Improved control of a humanoid arm driven by pneumatic actuators

Joachim Schröder[1,2], Kazuhiko Kawamura[2], Tilo Gockel[1], and Rüdiger Dillmann[1]

[1] Industrial Applications of Informatics and Microsystems
University of Karlsruhe (TH), Karlsruhe, Germany
[2] Center for Intelligent Systems, Vanderbilt University, Nashville, TN, USA

Abstract. Favorable characteristics of pneumatic actuators, such as high force-to-weight ratio and safe interaction with humans are very suitable for robotic applications. At the Center for Intelligent Systems at Vanderbilt University, pneumatic actuators are used in our humanoid robot, called Intelligent Soft Arm Control (ISAC) and have been subject in several papers in the past [1–4]. Due to the high nonlinearity of the system combined with the serial kinematics, a fast and robust control is necessary to achieve the desired motion. Regular controllers seem to be overstrained with handling these problems, thus special control techniques are required. In this paper, an approach to improve a conventional PID control by implementing a cascaded model-based controller is presented. To judge the performances of both control types, a testbed was built and experiments were performed. The test results are discussed in this paper.

1 Introduction

1.1 ISAC Humanoid Robot System

At the Center for Intelligent Systems at Vanderbilt University, one research emphasis is the development and control of humanoid robots. A testbed is the Intelligent Soft Arm Control (ISAC, Figure 1). ISAC consists of a human-like upper body with two six-degree-of-freedom manipulators, called Soft Arms [5]. Each Soft Arm is a robotic manipulator that is activated by artificial muscles. These kinds of actuators feature special properties, e.g. spring-like characteristics and cause a very natural movement. They are lightweight, safe to operate and have therefore a high potential to be used in direct interaction with humans.

Especially the fulfillment of safety precautions offer a wide range of applications in the service sector for ISAC. In this field, ISAC can be used as a human aid in hospitals or even to assist lightly disabled persons at home.

1.2 ISAC Architecture

Software architecture for ISAC is a real-time multi-agent based distributed system called the Intelligent Machine Architecture (IMA) [6] as shown in Figure 2.

The modules are encapsulated in different agents, which handle a variety of human-robot interaction and real-time robot control tasks. A set of agents are able to run on different machines with different operating systems. In our case, the real-time applications like the control are running under QNX, while speech or video processing is under Windows. The DCOM/TAO bridge is used to complete the communication between those systems. Both arm controls are running separately from each other in right and left arm agents. Instead of using a real arm, the input for the controller can be send to a simulator where the same control algorithms, combined with dynmical actuator and arm models, are implemented.

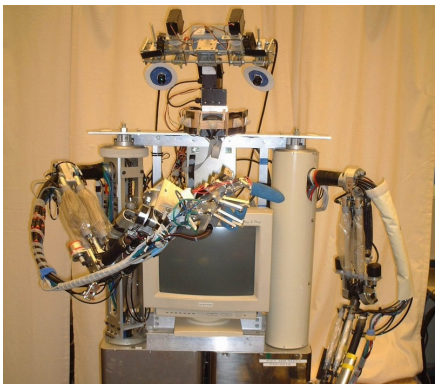


Fig. 1. Humanoid robot system ISAC

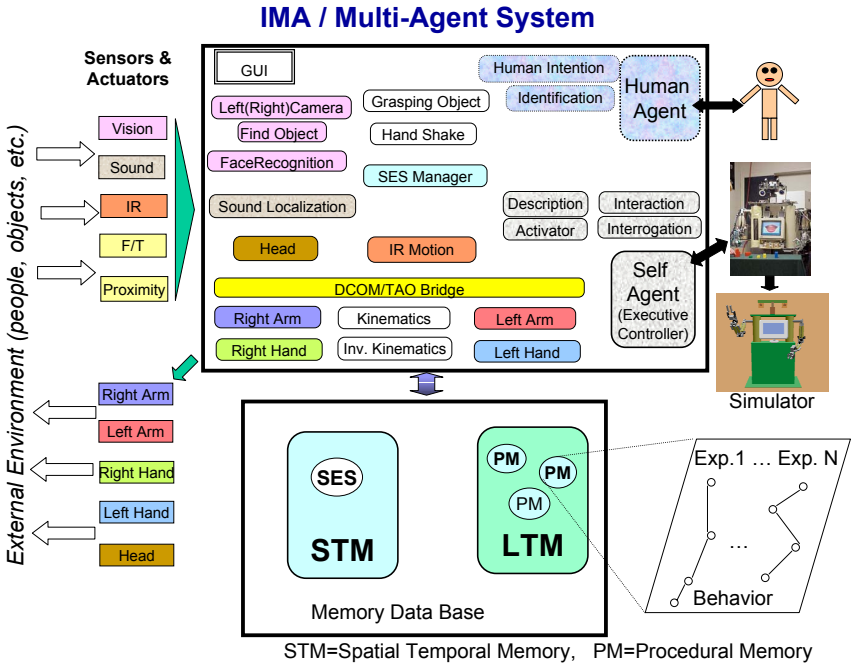
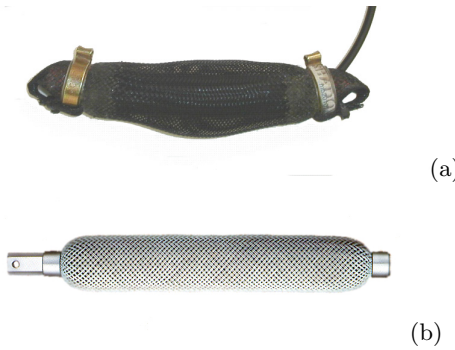


Fig. 2. Intelligent Machine Architecture (IMA)

1.3 Artificial Muscles

In 1957, the physician Dr. Joseph L. Mc Kibben developed an artificial actuator to improve his daughter's grasping ability [7]. The pneumatically actuated muscle consisted of a inner rubber tube and a outer cord netting. The inner tube was expanding when inflated, and the cord which was inexpandable transferred this change of volume in a change of length. The compressible air made the muscle elastic and safe for direct interaction with humans. Construction and materials differ from manufacturer to manufacturer, but the principle remained always same. Figure 3 shows two different artificial muscles manufactured by Shadow and Bridgestone.



