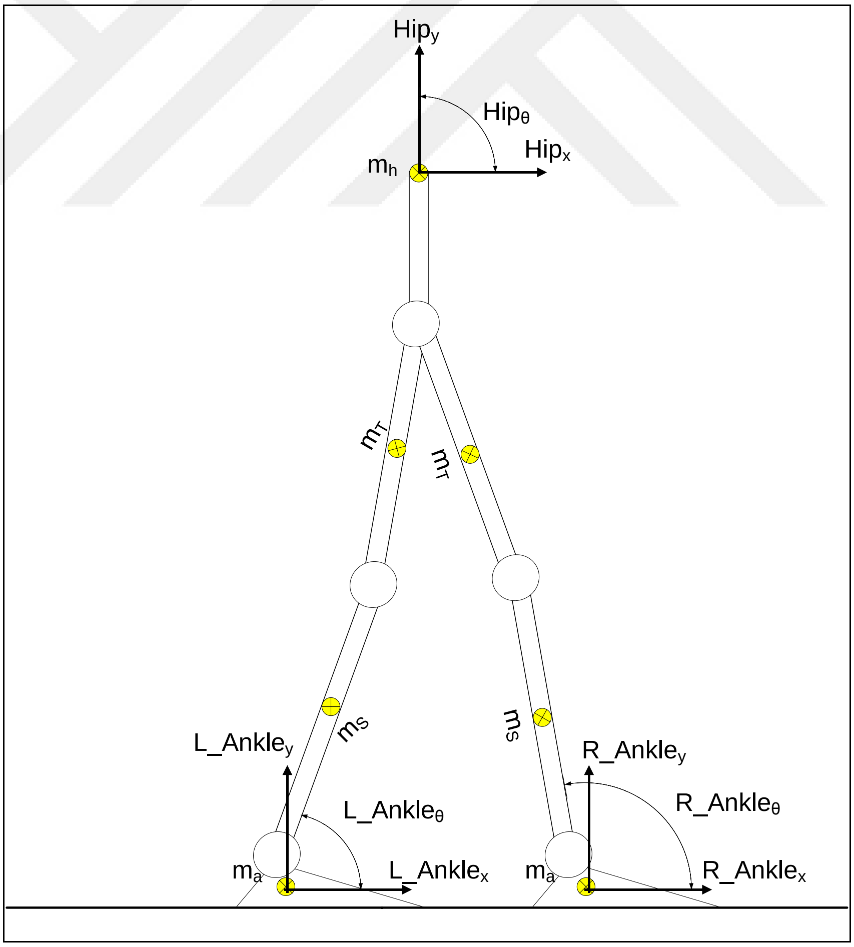

Figure 3.4. Gait parameters

These changes are then sent to the joint controllers. Kinematic solver calculates the changes in the joints while maintaining the vertical hip position and hip orientation.

3.4. SIMULATION

In our approach we first created a block diagram of YU-BiBot in order to evaluate its response to the implemented balance control method and to serve as a test bed for further developments. When creating the block diagram, a modular approach was implemented so that same controller algorithm can be used when controlling the real robot. Simulation block diagram was created in SimMechanics toolbox of MATLAB®. The overall Simulink model is shown in figure 3.5.

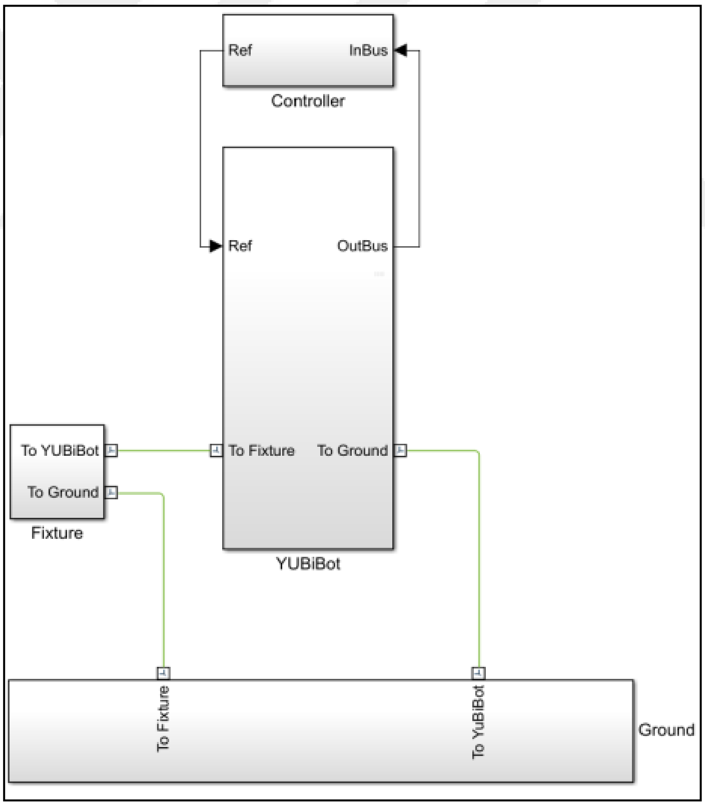

Figure 3.5. Simulink model of YU-Bibot.

Simulation consists of four main subsystems. Ground subsystem includes the planar body which YU-BiBot stands on and some utility blocks which are necessary for the simulation. YU-BiBot is a 2D bipedal robot and all of its motions remain in the sagittal plane. In order to meet this condition, a fixture is used in real life. Fixture subsystem is used to model the real fixture. YU-BiBot subsystem includes the bipedal mechanism. This is achieved by using body blocks for body parts, using rigid transform blocks to define each part's spatial information, and using joint blocks to allow motion of these body parts. Parts of the robot, which were created in SolidWorks®, were imported to the model and connected to each other with revolute joints as shown in figure 3.6.

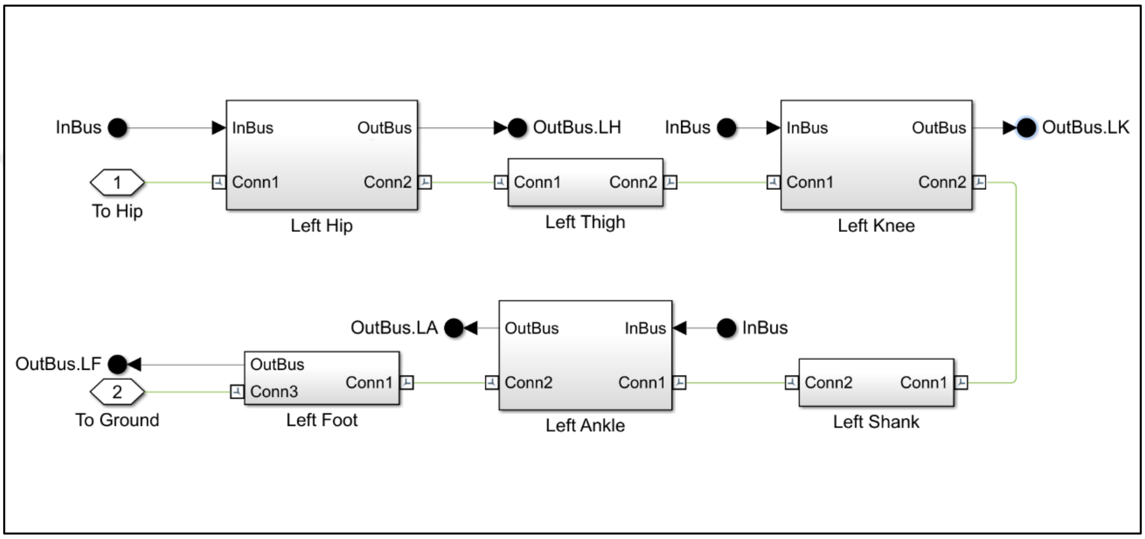

Figure 3.6. Leg subsystem.

The real robot stands on ground on two feet, and each foot is on top of 4 FSRs. In simulation contacts were modeled for these 4 points as shown in figure 3.7 and normal forces on these points were read to replicate FSRs in reality.

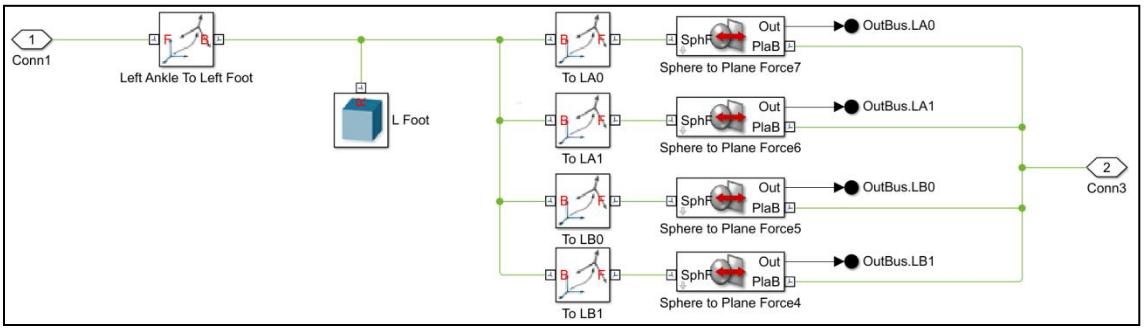

Figure 3.7. FSR models of the left foot.

Position, velocity, and acceleration of each joint and normal force readings at both feet are sent to the controller subsystem. Using the normal force readings, local ZMP positions at each feet are calculated in state calculations subsystem. Then each of the local ZMP values are filtered with a Kalman filter. Both filtered and unfiltered ZMP positions are then sent to absolute ZMP subsystem where the overall ZMP of YU-BiBot is calculated as shown in figure 3.8

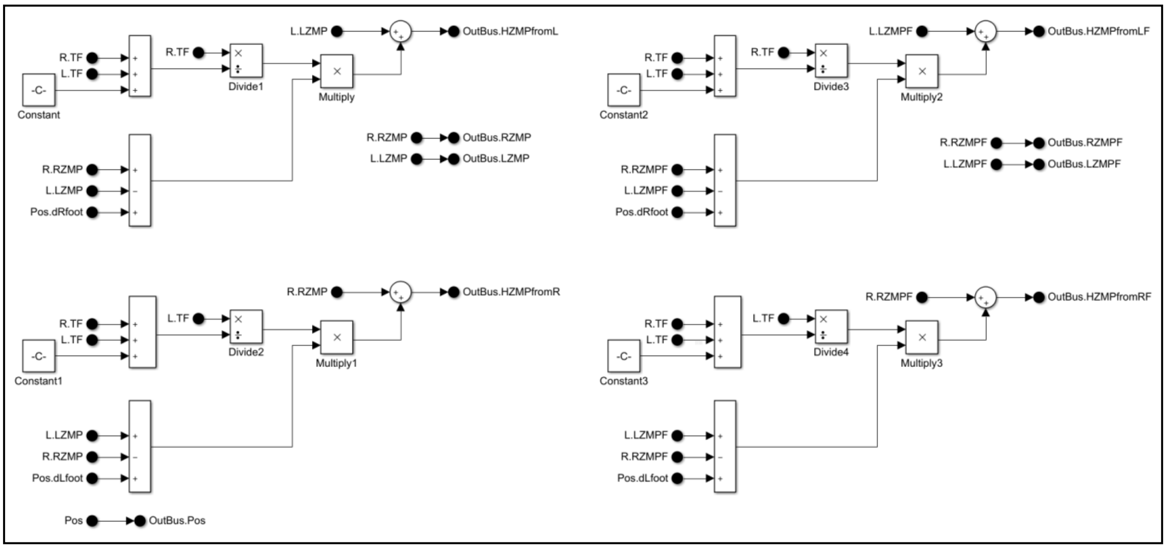

Figure 3.8. Hip ZMP.

Absolute ZMP location with respect to the right and left foot is calculated by using filtered and unfiltered local ZMP values. Although different ZMP values are calculated only the filtered hip ZMP values are necessary, any unfiltered hip ZMP values are used to inspect the effect of the filter. During the robot's swing phase, one foot is lifted, swung forward, and lowered on the ground. Hip ZMP with respect to the foot which is on the ground is used in this phase. This ZMP position is then compared to the reference ZMP to find the error in ZMP in controller method subsystem as shown in figure 3.9.

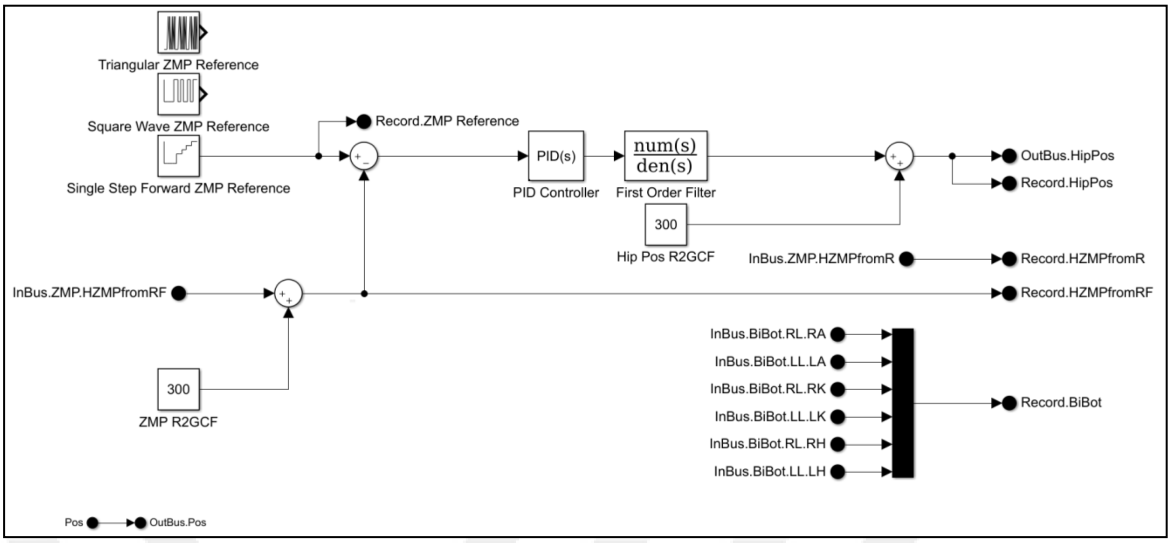

Figure 3.9. Control method.

In the first approach this error was filtered through only a first order proportional controller to calculate the required change of horizontal hip position. In the second approach a PID controller was introduced before the filter. Regardless of the approach, the filtered result is deemed as the required hip position which would eliminate the error. This new position value and a state input are then sent to an inverse kinematic solver to calculate the change in joint positions, which are sent to their respective joints in leg subsystems.

