

Razvoj realnočasovnega razporejevalnika s prikazom delovanja na pogonskem sklopu vozila L7

Urban Rupnik¹, Roman Manko, Alen Alić, Mario Vukotić, Damijan Miljavec

¹ Univerza v Ljubljani, Fakulteta za elektrotehniko, Tržaška cesta 25, 1000 Ljubljana, Slovenija
E-pošta: urban.rupnik@fe.uni-lj.si

Implementation of Real-time scheduler on L7-type vehicle powertrain

Abstract. *The aim of this paper is to present the implementation and usage of the real-time scheduler based on Earliest deadline first (EDF) type using Matlab and a personal computer (PC). By upgrading the scheduler core with communication drivers, which are already developed by Mathworks, converts a PC to a powerful real-time master controller in different measurement and control setups. A high flexibility program distribution is achieved by using single PC as development environment as well as controlling device. The ability of real-time scheduler to control high number of external devices is presented on a L7-type vehicle power train test bench (i.e. inverter, battery management system (BMS), data acquisition systems, power supply, etc.). Presented real-time scheduler also successfully reduce test bench costs due to no need of additional external control devices.*

1 Uvod

Po končanem konstruiranju prototipov (npr. elektromotor) oziroma aplikacij (npr. pogonski sklop L7 električnega vozila) so za validacijo potrebne meritve. Te se glede na zahteve izvajajo na različne načine. Bodisi je potrebna meritev v posameznih delovnih točkah ali pa v naprej določenih voznih ciklih. Izvajanje takih meritev se lahko loči na ročne ali avtomatske meritve. Pri ročnih meritvah vse uporabljene naprave nastavimo v želena delovna točka, nato pa iz merilnih instrumentov odčitamo rezultate. Za večjo količino rezultatov, s katerimi ustrezneje ovrednotimo izdelek, pa se vse bolj uporabljajo avtomatske meritve. Enako kot pri ročnih je to nastavljanje delovnih točk, le da v tem primeru namesto operaterja nastavljanje opravlja krmilni sistem. S tem lahko merjenec poženemo v poljubna obratovalna stanja ali pa serijo stanj oziroma po v naprej določenih voznih ciklih. Dodatno se lahko v krmilne sisteme vgradi tudi simulacijske modele vse bolj aktualnih digitalnih dvojčkov. Na ta način dobimo sistem v zanki (ang. *Hardware-in-the-Loop*, HIL). Z dodanimi zunanji močnostnimi enotami pa močnostni sistem v zanki (ang. *Power-Hardware-in-the-Loop*, PHIL).

Za izvedbo krmiljenja sistemov in izračunov digitalnih dvojčkov je na trgu na veliko različnih produktov. Pomembnejše specifikacije, ki vplivajo na izbor so hitrost izvajanja programa, realnočasovnost, velikost programa, kompatibilnost s perifernimi enotami oz. komunikacijskimi protokoli, cenovna dostopnost, potrebna dodatna zunanja oprema, licenciranje, podpora, predznanja, idr.

Produkti podjetij dSpace [1] in SpeedGoat [2] so zagotovo med najzmogljivejšimi komercialno dostopnimi izdelki namenjenim za krmiljenje in regulacijo sistemov. Fiksni čas izračuna koraka modela in možnost izdelave programa na polju programirljivih nizov vrat (ang. *Field programmable gate array*, FPGA) omogočajo visoko natančnost in hitrost izračunov modelov. Visoka zmogljivost pa posledično pomeni tudi višjo ceno za napravo in za razširitve s katerimi lahko uporabljamo različne komunikacijske protokole.

Industrijski krmilniki (ang. *programmable logic controller*, PLC) so naprave namenjene krmiljenju in se množično uporabljajo v industriji [3] [4] [5] [6]. So nižjega cenovnega ramga od prej omenjenih. V primeru PLC ta že sam omogoča komunikacijo z zunanjimi napravami prek različnih protokolov ali pa je to dosegljivo z dodatnimi moduli.

V širši skupini razvojnih plošč imamo na voljo največ produktov različnih proizvajalcev in z različnimi zmogljivostmi [7] [8] [9] [10]. Nekatere izmed njih so namenjene učenju nekatere pa uporabi v vgrajenih sistemih. Vsem je skupno, da za uporabo v krmilnih sistemih, kjer je poleg zanesljivosti potreben tudi določen nivo robustnosti, potrebujejo nadgranjo z ohišjem, konektorji in napajalno logiko. Hkrati so ti izdelki na koncu opremljeni s komunikacijskim protokoli namenjenim le dotični aplikaciji in so tako manj fleksibilni.