(a)

(b)

Fig. 3. Shadow (a) and Bridgestone (b) artificial muscles

In the 1960's, electrical motors replaced the Mc Kibben muscle as an orthodic device, because of the smaller design and the easier way to control. The interest in artificial muscles became higher when Bridgestone Company redesigned the Mc Kibben muscles and distributed them as Rubbertuators.

The advantages are, as described in the section above, a spring-like behavior of the muscles, lower stiffness compared to industrial robots and hereby a safer use in human environments. Artificial muscles cause, due to the elasticity, a very natural and smooth movement of the arm. But working with artificial muscles means also dealing with some handicaps. Due to friction between the outer cord and the inner rubber tube, the relation between pressure, contraction and force contains a hysteresis. The nonlinear behavior complicates the control of the muscles even more, so that higher control techniques are required to operate these actuators.

1.4 Control of Artificial Muscles

Due to the complexity of artificial muscles, no obvious control architecture is recommended. In fact, many different control strategies are possible, also depending on the application. In [8], a regular PID controller was implemented to

control pneumatic muscles. The performance of this controller was quite sensitive to errors caused by the muscle hysteresis. The same authors followed another approach in [9], where a pole-placement controller was developed. A polynomial model whose parameters were estimated at each sampling interval was used to determine the controller poles.

One problem in controlling artificial muscles is the development of an accurate muscle model. In [5], a feed-forward neural network was applied to one joint driven by pneumatic muscles. Other researchers implemented a Kohonen network to solve position control problems [10]. These approaches did not assert themselves because they needed too many training examples, and therefore too much time to teach the neural network. An additional problem is that neural networks are trained for specific inputs. It is impossible within a limited time to teach a neural network all possible inputs of a six-degree-of-freedom arm to guarantee robustness and safety of the control. The application of neural networks for controlling individual joints with repeating control inputs might in contrast be useful.

In this paper, the classical position control approach is retained because it has proven itself to be reliable for many years. To accommodate the special properties of pneumatic actuators, the standard PID position control is extended to a cascaded control architecture.

2 Actuator Testbed

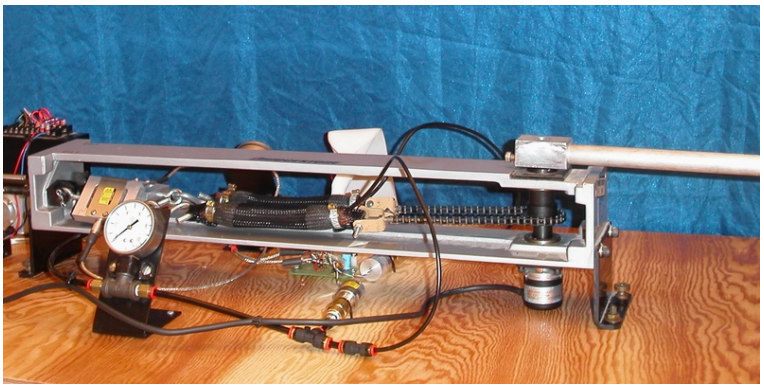


Fig. 4. Actuator Testbed

To generate an actuator model and to verify it, a testbed was built (Figure 4). This testbed was also used to implement the joint control, for tuning the control parameters and to compare the different architectures. To measure the resulting torque, two force sensors (Futek L2353 [11]) are integrated between the end of the

muscles and the testbed frame. External pressure sensors (Futek P4010, [11]) are placed as close to the muscles as possible to measure the real muscle pressures and to avoid time delays or pressure loss because of long tubes. An optical encoder (Sumtek, 8000 steps/turn) is used to determine the actual joint angle. Over an encoder buffer, the encoder signal is read by a PC interface board, which has been developed in the CIS lab [12]. Figure 5 illustrates an overview of the components. Pressure is given from a Bridgestone servo valve unit (SVO 102-A06) that receives the reference pressures via a 4-20mA input signal from the PC interface card. A National Instruments multi-IO card (Lab PC 1200) reads the signal of force and pressure sensors into the PC.

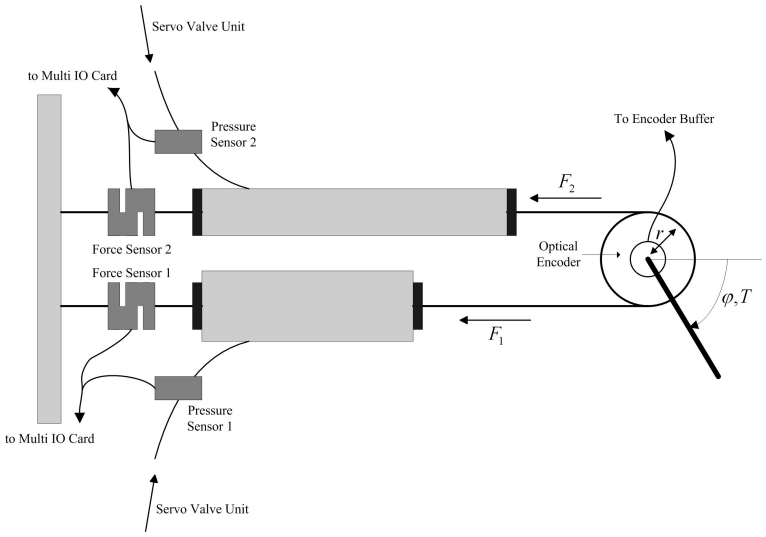


Fig. 5. Testbed Components

The Shadow Robot Company offers three different types of muscles with initial lengths of 150 mm, 210 mm and 290 mm. In the testbed, 210 mm muscles (l_{max}) are mounted with a pre-tension of about 15%, half of the maximum contraction ratio which is 30%. The diameter of the chain wheel is r=0.019 m. The muscles are driven with a initial pressure of 3 kg/cm^2, Δp varies between ± 2 kg/cm^2. The initial muscle length is l_0 =0.85 l_{max}, because of a chosen pre-tension of 15%.

