A priedas. Pilnas krantinės krano matematinis modelis

Pirmojo kūno (krantinės krano vežimėlio) slenkamojo judesio judėjimo lygčių sistema:

$$\begin{aligned}&[M_{C1}]\{\ddot{q}_{C1}\} = \{Q_{C1,31}\} + \{Q_{C1,41}\} + \{Q_{C1,5}\} + \{Q_{C1,6}\} + \\ &+ \{Q_{C1,7}\} + \{Q_{C1,8}\} + \{Q_{C1,10}\} + \{Q_{C1,20}\} + \{Q_{C1,w}\}.\end{aligned} \tag{A1}$$

Krantinės krano vežimėlio sukimosi apie X_{c1} ašį judėjimo lygtis:

$$I_{C1,X}\ddot{\phi}_1 = Q_{\phi 1,31} + Q_{C1,41} + Q_{\phi 1,5} + Q_{\phi 1,6} + Q_{\phi 1,7} + Q_{\phi 1,8} + Q_{\phi 1,10} + Q_{\phi 1,20}. \tag{A2}$$

Krantinės krano griebtuvo skriemulių sukimosi apie ašį judėjimo lygtis:

$$\begin{aligned}I_{21}\ddot{\phi}_{21} &= (F_5 - F_6) R_{21} - M_{21,R}(\dot{\phi}_{21}), \\ I_{22}\ddot{\phi}_{22} &= (F_7 - F_8) R_{22} - M_{22,R}(\dot{\phi}_{22}).\end{aligned}, \tag{A3}$$

Vežimėlio kontaktiniuose taškuose ir perdavos diržo segmente veikiančios jėgos:

$$\{Q_{C1,31}\} = \begin{Bmatrix} -k_{31}\left(R_3\phi_3 + q_{C1y} - b_1\phi_1\right) - c_{31}\left(R_3\dot{\phi}_3 + \dot{q}_{C1y} - b_1\dot{\phi}_1\right) \\ 0 \end{Bmatrix},$$

$$\{Q_{C1,41}\} = \begin{Bmatrix} -k_{41}\left(R_4\phi_4 + q_{C1y} - b_2\phi_1\right) - c_{41}\left(R_4\dot{\phi}_4 + \dot{q}_{C1y} - b_2\dot{\phi}_1\right) \\ 0 \end{Bmatrix},$$

$$\{Q_{C1,10}\} = \begin{Bmatrix} 0 \\ -k_{10}\left[\left(q_{C1Z} + a_1\phi_1\right) - z_{11}\right] - c_{10}\left[\left(\dot{q}_{C1Z} + a_1\dot{\phi}_1\right) - \dot{z}_{11}\right] \end{Bmatrix},$$

$$\{Q_{C1,20}\} = \begin{Bmatrix} 0 \\ -k_{20}\left[\left(q_{C1Z} + a_2\phi_1\right) - z_{12}\right] - c_{20}\left[\left(\dot{q}_{C1Z} + a_2\dot{\phi}_1\right) - \dot{z}_{12}\right] \end{Bmatrix},$$

$$\{Q_{C1,w}\}^T = [0, -m_{C1}g]. \tag{A4}$$

Krantinės krano 5 lyne veikianti jėgos vektorius:

$$\{Q_{C1,5}\} = \frac{F_5}{L_6}\{R_{9,10}\}, \tag{A5}$$

$$F_5 = -\left(k_5 \Delta L_5 + c_5 \left(\Delta \dot{L}_5 - R_{21}\dot{\phi}_{21}\right)\right) H\left(\Delta L_5\right),\ L_5^2 = \left\{R_{9,10}\right\}^T \left\{R_{9,10}\right\},$$

$$\left\{R_{9,10}\right\} = \left\{R_9\right\} - \left\{R_{10}\right\};\ \left\{R_9\right\} = \left\{R_{o12}\right\} + R_{13}\begin{Bmatrix}\cos\beta \\ \sin\beta\end{Bmatrix},$$

$$\beta = \alpha - \gamma,\ \sin\alpha = \frac{R_{o21Z} - R_{o12Z}}{L_{o12,o21}},\ \sin\gamma = \frac{R_{13} - R_{21}}{L_{o12,o21}},$$

$$\left\{R_{o12}\right\} = \left\{R_{C10}\right\} + \left\{q_{C1}\right\} + \left[A\left(\phi_1\right)\right]\left\{r_{C1,12}\right\},$$
$$\left\{R_{o21}\right\} = \left\{R_{C20}\right\} + \left\{q_{C2}\right\} + \left[A\left(\phi_2\right)\right]\left\{r_{C2,21}\right\},$$

$$\left\{R_{10}\right\} = \left\{R_{21}\right\} + R_{21}\begin{Bmatrix}\cos\beta \\ \sin\beta\end{Bmatrix},\ \beta = \alpha - \gamma,\ \sin\alpha = \frac{R_{o21Z} - R_{o12Z}}{L_{o12,o21}},$$

$$\sin\gamma = \frac{R_{13} - R_{21}}{L_{o12,o21}},\ L_{12,21}^2 = \left\{R_{12,21}\right\}^T \left\{R_{12,21}\right\},$$

$$\left\{R_{12,21}\right\} = \left\{R_{C10}\right\} + \left\{q_{C1}\right\} + \left[A\left(\phi_1\right)\right]\left\{r_{C1,12}\right\} - \left\{R_{C20}\right\} + \left\{q_{C2}\right\} + \left[A\left(\phi_2\right)\right]\left\{r_{C2,21}\right\},$$

$$\Delta L_5 = L_5 - L_{50} - R_{21}\phi_{21},\ \Delta \dot{L}_5 = \frac{1}{L_5}\left\{\dot{R}_{9,10}\right\}^T \left\{R_{9,10}\right\} - R_{21}\dot{\phi}_{21}. \qquad \text{(A6)}$$

H(x) – Heavisido funkcija.