Vsem zgoraj omenjenim skupinam naprav za krmiljenje sistemov je skupna uporaba dodatnega osebnega računalnika. Ta je potreben bodisi za razvoj programa ali pa tudi pri samem delovanju. Na tak način se torej za vsako merilno mesto potrebuje krmilni sistem in osebni računalnik (PC). Dandanes se avtomatizirana merilna mesta uporablja praktično povsod. To pomeni, da potrebujemo bodisi večje število krmilnih sistemov za sočasno rabo ali pa premikanje le-teh. To pa predstavlja ali cenovni ali pa časovni problem.

V prispevku je predstavljena izdelava krmilnega sistema za merilna mesta oziroma regulirane sisteme. Za osnovne cilje smo si postavili uporabo osebnega računalnika, ki bo hkrati razvojno orodje in krmilnik. S tem omogočimo fleksibilnost in sočasno uporabo na večih merilnih mestih. Zaradi raznolikosti zunanjih naprav, ki se uporabljajo na različnih merilnih mestih mora biti omogočena enostavna povezljivost z zunanjimi enotami (npr. laboratorijski napajalniki, osciloskopi, frekvenčni pretvorniki, BMS). Pri večini izvedb avtomatskih meritev s krmilnimi sistemi ali v HIL oz. PHIL aplikacijah ne potrebujemo točnega realnočasovnega izvajanja opravil, ker natančne meritve izvajamo z dodatnim instrumentarijem (npr. Dewesoft,

National Instruments, itd.). Ti imajo lastno uro. V prispevku predstavljen krmilni sistem tako ne zahteva natančne periode pri izvajanju opravila. Izvedba je zaradi uporabe osebnega računalnika tudi cenovno ugodna.

Članek v prvem delu govori o jedru programskega dela krmilnega sistema, ki je razporejevalnik opravil razvit v programskem paketu Matlab. V tem delu je opisana razčlenitev programa in prikaz delovanja na primeru razporejanja štirih preprostih opravil. V drugem delu je izdelan razporejevalnik opravil razširjen v aplikacijo, za prikaz delovanja na primeru krmiljenja merilnega mesta pogonskega sklopa L7 električnega vozila.

V zaključku bo kritično ovrednotena smiselnost uporabe izdelanega programa iz vidika cenovne dostopnosti, vložnega časa, možnosti uporabe v drugih projektih, ter možnosti razširitev.

2 Razporejevalnik in prevajalnik opravil

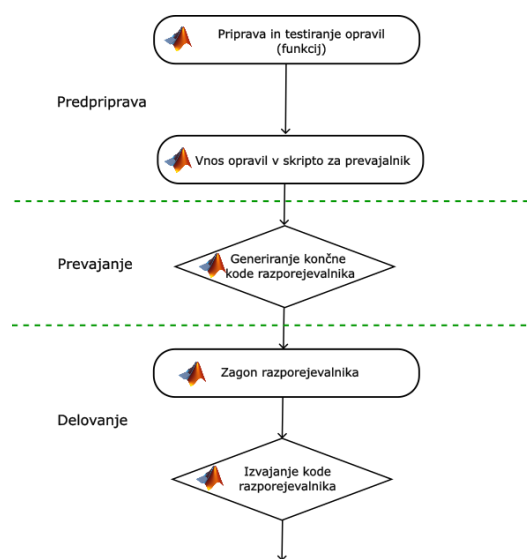
V krmilnih procesih se uporablja več osnovnih tipov razporejevalnikov opravil (npr. ang. *Rate-Monotonic Scheduling*, *Earliest Deadline First*, *Least Laxity First*, *Fixed Priority Pre-emptive Scheduling*, ...). Med seboj se razlikujejo po načinu dodeljevanja prioritet na podlagi katerih se določa zaporedje izvrševanja opravil. Bodisi so prioritete razdeljene glede na pomembnost opravila, na čas ki je potreben za izvedbo opravila ali pa na periodo opravila. Lahko so med posameznimi opravili dovoljene prekinitve, ki prožijo izvajanje drugega opravila, ali pa je dovoljeno zamuditi rok izvedbe opravila.