3 System Description

In this section, modeling of system parts is described. Transfer functions of the different system parts are required for controller design and to proof stability.

The system can be divided into pressure control unit, artificial muscles (actuator) and the joint itself. As shown in Figure 6, transfer functions of actuator and arm are coupled. The mathematical description of the actuator is therefore combined with the equation of motion of the joint to receive one transfer function.

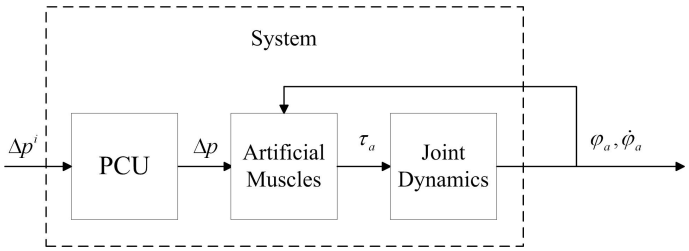


Fig. 6. Different system parts

3.1 Actuator Modeling

A mathematical description of the actuator, consisting of two artificial muscles, is developed in this part. The derivation is treated very briefly, because this was subject of another paper and is explained in [13] more detailed. Basis for the actuator model was the original Bridgestone muscle model.

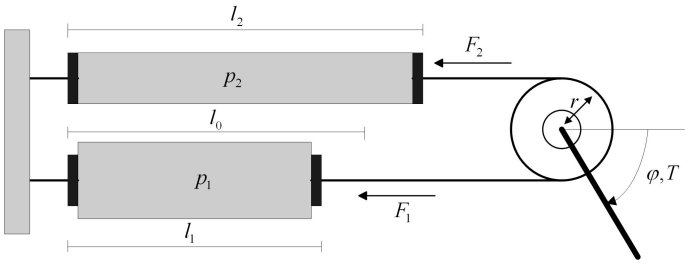


Fig. 7. Alignment of artificial muscles in one joint

The formula published by Bridgestone is:

$$F = {D_0}^2 p[a(1-\epsilon)^2 - b] \tag{1}$$

Where F is the force generated by the muscle, p is the internal muscle pressure, and ϵ is the contraction ratio. D_0 is the muscle diameter before displacement and a and b are muscle parameters.

According to the layout of the Rubbertuators, shown in Figure 7, and using the relations $\tau = (F_2 - F_1)r$, equation 1 as well as $\varphi = \frac{l_1 - l_2}{r}$ and $\epsilon = \frac{l_{max} - l}{l_{max}}$, the simplified description for one joint is:

$$\tau = a' \Delta p r^3 \varphi^2 + b' r^2 \varphi + c' \Delta p r \tag{2}$$

Where a', b' and c' are new joint parameters, r is the radius of the chain wheel.

The nonlinear part in equation 2 was only significant for larger joint angles (near joint limits) as well as high $\Delta p's$. Because these areas are not in the usual operating range, this part can be disregarded in the actuator model. A dynamic part was added because experiments showed that the muscles have damping characteristics, so that especially higher velocities change the actuator's behavior and have to be considered. The model was, according to experimental results, changed to:

$$\tau = Ar^2 \varphi + Br \Delta p + C\dot{\varphi}^3 + D\dot{\varphi} \tag{3}$$

Parameters of the model have been identified as:
$A = -23450 \frac{N}{m}$, $B = 174.2N \frac{cm^2}{kg}$, $C = 0.0026Nms$ and $D = 0.14Nms$.
The detailed derivation of this formula can be found in [13].

3.2 Modeling the control unit

The transfer function of the pressure control unit (PCU) was determined experimentally. The air tubes are assumed to be a part of the PCU to simplify the system identification. A reference current value, corresponding to 5 kg/cm^2 was given as a step input to the pressure control unit. Figure 8 shows the response, which identified the pressure control unit as a first order lag element.

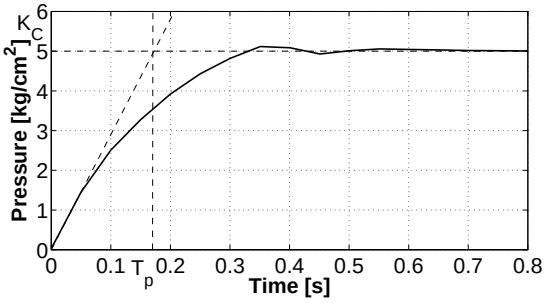


Fig. 8. Response of pressure control unit on a step input

The equation for the PCU transfer function is therefore:

$$F_C(s) = \frac{K_C}{1 + T_C s} \tag{4}$$

The two parameters K_C and T_C were identified as $K_C = 1$ and $T_C = 0.17$ s. The experiments were carried out with a tube length of about 5 m between pressure control unit and artificial muscles. This length matches approximately with the length of the air tubes in ISAC. The lag time T_C is, depending on the reference pressure or a changing tube length, sometimes shorter than $0.17s$, so that this is representing the worst case. Thus we deal with the actuator element, the real time lag is allowed to be shorter as assumed because the control speed and quality would benefit if so.

3.3 Joint Modeling

The simple equation of motion for the joint is:

$$J\ddot{\varphi} - \tau = 0 \tag{5}$$

or with equation 3 and disregard of $C\dot{\varphi}^3$ what affects only high velocities:

$$-\frac{J}{Ar^2}\ddot{\varphi} + \frac{D}{Ar^2}\dot{\varphi} + \varphi = -\frac{B}{Ar}\Delta p \tag{6}$$

Where J is the moment of inertia of the joint which is mainly given through a metal bar, used to indicate the joint position. The moment of inertia was computed as J=0.0085 kgm^2, referred to the joint axis. Equation 6 has the structure of a second order lag time element. A comparison of coefficients with a regular form of a second order element

$$T_J{}^2\ddot{\varphi} + 2\zeta T_J\dot{\varphi} + \varphi = K_J\Delta p \tag{7}$$

leads to the following values:

$T_J = \sqrt{-\frac{J}{Ar^2}} = 0.032s$, $\zeta = \frac{D}{2Ar^2\sqrt{-\frac{J}{Ar^2}}} = 0.26$ and $K_J = -\frac{B}{Ar} = 0.3$.