Another block diagram was created to control the real robot. Communication between YU-BiBot's CAN bus and block diagram is achieved with custom scripts. Block diagram use these scripts to gather data from the robot and sends the data to the controller subsystem. This subsystem is the same as its simulation block counterpart. Similarly, calculated hip position with respect to the state is then sent to a function which calculates the joint positions. These positions are then sent to YU-BiBot.

4. EXPERIMENTAL METHOD

Three testing procedures were conducted in this study. In the first two procedures YU-BiBot remains in double support phase and both a square wave and a triangular ZMP trajectories were used as a reference in both simulation and real robot. By using these relatively easier reference signals YU-BiBot's control parameters were calibrated before the swing phase in the third procedure. In the first test procedure, robot stands straight on its legs, which are at the same horizontal position as shown in figure 4.1.

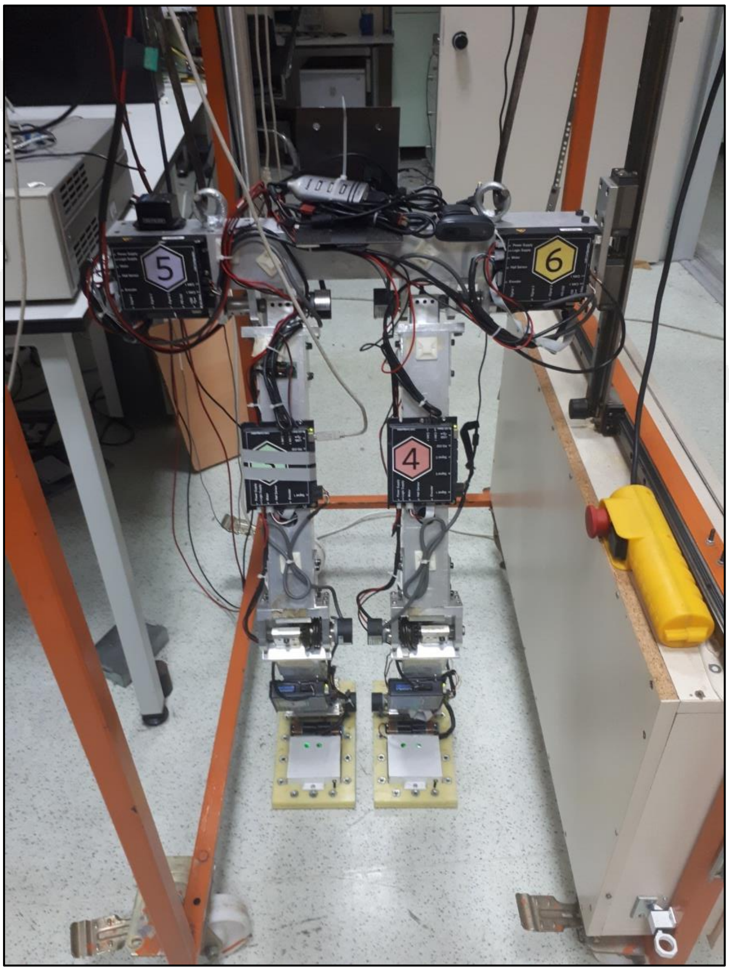

Figure 4.1. First test procedure.

Due to this posture both local and hip ZMP values are the same and effects of the first order filter was inspected. In these conducted experiments and simulations, reference ZMP periodically changes between 40 mm and 70 mm. Both the simulation and real robot change their horizontal hip position, while keeping vertical hip position at 900 mm, to match the actual ZMP position to the given reference ZMP. Posture control is achieved with K and τ constants. During the experiments and simulations different K and τ constants were used, in order to examine their effect.

After these initial tests a PID controller was implemented before the first order filter in order to improve robot's response. In the second test procedure robot stands on the ground with its legs apart, where right foot was placed on 30 cm forward of the left foot as shown in figure 4.2.

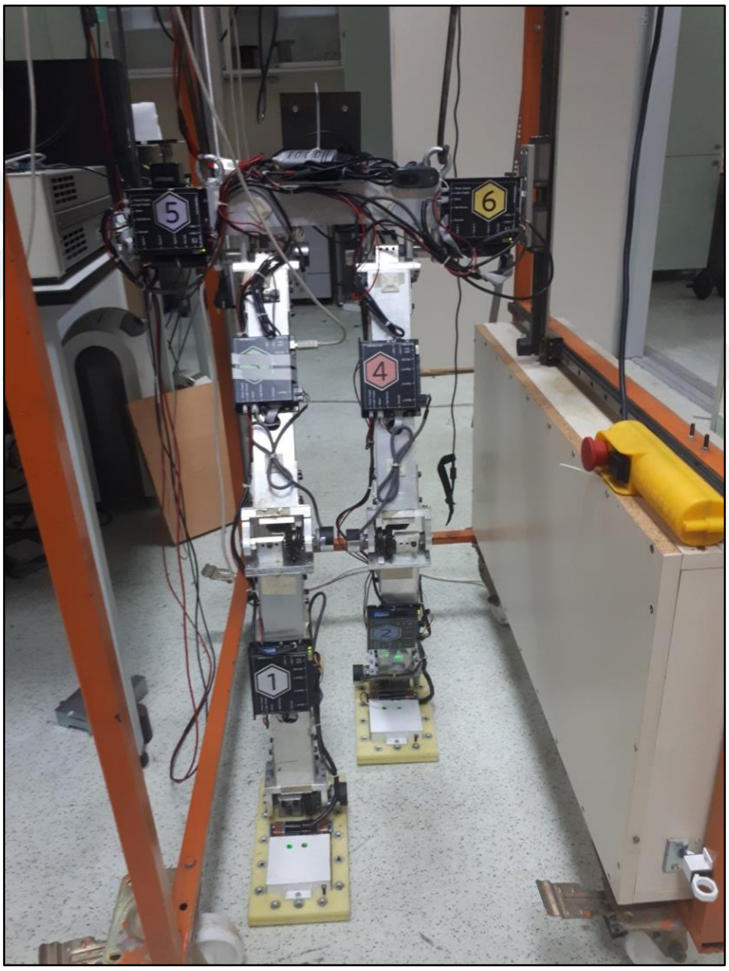

Figure 4.2. Second test procedure.

This posture was selected in order to investigate the double support phase of walking. Similarly a square wave and a triangular were given as the reference ZMP.

5. RESULTS AND DISCUSSION

5.1. PROCEDURE 1 SIMULATION RESULTS

Before the real experiments simulations were conducted. In these simulations virtual robot was provided with both square and triangular ZMP references. In each test different set of filter constants were used in order to see their effects.

5.1.1. Square Wave ZMP Reference

Figures from 5.1 to 5.9 show the results of YU-Bibot's response to square wave ZMP reference signal with different control parameters.