Krantinės krano 6 lyne veikianti tempimo jėgos vektorius:

$$\left\{Q_{C1,6}\right\} = \frac{F_6}{L_6}\left\{R_{12,11}\right\} \qquad \text{(A7)}$$

$$F_6 = -\left(k_6 \Delta L_6 + c_6 \left(\Delta \dot{L}_6 + R_{21}\dot{\phi}_{21}\right)\right) H\left(\Delta L_6\right),$$

$$\Delta L_6 = L_6 - L_{60} + R_{21}\phi_{21},$$

$$\Delta \dot{L}_6 = \dot{L}_6 + R_{21}\dot{\phi}_{21},\ L_6^2 = \left\{R_{12,11}\right\}^T \left\{R_{12,11}\right\},\ \left\{R_{12,11}\right\} = \left\{R_{12}\right\} - \left\{R_{11}\right\},$$

$$\left\{R_{12}\right\} = \left\{R_{C10}\right\} + \left\{q_{C1}\right\} + \left[A\left(\phi_1\right)\right]\left\{r_{C1,12}\right\},\ \left\{R_{11}\right\} = \left\{R_{21}\right\} + R_{21}\begin{Bmatrix}\cos\alpha_{11} \\ \sin\alpha_{11}\end{Bmatrix},$$

$$\alpha_{11} = \alpha - \beta,\ \sin\alpha = \frac{R_{o21Z} - R_{12Z}}{L_{12,21}},\ \sin\beta = \frac{R_{21}}{L_{12,21}},\ L_{12,o21}^2 = \left\{R_{12,21}\right\}^T \left\{R_{12,21}\right\},$$

$$\left\{R_{12,21}\right\}=\left\{R_{C10}\right\}+\left\{q_{C1}\right\}+\left[A\left(\phi_1\right)\right]\left\{r_{C1,12}\right\}-\left\{R_{C20}\right\}-\left\{q_{C2}\right\}-\left[A\left(\phi_2\right)\right]\left\{r_{C2,21}\right\},$$

$$\dot{L}_6=\frac{1}{L_6}\left\{\dot{R}_{12,11}\right\}^T\left\{R_{12,11}\right\}. \tag{A8}$$

Krantinės krano 7 lyne veikianti tempimo jėgos vektorius:

$$\left\{Q_{C1,7}\right\}=\frac{F_7}{L_7}\left\{R_{12,13}\right\}, \tag{A9}$$

$$F_7=-\left(k_7\Delta L_7+c_7\left(\Delta\dot{L}_7+R_{22}\dot{\phi}_{22}\right)\right)H\left(\Delta L_7\right),$$

$$\Delta L_7=L_7-L_{70}+R_{22}\phi_{22},$$

$$\Delta\dot{L}_7=\dot{L}_7-R_2\dot{\phi}_{22},\ L_7^2=\left\{R_{12,13}\right\}^T\left\{R_{12,13}\right\},\ \left\{R_{12,13}\right\}=\left\{R_{12}\right\}-\left\{R_{13}\right\},$$

$$\left\{R_{13}\right\}=\left\{R_{22}\right\}+R_{22}\begin{Bmatrix}\cos\alpha_{13}\\ \sin\alpha_{13}\end{Bmatrix},\ \alpha_{13}=\alpha-\beta,\ \sin\alpha=\frac{R_{12Z}-R_{22Z}}{L_{12,22}},$$

$$\sin\beta=\frac{R_{22}}{L_{12,22}},\ L_{12,22}^2=\left\{R_{12,o22}\right\}^T\left\{R_{12,o22}\right\},$$

$$\left\{R_{12,22}\right\}=\left\{R_{C10}\right\}+\left\{q_{C1}\right\}+\left[A\left(\phi_1\right)\right]\left\{r_{C1,12}\right\}-\left\{R_{C20}\right\}-\left\{q_{C2}\right\}-\left[A\left(\phi_2\right)\right]\left\{r_{C2,22}\right\},$$

$$\dot{L}_7=\frac{1}{L_7}\left\{\dot{R}_{12,13}\right\}^T\left\{R_{12,13}\right\}. \tag{A10}$$