The resulting transfer function of the joint is:

$$F_J(s) = \frac{K_J}{1 + 2\zeta T_J{\cdot}s + T_J{}^2{\cdot}s^2} \tag{8}$$

To judge the controllability of a system and to tune the control parameters, the relation of the parameters latency T_L and adjustment time T_A are helpful. The latency is the time that passes, until the changing input value causes an output, which is appreciable different from zero. This point is defined as the intersection of the time axis and the tangent in the inflection point of the system response. The adjustment time is the time that passes between the dead time until the tangent at the inflection point cuts the abscissa given through K_J.

To determine these two variables, the step response of the joint was recorded. A problem when accomplishing this experimentally is, that it is de facto not possible to give a step to the actuator.

experiment is to give a step input to the muscles by switching directly to an air supply with the reference pressure and a sufficient in the wires.

One way to see the response is simulating a step to the existing transfer function. With Matlab/Simulink, a step input of $\Delta p = 2kg/cm^2$ was given to the transfer function 8 of the joint. The system response is shown in Figure 9.

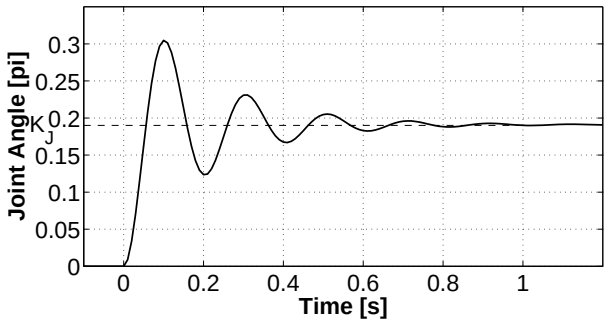


Fig. 9. Simulated step response of the joint transfer function

Figure 10 shows the parameters latency and adjustment time, which have been identified as $T_L = 0.015$ s and $T_A = 0.045$ s.

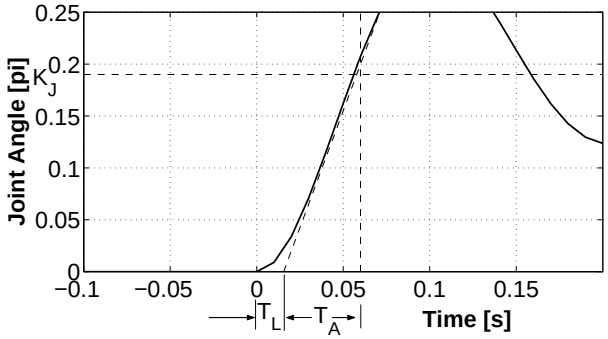


Fig. 10. Description

According to [14], the controllability of a system is getting more and more difficult, the higher T_L is, because the controller cannot interact and the system

is totally committed to an occurring disturbance, and the lower T_A is, what means that the system responds fast after the latency is over and the controller has to act also fast and adequate.

The relation T_L/T_A is an indicator for the controllability of a system. In our case, a relation of $T_L/T_A = 0.36$ is controllable, but needs a high control effort [14]. This could be an indicator that we have to fall back on higher control architectures. Easy controllable systems have a T_L/T_A ratio of 0-0.1, these with over 0.8 are hardly controllable. The parameters T_L and T_A can further be used to tune control parameters.

4 Controller Design

To design a suitable controller, the demands have to be defined first.

The controller is designed for living up to the following conditions:

1. Fast and exact following of changing command values (input of a new reference position)
2. Robust behavior on parameter fluctuations (lifting of a load resp. movement of load)
3. Low overshooting ($< 10\%$)

To analyze if the controller fits the needs described above, the following experiments were attended:

a. Response of the joint on a step input (Demand 1.)
b. Testing the ability to follow an oscillating reference value by giving a sinus curve to the controller input (Demand 1.)
c. Study the behavior on parameter fluctuations by giving a step input with different joint parameters, represented by attached load (Demand 2.)

Demand 3. is important for all types of experiments, but especially for experiment b) , because slow changing inputs will be the main case in the arm control.

As a first architecture, a standard PID controller was implemented on the testbed. In most control applications, a regular PID controller as shown in Figure 11 can solve occurring tasks satisfying, but the success of a PID controller is doubtful, according to the results in 3.3.

To reduce the error for the position control loop, caused by the nonlinearity and the hysteresis of the artificial muscles, the idea to implement a subsidiary torque control loop came up. This inner loop changes the system behavior that it appears to the outer control loop like a linear element. Controlling the system on the level of torques has several advantages. Dynamic models of arm parts can be added if necessary for all joints or just to consider gravity and masses of inertia in the shoulder and upper arm joint. The torque control loop can be run separately, for a passive arm-movement (e.g. shaking hands). Testing of the

inner control loop is possible independent from the rest of the control and allows exact parameter tuning.

The second architecture is a cascaded PI-PI controller, with separate control loops for position and torque. Figure 15 shows a possible implementation with a inner PI torque controller and a outer PI position controller. Torque or force sensors are required to realize the feedback control.

To realize the architecture in Figure 15 without using torque or force sensors in the joint, an actuator model (Equation 3.1) is used as feedback to predict the actual joint torque (Figure 16). Accurate modeling of the actuators is necessary to receive similar control results as above.

4.1 Implementing a regular PID controller

Figure 11 shows the implementation of a regular PID controller, where the system consists of the pressure control unit, the actuator itself and the arm. The controller output sets directly Δp in the pressure control unit. The only sensor information are encoder steps, from which is also the joint velocity derived.