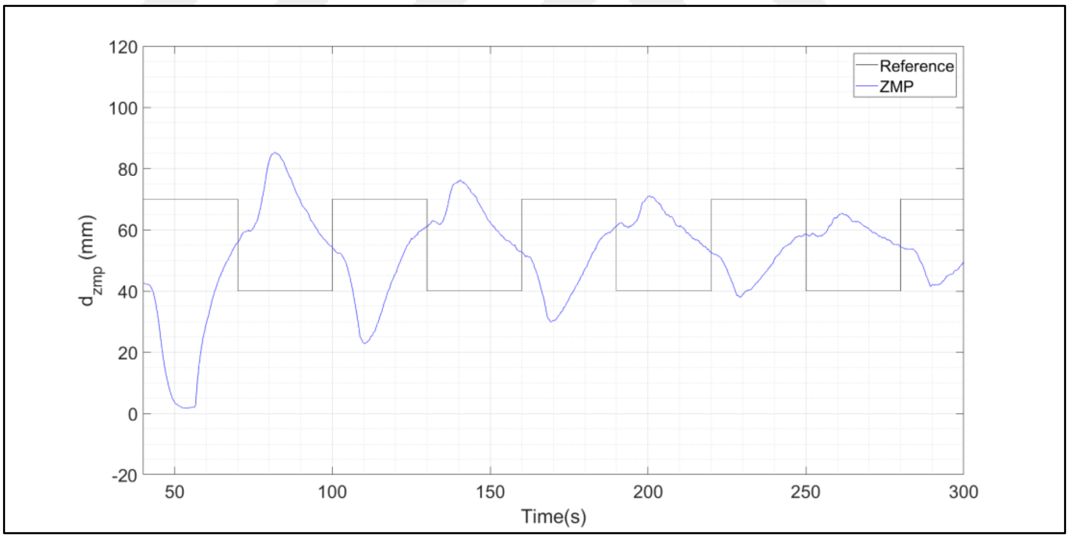

Figure 5.1. Square ZMP reference test of simulation for K=0.2 and τ=0.25

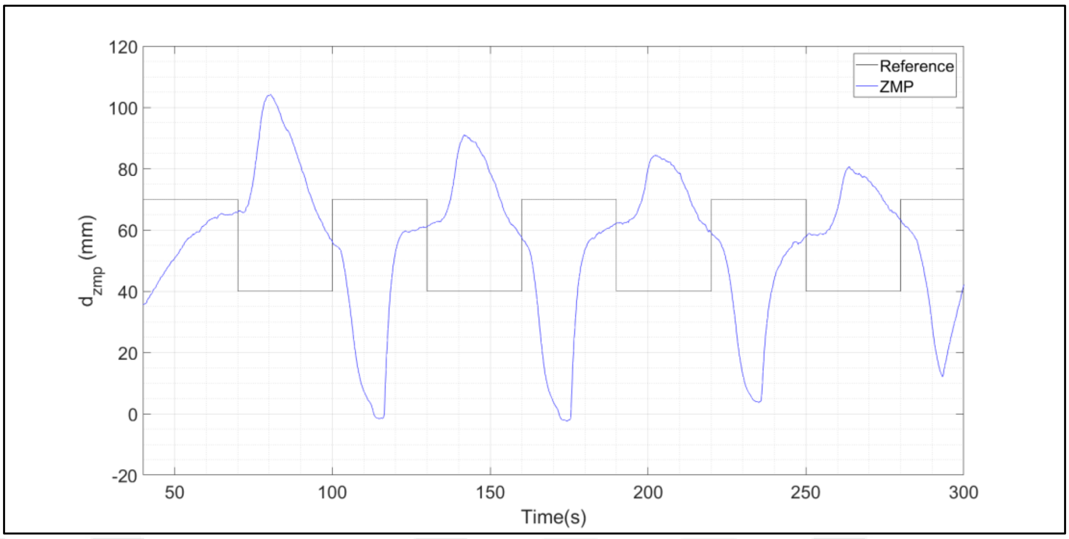

Figure 5.2. Square ZMP reference test of simulation for K=0.2 and τ =0.3

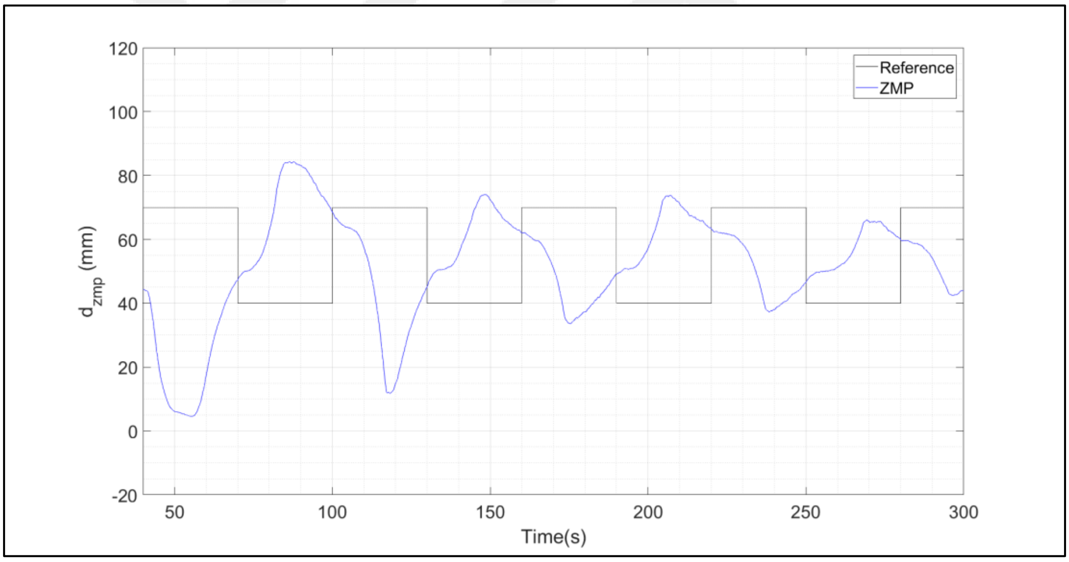

Figure 5.3. Square ZMP reference test of simulation for K=0.2 and τ=0.5

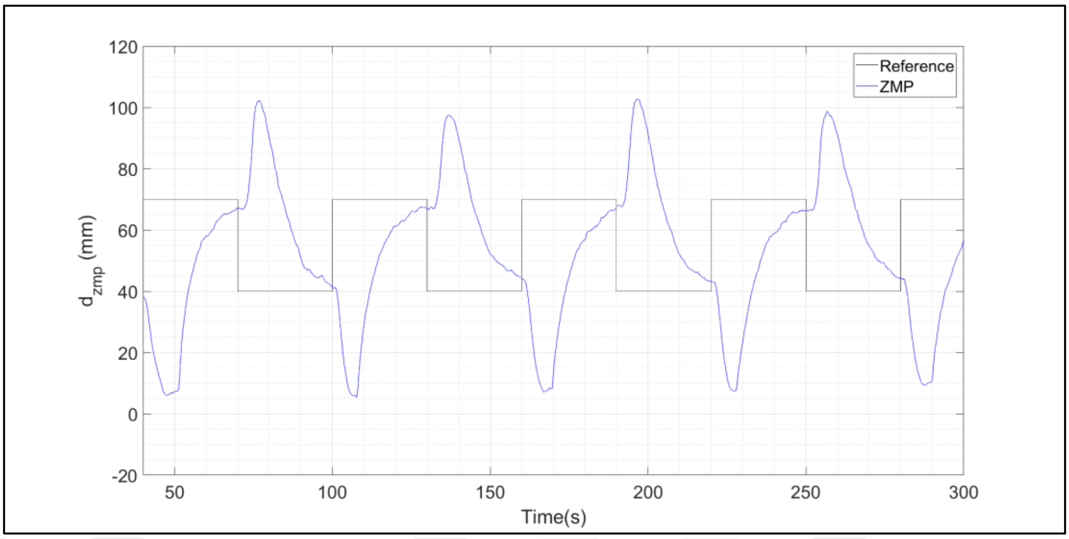

Figure 5.4. Square ZMP reference test of simulation for K=0.3 and τ=0.25

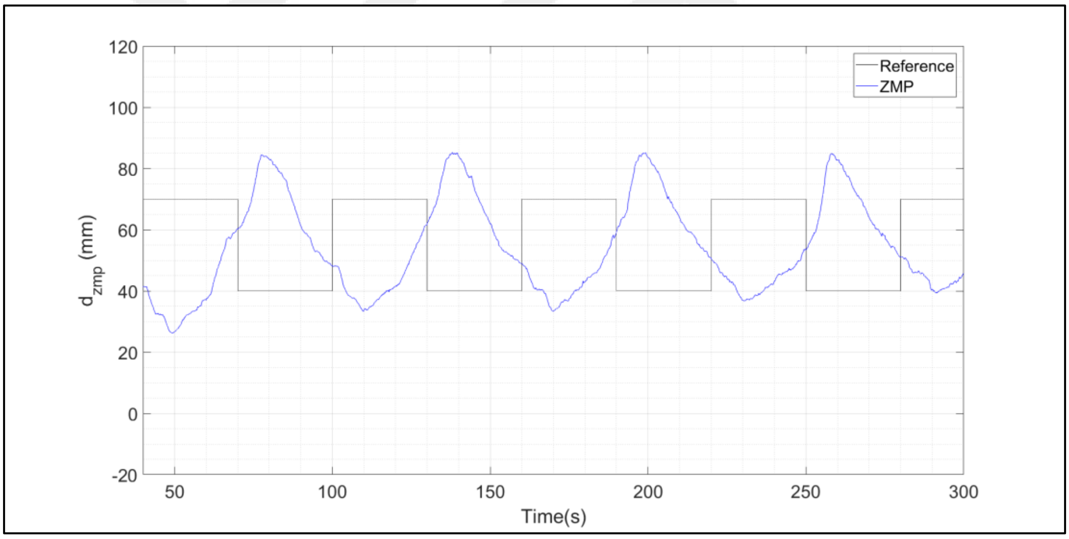

Figure 5.5. Square ZMP reference test of simulation for K=0.3 and τ =0.3

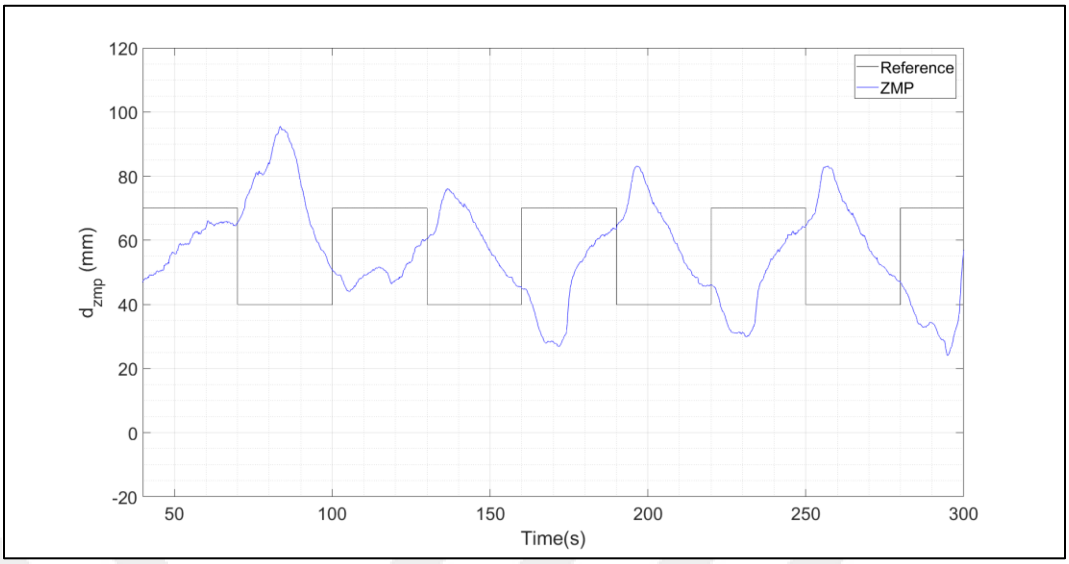

Figure 5.6. Square ZMP reference test of simulation for K=0.3 and τ=0.5

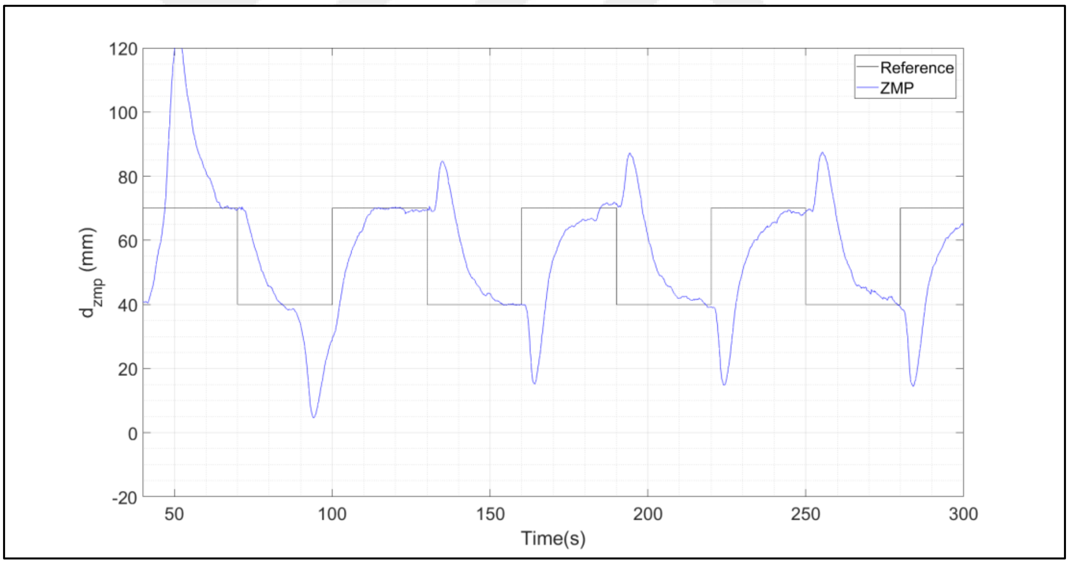

Figure 5.7. Square ZMP reference test of simulation for K=0.46 and τ=0.25

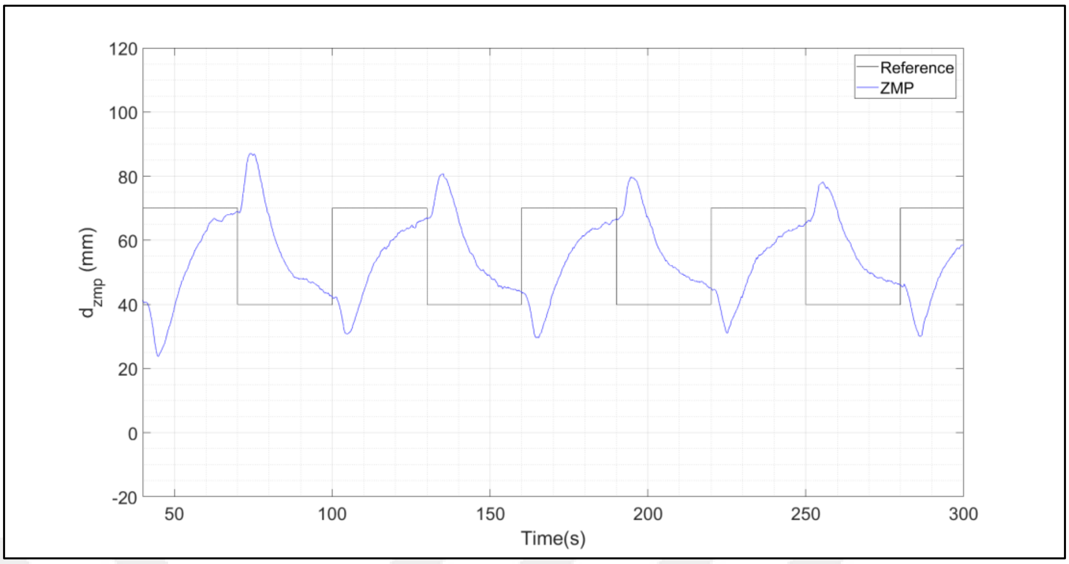

Figure 5.8. Square ZMP reference test of simulation for K=0.46 and τ =0.3

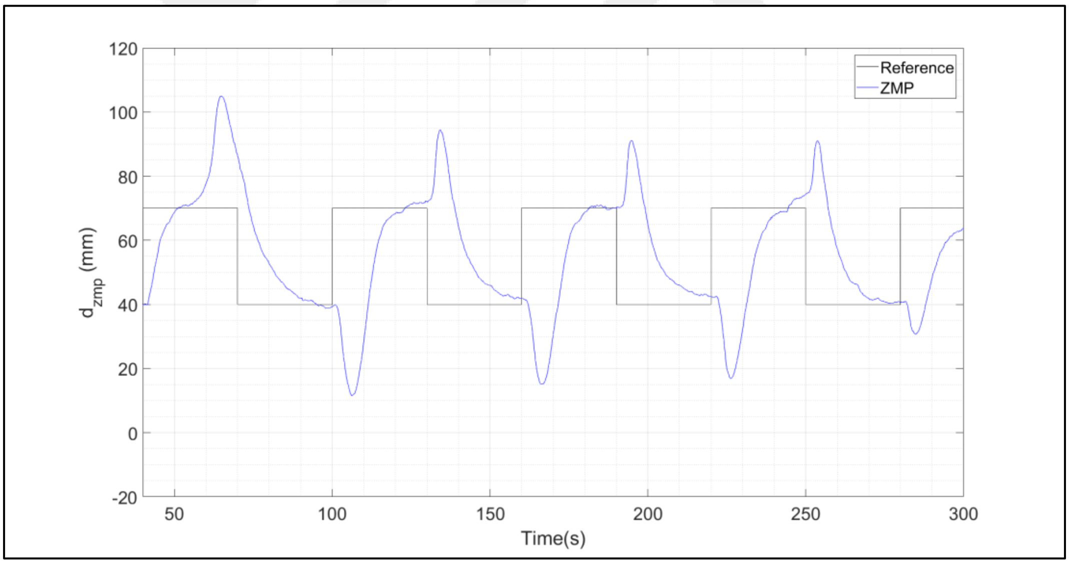

Figure 5.9. Square ZMP reference test of simulation for K=0.46 and τ=0.5

It is seen that robot performs a periodical movement to match the periodical ZMP reference. Also it is seen that higher **K** values with lower **τ** give the better results. However these results are far from satisfactory. System’s response to the reference change is slow and there are undershoots in the system.

5.1.2. Triangular Wave ZMP Reference

Figures from 5.10 to 5.18 show the results of YU-Bibot's response to triangular wave ZMP reference signal with different control parameters.