V primeru izdelave Matlab razporejevalnika je bil uporabljen razporejevalnik z najbližjim časovnim rokom (ang. »Earliest deadline scheduler«). Največjo prioriteto ima v tem primeru opravilo, katere rok za izvršitev je najbližji oziroma je najbolj pretekkel. Glavni prednosti tega razporejevalnika sta maksimalna izkoriščenost procesorskega časa in zagotovilo, da se vsa opravila izvajajo. Hkrati so v izdelanem razporejevalniku dovoljene tudi zamude časovnih rokov zato ni težav s stabilnostjo razporejevalnika. Pomembno je, da je shranjena informacija ob katerem času se je posamezno opravilo izvršilo.

Krmilni sistemi navadno zahtevajo več različnih opravil. Ročno vnašanje opravil v programsko kodo razporejevalnika je za uporabnika neprijazno, hkrati pa lahko privede do napak in časovno ne optimizirane kode. Opravila, ki so v programskem jeziku funkcije, vhodne in izhodne spremenljivke opravil bi lahko bili vhodni parametri v razporejevalnik. Tak pristop zaradi dodatnih operacij in počasnejšega klica funkcije doprinese k skupnemu času procesiranja (t.j. čas, ki ga uporabi razporejevalnik za dodeljevanje prioritet, izbiro opravila in klic opravila). Za minimizacijo časa procesiranja razporejevalnika opravil, enostavnost uporabe in preprečevanja napak je bil poleg razporejevalnika razvit tudi prevajalnik opravil.

Uporabnik v skripto vnese opravila v obliki imen funkcij ter definira njihovo zeleno periodo izvajanja. Poimenuje

tudi vhodne in izhodne spremenljivke. Nato prevajalnik napiše končno skripto razporejevalnika, ki vsebuje klice funkcij skupaj z ustreznimi vhodnimi in izhodnimi parametri. Delovni proces, ki se v celoti izvaja v Matlabu, je prikazan na sliki **Error! Reference source not found..** Izvajanje kode razporejevalnika je funkcija, ki glede na trenutni čas izvede ustrezno opravilo. Uporabljena ura je Matlabov interni časovnik, ki je vezan na procesorsko uro osebnega računalnika [12]. Ločljivost je tako odvisna od frekvence ure procesorja. Stabilnost delovanja razporejevalnika je v največji meri odvisna od ustrezno pripravljenih opravil, saj le napaka v programski kodi opravil in »sesutje« Matlabu lahko neželeno prekine izvajanje.



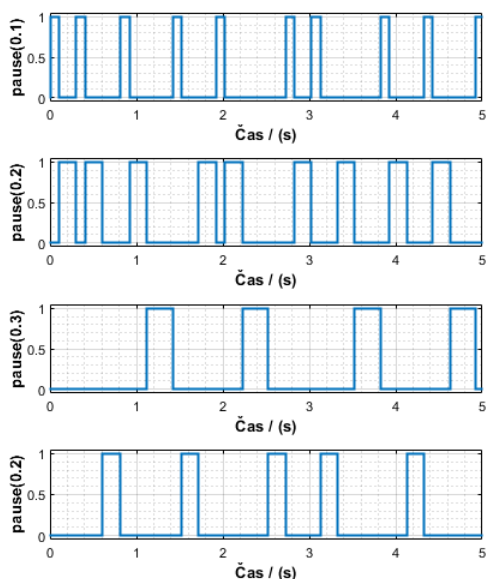
Slika 1: Potek dela z razporejevalnikom – od priprave opravil do izvajanja kode razporejevalnika.

Delovanje in meritve časa procesiranja razporejevalnika je bilo testirano s štirimi preprostimi opravili (tabela 1) na treh različnih osebnih računalnikih (tabela 2). Vsaka opravila predstavlja Matlabov ukaz za ustavitev programa (*pause*) z različnimi vhodnimi vrednostmi. Dolžine čakanja opravil in periode opravil so prikazane v tabeli 1. Zaporedje izvajanja opravil, kakor jih je razporedil razporejevalnik, je prikazano na sliki 2. Vrednost 1 pri posameznem opravilu predstavlja čas v katerem se opravilo izvaja.