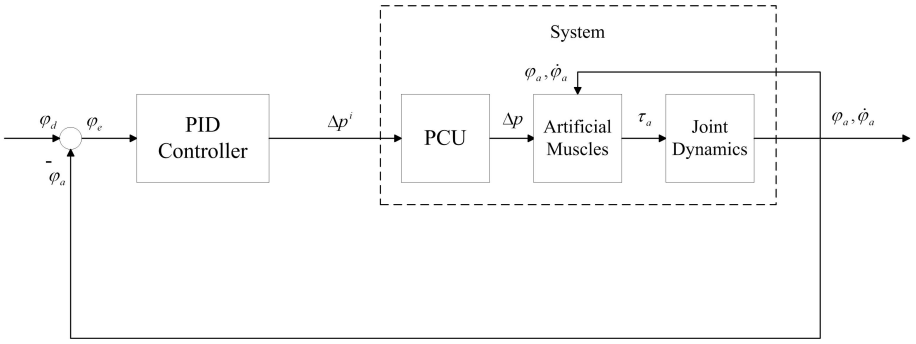


Fig. 11. Implementation of a regular PID controller

The transfer function for a PID controller is given by:

$$F_{PID}(s) = K_P + \frac{K_I}{s} + K_D s \tag{9}$$

The experiments were run under QNX with a sample rate of 2 kHz. To tune the PID parameters, two different methods, Ziegler-Nichols method (ZN) and Chien, Hrones and Reswick method (CHR) were applied.

The Ziegler-Nichols is a common method to tune PID parameters in process engineering where usually much higher time constants exist, but also gave satisfying results for our system. After increasing the proportional gain until a undamped oscillation occurred, the parameters were determined as $K_{P,crit} = 1.25$ and $T_{crit} = 0.175s$.

According to the Ziegler-Nichols approach, the control parameters have to be set as $K_P = 0.6 \cdot K_{p,crit} = 0,75$, $K_I = \frac{K_p}{0.5 T_{crit}} = 14.3$ and $K_D = 0.12 \cdot K_P \cdot T_{crit} = 0.016$. Due to instable behavior of the joint (because of the actuator input limit of $\pm 2 kg/cm^2$), the gain had to be reduced to $K_P = 0.4$ to guarantee stable movement. Therefore, other control parameters changed to $K_I = 7$ and $K_D = 0.08$.

The Chien, Rhones and Reswick method is based on Ziegler-Nichols, but improved on basis of simulation results. Control parameter tuning can be done according to certain criteria, depending if the control wants to be optimized for fast following of reference values, or for fast control of occurring disturbances. Experiments with both criteria showed that designing the controller for fast following of reference values gives better results to satisfy the controller demands. The tuning of this method is based on system parameters latency T_L and adjustment time T_A. These parameters are determined as described in section 3.3. The control parameters can be calculated as $K_P = \frac{0.6 T_A}{K_J T_L} = 6$, $K_I = \frac{K_P}{T_A} = 120$ and $K_D = 0.5 \cdot K_P \cdot T_L = 0.04$. The same instability problem like in the last paragraph required tuning down the proportional gain to $K_P = 0.2$, which decreased the other parameters to $K_I = 4.0$ and $K_D = 0.0014$.

In the first experiment (Figure 12), three step inputs were given to the controller. The Chien, Hrones and Reswick method showed a smoother, but slower reaching of the reference value. Fine-tuning of the Ziegler-Nichols parameters did not allow to get rid of the small spike.

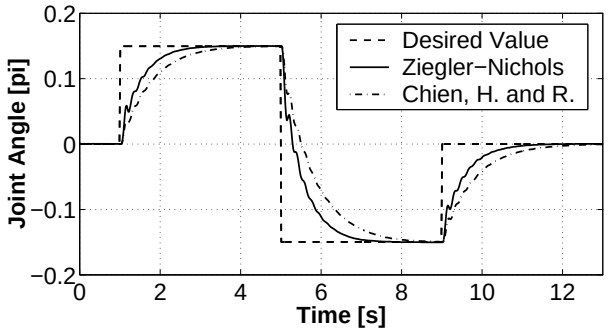


Fig. 12. Step response of the joint, using regular PID controller

The next experiment shows the ability to follow a slow changing reference value, represented by a sin curve with f(t)=0.15 sin(2t). Both parameter sets could not follow this trajectory satisfying, the difference was up to 18° between reference position and joint position (CHR tuning).

Figure 14 shows the step response of the joint with different parameters. In this case, a mass of 0.8 kg was added to the arm to change the mass of inertia from J=0.0085 kgm^2 to approximately J=0.036 kgm^2. The joint response

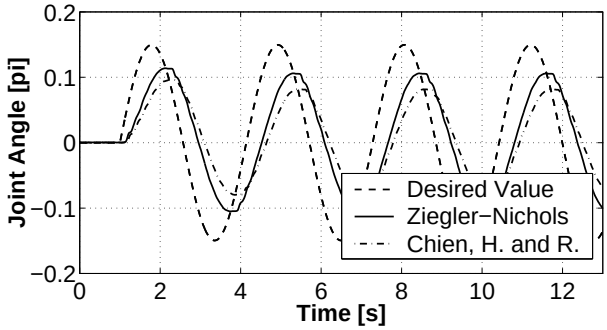


Fig. 13. Response of PID controlled joint on oscillating controller input

was oscillating after a step input was given as new reference value. These oscillations could be minimized again by decreasing the proportional gain, but the adjustment time stood in no relation to the smoother curve.

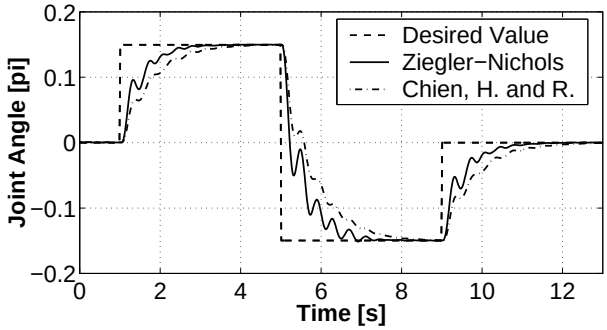


Fig. 14. Step response of PID controlled joint after changing mass of inertia

4.2 Implementation of cascaded controller (PI-PI)

A possible implementation of a cascaded controller with different control loops for torque and position is shown in Figure 15. The idea is to give the inner control loop a fast time response, so that the behavior is similar to a linear transfer element for the outer control loop. The sample times of inner loop to outer loop are chosen as 1/10 to provide the required time for inner loop to adjust the reference value.