Krantinės krano 8 lyne veikianti tempimo jėgos vektorius:

$$\left\{Q_{C1,8}\right\}=\frac{F_8}{L_8}\left\{R_{15,14}\right\}, \tag{A11}$$

$$F_8=-\left(k_8\Delta L_8+c_8\left(R_{22}\dot{\phi}_{22}\right)\right)H\left(\Delta L_8\right),\ \Delta L_8=L_8-L_{80}+R_{22}\phi_{22},$$

$$\Delta\dot{L}_8=\dot{L}_8-R_{22}\dot{\phi}_{22},\ L_8^2=\left\{R_{15,14}\right\}^T\left\{R_{15,14}\right\},\ \left\{R_{15,14}\right\}=\left\{R_{15}\right\}-\left\{R_{14}\right\},$$

$$\left\{R_{15}\right\}=\left\{R_{14}\right\}+R_{16}\begin{Bmatrix}\cos\beta_{15}\\ \sin\beta_{15}\end{Bmatrix},\ \beta_{15}=\alpha_{15}-\gamma_{15},\ \sin\alpha_{15}=\frac{R_{22Z}-R_{14Z}}{L_{22,14}},$$

$$\sin\gamma_{15}=\frac{R_{16}-R_{22}}{L_{22,14}},\ L_{22,14}^2=\left\{R_{14,22}\right\}^T\left\{R_{14,22}\right\},$$

$$\left\{R_{14,22}\right\}=\left\{R_{C10}\right\}+\left\{q_{C1}\right\}+\left[A\left(\phi_1\right)\right]\left\{r_{C1,14}\right\}-\left\{R_{C20}\right\}-\left\{q_{C2}\right\}-\left[A\left(\phi_2\right)\right]\left\{r_{C2,22}\right\},$$

$$\dot{L}_8=\frac{1}{L_8}\left\{\dot{R}_{15,14}\right\}^T\left\{R_{15,14}\right\}. \tag{A12}$$

Jėgos, veikiančios vežimėlio perdavos diržo segmente *31* ir *34*, lynuose *5,6,7,8* tempimo jėgų sukimo momentai apie C_1 tašką:

$$Q_{\phi 1,31}=k_{31}b_1\left(R_3\phi_3+q_{C1y}-b_1\phi_1\right)+c_{31}b_1\left(R_3\dot{\phi}_3+\dot{q}_{C1y}-b_1\dot{\phi}_1\right) \tag{A13}$$

$$Q_{\phi 1,34}=k_{41}b_2\left(R_4\phi_4+q_{C1y}-b_2\phi_1\right)+c_{41}b_2\left(R_4\dot{\phi}_4+\dot{q}_{C1y}-b_2\dot{\phi}_1\right) \tag{A14}$$

$$Q_{\phi 1,5}=-\frac{F_5}{L_5}\begin{bmatrix}r_{C1,9y}\left(R_{9,10y}\sin\phi_1+R_{9,10Z}\cos\phi_1\right)-\\-r_{C1,9Z}\left(R_{9,10y}\cos\phi_1+R_{9,10Z}\sin\phi_1\right)\end{bmatrix},$$

$$Q_{\phi 1,6}=-\frac{F_6}{L_6}\begin{bmatrix}r_{C1,12y}\left(R_{12,11y}\sin\phi_1+R_{12,11Z}\cos\phi_1\right)-\\-r_{C1,12Z}\left(R_{12,11y}\cos\phi_1+R_{12,11Z}\sin\phi_1\right)\end{bmatrix},$$

$$Q_{\phi 1,7}=-\frac{F_7}{L_7}\begin{bmatrix}r_{C1,12y}\left(R_{12,13y}\sin\phi_1+R_{12,13Z}\cos\phi_1\right)-\\-r_{C1,12Z}\left(R_{12,13y}\cos\phi_1+R_{12,13Z}\sin\phi_1\right)\end{bmatrix},$$

$$Q_{\phi 1,8}=-\frac{F_8}{L_8}\begin{bmatrix}r_{C1,15y}\left(R_{15,14y}\sin\phi_1+R_{15,14Z}\cos\phi_1\right)-\\-r_{C1,14Z}\left(R_{15,15y}\cos\phi_1+R_{15,14Z}\sin\phi_1\right)\end{bmatrix}. \tag{A15}$$

Vežimėlio atramų *10* ir *20* reakcijų jėgų sukimo momentai apie C_1 tašką:

$$Q_{C1,10}=-k_{10}a_1\left(q_{C1Z}+a_1\phi_1-z_{11}\right)+c_{10}a_1\left(\dot{q}_{C1Z}+a_1\dot{\phi}_1-\dot{z}_{11}\right), \tag{A16}$$

$$Q_{C1,20}=+k_{20}a_2\left(q_{C1Z}-a_2\phi_1-z_{12}\right)+c_{10}a_1\left(\dot{q}_{C1Z}-a_1\dot{\phi}_2-\dot{z}_{12}\right). \tag{A17}$$

Antrojo kūno (krantinės krano griebtuvo) slenkamojo judesio judėjimo lygčių sistema:

$$\begin{aligned}\left[M_{C2}\right]\left\{\ddot{q}_{C2}\right\}&=\left\{Q_{C2,5}\right\}+\left\{Q_{C2,6}\right\}+\left\{Q_{C2,7}\right\}+\\&+\left\{Q_{C2,8}\right\}+\left\{Q_{C2,aero}\right\}+\left\{Q_{C2,F1}\right\}+\left\{Q_{C2,w}\right\}.\end{aligned} \tag{A18}$$

Krantinės krano griebtuvo sukimosi apie X_{c2} ašį judėjimo lygtis:

$$I_{C2,X}\,\ddot{\phi}_2 = Q_{C2,5} + Q_{C2,6} + Q_{C2,7} + Q_{C2,8} + Q_{C2,aero} + Q_{C2,F1}. \qquad \text{(A19)}$$