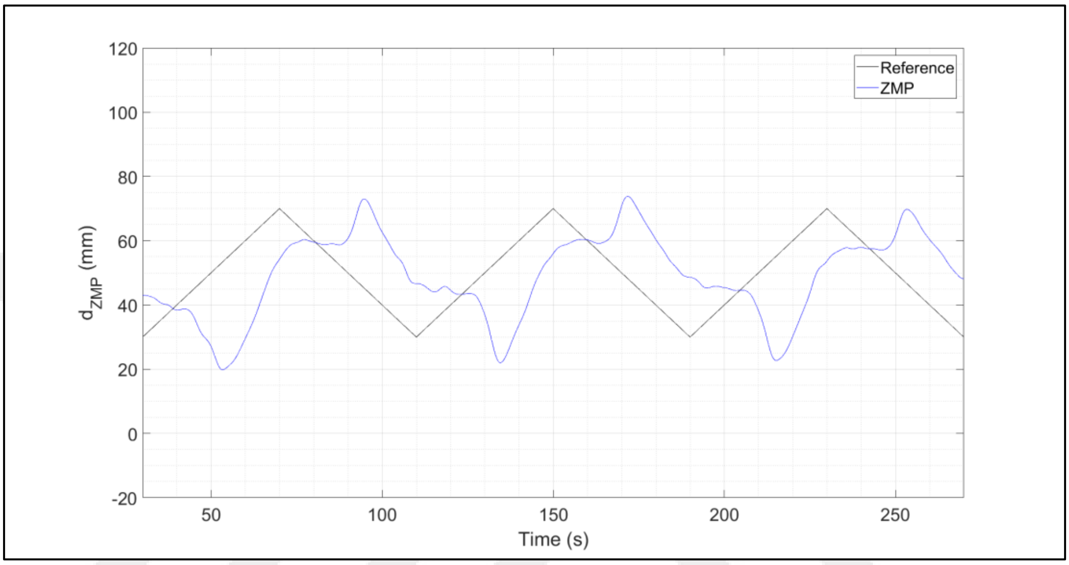

Figure 5.10. Triangular ZMP reference test of simulation for K=0.2 and τ=0.25

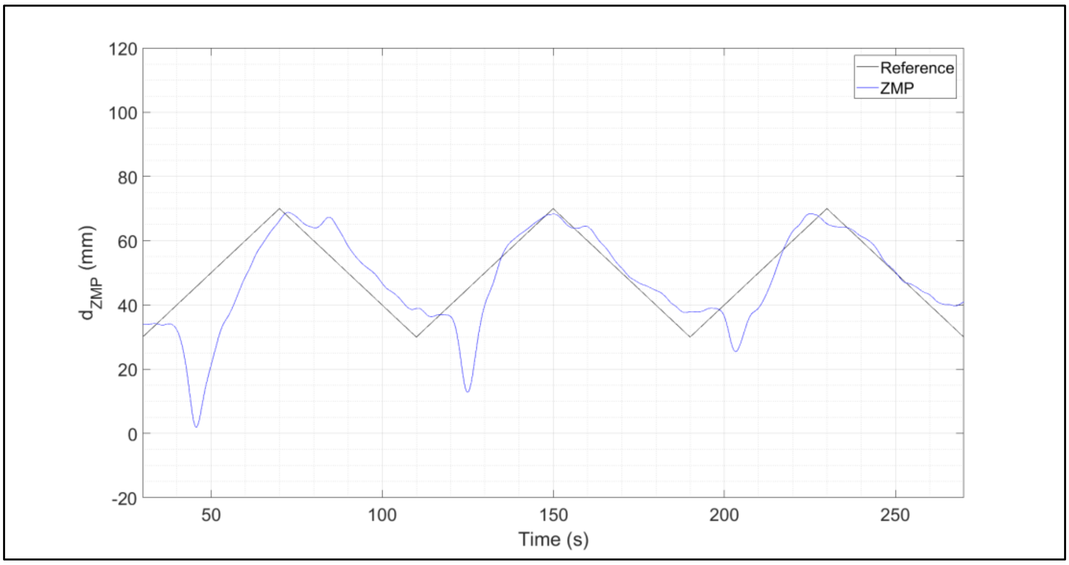

Figure 5.11. Triangular ZMP reference test of simulation for K=0.2 and τ =0.3

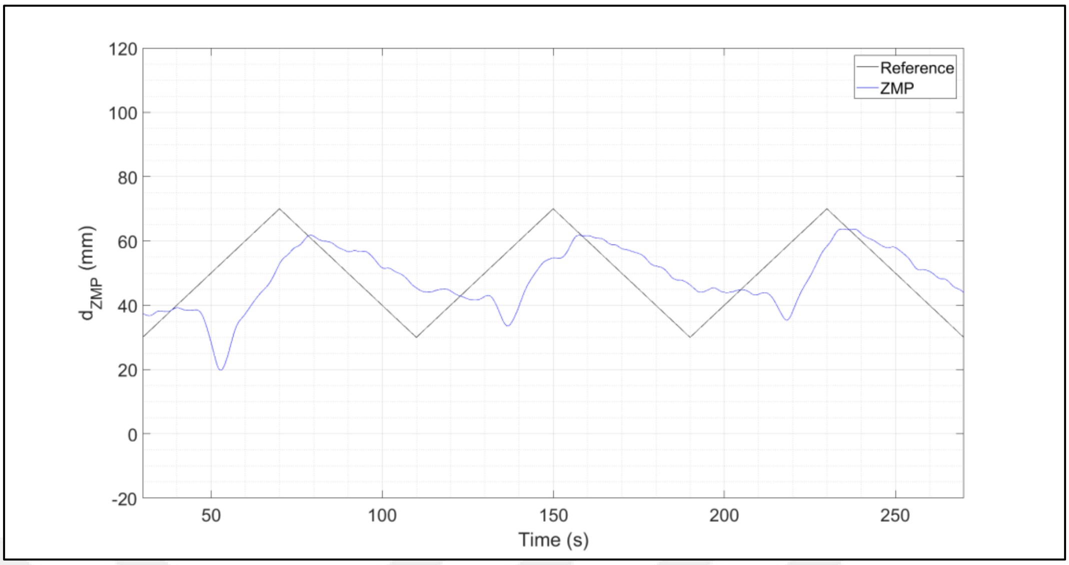

Figure 5.12. Triangular ZMP reference test of simulation for K=0.2 and τ=0.5

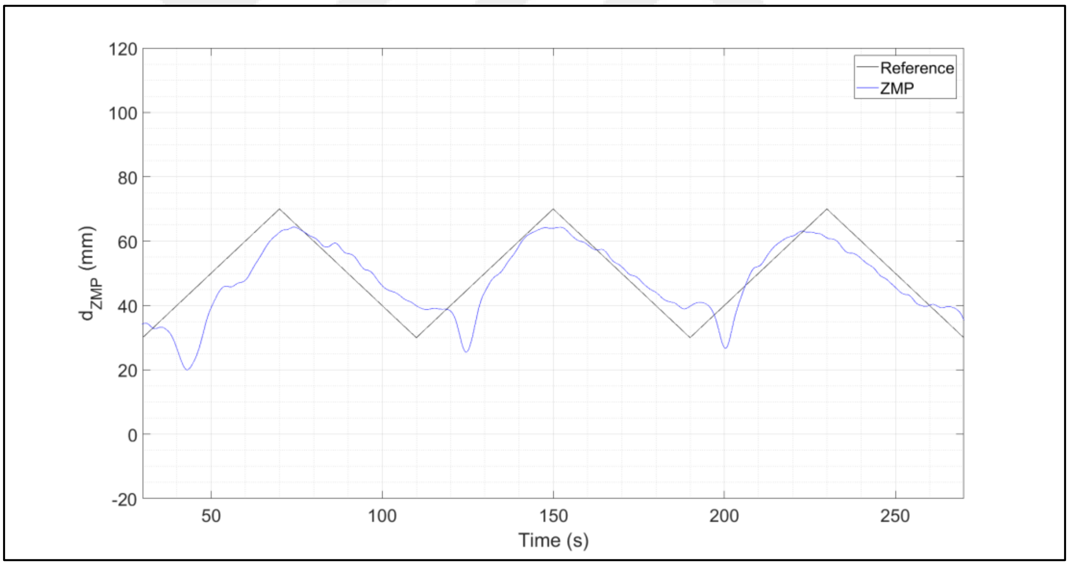

Figure 5.13. Triangular ZMP reference test of simulation for K=0.3 and τ=0.25

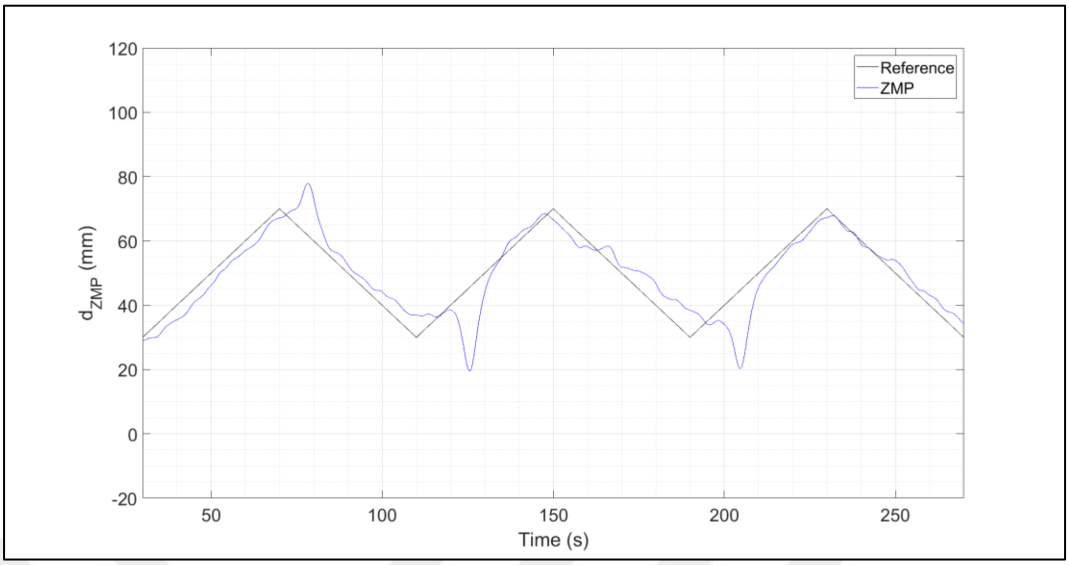

Figure 5.14. Triangular ZMP reference test of simulation for K=0.3 and τ =0.3

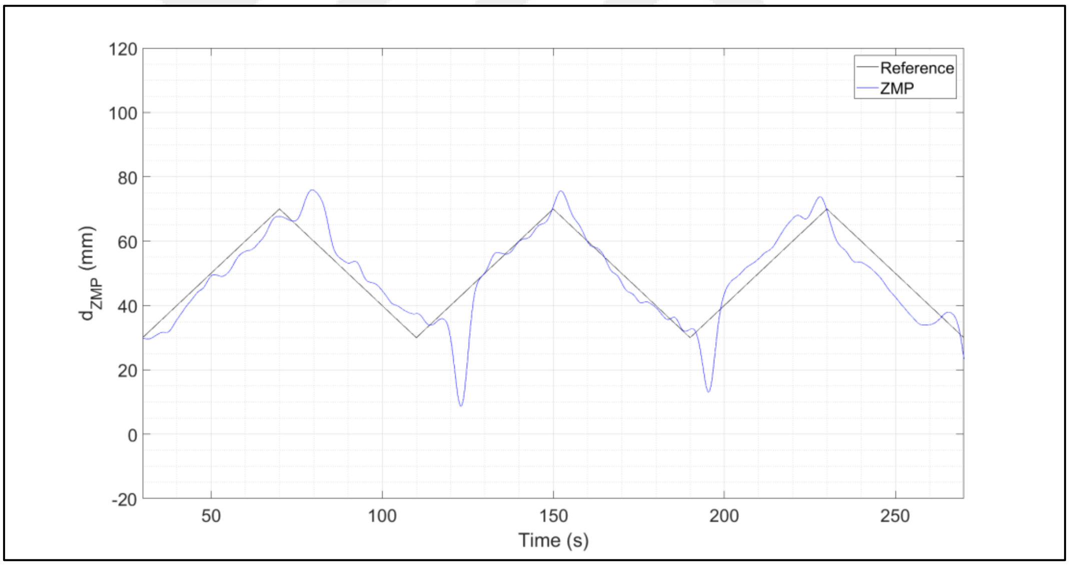

Figure 5.15. Triangular ZMP reference test of simulation for K=0.3 and τ=0.5

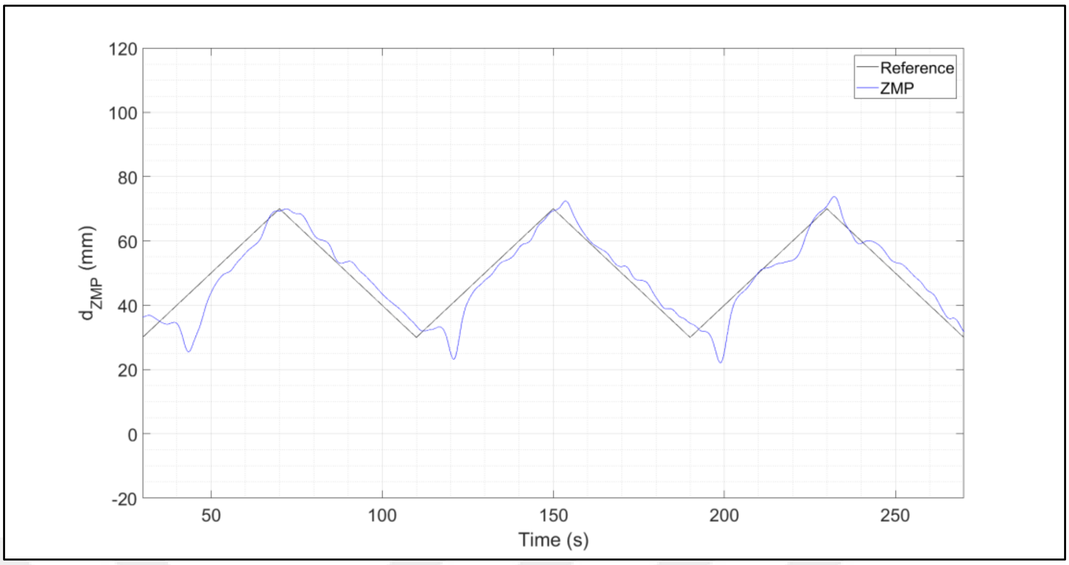

Figure 5.16. Triangular ZMP reference test of simulation for K=0.46 and τ=0.25

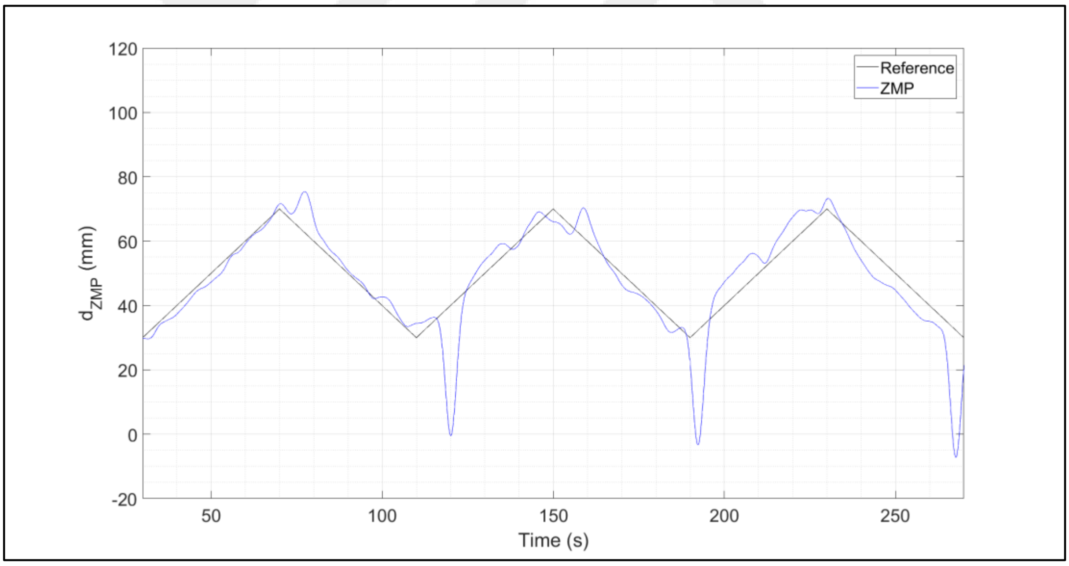

Figure 5.17. Triangular ZMP reference test of simulation for K=0.46 and τ =0.3

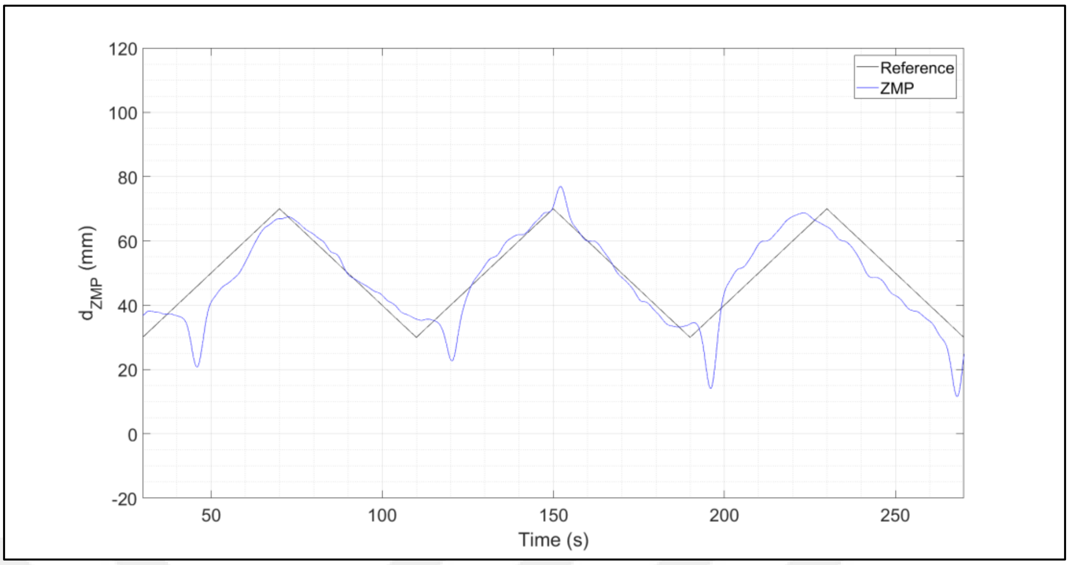

Figure 5.18. Triangular ZMP reference test of simulation for K=0.46 and τ=0.5

It is seen that higher **K** values with lower **τ** give the better results in triangular reference as well. However there are still undershoots in the system. In order to understand the effects of different *K* and *τ* constants, resulting experiment and simulation of actual ZMP trajectories were compared with their respective reference ZMP trajectories. From this comparison RMS error of the system were calculated and tabulated as shown in Table 5.1.