The torque controller is implemented as PI control, so that also disturbances of the inner loop are controlled in order that the loop can be run in stand-alone

torque-control mode. Even the outer controller has no D-part, because a fast part of the control is not necessary with the fast inner loop. The number of control parameters is also kept low because the tuning process would become more difficult.

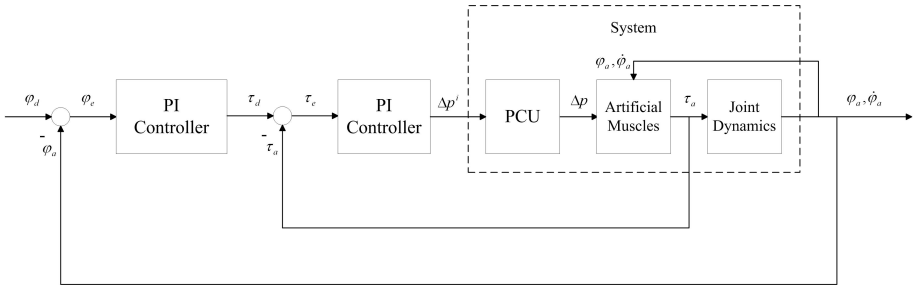


Fig. 15. PID controller with subsidiary torque control loop

The transfer functions of torque and position controllers are:

$$F_{C\tau} = \frac{1}{K_{P\tau} + \frac{K_{I\tau}}{s}} \tag{10}$$

$$F_{Cp} = \frac{1}{K_{Pp} + \frac{K_{Ip}}{s}} \tag{11}$$

To minimize cost and hardware effort, no force or torque sensors are used in the ISAC arm. But to realize a closed-loop control, torque feedback is required. This means the architecture in Figure 15 cannot be implemented in this way. The next section introduces another possibility to get a torque feedback.

4.3 Implementation of model-based cascaded controller (PI-MBPI)

One approach to realize the cascaded control architecture without measuring torque is shown in Figure 16. The information about torque is given by our actuator model, described in section 3.1. The fast torque control loop runs at a sample rate of 2 kHz, the slower position control loop with a rate of 0.2 kHz.

First step was the implementation of the torque control and inner loop PI parameter tuning. For this purpose, the joint was fixed in its position so that no movement was allowed. The parameters were adjusted as $K_{P,T} = 0.1$ and $K_{I,T} = 10$ by Ziegler-Nichols, which led immediately to a good result. After tuning the parameters, the torque controller was tested separately. Figure 17 shows a response of the torque controller for different step inputs.

The inner control loop has now a behavior like a first order lag time element with a time constant $T_T = 0.025$. An important fact is that the adjustment time

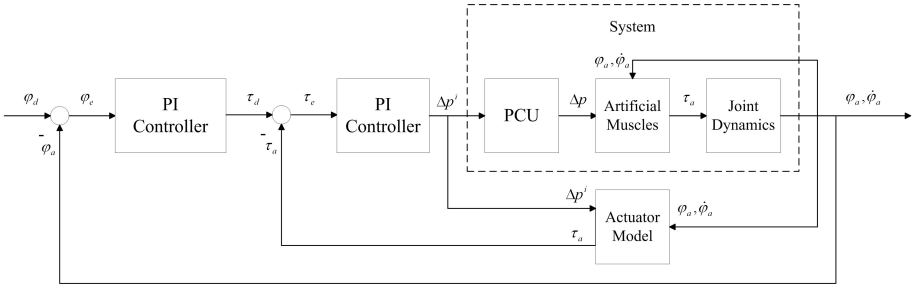


Fig. 16. Model-based controller

hardly depends on the amount of the reference torque. The torque was reached after 0.018 s to 0.02 s, little differences are caused by different filling times but do not need to be considered.

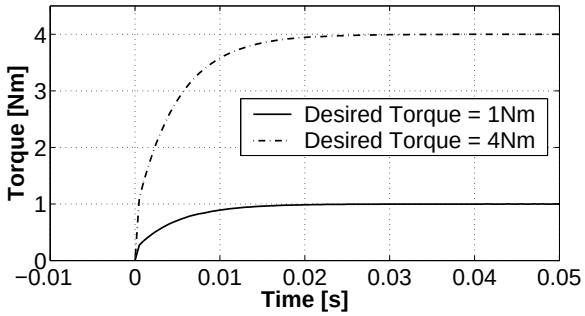


Fig. 17. Step response of the inner torque control loop

None of the common tuning methods seemed to work for the position control loop, this might have to do with the cascaded control structure and the limitations of torque and pressure in the controller outputs. Therefore, outer loop parameters were adjusted by trial and error as $K_{P,P} = 4.0$ and $K_{I,P} = 8.0$.

The joint response on a step input (Experiment a.), Figure 18) was now faster and smoother than with the regular PID controller. But much more important is the response on the sinus curve (Experiment b.)), given in Figure 19. The joint position follows the reference value nearly without lag. Due to the fact that the joint has to follow a trajectory, this is the biggest improvement compared to the PID controller.