Krantinės krano *5, 6, 7, 8* lynuose veikiančių tempimo jėgų vektoriai:

$$\left\{Q_{C2,5}\right\} = -\frac{F_5}{L_5}\left\{R_{9,10}\right\},\ \left\{Q_{C2,6}\right\} = -\frac{F_6}{L_6}\left\{R_{12,11}\right\},\ \left\{Q_{C2,7}\right\} = -\frac{F_7}{L_7}\left\{R_{12,13}\right\},$$

$$\left\{Q_{C2,8}\right\} = -\frac{F_8}{L_8}\left\{R_{15,14}\right\},\ \left\{Q_{C2,w}\right\}^T = \left[0, -m_{C2}\,g\right],$$

$$\left\{Q_{C2,aero}\right\} = \frac{1}{2}\,qC_{aero2}\left(\left\{V_{\Sigma C2}\right\}^T\left\{V_{\Sigma C2}\right\}\right)\frac{\left\{V_{\Sigma C2}\right\}}{\left|\left\{V_{\Sigma C2}\right\}\right|},$$

$$\left\{V_{\Sigma C2}\right\} = \left\{V_{C2}\right\} - \left\{V_{WIND}\right\},$$

$$\left\{Q_{C2,F1}\right\}^T = \left[0, -F_1\right], \qquad \text{(A20)}$$

$$Q_{\phi 2,5} = -\frac{F_5}{L_5}\begin{bmatrix} r_{C2,10y}\left(R_{9,10y}\sin\phi_2 + R_{9,10Z}\cos\phi_2\right) - \\ -r_{C2,10Z}\left(R_{9,10y}\cos\phi_2 + R_{9,10Z}\sin\phi_2\right)\end{bmatrix},$$

$$Q_{\phi 2,6} = -\frac{F_6}{L_6}\begin{bmatrix} r_{C2,11y}\left(R_{12,11y}\sin\phi_2 + R_{12,11Z}\cos\phi_2\right) - \\ -r_{C2,11Z}\left(R_{12,11y}\cos\phi_2 + R_{12,11Z}\sin\phi_2\right)\end{bmatrix},$$

$$Q_{\phi 2,7} = -\frac{F_7}{L_7}\begin{bmatrix} r_{C2,13y}\left(R_{12,13y}\sin\phi_2 + R_{12,13Z}\cos\phi_2\right) - \\ -r_{C2,13Z}\left(R_{12,13y}\cos\phi_2 + R_{12,13Z}\sin\phi_2\right)\end{bmatrix},$$

$$Q_{\phi 2,8} = -\frac{F_8}{L_8}\begin{bmatrix} r_{C2,14y}\left(R_{15,14y}\sin\phi_2 + R_{15,14Z}\cos\phi_2\right) - \\ -r_{C2,14Z}\left(R_{15,14y}\cos\phi_2 + R_{15,14Z}\sin\phi_2\right)\end{bmatrix}.$$

Krano griebtuvo su kroviniu, lynuose *5, 6, 7, 8* veikiančių tempimo jėgų momentai, apie ašį X_{C2}:

$$\begin{aligned} Q_{\phi 2,aero} &= r_{C2,py}\left(Q_{C2,aeroZ}\cos\phi_2 - Q_{C2,aeroy}\sin\phi_2\right) - \\ &-r_{C2,pZ}\left(Q_{C2,aeroy}\cos\phi_2 - Q_{C2,aeroZ}\sin\phi_2\right). \end{aligned} \qquad \text{(A21)}$$

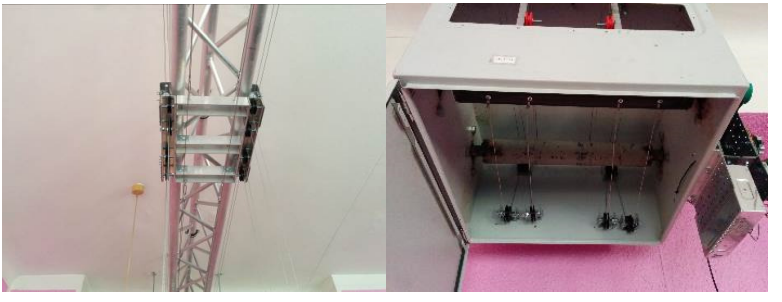
$\{r_{C2,p}\}$ – Pridėtas vėjo jėgos taškas veikiantis į antrą kūną bendroje koordinačių sistemoje $(X_{C2}, Y_{C2}, Z_{C2}, C_2)$:

$$Q_{\phi 2,F1} = -F_1 a_7 . \tag{A22}$$

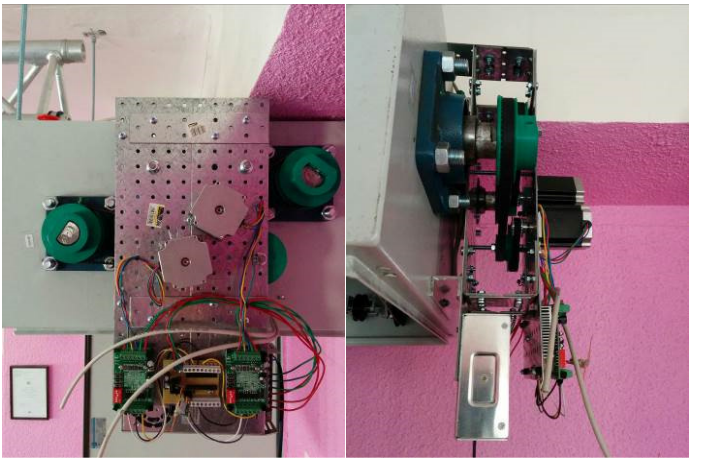
B priedas. Laboratorinis krantinės krano prototipas