Tabela 1: Opravila s predvidenim časom izvrševanja in zeleno periodo ponavljanja.

Opravilo	Čas izvrševanja (ms)	Perioda
pause(0.1)	100	300
pause(0.2)	200	300
pause(0.3)	300	1000
pause(0.2)	200	300

Rezultati meritev za 10 sekundni cikel oziroma 600 sekundni cikel so prikazani v tabeli 3 oziroma tabeli 4. Meritve porabljenega časa so bile opravljene z Matlabovim »Profiler« vtičnikom. Komentar rezultatov je podan v zaključku prispevka.



Slika 2: Zaporedje izvajanje opravil, kakor jih je razporedil razporejevalnik.

Tabela 2: Specifikacije računalnikov

Računalnik	1	2	3
Procesor	Intel i5-4590	Intel i5-4590	Intel i7-7700
Frekvenca	3,3 GHz	3,3 GHz	3,3 GHz
RAM	16 GB	24 GB	32 GB
OS	Windows 10	Windows 10	Windows 10
Matlab	R2023a	R2021a	R2024a

Tabela 3: 10 sekundni cikel

Računalnik	1	2	3
Končni čas	10,195 s	10,153 s	10,142 s
Št. Izvedenih opravil	54	56	54
Čas procesiranja	68 ms	26 ms	24 ms
Čas procesiranja/opravilo	1,3 ms	0,464 ms	0,429 ms

Tabela 4: 600 sekundni cikel

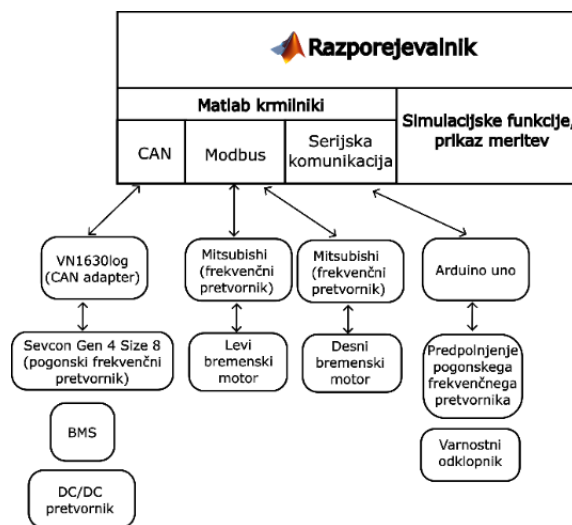
Računalnik	1	2	3
Končni čas	600,425 s	600,298 s	600,297 s
Št. Izvedenih opravil	3155	3247	3253
Čas procesiranja	1113 ms	629 ms	493 ms
Čas procesiranja/opravilo	0,352 ms	0,194 ms	0,151 ms

3 Primer delovanja razporejevalnika na pogonskem sklopu

Eden izmed ciljev, ki so bili zadani na začetku prispevka je enostavna uporaba razporejevalnika v aplikacijah, kot je na primer krmiljenje merilnega mesta pogonskega sklopa L7 električnega vozila. V ta namen je bilo potrebno razporejevalnik nadgraditi tako z možnostjo interakcije z merilnimi sistemi (NI DAQ, Agilent merilne

naprave, ...), viri moči kot so laboratorijski napajalniki, frekvenčnimi pretvorniki (nastavljanje želene hitrosti/navora), komunikacijskimi adapterji (CAN adapter, pisanje in branje) kot tudi z možnostjo izvajanja poljubnih Matlab funkcij (izračun modela vozila, termični model motorja, izris krivulj na grafični vmesnik, ...).