A change of parameters (Experiment c.), Figure 20) produces overshoot. This overshoot can be accepted because the controller input will never be a step of this size. Experiments with higher mass of inertia and the sinus trajectory

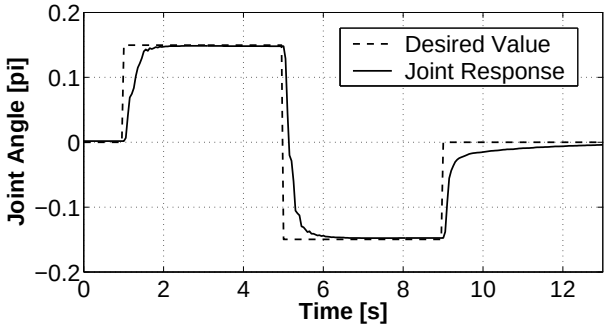


Fig. 18. Step response of the joint, using model-based PI-PI controller

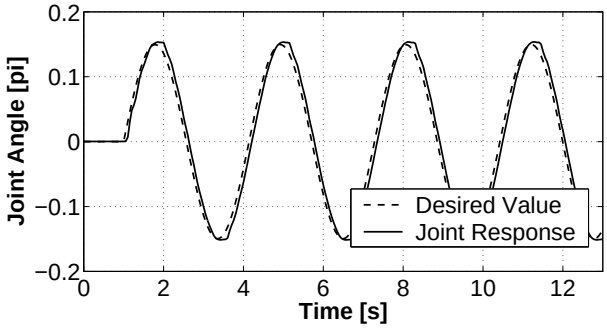


Fig. 19. Response of PI-MBPI controlled joint on oscillating controller input

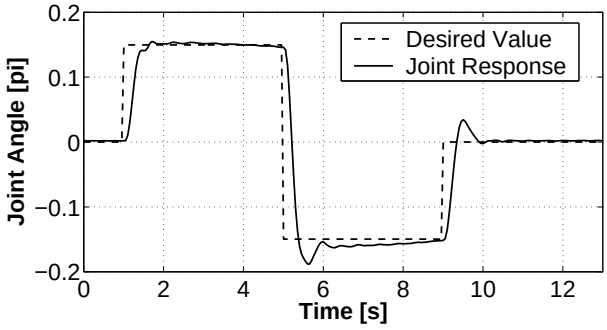


Fig. 20. Step response of PI-MBPI controlled joint after changing mass of inertia

in Experiment c.) showed that the overshot does not play any role when the controller input is a sequence of very small steps.

Stability of PI-MBPI controller

To prove stability of the control, the closed-loop transfer function of the system has to be derived. Instead of using the transfer function of the model-based controller, the easier architecture in Figure 15 is used. Provided that the actuator model is close to the real behavior, this assumption is allowed. The damping parts of the system have been ignored, that means a simplified version of the joint transfer function is used for proof of stability. This is possible because the damping part would take energy out of the system.

Pressure Control Unit	$F_C(s) = \frac{1}{1+T_C s}$
Torque Controller	$F_{C\tau}(s) = \frac{1}{K_{P\tau}+\frac{K_{I\tau}}{s}}$
Position Controller	$F_{Cp}(s) = \frac{1}{K_{Pp}+\frac{K_{Ip}}{s}}$
Joint (Actuator and Dynamics)	$F_J(s) = \frac{Br}{Js^2-Ar^2}$
Torque Feedback	$\tau_a = Ar^2\varphi_a + Br\Delta p^i$

Table 1. Transfer functions of different components and actuator equation

Because all transfer functions are linear, which means that our overall transfer function will be a linear equation, the proof of stability can be done with the Hurwitz criterion [14]. The idea is to locate the poles of the linear closed-loop function in the s-plane. All poles have to lie in the left half of the $s - j\omega$ diagram to prove stability. Therefore, coefficients of the characteristic equation have to be analyzed.

The closed loop transfer function is:

$$\frac{\varphi_a}{\varphi_d} = \frac{F_0(s)}{1 + F_0(s)} \tag{12}$$

To determine poles of the system, the denominator has to be zero:

$$1 + F_0(s) = 0 \tag{13}$$

The open loop transfer function $F_0(s)$ is derived from the individual transfer functions as:

$$F_0(s) = \frac{Br(K_{P\tau} + \frac{K_{I\tau}}{s})(K_{Pp} + \frac{K_{I\tau}}{s})}{((Js^2 - Ar^2)(1 + T_C s)) + Js^2 Br(K_{P\tau} + \frac{K_{I\tau}}{s})} \tag{14}$$

Inserted in Equation 13 leads to the characteristic formula:

$$\begin{aligned}(JT_C)s^5 + (J + JK_{P\tau}Br)s^4 + (JBrK_{I\tau} - Ar^2T_C)s^3 + \\ (BrK_{Pp}K_{P\tau} - Ar^2)s^2 + (BrK_{Pp}K_{I\tau} + K_{Ip}K_{Pp})s + \\ (BrK_{Ip}K_{I\tau}) = 0\end{aligned} \tag{15}$$

Comparison with the regular equation

$$a_0 s^5 + a_1 s^4 + a_2 s^3 + a_3 s^4 + a_4 s^5 + a_5 = 0 \tag{16}$$

gives the coefficients $a_0 - a_5$. The first condition of the Hurwitz criterion says that all coefficients have to be positive. Considering the fact that the parameter A is always negative, and all others are positive, this condition is fulfilled. The second condition demands that all determinants $H_2..H_5$ are positive. H_i are $[i, i]$ sub-determinants derived from the Hurwitz determinant starting at the upper left corner. The Hurwitz determinant for our system is the following:

$$H = \begin{vmatrix} a_1 & a_3 & a_5 & 0 & 0 \\ a_0 & a_2 & a_4 & 0 & 0 \\ 0 & a_1 & a_3 & a_5 & 0 \\ 0 & a_0 & a_2 & a_4 & 0 \\ 0 & 0 & a_1 & a_3 & a_5 \end{vmatrix} = \begin{vmatrix} 0.011 & 9.5 & 200 & 0 & 0 \\ 0.0015 & 1.7 & 100 & 0 & 0 \\ 0 & 0.011 & 9.5 & 200 & 0 \\ 0 & 0.015 & 1.66 & 100 & 0 \\ 0 & 0 & 0.011 & 9.5 & 200 \end{vmatrix}$$

Computing the determinants H_2-H_5 gives as result: $H_2 = 0.0040$, $H_3 = 0.030$, $H_4 = 5,4$ and $H_5 = 1100$, which means that all poles of the system are lying on the left side of the imaginary axis. The system is hereby always stable. Stability is also given for different values of J. Changing the mass of inertia to J=0.036 kgm^2 even stabilizes the system more by increasing the determinants.

