

# Nonlinear adaptive robust motion control for hydraulic winch in oil and gas wireline operation

Fanping Bu

Schlumberger Industrial Internet Center, Schlumberger, Sugar Land, TX 77478  
USA (Tel: 281-285-8135; e-mail: fbu2@slb.com).

**Abstract:** This paper presents the design and implementation of a nonlinear model based adaptive robust controller (ARC) for tool motion control driven by a hydrostatic transmission used in an Oil and Gas wireline operation. A detailed physical system model was built for controller design and testing. ARC controller was designed to address both parametric uncertainties and uncertain nonlinearities inherent in the nonlinear system dynamics. The controller software development and testing followed a Model-Based Design (MBD) procedure. A micro-service architecture based on docker containers was adopted for the controller software which facilitated continuous integration and deployment. The preliminary testing results show the effectiveness of the ARC controller design.

**Keywords:** Wireline operation in Oil & Gas, Hydrostatic transmission, Adaptive control, Nonlinear robust control, Model-Based Design, Docker container

## 1. INTRODUCTION

Wireline logging is widely used in the Oil and Gas industry to measure the properties of a formation using electronic instruments. Fig. 1 illustrates a typical wireline logging operation. A drum of electric cable is driven by a hydraulic winch with a tool string packed with different formation measurement sensors attached at the free end of the cable. The winch drum is rotated by a hydraulic motor that moves the tool string up and down along well bore. Sensors packaged inside the tool string conduct sensing measurements while moving and send back measurement results through the connected cable. During operation, an operator is required to control the hydraulic winch manually so that the tool string movement will follow a desired motion profile. As the first step toward automation of the wireline logging operation, it is necessary to control the hydraulic winch, or tool string motion, following a desired motion profile, automatically, without operator intervention.

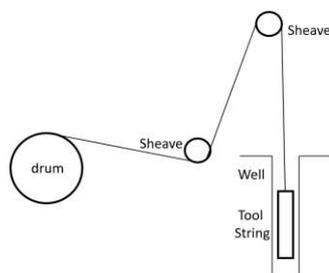


Figure 1 Wireline logging operation for oil and gas industry

The hydraulic winch drives the drum via a gear transmission. As shown in Fig. 2, the hydraulic winch is a hydrostatic transmission system consisting of a variable displacement pump, a variable displacement motor and a charge pump. The pumps are driven by a vehicle engine through gears. The drum is driven by variable displacement motor through transmission

gears. Hydrostatic transmission (Manning and Luecke 1998) has several advantages such as a high power-density, “continuously-variable transmissions” and the capability of delivering high torque at low speed. Therefore, it is widely used in applications like mining, construction and farming equipment. As pointed out in (Sun and Aschemann 2013), hydrostatic transmission systems are subject to different nonlinearities and uncertainties in industrial applications. Aside from typical nonlinearities and uncertainties seen in a hydrostatic transmission application, large forces in the oil & gas operation, and changing effective drum radius and mass/inertia due to reeling in or releasing cable will further complicate the controller design. Thus, a nonlinear model-based controller design is essential to deliver high performance of a closed loop controller for the system shown in Figs. 1 and 2. Nonlinear model-based controller design such as sliding mode control and flatness based control (Aschemann and Sun 2013, Sun and Aschemann 2013, Sun and Aschemann 2016) has been successfully applied to hydrostatic transmission control.

Adaptive robust control (ARC) (Yao and Tomizuka 1997) was developed to address control of nonlinear system with both parametric and nonlinear uncertainties. It has been applied to many different application areas such as linear motors (Xu and Yao 2001), electro-hydraulic system (Yao, Bu et al. 2000), vehicle control (Bu and Tan 2007) and drilling control (Bu and Dykstra 2014). It utilizes fast nonlinear robust feedback to attenuate overall system uncertainties for a guaranteed baseline performance. Parameter adaptation is applied to further improve system performance by reducing parametric uncertainties. Backstepping via (Krstic, Kanellakopoulos et al. 1995) Lyapunov functions is used to design the control law systematically.

This paper will be organized as follows: In Section 2, system dynamics modelling, and system identification of hydraulic

winch, drum and tool string motion will be presented. In Section 3, first the control problem will be formulated for the control of tool string motion. Second the ARC controller design, which includes feedforward compensation, feedback controller design and parameter adaptation law design, will be presented. Section 4 will present controller implementation and testing using model-based design with docker containers. Experimental results will be presented in Section 5. Section 6 will conclude the paper.