Šiame priede pateikiama daugiau krantinės krano prototipo nuotraukų. Šios nuotraukos leidžia suprasti disertacijoje aprašyto laboratorinio krantinės krano prototipo struktūrą pagrindinius mazgus ir susidaryti bendra sistemos vaizdą, kurioje atlikti eksperimentiniai matavimai:

B1 pav. Prototipo priekinis ir šoninis vaizdai
Fig. B1. Side and front views of prototype

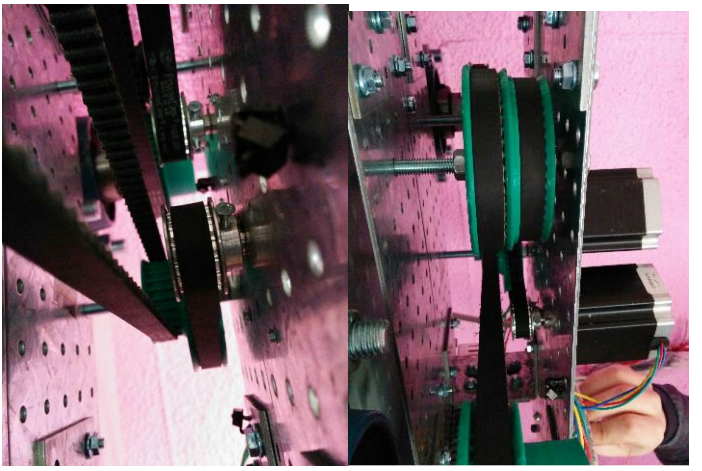


B2 pav. Vežimėlis ir variklių dėžė
Fig. B2. Trolley and motor box

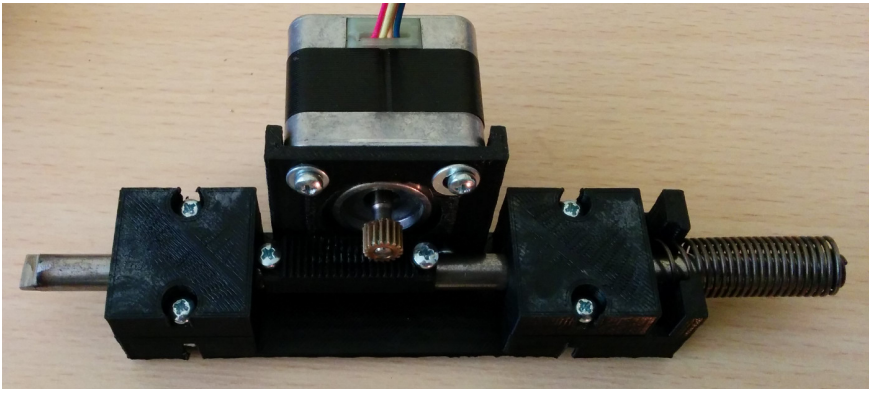


B3 pav. Valdymo blokas
Fig. B3. Control module

B4 pav. Kroviniai naudoti eksperimentuose
Fig. B4. Cargo types used in experiments



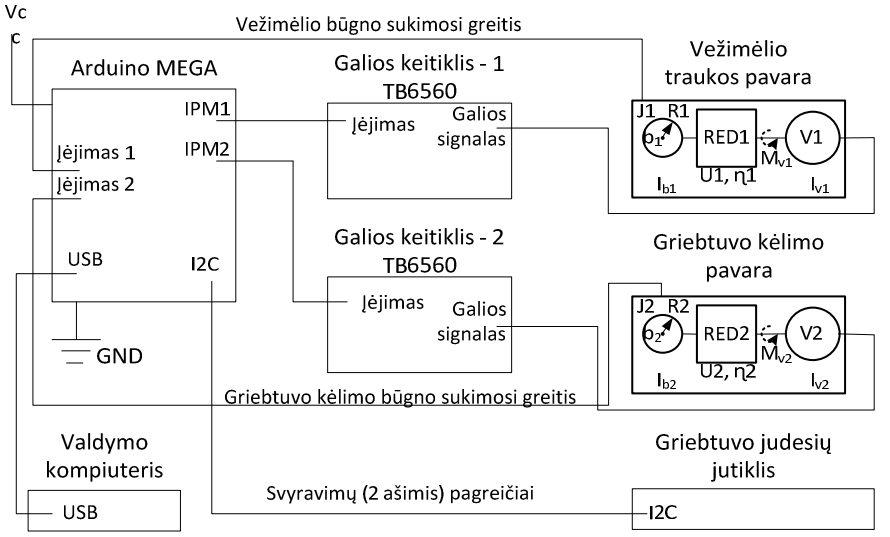
B5 pav. Perdavimo mechanizmai
Fig. B5. Reduction mechanisms



B6 pav. Avarinio stabdymo mechanizmas
Fig. B6. Mechanisms for emergency stop

C priedas. Valdymo algoritmo programos pagrindinis kodas

Šiame priede valdymo algoritmo programinis kodas parašytas laboratoriniam krantinės krano prototipui kurio valdymo blokine schema pateikiama žemiau (C1 paveikslas):



C1 pav. Valdymo sistemos blokine schema
Fig. C1. Block diagram of control system

Sistemoje naudojamų techninių priemonių parametrai pateikti lentelėje (C1 lentelė):

C1 lentelė. Valdymo sistemos techninių priemonių parametrai
Table C1. Technical parameters of main equipment