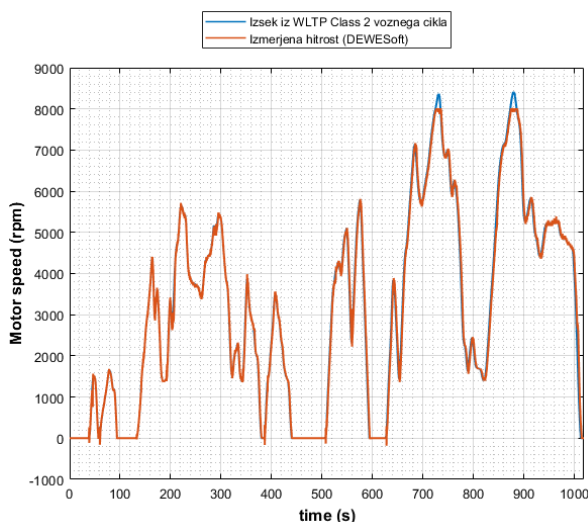
Razporejevalnik opravi skupaj s prevajalnikom za opravila je bil dalje razširjen v aplikacijo za namen meritev zmogljivosti pogonskega sklopa izdelanega za električno vozilo razreda L7. V predstavljenem primeru so se izvajale meritve zmogljivosti baterije in motorja na pogonskem sklopu v enem standardnem voznem ciklu za L7 vozilo (WLTP class 2). Za ta namen je bilo potrebno komunicirati (slika 3/Slika 3) s pogonskim frekvenčnim pretvornikom za zahtevo bremenskega navora na gredi pogonskega motorja; z BMS, za branje baterijske napetosti in toka; z DC/DC 350/12 V pretvornikom za napajanje logičnega dela komponent; z dvema Mitsubishijevima 4 kvadrantnima frekvenčnima pretvornikoma za nastavljanje trenutne hitrosti vozila; z Arduino razvojno ploščo za varno polnjenje kondenzatorjev pogonskega frekvenčnega pretvornika in krmiljenje varnostnega odklopnika v primeru napake. V razporejevalniku so bili uporabljeni že obstoječi Matlabovi krmilniki za CAN, Modbus in serijsko komunikacijo.



Slika 3: Blokovna shema aplikacije

Dodatno so bile z razporejevalnikom realizirane še Matlabove funkcije. Te so bile uporabljene za odčitavanje trenutne hitrosti in trenutnega navora na gredi pogonskega motorja iz pred pripravljenih tabel. Na enak način, z Matlab funkcijami, se je s kritičnimi informacijami pogonskega sistema (temperatura motorja, temperatura pogonskega frekvenčnega pretvornika, temperatura baterije, napetost baterije, itd.) posodabljal uporabniški vmesnik. Meritve hitrosti pogonskega motorja, navora, baterijskega toka, baterijske napetosti smo opravljali z DEWESoft merilno opremo. Za prikaz delovanja razporejevalnika je na sliki 4 prikazan izsek hitrosti iz WLTP Class 2 voznega cikla preračunan na hitrost motorja (v rpm) in eksperimentalno izmerjena

hitrost motorja z DEWESoft programom. Natančnejša analiza meritev ni tema prispevka.



Slika 4: Primerjava izseka voznega cikla WLTP Class 2 preračunane hitrosti in eksperimentalno izmerjena hitrost motorja z DEWESoft merilno opremo.

4 Zaključek

Izdelan razporejevalnik v kombinaciji s prevajalnikom opravil dosega cilje, ki so bili zadani na začetku. Z dodanim prevajalnikom uporabljenim za namen minimizacije časa procesiranja razporejevalnika smo dosegli do 0,151 ms (tabela 1) na opravilo. Ta čas in skupni čas razporejanja za vsa opravila sta odvisna od zmogljivosti osebnega računalnika. Večje število manj zahtevnih opravil omogoča večjo frekvenco izvajanja pomembnejših opravil ampak hkrati tudi večji delež celotnega časa, ki je porabljen za razporejanje opravil. Ravno obratno pa lahko celotni čas razporejanja opravil zmanjšamo z združevanjem manjših opravil, kar pa omeji največjo frekvenco za izvajanje pomembnejših opravil. S 1000 sekundnim vodenjem smo pokazali kvalitetno in stabilno delovanje razporejevalnika na kompleksnem krmilnem sistemu, kot je merilno mesto pogonskega sklopa L7 električnega vozila.