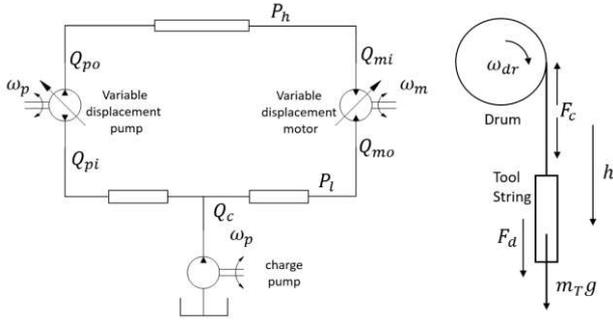


Figure 2 Schematics of a hydraulic winch

## 2. SYSTEM MODELING

A system dynamics modelling process provides us physical intuitions on the controller design. Fig. 2 shows the overall system schematics which can be separated into two parts: winch hydraulics and winch mechanics.

### 2.1 Winch hydraulics

Winch hydraulics include a variable displacement axial piston pump driven by engine through fixed gears. The hydraulic fluid flow going in and out of the pump is modelled by the following equations:

$$Q_{pi} = D_p \omega_p \quad D_p = G_p \alpha \quad \alpha = f(v) \quad Q_{po} = \eta_{vp} Q_{pi} \quad (1)$$

where  $Q_{pi}$  is the input flow to the pump,  $D_p$  is the displacement of the pump and  $\omega_p$  is the pump shaft rotation speed. The pump's displacement can be changed with an electronic command  $v \in [-1,1]$ . An electro-hydraulic actuator with displacement feedback control is used to tilt the cradle swashplate such that the normalized pump displacement  $\alpha \in [-1,1]$  will always follow the command  $v$ .  $G_p$  is a positive flow gain. Extensive testing reveals that the actuator dynamics is fast enough for the wireline application. Therefore, the actuator dynamics is ignored in the modelling and the following controller design. A nonlinear mapping function  $f$  is used to characterize the relationship between command  $v$  and actual displacement  $\alpha$  including deadband and nonlinear gain.  $Q_{po}$  represents the pump output flow rate and  $\eta_{vp}$  is the pump volumetric efficiency.

The flow through the hydraulic motor is modelled by the following equations:  $Q_{mo} = D_m \omega_m$  and  $Q_{mi} = Q_{mo} / \eta_{vm}$ , where  $Q_{mo}$  is the output flow of the motor and  $\omega_m$  is the motor speed. Although a variable displacement motor is used, the displacement has only two fixed values. The motor displacement  $D_m$  is modelled as a positive constant.  $Q_{mi}$  is the input flow rate of the motor and  $\eta_{vm}$  is the motor volumetric efficiency. The torque generated by the motor is described by

$$T_m = D_m (P_h - P_l) \eta_{tm} \quad (2)$$

where  $T_m$  is the motor torque,  $P_h$  and  $P_l$  represent the pressures on the high and low sides, and  $\eta_{tm}$  is the motor mechanical efficiency. The output flow of the charge pump is represented by  $Q_c = D_c \omega_p$ , where  $D_c$  is the displacement of the charge pump. Both charge pump and main pump are driven by the same shaft from the engine. The pressure dynamics is represented by the following equations:

$$V_h \dot{P}_h / \beta_e = Q_{po} - Q_{mi} - Q_{lh} \quad (3)$$

$$V_l \dot{P}_l / \beta_e = Q_{mo} + Q_c - Q_{pi} - Q_{ll}$$

where  $V_h$  and  $V_l$  represent the total fluid volumes on the high-pressure and low-pressure side.  $Q_{lh}$  and  $Q_{ll}$  represent leakage flow on the high-pressure and low-pressure side.  $\beta_e$  is the effective bulk modulus of the hydraulic fluid.

### 2.2 Winch mechanics

The output shaft of the hydraulic motor drives the drum through the gear transmission. The rotating speed  $\omega_{dr}$  and the torque  $T_{dr}$  acting on the drum is given by  $\omega_{dr} = \omega_m / n$  and  $T_{dr} = n T_m$ , where  $n$  is the transmission gear ratio. The drum motion is governed by:

$$(J_{dr} + J_c) \ddot{\omega}_{dr} = T_{dr} - F_c r \quad (4)$$

where  $J_{dr}$  represents drum inertia.  $J_c$  represents inertia due to the cables on the drum.  $r$  is the effective drum radius. Both  $J_c$  and  $r$  can change during operation when cable is reeled in or released from the drum, especially for deep wells.  $F_c$  is the cable force. The tool movement is governed by:

$$(\rho h + m_T) \ddot{h} = (\rho h + m_T) g - F_c + F_d \quad (5)$$

where  $\rho$  is the cable's line mass density,  $m_T$  is the tool mass,  $h$  is the tool displacement underground in the wellbore and  $F_d$  is the disturbance force acting on the tool and cable during operation including friction, buoyancy and other forces. If the cable is long enough, force and displacement will propagate according to a wave equation. As the first step toward the controller design, a cable is modelled as a rigid body and its dynamics is ignored. The relationship between drum speed and tool speed is given by  $\dot{h} = \omega_{dr} r$ . To simplify the controller development, it is reasonable to assume symmetry in the hydraulic loop which means  $V_T = V_h = V_l$ . The effective radius  $r$  changes slowly compared with system dynamics. It is assumed  $\dot{r} = 0$ . Defining the load pressure  $P_L = P_h - P_l$  and the state vector  $x = [x_1, x_2, x_3]^T = [h, \dot{h}, P_L]^T$ , the system dynamics equation (1)-(10) can be written as

$$\dot{x}_1 = x_2 \quad m_s \dot{x}_2 = A_m x_3 + (\rho x_1 + m_T) g + F_d \quad (6)$$

$$\dot{x}_3 = \frac{\beta_e}{V_T} [C_p \omega_p \alpha - C_m x_2 - D_c \omega_p + Q_L]$$

where  $Q_L = Q_{ll} - Q_{lh}$  represents the leakage flow, effective mass  $m_s = (J_d + J_c) / r^2 + \rho x_1 + m_T$ ,  $A_m = n D_m \eta_{tm} / r$ ,  $C_p = (1 + \eta_{vp}) G_p$ ,  $C_m = (1 + 1 / \eta_{tm}) D_m n / r$  and  $\alpha = f(v)$  with  $v$  as the actual control input command.

### 2.3 Model identification and validation

In the current system setup, the engine speed which is related to the pump shaft speed,  $\omega_p$ , by a transmission ratio, drum speed  $\omega_{dr}$ , pressures on the high side  $P_h$  and low side  $P_l$ , and tool displacement  $h$  and speed  $\dot{h}$  are measured. The effective drum radius  $r$  can be estimated from equation  $\dot{h} = \omega_{dr} r$  using a recursive least squares method.  $J_c$ , the inertia due to the cable on the drum, can be calculated using the estimated effective drum radius  $r$ . Extensive experiments were conducted to identify the system dynamics. As shown in Fig. 3, an open loop ramp test was conducted where the control input command  $v$  was ramped from 0 to +/-100 percent. The response of the drum rotating speed  $\omega_{dr}$  shows significant deadband and nonlinearities. Since the actuator dynamics is ignored, the nonlinear mapping  $f$  is modelled as a deadband and nonlinear gain function fitted with a polynomial. A system model was built using Matlab/Simulink/SimScape for control design, model-in-loop and software-in-loop testing. Fig. 5 shows a comparison between simulation and testing data.

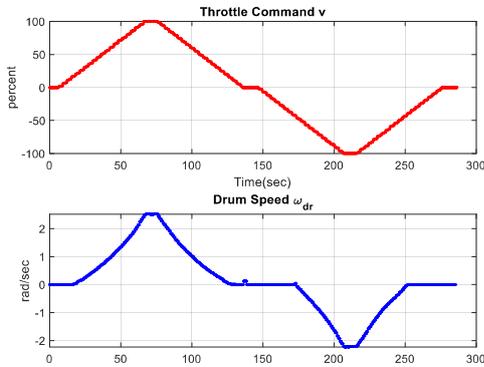


Figure 3 Open loop ramp testing

### 3. ARC CONTROLLER DESIGN

In this section, the control problem will be formulated first. The controller design difficulties will be illustrated with addressing strategies. Detailed controller design will be presented.

#### 3.1 Design model and issues to be addressed

In general, the system (6) has significant nonlinearities represented by a nonlinear mapping function  $f$  in Fig. 4, changing effective drum radius  $r$  entering system dynamics in different places and changing effective mass  $m_s$  due to reeling in or releasing cable. The system is also subjected to large uncertainties represented by a large disturbance force  $F_d$  when in operation and leakage flow  $Q_L$ . Both  $F_d$  and  $Q_L$  can also be treated as the modelling errors in the torque/force equations and hydraulic loop. To better compensate for their effects, both  $F_d$  and  $Q_L$  can be separated into two parts: a slow varying part which can be captured and compensated by parameter adaptation and a fast-changing part which needs to be attenuated by robust feedback. To utilize parameter adaptation, the system dynamics' equations need to be linearly parametrized by unknown parameters. Defining an unknown parameter set as  $\theta = [\theta_1, \theta_2]^T = [F_{dn}/m_s, Q_{Ln}\beta_e/V_T]^T$ , where  $F_{dn}$  and  $Q_{Ln}$  are nominal part (or low frequency part) of  $F_d$  and  $Q_L$ , the system equation (6) can be written as