Table 5.1 ZMP RMS error (mm) of virtual robot for different K and τ constants

Square Wave Reference		***K***			**Triangular Wave Reference**		***K***		
		0.2	0.3	0.46			0.2	0.3	0.46
τ	0.25	30.903	29.639	21.226	***τ***	0.25	23.421	22.429	17.853
	0.3	29.639	26.069	22.021		0.3	24.429	23.095	19.120
	0.5	40.767	28.938	23.723		0.5	26.271	25.192	20.939

5.2. PROCEDURE 1 EXPERIMENTAL RESULTS

Similar to the simulation tests, total of eighteen tests were performed. In nine of these tests a square wave ZMP reference were given to YU-Bibot and combination of three different K and τ parameters were used when observing the resulting ZMP trajectory. In the other nine of the tests a triangular wave ZMP reference were given to the same combination of control parameters.

5.2.1. Square Wave ZMP Reference

Figures from 5.19 to 5.27 show the results of YU-Bibot’s response to square wave ZMP reference signal with different control parameters.

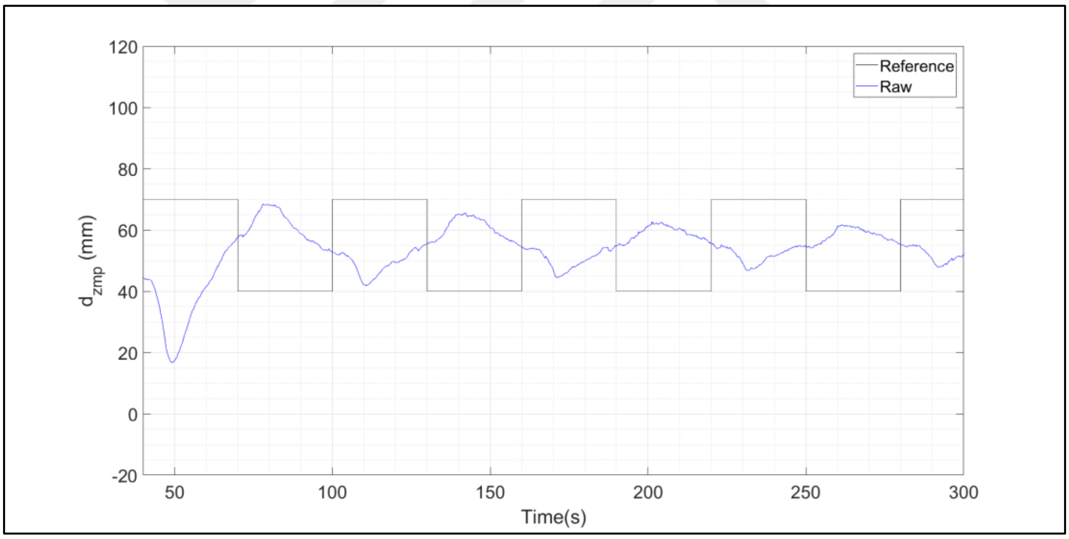

Figure 5.19. Square ZMP reference test of YU-Bibot for K=0.2 and τ=0.25

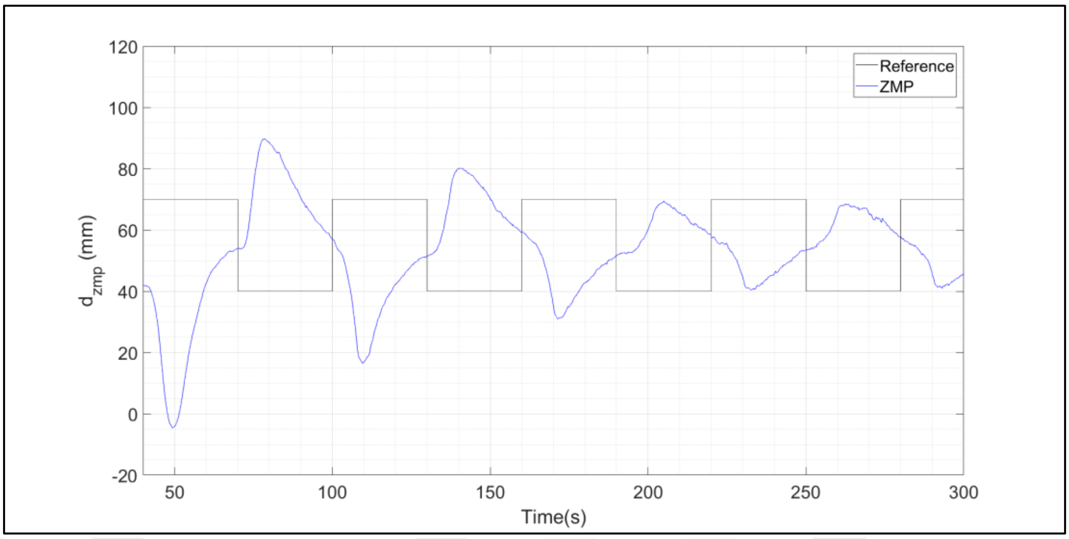

Figure 5.20. Square ZMP reference test of YU-Bibot for K=0.2 and τ =0.3

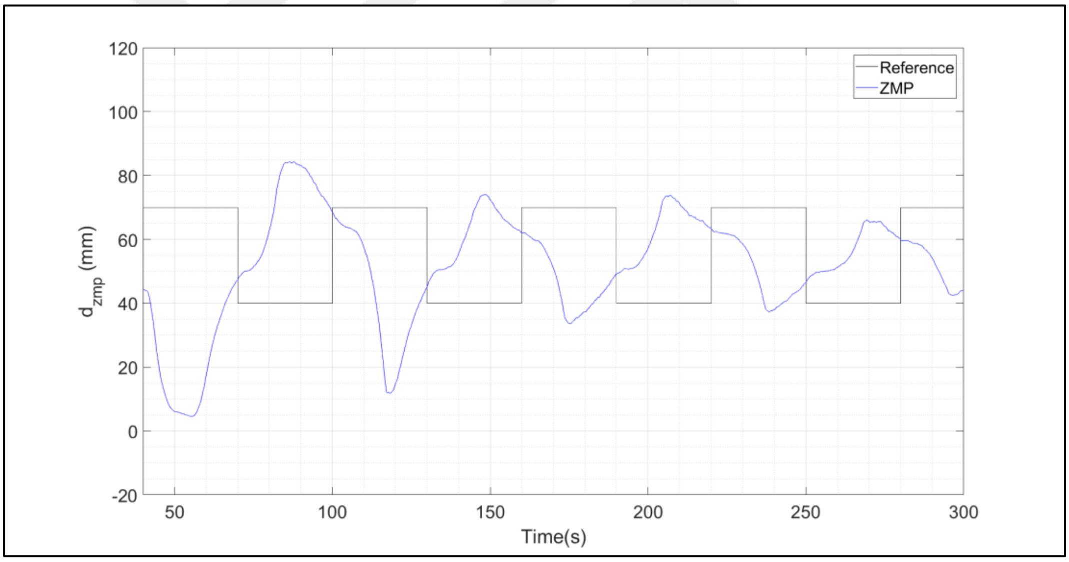

Figure 5.21. Square ZMP reference test of YU-Bibot for K=0.2 and τ=0.5

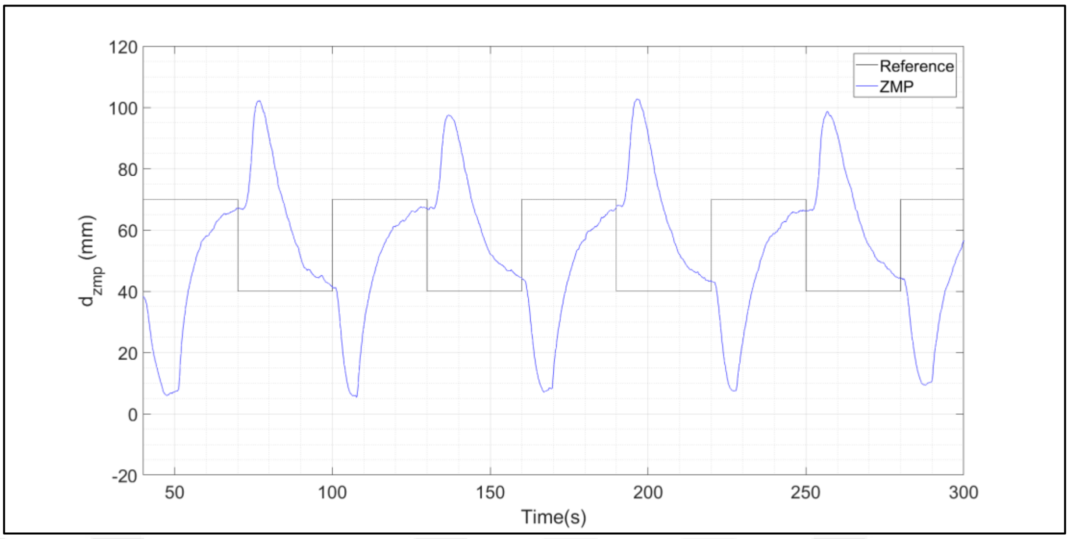

Figure 5.22. Square ZMP reference test of YU-Bibot for K=0.3 and τ=0.25

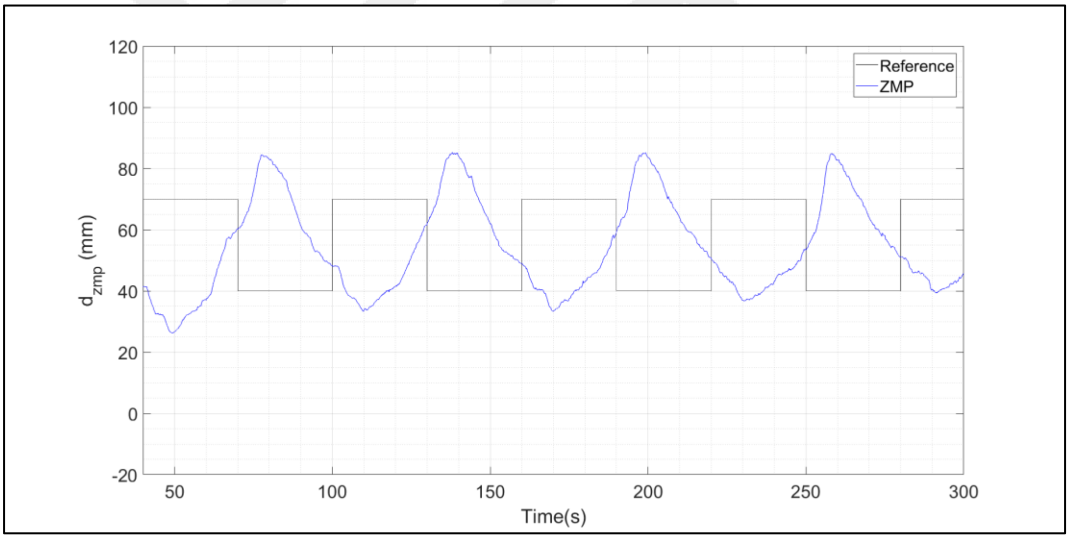

Figure 5.23. Square ZMP reference test of YU-Bibot for K=0.3 and τ =0.3

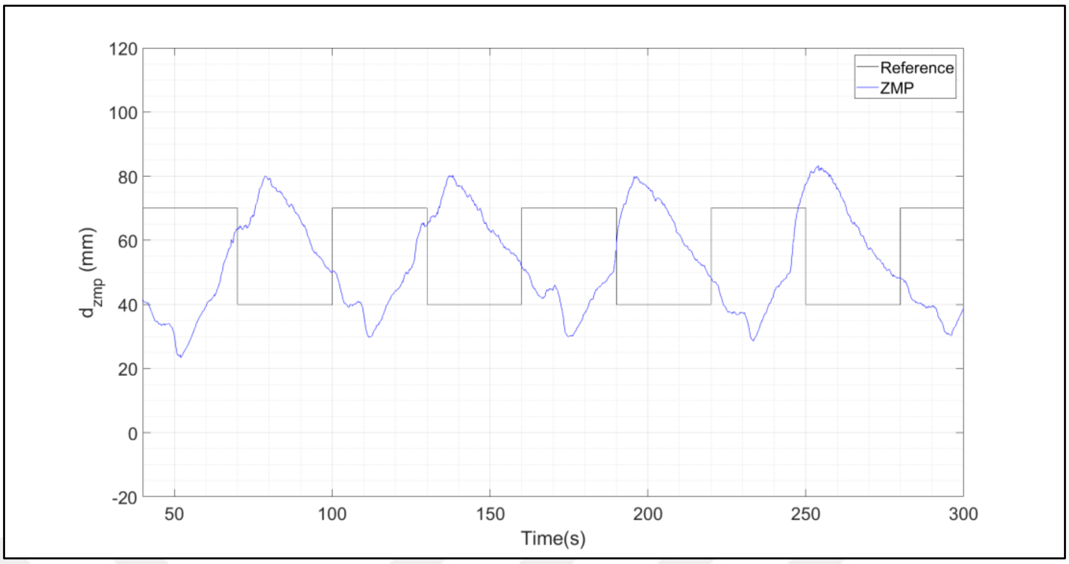

Figure 5.24. Square ZMP reference test of YU-Bibot for K=0.3 and τ=0.5

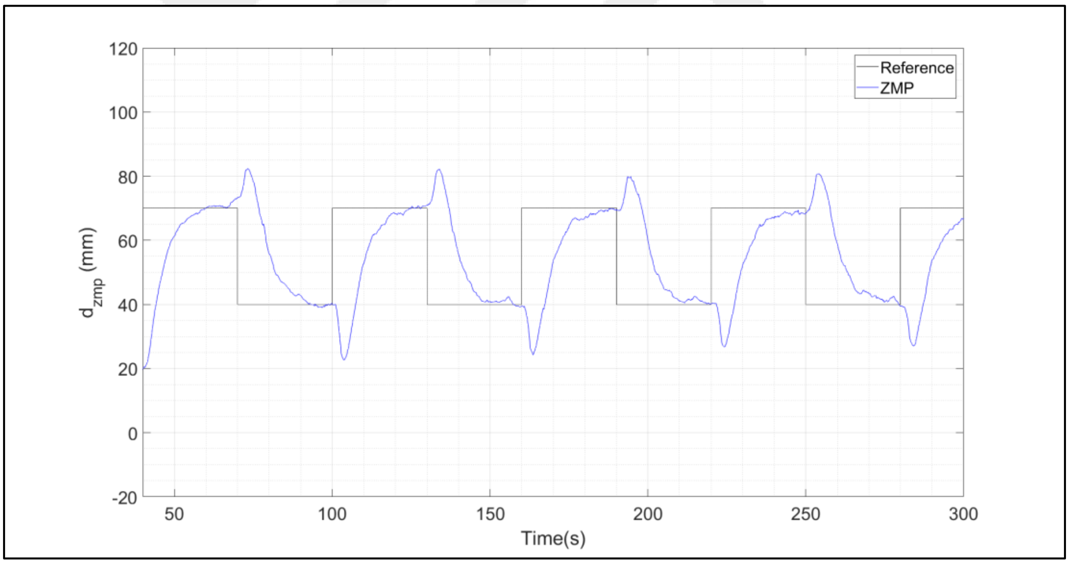

Figure 5.25. Square ZMP reference test of YU-Bibot for K=0.46 and τ=0.25

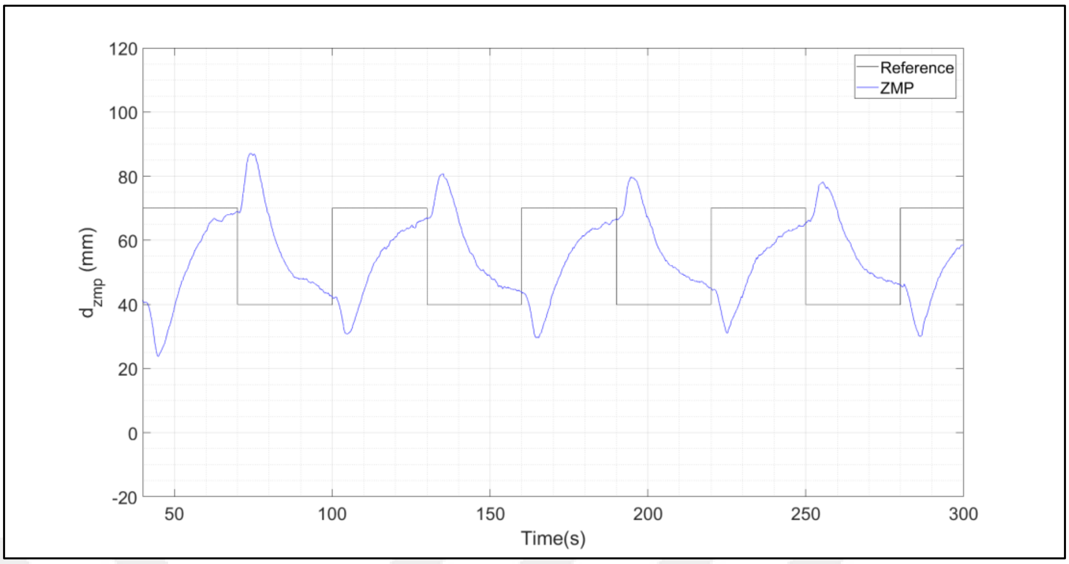

Figure 5.26. Square ZMP reference test of YU-Bibot for K=0.46 and τ =0.3

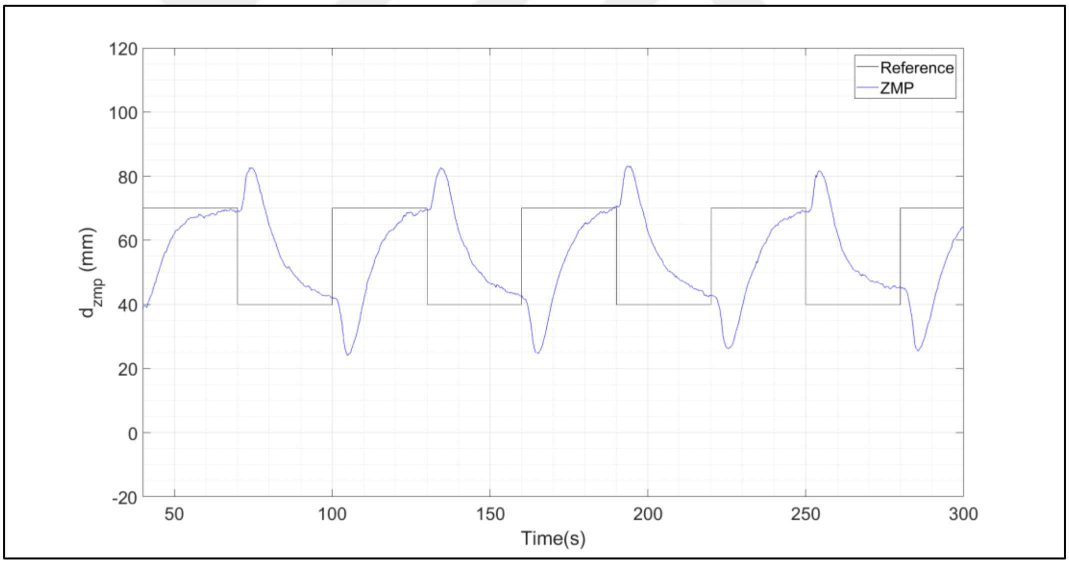

Figure 5.27. Square ZMP reference test of YU-Bibot for K=0.46 and τ=05

These control parameters cause a slow robot response. Also when reference ZMP signal changes its magnitude, undershoots happen in YU-Bibot. Because of the unideal control parameters and undershoot effects, YU-Bibot fails to catch up with reference ZMP signal. It can be observed that there is a significant steady state error and near 180° phase difference in YU-Bibot’s ZMP response.

5.2.2. Triangular Wave ZMP Reference

Figures from 5.28 to 5.36 show the results of YU-Bibot's response to triangular wave ZMP reference signal with different control parameters.

