# Simulation of VSA-qbmove

N. Parnenzini
Department of Information Engineering
University of Pisa
Email: nicholas.parnenzini@gmail.com

C. Rosales, G. Grioli
Research Center "E.Piaggio"
University of Pisa
Email: carlos.rosales@for.unipi.it
giorgio.grioli@centropiaggio.unipi.it, bicchi@ing.unipi.it

***Abstract*—This work shows how to simulate a VSA-qbmove module within the Gazebo simulator. Variable stiffness actuators are ubiquitous in soft-robotics, and the need to simulate their behavior is fundamental in the design process and to develop applications. Up to the authors knowledge, no current robotic simulator has them as a feature for joint actuation, hence the motivation of this work. The Gazebo simulator is widely used in robotics due to its connectivity with the Robot Operating System (ROS) and its relation to the DARPA Robotics challenge (DRC), which makes it an ideal candidate to host the proposed simulation module. Thus, we briefly describe the inner mechanism of a single VSA-qbmove module for completeness, and then how that was implemented using the simulator capabilities. Furthermore, examples of how the module can be readily assembled in more complex structures are provided. The simulation module and assembled examples are freely available in the form a easy-to-use plug-in, and released open-source under the Apache license v2.0[1].**

***Keywords*—*Variable Stiffness Actuators, Gazebo, robotic simulation, modeling of soft-bodied robots***

## I. The VSA-qbmove module

The qbmove is a rotational variable stiffness actuator described in [1]. The module can be seen as a "black box" in the sense that set-points for position and stiffness refer to the output shaft. However, we briefly describe the inner mechanism to have an in-sight for simulation module implementation. A VSA-qbmove has two servomotors which can rotate in clockwise and counterclockwise direction. Two pulleys are attached to each of the servomotors. The elastic transmission is realized via four tendons and four extension springs. One end of each tendon is wrapped and locked on the output shaft, while the other end is wrapped and locked on a pulley. This aims to realize a nonlinear transmission between motors and output shaft. The result is an agonistic-antagonistic behaviour, depending on relative rotations of the two pulleys. Stiffness is varied via opposite rotations of the pulleys, loading or unloading the spring through tendon manipulation. Position is varied via identical rotation of the pulleys, moving the output shaft with the stiffness set as before.

## II. VSA-qbmove plug-in for Gazebo

Fig. 1 shows the simulation module corrsponding to a real VSA-qbmove module. Gazebo is a robotics simulator furnished with a powerful 3D viewer and open-source physics engines [2]. Models in Gazebo consists of a simulation description file (SDF) and a custom dynamic behavior can be written C++ in a plug-in style. The components of the simulation module are: (a) A *Mechanical interface* to simulate mechanical behaviour of the output shaft (Sec II-C); (b) An *Electrical interface* to simulate the control board (Sec II-B); and (c) A *Command interface* to simulate how set-points for output shaft are regulated (Sec II-A).

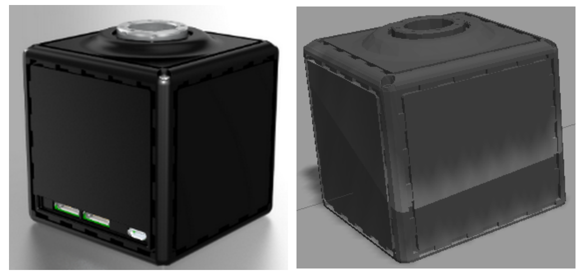

Fig. 1. The real VSA-qbmove module (left) and the simulation module in Gazebo (right)

### A. Command interface

Let $q$ and $s$ be the reference for position and stiffness of the output shaft, this block computes the reference position of the servomotors 1 and 2 as $q_{1_{ref}} = 2(q+s)$ and $q_{2_{ref}} = 2(q-s)$, respectively.

The reference position for each servomotor is then handled by a PID controller on the position error using a sampling time of $T_s = 0.001$, using $K_p = 0.001$, $K_i = 0$, and $K_d = 0.008$, the proportional, integral and derivative gains, respectively, to obtain a voltage command.

### B. Electrical interface

PWM modulation effects and h-bridge driver have been implemented as a dead zone and a saturation limit, with parameters set to $[-0.3, 0.3]$ and $[-1, 1]$, respectively.

Each servomotor is modeled using a state-space representation of a standard DC motor. The relevant parameters in the model are the motor inertia, $J = 0.001\text{Kg}\cdot\text{m}^2$, the motor torque constant, $K_t = 0.8$, the back-emf constant $K_b = 0.3$, the resistance, $R = 2.3\Omega$, and the inductance, $L = 2.4\text{H}$.

Electrical dynamics is too fast to be simulated by using Gazebo sampling time. Hence, we use a simplified 2-state model, considering position, $\theta$, and speed, $\dot{\theta}$, of each DC motor. The current, $i$, can be readily computed using $i = \frac{V}{R} - \frac{K_b}{R}\dot{\theta}$.