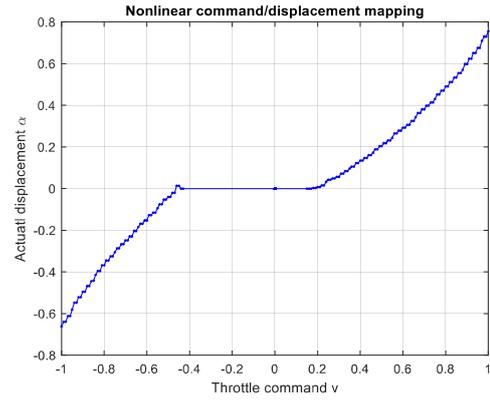


Figure 4 Nonlinear mapping function  $f$

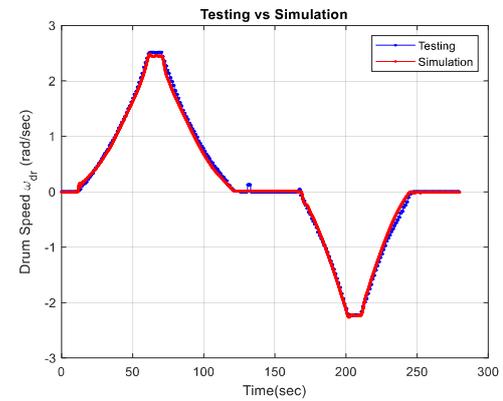


Figure 5 Simulation vs testing data

$$\dot{x}_1 = x_2 \quad (7)$$

$$\dot{x}_2 = A_m x_3 / m_s + (\rho x_1 + m_T) g / m_s + \theta_1 + \tilde{F}_d$$

$$\dot{x}_3 = \beta_e (C_p \omega_p \alpha - C_m x_2 - D_c \omega_p) / V_T + \theta_2 + \tilde{Q}_L$$

where  $\tilde{F}_d = (F_d - F_{dn})/m_s$  and  $\tilde{Q}_L = \beta_e (Q_L - Q_{Ln})/V_T$  represent uncertain nonlinearities, and the real control input, electronic pump displacement command  $v$ , is related to the actual pump displacement  $\alpha$  with nonlinear function  $\alpha = f(v)$ . The control problem is formulated as: given a desired motion trajectory  $x_{1d}$ , design a control law  $u$  for system control input  $v$  such that the tool motion output  $x_1$  will track desired trajectory  $x_{1d}$  as close as possible.

Since the extent of parametric uncertainties and uncertain nonlinearities are known, the following practical assumptions are made:

$$\theta \in \Omega_\theta \triangleq \{\theta_{min} < \theta < \theta_{max}\} \quad (8)$$

$$|\tilde{F}_d(t, x_1, x_2)| \leq \delta_F(t, x_1, x_2) \quad |\tilde{Q}_L(t, x)| \leq \delta_Q(t, x)$$

At this stage, the major difficulties in designing controller for system (7) are: 1) the system is nonlinear as represented by nonlinear actuator mapping, changing effective drum radius and effective mass; 2) the system has large parametric uncertainties including large load changes  $F_{dn}$  and hydraulic modelling error  $Q_{Ln}$ ; 3) the model uncertainties are mismatched, i.e. both parameter uncertainties and uncertain

nonlinearities appear in the equation which is not directly related to the control input  $u = v$ . To address these challenges, the following strategies are adopted: 1) physical model based nonlinear analysis and synthesis will be employed to address nonlinearities in the system dynamics; 2) ARC will be used to handle both parametric uncertainties and uncertain nonlinearities; 3) backstepping design via ARC Lyapunov function will be used to overcome the design difficulties introduced by unmatched model uncertainties.