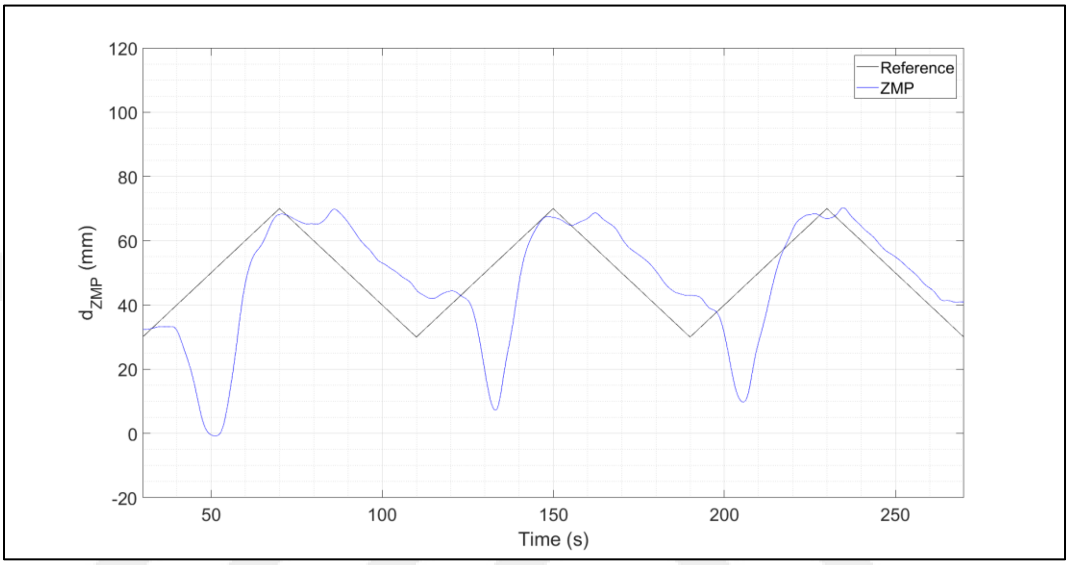

Figure 5.28. Triangular ZMP reference testof YU-Bibot for K=0.2 and τ=0.25

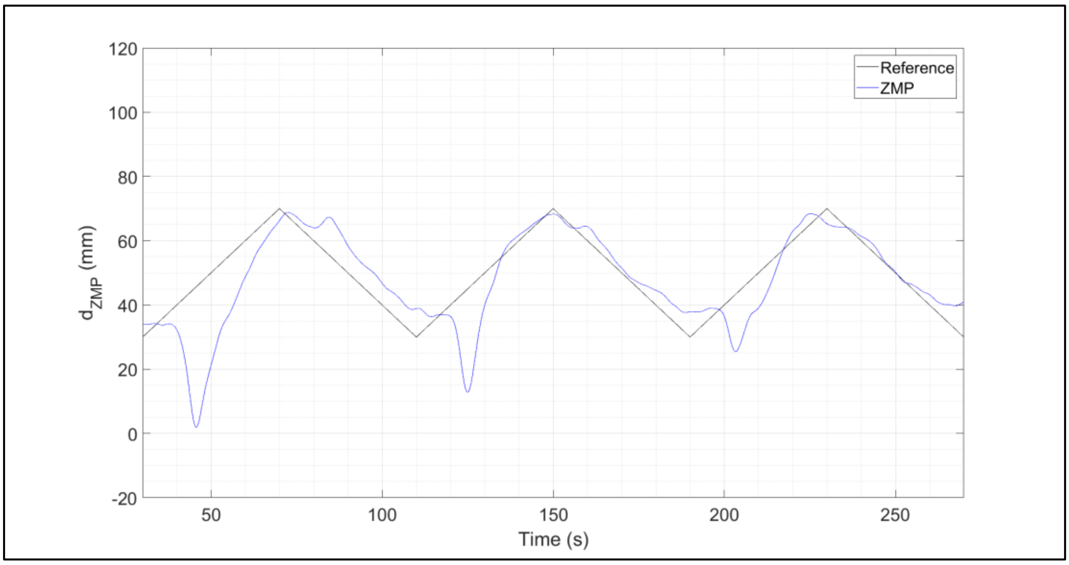

Figure 5.29. Triangular ZMP reference test of YU-Bibot for K=0.2 and τ =0.3

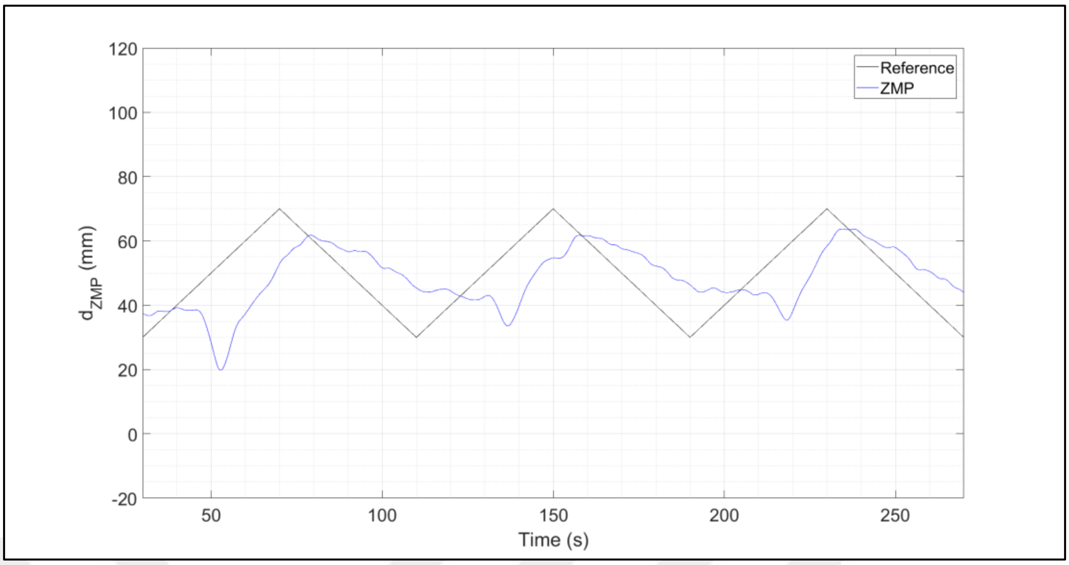

Figure 5.30. Triangular ZMP reference test of YU-Bibot for K=0.2 and τ=0.5

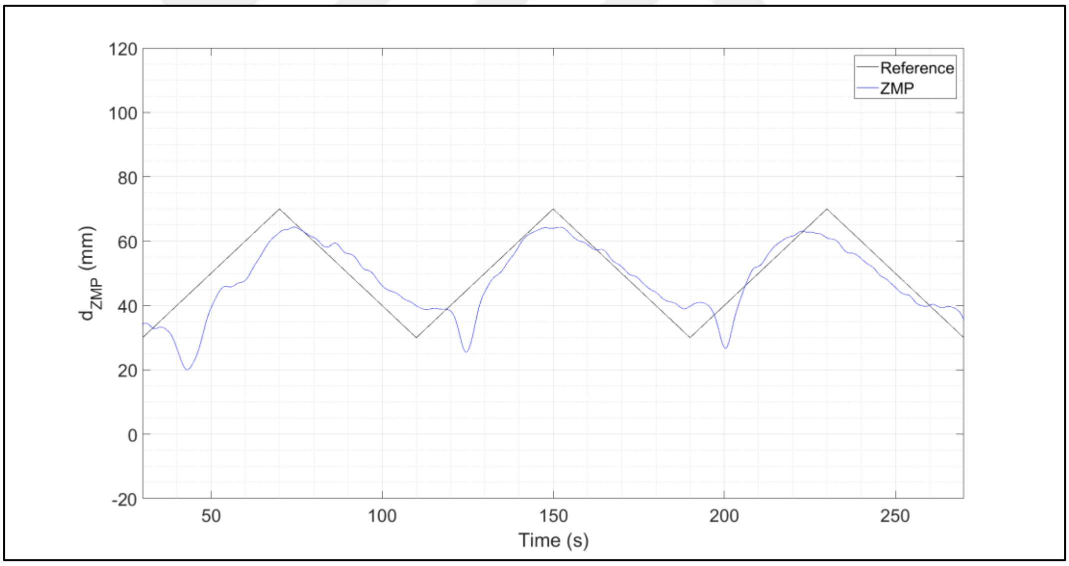

Figure 5.31. Triangular ZMP reference test of YU-Bibot for K=0.3 and τ=0.25

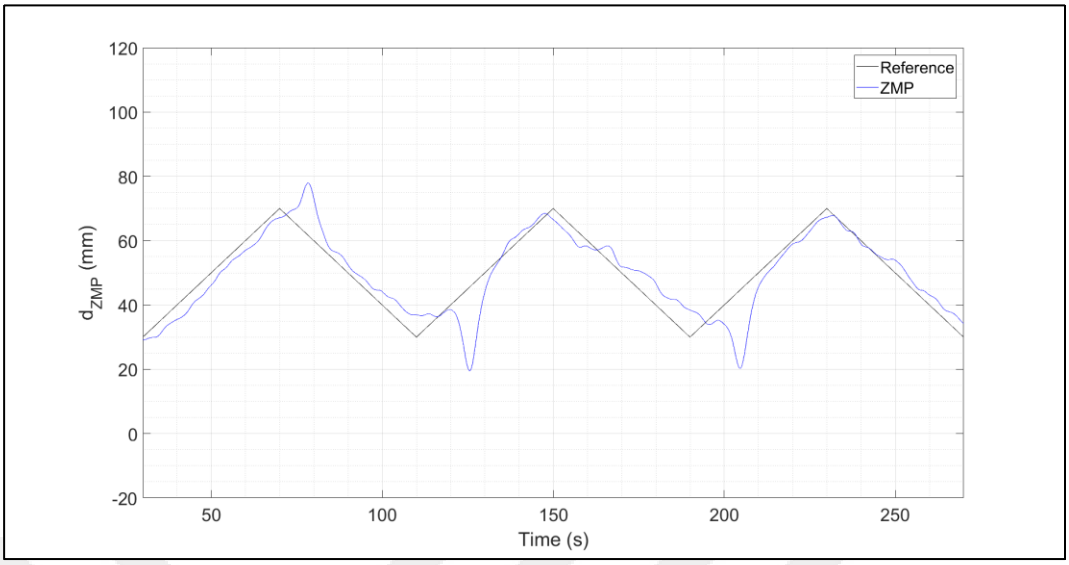

Figure 5.32. Triangular ZMP reference test of YU-Bibot for K=0.3 and τ =0.3

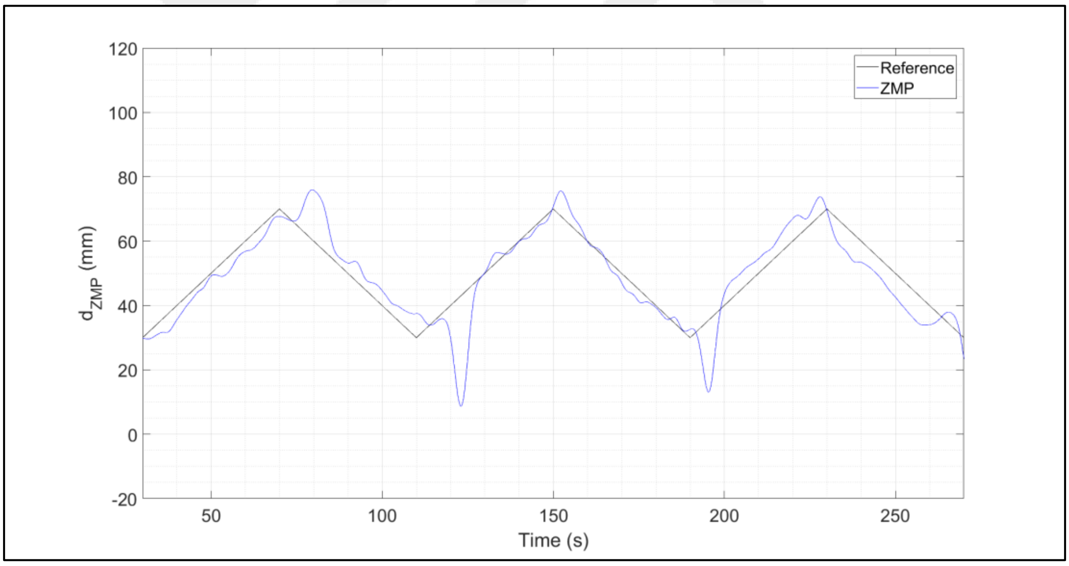

Figure 5.33. Triangular ZMP reference test of YU-Bibot for K=0.3 and τ=0.5

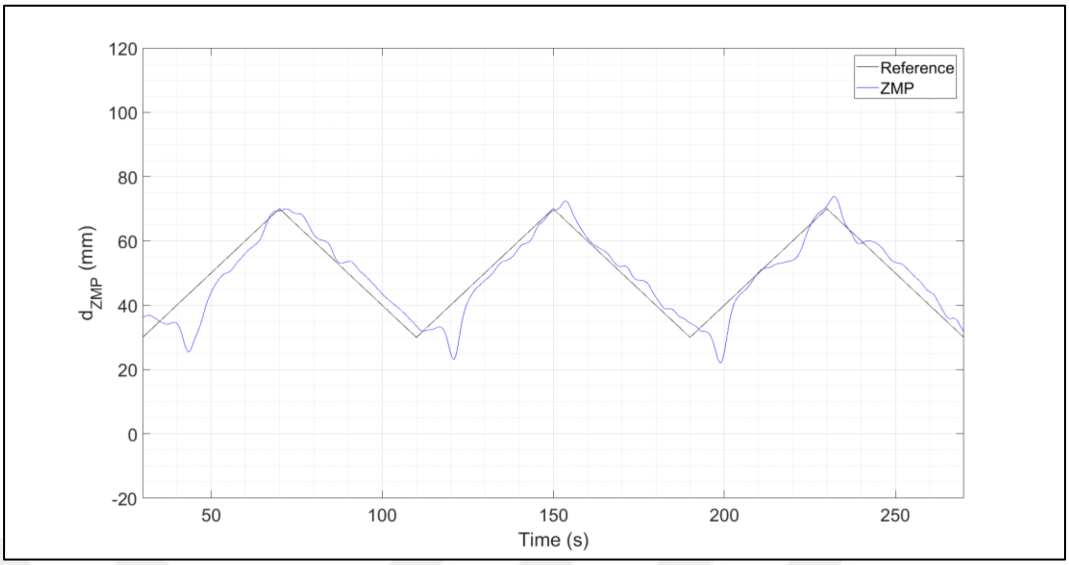

Figure 5.34. Triangular ZMP reference test of YU-Bibot for K=0.46 and τ=0.25

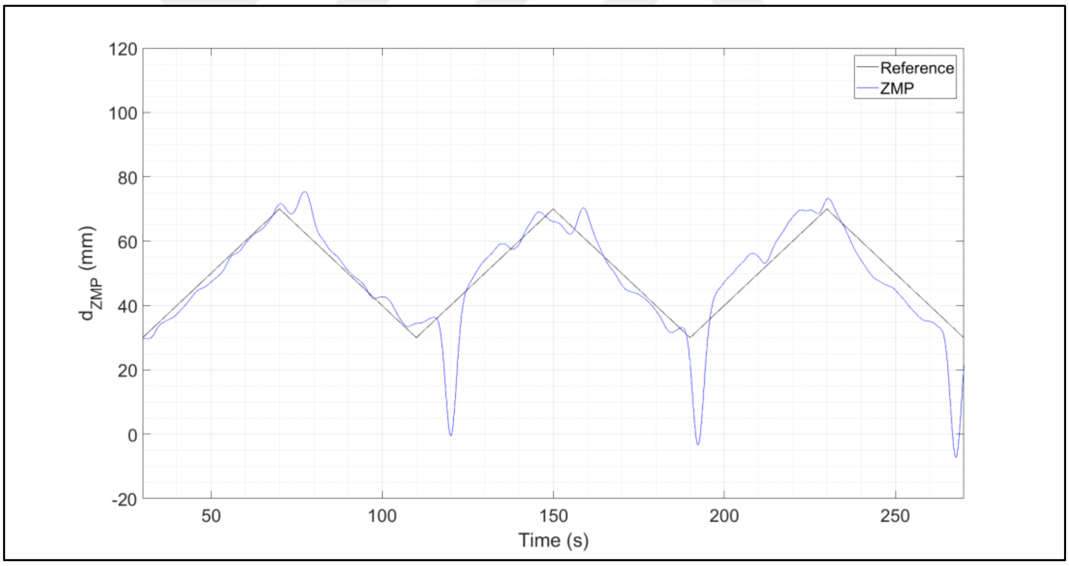

Figure 5.35. Triangular ZMP reference test of YU-Bibot for K=0.46 and τ =0.3

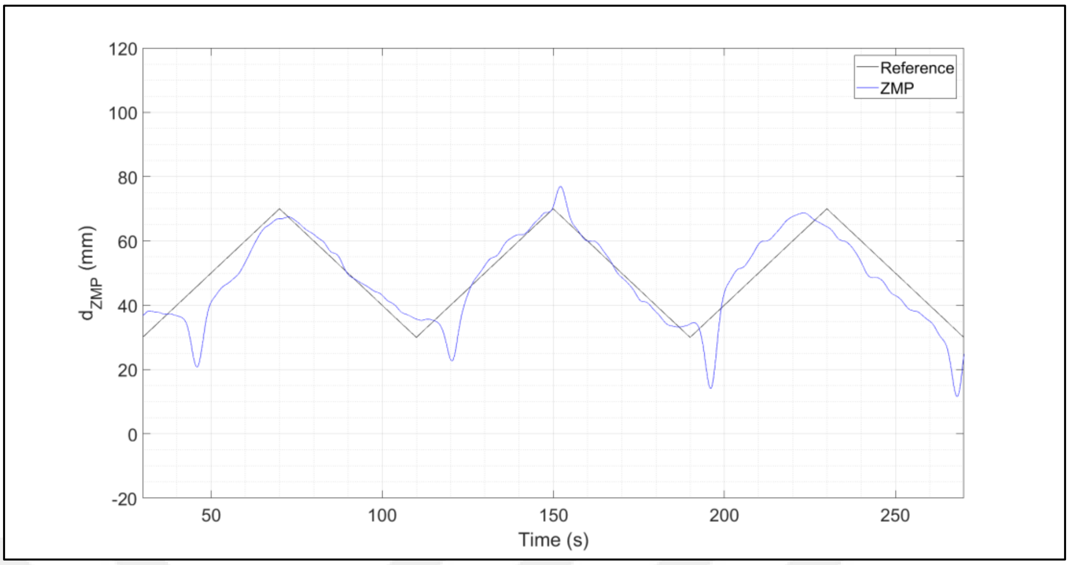

Figure 5.36. Triangular ZMP reference test of YU-Bibot for K=0.46 and τ=05

Similarly high **K** values with low ***τ*** give the best results. The effects of different *K* and *τ* constants were inspected by calculating RMS error of the system as shown in Table 5.2.

Table 5.2. ZMP RMS error (mm) of real robot for different K and τ constants

Square Wave Reference		***K***			**Triangular Wave Reference**		***K***		
		0.2	0.3	0.46			0.2	0.3	0.46