Parametras	Vertė / apibūdinimas
Arduino MEGA techniniai parametrai	
Mikrovaldiklis	ATmega2560
Dažnis	16 MHz
Skaitmeninai įvesties/išvesties uostai	54 (iš jų 15 IPM tipo)
Analoginiai įvesties/išvesties uostai	16
Maitinimo įtampa	5 VDC
Galios keitiklis	
Mikrovaldiklis	Toshiba TB6560AHQ
Maitinimo įtampa	10–35 VDC
Maksimali leistina apkrovos srovė	3A (maks 3,5A), reguliuojama apkrovos srovė (0,3–3 A)

C1 lentelės pabaiga

Parametras	Vertė / apibūdinimas
Paskirtis	Dviejų fazių žingsninio variklio valdymui
Darbo rėžimai	Pilno, pusės, 1/8, 1/16 žingsnio
Žingsninis variklis	
Modelis	NEMA 23
Pilnas žingsnis	1,8°
Maitinimo įtampa	3 VDC
Nominali srovė	3 A
Fazės varža	1 Omh
Fazės induktyvumas	3,5 mH
Trumpo jungimo momentas	1,9 Nm
Variklio rotoriaus inercijos momentas	$4{,}8 \times 10^{-5}$ kg·m^2
Greičio jutiklis	
Tipas	Optinis
Modelis	600P/R
Maitinimo įtampa	5–24 VDC
Impulsų skaičius per apsisukimą 1 fazei	600
Impulsų skaičius per apsisukimą 2 fazėms	2400
Maksimalus leistinas greitis	5000 aps/min
Darbo dažnio diapazonas	0–20 kHz
Griebtuvo judesių jutiklis	
Modelis	MPU–6050
Maitinimo įtampa	3–5 VDC
Matavimai3 ašių giroskopas	diapazonas: + 250 500 1000 2000 ° /s, 3 a-šių akselerometras (diapazonas: ± 2 ± 4 ± 8 ± 16 g)