### 3.2 Notations

The following notations will be used throughout the control design process. Let  $\hat{\theta}$  denote the estimation of  $\theta$  and  $\tilde{\theta}$  the estimation error ( $\tilde{\theta} = \hat{\theta} - \theta$ ). From (8), a simple discontinuous projection can be defined as (Sastry and Bodson 1989):

$$Proj_{\theta_j}(\hat{\theta}_j) = \begin{cases} 0 & \text{if } \hat{\theta}_j = \theta_{jmax} \text{ and } \dot{\hat{\theta}}_j > 0 \\ 0 & \text{if } \hat{\theta}_j = \theta_{jmin} \text{ and } \dot{\hat{\theta}}_j < 0 \\ \hat{\theta}_j & \text{otherwise} \end{cases} \quad (9)$$

By using an adaptation law given by:

$$\dot{\hat{\theta}} = Proj_{\hat{\theta}}(\Gamma\tau) \quad (10)$$

where  $\Gamma > 0$  is a diagonal adaptive gain matrix,  $\tau$  is the adaptation function to be synthesized later, it can be shown that (Yao and Tomizuka 1994):

$$\begin{aligned} \hat{\theta} &\in \Omega_{\theta} \\ \tilde{\theta}^T(\Gamma^{-1}Proj_{\hat{\theta}}(\Gamma\tau) - \tau) &\leq 0 \end{aligned} \quad (11)$$

### 3.3 ARC controller design

*Step 1:* At this step, a desired load pressure  $\alpha_2$  is designed for the system load pressure  $x_3$  such that the tool motion  $x_1$  will follow the desired motion trajectory  $x_{1d}$ . Define an output tracking error  $z_1 = x_1 - x_{1d}$  and a virtual speed command  $x_{2eq} = \dot{x}_{1d} - k_1 z_1$ , the speed tracking error can be written as  $z_2 = x_2 - x_{2eq}$ . Its derivative is given by:

$$\dot{z}_2 = \frac{A_m}{m_s} x_3 + \frac{\rho x_1 + m_T}{m_s} g + \theta_1 + \tilde{F}_d - \dot{x}_{2eq} \quad (12)$$

In equation (12), load pressure  $x_3$  can be treated as a virtual control input function at this step. A virtual control law  $\alpha_2$  will be synthesized for the load pressure  $x_3$  such that the output tracking error  $z_1$  will converge to a small value with guaranteed transient performance. The resulting control function  $\alpha_2$  is given by:

$$\alpha_2 = \alpha_{2a} + \alpha_{2s} \quad \alpha_{2s} = \alpha_{2s1} + \alpha_{2s2} \quad (13)$$

$$\alpha_{2a} = \frac{m_s}{A_m} \left( -\frac{\rho x_1 + m_T}{m_s} g - \hat{\theta}_1 + \dot{x}_{2eq} \right)$$

$$\alpha_{2s1} = -\frac{m_s}{A_m} k_2 z_2$$

In (13),  $\alpha_{2a}$  functions as the adaptive control part used to achieve better model compensation through online parameter adaptation.  $\alpha_{2s}$  is the feedback control part where  $\alpha_{2s1}$  is a regular linear feedback part and  $\alpha_{2s2}$  is a robust feedback part that satisfies the following conditions:

$$\begin{aligned} z_2 \left( \frac{A_m}{m_s} \alpha_{2s2} - \tilde{\theta}^T \varphi_2 + \tilde{F}_d \right) &\leq \varepsilon_2 \\ z_2 \alpha_{2s2} &\leq 0 \end{aligned} \quad (14)$$

where  $\varepsilon_2 > 0$  is a design parameter and  $\varphi_2 = [1, 0]^T$ . How to synthesize  $\alpha_{2s2}$  to satisfy condition (14) can be found in (Yao and Tomizuka 1997). Let  $z_3 = x_3 - \alpha_2$  denote input discrepancy. For a positive-semidefinite function  $V_2 = \frac{1}{2} \omega_2 z_2^2$ , its derivative is given by:

$$\begin{aligned} \dot{V}_2 &= \omega_2 \frac{A_m}{m_s} z_2 z_3 + \omega_2 z_2 \left( \frac{A_m}{m_s} \alpha_{2s2} - \tilde{\theta}^T \varphi_2 + \tilde{F}_d \right) \\ &\quad - \omega_2 k_2 z_2^2 \end{aligned} \quad (15)$$

*Step 2:* In this step, the objective is to synthesize a control law  $\alpha_3$  for the actual pump displacement  $\alpha$  such that the load pressure  $x_3$  will track the virtual control function  $\alpha_2$  designed in the first step. The actual control input  $u$  for the electronic throttle input  $v$  can be calculated from the inverse nonlinear mapping  $f^{-1}$ . The derivative of input discrepancy  $z_2$  in *Step 1* can be written as:

$$\begin{aligned} \dot{z}_2 &= \frac{\beta_e}{V_T} (C_p \omega_p \alpha - C_m x_2 - D_c \omega_p) + \theta_2 + \tilde{Q}_L \\ &\quad - \dot{\alpha}_{2c} - \dot{\alpha}_{2u} \end{aligned} \quad (16)$$