τ	0.25	20.75	31.40	20.71	***τ***	0.25	21.089	20.251	18.764
	0.3	29.65	24.52	21.62		0.3	22.511	21.247	20.064
	0.5	31.77	27.35	23.57		0.5	25.923	22.975	20.673

After these initial tests a PID controller was implemented before the first order filter in order to improve robot's response. In second test procedure robot's stands on the ground with its legs apart, where right foot was placed on forward. Similarly a square wave and a triangular were given as the reference ZMP.

5.3. PROCEDURE 2

After these initial tests a PID controller was implemented before the first order filter in order to improve robot's response. In second test procedure robot's stands on the ground with its legs apart, where the right foot was placed on forward. Similarly a square wave and a triangular were given as the reference ZMP. First the simulation of this procedure was performed which gave the following results in Figure 5.37 and Figure 5.38.

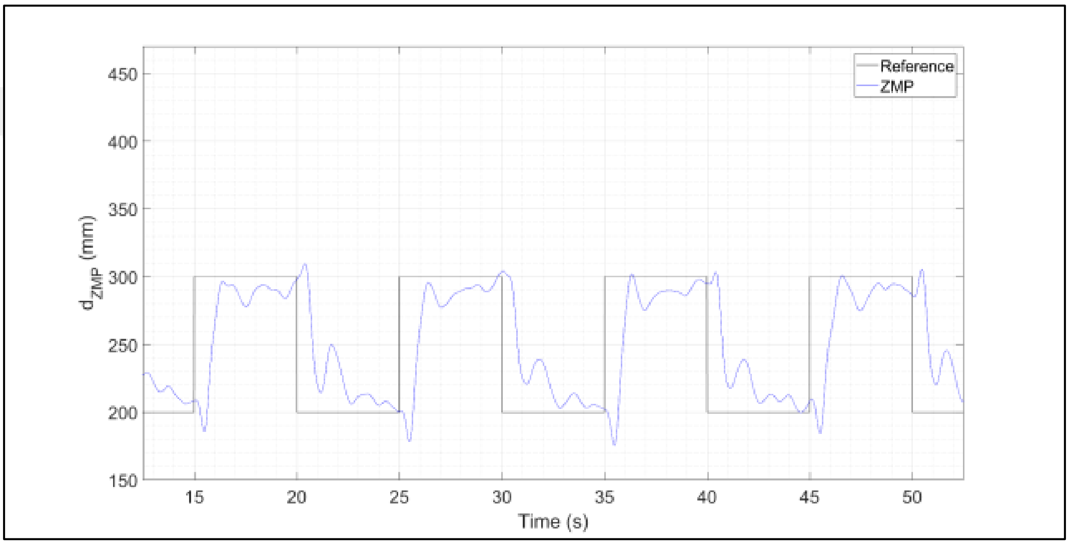

Figure 5.37. Square ZMP reference and acquired control response of the virtual robot

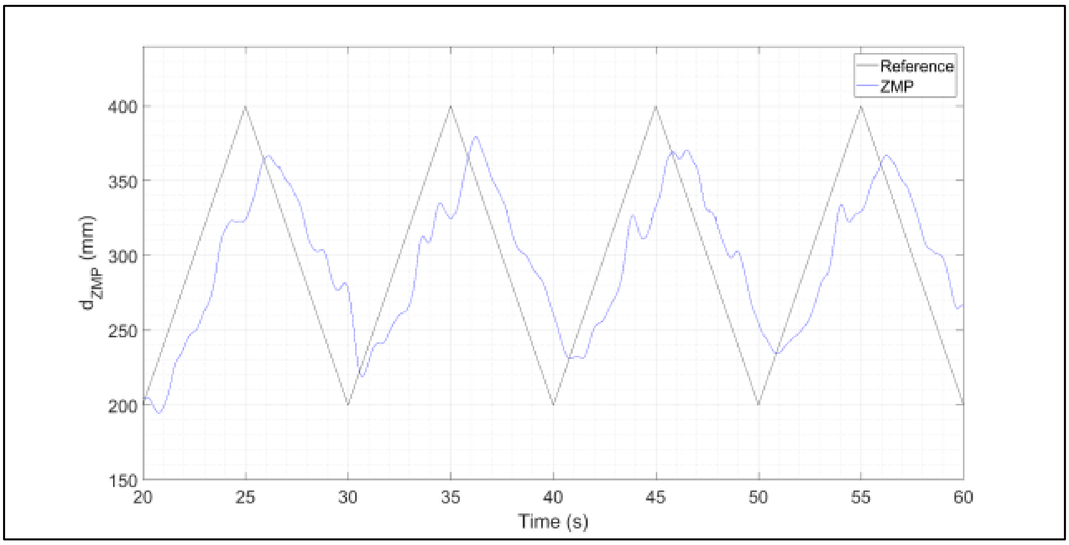

Figure 5.38. Triangular ZMP reference and acquired control response of the virtual robot

It can be seen that simulation results were promising where simulated YU-BiBot followed the given reference ZMP signal albeit some errors. Same procedure was then repeated in the real robot which gave similar performance as can be seen from the Figure 5.39 and Figure 5.40.