### C. Mechanical interface

At each servomotor, an external torque $\tau_{ext}$ is computed as $\tau_{ext} = \hat{\tau} - \tilde{\tau}$, where $\tau_k$ is a nonlinear spring torque

[1] Available online at https://github.com/valeria-parnenzini/qbmove_plugin

given by $\hat{\tau} = k\sinh[a(q - q_{motor})]$, with elastic constant $k = 0.022$N·m/rad and response parameter $a = 6.85$, that yields a torque proportional to the relative angle between the output shaft angle $q$ and the motor angle $q_{motor}$. Friction is considered in both the motors and the output shafts. In both cases, the model is taken from [3]. Force is modeled as a spring, with one end wrapped to the rotor, and the other end wrapped to the servomotor frame. The second end of the spring moves folllowing the movement of the first one when spring force is higher than static friction torque. Static friction is different between motors (0.8 Nm) and output shaft (0.01 Nm), because motor reductors friction is also included in motor models. Computing the extern torque of each servomotor as: $\tau_{ext_i}$ = $\hat{\tau}_i$ - $\tilde{\tau}_i$ as specified before, with $\hat{\tau}_i$ a the nonlinear spring torque and $\tilde{\tau}_i$ is friction torque of *i*-th motor, the *output shaft torque* $\tau_{out}$ can be computed as $\tau_{out}$ = - $\hat{\tau}_1$ - $\hat{\tau}_2$ - $\tau_{ft}$ and $\tau_{ft}$ is friction torque of output shaft, which is different from friction torque computed for servomotors.

## III. Validation

Figs. 2 and 3 show the output shaft tracking and step response using the simulation module w.r.t. the ground truth. The ground truth for the validation is the output of a Simulink© block already validated in our lab.

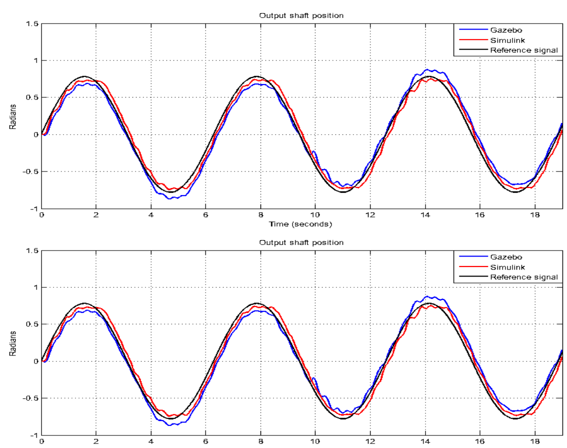


Fig. 2. Tracking-response of the output shaft position for set-point $q = A\sin(wt)$rad and two different stiffnesses: $s = 0.0$N/m (top) and $s = HIGH$N/m (bottom)

## IV. Applications of the VSA-qbmove plugin

We show two examples of how the VSA-qbmode can be readily assembled to form more complex structures in Fig. 4.

## V. Conclusion

We implemented a simulation module of a VSA-qbmove [1] in the Gazebo simulator [2] in the form of an easy-to-use plug-in[2]. The module was validated using previously accepted tools, and it is ready to be used in the design and applications involving this module. Complex structure are readily assembled and access to the command interface is done using the Gazebo interface.

[2]Available online at https://github.com/valeria-parnenzini/qbmove_plugin

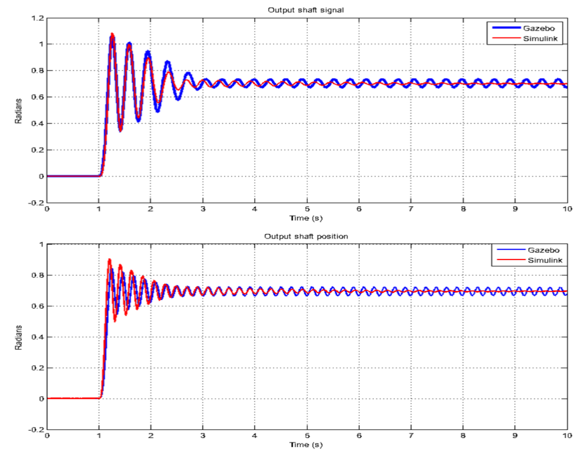


Fig. 3. Step-response of the output shaft position for set-points $q = 0.7$rad and two different stiffnesses: $s = 0.3$N/m (top) and $s = HIGH$N/m (bottom)

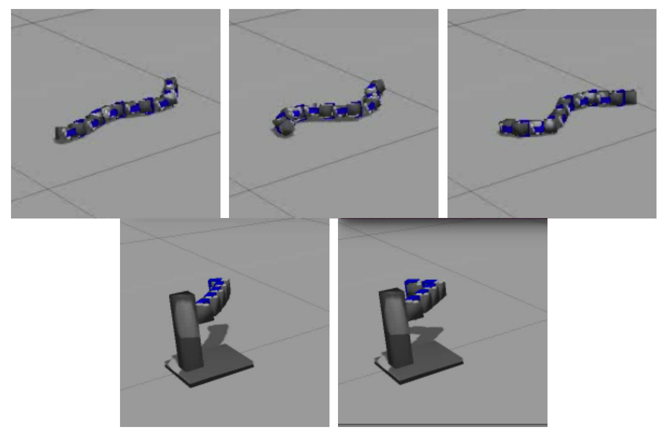

Fig. 4. A snake model composed of 12 VSA-qbmove modules (top) and a 3R manipulator arm composed of 3 VSA-qbmove modules (bottom)

## Acknowledgment

This work is supported by EU FP7-ICT project, WALKMAN, “Whole-body Adaptive Locomotion and Manipulation”, no. 611832.

## References

[1] M.G.Catalano, G.Grioli, M.Garabini, A.Bicchi: ”VSA-CubeBot: a modular variable stiffness platform for multiple degrees of freedom robots”, in Robotics and Automation (ICRA), IEEE International Conference on, may. 2011, pp. 5090 - 5095.

[2] Gazebo Robot Simulator, http://gazebosim.org/.

[3] Hayward, V. and Armstrong, B. ”A New Computational Model Of Friction Applied To Haptic Rendering”.