where  $\dot{\alpha}_{2c} = \frac{\partial \alpha_2}{\partial x_1} x_2 + \frac{\partial \alpha_2}{\partial x_2} \hat{x}_2 + \frac{\partial \alpha_2}{\partial \hat{\theta}} \dot{\hat{\theta}} + \frac{\partial \alpha_2}{\partial t}$  is the calculable part of  $\dot{\alpha}_2$  and can be used in control function design.  $\dot{\alpha}_{2u} = \frac{\partial \alpha_2}{\partial x_2} (-\tilde{\theta}_1 + \tilde{F}_d)$  is the uncertain part of  $\dot{\alpha}_2$  which needs to be addressed with robust feedback.  $\hat{x}_2 = \frac{A_m}{m_s} x_3 + \frac{(\rho x_1 + m_T)}{m_s} g + \hat{\theta}_1$  is the calculable part of  $\dot{x}_2$ . Define a positive-semidefinite function  $V_3 = V_2 + \frac{1}{2} \omega_3 z_3^2$ , its derivative can be written as:

$$\begin{aligned} \dot{V}_3 &= \dot{V}_2|_{x_3=\alpha_3} + \omega_3 z_3 \left[ \frac{\beta_e}{V_T} (C_p \omega_p \alpha - C_m x_2 - D_c \omega_p) + \theta_2 + \tilde{Q}_L - \dot{\alpha}_{2c} - \dot{\alpha}_{2u} + \frac{\omega_2 A_m}{\omega_3 m_s} z_2 \right] \end{aligned} \quad (17)$$

Let control function  $\alpha = \alpha_3$ :

$$\alpha_3 = \alpha_{3a} + \alpha_{3s} \quad \alpha_{3s} = \alpha_{3s1} + \alpha_{3s2} \quad (18)$$

$$\begin{aligned} \alpha_{3a} &= \frac{1}{C_p \omega_p} [C_m x_2 + D_c \omega_p \\ &\quad + \frac{V_T}{\beta_3} \left( -\hat{\theta}_2 + \dot{\alpha}_{2c} - \frac{\omega_2 A_m}{\omega_3 m_s} z_2 \right)] \\ \alpha_{3s1} &= -\frac{1}{C_p \omega_p \beta_3} k_3 z_3 \end{aligned}$$

$\alpha_{3s2}$  is designed to satisfy following conditions using similar process for  $\alpha_{2s2}$  in *Step 1*:

$$\begin{aligned} z_3 \left( \frac{\beta_e}{V_T} C_p \omega_p \alpha_{3s2} - \tilde{\theta}^T \varphi_3 + \tilde{Q}_L - \frac{\partial \alpha_2}{\partial x_2} \tilde{F}_d \right) &\leq \varepsilon_3 \\ z_3 \alpha_{3s2} &\leq 0 \end{aligned} \quad (19)$$

where  $\varphi_3 = [-\frac{\partial \alpha_2}{\partial x_2}, 1]^T$ . From (17) and (18):

$$\begin{aligned} \dot{V}_3 &= \dot{V}_2|_{x_2=\alpha_2} - \omega_3 k_3 z_3^2 + \omega_3 z_3 \left( \frac{\beta_e}{V_T} C_p \omega_p \alpha_{3s2} \right. \\ &\quad \left. - \tilde{\theta}^T \varphi_3 + \tilde{Q}_L - \frac{\partial \alpha_2}{\partial x_2} \tilde{F}_d \right) \end{aligned} \quad (20)$$

The actual control command  $v$  can be calculated from inversed nonlinear mapping  $v = f^{-1}(\alpha_3)$

### 3.4 Theoretical performance results

**Theorem 1:** Let parameter estimation be update by adaptation law (16) in which  $\tau = \sum_{j=2}^3 \omega_j \varphi_j z_j$ , the control law (18) will guarantee that:

1. The output tracking error  $z = [z_1, z_2, z_3]^T$  are bounded and transient performance satisfies

$$V_3(t) \leq \exp(-\lambda_v t) V_3(0) + \frac{\varepsilon_v}{\lambda_v} [1 - \exp(-\lambda_v t)] \quad (21)$$

where  $\lambda_v = 2 \min \{k_2, k_3\}$  and  $\varepsilon_v = \omega_2 \varepsilon_2 + \omega_3 \varepsilon_3$

2. If after a finite time  $t_0, \tilde{F}_d = \tilde{Q}_L = 0$ , which means only parametric uncertainties are presented, asymptotic tracking can be achieved.

Detailed proof can be found in (Yao, Bu et al. 2000)

## 4. CONTROLLER SOFTWARE IMPLEMENTATION

In this section, the controller software architecture based on micro-services will be introduced with each component. The controller software development process which follows a Model-based design process will be detailed.