5 Future Work

The next step is the implementation of the PI-MBPI control on the ISAC arm. To reduce control errors, the outer loop PI controller will be extended with nonlinear terms to consider gravitation and masses of inertia in some of the joints. This will be necessary in any case for the shoulder joint because of the long lever arm.

Once the control is implemented on the arm, the algorithm will also be used together with the actuator model and a simplified dynamic arm model in a

simulator. This simulator allows the imaginary use of the arm and can be used to test higher level components to verify the controller input, or as a part in a cognitive robot for learning or motion improvements without using the real arm.

The arm could be equipped with force or torque sensors supplementary. Then an exact actuator modeling would become redundant and the control would be improved, especially relating to the interference immunity.

6 Conclusions

On the basis of the received results, the cascaded model-based controller (PI-MBPI) is preferred instead of using a regular PID control. In particular the fast following of reference values and the handling of parameter changes are the deciding features. Practical reasons, like easy extension upon dynamic models and the ability to run the arm in torque-control-mode, support this choice.

It was shown that the realization of a torque control loop is possible without using any torque or force sensors. The feedback can be given by an actuator model, which reduces the required hardware and with it costs.

Acknowledgement

This research is conducted with the financial support of the Center for Intelligent Systems (CIS) and the Institute for Industrial Applications of Informatics and Microsystems (IAIM). The research is subject of Mr. Schroeder's Diploma Thesis at the University of Karlsruhe (TH), written as a visiting scholar at Vanderbilt University.

References

1. K. Kawamura, R. A. Peters II, S. Bagchi, M. Iskarous and M. Bishay. Intelligent robotic systems in service of the disabled. *IEEE Transactions on Rehabilitation Engineering*, 3(1) , pp. 14–21, March 1995.
2. N. Sarkar S. Thongchai, M. Goldfarb and K. Kawamura. A frequency modeling method of Rubbertuators for control application in an IMA framework. *Proceedings of American Control Conference, Arlington, Virginia*, pp. 1710–1714, June 2001.
3. A. Kara, K. Kawamura et al. Reflex control of a robotic aid system for the physically disabled. *IEEE Control Systems Magazine*, 12(3), pp. 71–77, June 1992.
4. S. Northrup, N. Sarkar and K. Kawamura. Biologically Inspired Control Architecture for a Humanoid Robot. *IROS 2001*, Maui, Hawaii, 2001.
5. F. Groen, P. van der Smagt and K. Schulten. Analysis and control of a Rubbertuator arm. *Biological Cybernetics*, Springer-Verlag, 75, pp. 433–440, 1996.
6. R.T. Pack, D.M. Wilkes and K. Kawamura. A Software Architecture for Integrated Service Robot Development. *IEEE Conference on Systems, Man, and Cybernetics*, Orlando, pp. 3774-3779, September, 1997.

7. Steve Northrup. Biologically-Inspired Control of a Humanoid Robot. *Dissertation*, Graduate School of Vanderbilt University, Nashville, Tennessee, August 2001.
8. D.G. Caldwell, A. Razak, and M.J. Goodwin. Braided Pneumatic Muscle Actuators. *IFAC Conference on Int. Autonomous Vehicles*, Southampton, UK, April 1993.
9. Darwin G. Caldwell, Gustavo A. Medrano-Cerda, and Mike Goodwin. Control of pneumatic muscle actuators. *IEEE Control Systems*, pp. 40–47, February 1995.
10. P. van der Smagt, K. Schulten, T. Hesselroth and K. Sarkar. Neural network control of a pneumatic robot arm. *IEEE Trans. Syst. Man., Cybernetics*, 24, pp. 28–38, 1994.
11. Futek Advanced Sensor Technology. www.futek.com/
12. Steve Northrup. A PC-based controller for the Soft Arm robot. *Master's thesis*, Vanderbilt University, Center for Intelligent Systems, Nashville, TN, USA, March 1997.
13. Joachim Schröder. Dynamic Pneumatic Actuator Model for a Model-Based Torque Controller. *Submitted Paper CIRA Conference*, July 2003.
14. L. Merz, H. Jaschek. Grundkurs der Regelungstechnik. Oldenbourg Verlag, 1996.
15. Ching-Ping Chou and Blake Hannaford. Static and dynamic characteristics of McKibben pneumatic artificial muscles. *International Conference on Robotics and Automation*, 1, pp. 281–286, 1994.
16. Ching-Ping Chou and Blake Hannaford. Measurement and modeling of McKibben pneumatic artificial muscles. *IEEE Transactions on Robotics and Automation*, 12(1), pp. 90–102, February 1996.
17. H.F. Schulte Jr. The characteristics of the McKibben artificial muscle. *The Application of External Power in Prosthetics and Orthotics*, National Acamy of Sciences - National Research Council, Washington D.C., 1961.
18. Bridgestone Company. Rubbertuators and Applications for Robotics. *Technical Guide*, No.1, 1986.
19. Shadow Robot Company. www.shadow.org.uk/
20. Research Center for Information Technologies. University of Karlsruhe (TH). wwwneu.fzi.de/ids/eng/
21. J.J. Craig. Introduction to Robotics. Addison-Wesley, 1998.
22. Krishna C. Gupta. Mechanics and Control of Robots. Springer, New York, 1997.
23. J. M. Skowronski. Control Dynamics of Robotic Manipulators. *Academic Press*, Orlando, Florida, 1986.
24. R. Pack, M. Iskarous et al. Comparison of fuzzy and nonlinear control techniques for a flexible rubbertuator-based robot. *Proceedings of the Third International Fuzzy Systems and Intelligent Control Conference*, Louisville, KY, USA, pp. 361-370, March 1994.