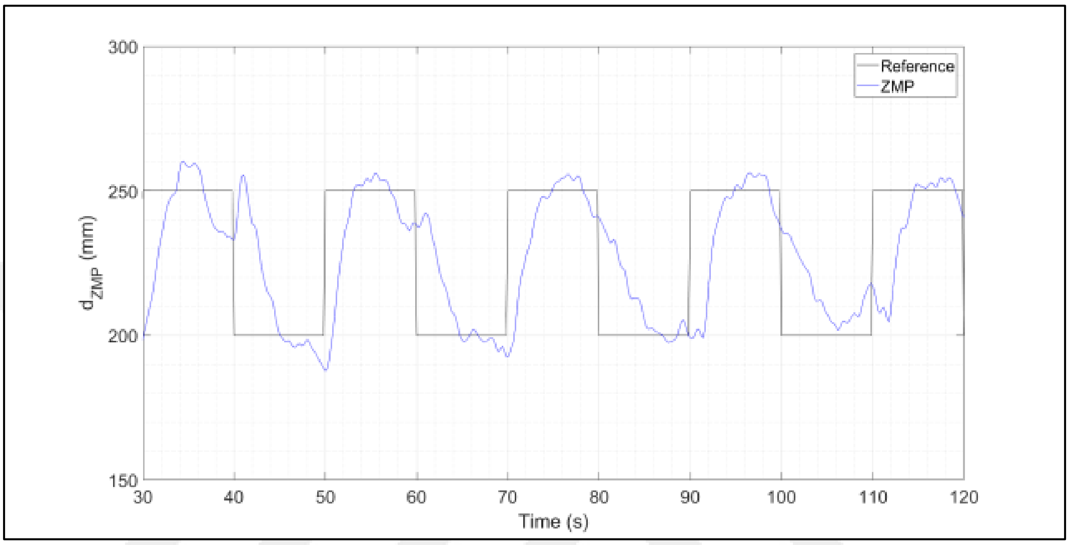

Figure 5.39. Square ZMP reference and acquired control response of the real robot

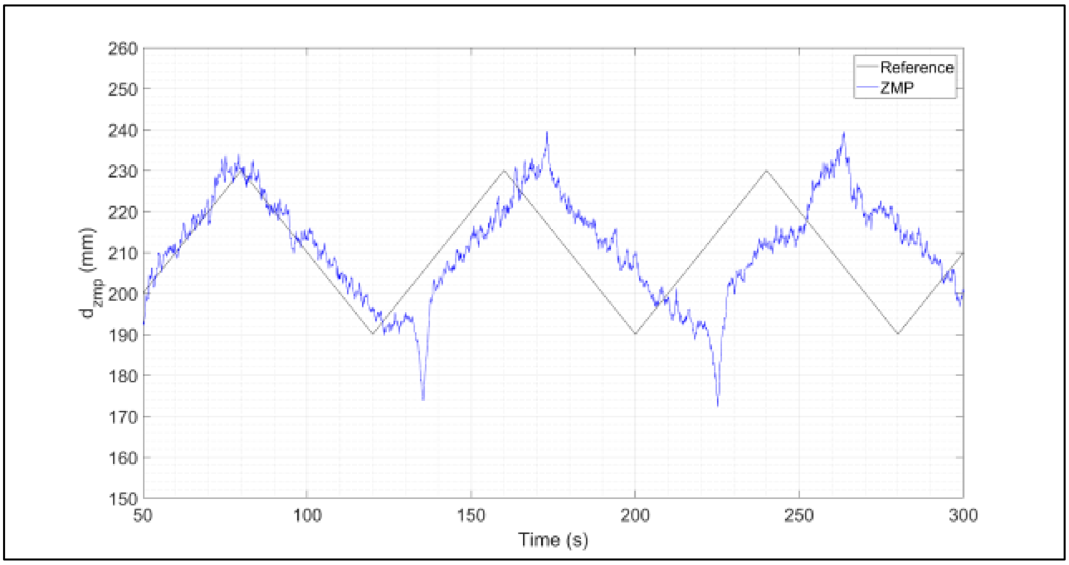

Figure 5.40. Triangular ZMP reference and acquired control response of the real robot

Results of the simulations and real experiments were than compared with their reference values and RMS errors were calculated and tabulated as shown in table 5.2

Table 5.3. ZMP RMS error (mm) of real and virtual robot

	Real Robot	**Virtual Robot**
Square wave	12.68	20.03
Triangular wave	19.18	22.76

5.4. PROCEDURE 3

In the third test procedure robot starts its movement similar to the previous procedure. However during the procedure robot lifts its left foot and swings it forward hence robot takes a single step forward. In order to achieve this procedure two reference signals are given to the system. First reference signal is sent to the inverse kinematic solver and it states the robot's swinging foot's condition. Second reference is a stair ZMP signal, which is used to adjust the hip position for the stability. Simulated robot was able to follow the given reference ZMP signal and performed its task while maintaining its balance as can be seen from figure 5.41.

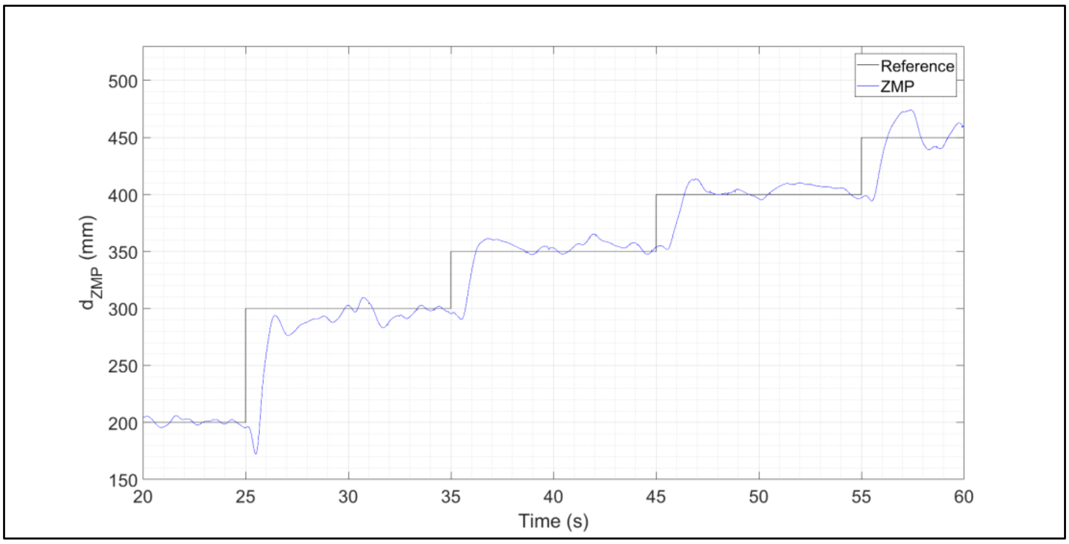

Figure 5.41. ZMP reference and acquired control response of the virtual robot

Although the performance of the simulation was promising, real robot failed to perform this task. In real experiments, when the left foot was raised above right ankle joint failed to adjust the hip position.

5.5. DISCUSSION

In the first two procedures, YU-Bibot was able to follow the given reference ZMP trajectory with a fairly good accuracy by using ZMP feedback, although the sensors employed were able to measure reaction forces under the feet with 10-15% accuracy. Comparing the simulation results with the real experiment results, it is presumed that imperfections in the robots joints and nonlinear friction effects cause the increased undershoots. In the third procedure robot failed its operation after elevating its left feet. Inconsistency between the real robot and simulation might be caused by some factors which were ignored in the simulation. The real robot is connected to a fixture system which consists of linear guides. It is presumed that the force transferred to the linear guides changes during the robot's motion. Therefore friction in the linear guide changes between static and dynamic friction types and stick-slip occurs. This friction effect and the spring in the ankles were not implemented in the simulation. During the operation measured ZMP started to decrease when the left foot was elevated as expected. It is thought that, the initial calculated response to this rapid decrease was not enough to overcome these added effects of the friction forces and the spring. This discrepancy can be attributed to the current and speed limitations in the robot joints. Therefore robot started to lean backwards and required torque to move the hip increased. Actuators in the right foot fail to provide this torque levels and therefore robot's operation fails.

.

6. CONCLUSION AND FUTURE WORK

A stable balance control method for a planar robot was proposed, simulated, tested, and verified in this study. It was shown that force sensitive resistors, placed underneath the robot feet, were suitable to provide required data for real time control. First procedure showed that higher filter gain values with lower filter time constants give better results. Second procedure showed that including a PID controller to the system decreases the RMS error. Third procedure showed that the proposed strategy can be used when system goes from double support phase to single support phase in theory. However real experiments of the third procedure had a failure. It is expected that, mechanical improvements in the joints and the fixture yield more precise control of the robot and will result in successful testing of the third procedure. After successfully completing the third procedure, a fourth procedure can be conducted where various ZMP references for walking gait can be given to the system and the optimum ZMP reference can be searched.

REFERENCES

1. Stephens B. Integral control of humanoid balance. *Intelligent Robots and Systems (IROS), 2007 IEEE/RSJ International Conference on;* 2007: IEEE.

2. Winter DA, Patla AE, Ishac M, Gage WH. Motor mechanisms of balance during quiet standing. *Journal of Electromyography and Kinesiology*. 2003;13(1):49-56.

3. Vukobratović M, Borovac B. Zero-moment point - thirty-five years of its life. *International Journal of Humanoid Robotics*. 2004;1(1):157-173.

4. Kajita S, Kanehiro F, Kaneko K, Fujiwara K, Harada K, Yokoi K. Biped walking pattern generation by using preview control of zero-moment point. *International Conference on Robotics & Automation, Proceedings of the IEEE*; 2003: IEEE

5. Sugihara T, Nakamura Y, Inoue H. Real-time humanoid motion generation through ZMP manipulation based on inverted pendulum control. *IEEE International Conference on Robotics and Automation, Proceedings of the IEEE*; 2002: IEEE.

6. Park JH, Rhee YK. ZMP trajectory generation for reduced trunk motions of biped robots. *International Conference on Intelligent Robots and Systems Innovations in Theory, Practice and Applications, Proceedings of IEEE/RSJ*; 1998: IEEE.

7. Zhang Y, Wang Q, Qiang W, Fu P. A new method of desired gait synthesis in biped robot. *The 3rd World Congress on Intelligent Control and Automation, Proceedings of IEEE*; 2000: IEEE.

8. Lingyun H, Zengqi S. Reinforcement learning method-based stable gait synthesis for biped robot. *Control, Automation, Robotics and Vision Conference, 2004 8th International Conference on*; 2004: IEEE.

9. Er MJ, Zhou Y, Chien CJ. Gait synthesis self-generation by dynamic fuzzy q-learning control of humanoid robots. *Systems, Man and Cybernetics, 2006 IEEE International Conference on*; 2006: IEEE.

10. Ding J, Yang M, Zhou J, Yao D, Xiao X. Robust real-time walking pattern generation with dynamical consistency: An analytical method combined with optimal solution. *Robotics and Biomimetics (ROBIO), 2017 IEEE International Conference on*; 2017: IEEE.

11. Hollinger A, Wanderley M. Evaluation of commercial force-sensing resistors. *New Interfaces for Musical Expression, 2006 International Conference on*; 2011.

12. Erbatur K, Okazaki A, Obiya K, Takahashi T, Kawamura A. A study on the zero moment point measurement for biped walking robots. *Advanced Motion Control Proceedings, 7th International Workshop on*; 2002: IEEE.

13. Kim D, Park GT, Seo SJ. Zero-moment point trajectory modelling of a biped walking robot using an adaptive neuro-fuzzy system. *Control Theory and Applications, Proceedings of the IEEE*; 2005: IEEE.

14. Choi K, Lee H, Lee M. Fuzzy posture control for biped walking robot based on force sensor for ZMP. *2006 SICE-ICASE International Joint Conference*; 2006: IEEE.

15. Sheng B, Huaqing M, Min D, Zhongjie Z. Modelling and control for a bipedal robot on slopes. *Transactions of the Institute of Measurement and Control*. 2012;35(7):910-921.

16. Lai WZ, Huang HP, Chen JH. Real-time control of a humanoid robot. *Advanced Robotics and Intelligent Systems (ARIS), 2017 International Conference on*; 2017: IEEE.

17. Li THS, Su YT, Liu SH, Hu JJ, Chen CC. Dynamic balance control for biped robot walking using sensor fusion, kalman filter, and fuzzy logic. *Transactions on Industrial Electronics*. 2012;59(11):4394-4408.

18. Rodriguez D, Rodriguez DA, Ramirez RE. An underactuated model of bipedal gait based on a biomechanical analysis. *22nd International Congress of Mechanical Engineering (COBEM 2013)*; 2013.

19. Song S, Tang C, Wang Z, Yan G. Active disturbance rejection controller design for stable walking of a compass-like biped. *Transactions of the Institute of Measurement and Control*. 2017.

20. Şafak KK, Baturalp TB. Design and analysis of a foot contact sensor for posture control of a biped robot. *Engineering Systems Design and Analysis, ASME 2010 10th Biennial Conference on*; 2010:693-701.

21. Çıgay CP, Şafak KK and Karadağ V. Control of balance using force sensitive resistors. *Turkish National Comittee for Automatic Control*; 2015:1120-1124.

22. CiA. CAN in Automation; [cited 2017 10 October]. Available from: https://www.can-cia.org/can-knowledge/can/can-history/.

23. Wikipedia. CAN bus; [cited 2017 10 October]. Available from: https://en.wikipedia.org/wiki/CAN_bus.

24. Canbus. Data link layer; [cited 2017 10 October]. Available from: http://canbus.pl/index.php?id=4&lang=ena