### 4.1 Controller software architecture

A micro-service based architecture has many advantages (Newman 2015) and is popular for cloud and web development. As shown in Fig. 6, the controller software is implemented based on a micro-services architecture. Each hexagon represents a micro-service packaged in a docker container (Turnbull 2014). Their detailed functions are illustrated as follows: 1) A hardware interface service interacts with hardware through Serial, CAN and ProfiBus to acquire sensing information such as tool position/speed, drum speed and hydraulic pressures. It also sends out a pump displacement command  $v$  via CAN. The hardware interface service is active when running in actual operation; 2) Simulator, built from Simlink/SimScape system dynamics model detailed in Section 2, provides system behaviour information when the system is running in a software-in-loop testing mode; 3) ARC controller, designed in Section 3, is also a micro-service packaged in a docker container. It receives sensing information and sends out control commands; 4) To shared data among different micro-services, RabbitMQ (Bender, Ward et al. 2016), an open source message broker, is used to routing messages with data among different micro-services.

### 4.2 Model-based design process

The controller software development follows a Model-Based Design process. The advantages of a MBD process is well

documented (Aarenstrup 2016). It is the de facto standard for the embedded system development in various industries such as automotive, aerospace and medical electronics.

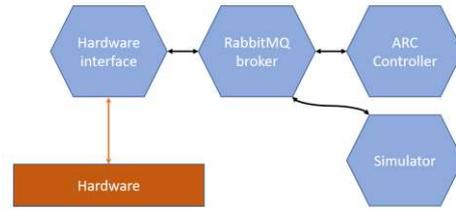


Figure 6 Controller software architecture

In the ARC controller development and implementation, a system model, including both winch hydraulics and winch mechanics (Section 2), was developed using Matlab, Simulink and SimScape first. To enable the software-in-loop testing, the system model, constructed using Matlab/Simulink/SimScape, is built into a Simulator in Fig. 6, a micro-service packed in a docker container. The process, that converts Simulink and SimScape blocks to an MBD micro-service packaged in a docker container, is illustrated in Fig. 7. It utilizes the code generation capabilities of Simulink/SimScape together with a Java wrapper developed in-house.

The ARC controller designed in Section 3 is constructed in Simulink. Before it was built into a micro-service, extensive model-in-loop tests were conducted. The controller is built into a docker container using the same process illustrated in Fig. 7.

Once the Simulator and/or ARC controller's docker images are published, various software-in-loop tests will be running before actual operation test. The whole build and test process are conducted in a Continuous Integration (CI) server automatically when a controller or simulator modification is pushed to the code repository.

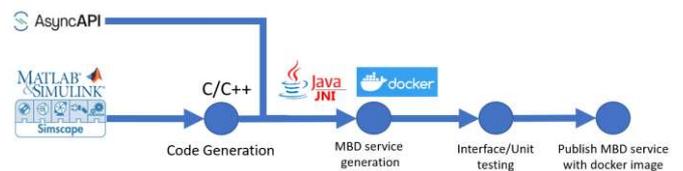


Figure 7 From Simulink/SimScape blocks to micro-service in docker container

## 5. PRELIMINARY EXPERIMENTAL TESTING RESULTS

A preliminary controller was designed following design steps in Section 3. Due to the short development and testing time, the robust control parts (Eq. (14) and Eq. (19)) and the adaptation for the flow rate error  $\theta_2$  are not turned on in the preliminary controller testing. Testing was conducted at a vertical testing well. Fig. 8 shows the test result moving the tool from surface to 80 meters deep, in the well. The first two plots show the desired depth and speed. The desired trajectory is generated from a linear filter with speed and acceleration limits. The 3<sup>rd</sup> plot shows the tracking error. The error grows

when the tool starts accelerating initially. The parameter adaptation reduces the error once the tool enters the cruising phase with the maximal speed. The tracking error increases first when tool starts decelerating. It decreases when the tool approaches the target depth. The final tracking error is well within the 0.1m requirement for the operation. Fig. 8 also shows significant sensing noise in the depth measurement.

## 6. CONCLUSIONS

In this paper, an adaptive robust controller is designed for the tool motion control in wireline operation in the oil & gas industry. Driven by a hydrostatic transmission with variable displacement hydraulic pump and motor, the system dynamics are highly nonlinear and subject to parametric uncertainties and uncertain nonlinearities. The designed ARC controller takes into account system nonlinearities as well as uncertainties by using nonlinear robust control, parameter adaptation and backstepping design. Implemented as a micro-service packaged in a docker container, the controller software development and implementation follows the MBD process. Preliminary experimental results show the effectiveness of the ARC controller design, and controller software development process.