Z izdelanim programom smo se uspešno izognili uporabi dražjih in za pripravo vodenja časovno potratnejših krmilnih naprav. Razporejevalnik se lahko uporablja praktično v vseh krmilnih sistemih, kjer je potrebno krmiljenje ali branje iz ene ali več zunanjih naprav. Za delo potrebujemo le osebni računalnik in osnovno Matlab licenco, ki je na UL prosto dostopna splošnemu uporabniku pa tudi cenovno dosegljiva [13]). Na sliki 4 lahko na koncu WLTP cikla opazimo odstopanje merjene hitrosti (DEWESoft merilna oprema, oranžna) od zelene (modra). Razlog je integracijska časovna napaka razporejevalnika, ki jo bomo v prihodnje odpravili. Po ciklu 1000 sekund podana hitrost z razporejevalnikom zamuja za trenutno zeleno hitrostjo WLTP cikla za 1,9 sekunde, kar je razvidno iz meritev. Kljub tej napaki pa so meritve pravilne s časovno natančnostjo kot jo določa merilna oprema. Za

izboljšanje uporabniške izkušnje bomo v nadaljevanju pripravili možnost vnosa opravil prek uporabniškega vmesnika.

V nadaljnjem delu je tudi smiselno razmisliti kako še povečati zmogljivost, robustnost in zanesljivost razporejevalnika opravil. Za povečanje zmogljivosti se ponuja možnost paralelizacije opravil, s čimer bi zmanjšali periodo izvajanja opravil. Robustnost in zanesljivost pa lahko povečamo, če izdelan razporejevalnik opravil uporabljamo na PC v kombinaciji z operacijskim sistemom, ki poleg razporejevalnika izvaja kar najmanj dodatnih opravil.

Zahvala

Raziskavo je finančno podprla Javna agencija za znanstveno-raziskovalno in inovacijsko dejavnost Republike Slovenije (ARIS šifra L2-50084).

Literatura

- [1] dSpace, „Hardware-in-the-loop Testing,“ [Elektronski]. Dostop: <https://www.dspace.com/en/pub/home/applicationfields/foo/hil-testing.cfm>. [Poskus dostopa 1. 7. 2024].
- [2] speedgoat, „Products & Services,“ [Elektronski]. Dostop: <https://www.speedgoat.com/products-services>. [Poskus dostopa 1. 7. 2024].
- [3] Siemens, „Controller,“ Siemens, [Elektronski]. Dostop: <https://www.siemens.com/global/en/products/automation/systems/industrial/plc.html>. [Poskus dostopa 1. 7. 2024].
- [4] LS Electric, „PLC,“ [Elektronski]. Dostop: https://www.ls-electric.com/products/category/Smart_Automation_Solution/PLC. [Poskus dostopa 1. 7. 2024].
- [5] Omron, „Programmable Logic Controllers,“ [Elektronski]. Dostop: <https://industrial.omron.eu/en/products/programmable-logic-controllers>. [Poskus dostopa 1. 7. 2024].
- [6] Weidmuller, „PLC modules,“ [Elektronski]. Dostop: https://www.weidmuller.com/int/products/connectivity/plc_interf_aces_and_migration_solutions/index.jsp#/Segments/prod186196661062730. [Poskus dostopa 1. 7. 2024].
- [7] Arduino, „Hardware,“ [Elektronski]. Dostop: <https://www.arduino.cc/en/hardware>. [Poskus dostopa 1. 7. 2024].
- [8] infineon, „AURIX Microcontroller Board,“ [Elektronski]. Dostop: <https://www.infineon.com/cms/en/product/promopages/AURIX-microcontroller-boards/>. [Poskus dostopa 1. 7. 2024].
- [9] Texas Instruments, „C2000 real-time microcontrollers,“ [Elektronski]. Dostop: <https://www.ti.com/design-development/embedded-development/c2000-mcus.html>. [Poskus dostopa 1. 7. 2024].
- [10] Raspberry PI, „Products,“ [Elektronski]. Dostop: <https://www.raspberrypi.com/products/>. [Poskus dostopa 1. 7. 2024].
- [11] Mathworks, „Simulink Real-Time,“ [Elektronski]. Dostop: <https://www.mathworks.com/help/srealtime/ref/simulinkrealttime-app.html>. [Poskus dostopa 1. 7. 2024].
- [12] Mathworks, „Improvements to tic and toc Functions for Measuring Absolute Elapsed Time Performance in MATLAB,“ [Elektronski]. Dostop: <https://www.mathworks.com/company/technical-articles/improvements-to-tic-and-toc-functions-for-measuring-absolute-elapsed-time-performance-in-matlab.html>. [Poskus dostopa 1. 7. 2024].
- [13] Mathworks, „Matlab Pricing,“ [Elektronski]. Dostop: <https://www.mathworks.com/pricing-licensing.html>. [Poskus dostopa 1. 7. 2024].