Toliau pateikiamas valdymo algoritmo programinis kodas:

```
#include <TimerOne.h>
#include <TimerFour.h>
#include <stdlib.h>
#include <PID_v1.h>
#include "LPfilter.h"
#include "HPfilter.h"
```

```
#include "I2Cdev.h"
#include "MPU6050.h"
#if I2CDEV_IMPLEMENTATION == I2CDEV_ARDUINO_WIRE
#include "Wire.h"
#endif

MPU6050 accelgyro1(0x68);  // viršus
MPU6050 accelgyro2(0x69); // konteneris
//MPU6050 accelgyro(0x69); // <-- use for AD0 high
float last1 = 0.0;
float last3, last4 = 0.0;
float vy2HP = 0.0;
float vy2Old = 0.0;
float vy2Sum, vy2Count, vy2Avg = 0.0;
float vy2HPAbs, vKont = 0.0;

volatile float velocity YNew = 0.0;

float koef1 = 0.95;
int16_t ax1, ay1, az1;
float vx1, vy1, vz1;
int16_t gx1, gy1, gz1;

float last2 = 0.0;
float koef2 = 0.9;
int16_t ax2, ay2, az2;
float vx2, vy2, vz2 = 0.0;
int16_t gx2, gy2, gz2;

float gyro2Y = 0.0;

#define ENC_Y_KOEF  0.069115 //0.068 // 1 imp [mm]
#define ENC_Z_KOEF  0.0701622 //0.068 // 1 imp [mm]
#define DUTY          512
#define STEPSREV   200
#define DT              50     // ms
#define DTPID       50     // ms
#define DTLOG         50     // ms
#define DTVELOCITY     50

class Motor {
    typedef enum { CW, CCW } Direction;
    String name;
    boolean useTimer3 = false;

    int pinEn, pinStep, pinDir, pinStopCW, pinStopCCW; // pins

    float d; // būgno skersmuo mm
    int stepSize = 2; // motor driver step size Step size: 1 / [1,
2, 8, 16]
    float gearRatio;
```

```
    int trosoKoef;
    float systemKoef;

    /*** PID *******/
    PID pid;
    PID pid2;

    Direction curDir;
    boolean invertAxis;
    int curStopCW, curStopCCW; // curent stop values

    float startX = 0.0;
    float maxV = 0.0;
    float targetV = 0.0;
    float stopV = 0.0;
    int jerkState = 0;
    unsigned long curDT = 0;
    unsigned long previousMillis;

    float s = 0.5;
    float t_accel = 0.0;
    float a_vid = 0.0;
    float a_max = 0.0;
    float j = 0.0;

    float t_st1, t_st2, t_st3 = 0.0;
    float kp, ki, kd = 0.0;
    float kp2, ki2, kd2 = 0.0;
    float korekcija = 0.0;
    boolean usePid, usePid2, trapezoidal = false;
    unsigned long time_st_changed = 0;

  public:
    typedef enum { HOLD, J1, J2, J3, RUN, J5, J6, J7 } State;
    State curState;
    float curJ, curA, curV, curX = 0.0;
    float target = 0.0;
    float stopDist = 0.0;
    boolean flagMove = false;
    double output = 0.0;
    double input1, setpoint1, output1 = 0.0;
    double input2, setpoint2, output2 = 0.0;

    Motor(boolean usePid, boolean usePid2, String name, boolean
invertDir, int en, int dir, int step, int stopCW, int stopCCW,
float maxVelocity , float duration_accel, float diameter, float
gearRadio, int stepSize, int trosoKoef, boolean timer3)
      : pid(&input1, &output1, &setpoint1, 1.5, 8.0, 0.0, 0),
pid2(&input2, &output2, &setpoint2, 1.0, 0.0, 0.0, 0)
    {
      pid.SetOutputLimits(-0.24, 0.24);
```

```
      pid.SetSampleTime(DTPID);
      (usePid) ? pid.SetMode(AUTOMATIC) : pid.SetMode(0);

      setpoint2 = 0.00;
      pid2.SetOutputLimits(0.0, 0.1);
      pid2.SetSampleTime(DTPID);
      (usePid2) ? pid2.SetMode(AUTOMATIC) : pid2.SetMode(0);

      this->usePid = usePid;
      this->usePid2 = usePid2;
      this->name = name;
      this->invertAxis = invertDir;
      this->pinEn = en;
      this->pinDir = dir;
      this->pinStep = step;
      this->pinStopCW = stopCW;
      this->pinStopCCW = stopCCW;
      this->useTimer3 = timer3;

      pinMode(pinEn, OUTPUT);
      pinMode(pinDir, OUTPUT);
      pinMode(pinStep, OUTPUT);
      pinMode(pinStopCW, INPUT_PULLUP);
      pinMode(pinStopCCW, INPUT_PULLUP);
      this->curStopCW = digitalRead(pinStopCW);
      this->curStopCCW = digitalRead(pinStopCCW);

      this->maxV = maxVelocity ;
      this->t_accel = duration_accel;
      this->d = diameter;
      this->gearRatio = gearRadio;
      this->stepSize = stepSize;
      this->trosoKoef = trosoKoef;

      systemKoef = (15708 * diameter) / (5 * STEPSREV * stepSize *
gearRatio * trosoKoef);

      calculateProfileKoef();

      setTrapezoidalMode(trapezoidal);

      this->previousMillis = millis();
      setDir((invertDir) ? CCW : CW);
      hold();
    }

    /** CHANGE STATE ******************************/
    void hold() {…}

    void startCW() {…}
```

```
    void startCCW() {…}

    void run() {…}

    void stop() {…}

    /** UPDATE ***************************************/
    void update() {
      unsigned long currentMillis = millis();

      int newCurStopCW = digitalRead(pinStopCW);
      if (newCurStopCW != curStopCW) {
        curStopCW = newCurStopCW;
        Serial3.print("EndStop CW ");
        Serial3.print(name);
        Serial3.print(" ");
        Serial3.println(curStopCW);
      }
      if (curStopCW == LOW && getState() != HOLD && getDir() ==
CW) {
        target = 0.0;
        hold();
      }

      int newCurStopCCW = digitalRead(pinStopCCW);
      if (newCurStopCCW != curStopCCW) {
        curStopCCW = newCurStopCCW;
        Serial3.print("EndStop CCW ");
        Serial3.print(name);
        Serial3.print(" ");
        Serial3.println(curStopCCW);
      }
      curStopCCW = digitalRead(pinStopCCW);
      if (curStopCCW == LOW && getState() != HOLD && getDir() ==
CCW) {
        target = 0.0;
        hold();
      }

      // stop moveTo if target is reached    abs(y.curX) +
y.stopDist >= abs(targetY)
      if (flagMove && getState() != J5 && getState() != J6 &&
getState() != J7 && getState() != HOLD && abs(target - curX) -
stopDist <= 0.020) {
        stop();
        //        Serial3.println("TARGET REACHED. STOPPING...");
        flagMove = false;
      }

      if (getState() != HOLD) {
        pid.Compute();
```

```
        //pid2.Compute();
      }

      pid2.Compute();

      curDT = currentMillis - previousMillis;
      switch (getState()) {…}

  private:
    void setJerk(float jerk) {