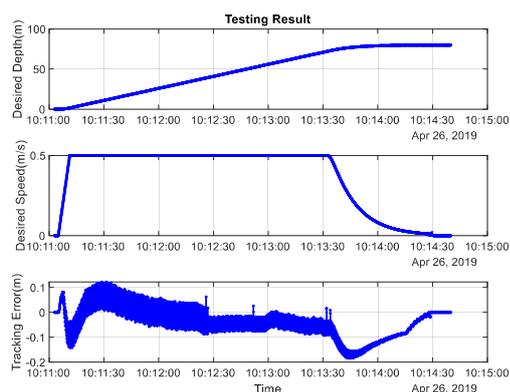


Figure 8 Testing results

The current controller design still has several drawbacks that require further refinements. First, cable dynamics is ignored in both modelling and controller design. This works acceptably when the well is shallow and the released cable is short. When the released cable is longer, the cable dynamics, described by a partial differential equation (PDE), wave equation, will be dominant. Motion measurements acquired by surface sensors no longer describe downhole tool string movement; Second, current tool motion trajectory is generated through a linear filter with speed and acceleration limits. To make the trajectory physically feasible, the parameters and limits are often conservative. It is necessary to introduce certain optimizations with system dynamics to minimize tripping time and still maintain control accuracy. Finally, there are many sensors measuring various physical aspects of the whole system. To fully utilize the information provided by these sensors, a fusion of their measurements using a Kalman type filter is desirable for better controller performance, fault tolerance and diagnostics.

## ACKNOWLEDGEMENT

The author would like to thank Stephen Georget for discussions and collaborations on system modelling and controller design, Vassilis Varveropoulos and Ken Ditlefsen for help on controller implementation, and Weijia Yang for actual operational test.

## REFERENCES

- Aarenstrup, R. (2016). Managing Model-Based Design.
- Aschemann, H. and H. Sun (2013). Decentralised flatness-based control of a hydrostatic drive train subject to actuator uncertainty and disturbances. 2013 18th International Conference on Methods & Models in Automation & Robotics (MMAR).
- Bender, A., J. R. Ward, S. Worrall, M. L. Moreyra, S. G. Konrad, F. Masson and E. M. Nebot (2016). "A Flexible System Architecture for Acquisition and Storage of Naturalistic Driving Data." IEEE Transactions on Intelligent Transportation Systems 17(6): 1748-1761.
- Bu, F. and J. Dykstra (2014). Indirect Adaptive Robust Controller Design for Drilling Rotary Motion Control. ASME 2014 Dynamic Systems and Control Conference. San Antonio, TX.
- Bu, F. and H.-S. Tan (2007). "Pneumatic brake control for precision stopping of heavy-duty vehicles." IEEE Transactions on Control Systems Technology 15(1): 53-64.
- Krstic, M., I. Kanellakopoulos and P. Kokotovic (1995). Nonlinear and Adaptive Control Design.
- Manring, N. D. and G. R. Luecke (1998). "Modeling and Designing a Hydrostatic Transmission With a Fixed-Displacement Motor." Journal of Dynamic Systems, Measurement, and Control 120(1): 45-49.
- Newman, S. (2015). Building Microservices, O'Reilly Media, Inc.
- Sastry, S. and M. Bodson (1989). Adaptive Control: Stability, Convergence, and Robustness.
- Sun, H. and H. Aschemann (2013). Sliding-mode control of a hydrostatic drive train with uncertain actuator dynamics. 2013 European Control Conference (ECC).
- Sun, H. and H. Aschemann (2016). "A Backstepping Sliding Mode Control for a Hydrostatic Transmission with Unknown Disturbances." IFAC-PapersOnLine 49(18): 879-884.
- Turnbull, J. (2014). The Docker Book: Containerization is the new virtualization.
- Xu, L. and B. Yao (2001). "Output feedback adaptive robust precision motion control of linear motors." Automatica 37(7): 1029-1039.
- Yao, B., F. Bu, J. Reedy and G. T. C. Chiu (2000). "Adaptive robust motion control of single-rod hydraulic actuators: Theory and experiments." IEEE/ASME Transactions on Mechatronics 5(1): 79-91.
- Yao, B. and M. Tomizuka (1994). Smooth robust adaptive sliding mode control of manipulators with guaranteed transient performance. Proceedings of 1994 American Control Conference - ACC '94.
- Yao, B. and M. Tomizuka (1997). "Adaptive robust control of SISO nonlinear systems in a semi-strict feedback form." Automatica 33(5): 893-900.