      if ((getDir() == CCW && !invertAxis) || (getDir() == CW &&
invertAxis)) jerk = -jerk;

      float dt = curDT / 1000.0;
      curJ = jerk;
      if (this->trapezoidal) { // trapecinis
        curA = a_vid;
        if ((getDir() == CCW && !invertAxis) || (getDir() == CW &&
invertAxis)) curA = -curA;
        if (getState() == J6) curA = -curA;
      } else { // s-shaped
        curA += curJ * dt;
      }

      if (getState() == HOLD || getState() == RUN) {
        curJ = 0.0;
        curA = 0.0;
        if (getState() == HOLD) curV = 0.0;
      }
      //      curV += curA * dt + curJ * dt * dt / 2;
      //      curX += curV * dt + curA * dt * dt / 2 + curJ * dt *
dt * dt / 6;
      curV += (curA * dt + curJ * dt * dt / 2) / 1.0;
      curX += (1000 * curV * dt + 1000 * curA * dt * dt / 2 + 1000
* curJ * dt * dt * dt / 6 ) / 1000.0;
      //      float jerk_stop = ((getDir() == CCW && !invertAxis)
|| (getDir() == CW && invertAxis)) ? -maxJ : maxJ;
      //      float t_stop = -curA / jerk_stop;
      //      stopV =  abs(curA * t_stop + jerk_stop * t_stop *
t_stop / 2);

      float decel = ((getDir() == CCW && !invertAxis) || (getDir()
== CW && invertAxis)) ? -a_vid : a_vid;
      float t = -curV / decel;
      stopDist = abs(curV * t + decel * t * t / 2) - 0.03;
      //stopDist = 0.0;

      setpoint1 = curV;
    }

    void updatePWM() {
```

```
      if (usePid) {
        (getState() == HOLD) ? pid.SetMode(0) :
pid.SetMode(AUTOMATIC);
        //output += korekcija;
        output = (usePid2) ? output1 - output2 : output1;

        if (output > 0.24) {
          output = 0.24;
        } else if (output < -0.24) {
          output = -0.24;
        } else {}

        if (setpoint1 * output <= 0.0) {
          output = 0.0;
        }
        if (setpoint1 * output1 <= 0.0) {
          output1 = 0.0;
        }

        (useTimer3) ? Timer4.pwm(pinStep, DUTY, getPeriod(output))
: Timer1.pwm(pinStep, DUTY, getPeriod(output));

      } else if (usePid2) {…}
      else {
        (useTimer3) ? Timer4.pwm(pinStep, DUTY, getPeriod(curV)) :
Timer1.pwm(pinStep, DUTY, getPeriod(curV));
      }
    }

    unsigned int getPeriod(float v) {
      if (v != 0.0) return int(systemKoef / abs(v));
      else return 0;
    }
    /*********************************************/
  public:
    void setState( State newState) {…}
    void setDir( Direction newDir) {…}
    State getState() {…}
    Direction getDir() {…}

    void moveTo(float position) {…}

    void setTrapezoidalMode(boolean mode) {
      this->trapezoidal = mode;
      if (this->trapezoidal) { // trapecinis
        t_st1 = 0;
        t_st2 = t_accel * 1000;
        t_st3 = t_st1;
      } else { // s-shaped
        t_st1 = t_accel * s / 2 * 1000;
        t_st2 = t_accel * (1 - s) * 1000;
```

```
        t_st3 = t_st1;
      }
    }
    void setInput1(float input1) {…}
    void setInput2(float input2) {…}
    void setUsePid(boolean usePid) {…}
    void setUsePid2(boolean usePid2) {…}
    void setvMax(float input) {…}

    void setAccelDuration(float input) {…}

    void setKp(float kp) {…}
    void setKi(float ki) {…}
    void setKd(float kd) {…}
    void setKp2(float kp2) {…}
    void setKi2(float ki2) {…}
    void setKd2(float kd2) {…}

    void calculateProfileKoef() {
      this->a_vid = this->maxV / this->t_accel;
      this->a_max = this->a_vid / (1 - s * 0.5);
      this->j = 2 * this->a_max / (this->t_accel * s);
      setTrapezoidalMode(trapezoidal);
    }
    float getvMax() {…}
    float getJerk() {…}
    float setKorekcija(float korekcija) {…}
}; /*** End of Motor class
/*** GLOBAL  ****************************/

/*** ENCODER *******/
const int pinEncYA = 3;
const int pinEncYB = 2;
const int pinEncZA = 18;
const int pinEncZB = 19;

boolean encYASet, encYBSet, encZASet, encZBSet = false;

volatile long impY, impZ = 0;              // for velocity
calculation
volatile long posY, posZ = 0;             // imps
volatile float realPosY, realPosZ = 0.0;  // m
volatile float velocity Y, velocity Z = 0.0;      // m/s

/*** TIMERS *******/
unsigned long lastLog, lastVelocity Jutikliniu = 0;
unsigned long lastAVTime = 0;
float lastAV = 0.0;
float last2old = 0.0;
float korekcija = 0.0;
boolean sendLog = false;
```

```
boolean moveContainerFlag = false;
boolean moveContainerBackFlag = false;

/*** COMMANDS *******/
String inputCmd, inputValue = "";
boolean stringComplete, readingValue = false;

//Motor [xyz] (usePid, name, invertAxisDir, en, dir, step, stopCW,
stopCCW, maxVelocity , acceleration, deceleration, diameter,
gearRadio, stepSize, trosoKoef, useTimer3)
Motor y (true, true, "Y" , true, 38, 40, 11, 22, 26, 0.20, 1.4,
76.5, 3.0, 2, 1, false); // CCW positive
Motor z (false, false, "Z", false, 42, 44, 6, 30, 34, 0.12, 3.2,
90.0, 6.0, 2, 2, true); // CW positive

/*** FUNCTIONS *********************/

/*** ENCODER ******************/
void intYA() {…}
void intYB() {…}
void intZA() {…}
void intZB() {…}
void velocity Jutikliniu(unsigned long now) {…}

/*** LOG ******************/
void log(unsigned long now) {…}

void toggleLog(boolean newState = false) {…}

void serial3Event() {…}
}

float targetY = 1.9; float targetZ = 0.0; float targetZbottom =
0.0;
//float targetY = 1.9; float targetZ = 1.9; float targetZbottom =
1.4;

float targetPercent = 0.75;
boolean movingContainer = false;
boolean movingContainerBack = false;
boolean trapezoidal = false;
int movingState = 0;
int movingStateBack = 0;

void moveContainer() {…}

void moveContainerBack() {…}

/*** START HERE*****************************/
void setup() {…}
```

```
void loop() {
  unsigned long now = millis();
  serial3Event();
  if (stringComplete) {…}

  y.update();
  z.update();
  velocity Jutikliniu(now);

  if (moveContainerFlag) {
    moveContainer();
  }
  if (moveContainerBackFlag) {
    moveContainerBack();
  }

  log(now);
}
```